CSE250B Name: Priyanka Dighe PID: A53090004

Homework 3

Problem1:

a. $\mu_x = 2, \ \Sigma_{xx} = 1, \mu_y = 2, \ \Sigma_{yy} = 0.25,$

$corr(x,y) = \frac{cov(x,y)}{std(x)std(y)}$

$=> \ -0.5 = \frac{cov(x,y)}{1*0.5}$

$=> cov(x,y) = \Sigma_{xy} = -0.25$

A distribution over (x,y) ϵ R² is parametrized by

- $Mean\ \mu = (\mu_x, \mu_y) = (2,2)\ \epsilon\ R^2$
- $Covariance\ matrix\ \Sigma = \begin{matrix} \Sigma_{xx} & \Sigma_{xy} \\ \Sigma_{yx} & \Sigma_{yy} \end{matrix} = \begin{matrix} 1 & -0.25 \\ -0.25 & 0.25 \end{matrix}$

b. $\mu_x = \mu_y = 1, \ \Sigma_{yy} = \Sigma_{xx} = 1 \ , since\ x\ is\ equal\ to\ y$

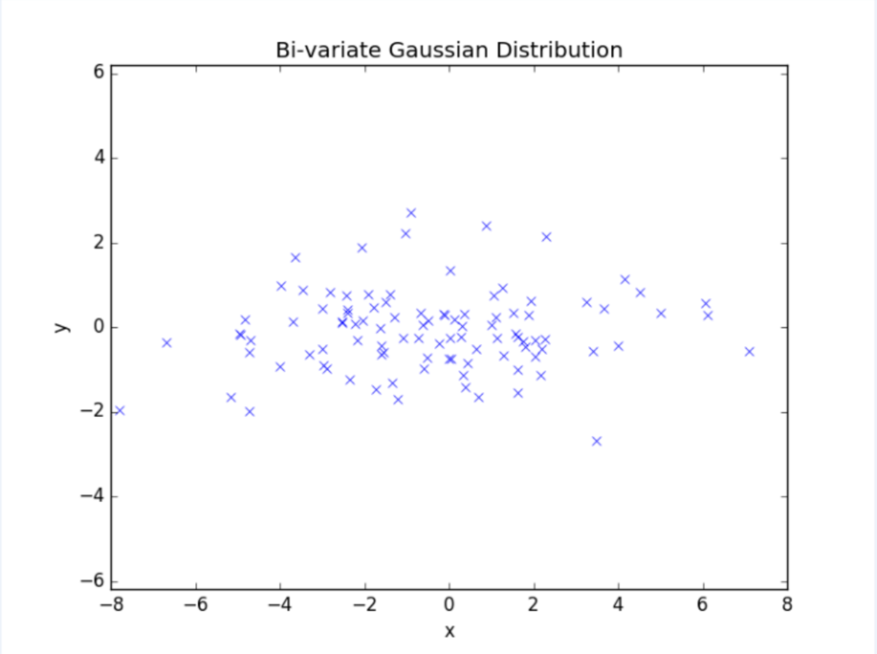
$Therefore, \Sigma_{xy} = \Sigma_{yx} = 1$

