

# CSE 6242 - Data and Visual Analytics

## HW3: Logistic Regression

Ebeid ElSayed - Ebeid@gatech.edu

March 29, 2017

### 0. Data Preprocessing

*a. Download the CSV files for the provided dataset.*

*b. Read  $mnist_{train}.csv$  and  $mnist_{test}.csv$  separately.*

```
> setwd("C:/Users/eelsayed/Google Drive/CSE6242")
> rawDataLoaded <- TRUE
> if(file.exists("mnist_train.csv")){
+   train <- read.csv(file="mnist_train.csv", header = FALSE)
+ }else{
+   rawDataLoaded <- FALSE
+ }
> if(file.exists("mnist_test.csv")){
+   test <- read.csv(file="mnist_test.csv", header = FALSE)
+ }else{
+   rawDataLoaded <- FALSE
+ }
> if(!rawDataLoaded){
+   print("Data wasn't loaded correctly.")
+ }
> train <- as.data.frame(t(train))
> names(train)[785] <- "Label"
> test <- as.data.frame(t(test))
> names(test)[785] <- "Label"
>
```

*c. Partition the training set for classification of 0, 1 and 3, 5 classes based on the class label (last row 785) :  $train_{01}$ ,  $train_{35}$ .*

```
> train_0_1 <- train[(train$Label == 0) | (train$Label == 1),]
> train_3_5 <- train[(train$Label == 3) | (train$Label == 5),]
```

*d. Do the same for the test set :  $test_{01}$ ,  $test_{35}$ .*

```
> test_0_1 <- test[(test$Label == 0) | (test$Label == 1),]
> test_3_5 <- test[(test$Label == 3) | (test$Label == 5),]
```

*e. Separate the class label from all the partitions created (remove row 785 from the actual data and store it as a separate vector).*

```

> true_label_train_0_1 <- train_0_1$Label
> train_0_1 <- subset(train_0_1, select = names(train_0_1) != "Label" )
> true_label_train_3_5 <- train_3_5$Label
> train_3_5 <- subset(train_3_5, select = names(train_3_5) != "Label" )
> true_label_test_0_1 <- test_0_1$Label
> test_0_1 <- subset(test_0_1, select = names(test_0_1) != "Label" )
> true_label_test_3_5 <- test_3_5$Label
> test_3_5 <- subset(test_3_5, select = names(test_3_5) != "Label" )

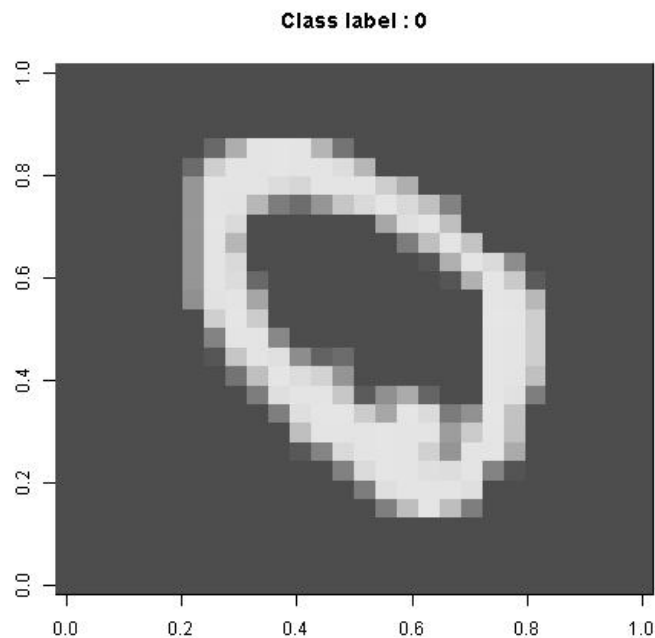
```

*g. Visualize image from each class to ensure you have read in the data correctly.*

