Assignment 2

Adriana Salcedo

November 12, 2017

1 Q1

1.1

- y is a response for a single observation
- $\mathbf{x}$ is an N x d vector, in the case of a single observation (below) it is a 1 x d vector
- $\boldsymbol{\mu}_k$ is a 1 x d vector of the means of each feature for class k
- $\boldsymbol{\sigma}$ is a d x 1 vector of the variances of each feature
- $\boldsymbol{\alpha}_k$ is a d x 1 vector of the prior for each class

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

$$= \frac{\mathbb{1}\{y_k = k\}(2\pi\boldsymbol{\sigma}^T\boldsymbol{\sigma})^{-\frac{1}{2}} exp\{-\frac{1}{2\boldsymbol{\sigma}^T\boldsymbol{\sigma}}(\mathbf{x} - \boldsymbol{\mu}_k)(\mathbf{x} - \boldsymbol{\mu}_k)^T\}\boldsymbol{\alpha}_k}{\sum_{k=1}^{K} p(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\sigma})\boldsymbol{\alpha}_k}$$

$$= \frac{\mathbb{1}\{y_k = k\}(2\pi\boldsymbol{\sigma}^T\boldsymbol{\sigma})^{-\frac{1}{2}} exp\{-\frac{1}{2\boldsymbol{\sigma}^T\boldsymbol{\sigma}}(\mathbf{x} - \boldsymbol{\mu}_k)(\mathbf{x} - \boldsymbol{\mu}_k)^T\}\boldsymbol{\alpha}_k}{\sum_{k=1}^{K}(2\pi\boldsymbol{\sigma}^T\boldsymbol{\sigma})^{-\frac{1}{2}} exp\{-\frac{1}{2\boldsymbol{\sigma}^T\boldsymbol{\sigma}}(\mathbf{x} - \boldsymbol{\mu}_k)(\mathbf{x} - \boldsymbol{\mu}_k)^T\}\boldsymbol{\alpha}_k}$$

1.2

For each class k

$$\ell(\boldsymbol{\theta}, D) = \prod_{n=1}^{N} \mathbb{1}\{t_k = k\}p(\mathbf{x}_n|y_n = k, \boldsymbol{\mu}_k, \boldsymbol{\sigma})p(y = k)$$

$$\prod_{n=1}^{N} \mathbb{1}\{y_k = k\}(2\pi\boldsymbol{\sigma}^T\boldsymbol{\sigma})^{-\frac{1}{2}} exp\{-\frac{1}{2\boldsymbol{\sigma}^T\boldsymbol{\sigma}}(\mathbf{x} - \boldsymbol{\mu}_k)(\mathbf{x} - \boldsymbol{\mu}_k)^T\}\boldsymbol{\alpha}_k$$

Apply negative log

$$= -\sum_{i}^{N} \mathbb{1}\{y_k = k\} \log p(\mathbf{x}_n|y_n = k, \boldsymbol{\mu}_{\mathbf{k}}, \boldsymbol{\sigma}) - \mathbb{1}\{y_k = k\} \log \alpha_k$$

$$= -\sum_{i}^{N} \mathbb{1}\{y_k = k\} \log{(2\pi \boldsymbol{\sigma}^T \boldsymbol{\sigma})^{-\frac{1}{2}}} + \mathbb{1}\{y_k = k\} \frac{1}{2\boldsymbol{\sigma}^T \boldsymbol{\sigma}} (\mathbf{x} - \boldsymbol{\mu}_k)(\mathbf{x} - \boldsymbol{\mu}_k)^T - \mathbb{1}\{y_k = k\} \log \boldsymbol{\alpha}_k$$

$$= \sum_{i}^{N} \mathbb{1}\{y_k = k\} \frac{1}{2} \log{(2\pi \boldsymbol{\sigma}^T \boldsymbol{\sigma})} + \sum_{i}^{N} \mathbb{1}\{y_k = k\} \frac{1}{2\boldsymbol{\sigma}^T \boldsymbol{\sigma}} (\mathbf{x}_n \mathbf{x}_n^T - 2\mathbf{x}_n \boldsymbol{\mu}_k^T + \boldsymbol{\mu}_k \boldsymbol{\mu}_k^T) - \mathbb{1}\{y_k = k\} \log \boldsymbol{\alpha}_k$$

Taking the derivative with respect to μ_{ki}

$$\sum_{i}^{N} \mathbb{1}\{y_k = k\} \frac{1}{2\boldsymbol{\sigma}^T \boldsymbol{\sigma}} (-2x_{ki} + 2\mu_{ki})$$

