Assignment 2

Tanuj Kumar ta.kumar@mail.utoronto.ca 1002197133

November 2017

1 Question 1

1.1 Definitions

- $y \in (1, 2, ...K)$
- $\mathbf{x} = (x_1, x_2, ...x_d)$
- $p(y = k|\boldsymbol{\mu}, \boldsymbol{\sigma}) = \alpha_k$

$$p(\mathbf{x}|y = k, \boldsymbol{\mu}, \boldsymbol{\sigma})$$

given in the sheet.

1.2 Class-Conditional Probability Derivation

Our goal is to derive an expression for $p(y = k|\mathbf{x}, \boldsymbol{\mu}, \boldsymbol{\sigma})$. Using Bayes Rule, we know:

$$p(y = k|\mathbf{x}, \boldsymbol{\mu}, \boldsymbol{\sigma}) = \frac{p(\mathbf{x}|y = k, \boldsymbol{\mu}, \boldsymbol{\sigma})p(y = k|\boldsymbol{\mu}, \boldsymbol{\sigma})}{p(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\sigma})}$$

Recall from the **law of total probability** the following:

$$p(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\sigma}) = \sum_{a=1}^{K} p(\mathbf{x}|y = a, \boldsymbol{\mu}, \boldsymbol{\sigma})p(y = a|\boldsymbol{\mu}, \boldsymbol{\sigma})$$

Where a takes on the value of every class, giving us:

$$p(y = k|\mathbf{x}, \boldsymbol{\mu}, \boldsymbol{\sigma}) = \frac{p(\mathbf{x}|y = k, \boldsymbol{\mu}, \boldsymbol{\sigma})p(y = k|\boldsymbol{\mu}, \boldsymbol{\sigma})}{\sum_{a=1}^{K} p(\mathbf{x}|y = a, \boldsymbol{\mu}, \boldsymbol{\sigma})p(y = a|\boldsymbol{\mu}, \boldsymbol{\sigma})}$$

From our definitions, we can define:

$$p(y = k|\boldsymbol{\mu}, \boldsymbol{\sigma}) = \alpha_k$$

$$p(\mathbf{x}|y = k, \boldsymbol{\mu}, \boldsymbol{\sigma}) =$$

$$(\prod_{i=1}^{D} 2\pi\sigma_i^2)^{-1/2}\mathsf{exp}(-\sum_{i=1}^{D}(2\sigma_i^2)^{-1}(x_i - \mu_{k_i})^2)$$

Therefore, our expression is:

$$p(y = k|\mathbf{x}, \boldsymbol{\mu}, \boldsymbol{\sigma}) = \frac{(\prod_{i=1}^{D} 2\pi\sigma_i^2)^{-1/2}\mathsf{exp}(-\sum_{i=1}^{D}(2\sigma_i^2)^{-1}(x_i - \mu_{k_i})^2)(\alpha_k)}{\sum_{j=a}^{K}(2\pi\prod_{i=1}^{D}\sigma_i^2)^{-1/2}\mathsf{exp}(-\sum_{i=1}^{D}(2\sigma_i^2)^{-1}(x_i - \mu_{a_i})^2)(\alpha_a)}$$

1.3 Negative Log Likelihood Expression

Let D be the dataset consisting of points $(y^i, \mathbf{x}^i), i \in (1, ..., N)$ and $\boldsymbol{\theta} = (\boldsymbol{\alpha}, \boldsymbol{\mu}, \boldsymbol{\sigma})$. Assume that the points in D are **IID**.

