german_credit_gradient_descent

June 9, 2018

Demostration of Gradient Descent using R:
Kumar Rahul

In this exercise, we will use the German Credit Rating data to demostrate Gradient Descent Algorithm.The Dataset can be dowloaded from:

https://archive.ics.uci.edu/ml/datasets/statlog+(german+credit+data)

1 Code starts here

We are going to use below mentioned libraries for demonstrating plot the cost function:

```
In [1]: library(ggplot2)
```

1.1 Data Import and Manipulation

1.1.1 1. Import a data set

```
In [2]: raw_data <- read.csv("/Users/Rahul/Documents/Datasets/german_credit_rating_data 2010.c
                             header = TRUE,sep = ",",na.strings = c(""," ", "NA"))

        filter_data <- raw_data
        filter_data <- raw_data[complete.cases(raw_data),]
        filter_data <- raw_data[,c(-1)]
```

1.1.2 2. Create dummy variable for factors

```
In [3]: n <- ncol(filter_data)
        #for creating X0 variable
        processed_data <- cbind(rep(1, nrow(filter_data)))
        for (i in 1:n ) {
          if (is.factor(filter_data[,i])) {
            #creating dummies for the factors and storing in temp
            temp <- model.matrix(~filter_data[,i])
            #removing the first column which is an intercept term in dummy coding
            temp <- subset(temp, select = -c(1) )
            #loop through to rename the column names appropriately
              for (c in 1:ncol(temp)) {
                str <- colnames(temp)[c]
```

```
        colnames(temp)[c] <- sub("filter_data\\[, i\\]"," ",colnames(temp)[c])
        colnames(temp)[c] <- paste(colnames(filter_data)[i], "- ", colnames(temp)[c])
      }
    #the factors after dummy variable creation is appeneded to the processed_data
    processed_data <- cbind(processed_data,temp)
  }
  #in case not a factor, store the variable in processed_data directly
  else {
    processed_data <- cbind(processed_data,filter_data[,i])
    len <- ncol(processed_data)
    colnames(processed_data)[len] <- colnames(filter_data[i])

    #standardize the numeric variable for faster descent. (Feature scaling)
    processed_data[,len] <- scale(processed_data[,len])
  }
}
```

1.1.3 3. Seperate x and y variable and converting to matrix form

```
In [4]: x <- processed_data
        y <- as.matrix(x[,ncol(x)])
        colnames(y)[1] <- colnames(x)[ncol(x)]
        x <- x[,c(-ncol(x))]
        colnames(x)[1] <- "X0"
        x <- as.matrix(x)
```

1.2 Create functions

1.2.1 1. Define Sigmoid function

```
In [5]: sigmoid <- function(z) {
        1/(1 + exp(-z))
        }
```

1.2.2 2. Define Hypothesis function

```
In [6]: hypothesisLr <- function(theta, x) {
        sigmoid(x %*% theta) #matrix multiplication (x is 1000*75 and theta is 75*1, result 75
        }
```

1.2.3 3. Define Cost function

```
In [7]: computeCost <- function(theta, x, y) {
        m <- length(y)
        s <- sapply(1:m, function(i)
        y[i]*log(hypothesisLr(theta,x[i,])) + (1 - y[i])*log(1 - hypothesisLr(theta,x[i,]))
        )
        j <- -1/m * sum(s)
```

```
        return(j)
        }
```

1.2.4 4. Define Gradient Descent function

```
In [8]: gradAlgo <- function(theta, x, y) {
          m <- length(y)
          g <- 1/m * t(x) %*% (hypothesisLr(theta,x) - y)
          return(g)
        }
```

1.3 Call functions

1.3.1 1. Initialize α, θ and compute cost

Run normal gradient descent to compute theta and cost function J for each iteration