A distribution over (x,y) ϵ R² is parametrized by

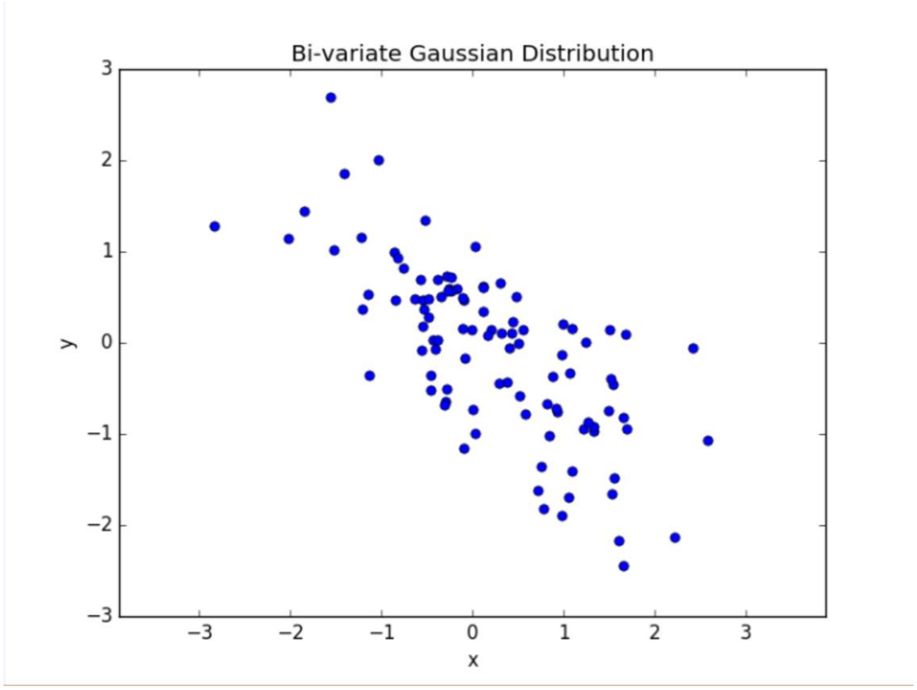
- $Mean\ \mu = (\mu_x, \mu_y) = (1,1)\ \epsilon\ R^2$
- $Covariance\ matrix\ \Sigma = \begin{matrix} \Sigma_{xx} & \Sigma_{xy} \\ \Sigma_{yx} & \Sigma_{yy} \end{matrix} = \begin{matrix} 1 & 1 \\ 1 & 1 \end{matrix}$

Problem2:

a. Graph of 100 random samples is as follows:

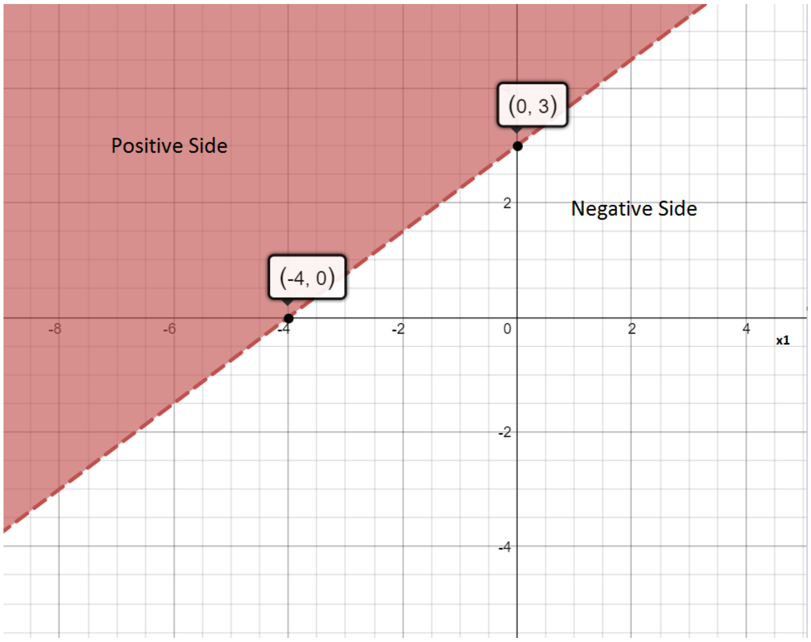


b. Graph of 100 random samples is as follows:



Problem3:

The decision boundary for linear classifier is as follows:
(The colored side is the poistive side)



Problem4:

a. ∑ is invertible if det(∑) != 0. W can write ∑ in terms of its eigen values and eigen vectors as follows:

$$\Sigma = \begin{pmatrix} \vdots & \vdots & \vdots \\ u_1 & \dots & u_p \\ \vdots & \vdots & \vdots \end{pmatrix} \begin{pmatrix} \lambda_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \lambda_p \end{pmatrix} \begin{pmatrix} \dots & u_1 & \dots \\ \dots & \vdots & \dots \\ \dots & u_p & \dots \end{pmatrix}$$

$Let\ \Sigma = U\,\Lambda\,U^T \rightarrow\ \det(\Sigma) = \det(U) * \det(\Lambda) * \det(U^T)$

$Since\ U\ and\ U^T\ are\ eigen\ vectors\ \det(U) = \det(U^T) = 1\ and\ \det(\Lambda) = \prod_{i=1}^{p} \lambda_i$
$Therefore\ we\ can\ say\ that\ \Sigma\ is\ invertible\ if\ \prod_{i=1}^{p} \lambda_i\ != 0$

