Project 1: Density Estimation and Classification

Youmi Koh

Aug 30, 2023

This project involves implementing supervised, unsupervised, and deep learning techniques for density estimation and classification. The project focuses on a subset of the MNIST dataset containing images of digits ”0” and ”1”, and demonstrates feature extraction, parameter calculation, implementation of Naïve Bayes classifiers, and prediction of labels for the test data using the classifiers. Finally, accuracy of the predictions are calculated and observations are presented.

Maximum Likelihood Estimation

The average brightness (`f1`) and standard deviation of brightness (`f2`) of images are extracted as features from the training data. We assume that these two features are independent and that each image is drawn from a normal distribution. Images depict either the digit zero (`d0`) or one (`d1`), and these digit labels are used to estimate the parameters of a normal distribution (`mean`, `var`) for each feature/digit pair.

Likelihood function for a normal distribution: $L(\mu, \sigma) = p(X \mid \mu, \sigma) = \left(\frac{1}{\sigma\sqrt{2\pi}}\right)^n \prod_i^n \exp\left(-\frac{(x_i-\mu)^2}{2\sigma^2}\right)$

Log-likelihood function for convenience: $l(\mu, \sigma) = \log p(X \mid \mu, \sigma) = -n\log(\sigma\sqrt{2\pi}) \sum_i^n \exp\left(-\frac{(x_i-\mu)^2}{2\sigma^2}\right)$

Maximum likelihood estimates for mean μ and variance σ^2:

$$\hat{\mu} = \underset{\mu}{\operatorname{argmax}}\ l(\mu, \sigma) = \frac{1}{n}\sum_i^n x_i \qquad \hat{\sigma} = \underset{\sigma}{\operatorname{argmax}}\ l(\mu, \sigma) \ \Rightarrow\ \widehat{\sigma^2} = \frac{1}{n}\sum_i^n (x_i - \mu)^2$$

In the results below, the mean of `f1` (average brightness) is notably higher for `d0` than `d1`, which suggests the images of zeroes are generally brighter than images of ones. The mean of `f2` (standard deviation of brightness) is also significantly higher for `d0` than `d1`, indicating that the brightness values within each image of zeroes vary more widely. Variance for `d0` is larger than `d1` for both features, and this may impact the derived classifier.

```
    Mean_of_feature1_for_digit0: 44.20389362244898
    Variance_of_feature1_for_digit0: 116.7009513315956
    Mean_of_feature2_for_digit0: 87.41907699909902
    Variance_of_feature2_for_digit0: 102.46490855207279

    Mean_of_feature1_for_digit1: 19.43962244897959
    Variance_of_feature1_for_digit1: 31.98890083543315
    Mean_of_feature2_for_digit1: 61.44770668123185
    Variance_of_feature2_for_digit1: 83.54555141482662
```

Naive Bayes Classification

The above estimates serve parameters in the normal probability density functions that calculate conditional probabilities for naive Bayes classification.

$$p(\texttt{digit} \mid \texttt{features}) = \frac{p(\texttt{digit}) \cdot p(\texttt{features} \mid \texttt{digit})}{p(\texttt{features})} \Leftrightarrow \text{posterior} = \frac{\text{prior} \times \text{likelihood}}{\text{evidence}}$$

Although `f1` and `f2` are not necessarily independent, we assume that these features are conditionally independent given a class `d`. This assumption allows us to simplify the likelihood function to the product of the likelihoods of each feature given the digit.

$$p(\texttt{f}_i \mid \texttt{f}_j, \texttt{d}_k) = p(\texttt{f}_i \mid \texttt{d}_k) \Rightarrow \quad \begin{aligned} p(\texttt{d}_k \mid \texttt{f1}, \texttt{f2}) &\propto p(\texttt{d}_k, \texttt{f1}, \texttt{f2}) \\ &\propto p(\texttt{d}_k) \cdot p(\texttt{f1} \mid \texttt{d}_k) \cdot p(\texttt{f2} \mid \texttt{d}_k) \end{aligned} \quad k \in \{0, 1\}$$