Taking the derivative with respect to σ_{ki}

$$= \sum_{i}^{N} \mathbb{1}\{y_k = k\} \frac{1}{2} \frac{1}{(2\pi \boldsymbol{\sigma}^T \boldsymbol{\sigma})} 2\pi - \sum_{i}^{N} \mathbb{1}\{y_k = k\} \frac{1}{2} (\boldsymbol{\sigma}^T \boldsymbol{\sigma})^{-2} (\mathbf{x}_n \mathbf{x}_n^T - 2\mathbf{x}_n \boldsymbol{\mu}_k^T + \boldsymbol{\mu}_k \boldsymbol{\mu}_k^T)$$

$$= \sum_{i}^{N} \mathbb{1}\{y_k = k\} \frac{1}{2} \frac{1}{(\boldsymbol{\sigma}^T \boldsymbol{\sigma})} - \sum_{i}^{N} \mathbb{1}\{y_k = k\} \frac{1}{2} (\boldsymbol{\sigma}^T \boldsymbol{\sigma})^{-2} (\mathbf{x} - \boldsymbol{\mu}_k)(\mathbf{x} - \boldsymbol{\mu}_k)^T$$

1.3

Setting the derivative with respect to μ_{ki} to zero

$$0 = \sum_{i}^{N} \mathbb{1}\{y_k = k\} \frac{1}{2\boldsymbol{\sigma}^T \boldsymbol{\sigma}} 2x_{ki} + \sum_{i}^{N} \mathbb{1}\{y_k = k\} \frac{1}{2\boldsymbol{\sigma}^T \boldsymbol{\sigma}} 2\mu_{ki}$$

$$\sum_{i}^{N} \mathbb{1}\{y_k = k\} \frac{1}{\boldsymbol{\sigma}^T \boldsymbol{\sigma}} \mu_{ki} = \sum_{i}^{N} \mathbb{1}\{y_k = k\} \frac{1}{\boldsymbol{\sigma}^T \boldsymbol{\sigma}} x_{ki}$$

$$\mu_{ki} = \frac{\sum_{i}^{N} \mathbb{1}\{y_k = k\} x_{ki}}{\sum_{i}^{N} \mathbb{1}\{y_k = k\}}$$

For for feature i=1 to i=d, and for each class k =1 to k = k

$$\boldsymbol{\mu}_k = \begin{bmatrix} \frac{\sum_{i}^{N} \mathbb{1}\{y_k=k\} x_{k=1,i=1}}{\sum_{i}^{N} \mathbb{1}\{y_k=k\}} & \frac{\sum_{i}^{N} \mathbb{1}\{y_k=k\} x_{k=1,i=2}}{\sum_{i}^{N} \mathbb{1}\{y_k=k\}} & ... & \frac{\sum_{i}^{N} \mathbb{1}\{y_k=k\} x_{k=1,i=d}}{\sum_{i}^{N} \mathbb{1}\{y_k=k\}} \\ \frac{\sum_{i}^{N} \mathbb{1}\{y_k=k\} x_{k=2,i=1}}{\sum_{i}^{N} \mathbb{1}\{y_k=k\}} & \frac{\sum_{i}^{N} \mathbb{1}\{y_k=k\} x_{k=2,i=2}}{\sum_{i}^{N} \mathbb{1}\{y_k=k\}} & ... & \frac{\sum_{i}^{N} \mathbb{1}\{y_k=k\} x_{k=2,i=d}}{\sum_{i}^{N} \mathbb{1}\{y_k=k\}} \\ ... \\ \frac{\sum_{i}^{N} \mathbb{1}\{y_k=k\} x_{k=k,i=1}}{\sum_{i}^{N} \mathbb{1}\{y_k=k\}} & \frac{\sum_{i}^{N} \mathbb{1}\{y_k=k\} x_{k=k,i=2}}{\sum_{i}^{N} \mathbb{1}\{y_k=k\}} & ... & \frac{\sum_{i}^{N} \mathbb{1}\{y_k=k\} x_{k=k,i=d}}{\sum_{i}^{N} \mathbb{1}\{y_k=k\}} \end{bmatrix}$$

