Logistic Lasso optimization

Hun, Jimmy

3/19/2022

```
cancer_df <- read.csv("~/Downloads/breast-cancer.csv") %>% janitor::clean_names()
```

```
data <-
  cancer_df %>% dplyr::select(-id, -x) %>%
  mutate(diagnosis = ifelse(diagnosis == "M", 1, 0)) %>% distinct()

set.seed(1)
split <- initial_split(data, prop = 0.8)

training_df <- split %>% training()

testing_df <- split %>% testing()
```

Standardization

```
for (i in 2:length(training_df)) {
  training_df[,i] = (training_df[,i] - mean(training_df[,i]))/sd(training_df[,i])
}
```

Checking standardization

```
summary <- skimr::skim_without_charts(training_df) %>% data.frame()
summary[1:6] %>% kable()
```

skim_type	skim_variable	n_missing	complete_rate	numeric.mean	numeric.sd
numeric	diagnosis	0	1	0.3714286	0.4837186
numeric	radius_mean	0	1	0.0000000	1.0000000
numeric	texture_mean	0	1	0.0000000	1.0000000
numeric	perimeter_mean	0	1	0.0000000	1.0000000
numeric	area_mean	0	1	0.0000000	1.0000000
numeric	smoothness_mean	0	1	0.0000000	1.0000000
numeric	compactness_mean	0	1	0.0000000	1.0000000
numeric	concavity_mean	0	1	0.0000000	1.0000000
numeric	concave_points_mean	0	1	0.0000000	1.0000000
numeric	symmetry_mean	0	1	0.0000000	1.0000000
numeric	fractal_dimension_mean	0	1	0.0000000	1.0000000
numeric	radius_se	0	1	0.0000000	1.0000000
numeric	texture_se	0	1	0.0000000	1.0000000
numeric	perimeter_se	0	1	0.0000000	1.0000000
numeric	area_se	0	1	0.0000000	1.0000000
numeric	smoothness_se	0	1	0.0000000	1.0000000
numeric	compactness_se	0	1	0.0000000	1.0000000
numeric	concavity_se	0	1	0.0000000	1.0000000
numeric	concave_points_se	0	1	0.0000000	1.0000000
numeric	symmetry_se	0	1	0.0000000	1.0000000
numeric	fractal_dimension_se	0	1	0.0000000	1.0000000
numeric	radius_worst	0	1	0.0000000	1.0000000
numeric	texture_worst	0	1	0.0000000	1.0000000
numeric	perimeter_worst	0	1	0.0000000	1.0000000
numeric	area_worst	0	1	0.0000000	1.0000000
numeric	smoothness_worst	0	1	0.0000000	1.0000000
numeric	compactness_worst	0	1	0.0000000	1.0000000
numeric	concavity_worst	0	1	0.0000000	1.0000000
numeric	concave_points_worst	0	1	0.0000000	1.0000000
numeric	symmetry_worst	0	1	0.0000000	1.0000000
numeric	fractal_dimension_worst	0	1	0.0000000	1.0000000

```
X <- training_df %>% dplyr::select(-diagnosis) %>% as.matrix()
y <- training_df$diagnosis %>% as.matrix()

dim(X)
```

```
## [1] 455  30
```

```
dim(y)
```

```
## [1] 455   1
```

Helper Functions

```
# logistic function
logistic <- function(x) 1 / (1 + exp(-x))

# soft threshold
```

```r
S <- function(beta, gamma) {
  if (beta > 0 && gamma < abs(beta)) {
    beta - gamma
  } else if (beta < 0 && gamma < abs(beta)) {
    beta + gamma
  } else {
    0
  }
}


# probability adjustment function
p_adj <- function(p, epsilon) {
  if (p < epsilon) {
    0
  } else if (p > 1 - epsilon) {
    1
  } else {
    p
  }
}

# weight adjustment function
w_adj <- function(p, epsilon) {
  if ((p < epsilon) | (p > 1 - epsilon)) {
