CS584 – Machine Learning

Topic: Decision Trees

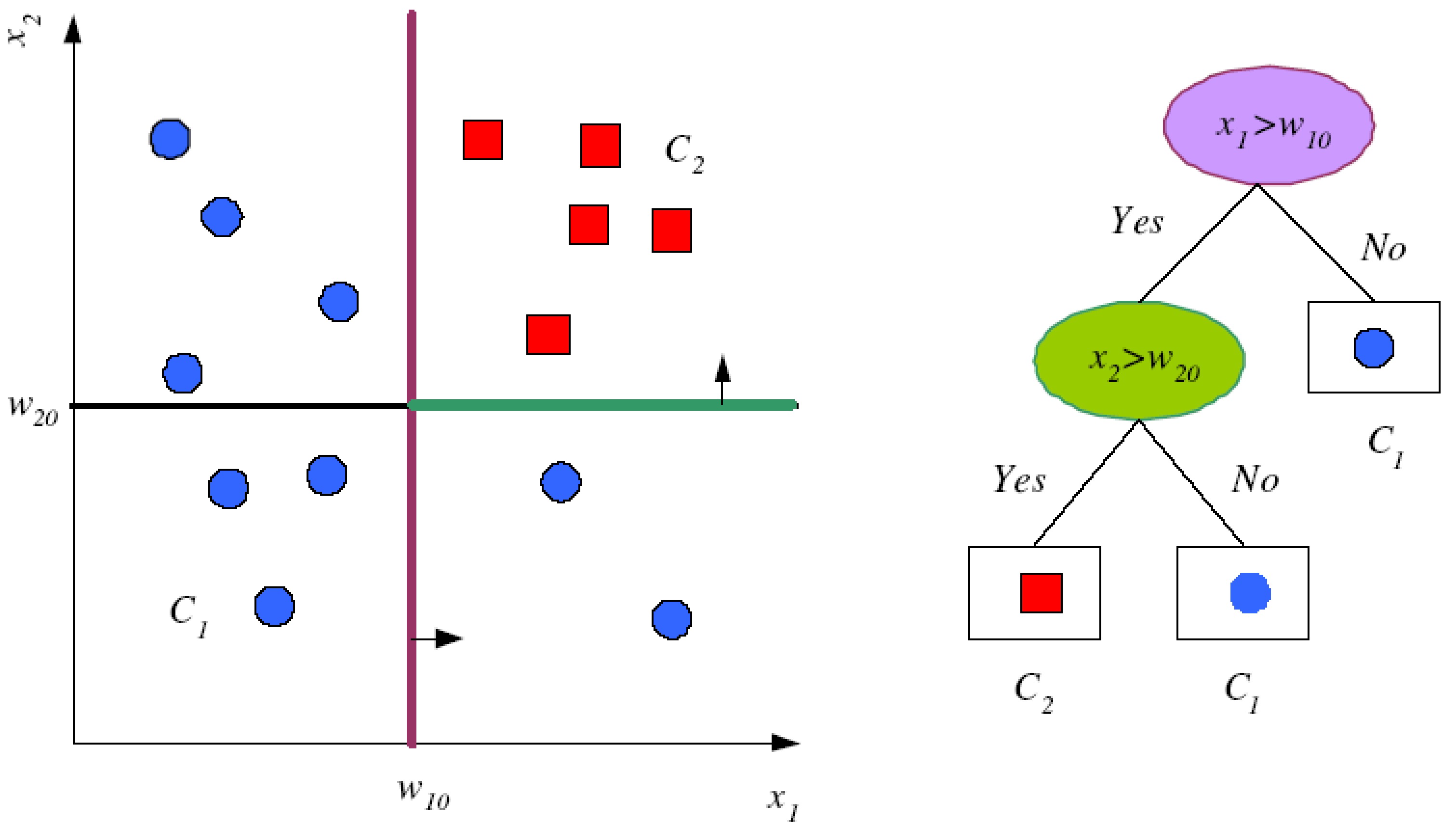
Mustafa Bilgic

🔗 http://www.cs.iit.edu/~mbilgic

https://twitter.com/bilgicm

CS584 – Machine Learning – Illinois Institute of Technology – Please do not distribute.