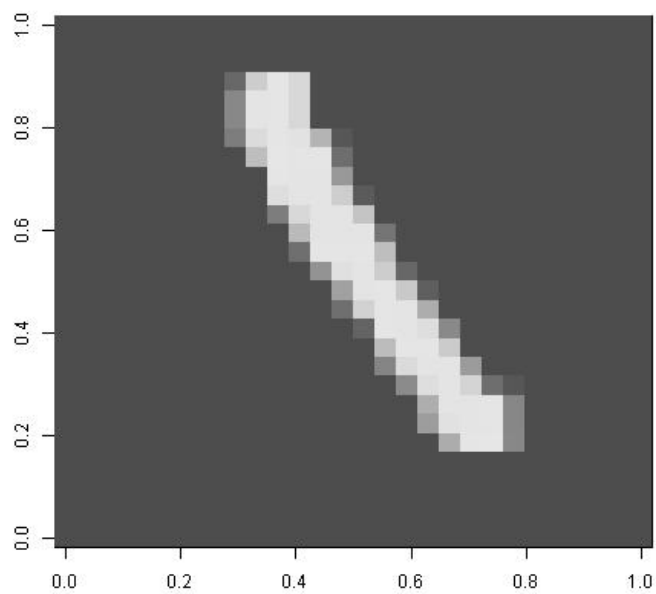
```

> save_digit_image <- function(df, digitClass, imageTitle, fileName) {
+   tmp <- df[df$Label == digitClass,]
+   m <- matrix(unlist(tmp[1,1:784])), ncol = 28, byrow = TRUE)
+
+   jpeg(filename = fileName)
+   image(z = m, col = gray.colors(256))
+   title(main = imageTitle)
+   dev.off()
+ }
> save_digit_image(train, 0, "Class label : 0", "0.jpg")
> save_digit_image(train, 1, "Class label : 1", "1.jpg")
> save_digit_image(train, 3, "Class label : 3", "3.jpg")
> save_digit_image(train, 5, "Class label : 5", "5.jpg")

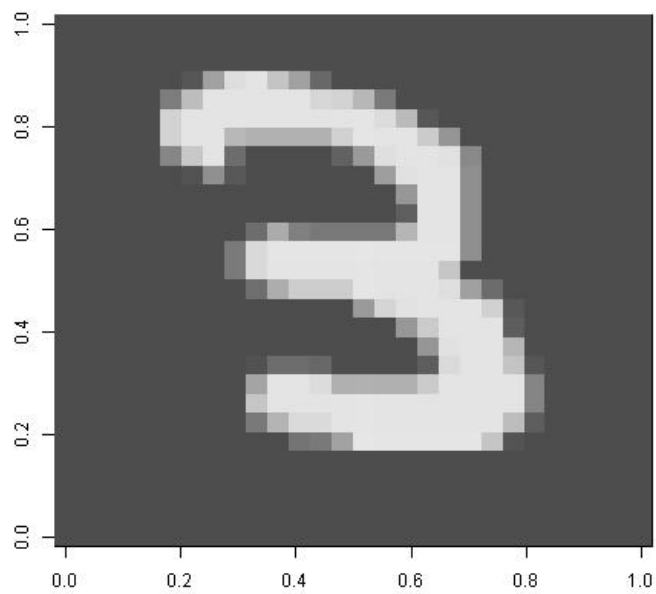
```

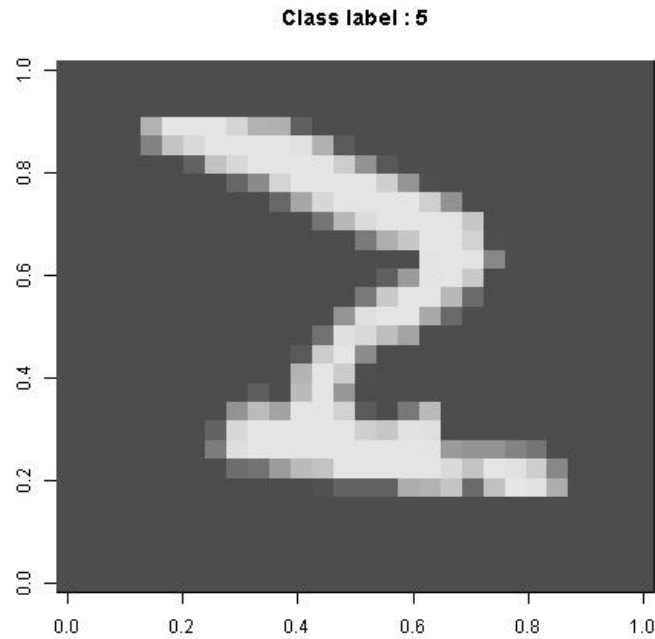


**Class label : 1**



**Class label : 3**





## 1 Theory

**a** Write down the formula for computing the gradient of the loss function used in Logistic Regression. Specify what each variable represents in the equation.

