CAP 6610 Homework 2

Siddhant Mittal (6061-8545)

1. Given two parallel hyperplanes h1 and h2:

$$h_1 : a^T x_1 = b_1$$
$$h_2 : a^T x_2 = b_2$$

Goal is to find the distance between h_1 and h_2.

If we write a equation of a line L which is passing through x_1 and is normal to hyperplane h_1 we will get

$$L : x_1 + at; t \in R$$

L will intersect h_2 normally as well at some point x_2. The distance $||x_2 - x_1||$ is the distance between h_1 and h_2.

Finding x2

$$a^T x_2 = b_2$$
$$a^T(x_1 + at) = b_2$$
$$a^T x_1 + a^T at = b_2$$
$$a^T at = b_2 - a^T x_2$$
$$t = \frac{b_2 - a^T x_2}{a^T a}$$

$a^T x_1 = b_1$

$$t = \frac{b_2 - b_1}{a^T a}$$

Hence

$$x_2 = x_1 + \frac{(b_2 - b_1)a}{a^T a}$$

$$||x_2 - x_1|| = \frac{||b_2 - b_1||.||a||}{a^T a}$$

Distance between the two parallel hyperplanes is $\frac{||b_2-b_1||}{||a||}$ as $a^T a = ||a||^2$

2. (a) Expected value is given by

$$E[x] = \sum_{i=1}^{n=k} a_i Pr[a_i]$$

It can be seen that expected value is a linear function hence it can be both convex and concave

(b) Pr[x > α] and (c) Pr[α < x < β]
Both are linear function so both convex and concave.

(c) Entropy of this distribution can be defined as:

$$-\sum_{i=1}^{n=k} p_i \log p_i$$

Showing convexity using second derivative test. Consider the function $f(x) = x \log x x > 0$. Then,

$$f(x) = \log x + 1$$

$$f(x) = \frac{1}{x} > 0$$

Thus, f(x) is a convex function. Now we know summation preserve convexity hence $\sum_{i=1}^{n=k} p_i \log p_i$ is convex. Now using the property negative of a convex function is concave we can say that entropy of this distribution is concave.

(b) var(x) is a concave function which can be proved in the following way.

$$var_p(x) = E_p[x^2] - E_p[x]^2$$

Hence for any $\theta \in$ [0,1] we can write

$$E_{\theta p+(1-\theta)q}[x^2] = \theta E_p[x^2] + (1-\theta)E_q[x^2]$$

Convexity of square function and Jensen's inequality $\phi(E[x]) \geq E(\phi X)$ then yields

$$\theta E_p[x]^2 + (1-\theta)E_q[x]^2 \geq (\theta E_p[x] + (1-\theta)E_q[x])^2$$

Right hand side is nothing but $E_{\theta p+(1-\theta)q}[x]^2$ Thus we deduce that

$$var_{\theta p+(1-\theta)}(x) = E_{\theta p+(1-\theta)q}[x^2] - E_{\theta p+(1-\theta)q}[x]^2 \geq \theta var_p(x) + (1-\theta)var_q(x)$$

Which satisfy the property of a concave function hence variance is concave.

3. Two optimization models are equivalent if the solution to one quickly leads to a solution to the other, and vice versa. Some common transformations that preserve convexity is the core idea behind equivalency between two optimization problem.

   Let's start fixing the quadratic program optimization problem. Note below statement are equivalent.

$$-u_i - v_i \leq \phi_i^T\theta - \psi_i \leq u_i + v_i$$
$$|\phi_i^T\theta - \psi_i| \leq v_i + u_i$$

   The minimization is achieved at a optimum value of $u_i + v_i = |\phi_i^T\theta - \psi_i|$. This is because otherwise objective will change as we can go much lower as u_i and v_i are both greater than zero.

   At optimum value $v_i = |\phi_i^T\theta - \psi_i| - u_i$, we can substitute this variable in our objective function to eliminate a variable preserving the convexity. Now the new objective function is

   minimize:

$$\sum_{i=1}^{m} u_i^2 - 2Mu_i + 2M|\phi_i^T\theta - \psi_i|$$

   subject to:

$$0 \leq u_i \leq \min(M, |\phi_i^T\theta - \psi_i|)$$

   Now there are two case if $|\phi_i^T\theta - \psi_i|$ is less than equal to M or greater than M.

   if $|\phi_i^T\theta - \psi_i| \leq M$ then objective function is $|\phi_i^T\theta - \psi_i|^2$.

   if $|\phi_i^T\theta - \psi_i| > MM$ then objective function is $2M|\phi_i^T\theta - \psi_i| - M^2$

   Let's call the objective function as t then optimal value of this QP problem for a fixed x is given by

$$\sum_{i=1}^{m}(|\phi_i^T\theta - \psi_i|)$$

Which is equivalent to solving robust least square problem of (a).

4. Linear model is constructed in the following way:

$$y_i = x_i^T\theta + \beta$$

where,
y_i = value of quality of red wine
x_i = vector of features of the given wine
θ = coefficients of unknowns.
β = scalar-valued bias.

This can also be represented as

$$y_i = \theta_1 * x_i 1 + \theta_2 * x_i 2 + ... + \theta_1 1 * x_i 11 + \beta$$

To make the model more easy to use adding a 1 value feature vector to x_i thus making $\beta = \theta_0$ So the problem reduces to

$$x_i^T\theta = y_i$$

Now to make linear regression model we use three type of loss functions; least squares, huber penalty and hinge loss. Their implementation can be seen in the code.

Below table shows mean absolution error(MAE) for three regression models.

Model	MAE
Least square loss	0.53297
Huber loss	0.53272
Hinge loss	0.54811

5. We have to create a binary classifier for which good('g') is +1 and bad('b') is -1. Binary linear classifier work is simple: they compute a linear function of the inputs, and determine whether or not the value is larger than some threshold r.

$$z_i = \theta_1 * x_i 1 + \theta_2 * x_i 2 + ... + \theta_1 1 * x_i 34 + \beta$$

where,
z_i = value of quality
x_i = vector of features
θ = coefficients of unknowns.
β = scalar-valued bias.
Therefore, the prediction y_i can be computed as follows:

$$z = x_i^T\theta + \beta$$

$$y_i = \begin{cases} 1 & z \geq r \\ -1 & z < r \end{cases}$$

We can obtain an equivalent model by replacing the bias with b - r and setting r to 0. From now on, we'll assume (without loss of generality) that the threshold is 0. In fact, it's possible to eliminate the bias as well. We simply add another input dimension x_0, called a dummy feature, which always takes the value 1. Therefore our model becomes

$$z = x_i^T\theta$$

$$y_i = \begin{cases} 1 & z \geq 0 \\ -1 & z < 0 \end{cases}$$

We use this model with three loss function to create a binary linear classifier who prediction accuracy in percentage is show in below table.

Model	Prediction Accuracy(%)
Least square loss	100
Logistic loss	100
Hinge loss	100

Proof of concept: The explanation of 100 percentage result. The last 51 records used for testing contains all good values that is 'g', if we have randomly separated training and testing then prediction accuracy varies from 90-100 percent which I checked but since question specify the separating condition for training and testing the result are shown accordingly.