Tree Uses Nodes and Leaves



Credit: Ethem Alpaydin. Introduction to Machine Learning. 3rd Edition. http://www.cmpe.boun.edu.tr/~ethem/i2ml3e

Divide and Conquer

- Internal decision nodes
  - Univariate: Uses a single attribute, x_i
    - Numeric x_i : Binary split : $x_i > w_m$
    - Discrete x_i : n-way split for n possible values
  - Multivariate: Uses all attributes, $\boldsymbol{x}$
- Leaves
  - Classification: Class labels, or proportions
  - Regression: Numeric; r average, or local fit
- Learning is greedy; find the best split recursively (Breiman et al, 1984; Quinlan, 1986, 1993)

Credit: Ethem Alpaydin. Introduction to Machine Learning. 3rd Edition. http://www.cmpe.boun.edu.tr/~ethem/i2ml3e

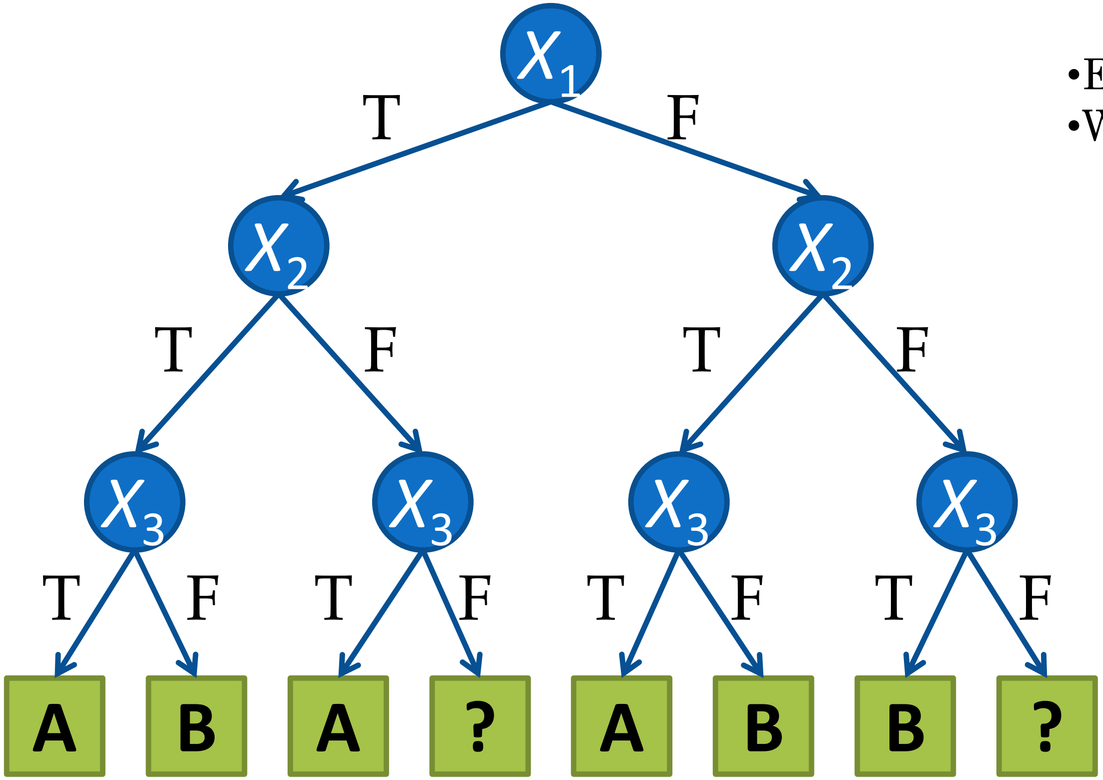
How would you learn a DT?