In other words, $\sum$ is invertible if no λ_i has the value 0

b. For an identity matrix *I*, all eigen values $\lambda_i = 1$ and all vectors are eigen vectors of *I*.
Let $u_1, u_2 \dots u_p\ be\ the\ eigen\ vectors\ of\ I$
Therefor we can write *I* as:

$$\mathrm{I} = \begin{pmatrix} \vdots & \vdots & \vdots \\ u_1 & \dots & u_p \\ \vdots & \vdots & \vdots \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \dots & u_1 & \dots \\ \dots & \vdots & \dots \\ \dots & u_p & \dots \end{pmatrix}$$, Using spectral decomposition

Substituting the value of $\sum$ from part a in the following equation:

$$\Sigma + \mathrm{cI} = \begin{pmatrix} \vdots & \vdots & \vdots \\ u_1 & \dots & u_p \\ \vdots & \vdots & \vdots \end{pmatrix} \begin{pmatrix} \lambda_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \lambda_p \end{pmatrix} \begin{pmatrix} \dots & u_1 & \dots \\ \dots & \vdots & \dots \\ \dots & u_p & \dots \end{pmatrix} + \begin{pmatrix} \vdots & \vdots & \vdots \\ u_1 & \dots & u_p \\ \vdots & \vdots & \vdots \end{pmatrix} \begin{pmatrix} c & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & c \end{pmatrix} \begin{pmatrix} \dots & u_1 & \dots \\ \dots & \vdots & \dots \\ \dots & u_p & \dots \end{pmatrix}$$

$$\Sigma + \mathrm{cI} = \begin{pmatrix} \vdots & \vdots & \vdots \\ u_1 & \dots & u_p \\ \vdots & \vdots & \vdots \end{pmatrix} \left(\begin{pmatrix} \lambda_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \lambda_p \end{pmatrix} + \begin{pmatrix} c & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & c \end{pmatrix} \right) \begin{pmatrix} \dots & u_1 & \dots \\ \dots & \vdots & \dots \\ \dots & u_p & \dots \end{pmatrix}$$

$$\Sigma + \mathrm{cI} = \begin{pmatrix} \vdots & \vdots & \vdots \\ u_1 & \dots & u_p \\ \vdots & \vdots & \vdots \end{pmatrix} \begin{pmatrix} \lambda_1 + c & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \lambda_p + c \end{pmatrix} \begin{pmatrix} \dots & u_1 & \dots \\ \dots & \vdots & \dots \\ \dots & u_p & \dots \end{pmatrix}$$

$\boldsymbol{Therefore, the\ eigen\ values\ of\ \Sigma + \mathbf{cI}\ \mathbf{are}\ (\lambda_1 + c, \lambda_2 + c, \dots . \lambda_p + c)}$

c. From part a $\sum$ can be written as:

$$\Sigma = \begin{pmatrix} \vdots & \vdots & \vdots \\ u_1 & \dots & u_p \\ \vdots & \vdots & \vdots \end{pmatrix} \begin{pmatrix} \lambda_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \lambda_p \end{pmatrix} \begin{pmatrix} \dots & u_1 & \dots \\ \dots & \vdots & \dots \\ \dots & u_p & \dots \end{pmatrix}$$

$Let\ \Sigma = U\,\Lambda\,U^T,\ \ Multplying\ RHS\ by\ U\,\Lambda^{-1}\,U^T\ , we\ get$
$RHS = (U\,\Lambda\,U^T)(U\,\Lambda^{-1}\,U^T) = \ (U\,\Lambda\,(U^T U)\,\Lambda^{-1}\,U^T\ ) = \ \ U\,(\Lambda\,\Lambda^{-1})U^T = UU^T = \mathrm{I}$
$Therefore, U\,\Lambda^{-1}\,U^T\ is\ the\ inverse\ of\ \Sigma$
$Since\ \Lambda$ is a diagonal vector given by,

$$\Lambda = \begin{pmatrix} \lambda_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \lambda_p \end{pmatrix}$$, we can write $\Lambda^{-1}\ as$:

$$\Lambda^{-1} = \begin{pmatrix} 1/\lambda_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & 1/\lambda_p \end{pmatrix}$$

Therefore, the eigen values of Σ^{-1} are $1/\lambda_1, 1/\lambda_p, \ldots\ldots 1/\lambda_p$

