

CSC411 Fall 2017

Assignment 2 Report

Tianbao Li

2017/11/04

1 Q1: Class-conditional gaussians

Given:

$$p(y = k) = a_k \quad (1)$$

$$p(\mathbf{x}|y = k, \mu, \sigma) = \left(\prod_{i=1}^d 2\pi\sigma_i^2\right)^{-1/2} \exp\left\{-\sum_{i=1}^d \frac{1}{2\sigma_i^2} (x_i - \mu_{ki})^2\right\} \quad (2)$$

1.1 Bayes' rule derivation

$$\begin{aligned} p(y = k|\mathbf{x}, \mu, \sigma) &= \frac{p(\mathbf{x}|y = k, \mu, \sigma)p(y = k)}{p(\mathbf{x}|\mu, \sigma)} \\ &= \frac{p(\mathbf{x}|y = k, \mu, \sigma)p(y = k)}{\sum_{j=1}^K p(\mathbf{x}|y = j, \mu, \sigma)} \\ &= \frac{(\prod_{i=1}^d 2\pi\sigma_i^2)^{-1/2} \exp\{-\sum_{i=1}^d \frac{1}{2\sigma_i^2} (x_i - \mu_{ki})^2\} a_k}{\sum_{j=1}^K (\prod_{i=1}^d 2\pi\sigma_i^2)^{-1/2} \exp\{-\sum_{i=1}^d \frac{1}{2\sigma_i^2} (x_i - \mu_{ji})^2\}} \end{aligned}$$

1.2 Negative likelihood function (NLL)

$$\begin{aligned} \ell(\theta; D) &= -\log p(y^{(1)}, \mathbf{x}^{(1)}, y^{(2)}, \mathbf{x}^{(2)}, \dots, y^{(N)}, \mathbf{x}^{(N)}|\theta) \\ &= -\log \prod_{n=1}^N p(y^{(n)}, x^{(n)}|\theta) \\ &= -\sum_{n=1}^N (\log p(x^{(n)}|y^{(n)}, \theta) + \log p(y^{(n)}|\theta)) \\ &= -\sum_{n=1}^N (\log((\prod_{i=1}^d 2\pi\sigma_i^2)^{-1/2} \exp\{-\sum_{i=1}^d \frac{1}{2\sigma_i^2} (x_i^{(n)} - \mu_{ki})^2\}) + \log \alpha_k) \\ &= -\sum_{n=1}^N (-\frac{1}{2} \sum_{i=1}^d 2\pi\sigma_i^2 - \sum_{i=1}^d \frac{1}{2\sigma_i^2} (x_i^{(n)} - \mu_{ki})^2 + \log \alpha_k) \\ &= \sum_{n=1}^N (\sum_{i=1}^d \pi\sigma_i^2 + \sum_{i=1}^d \frac{1}{2\sigma_i^2} (x_i^{(n)} - \mu_{ki})^2 - \log \alpha_k) \end{aligned}$$

1.3 Partial derivatives

$$\begin{aligned}
\frac{\partial \ell}{\partial \mu_{ki}} &= \frac{\partial (\sum_{n=1}^N \sum_{i=1}^d \frac{1}{2\sigma_i^2} (x_i^{(n)} - \mu_{ki})^2)}{\partial \mu_{ki}} \\
&= - \sum_{n=1}^N \sum_{i=1}^d \frac{1}{\sigma_i^2} \mathbb{1}(y^{(n)} = k) (x_i^{(n)} - \mu_{ki}) \\
\frac{\partial \ell}{\partial \sigma_i^2} &= \frac{\partial (\sum_{n=1}^N (\sum_{i=1}^d \pi \sigma_i^2 + \sum_{i=1}^d \frac{1}{2\sigma_i^2} (x_i^{(n)} - \mu_{ki})^2))}{\partial \sigma_i^2} \\
&= \sum_{n=1}^N \sum_{i=1}^d \mathbb{1}(y^{(n)} = k) (2\pi - \frac{1}{2\sigma_i^4} (x_i^{(n)} - \mu_{ki})^2)
\end{aligned}$$

1.4 Estimation

$$\begin{aligned}
\frac{\partial \ell}{\partial \mu_{ki}} &= 0 \\
\Rightarrow \mu_{ki} &= \frac{\sum_{n=1}^N \sum_{i=1}^d \mathbb{1}(y^{(n)} = k) x_i^{(n)}}{\sum_{n=1}^N \sum_{i=1}^d \mathbb{1}(y^{(n)} = k)} \\
\frac{\partial \ell}{\partial \sigma_i^2} &= 0 \\
\Rightarrow \sigma_i &= \sqrt[4]{\frac{\sum_{n=1}^N \sum_{i=1}^d \mathbb{1}(y^{(n)} = k) * (x_i^{(n)} - \mu_{ki})^2}{\sum_{n=1}^N \sum_{i=1}^d \mathbb{1}(y^{(n)} = k) * 4\pi}}
\end{aligned}$$

2 Handwritten digit classification

In this assignment, we use the dataset of handwritten digit. To get a macro view of the data, here we plot the means for each data classes in the training set, shown in Figure 1.

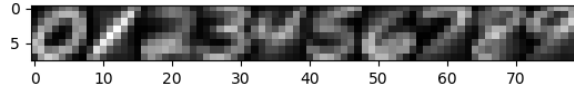


Figure 1: Mean for each class

2.1 K-NN classifier

First, we build a K nearest neighbor classifier using Euclidean distance on the data.

2.1.1 Accuracy for certain K values

- for K = 1
 - train classification accuracy: 1.0
 - test classification accuracy: 0.96875
- for K = 15

- train classification accuracy: 0.961
- test classification accuracy: 0.959

2.1.2 Ties solution

During the implementation, one important problem is that ties usually happens and need to be broken. The solution is shown as follows.

```
def query_knn(self , test_point , k):
    digit = None
    dist = self.l2_distance(test_point)
    noTies = False
    while noTies == False:
        mink = np.argpartition(dist , k)[:k]
        counts = np.bincount(self.train_labels[mink].astype(int))
        digit = np.argmax(counts == np.amax(counts))
        if len(digit) > 1:
            k = k - 1
        else:
            noTies = True
    return digit
```

While running KNN, if ties happen (which means two classes has the same maximum count), we try to decrease k by 1 and run the KNN again until no ties. As the class label should focus on one or a few classes, ties are unlikely to happen when $K = 1$, so this should be a solution to get an answer without ties.

2.1.3 Best k

Here we implement 10 fold cross validation in range 1-15 to find the optimal K. By finding minimum validation error, optimal K for this problem is that $K = 3$ and accuracies are shown as follows.

- train classification accuracy: 1.0
- validation classification accuracy: 0.96514286
- test classification accuracy: 0.96875