CS 584 – Machine Learning

Assignment -2 Report

1) 1-D 2 class Gaussian Determinant Analysis

a) Data set used: Iris.csv. Here I selected only two classes namely (Iris – Virginica and Iris-Versicolor) as the two classes and the first feature out of the four (sepal-length) as my feature.

b) Model parameters:

Since we assume the distribution to be Gaussian and the features to be 1-D we have the parameters 'μ' and 'σ' for the respective classes.

Prior probability of the classes:

```
[('Iris-versicolor', 0.5), ('Iris-virginica', 0.5)]
```

Mean of the features (μ):

```
[sepal_length    5.936
dtype: float64, sepal_length    6.588
dtype: float64]
```

Standard Deviation of the features (σ):

```
[sepal_length    0.510983
dtype: float64, sepal_length    0.629489
dtype: float64]
```

Discriminate function:

D(x) = g0(x) – g1(x)

Where $g_j(x) = -\log(\sigma_j) - (X-\mu_j)^2/2\sigma_j^2 + \log(\alpha_j)$

c) Classifying examples:

Based on the determinant function, the examples are classified. If D(x) > 0 class 0 or else class 1

Measurement of confusion matrix, accuracy, error, precision, recall and F-measure.

Performed 10-fold ross-validation on the data and below are the results:

```
Predicted        Iris-versicolor  Iris-virginica
Truth
Iris-versicolor                4               1
Iris-virginica                 2               3
Predicted        Iris-versicolor  Iris-virginica
Truth
Iris-versicolor                4               0
Iris-virginica                 2               4
Predicted        Iris-versicolor  Iris-virginica
Truth
Iris-versicolor                3               2
Iris-virginica                 3               2
Predicted        Iris-versicolor  Iris-virginica
Truth
Iris-versicolor                5               1
Iris-virginica                 0               4
Predicted        Iris-versicolor  Iris-virginica
Truth
Iris-versicolor                2               1
Iris-virginica                 4               3
Predicted        Iris-versicolor  Iris-virginica
Truth
Iris-versicolor                8               0
Iris-virginica                 1               1
Predicted        Iris-versicolor  Iris-virginica
Truth
Iris-versicolor                3               2
Iris-virginica                 2               3
Predicted        Iris-versicolor  Iris-virginica
Truth
Iris-versicolor                5               1
Iris-virginica                 1               3
Predicted        Iris-versicolor  Iris-virginica
Truth
Iris-versicolor                4               0
Iris-virginica                 4               2
Predicted        Iris-versicolor  Iris-virginica
Truth
Iris-versicolor                3               1
Iris-virginica                 3               3
##############################
Average of the model parameters
##############################
Error rate: 0.31
Accuracy: 0.69
Precision: [ 0.80833333  0.57785714]
Recall: [ 0.64888889  0.79      ]
F-measure: [ 0.70674391  0.64621212]
```

2) n-D 2 class Gaussian Determinant Analysis

a) Data set used: Iris.csv. Here I selected only two classes namely (Iris – Virginica and Iris-Versicolor) as the two classes and the all the features as my feature.

b) Model parameters:

Since we assume the distribution to be Gaussian and the features to be 1-D we have the parameters 'µ' and '∑' for the respective classes.

Prior probability of the classes:

```
[('Iris-versicolor', 0.5), ('Iris-virginica', 0.5)]
```

Mean of the features (µ):

```
[sepal_length    5.936
sepal_width     2.770
petal_length    4.260
petal_width     1.326
dtype: float64, sepal_length    6.588
sepal_width     2.974
petal_length    5.552
petal_width     2.026
dtype: float64]
```

Co-variance matrix of the features (∑):

```
sepal_length  sepal_width  petal_length  petal_width
sepal_length      0.266433     0.085184      0.182898     0.055780
sepal_width       0.085184     0.098469      0.082653     0.041204
petal_length      0.182898     0.082653      0.220816     0.073102
petal_width       0.055780     0.041204      0.073102     0.039106
              sepal_length  sepal_width  petal_length  petal_width
sepal_length      0.404343     0.093763      0.303290     0.049094
sepal_width       0.093763     0.104004      0.071380     0.047629
petal_length      0.303290     0.071380      0.304588     0.048824
petal_width       0.049094     0.047629      0.048824     0.075433
```

