1. CSE 6740: HW1

Note: This assignment is submitted by: **Somdut Roy (GTID: sroy86)**

1. **MAP Estimation**

   (a) $D = (x_1, x_2, ..., x_n) \in \mathbb{R} \sim \mathcal{N}(\mu, \sigma^2)$ given that $\mu \sim \mathcal{N}(m, s^2)$.
   $\hat{\mu}_{MAP} = \text{argmax}_\mu \; p(\mu|D) = \text{argmax}_\mu \; \frac{p(D|\mu)p(\mu)}{p(D)}$
   $\Rightarrow 0 = \frac{\delta}{\delta\mu}(log(p(D|\mu)) + log(p(\mu)))$.....(1)
   $\frac{\delta}{\delta\mu} log(p(D|\mu)) = \frac{\delta}{\delta\mu} \sum_{i=1}^{n} \frac{-(x_i-\mu)^2}{2\sigma^2} = \frac{1}{\sigma^2}(\sum_{i=1}^{n} x_i - n\mu)$.....(2)
   $\frac{\delta}{\delta\mu} log(p(\mu)) = \frac{\delta}{\delta\mu} \frac{-(\mu-m)^2}{2s^2} = \frac{(m-\mu)}{s^2}$.....(3)

   Putting derivations from (2) and (3) in (1):
   $\frac{1}{\sigma^2}(\sum_{i=1}^{n} x_i - n\mu) + \frac{(m-\mu)}{s^2} = 0$
   $\Rightarrow \hat{\mu}_{MAP} = \frac{s^2 \sum_{i=1}^{n} x_i + \sigma^2 m}{s^2 n + \sigma^2}$.....(4)

   (b) From (4), $\hat{\mu}_{MAP} = \frac{\sum_{i=1}^{n} x_i}{n + \frac{\sigma^2}{s^2}} + \frac{\sigma^2 m}{s^2 n + \sigma^2}$
   $\Rightarrow \lim_{n\to\infty} \hat{\mu}_{MAP} = \frac{\sum_{i=1}^{n} x_i}{n}$ = sample mean.
   Therefore it is evident that with increasing sample size, MAP estimate converges to MLE.

   (c) From expression in (4), clearly $\lim_{s^2\to\infty} \hat{\mu}_{MAP} = \frac{\sum_{i=1}^{n} x_i}{n}$ = sample mean.

   (d) From expression in (4), clearly $\lim_{s^2\to 0} \hat{\mu}_{MAP} = \frac{\sigma^2 m}{\sigma^2} = m$ = mean of distribution of unknown mean of Gaussian distribution of sample.

2. **Completing Squares**
   $X_1 \sim \mathcal{N}(x|\mu_1, \Sigma_1)$, $X_2 \sim \mathcal{N}(x|\mu_2, \Sigma_2)$. Let $Z = X_1 + X_2$.

$$f_Z(z) = \int_{-\infty}^{\infty} f_{X_2}(Z - X_1) f_{X_1}(x_1) dx_1$$
$$= \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi\Sigma_2}} \exp(-\frac{(z-x_1-\mu_2)^2}{2\Sigma_2}) \frac{1}{\sqrt{2\pi\Sigma_1}} \exp(-\frac{(x_1-\mu_1)^2}{2\Sigma_1}) dx_1$$
$$= \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi 2\pi\Sigma_1\Sigma_2}} \exp(-\frac{x_1^2(\Sigma_1+\Sigma_2) - 2x_1(\Sigma_1(z-\mu_2)+\Sigma_2\mu_1) + \sigma_1(z^2+\mu_2^2-2z\mu_2)+\Sigma_2\mu_1^2}{2\Sigma_1\Sigma_2}) dx_1$$

