1 IRIS DATA

Table 1: OOB error for PP.bagging with bootstrap samples without strata and with strata

	aux	error	error.st
1	5.00000	0.07200	0.06667
2	10.00000	0.05970	0.03704
3	50.00000	0.04444	0.04444
4	100.00000	0.05185	0.03704
5	500.00000	0.04444	0.03704

Ussing PPtree with LDA the error is 0

```
library(PPtree)
Tree.result <- PP.Tree("LDA",iris[training,5],iris[training,1:4])
test <- iris[-training,]
res<-PP.classify(test[,1:4],test[,5],Tree.result,1)
print(res[[1]]/length(res[[2]]))

## [1] 0
```

Ussing the same data we run a random forest and the oob error is around 5%.

```
##
## Call:
##  randomForest(formula = Species ~ ., data = iris[training, ],      importance = TRUE, pro
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 2
##
##         OOB estimate of  error rate: 5.19%
## Confusion matrix:
##            setosa versicolor virginica class.error
## setosa         45          0         0     0.00000
## versicolor      0         43         2     0.04444
## virginica       0          5        40     0.11111
```

2 OLIVE DATA

Table 2: OOB error for PP.bagging with bootstrap samples without strata and with strata

	aux	error	error.st
1	5.00000	0.00000	0.00583
2	10.00000	0.00535	0.00000
3	50.00000	0.00000	0.00000
4	100.00000	0.00000	0.00000
5	500.00000	0.00000	0.00000

```
## [1] 0
```

Using random forest the OOb error us 0%

```
##
## Call:
##  randomForest(formula = Region ~ ., data = d.olive2[training,      ], importance = TRUE,
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 2
##
##         OOB estimate of  error rate: 0%
## Confusion matrix:
##     1  2 class.error
## 1 290  0           0
## 2   0 88           0
```

Basic Output of PPforest for ntree=50

```
## Using  as id variables
## Using  as id variables

## $`OOB estimate or error rate`
## [1] 0
##
## $`Training error`
## [1] 0
##
## $Predicted
##   [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [36] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [71] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [106] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [141] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [176] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [211] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
```

```
## [246] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [281] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [316] 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [351] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [386] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [421] 2
##
## $`Confusion matrix`
##   1 2 class.error
## 1 1 0           0
## 2 0 1           0
##
## $`OOB error Tree`
##  [1] 0.000000 0.000000 0.000000 0.000000 0.007353 0.000000 0.000000
##  [8] 0.000000 0.000000 0.006711 0.000000 0.000000 0.000000 0.000000
## [15] 0.000000 0.021583 0.000000 0.000000 0.000000 0.000000 0.000000
## [22] 0.000000 0.014286 0.000000 0.000000 0.000000 0.000000 0.021127
## [29] 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.013889
## [36] 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.007092
## [43] 0.021429 0.000000 0.000000 0.007634 0.000000 0.000000 0.000000
## [50] 0.000000
```