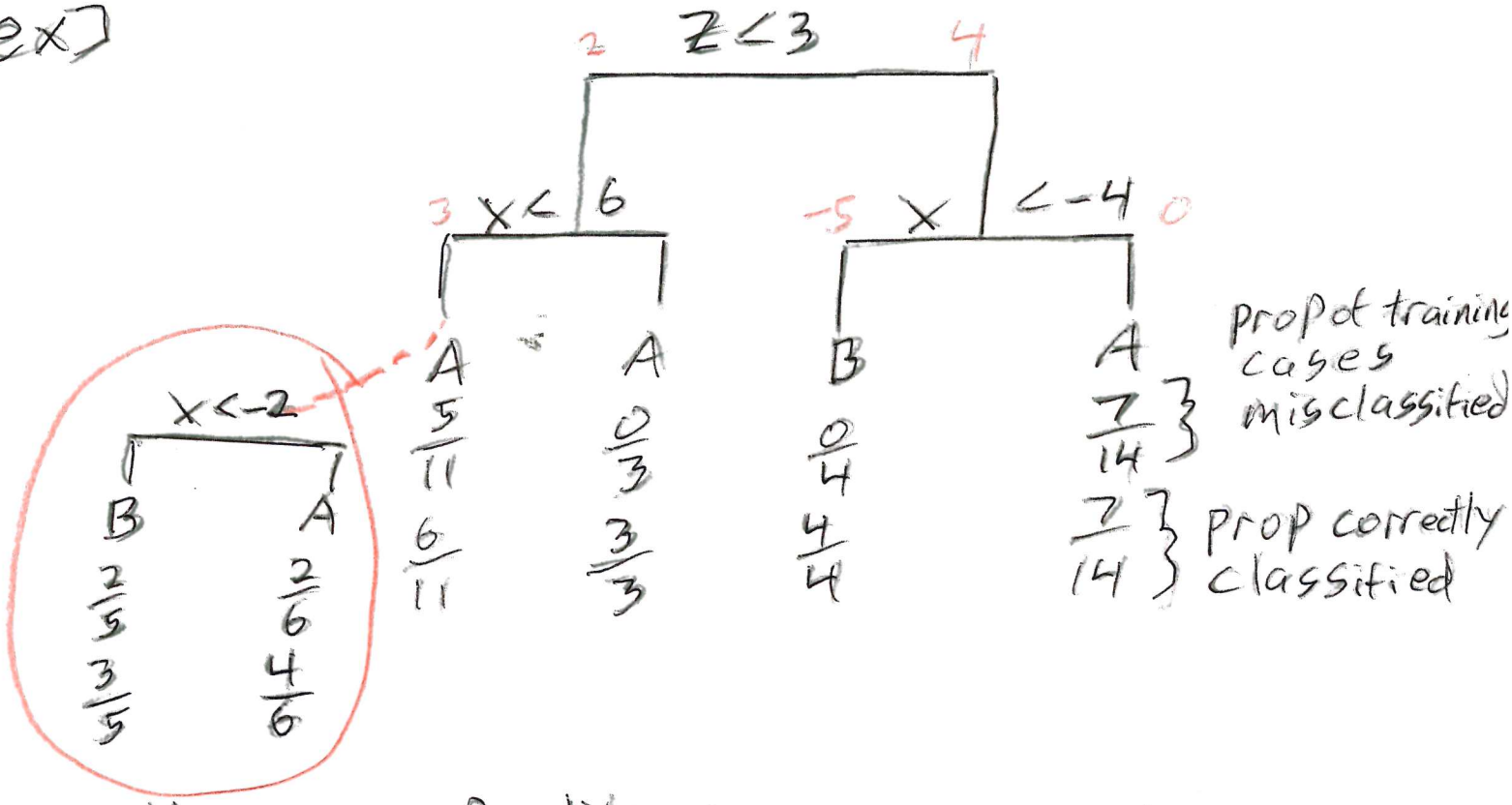
A single vertical line at $z = 3$ is the 1st partition.