We define $\Sigma_Z = \Sigma_1 + \Sigma_2$ and use "completing the squares" technique:

$$f_Z(z) = \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi\Sigma_Z}} \frac{1}{\sqrt{2\pi\frac{\Sigma_1\Sigma_2}{\Sigma_Z}}} \exp(-\frac{x_1^2 - 2x_1\frac{\Sigma_1(z-\mu_1)+\Sigma_2\mu_1}{\Sigma_Z} + \frac{\Sigma_1(z^2+\mu_2^2-2z\mu_2)+\Sigma_2\mu_1^2}{\Sigma_Z}}{\frac{2\Sigma_1\Sigma_2}{\Sigma_Z}}) dx_1$$
$$= \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi\Sigma_Z}} \frac{1}{\sqrt{2\pi\frac{\Sigma_1\Sigma_2}{\Sigma_Z}}} \exp(-\frac{(x_1 - \frac{\Sigma_1(z-\mu_2)+\Sigma_2\mu_1}{\Sigma_Z})^2 - (\frac{\Sigma_1(z-\mu_2)+\Sigma_2\mu_1}{\Sigma_Z})^2 + \frac{\Sigma_1(z-\mu_2)^2+\Sigma_2\mu_1^2}{\Sigma_Z}}{\frac{2\Sigma_1\Sigma_2}{\Sigma_Z}}) dx_1$$
$$= \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi\Sigma_Z}} \exp(-\frac{\Sigma_Z(\Sigma_1(z-\mu_2)^2+\Sigma_2\mu_1^2) - (\Sigma_1(z-\mu_2)+\Sigma_2\mu_1)^2}{2\Sigma_Z\Sigma_1\Sigma_2}) \frac{1}{\sqrt{2\pi\frac{\Sigma_1\Sigma_2}{\Sigma_Z}}} \exp(-\frac{(x_1 - \frac{\Sigma_1(z-\mu_2)+\Sigma_2\mu_1}{\Sigma_Z})^2}{2\frac{\Sigma_1\Sigma_2}{\Sigma_Z}}) dx_1$$
$$= \frac{1}{\sqrt{2\pi\Sigma_Z}} \exp(-\frac{(z-(\mu_1+\mu_2))^2}{2\Sigma_Z}) \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi\frac{\Sigma_1\Sigma_2}{\Sigma_Z}}} \exp(-\frac{(x_1 - \frac{\Sigma_1(z-\mu_2)+\Sigma_2\mu_1}{\Sigma_Z})^2}{2\frac{\Sigma_1\Sigma_2}{\Sigma_Z}}) dx_1$$
$$= \frac{1}{\sqrt{2\pi\Sigma_Z}} \exp(-\frac{(z-(\mu_1+\mu_2))^2}{2\Sigma_Z}) \int_{-\infty}^{\infty} \mathcal{N}(x_1 | \frac{\Sigma_1(z-\mu_2)+\Sigma_2\mu_1}{\Sigma_Z}, \frac{\Sigma_1\Sigma_2}{\Sigma_Z}) dx_1$$
$$= \frac{1}{\sqrt{2\pi\Sigma_Z}} \exp(-\frac{(z-(\mu_1+\mu_2))^2}{2\Sigma_Z})$$
$$= \mathcal{N}(z | (\mu_1+\mu_2), \Sigma_Z)$$

Therefore $(X_1 + X_2) \sim \mathcal{N}(\mu_1 + \mu_2, \Sigma_1 + \Sigma_2)$.

3. Bayesian Linear Regression

(a) We calculate $\mathbf{w}^T = [\hat{w_0}\ \hat{w_1}] \approx [-3.25643\ 0.04265]$. Using equation 7.107 given in the problem, the unbiased estimate of σ^2, $\hat{\sigma^2} \approx 0.01697$.

```
import numpy as np

#question 3, part (a)
x=[94,96,94,95,104,106,108,113,115,121,131]
y=[.47,.75,.83,.98,1.18,1.29,1.4,1.6,1.75,1.9,2.23]
cov_mat=np.cov(x,y)
w1=cov_mat[0][1]/cov_mat[0][0]
w0=np.average(y)-w1*np.average(x)
N=len(x)
print w0, w1
sigma_sq=np.sum([(y[i]-w1*x[i]-w0)**2 for i in range(N)])/(N-2)
print sigma_sq
```