Define the following:

$$p(y^{(1)}, \mathbf{x}^{(1)}, ..., y^{(N)}, \mathbf{x}^{(N)}|\boldsymbol{\theta})$$

Because the data is under the **IID** assumption, we can rewrite the above as a product:

$$\prod_{i=1}^{N} p(\mathbf{x}^{(i)}, y^{(i)}|\boldsymbol{\theta})$$

Recall for an individual i:

$$p(\mathbf{x}^{(i)}, y^{(i)}|\boldsymbol{\theta}) = p(\mathbf{x}^{(i)}|y^{(i)}, \boldsymbol{\theta})p(y^i)$$

$$p(\mathbf{x}^{(i)}, y^{(i)}|\boldsymbol{\theta}) = p(\mathbf{x}^{(i)}|y^{(i)} = k^i, \boldsymbol{\theta})p(y^i = k^i|\boldsymbol{\theta})$$

Where k^i is the **class assigned to the specific point** $y^{(i)}$. Why this and not over a sum $1...K$? Because we are specifically looking at a point $(\mathbf{x}^{(i)}, y^{(i)})$ where each $\mathbf{x}^{(i)} = (x_1^i...x_d^i)$ is explicitly defined and $y^{(i)}$ only takes the value of k^i. In other words, the joint probability of getting this already-observed $\mathbf{x}^{(i)}$ value with any other $y^{(i)}$ would be zero, so we only care about this situation.

$$p(\mathbf{x}^{(i)}, y^{(i)}|\boldsymbol{\theta}) = ((2\pi\prod_{j=1}^{D}\sigma_j^2)^{-1/2}\mathsf{exp}(-\sum_{j=1}^{D}(2\sigma_j^2)^{-1}(x_j^i - \mu_{k_j^i})^2))(\alpha_{k^i})$$

Therefore, the expression is:

$$p(y^{(1)}, \mathbf{x}^{(1)}, ..., y^{(N)}, \mathbf{x}^{(N)}|\boldsymbol{\theta}) = \prod_{i=1}^{N}((\prod_{j=1}^{D} 2\pi\sigma_j^2)^{-1/2}\mathsf{exp}(-\sum_{j=1}^{D}(2\sigma_j^2)^{-1}(x_j^i - \mu_{k_j^i})^2))(\alpha_{k^i})$$

Now we will neg-log this expression. The result is:

$$-\log p(y^{(1)}, \mathsf{x}^{(1)}, ..., y^{(N)}, \mathsf{x}^{(N)}|\boldsymbol{\theta}) = -\log \prod_{i=1}^{N}((\prod_{j=1}^{D} 2\pi\sigma_j^2)^{-1/2}\mathsf{exp}(-\sum_{j=1}^{D}(2\sigma_j^2)^{-1}(x_j^i - \mu_{k_j^i})^2))(\alpha_{k^i})$$

$$= -\log[(\prod_{j=1}^{D} 2\pi\sigma_j^2)^{-N/2}\prod_{i=1}^{N}[(\alpha_{k^i})\mathsf{exp}(-\sum_{j=1}^{D}(2\sigma_j^2)^{-1}(x_j^i - \mu_{k_j^i})^2)]]$$

$$= -\log[(\prod_{j=1}^{D} 2\pi\sigma_j^2)^{-N/2}[(\alpha_{k^1} \cdot ... \cdot \alpha_{k^N}) \cdot \mathsf{exp}(-\sum_{j=1}^{D}(2\sigma_j^2)^{-1}(x_j^1 - \mu_{k_j^1})^2) \cdot ... \cdot \mathsf{exp}(-\sum_{j=1}^{D}(2\sigma_j^2)^{-1}(x_j^N - \mu_{k_j^N})^2)]]$$

Recall that $e^x \cdot e^y = e^{x+y}$. Thus, we condense the Euler products into one Euler term.

$$= -\log[(\prod_{j=1}^{D} 2\pi\sigma_j^2)^{-N/2}[(\alpha_{k^1} \cdot ... \cdot \alpha_{k^N}) \cdot \mathsf{exp}(-\sum_{i=1}^{N}\sum_{j=1}^{D}(2\sigma_j^2)^{-1}(x_j^i - \mu_{k_j^i})^2)]]$$