Problem5:

Problem 5.a:

Pseudocode:

```
// Load test data(ts) from MNIST
ts_dataset(ts_images, ts_labels) = loadMNIST(testing)

// For each digit, (0-9) load the training data labelled with that digit
for digit in range(10):
        train_dataset(train_images, train_labels) = loadMNIST(training, digit)
        // Shuffle the tr_dataset randomly

        // split training data into 80:20 ratio for validation
        Split  = 0.8*(len(train_dataset))
        val_dataset(v_images, v_labels) = train_dataset[split:]     // Used as validation dataset
        tr_dataset(tr_images, tr_labels) = train_dataset[:split]    // This is used as training dataset

        // Calculate the gaussian parameters for each digit for data points(images) $x^{(1)}, x^{(2)}, \ldots x^{(m)}$
        $\mu[digit] = \frac{1}{m}\sum x_i$        // each μ[digit] is a  784 * 1 matrix
        $SIGMA[digit] = cI + \frac{1}{m}\sum_1^m \left(x^{(i)} - \mu[digit]\right)\left(x^{(i)} - \mu[digit]\right)^T$   // each SIGMA[digit] is a 784 * 784 matrix

// We have now modelled the Gaussian parameters using dataset.
// Test the validation set using the Gaussion formula for classification

$px[i] = P(Y = i|x) = \frac{P(x|Y = i).P(Y=i)}{P(x)}$
label(x) = argmax(px)

// The value of c is set by performing several experiments on val_dataset to get the most minimum error
```

Error rate on MNIST test set for different values of c is :

C value	Error rate in (%) in validation set
1	15.84
100	8.06
1000	4.89
3100	4.57
3225	4.61
3250	4.79
3500	4.74

Minimum error on validation set was seen at **c = 3100**

Problem 5.b

Error rate on MNIST test set using c = 3100 : **4.32%**

Problem 5.c: Posterior probabilities of 5 randomly selected mismatches are:

Sr. No	Sample Label	Classified Label	Posterior probabilities
1.	9	3	1.26923757e-033 9.39887850e-147 7.38987780e-017 1.00000000e+000 1.34477714e-054 4.37402582e-035 9.44827285e-092 2.73801918e-046 1.07374439e-024 2.07212828e-021
2.	3	8	9.86855545e-42 1.05798638e-50 2.14351703e-13 9.63094897e-02 1.87238369e-29 3.55870957e-19 1.81962332e-57 5.09742305e-45 9.03690510e-01 5.87606089e-20
3.	3	8	2.96495268e-34 8.24909698e-35 3.22842037e-24 3.34855832e-01 4.52169761e-39 5.73817143e-17 1.51039983e-60 2.73044558e-55 6.65144168e-01 5.37296381e-26
4.	3	8	2.38493412e-30 6.27259641e-09 2.38303545e-15 1.01444645e-01 8.18269897e-18 8.97980455e-14 1.30681358e-36 1.16017269e-27 8.98555349e-01 6.76975341e-14
5.	7	1	2.90172383e-68 1.00000000e+00 1.18761259e-42 1.22048359e-42 1.42921336e-29 8.27463066e-50 7.30369891e-62 5.13803037e-28 2.03627629e-33 6.12381632e-35

Problem 6:

Problem 6.a: Strategy

The problem statement is to design a model which is allowed to abstain a fraction of *f* instances. The idea is learn to a threshold (t) from validation data which will be used on the test data to abstain from prediction. We assume that the validation data and the test data come from the same underlying distribution.

Normally, we use the following formula for prediction for each data point:

$$px[i] = P(Y = i|x) = \log\left(\frac{P(x|Y = i).P(Y=i)}{P(x)}\right)$$

```
label(x) = argmax(px)
```

The model should abstain from prediction if it is not "certain" of its prediction. We measure this certainty using the difference between the top two probabilities in the array px. If the difference between the top two probabilities (px[i], px[j]) is less than a certain threshold (t), it means that the two labels have similar probabilities and hence the model should abstain from prediction. If the difference between 2 max probabilities is high, it means that the model is more certain of its classification. Thus, the classification rule can be written as:

$$h(x) = \begin{Bmatrix} i \text{ } if \text{ } px[i] - px[j] < t \\ abstain \text{ } otherwise \end{Bmatrix}$$

Here we assume that px[i] and px[j] are the largest and second largest probabilities respectively.

