Logistic-Lasso Coordinate Descent Algorithm

Jimmy Kelliher (UNI: jmk2303)

2022-03-20

Theory

Lemma 1. Consider the optimization problem

$$\min_{x \in \mathbb{R}} \left\{ \frac{1}{2}(x-b)^2 + c|x| \right\}$$

for $b \in \mathbb{R}$ and $c \in \mathbb{R}_{++}$. It follows that the minimizer is given by

$$\hat{x} = S(b, c),$$

where S is the soft-thresholding operator.

Lemma 2. Consider the optimization problem

$$\min_{\beta_k \in \mathbb{R}} \left\{ \frac{1}{2n} \sum_{i=1}^{n} w_i \left(z_i - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 \right\}$$

for some $k \in \{1, \ldots, p\}$. It follows that the minimizer is given by

$$\hat{\beta}_k = \left(\sum_{i=1}^{n} w_i x_{ik}^2 \right)^{-1} \sum_{i=1}^{n} w_i x_{ik} \left(z_i - \sum_{j \neq k} \beta_j x_{ij} \right).$$

Lemma 3. With $\hat{\beta}_k$ defined as above,

$$\min_{\beta_k \in \mathbb{R}} \left\{ \frac{1}{2n} \sum_{i=1}^{n} w_i \left(z_i - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^{p} |\beta_j| \right\} = \min_{\beta_k \in \mathbb{R}} \left\{ \frac{1}{2}(\beta_k - \hat{\beta}_k)^2 + \left(\frac{1}{n} \sum_{i=1}^{n} w_i x_{ik}^2 \right)^{-1} \lambda |\beta_k| \right\}.$$

Proposition. By Lemma 1 and Lemma 3,

$$\arg\min_{\beta_k \in \mathbb{R}} \left\{ \frac{1}{2n} \sum_{i=1}^{n} w_i \left(z_i - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^{p} |\beta_j| \right\} = S \left(\hat{\beta}_k, \left(\frac{1}{n} \sum_{i=1}^{n} w_i x_{ik}^2 \right)^{-1} \lambda \right)$$

Praxis

```
data <-
  read_csv("data/breast-cancer.csv") %>%
  mutate(diagnosis = 1 * (diagnosis == "M"))
```

```
## Rows: 569 Columns: 32
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (1): diagnosis
## dbl (31): id, radius_mean, texture_mean, perimeter_mean, area_mean, smoothne...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

Helper Functions

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

# shrinkage function
S <- function(beta, gamma) {
  if(abs(beta) <= gamma) {
    0
  } else if(beta > 0) {
    beta - gamma
  } else {
    beta + gamma
  }
}

# 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)
  }
}
```

Toy Example

```r
set.seed(1)
lambda <- 0 #0.0125
epsilon <- 10^(-5)

q    <- 30 - 1
n    <- 1000
X    <- scale(matrix(rnorm(q * n), c(n, q)))
X    <- as.matrix(cbind(rep(1, n), X))
y    <- 1 * (runif(n) > 0.5)

# initialize parameters
beta <- rep(0.25, ncol(X))

for(outer in 1:10) {

p <- map_dbl(logistic(X %*% beta), p_adj, epsilon)
w <- map_dbl(p, w_adj, epsilon)
z <- X %*% beta + (y - p) / w

terminate <- 0
iter <- 1
while(terminate < 1) {

  beta_old <- beta

  for(k in 1:ncol(X)) {
    x_k    <- X[ , k]
    x_notk <- X[ , -k]
    b_notk <- beta[-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))
    # update beta vector along with other parameters
    beta[k] <- b_k
    p <- map_dbl(logistic(X %*% beta), p_adj, epsilon)
    w <- map_dbl(p, w_adj, epsilon)
    z <- X %*% beta + (y - p) / w
  }

  iter <- iter + 1

  if(iter == 100 | max(abs(beta - beta_old)) < 10^(-10)) {
    print(iter)
    terminate <- 1
  }

}

#print(beta)
```

```
}
```

```
## [1] 12
## [1] 2
## [1] 2
## [1] 2
## [1] 2
## [1] 2
## [1] 2
## [1] 2
## [1] 2
## [1] 2
```

```
# Estimates from Coordinate Descent
#print(beta)

# 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, thresh = 10^-10)
#as.vector(fit$beta[ , ncol(fit$beta)])

# results
results <- tibble(
    Variable   = 1:length(beta)
  , Jimmy      = beta
  , GLM        = as.vector(glm(y ~ X[ , -1], family = binomial)$coefficients)
  , GLMNET     = as.vector(fit$beta[ , ncol(fit$beta)])
  , Difference = abs(Jimmy - GLM)
  , Change     = (Jimmy - GLM) / GLM
) %>%
  filter(Jimmy != 0 | GLMNET != 0)

results %>% knitr::kable()
```

