AdaBoost and K-Fold Cross-Validation on Hand-Written Digits

KK Feng

1 Problem

1.1 Description

Classify grayscale images for hand-written digits.

1.2 Data

- **uspsdata.txt**: contains a matrix with one data point (= vector of length 256) per row. The 256-vector in each row represents a 16 by 16 image of a handwritten number.
- **uspscl.txt**: contains the corresponding class labels. The data contains two classes - the digits 5 and 6 - so the class labels are stored as -1 and +1, respectively.

1.3 Idea

- Adaptive Boosting algorithm with decision stumps as weak learners.
- K-Fold Cross-Validation to tune the number of weak learners.

2 Solution

2.1 Implementation

To train decision stumps, we implement following algorithm

Algorithm 1 A simple training algorithm for decision stumps

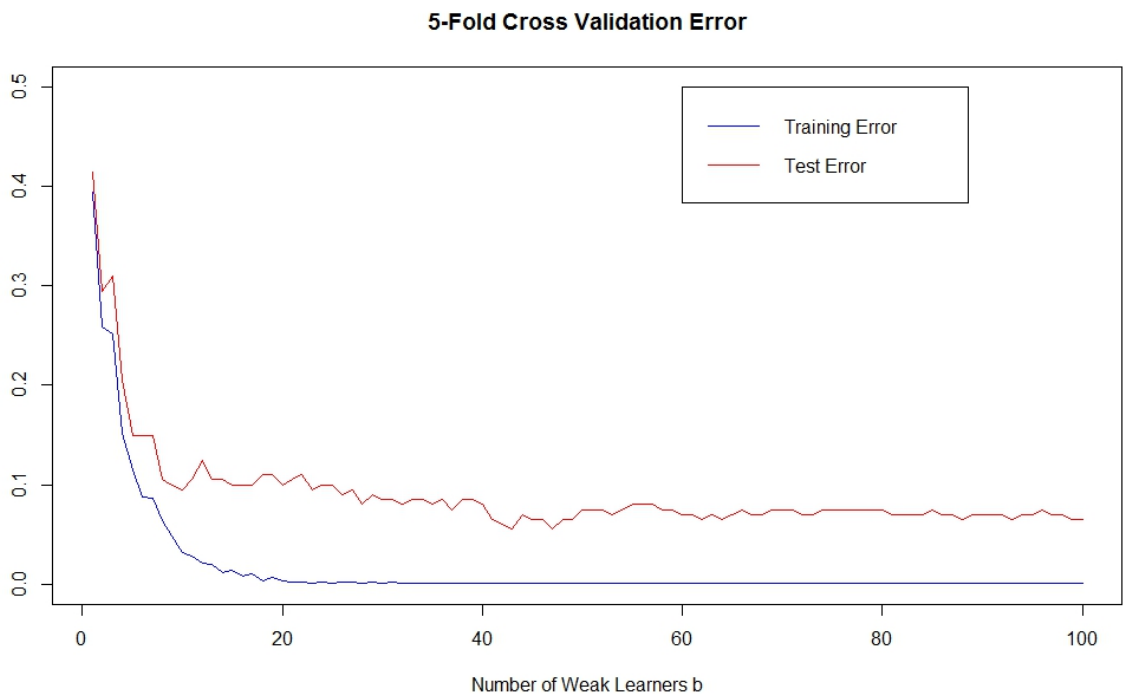
Require: Data $X = (x_1, \cdots, x_n)$ where $x_i \in \mathbb{R}^d$, weight w, label y
for $j = 1 : d$ **do**
    Sort samples x_i in ascending order along dimension j
    for $i = 1 : n$ **do**
        Compute cumulative sums $cum_i^j = \sum_{k=1}^{i} w_k y_k$
    end for
    Threshold θ_j is obtained at the extrema of cum_i^j
    Label m_j is obtained from the sign of cumulative sum at extrema
    Compute the error rate of classifier (θ_j, m_j) along dimension j
end for
Find optimal j^*, θ^* in which the classifier (θ_j, m_j) gives the minimum error rate

---

(Reference: http://ais.informatik.uni-freiburg.de/teaching/ws09/robotics2/pdfs/rob2-10-adaboost.pdf)

2.2 Plot

Please find the plots for training error and test error as a function of b (number of weak learners) in the following. Note that the cross validation error is the average of errors of 5 folds.



From the plot, we can see that for the USPS data and using 5-fold cross validation, the training error reaches bottom of the curve when we use approximately 20 weak learners, and the training error curve become flat when number of weak learners go larger than 20. On the other hand, we need around 40 weak learners to ensure that we have the optimal test error. If number of weak learners go larger than 40, the test error will just have small oscillations around the optimal test error we get at 40 weak learners.