The horizontal line at $x = -4$ is the 2nd partition.

" $x = 6$ 3rd "

ex]



add one more partition

6] Trees i) give prediction rules that can be rapidly and repeatedly evaluated,

ii) are useful for screening predictors (variable selection, interactions = products of variables like $x_1 x_2$, $x_3 x_5 x_7$)

iii) can be used to summarize large multivariate data sets

iv) regression trees can be used to assess the adequacy of linear models.

7] If Y is a categorical factor with levels 1, ..., G, classification rules are of the form "if $x_1 \leq 2.3$ and $x_3 \in \{A, B\}$, then Y is most likely to be in level 5."

If Y is numerical, regression rules are of the form "if $x_2 \leq 1.7$ and $x_9 \in \{C, D, F\}$ and $x_5 \leq 3.5$, then $\hat{Y} = 4.75$."

written $x_9 = C, D, F$

8] Trees can be easier to interpret than MLR when some predictors are numerical and some categorical. Trees are invariant to monotone (increasing or decreasing) transformations of the predictors x_i.

Trees can handle complex unknown interactions.

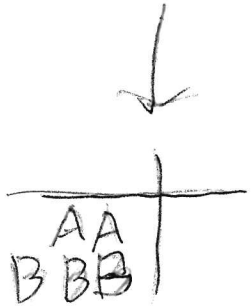
9] Regression trees handle missing values better

than MLR, and can beat MLR if there is nonadditive behavior ($Y = m(x) + e$).

10] In growing a tree, the binary partitioning algorithm recursively splits the data in each node until either the node is homogeneous (roughly 0 training data misclassifications for a classification tree, $Y \simeq$ constant for a regression tree) or the node contains too few observations (default ≤ 5).

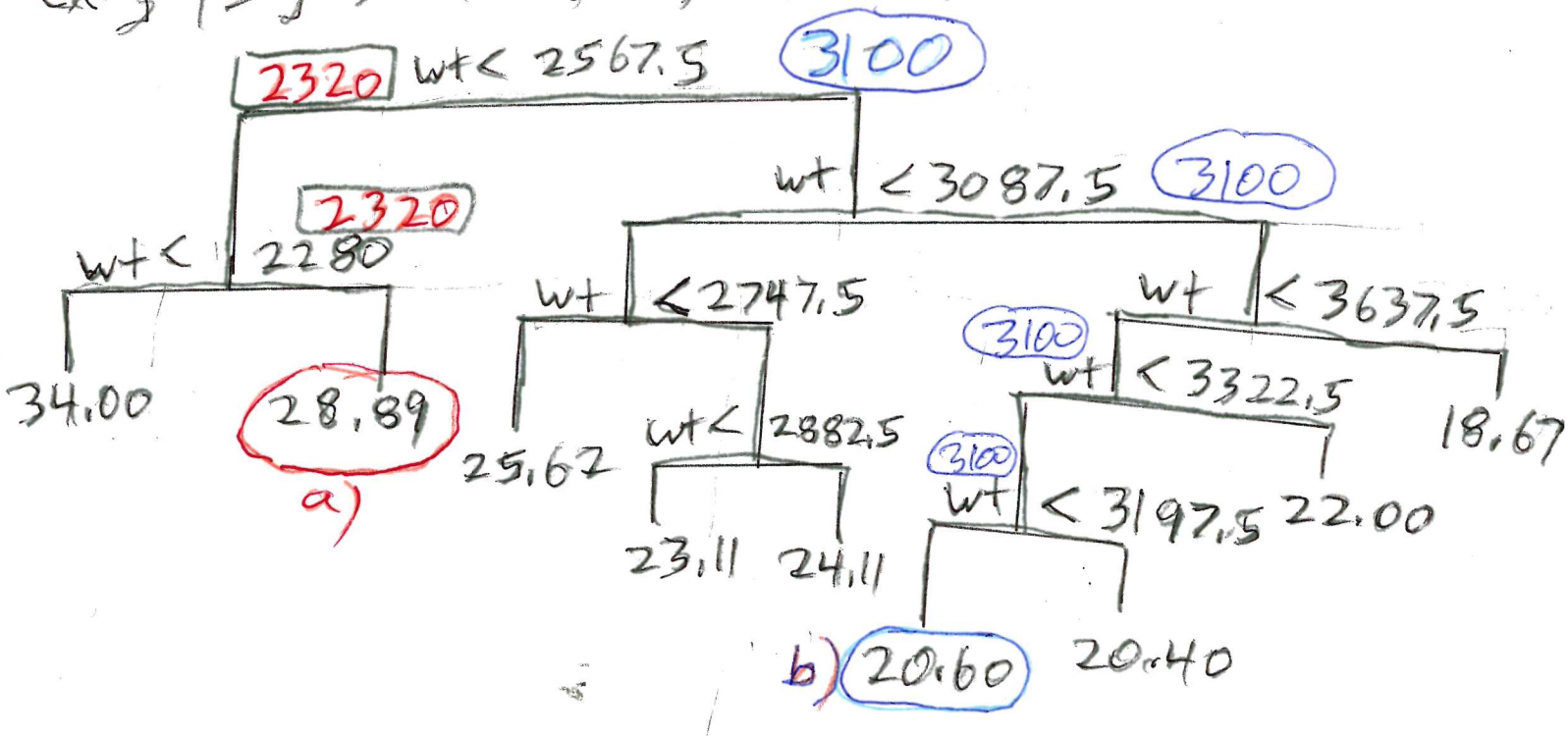
11] The deviance is a measure of node homogeneity, and deviance = 0 for a perfectly homogeneous node.