Setting the derivative with respect to σ_i^2 to zero

$$0 = \sum_i^N \mathbb{1}\{y_k = k\}\frac{1}{2}\frac{1}{(\boldsymbol{\sigma}^T\boldsymbol{\sigma})} - \sum_i^N \mathbb{1}\{y_k = k\}\frac{1}{2}(\boldsymbol{\sigma}^T\boldsymbol{\sigma})^{-2}(\mathbf{x} - \boldsymbol{\mu}_k)(\mathbf{x} - \boldsymbol{\mu}_k)^T$$

$$\sum_i^N \mathbb{1}\{y_k = k\}\frac{1}{2}\frac{1}{(\boldsymbol{\sigma}^T\boldsymbol{\sigma})} = \sum_i^N \mathbb{1}\{y_k = k\}\frac{1}{2}\frac{1}{\boldsymbol{\sigma}^T\boldsymbol{\sigma}^2}(\mathbf{x} - \boldsymbol{\mu}_k)(\mathbf{x} - \boldsymbol{\mu}_k)^T$$

$$\sum_i^N \mathbb{1}\{y_k = k\} = \sum_i^N \mathbb{1}\{y_k = k\}\frac{1}{\boldsymbol{\sigma}^T\boldsymbol{\sigma}}(\mathbf{x} - \boldsymbol{\mu}_k)(\mathbf{x} - \boldsymbol{\mu}_k)^T$$

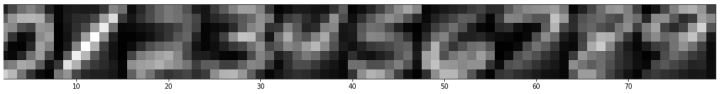
$$\boldsymbol{\sigma}^T\boldsymbol{\sigma} = \frac{\sum_i^N \mathbb{1}\{y_k = k\}(\mathbf{x} - \boldsymbol{\mu}_k)(\mathbf{x} - \boldsymbol{\mu}_k)^T}{\sum_i^N \mathbb{1}\{y_k = k\}}$$

For each feature i

$$\sigma_i^2 = \frac{\sum_i^N \mathbb{1}\{y_k = k\}(x_i - \mu_{ki})^2}{\sum_i^N \mathbb{1}\{y_k = k\}}$$

2

2.0.1



Means for each feature for each class

2.1

2.1.1

K=1 train accuracy: 0.9998, test accuracy: 0.967
K=15 train accuracy: 0.957 test accuracy: 0.9512

2.1.2

I chose to break ties by randomly selecting a class from among the tied classes. I chose this approach as it gave each of the classes tied for the most votes an equal chance of being selected, allowed the algorithm to use close to the same k for every point (as opposed to using k=1 when there are ties, or dropping k until the tie is broken), did not exclude any test points, and always gave a class for each test point (ie there were no cases where a class was undefined).

2.1.3

The optimal k was 3. train accuracy: 0.9825, test accuracy: 0.9645

k	Accuracy
1	0.959714
2	0.948571
3	0.961857
4	0.956857
5	0.957571
6	0.952143
7	0.953857
8	0.949571
9	0.948714
10	0.946429
11	0.946143
12	0.944286
13	0.943857
14	0.942857
15	0.942143

2.2

MLE estimates for μ_k and Σ_k

$\mathbf{x}_i$ is a d x 1 vector of the features for one observation and μ_{ki} is a d x 1 vector of the means of the features for class k. y is a N x 1 vector of responses for each observation indicating the class

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

Applying the log

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

$$= \sum_{i=1}^{N} \mathbb{1}\{y_k = k\} \log{(2\pi)^{-\frac{d}{2}}|\Sigma_k|^{-\frac{1}{2}}} - \sum_{i=1}^{N} \mathbb{1}\{y_k = k\} \frac{1}{2}(\mathbf{x}_i - \boldsymbol{\mu}_k)^T \Sigma_k^{-1}(\mathbf{x}_i - \boldsymbol{\mu}_k) + \log \frac{1}{10}$$

$$= -\sum_{i=1}^{N} \mathbb{1}\{y_k = k\} \log{(2\pi)^{\frac{d}{2}}|\Sigma_k|^{\frac{1}{2}}} - \sum_{i=1}^{N} \mathbb{1}\{y_k = k\} \frac{1}{2}(\mathbf{x}_i - \boldsymbol{\mu}_k)^T \Sigma_k^{-1}(\mathbf{x}_i - \boldsymbol{\mu}_k) + \log \frac{1}{10}$$

taking the derivative w.r.t μ_k and setting it to zero

$$0 = -\sum_{i=1}^{N} \mathbb{1}\{y_k = k\} \Sigma_k^{-1} \frac{1}{2}(-2\mathbf{x}_i + 2\boldsymbol{\mu}_k)$$

