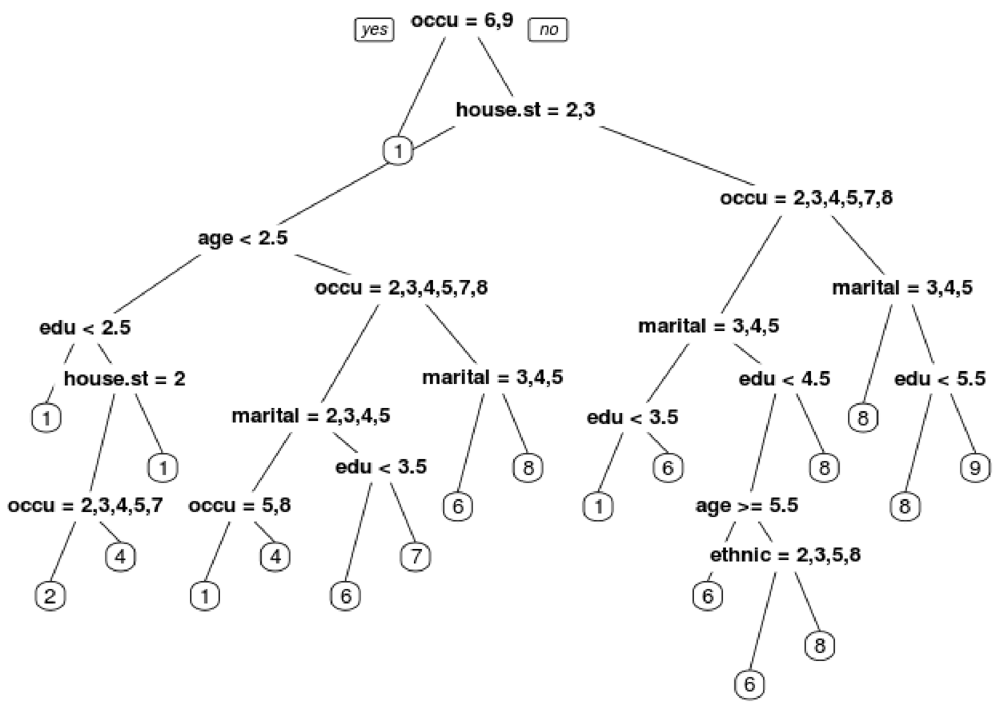
Statistics 315B
Homework 01

Amy Zhang (amyxz@stanford.edu)
Mina Jean Hanna (mjhanna@stanford.edu)
Wei Xia (wei4@stanford.edu)

April 28, 2016

1. Income Tree:



The primary split is on Occupation; being unemployed or a student was very predictive of low income. To name a few other variables, owning a house tended to indicate higher income, as did an older age and greater education.

When cp = 7.5883e-04, the pruned tree is optimal in terms of minimal cross validation error. But the tree has 36 splits, which is not easy to print here. In order to show the tree below, we chose cp = 1.7936e-03:

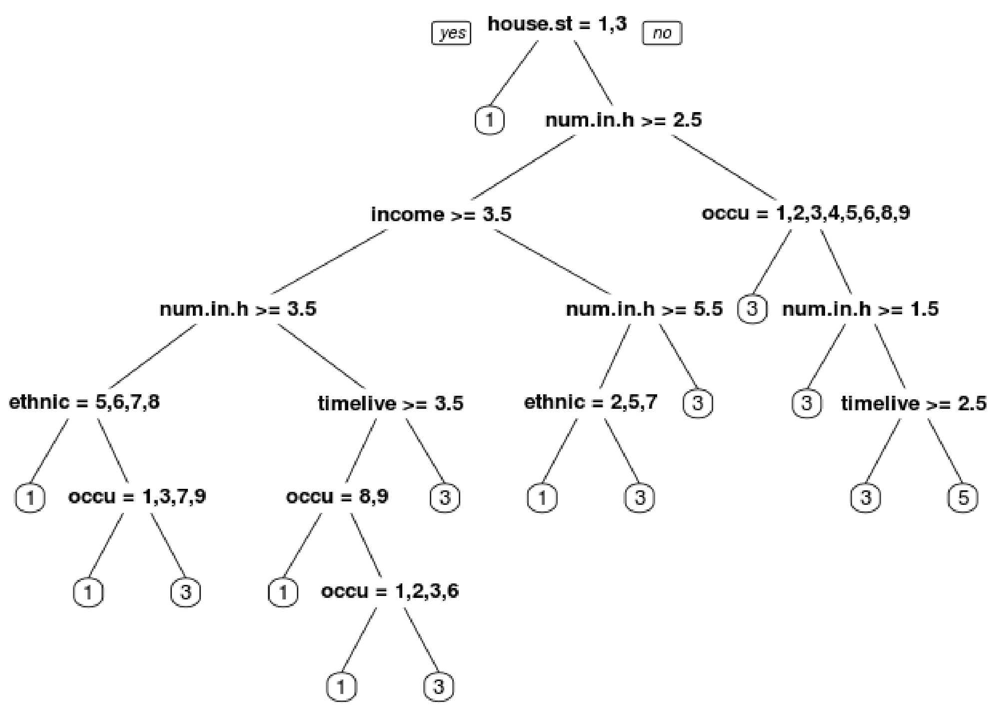
(a) Surrogate splits were used. A surrogate split is an alternative split in the decision tree to be used in the case that the primary split variable is missing. The surrogate is the best mimic/predictor of the primary split because it is correlated with the primary split.
An example is the root node, with a primary split at Occupation, but a surrogate of Age :

```
Node number 1: 8993 observations,    complexity param=0.08498896
```

```
predicted class=1
Primary splits:
    OCCUPATION         splits as  RRRRRLRRL, improve=495.5213, (136 missing)
    AGE                < 1.5 to the left,    improve=495.4613, (0 missing)
    EDUCATION          < 2.5 to the left,    improve=400.2626, (86 missing)
    HOUSEHOLDER.STATUS splits as  RRL,       improve=396.2327, (240 missing)
    MARITAL.STATUS     splits as  RRRRL,     improve=295.3057, (160 missing)
Surrogate splits:
    AGE                < 1.5 to the left,  agree=0.859, adj=0.314, (136 split)
    EDUCATION          < 2.5 to the left,  agree=0.832, adj=0.185, (0 split)
    HOUSEHOLDER.STATUS splits as  RRL,     agree=0.832, adj=0.185, (0 split)
```