    epsilon
  } else {
    p * (1 - p)
  }
}
```

Computing lambda max in two ways and defining lambda sequence

```r
alpha = 1
n = length(y)

lambda_max <- max(abs(t(X) %*% y))/n

same_lambda_max <- max(abs(t(y - mean(y)*(1 - mean(y))) %*% X ))/(alpha*n)

lambda_max
```

```
## [1] 0.386795
```

```r
same_lambda_max
```

```
## [1] 0.386795
```

```r
lambda_max <- lambda_max %>% round(digits = 2)
```

```
epsilon = 1e-5
lambda_seq <- seq(lambda_max, 1e-5*lambda_max, length = 100)
```

Toy example

```
set.seed(1)
n    <- 1000
X    <- scale(matrix(rnorm(3 * n), c(n, 3)))
X    <- as.matrix(cbind(rep(1, n), X))
y    <- 1 * (runif(n) > 0.5)

lambda_max <- round(max(abs(t(X) %*% y))/n,2)
epsilon <- 1e-5

lambda_seq <- seq(lambda_max, epsilon*lambda_max, length = 100)

# initialize parameters
beta <- list()
beta_old <- list()
b_k_temp <- list()

# creating initial beta list
for (i in 1:length(lambda_seq)) {
  beta[[i]] <- rep(0, ncol(X))
  beta_old[[i]] <- rep(NA, ncol(X))
}

# outer loop for lambda decrement
for (i in 1:length(lambda_seq)) {
  lambda <- lambda_seq[i]
# inner loop for updating quadratic approximation ingredient
  for (inner in 1:100) {
    p <- map_dbl(logistic(X %*% beta[[i]]), p_adj, epsilon)
    w <- map_dbl(p, w_adj, epsilon)
    z <- X %*% beta[[i]] + (y - p) / w

    terminate <- 0
    iter <- 1
    # while loop for beta convergence
    while (terminate < 1) {
      beta_old[[i]] <- beta[[i]]
      # running coordinate descending algorithm
      for (k in 1:ncol(X)) {
        x_k    <- X[ , k]
        x_notk <- X[ , -k]
        b_notk <- beta[[i]][-k]

        # un-penalized coefficient update
        b_k_temp <- sum(w * (z - x_notk %*% b_notk) * x_k) / sum(w * x_k^2)
        # shrinkage update
        b_k          <- S(b_k_temp, lambda * (k > 1) / mean(w * x_k^2))
```

```
        #b_k      <- S(sum(w*x_k*(z - x_notk %*% b_notk)), lambda * (k > 1)) / sum(w * x_k^2)

        # update beta list along with other parameters
        beta[[i]][k] <- b_k
        iter <- iter + 1
        }
      if (iter == 200 | max(abs(beta[[i]] - beta_old[[i]])) < 1e-10) {
        terminate <- 1
      }
    }
  }
}
# Estimates from Coordinate Descent
beta <- data.frame(beta)

for (i in 1:length(lambda_seq)) {
  colnames(beta)[i] <- paste0(lambda_seq[i] %>% round(digits = 6))
}

beta <- t(beta) %>% data.frame()

beta %>% mutate(lambda = rownames(beta)) %>% relocate(lambda) %>% `rownames<-`(NULL)
```

```
##       lambda        X1         X2          X3           X4
## 1       0.53 0.1040939 0.00000000 0.000000000  0.000000000
## 2   0.524647 0.1040939 0.00000000 0.000000000  0.000000000
## 3   0.519293 0.1040939 0.00000000 0.000000000  0.000000000
## 4    0.51394 0.1040939 0.00000000 0.000000000  0.000000000
## 5   0.508586 0.1040939 0.00000000 0.000000000  0.000000000
## 6   0.503233 0.1040939 0.00000000 0.000000000  0.000000000
## 7   0.497879 0.1040939 0.00000000 