Discriminate function:

$D(x) = g0(x) - g1(x)$

Where $g_j(x) = -1/2[\log(|\sum_j|)] - 1/2[(X-\mu_j)^T\sum_j{}^{-1}(X-\mu_j)]+ \log(\alpha_j)$

c) Classifying examples:

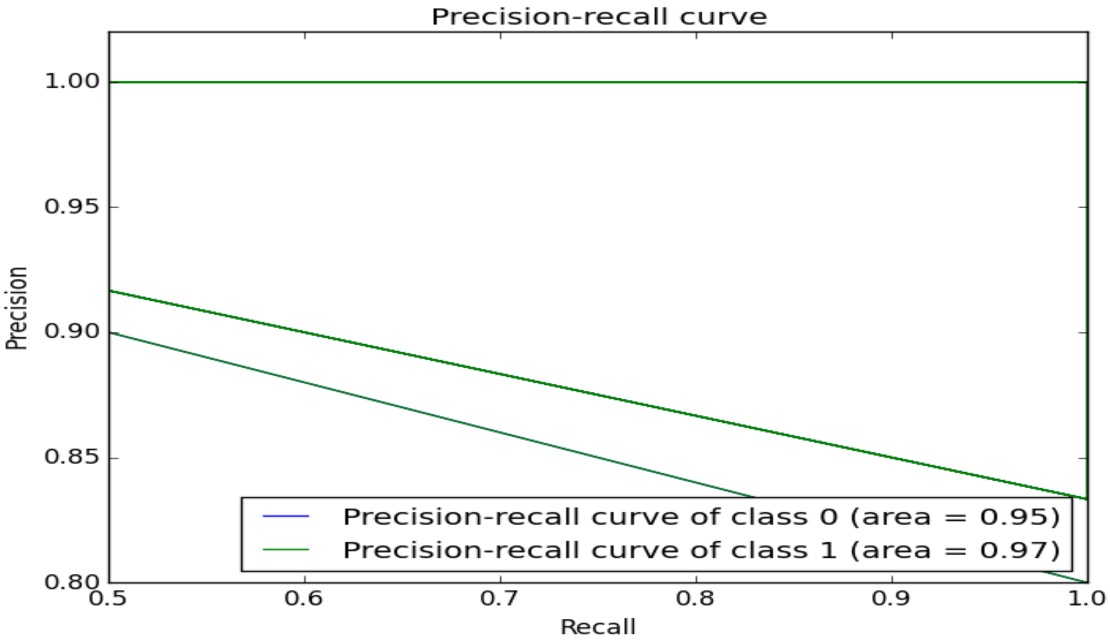
Based on the determinant function, the examples are classified. If D(x) > 0 class 0 or else class 1

Measurement of confusion matrix, accuracy, error, precision, recall and F-measure.

Performed 10-fold cross-validation on the data and below are the results:

```
(First 5)
Predicted        Iris-versicolor  Iris-virginica
Truth
Iris-versicolor                4               1
Iris-virginica                 0               5
Predicted        Iris-versicolor  Iris-virginica
Truth
Iris-versicolor                4               0
Iris-virginica                 0               6
Predicted        Iris-versicolor  Iris-virginica
Truth
Iris-versicolor                5               0
Iris-virginica                 0               5
Predicted        Iris-versicolor  Iris-virginica
Truth
Iris-versicolor                5               1
Iris-virginica                 0               4
Predicted        Iris-versicolor  Iris-virginica
Truth
Iris-versicolor                3               0
Iris-virginica                 0               7
Predicted        Iris-versicolor  Iris-virginica
Truth
Iris-versicolor                8               0
Iris-virginica                 0               2
################################
Average of the model parameters
################################
Error rate: 0.04
Accuracy: 0.96
Precision: [ 0.94666667  0.98333333]
Recall: [ 0.98        0.94333333]
F-measure: [ 0.95959596  0.95959596]
```

d) Precision – Recall curve:



There is a trade-off between precision and recall clearly from the above graph and the average area under the curve is 0.96 which equals to the average of the accuracies.