Bringing in the log,

$$= -[\log[(\prod_{j=1}^{D} 2\pi\sigma_j^2)^{-N/2}[(\alpha_{k^1} \cdot ... \cdot \alpha_{k^N}) \cdot \mathsf{exp}(-\sum_{i=1}^{N}\sum_{j=1}^{D}(2\sigma_j^2)^{-1}(x_j^i - \mu_{k_j^i})^2)]]]$$

$$= -[\log[(\prod_{j=1}^{D} 2\pi\sigma_j^2)^{-N/2}] + \log[(\alpha_{k^1} \cdot ... \cdot \alpha_{k^N}) \cdot \mathsf{exp}(-\sum_{i=1}^{N}\sum_{j=1}^{D}(2\sigma_j^2)^{-1}(x_j^i - \mu_{k_j^i})^2)]]$$

$$= -[\frac{-N}{2}\log[(\prod_{j=1}^{D} 2\pi\sigma_j^2)] + \log[(\alpha_{k^1} \cdot ... \cdot \alpha_{k^N})]\log[\mathsf{exp}(-\sum_{i=1}^{N}\sum_{j=1}^{D}(2\sigma_j^2)^{-1}(x_j^i - \mu_{k_j^i})^2)]]$$

$$= -[\frac{-N}{2}\log[(\prod_{j=1}^{D} 2\pi\sigma_j^2)] + \log[(\alpha_{k^1} \cdot ... \cdot \alpha_{k^N})] - \sum_{i=1}^{N}\sum_{j=1}^{D}(2\sigma_j^2)^{-1}((x_j^i - \mu_{k_j^i})^2)]$$

$$= \frac{N}{2}\log[(\prod_{j=1}^{D} 2\pi\sigma_j^2)] - \log[\prod_{i=1}^{N}\alpha_{k^i}] + \sum_{i=1}^{N}\sum_{j=1}^{D}(2\sigma_j^2)^{-1}((x_j^i - \mu_{k_j^i})^2)]$$

We can further simplify this to turn $log[(\prod_{j=1}^{D} 2\pi\sigma_j^2)] = \sum_{j=1}^{D}\log 2\pi\sigma_j^2$ and $\log[\prod_{i=1}^{N}\alpha_{k^i}]$ to $\sum_{i=1}^{N}\log[\alpha_{k^i}]$ as necessary, giving us:

$$= \frac{N}{2}\sum_{j=1}^{D} \log\left[2\pi\sigma_j^2\right] - \sum_{i=1}^{N} \log[\alpha_{k^i}] + \sum_{i=1}^{N}\sum_{j=1}^{D}[(2\sigma_j^2)^{-1}((x_j^i - \mu_{k_j^i})^2)]$$

And finally, we simplify the remaining logs:

$$= \frac{N}{2}\sum_{j=1}^{D}[\log\left[2\pi\right] + \log\left[\sigma_j^2\right]] - \sum_{i=1}^{N} \log[\alpha_{k^i}] + \sum_{i=1}^{N}\sum_{j=1}^{D}[(2\sigma_j^2)^{-1}((x_j^i - \mu_{k_j^i})^2)]$$

1.4 Parameter-based Partial Derivatives

From above we have,

$$l(\boldsymbol{\theta}, D) = \frac{N}{2}\sum_{j=1}^{D}[\log\left[2\pi\right] + \log\left[\sigma_j^2\right]] - \sum_{i=1}^{N} \log[\alpha_{k^i}] + \sum_{i=1}^{N}\sum_{j=1}^{D}[(2\sigma_j^2)^{-1}((x_j^i - \mu_{k_j^i})^2)]$$

$$l(\boldsymbol{\theta}, D) = \frac{1}{2}Nd\log 2\pi + \frac{1}{2}N\log\sigma_1^2 + ... + \frac{1}{2}N\log\sigma_d^2 - \sum_{i=1}^{N}\log[\alpha_{k^i}] + \sum_{i=1}^{N}\sum_{j=1}^{D}[(2\sigma_j^2)^{-1}((x_j^i - \mu_{k_j^i})^2)]$$