(b) $p(w_0) \propto 1$ and $p(w_1) \sim \mathcal{N}(0,1)$.
w_0 can be equated to a Normal distribution with very large variance σ thereby nullifying the exponential term in the prior.
$\mathbf{w} = \begin{bmatrix} w_0 \\ w_1 \end{bmatrix} \sim \mathcal{N}(\mathbf{w}_0, \mathbf{V}_0)$, where $\mathbf{w}_0 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$ and $\mathbf{V}_0 = \begin{bmatrix} \sigma & 0 \\ 0 & 1 \end{bmatrix}$.

(c) Using equations 7.56 and 7.58 derived in 7.6.1 of textbook
$[\mathbf{w}_N\ \mathbf{V}_N] \approx [0.04262\ 1.03845\text{x}10^{-5}]$. (Given that $V_0 = 1$ for the figure

below).

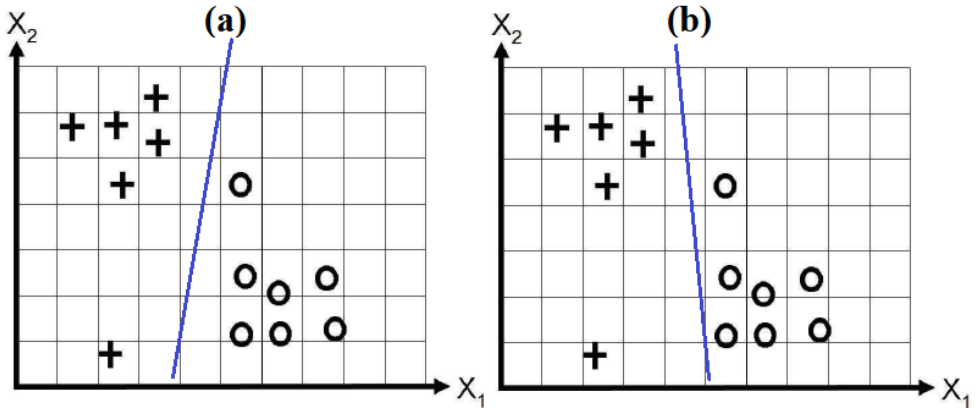
```
#question 3, part (c)
#using equations derived in 7.6.1 of textbook
w1_var=sigma_sq/(sigma_sq+N*cov_mat[0][0])
w1_mean=w1_var*w0+w1_var*cov_mat[0][1]*N/sigma_sq
print w1_mean, w1_var
```

(d) The 95 percent confidence interval for w_1 is given by $\mathbf{w}_N \pm 1.96\sqrt{\mathbf{V}_N}$. Therefore the confidence interval $\sim$ [0.03630 0.04893].

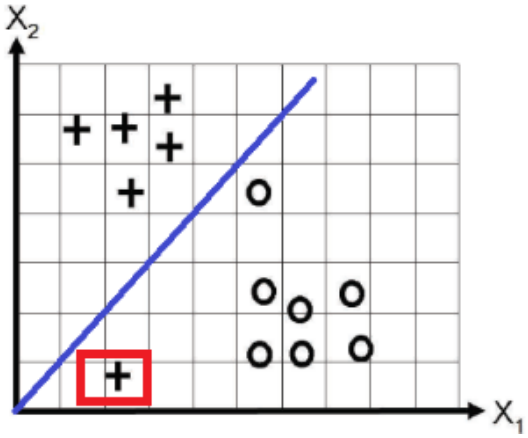
```
#question 3, part (d)
# +/- 1.96 for 95% confidence interval
low_bound, high_bound=w1_mean-1.96*w1_var**.5, w1_mean+1.96*w1_var**.5
print "confidence interval =["+str(low_bound)+', '+str(high_bound)+']'
```

4. **Regularization in Logistic Regression**

(a) Figure (a) (left) below depicts a line representing a decision boundary. Figure (b) gives another such option. **No classification errors** are seen in both cases. Also it is mention-worthy there can be **infinite number of possible lines** drawn like the ones below so as to classify the points correctly. Hence there is **no unique solution**.

