Bagging

= B_ootstrap + Agg_regate

Jeffrey Leek
Johns Hopkins Bloomberg School of Public Health

Bootstrap aggregating (bagging)

Basic idea:

1. Resample cases and recalculate predictions
2. Average or majority vote

Notes:

- Similar bias , but:
- Reduced variance
- More useful for non-linear functions

Proovable:

The bias you get from bagging is similar, but it has always reduced variance (which is good).

Ozone data

```
library(ElemStatLearn); data(ozone,package="ElemStatLearn")
ozone <- ozone[order(ozone$ozone),] order by outcome (ozone)
head(ozone)
```

```
    ozone radiation temperature wind
17      1         8          59  9.7
19      4        25          61  9.7
14      6        78          57 18.4
45      7        48          80 14.3
106     7        49          69 10.3
7       8        19          61 20.1
```

http://en.wikipedia.org/wiki/Bootstrap_aggregating

Bagged loess

```
ll <- matrix(NA,nrow=10,ncol=155)
for(i in 1:10){
  ss <- sample(1:dim(ozone)[1],replace=T)
  ozone0 <- ozone[ss,]; ozone0 <- ozone0[order(ozone0$ozone),]
  loess0 <- loess(temperature ~ ozone,data=ozone0,span=0.2)
  ll[i,] <- predict(loess0,newdata=data.frame(ozone=1:155))
}
```

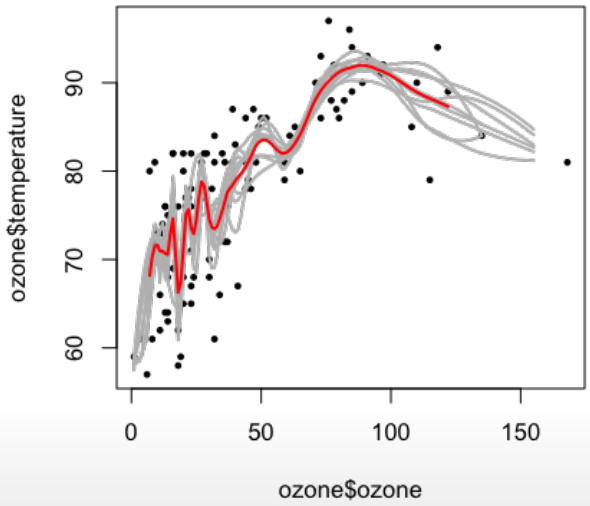
sample w/ replacement

choose and sort again by ozn

fit a 'loess' curve (similar to spline)
span is a measure for smoothness of curve
We're predicting temperature from ozone,
10 times

Bagged loess

```
plot(ozone$ozone,ozone$temperature,pch=19,cex=0.5)
for(i in 1:10){lines(1:155,ll[i,],col="grey",lwd=2)} plot the 10 predicted curves.
lines(1:155,apply(ll,2,mean),col="red",lwd=2) plot the mean of the 10 predictions = bagged loess curve
```



Bagging in caret

- Some models perform bagging for you, in `train` function consider `method` options
  - `bagEarth`
  - `treebag`
  - `bagFDA`
- Alternatively you can bag any model you choose using the `bag` function

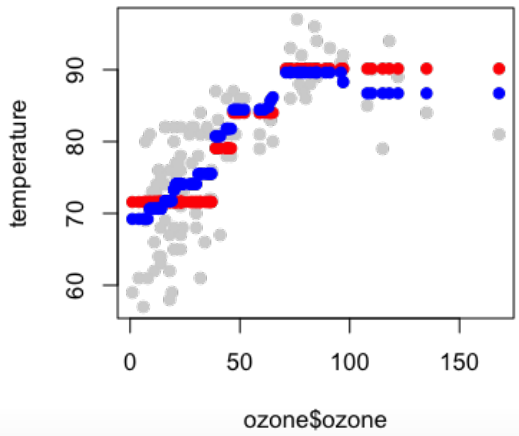
More bagging in caret

```
predictors = data.frame(ozone=ozone$ozone)
temperature = ozone$temperature
                                outcome      num iterations
treebag <- bag(predictors, temperature, B = 10,
               bagControl = bagControl(fit = ctreeBag$fit,
                                       predict = ctreeBag$pred,
                                       aggregate = ctreeBag$aggregate))
```

http://www.inside-r.org/packages/cran/caret/docs/nbBag

Example of custom bagging (continued)

```
plot(ozone$ozone,temperature,col='lightgrey',pch=19)
points(ozone$ozone,predict(treebag$fits[[1]]$fit,predictors),pch=19,col="red")
points(ozone$ozone,predict(treebag,predictors),pch=19,col="blue")
```



Parts of bagging

```
ctreeBag$fit
```

```
function (x, y, ...)
{
    library(party)
    data <- as.data.frame(x)      the preditors x
    data$y <- y                   the outcome y
    ctree(y ~ ., data = data)     then returns the result of the ctree function
}
<environment: namespace:caret>
```

Parts of bagging

```
ctreeBag$pred
```

```
          object is a ctree$model-fit, x is a predictor matrix
```

```
function (object, x)
{
    obsLevels <- levels(object@data@get("response")[, 1])
    if (!is.null(obsLevels)) {
        rawProbs <- treeresponse(object, x)
        probMatrix <- matrix(unlist(rawProbs), ncol = length(obsLevels),
            byrow = TRUE)
        out <- data.frame(probMatrix)
        colnames(out) <- obsLevels
        rownames(out) <- NULL
    }
    else out <- unlist(treeresponse(object, x))
    out
}
<environment: namespace:caret>
```

Parts of bagging

```
ctreeBag$aggregate
```

```
function (x, type = "class")
{
    if (is.matrix(x[[1]]) | is.data.frame(x[[1]])) {
        pooled <- x[[1]] & NA
        classes <- colnames(pooled)
        for (i in 1:ncol(pooled)) {
            tmp <- lapply(x, function(y, col) y[, col], col = i)
            tmp <- do.call("rbind", tmp)
            pooled[, i] <- apply(tmp, 2, median)
        }
        if (type == "class") {
            out <- factor(classes[apply(pooled, 1, which.max)],
                levels = classes)
        }
        else out <- as.data.frame(pooled)
    }
    else {
        x <- matrix(unlist(x), ncol = length(x))
```

Notes and further resources

Notes:

- Bagging is most useful for nonlinear models
- Often used with trees - an extension is random forests
- Several models use bagging in caret's *train* function

Further resources:

- Bagging
- Bagging and boosting
- Elements of Statistical Learning