From the above results, we can clearly see that the performance has increased significantly which can be attributed to the addition of new features, which assisted in classifying the examples more accurately.

3) n-D nclass Gaussian Determinant Analysis

a) Data set used: Iris.csv. Here I selected only two classes namely (Iris – Virginica and Iris-Versicolor) as the two classes and the all the features as my feature.

b) Model parameters:

Since we assume the distribution to be Gaussian and the features to be 1-D we have the parameters 'μ' and '∑' for the respective classes.

Prior probability of the classes:

```
[('Iris-setosa', 0.33333333333333331), ('Iris-versicolor',
0.33333333333333331), ('Iris-virginica', 0.33333333333333331)]
```

Mean of the features (μ):

```
[sepal_length    5.006
sepal_width     3.418
petal_length    1.464
petal_width     0.244
dtype: float64, sepal_length    5.936
sepal_width     2.770
petal_length    4.260
petal_width     1.326
dtype: float64, sepal_length    6.588
sepal_width     2.974
petal_length    5.552
petal_width     2.026
dtype: float64]
```

Co-variance matrix of the features (∑):

```
              sepal_length  sepal_width  petal_length  petal_width
sepal_length      0.124249     0.100298      0.016139     0.010547
sepal_width       0.100298     0.145180      0.011682     0.011437
petal_length      0.016139     0.011682      0.030106     0.005698
petal_width       0.010547     0.011437      0.005698     0.011494
              sepal_length  sepal_width  petal_length  petal_width
sepal_length      0.266433     0.085184      0.182898     0.055780
sepal_width       0.085184     0.098469      0.082653     0.041204
petal_length      0.182898     0.082653      0.220816     0.073102
petal_width       0.055780     0.041204      0.073102     0.039106
               sepal_length  sepal_width  petal_length  petal_width
sepal_length      0.404343     0.093763      0.303290     0.049094
sepal_width       0.093763     0.104004      0.071380     0.047629
petal_length      0.303290     0.071380      0.304588     0.048824
petal_width       0.049094     0.047629      0.048824     0.075433
```

Discriminate function:

Since we have more than 2 classes, we cannot go for the normal approach. So,

$D(x) = \max[g_j(x)]$

Where $g_j(x) = -1/2[\log(|\sum_j|)] – 1/2[(X-\mu_j)^T \sum_j^{-1}(X-\mu_j)]+ \log(\alpha_j)$

c) Classifying examples:

Based on the determinant function, the examples are classified. Get the max of $g_j(x)$'s index and math it with the index corresponding to the class-label to get the predicted results

Measurement of confusion matrix, accuracy, error, precision, recall and F-measure.

Performed 10-fold cross-validation on the data and below are the results:

```
(First 5)
Predicted        Iris-setosa  Iris-versicolor  Iris-virginica
Truth
Iris-setosa                6                0               0
Iris-versicolor            0                4               0
Iris-virginica             0                0               5
Predicted        Iris-setosa  Iris-versicolor  Iris-virginica
Truth
Iris-setosa                6                0               0
Iris-versicolor            0                4               0
Iris-virginica             0                1               4
Predicted        Iris-setosa  Iris-versicolor  Iris-virginica
Truth
Iris-setosa                6                0               0
Iris-versicolor            0                6               0
Iris-virginica             0                0               3
Predicted        Iris-setosa  Iris-versicolor  Iris-virginica
Truth
Iris-setosa                3                0               0
Iris-versicolor            0                4               0
Iris-virginica             0                0               8
Predicted        Iris-setosa  Iris-versicolor  Iris-virginica
Truth
Iris-setosa                6                0               0
Iris-versicolor            0                4               1
Iris-virginica             0                0               4
Average of the model parameters
################################
Error rate: 0.0333333333333
Accuracy: 0.966666666667
Precision: [ 1.         0.98        0.91988095]
Recall: [ 1.          0.92166667  0.98      ]
F-measure: [ 1.         0.94531025  0.9434188 ]
```

From the above results we can conclude that the developed model is efficient enough to predict new instances of data.