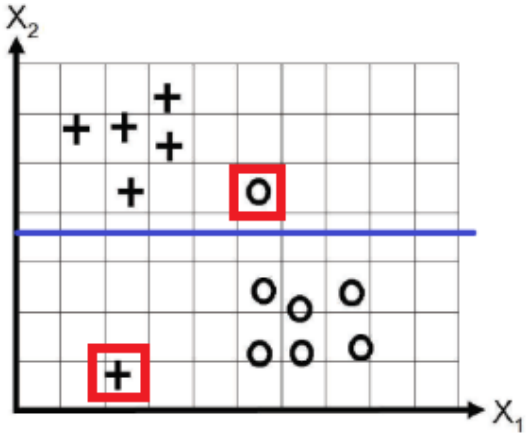


(b) A decision boundary for this case will be a line passing through the origin. The figure below represents one such line.



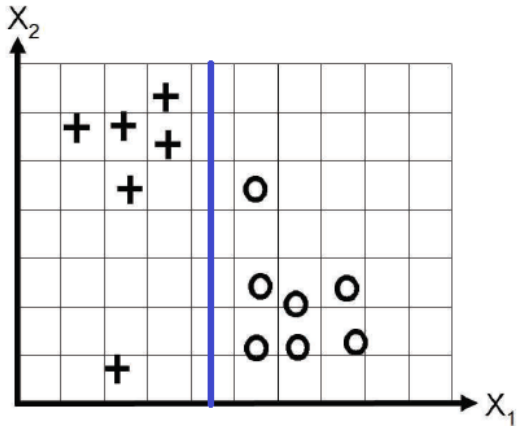
The point shown in red box is wrongly classified.

(c) A decision boundary for this case will be a line parallel to the $\mathbf{X}_1$ axis. The figure below represents one such line.



The point shown in red box is wrongly classified.

(d) A decision boundary for this case will be a line parallel to the $\mathbf{X}_2$ axis. The figure below represents one such line.



As evident from the figure above, there are no classification errors in this case.

5. **Residual Sum of Squares**

$RSS = ||\mathbf{Xw} - \mathbf{y}||_2^2 = \sum_{i=1}^{n}(\sum_{j=1}^{n} w_j x_{ij} - y_i)^2$

(a) $\frac{\delta}{\delta w_k} RSS = \sum_{i=1}^{n} 2(\sum_{j=1}^{n} w_j x_{ij} - y_i)x_{ik}$

$= w_k(\sum_{i=1}^{n} x_{ik}^2) - \sum_{i=1}^{n}(y_i x_{ik} - \sum_{\substack{j=0 \\ j\neq k}}^{n} w_j x_{ij} x_{ik})$

$= a_k w_k - c_k.$

Comparing terms with and without w_k,

$a_k = 2\sum_{i=1}^{n} x_{ik}^2 = 2||\mathbf{x}_{:,k}||^2$

$c_k = \sum_{i=1}^{n}(y_i x_{ik} - \sum_{\substack{j=0 \\ j\neq k}}^{n} w_j x_{ij} x_{ik})$

$= 2\sum_{i=1}^{n} x_{ik}(y_i - \mathbf{w}_{-k}^T \mathbf{x}_{i,-k}) = 2\mathbf{x}_{:,k}^T(\mathbf{y} - \mathbf{w}_{-k}^T \mathbf{X}_{:,-k}) = 2\mathbf{x}_{:,k}^T \mathbf{r}_k$

(b) $\frac{\delta}{\delta w_k} RSS = 0$

$\Rightarrow a_k \hat{w}_k - c_k = 0$

$\Rightarrow \hat{w_k} = \frac{c_k}{a_k}$

Putting values of a_k and c_k derived in part (a), we get,

$\hat{w_k} = \frac{2\mathbf{x}_{:,k}^T \mathbf{r}_k}{2||\mathbf{x}_{:,k}||^2} = \frac{\mathbf{x}_{:,k}^T \mathbf{r}_k}{||\mathbf{x}_{:,k}||^2}$.