1.4.1 Derivative with respect to $\boldsymbol{\sigma}^2$

$$\frac{\partial l(\boldsymbol{\theta}, D)}{\partial \sigma_I^2}$$

$$\frac{\partial}{\partial \sigma_I^2}\frac{1}{2}Nd\log 2\pi + \frac{1}{2}N\log\sigma_1^2 + ... + \frac{1}{2}N\log\sigma_d^2 - \sum_{i=1}^{N}\log[\alpha_{k^i}] + \sum_{i=1}^{N}\sum_{j=1}^{D}[(2\sigma_j^2)^{-1}((x_j^i - \mu_{k_j^i})^2)]$$

Through the partial derivative, we can remove the treated-constants $\frac{1}{2}Nd\log 2\pi$, every $\frac{1}{2}N\log\sigma_j^2$ where $j \neq I$, and the α_k term. In the double-sum over N and D at the farthest end of the expression, note that we can treat every other $j \neq I$ as constant, cancelling these terms out through the differentiation, leaving us with only

$$\sum_{i=1}^{N}[(2\sigma_j^2)^{-1}((x_I^i - \mu_{k_I^i})^2)]$$

where we look at component $j = I$ for all points $1...N$. The derivative of this with respect to σ_I^2 is just

$$-\sum_{i=1}^{N}[2(2\sigma_I^2)^{-2}((x_I^i - \mu_{k_I^i})^2)]$$

Thus, our final result is:

$$\frac{\partial l(\boldsymbol{\theta}, D)}{\partial \sigma_I^2} = \frac{1}{2}N\frac{1}{\sigma_I^2} - \sum_{i=1}^{N}[2(2\sigma_I^2)^{-2}((x_I^i - \mu_{k_I^i})^2)]$$

1.4.2 Derivative with respect to $\boldsymbol{\mu}$

$$\frac{\partial l(\boldsymbol{\theta}, D)}{\partial \mu_{k_j}}$$

$$\frac{\partial}{\partial \mu_{k_j}}\frac{1}{2}Nd\log 2\pi+\frac{1}{2}N\log\sigma_1^2+...+\frac{1}{2}N\log\sigma_d^2-\sum_{i=1}^{N}\log[\alpha_{k^i}]+\sum_{i=1}^{N}\sum_{j=1}^{D}[(2\sigma_j^2)^{-1}((x_j^i-\mu_{k_j^i})^2)]$$

Because the first terms before the double-sum are all treated as constants, we can cancel them, leaving us with:

$$\frac{\partial}{\partial \mu_{k_j}}\sum_{i=1}^{N}\sum_{j=1}^{D}[(2\sigma_j^2)^{-1}((x_j^i - \mu_{k_j^i})^2)]$$

Recall the definition of the $\mu_{k_j^i}$ terms: they are simply μ_{k_j} when k is the *specified class* of $y^{(i)}$ in a point $(\mathbf{x}^{(i)}, y^{(i)})$. Therefore, when we are taking the derivative with respect to μ_{k_j}, what we are in fact doing is we care only about the $\mu_{k_j^i}$ corresponding to points $(\mathbf{x}^{(i)}, y^{(i)})$ that will have this mean, and take all others as constants. Thus, define the set $D' \subseteq D$ of size $n \leq N$ which consists of points $(\mathbf{x}^{(i)}, y^{(i)})$ for which $y^{(i)} = k$ and therefore $\mu_{k_j^i}$ exists. Differentiating with respect to this, we get the following term:

$$\frac{\partial l(\boldsymbol{\theta}, D)}{\partial \mu_{k_j}} = \sum_{i=1}^{n}\frac{-2(x_j^i - \mu_{k_j^i})}{2\sigma_j^2}$$

$$\frac{\partial l(\boldsymbol{\theta}, D)}{\partial \mu_{k_j}} = -\sum_{i=1}^{n}\frac{(x_j^i - \mu_{k_j^i})}{\sigma_j^2}$$

