CSE 575 - Statistical Machine Learning

Project 1: Density Estimation and Classification

Skanda Suresh
MS in Computer Science
ASU Id: 1217132644

Problem Statement

This project attempts to perform parameter estimation for the given dataset (which is a subset from the MNIST dataset). The MNIST dataset contains 70,000 images of handwritten digits, divided into 60,000 training images and 10,000 testing images. We use only images for digit "7" and digit "8" in this project. Further after estimating the parameters, the Naive Bayes and Logistic Regression algorithms are implemented on this dataset. Dataset statistics are shown below.

	Train	**Test**
Class 7	6265	1028
Class 8	5851	974
TOTAL	**11216**	**2002**

Feature Extraction

The given dataset consists of images of classes **7** and **8.** Each image is represented as a vector of size **784**. The training set is of dimension **11216 x 784.** The train labels are either **0 (class 7)** or **1 (Class 8),** and it's dimensions are **11216 x 1.** The mean (μ) and standard deviation (σ) of each image are extracted, and these are used as the features for the experiments below. The mean is calculated by taking the row-wise sum and dividing it by the number of features (784).
Standard deviation is calculated by taking the square root of the sum of squares of the observations and the mean as shown below and dividing by N.

$$\mu = \frac{\sum_{i=0}^{N} x_i}{N}, \quad \sigma = \frac{\sqrt{\sum_{i=0}^{N} (x_i - \mu)^2}}{N}$$

This results in the number of features being reduced from 784 to 2. We call the training features, test features, train labels and test labels as **X, X_test, y, and y_test** respectively for the rest of this report.

Maximum Likelihood Estimation

Maximum likelihood estimation involves estimating the parameters that describe the distribution of data. Thus given and image **X** and the corresponding label **y,** we would like to estimate the density that **X** belongs to label **y,** i.e

$$P_{\theta}(X \mid y)$$

Here θ is the parameter we try to estimate. The above equation is the likelihood equation and we estimate θ by maximizing the Log-Likelihood, i.e,

$$\widehat{\theta} = argmax\,(P(X \mid \theta)\,.$$

For the purpose of this project, the features are assumed to be independent and follow a 2D Normal distribution. Thus the parameters for a normal distribution are it's mean and standard deviation, μ and σ. The formulation of a bivariate normal distribution is shown below.

$$P(x;\ \mu;\ \Sigma) = \frac{1}{\sqrt{2\Pi\Sigma}^{1/2}} exp(-\frac{1}{2}(x - \mu)^T \Sigma^{-1}(x - \mu)$$

The parameters μ and Σ are estimated as:

$$\mu = \frac{\sum_{i=0}^{N} x_i}{N}\ ,\ \Sigma = cov(x,\ y) = \frac{\Sigma(x - \bar{x})(y - \bar{y})}{N}$$

Here x and y refer to the 2 features

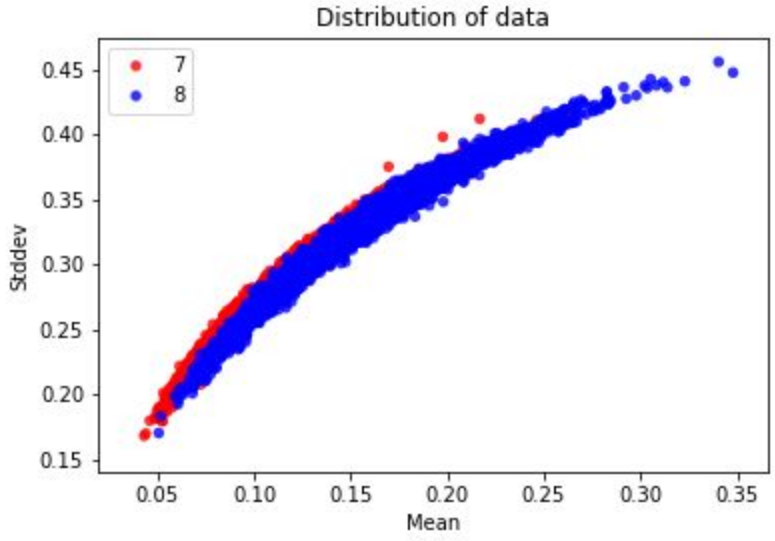
Results

In our case, we estimate parameters for digits 7 and 8 separately. The results are as shown below.

	(μ_1, μ_2)	$(\sigma_1,\ \sigma_2)$
Digit 7	[0.1145277 , 0.28755657]	[0.0306324 , 0.03820108],
Digit 8	[0.15015598, 0.32047584]	[0.03863249, 0.03996007]

Here $(\sigma_1,\ \sigma_2)$ form the square root of the diagonal of Σ, since features are independent the correlation between the features is 0.