12] Often { use the mean of the region for $\hat{Y}$, reg tree
mode $\hat{Y}$, classification tree



mode = B

ex} Y = gas mileage, x = weight of car



a) Predict mileage if wt = 2320

$\hat{Y} = 28.89$

see red: 2320 left
2320 right

b) predict mileage if wt = 3100

$\hat{Y} = 20.60$

see blue R right R left

13} a) Divide the predictor space

(= set of possible values for $\underbrace{X_1, \ldots, X_p}_{\text{training data}}$) into

J distinct and nonoverlapping regions

$R_1, \ldots, R_J$.

b) For every observation that falls in

region R_j, make the same prediction HD 93

$$\hat{Y}_{R_j} = \begin{cases} \bar{Y}_j = \text{sample mean of } Y_i \text{ in } R_j & \text{reg tree} \\ \text{mode}_j = \text{mode} & \text{class. tree} \end{cases}$$

14) The R_j are high dimensional boxes rectangles.

Choose R_j so $RSS \overset{\text{reg}}{=} \sum_{j=1}^{J} \sum_{i \in R_j} (Y_i - \hat{Y}_{R_j})^2$

or $RSS \overset{\text{class.}}{=} \sum_{j=1}^{J} \sum_{i \in R_j} I(Y_i \neq \hat{Y}_{R_j})$

is small.

15] Let $\{\underline{x} \mid x_j < s\}$ be the region in the predictor space such that $x_j < s$ where $\underline{x} = (x_1, \ldots, x_p)^T$. Define 2 regions

$R_1(j,s) = \{\underline{x} \mid x_j < s\}$, $R_2(j,s) = \{\underline{x} \mid x_j \geq s\}$

and seek "cutpoint" s and j to minimize

$$\sum_{i:\, \underline{x}_i \in R_1(j,s)} (y_i - \hat{y}_{R_1})^2 + \sum_{i:\, \underline{x}_i \in R_2(j,s)} (y_i - \hat{y}_{R_2})^2$$

for reg trees.

This can be done "quickly" if p is small (could use order statistics).

Then repeat the process looking for the best predictor and best cutpoint in order to split the data further so as to minimize the RSS within each resulting region. Only split one of the regions. So now there are regions R_1, R_2, and R_3. Continue this process until a stopping criterion is reached such as no region contains less than 5 obs's (and stop splitting if the region is homogeneous, eg all Y in region belong to class k for class tree).