- That is, how would you choose the nodes and leaves?
  - Which node(s) would you split on?
  - When would you stop splitting?
- Here is a DT learning algorithm
  - The node at the i^{th} level is the i^{th} feature
  - The leaf is the last feature
  - What's the empirical error (error on training data)?
  - Can you use it for prediction?
  - Is it interpretable?
  - Does this provide any compression?
  - Can you do any better?

Let's Apply it: Data1

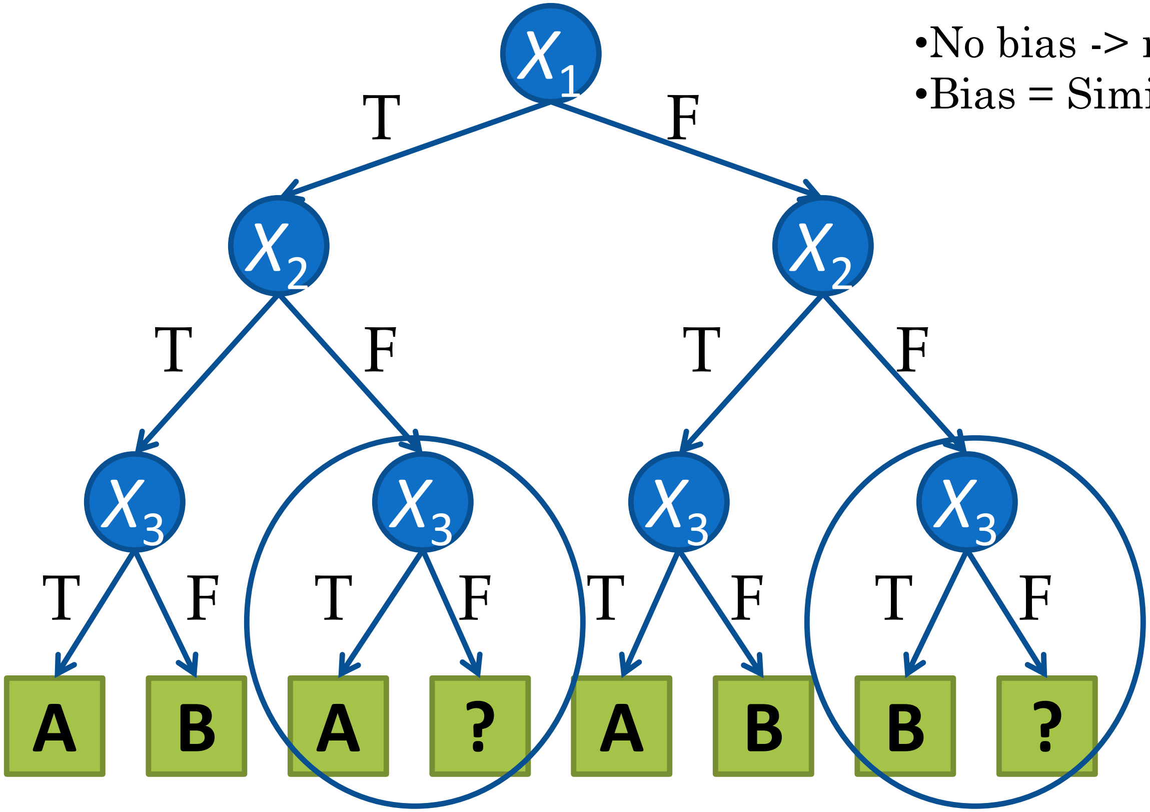
X_1	X_2	X_3	Y
T	T	T	A
T	T	F	B
T	F	T	A
F	T	T	A
F	T	F	B
F	F	T	B