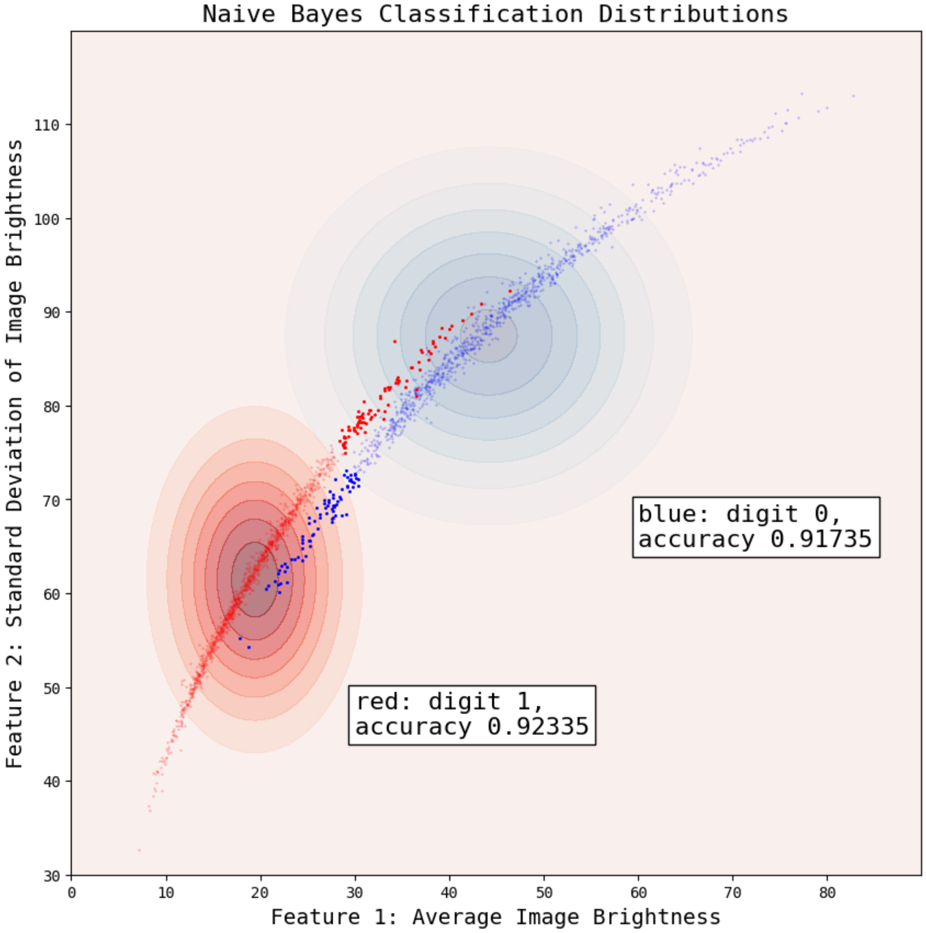
Prior probabilites $p(\texttt{d}_k)$ are derived from the training data, and the likelihood conditional probabilities $p(\texttt{f}_i \mid \texttt{d}_k)$ are calculated using the estimated normal distribution parameters $\texttt{mean}_{(\texttt{f};\texttt{d})}$ and $\texttt{var}_{(\texttt{f};\texttt{d})}$. Then posterior probabilities are calculated for each digit, and the digit with the highest probability is selected as the classification.

Accuracy

Despite the naive assumption, the classifier performs well on the test data.

```
Accuracy_for_digit0testset: 0.9173469387755102
Accuracy_for_digit1testset: 0.9233480176211454
```

`d1` slightly outperforms `d0` as a result of the smaller MLE variances. The below plot illustrates density contours for digits (0,1) learned from the training set, and highlights the overlay of test data. We can see `d1` density is more concentrated, and the spread of test set data that was misclassified.


Naive Bayes Classification Distributions
Feature 2: Standard Deviation of Image Brightness
Feature 1: Average Image Brightness
110
100
90
80
70
60
50
40
30
0
10
20
30
40
50
60
70
80
blue: digit 0,
accuracy 0.91735
red: digit 1,
accuracy 0.92335