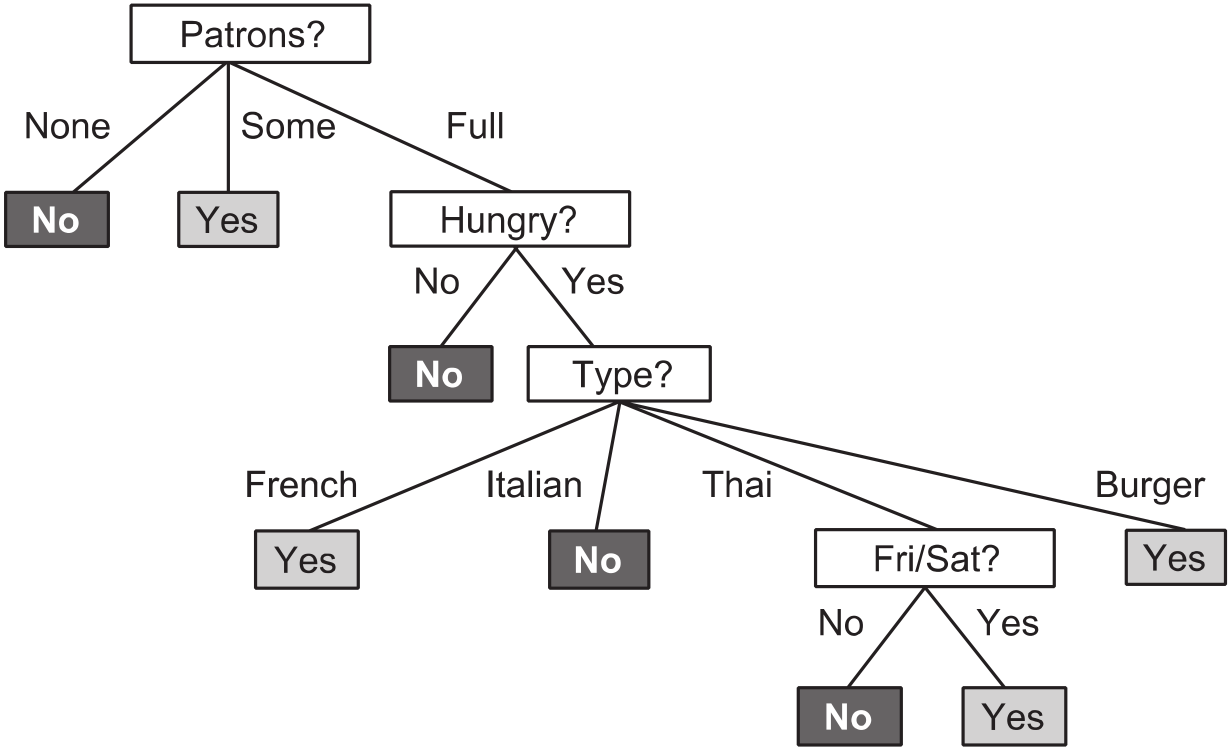
Lecture 4: AdaBoost

Course Teacher: Md. Shariful Islam Bhuyan

Decision tree



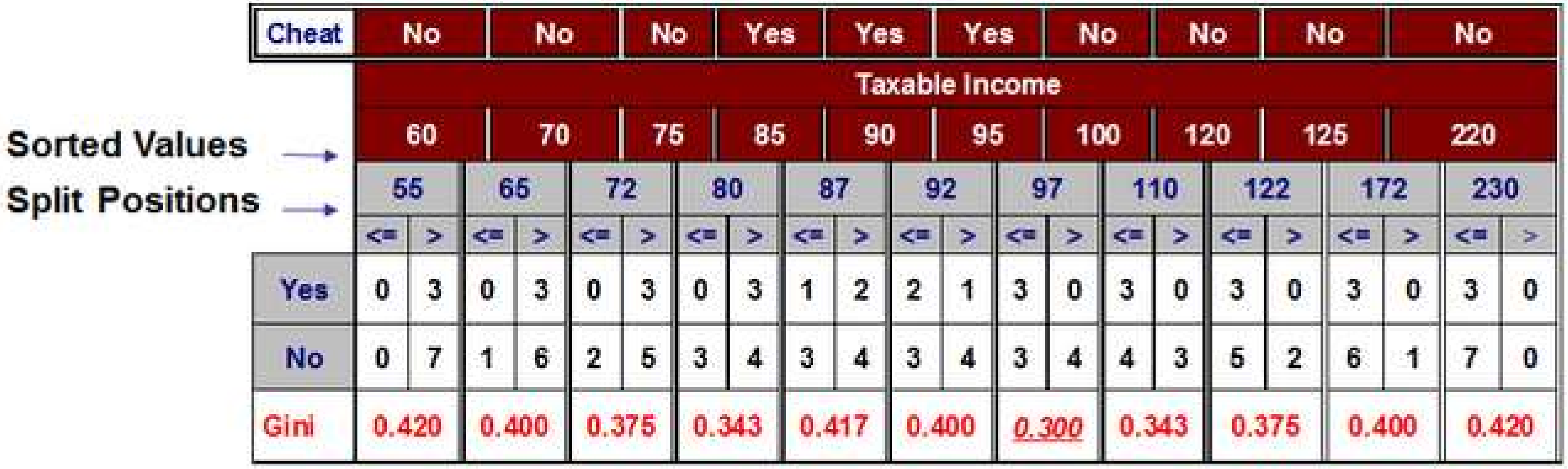
Detecting Overfitting

- Train-test split/holdout cross validation
- Poor performance on test data
- Did not learn to generalize
    - Extreme case: table lookup
- Peeking
- Combat pruning

Continuous Valued Input

- Find the split point that gives the highest information gain
- Sort examples according to attribute values
- Consider only split points that are between two examples in sorted order that have different classifications
- Keep track of the running totals of positive and negative examples on each side of the split point

Example



Cheat	No	No	No	Yes	Yes	Yes	No	No	No	No
	Taxable Income									
Sorted Values →	60	70	75	85	90	95	100	120	125	220

Split Positions →	55		65		72		80		87		92		97		110		122		172		230	
	<=	>	<=	>	<=	>	<=	>	<=	>	<=	>	<=	>	<=	>	<=	>	<=	>	<=	>
Yes	0	3	0	3	0	3	0	3	1	2	2	1	3	0	3	0	3	0	3	0	3	0
No	0	7	1	6	2	5	3	4	3	4	3	4	3	4	4	3	5	2	6	1	7	0
Gini	0.420		0.400		0.375		0.343		0.417		0.400		*0.300*		0.343		0.375		0.400		0.420	

Ensemble Learning

- Collection, or ensemble, of hypotheses
- Combine predictions with function; majority, additive, multiplicative …
- Boosting
  - Weighted training set
  - Increase weight for misclassified examples
  - Decrease weight for correctly classifier examples
  - Resample new data set
  - Weighted-majority combination of all the K hypotheses

AdaBoost

```
function ADABOOST(examples, algorithm L, No of hypotheses K)  returns a weighted-majority hypothesis
    w, a vector of N example weights, initially 1/N
    for k = 1 to K do
            data ← resample(examples, w)
            h[k] ← L(data);      error ← 0
            for j = 1 to N do
                if $h[k](x_j) \neq y_j$ then error ← error + w[j]
            if error > .5 continue
            for j = 1 to N do
                if $h[k](x_j) = y_j$ then w[j] ← w[j] · error/(1 – error)
            w ← NORMALIZE(w)
            Z[k] ← log[(1 – error)/error]
      return WEIGHTED-MAJORITY(h, z)
```