2.3 Code

```
###### AdaBoost and K-Fold Cross-Validation on Hand-Written Digits ######

# Train: weak learner training routine
train <- function(X, w, y){
  n <- dim(X)[1]
  d <- dim(X)[2]
  theta <- rep(0,d)
  m <- rep(0,d)
  error <- rep(0,d)
  # find best stump classifier (theta_j, m_j) for each dimension j
  for (j in 1:d){
    x_order <- order(X[,j]) # get order of data along dimension j
    x <- X[x_order,j]
    cum <- rep(0,n) # compute cumulative sums cum_i^j = \sum_{k=1}^i w_ky_k
    weighted_label <- w[x_order] * y[x_order]
    cum[1] <- weighted_label[1]
    for (i in 2:n){
      cum[i] <- weighted_label[i] + cum[i-1]
    }
    index <- which.max(abs(cum))  # find theta_j, m_j
    theta[j] <- x[index]
    m[j] <- sign(cum[index])
```

```r
    yy <- rep(-m[j], n)
    yy[x > theta[j]] <- m[j]
    error[j] <- (yy != y) %*% w   # compute error rate of classifier (theta_j, m_j)
  }
  j_star <- which.min(error)  # find optimal dimension j
  pars <- list(j = j_star, theta = theta[j_star], m = m[j_star])
  return(pars)
}

# Classify: evaluates the weak learner on X using the parametrization pars
classify <- function(X, pars){
  label <- rep(-pars$m, dim(X)[1])
  label[X[,pars$j] > pars$theta] <- pars$m
  return(label)
}

# Agg_class: evaluates the boosting classifier ("aggregated classifier") on X.
agg_class <- function(X, alpha, allPars){
  n <- dim(X)[1]
  B <- length(alpha)
  label_sum <- rep(0,n)
  for (b in 1:B){
    label_sum <- label_sum + alpha[b] * classify(X, allPars[[b]]) # sum up weighted labels
  }
  c_hat <- sign(label_sum)
  return(c_hat)
}

# AdaBoost: implement the AdaBoost algorithm
AdaBoost <- function(X, y, B){
  n <- length(y)
  w <- rep(1/n, n) # Initialize weights
  alpha <- rep(0,B) # Initialize alphas
  allPars <- rep(list(list(0)), B)
  for (b in 1:B) {
    allPars[[b]] <- train(X, w, y)  # Train a weak learner c_b
    index <- y != classify(X, allPars[[b]]) # misclassification index
    error <- sum(w[index]) / sum(w)  # Compute error
    alpha[b] <- log((1-error)/error) # Compute voting weights
    w[index] <- w[index] * exp(alpha[b]) # Recompute weights
  }
  return(list(alpha = alpha, allPars = allPars)) # Return classifier
}

# Problem 1.3 Run algorithm on the USPS data and evaluate results using cross validation.
X <- read.table("uspsdata.txt")
y <- read.table("uspscl.txt")[,1]
n <- length(y)
B <- 100  # maximum number of weak learners
m <- 5  # 5-fold cross validation
train_error <- matrix(0, nrow = B, ncol = m)
test_error <- matrix(0, nrow = B, ncol = m)
for (i in 1:m){
  # generate train data and test data for fold i
  index <- round(n/m*(i-1)+1) : trunc(n/m*i)
  data_train <- X[-index,]
  y_train <- y[-index]
  data_test <- X[index,]
  y_test <- y[index]
  # get AdaBoost classifer
  AB <- AdaBoost(data_train, y_train, B)
  alpha <- AB$alpha
  allPars <- AB$allPars
  for (b in 1:B){
  # compute train and test error for fold i by AdaBoost with b weak learners
    train_error[b,i] <- sum(y_train != agg_class(data_train, alpha[1:b], allPars[1:b]))/
        length(y_train)
```

```r
    test_error[b,i] <- sum(y_test != agg_class(data_test, alpha[1:b], allPars[1:b]))/length(y_test)
  }
}
# compute cross validation error
cross_train_error <- rep(0, B)
cross_test_error <- rep(0, B)
for (b in 1:B)
{
  cross_train_error[b] <- mean(train_error[b,])
  cross_test_error[b] <- mean(test_error[b,])
}

# Plot the training error and the test error as a function of b.
plot(cross_train_error,type='l',ylim=c(0,0.5),col='blue',xlab='Number of Weak Learners b', ylab='', main='5-Fold Cross Validation Error')
lines(cross_test_error,col='red')
legend(60,0.5,c('Training Error','Test Error'),col=c('blue','red'),lty=1)
```