0.000000000  0.000000000
## 8   0.492526 0.1040939 0.00000000 0.000000000  0.000000000
## 9   0.487172 0.1040939 0.00000000 0.000000000  0.000000000
## 10  0.481819 0.1040939 0.00000000 0.000000000  0.000000000
## 11  0.476465 0.1040939 0.00000000 0.000000000  0.000000000
## 12  0.471112 0.1040939 0.00000000 0.000000000  0.000000000
## 13  0.465758 0.1040939 0.00000000 0.000000000  0.000000000
## 14  0.460405 0.1040939 0.00000000 0.000000000  0.000000000
## 15  0.455051 0.1040939 0.00000000 0.000000000  0.000000000
## 16  0.449698 0.1040939 0.00000000 0.000000000  0.000000000
## 17  0.444344 0.1040939 0.00000000 0.000000000  0.000000000
## 18  0.438991 0.1040939 0.00000000 0.000000000  0.000000000
## 19  0.433637 0.1040939 0.00000000 0.000000000  0.000000000
## 20  0.428284 0.1040939 0.00000000 0.000000000  0.000000000
## 21   0.42293 0.1040939 0.00000000 0.000000000  0.000000000
## 22  0.417577 0.1040939 0.00000000 0.000000000  0.000000000
## 23  0.412223 0.1040939 0.00000000 0.000000000  0.000000000
## 24   0.40687 0.1040939 0.00000000 0.000000000  0.000000000
## 25  0.401516 0.1040939 0.00000000 0.000000000  0.000000000
## 26  0.396163 0.1040939 0.00000000 0.000000000  0.000000000
## 27  0.390809 0.1040939 0.00000000 0.000000000  0.000000000
## 28  0.385456 0.1040939 0.00000000 0.000000000  0.000000000
## 29  0.380103 0.1040939 0.00000000 0.000000000  0.000000000
```

```
## 30  0.374749 0.1040939 0.00000000 0.000000000  0.000000000
## 31  0.369396 0.1040939 0.00000000 0.000000000  0.000000000
## 32  0.364042 0.1040939 0.00000000 0.000000000  0.000000000
## 33  0.358689 0.1040939 0.00000000 0.000000000  0.000000000
## 34  0.353335 0.1040939 0.00000000 0.000000000  0.000000000
## 35  0.347982 0.1040939 0.00000000 0.000000000  0.000000000
## 36  0.342628 0.1040939 0.00000000 0.000000000  0.000000000
## 37  0.337275 0.1040939 0.00000000 0.000000000  0.000000000
## 38  0.331921 0.1040939 0.00000000 0.000000000  0.000000000
## 39  0.326568 0.1040939 0.00000000 0.000000000  0.000000000
## 40  0.321214 0.1040939 0.00000000 0.000000000  0.000000000
## 41  0.315861 0.1040939 0.00000000 0.000000000  0.000000000
## 42  0.310507 0.1040939 0.00000000 0.000000000  0.000000000
## 43  0.305154 0.1040939 0.00000000 0.000000000  0.000000000
## 44    0.2998 0.1040939 0.00000000 0.000000000  0.000000000
## 45  0.294447 0.1040939 0.00000000 0.000000000  0.000000000
## 46  0.289093 0.1040939 0.00000000 0.000000000  0.000000000
## 47   0.28374 0.1040939 0.00000000 0.000000000  0.000000000
## 48  0.278386 0.1040939 0.00000000 0.000000000  0.000000000
## 49  0.273033 0.1040939 0.00000000 0.000000000  0.000000000
## 50  0.267679 0.1040939 0.00000000 0.000000000  0.000000000
## 51  0.262326 0.1040939 0.00000000 0.000000000  0.000000000
## 52  0.256972 0.1040939 0.00000000 0.000000000  0.000000000
## 53  0.251619 0.1040939 0.00000000 0.000000000  0.000000000
## 54  0.246265 0.1040939 0.00000000 