This document presents benchmark data analysis similar to Wang (2012) using R package \texttt{bst}. We compare the multi-class HingeBoost using three different algorithms for four benchmark data sets available from the UCI repository of machine learning data. We utilized regression trees as base learners in the HingeBoost. The number of terminal nodes is related to the depth of the tree, and the degree of interactions. To illustrate, we present the results for maximum tree depth 6.

\section{Image segmentation data}

```r
library("bst")

dat1 <- "segmentation.data"
dat1 <- read.delim(paste(tmp, dat1, sep = ""), sep = ",", header = FALSE, skip = 5)
dat2 <- "segmentation.test"
dat2 <- read.delim(paste(tmp, dat2, sep = ""), sep = ",", header = FALSE, skip = 5)
dat1[, 1] <- as.numeric(factor(dat1[, 1]))
dat2[, 1] <- as.numeric(factor(dat2[, 1]))
m <- 500
dat.m1 <- mbst(x = dat1[, -1], y = dat1[, 1], ctrl = bst_control(mstop = m),
  control.tree = list(maxdepth = 6), family = "hinge", learner = "tree")
err.te1 <- predict(dat.m1, newdata = dat2[, -1], newy = dat2[, 1], mstop = m,
  type = "error")
dat.m2 <- mbst(x = dat1[, -1], y = dat1[, 1], ctrl = bst_control(mstop = m),
  control.tree = list(maxdepth = 6), family = "hinge2", learner = "tree")
err.te2 <- predict(dat.m2, newdata = dat2[, -1], newy = dat2[, 1], mstop = m,
  type = "error")
dat.m3 <- mhingebst(x = dat1[, -1], y = dat1[, 1], ctrl = bst_control(mstop = m),
  control.tree = list(maxdepth = 6), family = "hinge", learner = "tree")
err.te3 <- predict(dat.m3, newdata = dat2[, -1], newy = dat2[, 1], mstop = m,
  type = "error")
plot(err.te1, type = "l", xlab = "Iteration", ylab = "Test Error", ylim = c(0.05, 0.12))
```
2 Thyroid disease classification

```
dat1 <- "ann-train.data"
dat1 <- read.table(paste(tmp, dat1, sep = ""))
dat2 <- "ann-test.data"
dat2 <- read.table(paste(tmp, dat2, sep = ""))
m <- 400
dat.m1 <- mbst(x = dat1[, -22], y = dat1[, 22], ctrl = bst_control(mstop = m),
control.tree = list(maxdepth = 6), family = "hinge", learner = "tree")
err.te1 <- predict(dat.m1, newdata = dat2[, -22], newy = dat2[, 22], mstop = m,
type = "error")
dat.m2 <- mbst(x = dat1[, -22], y = dat1[, 22], ctrl = bst_control(mstop = m),
control.tree = list(maxdepth = 6), family = "hinge2", learner = "tree")
err.te2 <- predict(dat.m2, newdata = dat2[, -22], newy = dat2[, 22], mstop = m,
type = "error")
dat.m3 <- mhingebst(x = dat1[, -22], y = dat1[, 22], ctrl = bst_control(mstop = m),
control.tree = list(maxdepth = 6), family = "hinge", learner = "tree")
err.te3 <- predict(dat.m3, newdata = dat2[, -22], newy = dat2[, 22], mstop = m,
type = "error")
plot(err.te1, type = "l", xlab = "Iteration", ylab = "Test Error", ylim = c(0.005, 0.01))
points(err.te2, type = "l", lty = "dashed", col = "blue")
points(err.te3, type = "l", lty = "dotted", col = "red")
legend("topright", c("mbst_hinge", "mbst_hinge2", "mhingebst"), lty = c("solid", "dashed", "dotted"), col = c("black", "blue", "red"))
```