16] classification trees are usually interior to LDA and SVM. Bagging, random forests, or boosting makes trees more competitive.

17] Trees use regions $R_1, \ldots, R_J$. If J is too large, the tree overfits (not good for test data). One

strategy is to grow a large tree T_0 with J_0 regions. Then "prune" it to get a subtree T_α with J_α regions.

18) cost complexity pruning = weakest link pruning:

Let $T \subseteq T_0$, $\alpha \geq 0$, and $|T|$ = # terminal nodes of tree T. Each terminal node corresponds to a region (hyperrectangle = box) R_i. Let R_m be the region corresponding to the mth terminal node and $\hat{y}_{R_m}$ be the predicted response for R_m.

For each value $\alpha > 0$, there corresponds a subtree $T \subseteq T_0$ such that

$$(*) \quad \sum_{m=1}^{|T|} \sum_{i : x_i \in R_m} (y_i - \hat{y}_{R_m})^2 + \alpha |T| \text{ is as small as possible.}$$

(use $I(y_i \neq \hat{y}_{R_m})$ for classification.) Note that $\alpha = 0$ has $T = T_0$ and $(*) = RSS(T_0) =$ training data

RSS for T_0.

19] Much like lasso, as α increases, there is a sequence of nested subtrees

(**) $T_0 \supseteq T_{\alpha_1} \supseteq T_{\alpha_2} \supseteq \cdots \supseteq T_{\alpha_m}$. Then branches get "pruned" from T_0 in a nested and predictable fashion.

20] a) Build tree T_0, stopping when each node has ≤ 5 obs's (or is homogeneous) further splitting would cause fewer than 5 obs's

b) Do 18) to obtain subsequence (**).

c) use K-fold CV to choose $\alpha = \alpha_j^*$.

For each $i \in 1, \ldots, K$

i) repeat steps a) and b) on all but the ith fold.

ii) Evaluate the mean square prediction error on data left out of fold i as a function of α. Average the results for each value of α and pick α_j^* to minimize the average error.

d) use tree $T_{\alpha_j^*}$ from b).

For a regression tree,

$$MSE_i = \frac{1}{n_i} \sum_{j=1}^{n_i} (Y_{ji} - \hat{Y}_j(i))^2$$ for data

Y_{ji} in the left out fold i.

$$CV_{(K)} = \frac{1}{K} \sum_{i=1}^{K} MSE_i.$$

For a classification tree, use

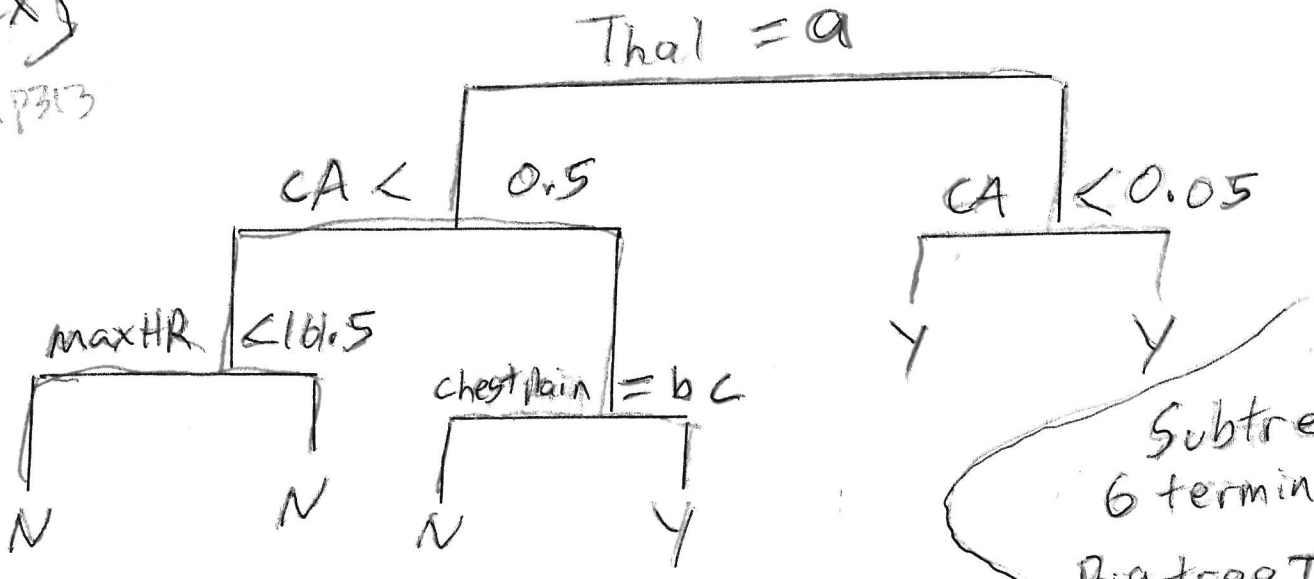
$$MSE_i = \frac{1}{n_i} \sum_{j=1}^{n_i} I\left[Y_{ji} \neq \hat{Y}_j(i)\right] =$$