(b) I am employed full time, I am not a home owner, my age is less than 30 years, I have graduate degree, and continuning down the tree, it predict 4. 20,000 - 24,9999. (However, this is wildly incorrect.)

2. Housetype Tree:



This tree indicates that whether you own, rent, or live with family, a good indication of the type of home that you have. This makes intuitive sense. In addition, a large number of individuals in your hour is associated with having a larger type of house, and individuals appear to live longer in larger houses.

Missclassification Error $\left(\frac{1}{N_m}\sum_{i\in R_m} I(y_i \neq k(m))\right)$:

printcp indicates what the root node error and relative error are. The missclassification error is the product of the root node error and relative error:

```
printcp(pruned.housetype)

Classification tree:
    rpart(formula = typeofhouse ~ ., data = df.housetypedata, method = "class",
```

```
    control = temp)

Variables actually used in tree construction:
[1] ethnic       house.status income       num.in.house occu         timelive

Root node error: 3694/9013 = 0.40985

n= 9013

          CP nsplit rel error  xerror     xstd
1  0.3286410      0   1.00000 1.00000 0.012640
2  0.0167840      1   0.67136 0.67136 0.011478
3  0.0078506      3   0.63779 0.64104 0.011311
4  0.0048728      4   0.62994 0.63563 0.011280
5  0.0025717      5   0.62507 0.63211 0.011260
6  0.0018950      7   0.61992 0.63021 0.011249
7  0.0017596      8   0.61803 0.63021 0.011249
8  0.0016243     10   0.61451 0.63102 0.011254
9  0.0012182     11   0.61289 0.63021 0.011249
10 0.0010828     13   0.61045 0.62859 0.011239
11 0.0010828     14   0.60937 0.62805 0.011236
```

At this cp (cp = 0.0010828), we get the optimal tree, so the
missclassification error = 0.60937 * 0.40985 = 0.25748

3. The target function is the function that minimizes the risk of incorrect prediction on future data.

   Tha target function will not be accurate if training data is used to calculate the target function, but then the generating function for data changes when producing future data. Unfortunately this is many real life senarios.

4. The empirical risk would not be accurate if the training data is not a representative sample of the population data.

5. In order to be useful, this search algorithm must been guaranteed to produce a function in a finite amount of time. Thus, the set of all functions that we search over must inherently be smaller than the infinite set of all possible functions.

6. The bias-variance tradeoff is the balancing act between fitting data closely, and being able to generalize.

   When we choose a function with excessively high bias, we underfit the data and have not modeled the complexity of the process. For example, if using linear regression to model an exponential process, large values of x may be underestimated.

   When we choose a function with excessively high variance, we overfit the data and have modeled nonexistance complexities in the data. For example, if using a 21 degree polynomial to model an linear process with random variance for which we only have 20 datapoints,

7. Withdrawn

8. We should not choose a surrogate split first because by definition the primary split is the split that is the most influential split on the data.