1.5 Parameter Derivation via Maximum Likelihood

1.5.1 Deriving $\boldsymbol{\mu}$

Recall from above:

$$\frac{\partial l(\boldsymbol{\theta}, D)}{\partial \mu_{k_j}} = -\sum_{i=1}^{n}\frac{(x_j^i - \mu_{k_j^i})}{\sigma_j^2} = 0$$

We will set it to zero to maximize and receive an expression for the components μ_{k_j} of $\boldsymbol{\mu}$. Then:

$$-\sum_{i=1}^{n} \frac{(x_j^i - \mu_{k_j^i})}{\sigma_j^2} = 0$$

Note that of the numerator term, the $\mu_{k_j^i}$ is in fact the same term, differing only because different points in the set D' use the identical mean in this context here. Therefore we can rewrite this as,

$$\frac{(x_j^1 + ... + x_j^n)}{\sigma_j^2} - \frac{n\mu_{k_j^i}}{\sigma_j^2} = 0$$

$$\frac{(x_j^1 + ... + x_j^n)}{\sigma_j^2} = \frac{n\mu_{k_j}}{\sigma_j^2}$$

$$(x_j^1 + ... + x_j^n) = \frac{n\mu_{k_j}\sigma_j^2}{\sigma_j^2}$$

$$(x_j^1 + ... + x_j^n) = n\mu_{k_j}$$

$$\mu_{k_j} = \frac{(x_j^1 + ... + x_j^n)}{n}$$

This intuitively means that the mean of class k conditioned on feature j is simply the sum of the j feature values of the n points in D whose class is assigned k, divided by the total number n, which is reasonable.

1.5.2 Deriving σ

Recall from above:

$$\frac{\partial l(\boldsymbol{\theta}, D)}{\partial \sigma_I^2} = \frac{1}{2}N\frac{1}{\sigma_I^2} - \sum_{i=1}^{N}[2(2\sigma_I^2)^{-2}((x_I^i - \mu_{k_I^i})^2)] = 0$$

$$\frac{1}{2}N\frac{1}{\sigma_I^2} = \sum_{i=1}^{N}[2(2\sigma_I^2)^{-2}((x_I^i - \mu_{k_I^i})^2)]$$

$$\frac{1}{4}N\frac{1}{\sigma_I^2} = (2\sigma_I^2)^{-2}\sum_{i=1}^{N}[((x_I^i - \mu_{k_I^i})^2)]$$

$$\frac{1}{4}N\frac{4\sigma_I^4}{\sigma_I^2} = \sum_{i=1}^{N}[((x_I^i - \mu_{k_I^i})^2)]$$

$$N\sigma_I^2 = \sum_{i=1}^{N}[((x_I^i - \mu_{k_I^i})^2)]$$

$$\sigma_I^2 = \frac{\sum_{i=1}^{N}[((x_I^i - \mu_{k_I^i})^2)]}{N}$$

This intuitively means that the shared variance of the feature I is simply the feature-value's squared distance from the feature mean of that particular point's class, summed and divided by the total number of points N, which is equivalent to the usual formula for variance.

2 Question 2

2.1 2.0 Mean Plot

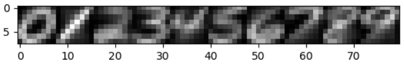

Figure 1: Plotted means

2.2 2.1 k-NN

2.2.1 Train and Test Classification Accuracies

- $K = 1$
  - Train: 0.98
  - Test: 0.97
- $K = 15$
  - Train: 0.96
  - Test: 0.95

2.2.2 Tiebreaking

In the situation of a tie, we simply tiebreak based on how the argmax function works: namely, we choose the label with the lowest index value. (See the choose label function) Because under a tiebreaking situation we have a situation where any of the label options are equally viable, choosing the lowest-index label should not have a marginally negative effect.