proportion misclassified in the ith fold.

If $n_i = \frac{n}{K}$, then

$$CV_{(K)} = \begin{cases} \frac{1}{n} \sum_{j=1}^{n} (Y_{ji} - \hat{Y}_j(i))^2, & \text{reg tree} \\ \frac{1}{n} \sum_{j=1}^{n} I\left[Y_{ji} \neq \hat{Y}_j(i)\right], & \text{class tree} \end{cases}$$

ex} J. p313



Subtree 6 terminal nodes

Big tree To added many branches and had 18 terminal nodes.

303 patients with chest pain

Y = heart disease

N = no heart disease

Thal = Thalium stress test: a = normal

Chestpain a = typical aginal, b = atypical aginal,

c = non-aginal pain, d = asymptomatic

maxHR maximum heart rate?

CA blood calcium?

21} For a tree T_α, $Y_i = m(\underline{x}_i) + \sigma_i e_i$

and $\hat{Y} = \hat{m}(\underline{x}_i) = \sum_{m=1}^{J_\alpha} c_m I(\underline{x}_i \in R_m)$ where

T_α uses regions $R_1, \ldots, R_{J_\alpha}$ and $c_m = \hat{Y}_m$ for $\underline{x}_i \in R_m$.

22] Trees can handle categorical variables (qualitative, factors) without creating indicators = dummy variables.

23] Bagging was used with the bootstrap: compute $T_1^*, \ldots, T_B^*$ with the bootstrap and the sample mean $\bar{T}^* = \frac{1}{B}\sum_{i=1}^{B} T_i^*$ is the bagging estimator.

24) For a regression tree, draw a sample of size n with replacement from $\underline{x}_1, \ldots, \underline{x}_m$. Fit the tree and find $\hat{f}_1(\underline{x})$. Repeat B times to get $\hat{f}_1^*(\underline{x}), \ldots, \hat{f}_B(\underline{x})$. Then the bagging estimator $\hat{f}_{bag}^*(\underline{x}) = \frac{1}{B}\sum_{i=1}^{B} \hat{f}_i(\underline{x})$.

The trees are not pruned, so terminate when each terminal node has 5 or fewer obs's or is homogeneous.

if further splitting would result in fewer than 5 obs's

25] For classification, draw a sample of size n_j from each group with replacement. Let

$\hat{f}_i^*(x) = j_i(x) \in \{1, \ldots, G\}$ where (groups)

Y takes on G levels $1, \ldots, G$.

Compute $\hat{f}_1^*(x), \ldots, \hat{f}_B^*(x)$ and let

$m_k = \#\, j_i(x) = k$ for $k = 1, \ldots, G$.

Take $\hat{f}_{bag}^*(x) = d$ where $m_d =$

$\max\{m_1, \ldots, m_G\}$.