$$0 = \sum_{i=1}^{N} \mathbb{1}\{y_k = k\} \Sigma_k^{-1}\mathbf{x}_i - \sum_{i=1}^{N} \mathbb{1}\{y_k = k\} \Sigma_k^{-1}\boldsymbol{\mu}_k$$

$$\sum_{i=1}^{N} \mathbb{1}\{y_k = k\}\Sigma_k^{-1}\mathbf{x}_i = \sum_{i=1}^{N} \mathbb{1}\{y_k = k\}\Sigma_k^{-1}\boldsymbol{\mu}_k$$

$$\boldsymbol{\mu}_k = \frac{\sum_{i=1}^{N} \mathbb{1}\{y_k = k\}\mathbf{x}_i}{\sum_{i=1}^{N} \mathbb{1}\{y_k = k\}}$$

To solve for the covariance

$$-\sum_{i=1}^{N} \mathbb{1}\{y_k = k\} \log{(2\pi)^{\frac{d}{2}}|\Sigma_k|^{\frac{1}{2}}} - \sum_{i=1}^{N} \mathbb{1}\{y_k = k\}\frac{1}{2}(x_i - \mu_{ki})^T\Sigma_k^{-1}(x_i - \mu_{ki}) + \log\frac{1}{10}$$

taking the derivative w.r.t Σ_{ki}^{-1} and setting it to zero

$$0 = -\sum_{i=1}^{N} \mathbb{1}\{y_k = k\}\frac{\partial \log{(2\pi)^{\frac{d}{2}}|\Sigma_k|^{\frac{1}{2}}}}{\partial\Sigma_k^{-1}} - \sum_{i=1}^{N} \mathbb{1}\{y_k = k\}\frac{1}{2}(\mathbf{x}_i - \boldsymbol{\mu}_k)(\mathbf{x}_i - \boldsymbol{\mu}_k)^T$$

$$0 = \sum_{i=1}^{N} \mathbb{1}\{y_k = k\}(2\pi)^{-\frac{d}{2}}|\Sigma_k|^{-\frac{1}{2}}(2\pi)^{\frac{d}{2}}\frac{\partial|\Sigma_k^{-1}|^{-\frac{1}{2}}}{\partial\Sigma_k^{-1}} - \sum_{i=1}^{N} \mathbb{1}\{y_k = k\}\frac{1}{2}(\mathbf{x}_i - \boldsymbol{\mu}_k)(\mathbf{x}_i - \boldsymbol{\mu}_k)^T$$

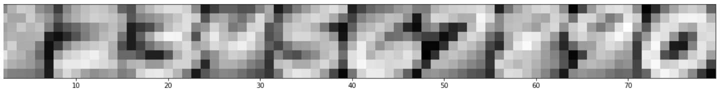
$$0 = \sum_{i=1}^{N} \mathbb{1}\{y_k = k\}|\Sigma_k^{-1}|^{\frac{1}{2}}(-\frac{1}{2})|\Sigma_k^{-1}|^{-\frac{3}{2}}|\Sigma_k^{-1}|\Sigma_k^T - \sum_{i=1}^{N} \mathbb{1}\{y_k = k\}\frac{1}{2}(\mathbf{x}_i - \boldsymbol{\mu}_k)(\mathbf{x}_i - \boldsymbol{\mu}_k)^T$$

$$0 = \sum_{i=1}^{N} \mathbb{1}\{y_k = k\}(-\frac{1}{2})\Sigma_k - \sum_{i=1}^{N} \mathbb{1}\{y_k = k\}\frac{1}{2}(\mathbf{x}_i - \boldsymbol{\mu}_k)(\mathbf{x}_i - \boldsymbol{\mu}_k)^T$$

$$\Sigma_k = \frac{\mathbb{1}\{y_k = k\}\sum_{i=1}^{N} \mathbb{1}\{y_k = k\}(\mathbf{x}_i - \boldsymbol{\mu}_k)(\mathbf{x}_i - \boldsymbol{\mu}_k)^T}{\sum_{i=1}^{N} \mathbb{1}\{y_k = k\}}$$

2.2.1

Covariance matrix diagonals (variances) for each feature



2.2.2

Training average conditional log likelihood: -0.1264
Testing average conditional log likelihood: -0.1995