Let's Apply it: Data1- Tree1



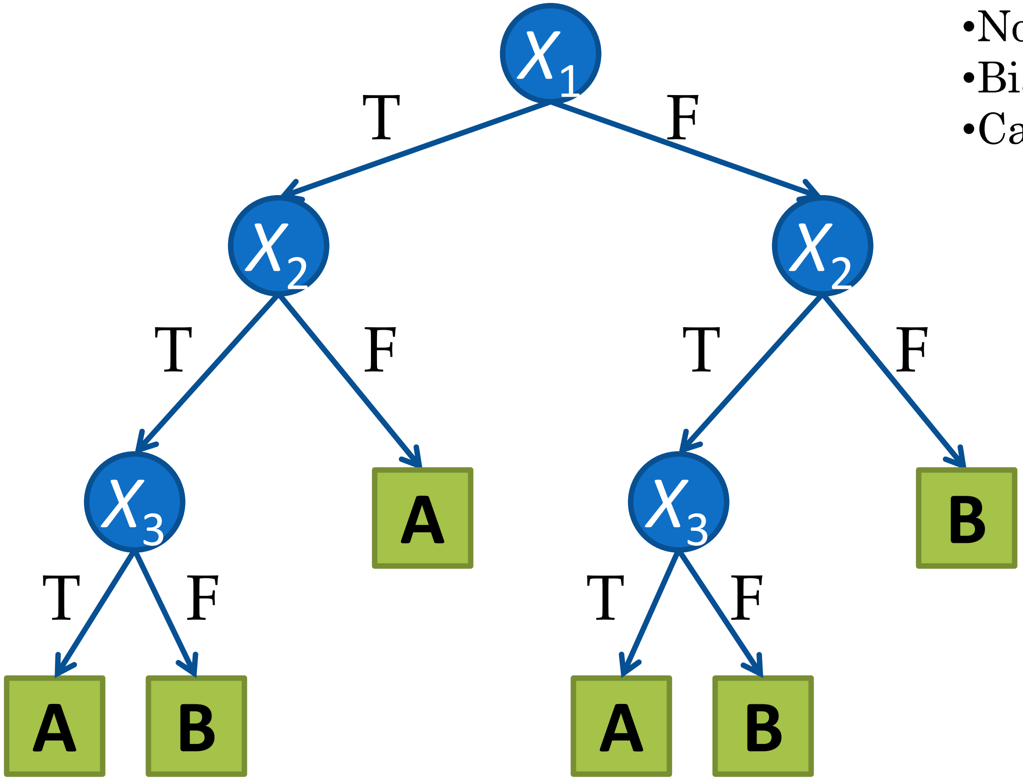
- Empirical error?
- What to do with unknown labels?
  - Reject?
    - Prediction capability?
  - Introduce bias
    - What kind of bias?

LET'S APPLY IT: DATA1- TREE1



- No bias -> no prediction
- Bias = Similar instances -> similar labels

LET'S APPLY IT: DATA1- TREE2



- No bias -> no prediction
- Bias = Similar instances -> similar labels
- Can we use this for prediction?
  - Yes!

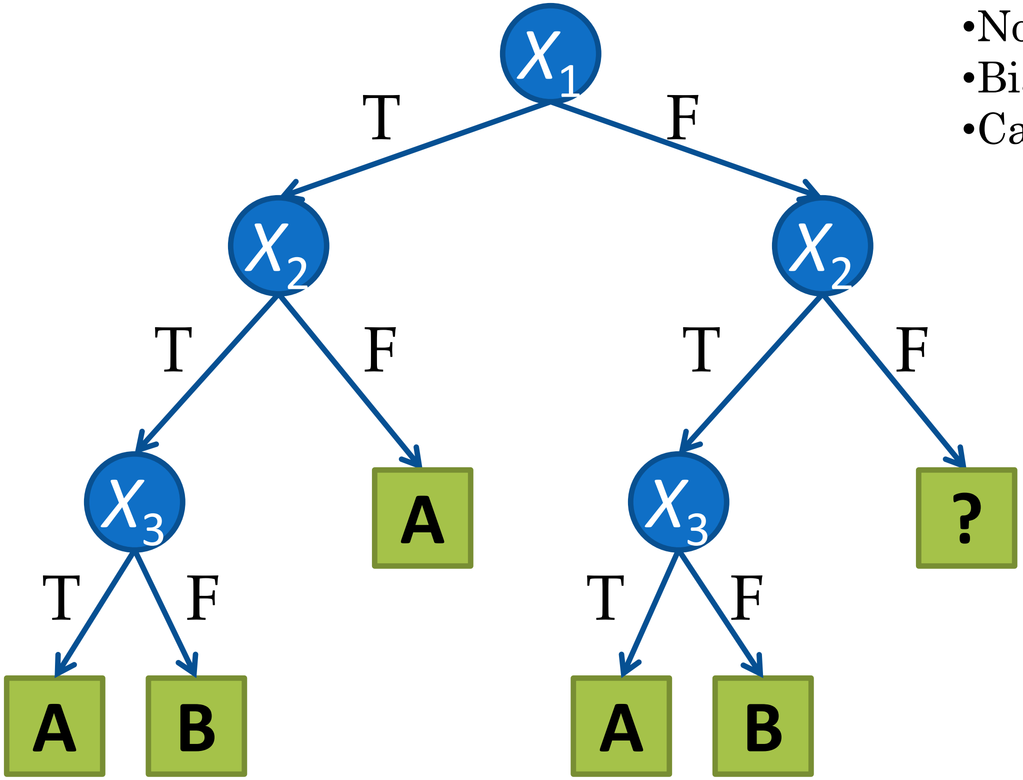
What if we had no case for X1=F, X2=F? How would you classify? A or B?