9. **Lemma :** Based on the Cauchy Inequality, it is easy to show: $\frac{\sum_{i=1}^{N} a_i^2}{N} \geq \left(\frac{\sum_{i=1}^{N} a_i}{N}\right)^2$

   Proof:

With the above inequality, we can get:

$$\begin{aligned} min\sum_{i=1}^{N}[y_i - F(x_i)]^2 &= \sum_{m=1}^{M}\left(\sum_{x_i\in R_m}[y_i - F(x_i)]^2\right) \\ &\geq \sum_{m=1}^{M}\frac{(\sum_{x_i\in R_m} y_i - \sum_{x_i\in R_m} F(x_i))^2}{\sum_{x_i\in R_m} I(x_i\in R_m)} \end{aligned} \tag{1}$$

Note that:

$$\begin{aligned} F(x_i) &= \sum_{m=1}^{M} c_m I(x_i\in R_m) \\ &= c_m I(x_i\in R_m) \\ \sum_{i\in R_m} F(x_i) &= \sum_{x_i\in R_m} F(x_i) I(x_i\in R_m) \\ &= c_m \sum_{x_i\in R_m} I(x_i\in R_m) \end{aligned} \tag{2}$$

In order to minimise equation (1), and substitute (2) into (1), we can get:

$$\begin{aligned} \left(\sum_{x_i\in R_m} y_i - \sum_{i\in R_m} F(x_i)\right) &= 0 \\ c_m \sum_{x_i\in R_m} I(x_i\in R_m) &= \sum_{i\in R_m} y_i \\ c_m &= \frac{\sum_{x_i\in R_m} y_i}{\sum_{x_i\in R_m} I(x_i\in R_m)} \end{aligned} \tag{3}$$

10.

$$\begin{aligned} Improvement &= -\left(\sum_{i\in R_l}(y_i - \bar{y_l})^2 + \sum_{i\in R_r}(y_i - \bar{y_r})^2 - \sum_{i\in R_m}(y_i - \bar{y_n})^2\right) \\ &= -\left(\sum_{i\in R_l}(\bar{y_l}^2 - 2y_i\bar{y_l}) + \sum_{i\in R_r}(\bar{y_i}^2 - 2y_i\bar{y_r}) - \sum_{i\in R_m}(\bar{y_i}^2 - 2y_i\bar{y_n})\right) \end{aligned} \tag{4}$$

Because $\sum_{i\in R_l}(\bar{y_l}) = n_l\bar{y_l}$ and $\sum_{i\in R_r}(\bar{y_r}) = n_r\bar{y_r}$

$\bar{y}_n = \frac{n_l\bar{y}_l + n_r\bar{y}_r}{n}$

$n = n_l + n_r$

Substitute above functions to improvments, then we can have

Improvment $= \frac{n_l n_r(\bar{y}_l - \bar{y}_r)^2}{n}$

11. Without loss of generality, assume that we move one observation y_* from R_l to R_r,

$$\begin{aligned} \bar{y_l}\prime &= \frac{n_l * \bar{y_l} - y_*}{(n_l - 1)} \\ \bar{y_r}\prime &= \frac{n_r * \bar{y_r} + y_*}{(n_r + 1)} \\ Improvement &= \frac{(n_l - 1)(n_r + 1)(\bar{y_l}\prime - \bar{y_r}\prime)^2}{n} - \frac{n_l n_r(\bar{y_l} - \bar{y_r})^2}{n} \end{aligned} \tag{5}$$

12. Surrogate Splits may not be effective if variables are uncorrelated, and another method may be more effective.

    As compared to Surrogate Splits, treating Missing As Terminal is more efficient, but has higher bias. Mising as Terminal will not continue the tree, and so will not model variation in data missing that value. This may be especially impactful if a variable at the root of the tree is missing, and instead of looking for otherways to predict y, we stop searching. However, if that value is especially predictive of the data and a meaningful prediction can not be made without it, Missing as Terminal may be advantageous. For example, if we

    As compared to Surrogate Splits, which uses the variable with the most similar mean, using Max Training Data uses the most common, or modal response of the split. Max Training Data may thus have an advantage in cases where the mode of a variable is highly indicative of an accurate prediction.

13. By including a branch for "missing," we loose the computational advantages of a binary tree. This strategy is likely to show a surrogate effect because the child split of the node that represents "missing" is likely to be on a variable correlated with the parent node – it does not, however, guarantee it and there is no mechanism that directly encourages it. A dataset without missing values with this strategy alone would not produce a meaningful tree, but we can try different techniques for dealing with them.

    One idea is to :

    (a) determine the primary split
    (b) pretend that the primary variable does not exist in the dataset
    (c) determine what the primary split would be in the complete absence of that variable
    (d) use the variable from (3) as the child of the "missing" branch

    This would produce an a way to proceed at any variable that is missing, and the next decision would likely be over another similarly meaningful variable.