2.2.3

Training accuracy: 0.98
Testing accuracy: 0.97

2.3

2.3.2

To get the MLE :

$$\ell(\mathbf{x}, \mathbf{y}|\boldsymbol{\eta}) = p(\mathbf{x}|\boldsymbol{\eta}_k, y_k = k)p(y_k = k)p\boldsymbol{\eta}_k$$

Incorporate the prior on η into the training data by adding a case of all positive and a case of all negative features to the training set.
For each class k:

$$= \prod_{i=1}^{N}\prod_{j=1}^{d} \mathbb{1}\{y_k = k\}(\eta_{jk})^{b_j}(1 - \eta_{jk})^{(1-b_j)}\frac{1}{10}$$

Taking the log

$$= \sum_{i=1}^{N}\sum_{j=1}^{d} b_j \mathbb{1}\{y_k = k\}\big\{ \log(\eta_{jk}) + (1 - b_j)\log(1 - \eta_{jk}) + \log\frac{1}{10}\big\}$$

Taking the derivative w.r.t. η_{jk} and setting it to zero

$$0 = \sum_{i=1}^{N} \mathbb{1}\{y_k = k\}\big\{b_j\frac{1}{\eta_{jk}} + (1 - b_j)\frac{1}{(1 - \eta_{jk})}(-1)\big\}$$

$$0 = \sum_{i=1}^{N} \mathbb{1}\{y_k = k\}\{\frac{b_j}{\eta_{jk}} - \frac{1}{(1-\eta_{jk})} + \frac{b_j}{(1-\eta_{jk})}\}$$

$$0 = \sum_{i=1}^{N} \mathbb{1}\{y_k = k\}\{\frac{b_j}{\eta_{jk}} - \frac{1-b_j}{(1-\eta_{jk})}\}$$

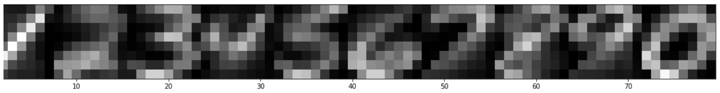
$$0 = \sum_{i=1}^{N} \mathbb{1}\{y_k = k\}\big\{b_j(1-\eta_{jk}) - \eta_{jk}(1-b_j)\big\}$$

$$0 = \sum_{i=1}^{N} \mathbb{1}\{y_k = k\}\big\{b_j - b_j\eta_{jk} - \eta_{jk} + b_j\eta_{jk}\big\}$$

$$\sum_{i=1}^{N} \mathbb{1}\{y_k = k\}\eta_{jk} = \sum_{i=1}^{N} \mathbb{1}\{y_k = k\}b_j$$

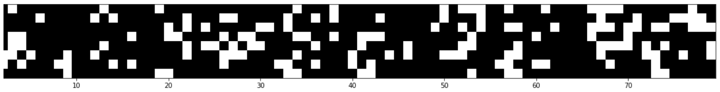
$$\eta_{jk} = \frac{\sum_{i=1}^{N} \mathbb{1}\{y_k = k\}b_j}{\sum_{i=1}^{N} \mathbb{1}\{y_k = k\}}$$

2.3.3



η for each class

2.3.4



New data point

2.3.5

Training average conditional log likelihood: -0.9437
Testing average conditional log likelihood: -0.9872

2.3.6

Training accuracy: 0.77
Testing accuracy: 0.76

2.4

All models performed quite well with the Gaussian Bayes classifier achieving the highest accuracy, and Naive Bayes the poorest. The optimal KNN classifier (k=1) achieved extremely high training accuracy, but accuracy dropped substantially on the testing data, indicative of some overfitting. KNN likely performed well because it is non-parametric and allowed for complex decision boundaries, even though it assumed features were uncorrelated. Its performance was probably limited by the high number of features relative to the number of observations, which gave optimal performance with a small k leading to some overfitting. The Gaussian Bayes classifier had very similar performance for the testing and training data indicating there was likely less overfitting. It probably performed well because it accounted for the correlations among features which were likely strong when features are adjacent pixels from images of handwritten digits. Its assumption that classes follow a multivariate Gaussian was likely justified in the MNIST dataset, and its quadratic decision boundaries allowed for more flexibility relative to a linear decision boundary. Naive Bayes also had similar performance for training and testing indicating there was little overfitting, but overall performance was poor. Poor performance was likely due to the assumption that features are uncorrelated was strongly violated in this dataset. Assuming linear decision boundaries likely further limited performance. Generally, the observed results make sense given the properties and assumptions of each algorithm,