26] As before, the prob that x_j is not in the bootstrap data set $\to e^{-1} \approx .3679$ $\approx \frac{1}{3}$ as $n \to \infty$ (case j)

$\frac{1-\frac{1}{n}}{1} \cdots \frac{1-\frac{1}{n}}{n}$

$\left(1-\frac{1}{n}\right)^n \to e^{-1}$

prob x_j is not the kth obs $= 1-\frac{1}{n}$ since each obs has $\frac{1}{n}$ chance of being selected for the kth position

27] For each bootstrap data set b,

let $\underline{x}_{i_1}, \ldots, \underline{x}_{i_{K_b}}$ be the K_b obs's not used in b. These are out of bag (OOB) obs's. Predict $\hat{y}$ for each OOB obs. Doing this for all B bootstrap data sets produces about $\frac{B}{3}$ predictions for each $\underline{x}_i$. Let $\hat{y}_{i0} = \begin{cases} \text{ave } \hat{y}_i & \text{reg tree} \\ \text{mode level} & \text{class} \end{cases}$

$$\text{OOB MSE} = \begin{cases} \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_{i0})^2 & \text{reg} \\ \frac{1}{n}\sum_{i=1}^{n} I(y_i \neq \hat{y}_{i0}) & \text{class} \end{cases}$$

The OOB MSE is "virtually equivalent" to the leave one out CV estimate for sufficiently large B.

28] Bagging typically gives better accuracy than a single tree.

29) For classification trees, let

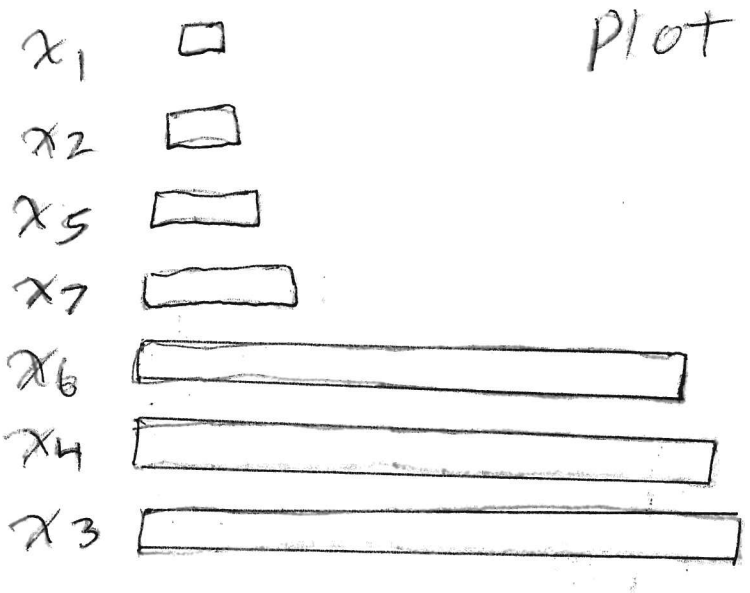
$\hat{p}_{mk}$ = proportion of training obs's in R_m from the kth class.

Gini's index $= \sum_{k=1}^{G} \hat{p}_{mk}(1-\hat{p}_{mk})$

is small if all $\hat{p}_{mk}$ are close to 0 or 1.

30) For bagging with B trees, a measure JP 3.4 of variable importance can be computed foreach variable using splits for each variable.