3 Satellite image classification

```
train <- "sat.trn"
train <- read.table(paste(tmp, train, sep = ""))
test <- "sat.tst"
test <- read.table(paste(tmp, test, sep = ""))
train[, 37] <- as.numeric(as.factor(train[, 37]))
test[, 37] <- as.numeric(as.factor(test[, 37]))
p <- 37
colnames(train)[1:(p - 1)] <- paste("x", 1:(p - 1), sep = "")
colnames(test)[1:(p - 1)] <- paste("x", 1:(p - 1), sep = ")
```
m <- 600

dat.m1 <- mbst(x = train[, -37], y = train[, 37], ctrl = bst_control(mstop = m),
        control.tree = list(fixed.depth = FALSE, maxdepth = 6, n.term.node = 6),
        family = "hinge", learner = "tree")
err.te1 <- predict(dat.m1, newdata = test[, -37], newy = test[, 37], mstop = m,
        type = "error")
dat.m2 <- mbst(x = train[, -37], y = train[, 37], ctrl = bst_control(mstop = m),
        control.tree = list(fixed.depth = FALSE, maxdepth = 6, n.term.node = 6),
        family = "hinge2", learner = "tree")
err.te2 <- predict(dat.m2, newdata = test[, -37], newy = test[, 37], mstop = m,
        type = "error")
dat.m3 <- mhingebst(x = train[, -37], y = train[, 37], ctrl = bst_control(mstop = m),
        control.tree = list(fixed.depth = FALSE, maxdepth = 6, n.term.node = 6),
        family = "hinge", learner = "tree")
err.te3 <- predict(dat.m3, newdata = test[, -37], newy = test[, 37], mstop = m,
        type = "error")
plot(err.te1, type = "l", xlab = "Iteration", ylab = "Test Error", ylim = c(0,
0.3))
points(err.te2, type = "l", lty = "dashed", col = "blue")
points(err.te3, type = "l", lty = "dotted", col = "red")
legend("topright", c("mbst_hinge", "mbst_hinge2", "mhingebst"), lty = c("solid",
"dashed", "dotted"), col = c("black", "blue", "red"))

4 Glass identification database

dat <- read.delim(dat, sep = ",", header = FALSE)[, -1]
### there is no class 4
table(dat[, 10])
### must recode class label such that the class labels are
### consecutive, which is how the code was designed to work
id <- dat[, 10] > 3
dat[id, 10] <- dat[id, 10] - 1
table(dat[, 10])
p <- ncol(dat)
colnames(dat)[1:(p - 1)] <- paste("x", 1:(p - 1), sep = ")
set.seed(153)
### generate 10 balanced training data and test data, using 9 folds
### for training and one for test
allfolds <- balanced.folds(dat[, 10], nfolds = 10)
omit <- allfolds[[1]]
train <- dat[-omit, ]
test <- dat[omit, ]
m <- 200
dat.m1 <- mbst(x = train[, -p], y = train[, p], ctrl = bst_control(mstop = m),
        control.tree = list(maxdepth = 6), family = "hinge", learner = "tree")
err.te1 <- predict(dat.m1, newdata = test[, -p], newy = test[, p], mstop = m,
type = "error")
dat.m2 <- mbst(x = train[, -p], y = train[, p], ctrl = bst_control(mstop = m),
control.tree = list(maxdepth = 6), family = "hinge2", learner = "tree")
err.te2 <- predict(dat.m2, newdata = test[, -p], newy = test[, p], mstop = m,
type = "error")
dat.m3 <- mhingebst(x = train[, -p], y = train[, p], ctrl = bst_control(mstop = m),
control.tree = list(maxdepth = 6), family = "hinge", learner = "tree")
err.te3 <- predict(dat.m3, newdata = test[, -p], newy = test[, p], mstop = m,
type = "error")
plot(err.te1, type = "l", xlab = "Iteration", ylab = "Test Error", ylim = c(0.15,
0.36))
points(err.te2, type = "l", lty = "dashed", col = "blue")
points(err.te3, type = "l", lty = "dotted", col = "red")
legend("topright", c("mbst_hinge", "mbst_hinge2", "mhingebst"), lty = c("solid",
"dashed", "dotted"), col = c("black", "blue", "red"))

References