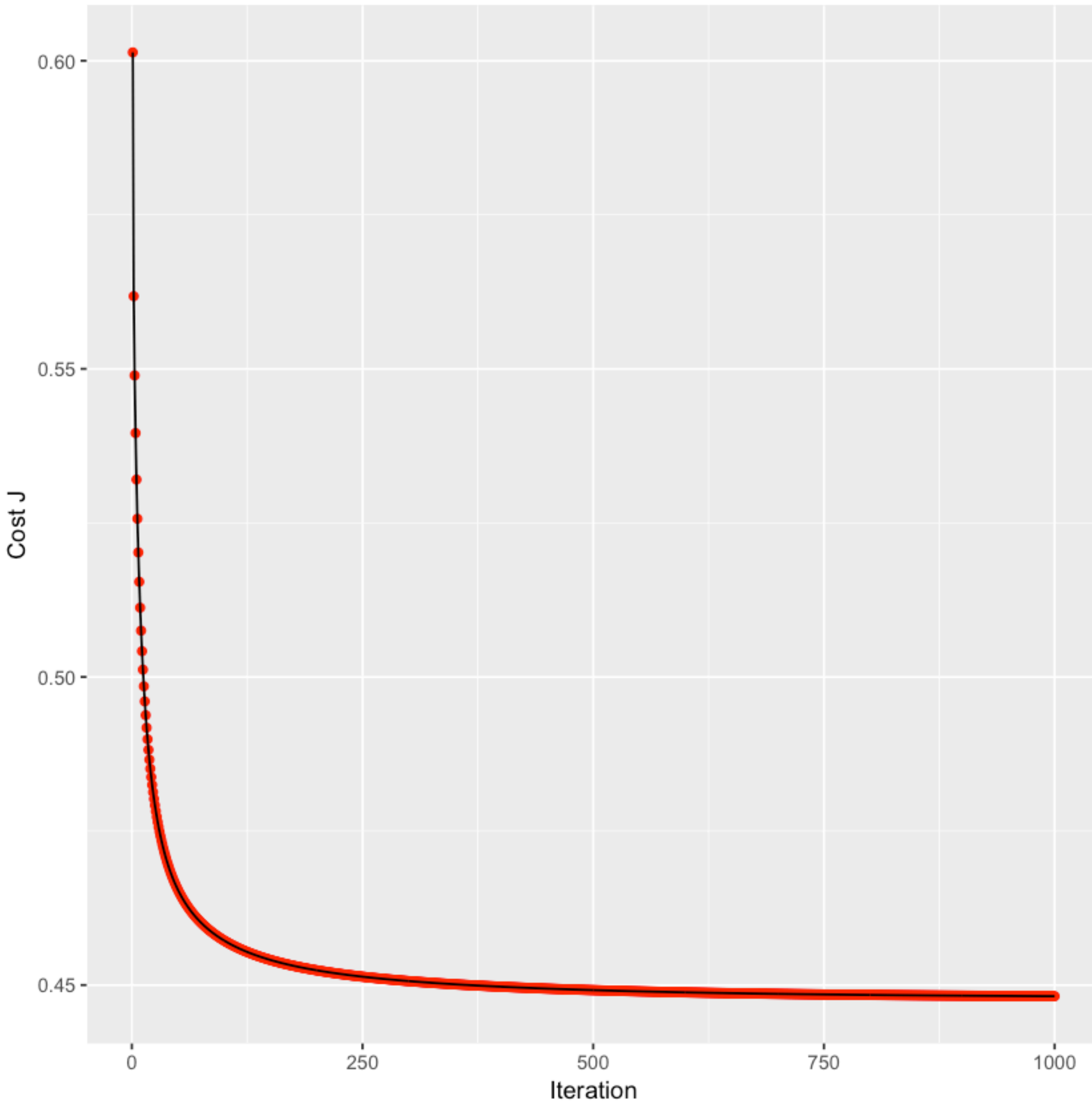
```
In [9]: alpha <- 1 #learning rate 1
        iter <- 1000 #no of interations for descent 5000
        #m <- length(y)
        theta <- rep(0, ncol(x))

        #cost at intial theta
        computeCost(theta, x, y)
        j <- rep(0,iter) #to maintain log of cost
        for (c in 1:iter) {
          theta <- theta - alpha * gradAlgo(theta,x,y)
          j[c] <- computeCost(theta,x,y)
        }
```

0.693147180559945

1.3.2 2. Plot Cost Value

Plot cost as a function of no of iterations to converge

```
In [10]: ggplot() +
           aes(x = 1:iter, y = j) +
           geom_point(colour = "red") +
           geom_path() + xlab("Iteration") +
           ylab("Cost J")
```



1.3.3 3. Build Classfication matrix

In [11]:
```
m <- nrow(y)
actual_y <- y
actual_y[actual_y == 0] <- "Bad"
actual_y[actual_y == 1] <- "Good"
predicted_y = rep("Bad", m);
predicted_y[hypothesisLr(theta,x) >= 0.5] = "Good";
addmargins(table(actual_y, predicted_y))
```

	Bad	Good	Sum
Bad	158	142	300
Good	73	627	700
Sum	231	769	1000

1.3.4 4. Score new cases

Scoring new profile pass all X's

```
In [12]: #new_score <- (sigmoid(c(1,<all X's>) %*% theta))
```