0.000000000  0.000000000
## 55  0.240912 0.1040939 0.00000000 0.000000000  0.000000000
## 56  0.235559 0.1040939 0.00000000 0.000000000  0.000000000
## 57  0.230205 0.1040939 0.00000000 0.000000000  0.000000000
## 58  0.224852 0.1040939 0.00000000 0.000000000  0.000000000
## 59  0.219498 0.1040939 0.00000000 0.000000000  0.000000000
## 60  0.214145 0.1040939 0.00000000 0.000000000  0.000000000
## 61  0.208791 0.1040939 0.00000000 0.000000000  0.000000000
## 62  0.203438 0.1040939 0.00000000 0.000000000  0.000000000
## 63  0.198084 0.1040939 0.00000000 0.000000000  0.000000000
## 64  0.192731 0.1040939 0.00000000 0.000000000  0.000000000
## 65  0.187377 0.1040939 0.00000000 0.000000000  0.000000000
## 66  0.182024 0.1040939 0.00000000 0.000000000  0.000000000
## 67   0.17667 0.1040939 0.00000000 0.000000000  0.000000000
## 68  0.171317 0.1040939 0.00000000 0.000000000  0.000000000
## 69  0.165963 0.1040939 0.00000000 0.000000000  0.000000000
## 70   0.16061 0.1040939 0.00000000 0.000000000  0.000000000
## 71  0.155256 0.1040939 0.00000000 0.000000000  0.000000000
## 72  0.149903 0.1040939 0.00000000 0.000000000  0.000000000
## 73  0.144549 0.1040939 0.00000000 0.000000000  0.000000000
## 74  0.139196 0.1040939 0.00000000 0.000000000  0.000000000
## 75  0.133842 0.1040939 0.00000000 0.000000000  0.000000000
## 76  0.128489 0.1040939 0.00000000 0.000000000  0.000000000
## 77  0.123135 0.1040939 0.00000000 0.000000000  0.000000000
## 78  0.117782 0.1040939 0.00000000 0.000000000  0.000000000
## 79  0.112428 0.1040939 0.00000000 0.000000000  0.000000000
## 80  0.107075 0.1040939 0.00000000 0.000000000  0.000000000
## 81  0.101721 0.1040939 0.00000000 0.000000000  0.000000000
## 82  0.096368 0.1040939 0.00000000 0.000000000  0.000000000
## 83  0.091014 0.1040939 0.00000000 0.000000000  0.000000000
```

```
## 84  0.085661 0.1040939 0.00000000 0.000000000  0.000000000
## 85  0.080308 0.1040939 0.00000000 0.000000000  0.000000000
## 86  0.074954 0.1040939 0.00000000 0.000000000  0.000000000
## 87  0.069601 0.1040939 0.00000000 0.000000000  0.000000000
## 88  0.064247 0.1040939 0.00000000 0.000000000  0.000000000
## 89  0.058894 0.1040939 0.00000000 0.000000000  0.000000000
## 90   0.05354 0.1040939 0.00000000 0.000000000  0.000000000
## 91  0.048187 0.1040939 0.00000000 0.000000000  0.000000000
## 92  0.042833 0.1040939 0.00000000 0.000000000  0.000000000
## 93   0.03748 0.1040939 0.00000000 0.000000000  0.000000000
## 94  0.032126 0.1040939 0.00000000 0.000000000  0.000000000
## 95  0.026773 0.1040939 0.00000000 0.000000000  0.000000000
## 96  0.021419 0.1040939 0.00000000 0.000000000  0.000000000
## 97  0.016066 0.1040939 0.00000000 0.000000000  0.000000000
## 98  0.010712 0.1040956 0.00000000 0.000000000 -0.008051621
## 99  0.005359 0.1041167 0.00000000 0.000000000 -0.029552984
## 100    5e-06 0.1041748 0.02045426 0.006292915 -0.052235167
```

```
