ENGR421 - Homework 1

Zeynep Öner, ID: 64912

March 19, 2021

Naïve Bayes' Classifier

This homework was an implementation of the Naïve Bayes Classifier in Python. After importing our image and label data sets and dividing them up to training and data sets, we calculate the needed parameters. We first calculate the sample means as:

$$\hat{\mu}_c = \frac{\sum_{i=1}^{N} x_i}{N} \tag{0.1}$$

The sample deviations are calculated as such:

$$\hat{\Sigma}_c = \sqrt{E[(X - E[X])^2]} \tag{0.2}$$

Here, I used the fact that the sample deviations are the square root of the variance.

Finally, we calculate the prior probabilities as:

$$P(y = c) = \frac{\texttt{number of samples with class c}}{\texttt{total number of samples}} \tag{0.3}$$

With these parameters, we can calculate the score function. As the section 5.7 in the textbook states, the score function can be calculated using the mean and the prior probabilities:

$$g_i(x) = \sum_j [x_j \log(p_{ij}) + (1 - x_j) \log(1 - p_{ij})] + \log(P(c_i)) \tag{0.4}$$

This results in a (200, 2) matrix in our problem, as the sets have 400 elements and we have 2 classes. We calculate the score functions for the training and test sets. After the data sets are generated, we calculated the predicted values by using the `np.argmax` function. We add `axis=1` as argument, because we want to calculate along the second axis.

The result is a list with 200 elements. The only thing left is to compare it with the true y values, which are the label sets for the corresponding image sets. We can perform this comparison with a confusion matrix, and this can be done with the same method we used in the lab (`pd.crosstab`). The confusion matrix outputs can be seen below:

```
Confusion matrix for the training set:
y_train   1    2
y_hat
1        15   28
2         5  152

Confusion matrix for the test set:
y_test   1    2
y_hat
1       13   28
2        7  152
```

The confusion matrix output is very close to the confusion matrix output seen in the homework description. Seeing as the parameters are also very close to the ones seen in the description, we can conclude that the algorithm works and the performance is agreeable.