4) Naïve – Baye's with Bernoulli Features:

a) Data set used: Imdb-labelled.txt

Data pre-processing stage:

The text document is read and the unique words in the documents are extracted. The given text is mapped to a feature vector containing examples as rows and the number of unique words as

columns. For. Eg.,if the document has 1000 samples and 3000 words a matrix of size 1000 X 3000 will be formed.
Iterate through each text and if the word is present mark it as ‘1’ else ‘0’, which denotes the word being present or absent. (i.e., binarizing the data)

b) Model parameters:

Since we use Bernoulli features, the parameters would be $\{ \alpha_{p|y=1} \}^{n}_{j=1}$ and $\alpha = p(y=1)$

So, $\alpha_i = \sum_{j=1}^{m} I(y = i)$

and $\alpha_{j|y=i} = \sum_{j=1}^{m} I(y = i) . Xj^{(i)} / \sum_{j=1}^{m} I(y = i)$

Prior probability:

```
[0.5, 0.5]
```

Likelihood:

```
        {0: array([ 0.28685259,  0.05976096,  0.00796813, ...,  0.00398406,
         0.00398406,  0.00398406]), 1: array([ 0.38844622,  0.05776892,
0.00199203, ...,  0.00199203, 0.00199203,  0.00199203])}
```

Determinant function:

$g_j(x) = [\sum_{j=1}^{m} Xjlog(\alpha_{j|y=i}) + (1-X_j)log(1- \alpha_{j|y=i})] + log(\alpha_i)$

c) Classifying the example:

The examples are classified using the formula, $y\hat{} = argmax_i(g_i(x))$

Measurement of confusion matrix, accuracy, error, precision, recall and F-measure.

```
     Predicted   0   1
Truth
0           50   2
1            6  42
Predicted    0   1
Truth
0           45   0
1            7  48
Predicted    0   1
Truth
0           43   1
1            7  49
Predicted    0   1
Truth
```

```
0           47   1
1           12  40
Predicted    0   1
Truth
0           53   1
1            9  37

################################
Average of the model parameters
################################
Error rate: 0.093
Accuracy: 0.907
Precision: [ 0.85305652  0.98402788]
Recall: [ 0.98577927  0.82701489]
F-measure: [ 0.9139618   0.89731896]
```

5) Naïve – Baye’s with Bernoulli Features:

b) Data set used: Imdb-labelled.txt

Data pre-processing stage:

The text document is read and the unique words in the documents are extracted. The given text is mapped to a feature vector containing examples as rows and the number of unique words as columns. For. Eg.,if the document has 1000 samples and 3000 words a matrix of size 1000 X 3000 will be formed.
Iterate through each text and on the corresponding word mark the no. of times the word appeared. This approach will be used to calculate the term-frequency as well as the document-frequency.
Here, stop words are removed, which might not be that much useful in predicting the examples and pushing the model for maximum accuracy.

c) Model parameters:

Since we use Bernoulli features, the parameters would be $\{ \alpha_{p|y=1} \}^{n}_{j=1}$ and $\alpha = p(y=1)$

So, $\alpha_i = \sum_{j=1}^{m} I(y = i)$

and $\alpha_{j|y=i} = \sum_{j=1}^{m} I(y = i) . Xj^{(i)} / \sum_{j=1}^{m} I(y = i).P$

where P is the total number of words in each class

Prior probability:

```
[0.5, 0.5]
```

Likelihood:

```
       {0: array([  3.28191449e-07,   3.28191449e-07,   3.28191449e-07,
...,
         9.84574346e-07,   3.28191449e-07,   2.95372304e-06]), 1: array([
2.80819837e-07,   2.80819837e-07,   8.42459510e-07, ...,
         2.80819837e-07,   8.42459510e-07,   2.52737853e-06])}
```

Determinant function:

Let $B_1 = \alpha^{X}_{j\ j|y=l}$, $B_2 = (1- \beta_1)^{P-X}_{j}$

$$g_j(x) =[\sum_{j=1}^{n} \log\binom{P}{Xj} . B_1. B_2] + \log(\alpha_l)$$

d) Classifying the example:

The examples are classified using the formula, $y\hat{} = \mathrm{argmax}_i(g_i(x))$

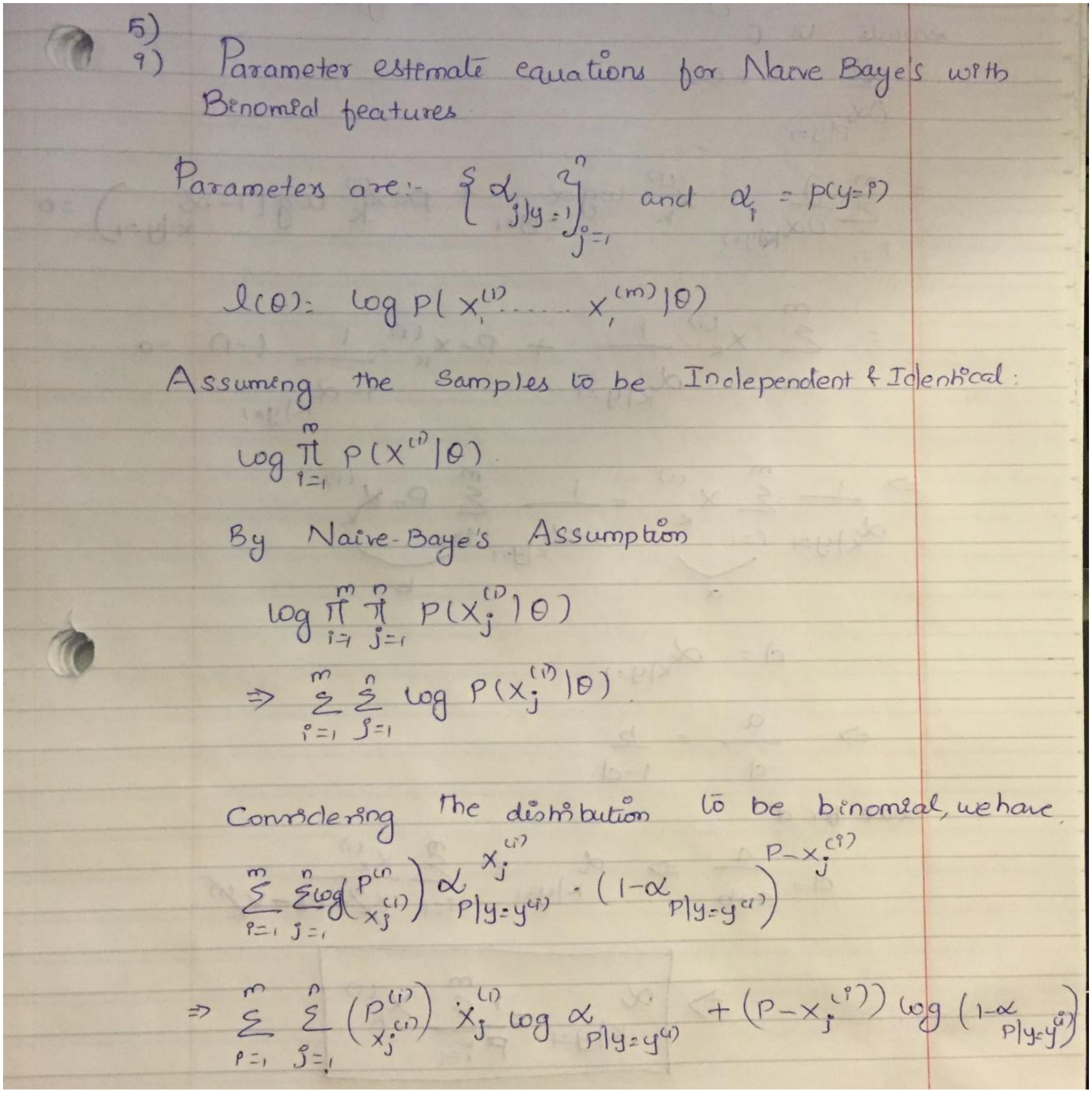
Measurement of confusion matrix, accuracy, error, precision, recall and F-measure.

```
     Predicted   0   1
Truth
0           49   3
1            1  47
Predicted   0    1
Truth
0           45   0
1            4  51
Predicted   0    1
Truth
0           40   4
1            2  54
Predicted   0    1
Truth
0           46   2
1            4  48
Predicted   0    1

################################
Average of the model parameters
################################
Error rate: 0.058
Accuracy: 0.942
Precision: [ 0.92696191  0.96179753]
Recall: [ 0.96124659  0.92110362]
F-measure: [ 0.94294667  0.94001407]
```