# True estimates from GLM
#as.vector(glm(y ~ X[ , -1], family = binomial)$coefficients)

# True estimates from GLMNET
fit <- glmnet(X, y, family = "binomial", standardize = FALSE, lambda = lambda_seq, thresh = 1e-10)

fit_result <- fit$beta %>% as.matrix() %>% t() %>% data.frame() %>% `rownames<-`(NULL)

fit_result %>% mutate(lambda = fit$lambda %>% round(digits = 6)) %>% relocate(lambda)
```

```
##       lambda V1         V2          V3          V4
## 1   0.530000  0 0.00000000 0.000000000  0.00000000
## 2   0.524647  0 0.00000000 0.000000000  0.00000000
## 3   0.519293  0 0.00000000 0.000000000  0.00000000
## 4   0.513940  0 0.00000000 0.000000000  0.00000000
## 5   0.508586  0 0.00000000 0.000000000  0.00000000
## 6   0.503233  0 0.00000000 0.000000000  0.00000000
## 7   0.497879  0 0.00000000 0.000000000  0.00000000
## 8   0.492526  0 0.00000000 0.000000000  0.00000000
## 9   0.487172  0 0.00000000 0.000000000  0.00000000
## 10  0.481819  0 0.00000000 0.000000000  0.00000000
## 11  0.476465  0 0.00000000 0.000000000  0.00000000
## 12  0.471112  0 0.00000000 0.000000000  0.00000000
## 13  0.465758  0 0.00000000 0.000000000  0.00000000
## 14  0.460405  0 0.00000000 0.000000000  0.00000000
## 15  0.455051  0 0.00000000 0.000000000  0.00000000
## 16  0.449698  0 0.00000000 0.000000000  0.00000000
## 17  0.444344  0 0.00000000 0.000000000  0.00000000
## 18  0.438991  0 0.00000000 0.000000000  0.00000000
## 19  0.433637  0 0.00000000 0.000000000  0.00000000
## 20  0.428284  0 0.00000000 0.000000000  0.00000000
## 21  0.422930  0 0.00000000 0.000000000  0.00000000
## 22  0.417577  0 0.00000000 0.000000000  0.00000000
## 23  0.412223  0 0.00000000 0.000000000  0.00000000
## 24  0.406870  0 0.00000000 0.000000000  0.00000000
## 25  0.401516  0 0.00000000 0.000000000  0.00000000
```

```
## 26  0.396163  0 0.00000000 0.000000000  0.00000000
## 27  0.390809  0 0.00000000 0.000000000  0.00000000
## 28  0.385456  0 0.00000000 0.000000000  0.00000000
## 29  0.380103  0 0.00000000 0.000000000  0.00000000
## 30  0.374749  0 0.00000000 0.000000000  0.00000000
## 31  0.369396  0 0.00000000 0.000000000  0.00000000
## 32  0.364042  0 0.00000000 0.000000000  0.00000000
## 33  0.358689  0 0.00000000 0.000000000  0.00000000
## 34  0.353335  0 0.00000000 0.000000000  0.00000000
## 35  0.347982  0 0.00000000 0.000000000  0.00000000
## 36  0.342628  0 0.00000000 0.000000000  0.00000000
## 37  0.337275  0 0.00000000 0.000000000  0.00000000
## 38  0.331921  0 0.00000000 0.000000000  0.00000000
## 39  0.326568  0 0.00000000 0.000000000  0.00000000
## 40  0.321214  0 0.00000000 0.000000000  0.00000000
## 41  0.315861  0 0.00000000 0.000000000  0.00000000
## 42  0.310507  0 0.00000000 0.000000000  0.00000000
## 43  0.305154  0 0.00000000 0.000000000  0.00000000
## 44  0.299800  0 0.00000000 0.000000000  0.00000000
## 45  0.294447  0 0.00000000 0.000000000  0.00000000
## 46  0.289093  0 0.00000000 0.000000000  0.00000000
## 