1.4 Compare the Gradient Descent with GLM

```
In [13]: lg_glm <- glm(filter_data$Credit.classification ~ .,data = filter_data, family = "bino
         summary(lg_glm)
         lg_predict_prob <- predict(lg_glm, filter_data[,1:20],"response")
         m <- length(filter_data[,21])
         actual_y <- filter_data[,21]

         predicted_y = rep("Bad", m);
         predicted_y[lg_predict_prob >= 0.5] = "Good";
         addmargins(table(actual_y, predicted_y))
```

```
Call:
glm(formula = filter_data$Credit.classification ~ ., family = "binomial",
    data = filter_data)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.6116  -0.7095   0.3752   0.6994   2.3410

Coefficients:
                                    Estimate Std. Error z value Pr(>|z|)
(Intercept)                        2.353e+00  1.330e+00   1.769 0.076938 .
CHK_ACCTless-200DM                 3.749e-01  2.179e-01   1.720 0.085400 .
CHK_ACCTno-account                 1.712e+00  2.322e-01   7.373 1.66e-13 ***
CHK_ACCTover-200DM                 9.657e-01  3.692e-01   2.616 0.008905 **
Duration                          -2.786e-02  9.296e-03  -2.997 0.002724 **
History bank-paid-duly            -1.434e-01  5.489e-01  -0.261 0.793921
History critical                   1.436e+00  4.399e-01   3.264 0.001099 **
History delay                      8.532e-01  4.717e-01   1.809 0.070470 .
History duly-till-now              5.861e-01  4.305e-01   1.362 0.173348
Purpose.of.credit domestic-app    -2.173e-01  8.041e-01  -0.270 0.786976
Purpose.of.credit education       -7.764e-01  4.660e-01  -1.666 0.095718 .
Purpose.of.credit furniture        5.152e-02  3.543e-01   0.145 0.884391
Purpose.of.credit new-car         -7.401e-01  3.339e-01  -2.216 0.026668 *
Purpose.of.credit others           7.487e-01  7.998e-01   0.936 0.349202
Purpose.of.credit radio-tv         1.515e-01  3.370e-01   0.450 0.653002
Purpose.of.credit repairs         -5.237e-01  5.933e-01  -0.883 0.377428
Purpose.of.credit retraining       1.319e+00  1.233e+00   1.070 0.284625
Purpose.of.credit used-car         9.264e-01  4.409e-01   2.101 0.035645 *
Credit.Amount                     -1.283e-04  4.444e-05  -2.887 0.003894 **
Balance.in.Savings.A.C less100DM  -3.761e-01  4.011e-01  -0.938 0.348476
```

```
Balance.in.Savings.A.C less500DM  -1.833e-02  4.656e-01  -0.039 0.968595
Balance.in.Savings.A.C over1000DM  9.631e-01  6.425e-01   1.499 0.133868
Balance.in.Savings.A.C unknown     5.706e-01  4.492e-01   1.270 0.203940
Employment one-year               -1.159e-01  2.423e-01  -0.478 0.632415
Employment over-seven              9.379e-02  2.510e-01   0.374 0.708653
Employment seven-years             6.482e-01  2.684e-01   2.415 0.015734 *
Employment unemployed             -1.828e-01  4.105e-01  -0.445 0.656049
Install_rate                      -3.301e-01  8.828e-02  -3.739 0.000185 ***
Marital.status male-divorced      -2.755e-01  3.865e-01  -0.713 0.476040
Marital.status married-male        9.162e-02  3.118e-01   0.294 0.768908
Marital.status single-male         5.406e-01  2.102e-01   2.572 0.010113 *
Co.applicant guarantor             1.415e+00  5.685e-01   2.488 0.012834 *
Co.applicant none                  4.360e-01  4.101e-01   1.063 0.287700
Present.Resident                  -4.776e-03  8.641e-02  -0.055 0.955920
Real.Estate car                    8.690e-02  2.313e-01   0.376 0.707115
Real.Estate none                  -4.490e-01  4.130e-01  -1.087 0.277005
Real.Estate real-estate            2.814e-01  2.534e-01   1.111 0.266630
Age                                1.454e-02  9.222e-03   1.576 0.114982
Other.installment none             6.463e-01  2.391e-01   2.703 0.006871 **
Other.installment stores           1.232e-01  4.119e-01   0.299 0.764878
Residence own                     -2.402e-01  4.503e-01  -0.534 0.593687
Residence rent                    -6.839e-01  4.770e-01  -1.434 0.151657
Num_Credits                       -2.721e-01  1.895e-01  -1.436 0.151109
Job skilled                       -7.524e-02  2.845e-01  -0.264 0.791419
Job unemployed-non-resident        4.795e-01  6.623e-01   0.724 0.469086
Job unskilled-resident            -5.666e-02  3.501e-01  -0.162 0.871450
No..dependents                    -2.647e-01  2.492e-01  -1.062 0.288249
Phone yes                         -3.000e-01  2.013e-01  -1.491 0.136060
Foreign yes                       -1.392e+00  6.258e-01  -2.225 0.026095 *
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1221.73  on 999  degrees of freedom
Residual deviance:  895.82  on 951  degrees of freedom
AIC: 993.82

Number of Fisher Scoring iterations: 5
```

	Bad	Good	Sum
bad.	160	140	300
good.	74	626	700
Sum	234	766	1000

End of Document