From the above results, it can be seen that the performance has improved while using the Binomial distribution and taking the word frequency into account. This is because a certain words in the corpus would be very well related to classifying the document. Furthermore, we can tweak the performance by using various settings such as different tokenizers, minimum document frequency, maximum document frequency,etc.,

5) a)



5)

i) Parameter estimate equations for Naive Baye's with Binomial features

Parameters are:- $\{\alpha_{j|y=i}\}_{j=1}^{n}$ and $\alpha_i = P(y=i)$

$l(\theta) = \log P(x_1^{(1)} \ldots \ldots x_1^{(m)} | \theta)$

Assuming the Samples to be Independent & Identical:

$\log \prod_{i=1}^{m} P(X^{(i)}|\theta)$.

By Naive-Baye's Assumption

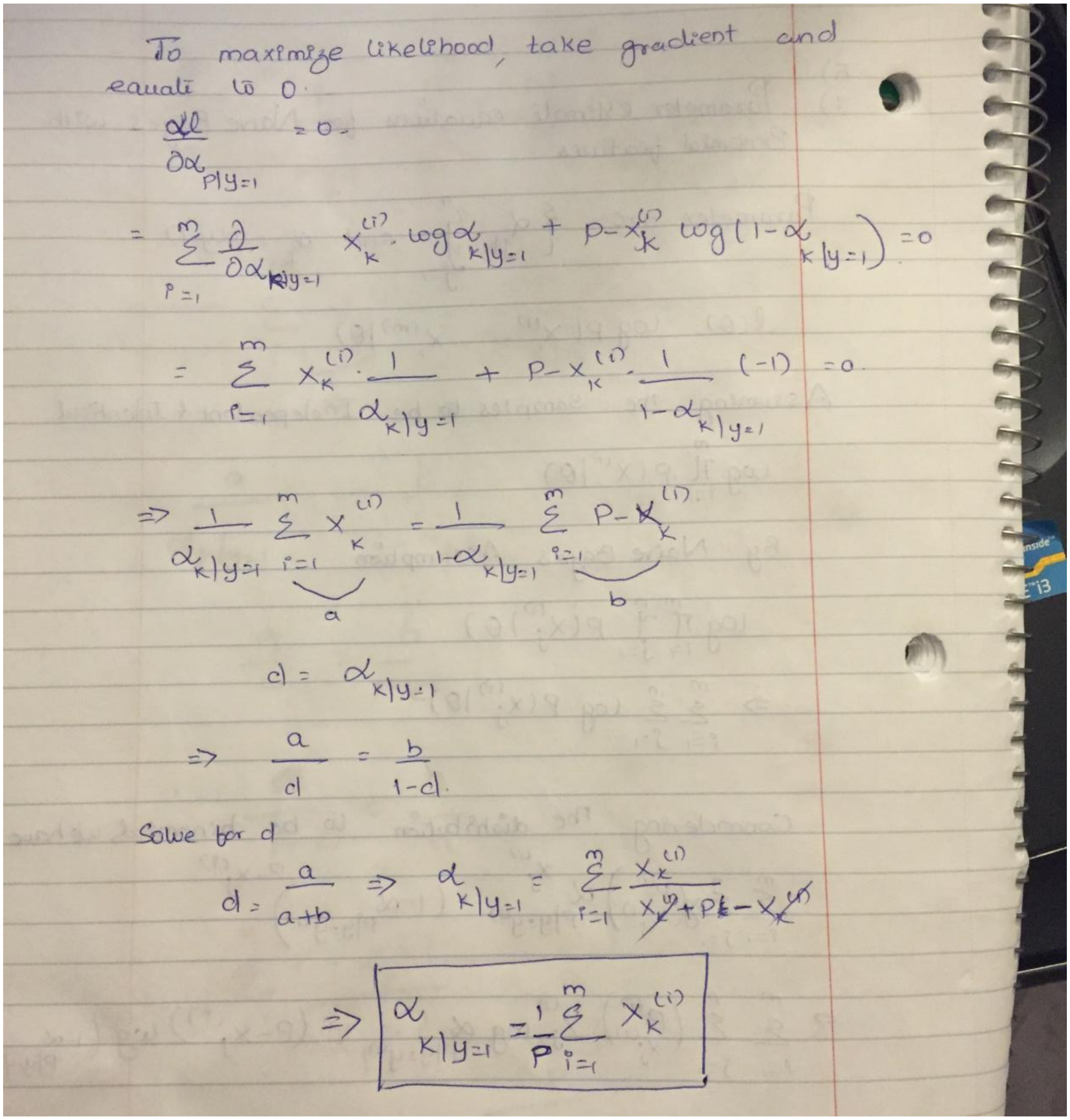
$\log \prod_{i=1}^{m} \prod_{j=1}^{n} P(X_j^{(i)}|\theta)$