Let's Apply it: Data2

X_1	X_2	X_3	Y
T	T	T	A
T	T	F	B
T	F	T	A
F	T	T	A
F	T	F	B

Same as Data1, except the last row is deleted

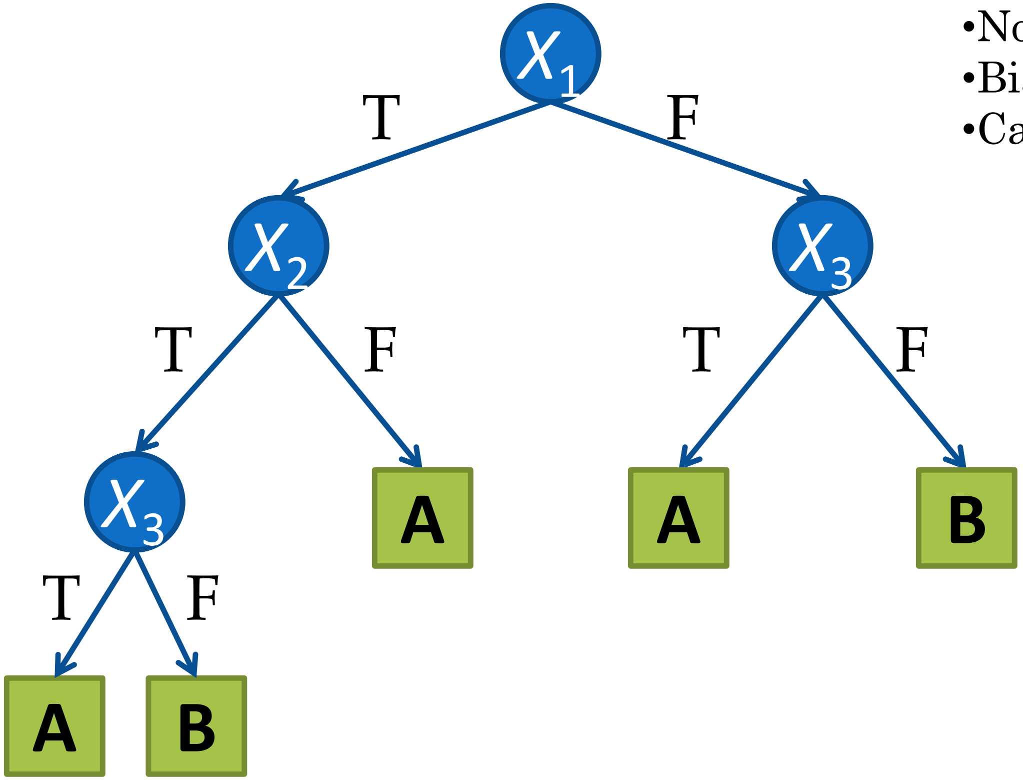
Let's Apply it: Data2- Tree1



- No bias -> no prediction
- Bias = Similar instances -> similar labels
- Can we use this for prediction?
  - Yes!