variable importance plot

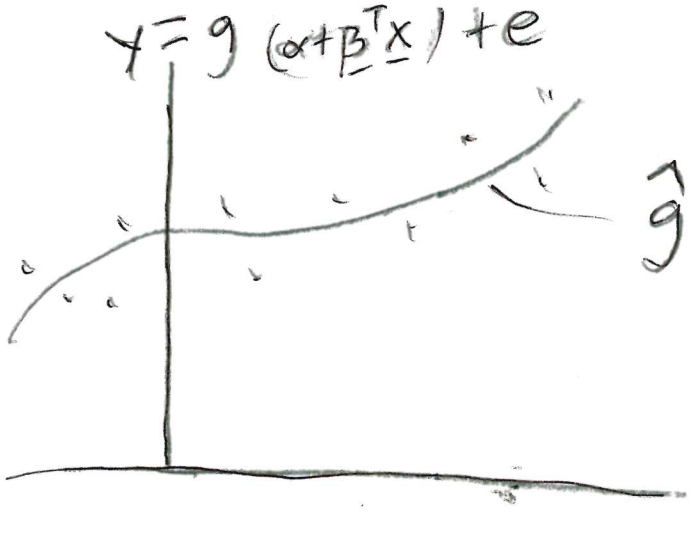


x_3, x_4, x_6 are the most important predictors for the tree

31) If $Y = \alpha + \sum_{j=1}^{p} s_j(x_j) + e$ or $Y = m(\underline{x}) + e$

with $m(\underline{x}) = g(\alpha + \underline{\beta}^T \underline{x})$, then slicing the ESP $\hat{\alpha} + \sum_{j=1}^{p} \hat{s}_j(x_j)$ or $\hat{\alpha} + \hat{\underline{\beta}}^T \underline{x}$

is more effective than partitioning the predictor space $\underline{x}$ with hyperboxes R_m (curse of dimensionality $\mathbb{R}^p$).



If $Y = g(w)$ and you plot w vs Y, then you "see g"!
so if $Y = g(SP) + e$, you "see g" up to noise of $ESP \approx SP \approx w$ and e_i.

32] MS project If $Y = m(\underline{x}) + e$, we can make prediction intervals for Y_f with the regression tree using $\hat{Y} = ESP = \hat{m}(\underline{x})$ and $r = Y - \hat{Y}$ as before.

Random Forests

33] For random forests, the bootstrap is used, but each time a split is considered, a random sample of $m = \lceil \sqrt{p} \rceil \approx \sqrt{p}$ predictors is chosen as split candidates.

Much faster than examining all p variables.

estimator

Random forests produces bootstrap trees that are less correlated than bagged trees (that use $m = p$) and the random forests estimator tends to have better test error and OOB error than the bagging estimator. 98.5

34] B around a few hundred seems to work.

35] If there is a single strong predictor, bagged trees tend to use that predictor in the 1st split. For random forests, the strong predictor is not considered for $\frac{p-m}{p}$ splits, on average. Trees from random forests also tend to be less correlated than bagged trees if there are many correlated predictors.

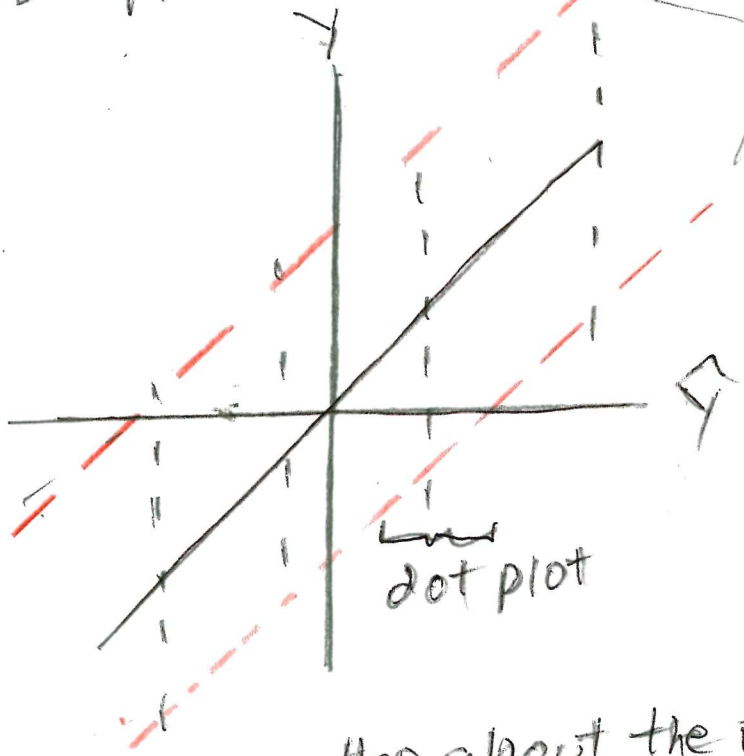
36] For classification, the null classifier has $\hat{y} = d$ where d is the dominant (mode) class. So if K% of obs's belong to the dominant class, the test error $= \frac{100-K}{100} \leq 1 - \frac{1}{G}$ where there are G groups, since $K \geq 100\frac{1}{G}$.