$\Rightarrow \sum_{i=1}^{m} \sum_{j=1}^{n} \log P(X_j^{(i)}|\theta)$.

Considering the distribution to be binomial, we have.

$$\sum_{i=1}^{m} \sum_{j=1}^{n} \log \binom{P^{(i)}}{x_j^{(i)}} \alpha_{P|y=y^{(i)}}^{x_j^{(i)}} \cdot (1-\alpha_{P|y=y^{(i)}})^{P-x_j^{(i)}}$$

$$\Rightarrow \sum_{i=1}^{m} \sum_{j=1}^{n} \binom{P^{(i)}}{x_j^{(i)}} x_j^{(i)} \log \alpha_{P|y=y^{(i)}} + (P-x_j^{(i)}) \log (1-\alpha_{P|y=y^{(i)}})$$



To maximize likelihood, take gradient and equate to 0.

$$\frac{\partial l}{\partial \alpha_{p|y=1}} = 0.$$

$$= \sum_{p=1}^{m} \frac{\partial}{\partial \alpha_{k|y=1}} x_k^{(i)} \cdot \log \alpha_{k|y=1} + p - x_k^{(i)} \log(1 - \alpha_{k|y=1}) = 0$$

$$= \sum_{i=1}^{m} x_k^{(i)} \cdot \frac{1}{\alpha_{k|y=1}} + p - x_k^{(i)} \cdot \frac{1}{1-\alpha_{k|y=1}} (-1) = 0.$$

$$\Rightarrow \frac{1}{\alpha_{k|y=1}} \underbrace{\sum_{i=1}^{m} x_k^{(i)}}_{a} = \frac{1}{1-\alpha_{k|y=1}} \underbrace{\sum_{i=1}^{m} p - x_k^{(i)}}_{b}$$

$$d = \alpha_{k|y=1}$$

$$\Rightarrow \frac{a}{d} = \frac{b}{1-d}.$$

Solve for d

$$d = \frac{a}{a+b} \Rightarrow \alpha_{k|y=1} = \sum_{i=1}^{m} \frac{x_k^{(i)}}{\cancel{x_k^{(i)}} + p\cancel{k} - \cancel{x_k^{(i)}}}$$

$$\Rightarrow \boxed{\alpha_{k|y=1} = \frac{1}{p} \sum_{i=1}^{m} x_k^{(i)}}$$