47  0.283740  0 0.00000000 0.000000000  0.00000000
## 48  0.278386  0 0.00000000 0.000000000  0.00000000
## 49  0.273033  0 0.00000000 0.000000000  0.00000000
## 50  0.267679  0 0.00000000 0.000000000  0.00000000
## 51  0.262326  0 0.00000000 0.000000000  0.00000000
## 52  0.256972  0 0.00000000 0.000000000  0.00000000
## 53  0.251619  0 0.00000000 0.000000000  0.00000000
## 54  0.246265  0 0.00000000 0.000000000  0.00000000
## 55  0.240912  0 0.00000000 0.000000000  0.00000000
## 56  0.235559  0 0.00000000 0.000000000  0.00000000
## 57  0.230205  0 0.00000000 0.000000000  0.00000000
## 58  0.224852  0 0.00000000 0.000000000  0.00000000
## 59  0.219498  0 0.00000000 0.000000000  0.00000000
## 60  0.214145  0 0.00000000 0.000000000  0.00000000
## 61  0.208791  0 0.00000000 0.000000000  0.00000000
## 62  0.203438  0 0.00000000 0.000000000  0.00000000
## 63  0.198084  0 0.00000000 0.000000000  0.00000000
## 64  0.192731  0 0.00000000 0.000000000  0.00000000
## 65  0.187377  0 0.00000000 0.000000000  0.00000000
## 66  0.182024  0 0.00000000 0.000000000  0.00000000
## 67  0.176670  0 0.00000000 0.000000000  0.00000000
## 68  0.171317  0 0.00000000 0.000000000  0.00000000
## 69  0.165963  0 0.00000000 0.000000000  0.00000000
## 70  0.160610  0 0.00000000 0.000000000  0.00000000
## 71  0.155256  0 0.00000000 0.000000000  0.00000000
## 72  0.149903  0 0.00000000 0.000000000  0.00000000
## 73  0.144549  0 0.00000000 0.000000000  0.00000000
## 74  0.139196  0 0.00000000 0.000000000  0.00000000
## 75  0.133842  0 0.00000000 0.000000000  0.00000000
## 76  0.128489  0 0.00000000 0.000000000  0.00000000
## 77  0.123135  0 0.00000000 0.000000000  0.00000000
## 78  0.117782  0 0.00000000 0.000000000  0.00000000
## 79  0.112428  0 0.00000000 0.000000000  0.00000000
```

```
## 80  0.107075  0 0.00000000 0.000000000  0.00000000
## 81  0.101721  0 0.00000000 0.000000000  0.00000000
## 82  0.096368  0 0.00000000 0.000000000  0.00000000
## 83  0.091014  0 0.00000000 0.000000000  0.00000000
## 84  0.085661  0 0.00000000 0.000000000  0.00000000
## 85  0.080308  0 0.00000000 0.000000000  0.00000000
## 86  0.074954  0 0.00000000 0.000000000  0.00000000
## 87  0.069601  0 0.00000000 0.000000000  0.00000000
## 88  0.064247  0 0.00000000 0.000000000  0.00000000
## 89  0.058894  0 0.00000000 0.000000000  0.00000000
## 90  0.053540  0 0.00000000 0.000000000  0.00000000
## 91  0.048187  0 0.00000000 0.000000000  0.00000000
## 92  0.042833  0 0.00000000 0.000000000  0.00000000
## 93  0.037480  0 0.00000000 0.000000000  0.00000000
## 94  0.032126  0 0.00000000 0.000000000  0.00000000
## 95  0.026773  0 0.00000000 0.000000000  0.00000000
## 96  0.021419  0 0.00000000 0.000000000  0.00000000
## 97  0.016066  0 0.00000000 0.000000000  0.00000000
## 98  0.010712  0 0.00000000 0.000000000 -0.00805162
## 99  0.005359  0 0.00000000 0.000000000 -0.02955296
## 100 0.000005  0 0.02045426 0.006292915 -0.05223517
```