Name: Arpit Parasottambhai Prajapati
Email: aprajap2@asu.edu
ASU ID: 1219477402

Project 1: Naïve Bayes and Logistic Regression

Two features for each image:

1. x_1: The average of all pixel values in the image
2. x_2: The standard deviation of all pixel values in the image

The formula that I used to estimate the parameters for the 2-D normal distribution for each class, using the training data:

$$l(\Sigma|\mathcal{D}) = -\frac{N}{2}\log|\Sigma| - \frac{1}{2}\sum_{i=1}^{N}(x_i-\mu)^T\Sigma^{-1}(x_i-\mu)$$

Taking derivative with respect to μ :

$$\frac{\partial l}{\partial \mu} = \sum_{i=1}^{N}(x_i-\mu)^T\Sigma^{-1}$$

Setting it as 0 will get us value of μ as:

$$\hat{\mu}_{ML} = \frac{1}{N}\sum_{i=1}^{N}x_i$$

Similarly taking derivative with respect to Σ and setting it to zero we get:

$$\hat{\Sigma}_{ML} = \frac{1}{N}\sum_{i=1}^{N}(x_i-\mu_{ML})(x_i-\mu_{ML})^T$$

If we have d dimensional input vector, the $\hat{\mu}_{ML}$ is d dimensional vector and $\hat{\Sigma}_{ML}$ is $d \times d$ dimensional matrix.

The estimated values for the parameters for both classes:

x_1 : The average value of all pixel values in training images.

x_2 : The standard deviation of all pixel values in training images.

Estimated value of μ:

	x_1	x_2
$Y = 0$	0.32560777	0.32003609
$Y = 1$	0.22290531	0.33394171

Covariance Matrix for $Y = 0$:

0.01285601	0.00897912
0.00897912	0.00774227

Covariance Matrix for $Y = 1$:

0.00324396	0.00253005
0.00253005	0.00325322

If we treat x_1 and x_2 as independent variable which is the requirement of Naïve Bayes, the μ per class, per feature remains the same but variance will become a d dimensional vector for each class.

σ^2	x_1	x_2
$Y = 0$	0.01285387	0.00774097
$Y = 1$	0.00324342	0.00325268

The expression for the estimated normal distribution. the expression for the estimated normal distributions, an explanation for how the distributions are used in classifying a testing sample.

Naïve Bayes

$$P(y|x) = \frac{P(y,x)}{p(x)}$$

$$\therefore P(y,x) = P(y)P(x|y)$$

$$\therefore P(y|x) = \frac{P(y)P(x|y)}{p(x)}$$

$$\therefore P(y|x) \propto P(y)P(x|y)$$

$$\therefore \hat{y} = argmax_y(P(y)P(x|y))$$

Here we have 2-dimensional input vector thus $P(x|y)$ can be written as follows: $d = 2$

$$P(x|y) = \prod_{i=1}^{d} P(x_i|y)$$

$$\hat{y} = argmax_y(P(y)\prod_{i=1}^{d} P(x_i|y))$$

We can take log of the above expression to make calculations easy.

$$\hat{y} = \underset{y}{argmax}\left[\, log(P(y)) + \sum_{i=1}^{d} log(P(x_i|y))\right]$$

Here $P(y)$ is a prior. Which can be calculated as follow:

$$P(y = i) = \frac{Number\ of\ traning\ examples\ having\ class = i}{Total\ training\ examples}$$

In out input, we have:

Number of classes = 2

Training examples per class = 6000

Number of training examples = 12000

For this project, $P(y = 0) = 0.5$ and $P(y = 1) = 0.5$

And $log(P(y = 0)) = log(P(y = 1)) = -0.6931471805599453$

$P(x_i|y)$ is the **maximum likelihood** of parameter x_i.

Calculation of **Estimated Normal Distribution** $log(P(x_i|y))$:

$$log\left(P\left(x_i\middle|y_j\right)\right) = log\left[\frac{1}{\sqrt{2\pi{\sigma_{ij}}^2}}e^{-\frac{(x_i-\mu_{ij})^2}{2{\sigma_{ij}}^2}}\right]$$

$$log\left(P\left(x_i\middle|y_j\right)\right) = -\frac{1}{2}log\left(2\pi{\sigma_{ij}}^2\right) - \frac{1}{2}\frac{(x_i-\mu_{ij})^2}{{\sigma_{ij}}^2}$$

$$\hat{y} = \underset{y_j}{argmax}\left[\, log(P(y_j)) + \sum_{i=1}^{d} -\frac{1}{2}log\left(2\pi{\sigma_{ij}}^2\right) - \frac{1}{2}\frac{(x_i-\mu_{ij})^2}{{\sigma_{ij}}^2}\right]$$

Here μ_j, x_i, σ_j are constants. Shape matches so we can do simple division or subtraction.

Logistic Regression

The sigmoid function $\sigma(t)$ can be written as follow:

$$\sigma(t) = \frac{e^t}{1+e^t}$$

$$P(Y=0|x) = \sigma(w^T x)$$

$$P(Y=1|x) = 1 - \sigma(w^T x)$$

Likelihood can be defined as:

$$l(w) = P(y|x) = (\hat{y})^y (1-\hat{y})^{1-y} \qquad where\ \hat{y} = \sigma(w^T x)$$

To make computation easier, we can take loglikelihood.

$$log\big(l(w)\big) = log((\sigma(w^T x))^y (1-\sigma(w^T x))^{1-y})$$

$$log\big(l(w)\big) = ylog\big(\sigma(w^T x)\big) + (1-y)log(1-\sigma(w^T x))$$

The derivative of log-likelihood with respect to w is:

$$\nabla_{w_j} log(l(w)) = \nabla_{w_j}(ylog\big(\sigma(w^T x)\big) + (1-y)log(1-\sigma(w^T x)))$$

$$\nabla_{w_j} log\big(l(w)\big) = y\frac{\partial}{\partial w_j} log(\sigma(w^T x) \ + \ (1-y)\frac{\partial}{\partial w} log(1-\sigma(w^T x)))$$

$$\nabla_{w_j} log\big(l(w)\big) = [y - \sigma(w^T x)]\, x_j$$

Now we can update the parameter w as follows:

$$w_j^{(k+1)} = \ w_j^{(k)} + \eta \nabla_{w_j^{(k)}} \log\,(l(w))$$

For bias term or w_0, we can update it as:

$$w_0^{(k+1)} = w_0^{(k)} + \eta[y - \sigma(w^T x)]$$

Learning rate (η)=**5** and Iterations to update weights = **25000**

Accuracy matrix:

Output Class	Naïve Bayes	Logistic Regression
Y=0	**0.784**	**0.917**
Y=1	**0.879**	**0.926**