JP 321

(assumes training data dominant class = that of test data)

classifiers that do not beat the null classifier are very bad.

37] Since a reg tree uses J_α regions, the response plot of $ESP = \hat{Y} = \hat{m}(x)$ vs Y looks like

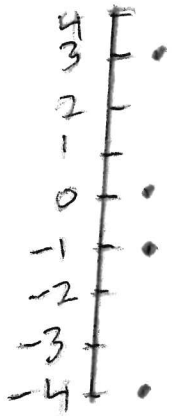


The plotted points scatter about the identity line, but there is a dot plot of n_m cases with $\hat{Y} = \hat{Y}_{Rm}$ for each of the J_α regions.

(One way Anova models have a similar dot plot, but each dot plot crosses the identity line at $\hat{Y}_i = \overline{Y}_{i0}$.)

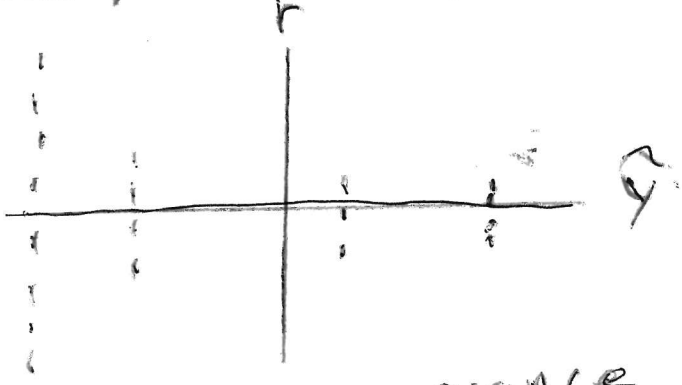
38] A dot plot of $Z_1, \ldots, Z_m$ consists of an axis and m points corresponding to the values of Z.

ex] $z_1 = 3, z_2 = -1, z_3 = 0,$
$z_4 = -4$



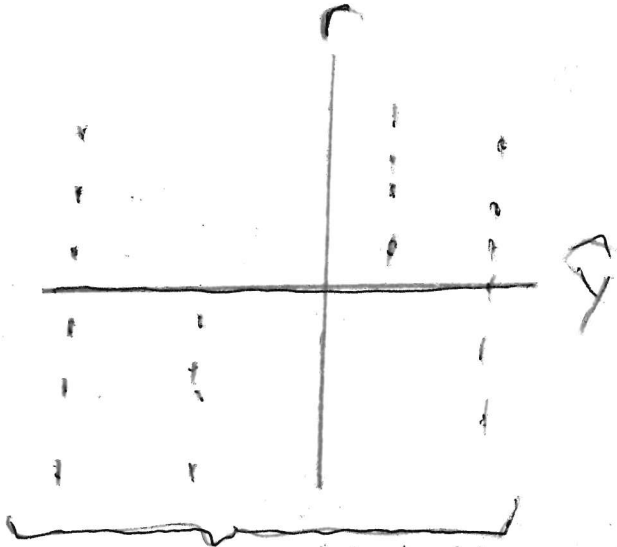
dot plot

39] The residual plot consists of dot plots that scatter about the $r = 0$ line.



nonconstant variance

$y_i = m(x_i) + \sigma_i \varepsilon_i$



evidence that the fit could be better

JS 8.2.3

40] The third technique for improving trees is <u>boosting</u>. Like bagging, boosting can be applied to many statistical methods, not just trees.

41] Boosting for regression trees

i). set $\hat{f}(x) \equiv 0$ and $r_i = y_i$, $i = 1, \ldots, n$
↑ training data