The formula for the gradient descent is:

$$\theta_j \leftarrow \theta_j - \alpha \sum_{i=1}^n \frac{1}{1 + \exp(-y^{(i)} < \theta, x^{(i)} >)}$$

where  $x^{(i)}$  is the data point represented in a vector of features,  $y^{(i)}$  is the class label, and  $\theta$  is the parameter vector. The goal is to reach the  $\theta$  that maximizes our likelihood function (given the data we use to train the model).

**b** Write pseudocode for training a model using Logistic Regression.

```

Data: Training data
1 convergence threshold:  $\eta$ 
2 step size:  $\alpha$ 
3 for  $j \leftarrow 0$  to  $d$  do
4   initialize  $\theta_j$ 
5   initialize  $\Delta\theta_j$ 
6 end
7 for  $i \leftarrow 1$  to  $n$  do
8    $x_0^{(i)} = 1$ 
9 end
10 while  $\cup_{j \in \{0,1,\dots,d\}} |\delta\theta_j/\theta_j| > \eta$  do
11   for  $i \leftarrow 1$  to  $n$  do
12      $z^{(i)} = \sum_{j=0}^d \theta_j x_j^{(i)}$ 
13   end
14   for  $j \leftarrow 0$  to  $d$  do
15      $\Delta = 0$ 
16     for  $i \leftarrow 1$  to  $n$  do
17        $\Delta \leftarrow \Delta + \frac{y^{(i)} x_j^{(i)}}{1 + \exp(-y^{(i)} z^{(i)})}$ 
18     end
19   end
20    $\theta_j \leftarrow \theta_j - \alpha \Delta$ 
21    $\delta\theta_j = \alpha \Delta$ 
22 end
23 return  $\{\theta_0, \dots, \theta_d\}$ 

```

c Calculate the number of operations per gradient descent iteration.  
Each gradient descent update iteration requires  $2n(d+1)$

## 2 Implementation

```

> # Gradient Descent
> gradient_descent <- function(X, Y, XY, theta){
+   z <- as.matrix(X) %*% theta
+   yz_exp <- 1 / (1 + exp(-Y * z))
+   # Precalculate that instead for performance
+   #tmp <- apply(X,2,function(x) X * Y)
+   delta_theta <- t(XY) %*% yz_exp
+   return(delta_theta)
+ }
> # Logistic Regression
> logistic_regression <- function(X, Y, alpha, eta, seed = 100, threshold = 100) {
+   # initializations
+   set.seed(seed)
+   X$intercept <- 1
+   theta <- rnorm(ncol(X), sd = 0.5)
+   delta_theta <- rep(10000, ncol(X))
+   numIterations <- 1

```

```

+
+   #XY <- apply(X,2,function(x) X * Y)
+   XY <- sweep(X, 1,Y, "*")
+   while (any(abs(alpha * delta_theta/(theta + 0.0001)) > eta) & numIterations < threshol
+     delta_theta <- gradient_descent(X,Y, XY,theta)
+     theta <- theta - alpha * delta_theta
+     numIterations <- numIterations + 1
+   }
+   return(theta)
+ }

```

### 3 Training

**a** Train 2 models, one on the  $\text{train}_{01}$  set and another on  $\text{train}_{35}$ , and report the training and test accuracies.

```

> # Calculate the predictions
> predict <- function(thetasFromTraining, classLabel){
+   classLabel$intercept <- 1
+   -sign(as.matrix(classLabel) %*% thetasFromTraining)
+ }
> # Calculate the accuracy
> accuracy <- function(model_function, X, Y, alpha = 0.001, eta = 0.000001, seed = 100){
+   model_output <- model_function(X, Y, alpha, eta, seed)
+   a <- predict(model_output, X)
+   tmp <- mean(Y == a)
+   return(tmp)
+ }
> accuracy(logistic_regression, train_0_1, true_label_train_0_1)
\end{Sinput}
\begin{Soutput}
[1] 0.5323332
\end{Soutput}
\begin{Sinput}
> accuracy(logistic_regression, train_3_5, true_label_train_3_5)
\end{Sinput}
\begin{Soutput}
[1] 0
\end{Soutput}
\begin{Sinput}
>
\end{Sinput}
\end{Schunk}

```