2.2.3 Cross-validated K and Accuracies

Our results, depicting classification accuracies for train and test data from $K = 1$ to $K = 15$

- **Training data**
  1. 1.0
  2. 0.9826
  3. 0.9834
  4. 0.9780
  5. 0.9776
  6. 0.9743
  7. 0.9737
  8. 0.9705
  9. 0.9693
  10. 0.9676
  11. 0.9654
  12. 0.9631
  13. 0.9624
  14. 0.9601
  15. 0.9594
- **Test data**
  1. 0.9685
  2. 0.9618
  3. 0.9665
  4. 0.9678
  5. 0.9645
  6. 0.9633
  7. 0.9615
  8. 0.9605

    9. 0.9610
    10. 0.9595
    11. 0.9583
    12. 0.9578
    13. 0.9573
    14. 0.9585
    15. 0.9501

- **Accuracies between folds**

    1. 0.96485714285714275
    2. 0.95785714285714274
    3. 0.96485714285714275
    4. 0.95985714285714285
    5. 0.96242857142857141
    6. 0.96028571428571419
    7. 0.95828571428571419
    8. 0.95528571428571429
    9. 0.9524285714285714
    10. 0.95371428571428574
    11. 0.95285714285714285
    12. 0.95057142857142851
    13. 0.95085714285714285
    14. 0.95114285714285707
    15. 0.94871428571428562

From this cross validation, it can be concluded that the best K value to use is in fact $K = 1$.

2.3 2.2 Class-Conditional Gaussians

2.3.1 Covariance Plot

2.3.2 Average Conditional Log-Likelihood

- **Train:** -0.1246
- **Test:** -0.1967

2.3.3 Train and Test Accuracy

- **Train:** 0.9814
- **Test:** 0.9728

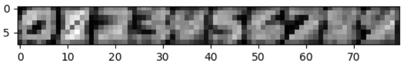

Figure 2: Plotted covariance diagonals

2.4 2.3 Naive Bayes

2.4.1 Eta Plot

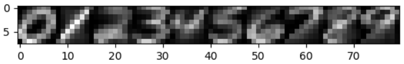

Figure 3: Plotted eta values

2.4.2 New Datapoint Plot

2.4.3 Average Conditional Log-Likelihood

- **Train:** -0.9438
- **Test:** -0.9873

2.4.4 Train and Test Accuracy

- **Train:** 0.7741
- **Test:** 0.7643

0
5
0
10
20
30
40
50
60
70

Figure 4: A sample point. This point's data was generated by randomly setting to 1 a specific feature based on the mean of that feature being on for the specific class. More details in code.

2.5 2.4 Model Comparison

- **k-NN**. Although it performed the second best, its ranking as second was only a difference by about one percentage point, and accounting for statistical variance it is entirely possible that the k-NN classifier essentially performs just as well as the class-conditional Gaussian model. For k-NN we made no assumptions and simply classified based on nearest distance, but the high likelihood of this result implies that there is a clear demarcation of regional class similarities based on feature values for data points, which supports the success of the Gaussian model below.

- **Class-Conditional Gaussians**. By a slight margin of 1% in its accuracies over k-NN, this model performed the best out of the three models. Because we made an assumption that the data roughly fit a Gaussian distribution with respect to individual classes, the high success of the class-conditional Gaussians appears to point to this assumption being highly reasonable.

- **Naive Bayes**. At around 77% accuracy, the Naive Bayes model was by far the worst classifier. The Naive Bayes assumption of complete independence and binary feature values over a Bernoulli distribution appeared to be excessively strong to the point of being detrimental to classification accuracy, meaning that these assumptions do not properly and accurately capture the reality of the data. Although the Naive Bayes model was fairly quick to train, the strong assumptions made weaken its reliability in classification.