The data distribution is as below



Naive Bayes

Naive Bayes algorithm works on the principle of the Bayes theorem. It makes a naive assumption on the conditional independence between every pair of features. It predicts the label *y* given a set of independent features, which in our case are x_1 and x_2. The general formulation of the Bayes theorem is as follows.

$$P(y \mid x_{1,\ldots\ldots} x_n) = \frac{P(y) \prod_{i=1}^{n} P(x_i \mid y)}{P(x_{1,\ldots\ldots} x_n)}$$

This probability is estimated for each class (y = 0 and y = 1 in our case), and the class resulting in largest probability is assigned as the predicted class. If we observe we notice that the denominator is the same for each class, hence the resultant class is proportional to the numerator. Thus we assign the predicted class as:

$$\widehat{y} = argmax_{(y)} \; P(y) \prod_{i=1}^{n} P(x_i \mid y)$$

We assume that the data follows a 2D normal distribution, thus to estimate $\prod_{i=1}^{n} P(x_i \mid y)$ we make use of the gaussian distribution function. It is worth noting that since the features are assumed to independent, we may break down the bivariate distribution as a product of 2 univariate distributions as shown below:

$$P(x;\ \mu_1;\ \mu_2;\ \sigma_1;\ \sigma_1) = \frac{1}{\sqrt{2\Pi\sigma_1^2}} exp\left(\frac{-(x_1-\mu_1)^2}{2\sigma_1^2}\right) \frac{1}{\sqrt{2\Pi\sigma_2^2}} exp\left(\frac{-(x_2-\mu_2)^2}{2\sigma_2^2}\right)$$

Thus this allows us to treat each feature as a univariate Gaussian distribution, and we may calculate the combined probability by multiplying each features univariate Probability Distribution Function Value.

Results

We make use of the parameters estimated using Maximum Likelihood Estimation as described in the above section. The Naive Bayes classifier gives an accuracy of **69.53%** on the test data. The statistics for each class are shown below.

	Digit 7	**Digit 8**	**Total**
N_Samples	781 / 1028	611 / 974	1392 / 2002
Accuracy	75.97%	62.7%	69.53%

Logistic Regression

Logistic regression is a discriminative classifier. It tries to model the probability $P(y \mid X)$ directly without making any assumptions about the distribution of *X.* The Logistic Regression classifier attempts to model the output *y* as a linear combination of its features *X.* In our case there are only 2 classes, hence it is a binary classification task. The output is converted into the probabilistic domain of [0, 1] using the sigmoid function.

$$h_\theta(x) = sigmoid\,(Z)$$

$$\textit{Where}\quad Z = WX + b$$

The sigmoid function is modeled as follows:

$$Sigmoid\,(Z) = \frac{1}{1 + exp(-Z)}$$

We need to compute the optimal values for the weights *W.* This is done using gradient ascent. In gradient ascent, we define a loss function, which is a function of the predicted outputs $\widehat{y}$ and y*.* We make use of the log loss function defined below.

$$L(y, \widehat{y}) = ylog(\widehat{y}) + (1-y)log(1-\widehat{y})$$

We attempt to maximize this function by differentiating it with respect to *W* and computing the gradients. The gradients give us the direction in which we need to move in order to maximize the loss function. Upon differentiating we compute the gradients and update the weights as follows. Note that prior to training we append a column of ones to the feature matrix X, to account for bias term *b.*

$$\Delta(W) = X^T(y - \widehat{y})\ ,\ \Delta\ refers\ to\ gradient$$

$$W^{t+1} = W^t + \eta.\Delta(W),\ where\ \eta\ is\ the\ learning\ rate$$

We repeat the above equation until the Loss function converges or a prespecified number of iterations is met.

Results

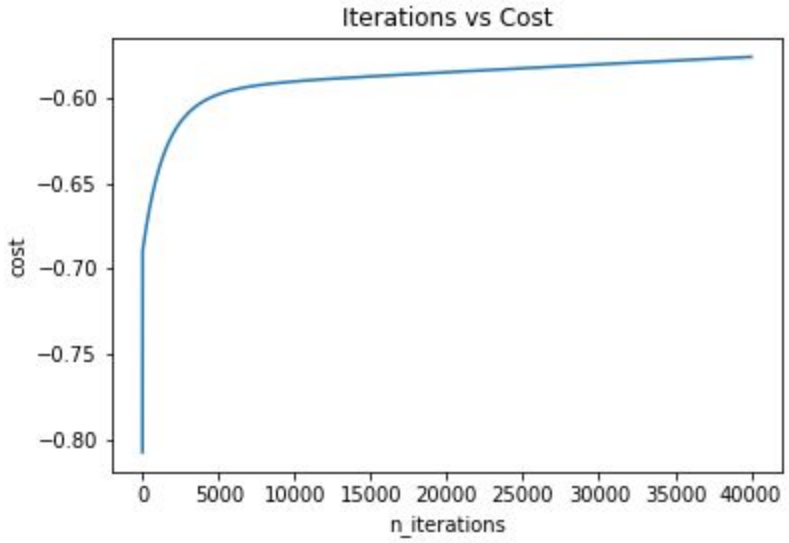
We present results for logistic regression below. Training is done for different learning rates (η) and iterations (*n_iter*). We present results for $\eta = 0.5$ and *n_iter = 40000* below.

	Digit 7	**Digit 8**	**Total**
N_Samples	776 / 1028	645 / 974	1421 / 2002
Accuracy	75.48%	66.22%	70.97%

The weights are initialized randomly. The coefficients for the regression equation are as shown below.

W1	**W2**	**b**
26.82897056	0.02775092	-3.589252241

We notice the cost function converges at a value of **- 0.5756.** The convergence plot is as below.



Conclusions

We notice that there is not a huge difference in accuracies between Naive Bayes and Logistic regression. Logistic regression outperforms Naive Bayes when it comes to predicting the Digit 8, whereas the predictions for Digit 7 are similar. The moderate accuracy may be due to the fact that we are only using 2 features (mean and std). Applying preprocessing techniques like normalization and binning may yield better performance.