Variable	Jimmy	GLM	GLMNET	Difference	Change
1	-0.0168645	-0.0168645	0.0000000	0	0
2	0.0076605	0.0076605	0.0076606	0	0
3	-0.0011113	-0.0011113	-0.0011112	0	0
4	0.0041578	0.0041578	0.0041575	0	0
5	0.0500018	0.0500018	0.0500013	0	0
6	0.0470723	0.0470723	0.0470715	0	0
7	0.0345528	0.0345528	0.0345528	0	0
8	-0.0233566	-0.0233566	-0.0233561	0	0
9	-0.0811655	-0.0811655	-0.0811654	0	0
10	0.0998044	0.0998044	0.0998046	0	0
11	-0.0208418	-0.0208418	-0.0208417	0	0
12	0.1030075	0.1030075	0.1030065	0	0
13	0.0421534	0.0421534	0.0421532	0	0
14	0.0196497	0.0196497	0.0196490	0	0
15	-0.0385209	-0.0385209	-0.0385213	0	0
16	-0.0835662	-0.0835662	-0.0835657	0	0
17	0.0565496	0.0565496	0.0565501	0	0

Variable	Jimmy	GLM	GLMNET	Difference	Change
18	-0.1303866	-0.1303866	-0.1303859	0	0
19	0.0447251	0.0447251	0.0447248	0	0
20	0.0162045	0.0162045	0.0162041	0	0
21	0.0573634	0.0573634	0.0573630	0	0
22	-0.0104509	-0.0104509	-0.0104510	0	0
23	0.1678573	0.1678573	0.1678568	0	0
24	0.0502613	0.0502613	0.0502613	0	0
25	-0.0231361	-0.0231361	-0.0231364	0	0
26	-0.0022917	-0.0022917	-0.0022916	0	0
27	0.0413491	0.0413491	0.0413487	0	0
28	0.0106969	0.0106969	0.0106971	0	0
29	-0.0729549	-0.0729549	-0.0729548	0	0
30	0.0916826	0.0916826	0.0916824	0	0

```r
c(mean(results$Difference[-1]), mean(results$Change[-1]), 1/n)
```

```
## [1] 1.68826e-15 4.01873e-14 1.00000e-03
```

Test with Actual Data

```r
set.seed(1)
epsilon <- 10^(-5)

n    <- nrow(data)
#X    <- scale(data[ , -c(1, 2)])
X    <- data[ , -c(1, 2)]
X    <- as.matrix(cbind(rep(1, n), X))
y    <- data$diagnosis

beta <- rep(0, ncol(X))

for(lambda in c(10)) { #, 1, 0.1, 0.01, 0.001)) {
# (max(t(X) %*% y) / n)

for(outer in 1:10) {

# initialize parameters
p <- map_dbl(logistic(X %*% beta), p_adj, epsilon)
w <- map_dbl(p, w_adj, epsilon)
z <- X %*% beta + (y - p) / w

terminate <- 0
iter <- 1
while(terminate < 1) {

  beta_old <- beta
  # initially go through all parameters
  K <- 1:ncol(X)
  #if(iter > 1) {
  #  K <- which(beta > 0)
```

```r
  #}

  for(k in K) {
    x_k    <- X[ , k]
    x_notk <- X[ , -k]
    b_notk <- beta[-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))

    # update beta vector along with other parameters
    beta[k] <- b_k
    #p <- map_dbl(logistic(- X %*% beta), p_adj, epsilon)
    #w <- map_dbl(p, w_adj, epsilon)
    #z <- X %*% beta + (y - p) / w
  }

  iter <- iter + 1

  if(iter == 1000 | max(abs(beta - beta_old)) < 10^-10) {
    print(iter)
    terminate <- 1
  }

}

}

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

# results
results <- tibble(
    Variable   = 1:length(beta)
  , Name       = c("intercept", names(data[ , -c(1, 2)]))
  , Jimmy      = beta
  , GLMNET     = as.vector(fit$beta[ , ncol(fit$beta)])
  , Difference = abs(Jimmy - GLMNET)
) %>%
  filter(Jimmy != 0 | GLMNET != 0)

print(paste0("lambda = ", lambda))
print(results %>% knitr::kable())
}
```

```
## [1] 116
## [1] 111
## [1] 158
## [1] 223
## [1] 253
## [1] 219
## [1] 130
```

```
## [1] 2
## [1] 18
## [1] 2
## [1] "lambda = 10"
##
##
## | Variable|Name       |      Jimmy|    GLMNET| Difference|
## |--------:|:----------|----------:|---------:|----------:|
## |        1|intercept  | -6.4985163| 0.0000000|   6.498516|
## |       25|area_worst |  0.0072571| 0.0072571|   0.000000|
```