Hence it is shown that as we sequentially add features, the optimal weight for feature k is computed by computing orthogonally projecting $x_{:,k}$ onto the current residual.

6. **Programming**
    1. **Ridge Regression**
        1. Follow *greedy.py* for this section.
        2. Average error for different values of $\lambda = \begin{bmatrix} 0.0125 & 0.025 & 0.05 & 0.1 & 0.2 \end{bmatrix}$ are $\begin{bmatrix} 5.94 & 5.95 & 6.00 & 6.20 & 6.88 \end{bmatrix}$. Therefore, optimal solution is attained for $\lambda = 0.0125$.
        3. The prediction error on "test-matrix.txt" for that model ≈ 7.22. $||\hat{\beta}^{Ridge} - \beta^*||_2^2 \approx 0.01454$.
    2. **Naive Bayes Classification**
        1. Fraction of test samples classified correctly=0.834862385321.
        2. Precision of class 1=0.951219512195.
        3. Recall of class 1=0.975.
        4. Precision of class 5=0.875.
        5. Recall of class 5=0.777777777778.

Bonus question

(a) One can use **Information Gain** for each feature to rank the features based on how much each of them is "relied upon". We reclassify the features with continuous attributes by taking the median as the split point, meaning any value below the median is re-branded as 0 and everything else as 1. The information gain in calculated $IG_Y = \sum_{y \in Y} P(Y) \sum_{x \in X} -P(X|Y) \log_2 P(X|Y)$.

(b) Using that principle, information gain was calculated for all the features and sorted in ascending order. So the classification relies on the features from less to more in the following order:

```
[('Feature', 'Information_Gain'),
 ('Feature_4', 1.4415509764774905),
 ('Feature_1', 1.5779529929113001),
 ('Feature_5', 1.6758399582738308),
 ('Feature_2', 1.690787857923655),
 ('Feature_3', 1.7261403740368739)]
```

Hence feature 4 is the least influential to the classifier and feature 3 is the most important.

(c) For the classifier to work, each class should have a respectable amount of representation in the training set data. There are more than 100 of class 1 in the data-set out of 218 while there are only 11 of class 5.

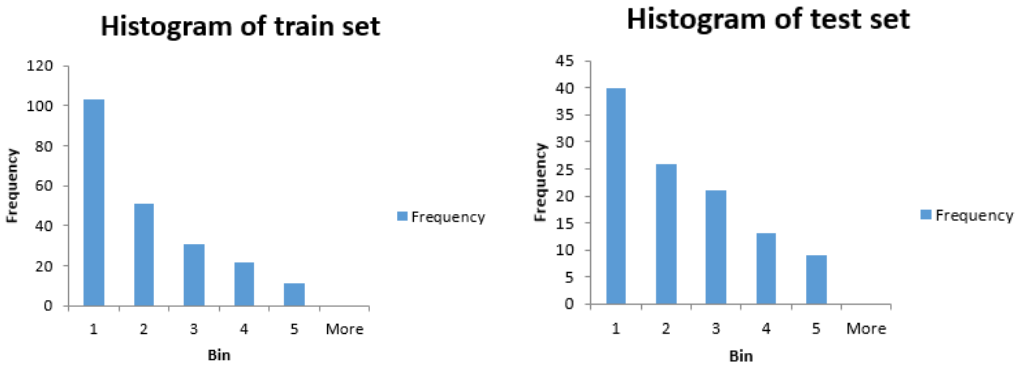


Figure: (from left) Histogram of classes in (a) train set and (b)test set

That is not enough information about class 5. Also the distribution of the classes in the train set and test set are not same. 47 percent of data in train set are classified 1 while 36 percent in train set are classified 1. This questions randomness of selecting training data-set and hence gives an indication of dataset bias. Also the median split point set in train set for feature 3 suggests that points in test set are skewed heavily towards less than median value.