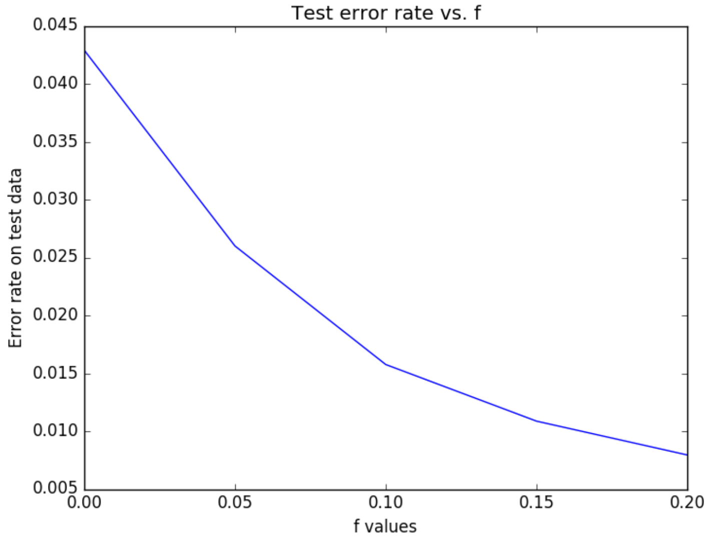
We use the validation data to find the threshold value (t) for any given value of f.

Problem 6.b: Pseudocode

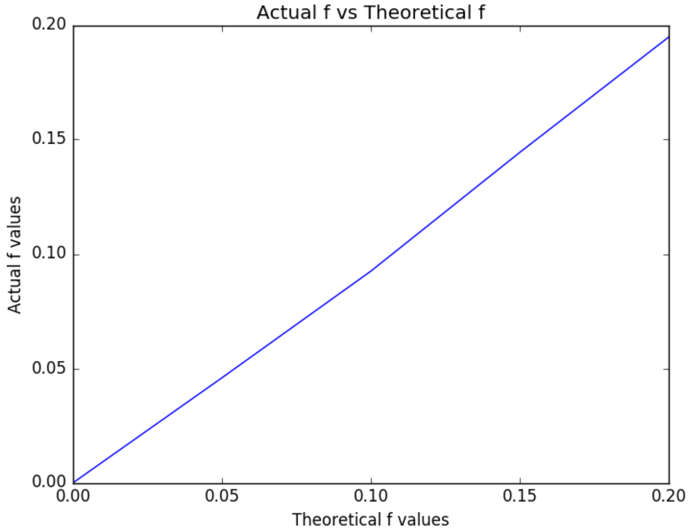
```
(i)    Divide the training set randomly in the ratio 80:20 (80% training data, 20%
       validation data)
(ii)   Model the Gaussian parameters μ, ∑ using the training data (as in problem 5)
(iii)  Use the validation data to model the parameter c which gives the minimum error
       rate (refer to problem 5). Here the optimal value of c is found to be 3100.
(iv)   We now need to find a threshold value (t) for a given value of f
(v)    For each prediction in the validation set:
           a. calculate px as follows:
              $px[i] = P(Y = i|x) = \log\left(\frac{P(x|Y = i).P(Y=i)}{P(x)}\right)$ for i = 0,1,2…9
           b. Find top 2 maximum probabilities in the array px. Let them be max1 and
              max2 respectively.
           c. Calculate the difference max1-max2 and save it in an array diff.
(vi)   Sort the array diff in a descending order.
(vii)  Given an f, split the array at index: split_index = int((1-f)*len(diff))
(viii)     threshold = diff[split_index]
(ix)   Classify the test data using the following classifier:
              $h(x) = \begin{Bmatrix} i \text{ } if \text{ } px[i] - px[j] < t \\ abstain \text{ } otherwise \end{Bmatrix}$
       Here we assume that px[i] and px[j] are the largest and second largest
       probabilities respectively.
(x)    Calculate the error rate of the test data using this classifier.
```

Problem 6.c:

(i) Graph of test error rate (on points actually classified) versus f:



(ii) Fraction of data on which classifier abstains versus f


Actual f vs Theoretical f
Actual f values
Theoretical f values