What if we had no case for X1=F, X2=F? How would you classify? A or B?

LET'S APPLY IT: DATA2- TREE2



- No bias -> no prediction
- Bias = Similar instances -> similar labels
- Can we use this for prediction?
  - Yes!

What if we had no case for X1=F, X2=F? How would you classify? A or B?

Intermediate Conclusions

- Need bias for prediction
- Given feature order is not necessarily the “best” order

WHAT KIND OF DT?

- We want the smallest tree? Why?
  - Mr. Occam says so
- “Entities are not be multiplied beyond necessity”
  - Father William of Ockham, Encyclopedia Britannica
- "We are to admit no more causes of natural things than such as are both true and sufficient to explain their appearances. Therefore, to the same natural effects we must, so far as possible, assign the same causes.“
  - Isaac Newton

Bad News

- Finding the smallest tree is NP-complete
- What do we do?
  - Be greedy!
  - Start with the “best” feature at the top and then the next “best” and then the next “best”
  - Is this guaranteed to be optimal?
    - Of course not.
- How do we measure how “good” a feature is?

PURITY

- A node is pure if it contains instances that belong to the same class
- Some *impurity* measures
  - First, let *p* represent the proportion of instances that belong to the positive class

$$\text{Entropy} = -p \log_2 p - (1-p)\log_2(1-p)$$

$$\text{Gini Index} = p(1-p)$$

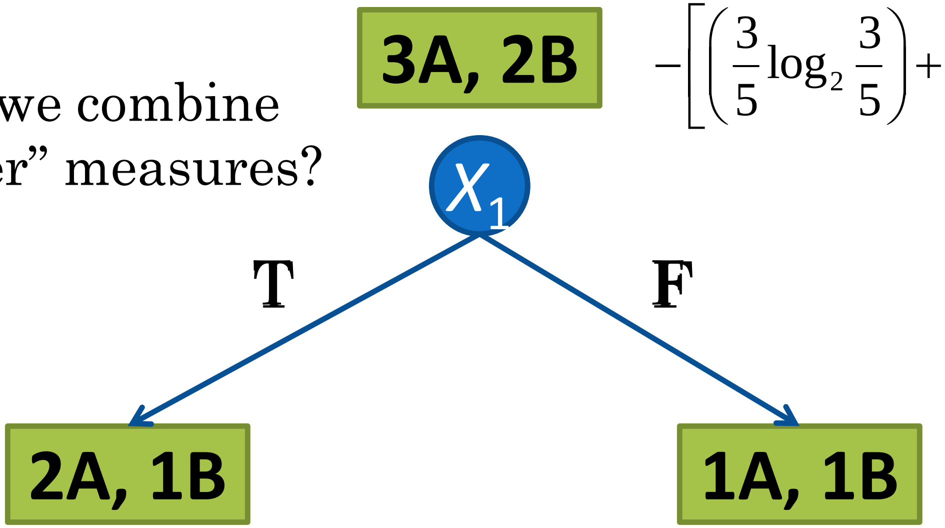
Locally Optimal Feature

- A feature, X_i, is locally optimal if the ***impurity*** is ***smallest after*** we split using X_i
- Example from Data2
  - Before split: 3A, 2B
  - Entropy =

$$-\left[\left(\frac{3}{5}\log_2\frac{3}{5}\right)+\left(\frac{2}{5}\log_2\frac{2}{5}\right)\right]=0.971$$

SPLIT ON X_1

Q1: How do we combine the two "after" measures?



3A, 2B

X_1

T

F

2A, 1B

1A, 1B

Before

$$-\left[\left(\frac{3}{5}\log_2\frac{3}{5}\right)+\left(\frac{2}{5}\log_2\frac{2}{5}\right)\right]=0.971$$

Q2: How do compare "before" and "after"?

$$-\left[\left(\frac{2}{3}\log_2\frac{2}{3}\right)+\left(\frac{1}{3}\log_2\frac{1}{3}\right)\right]=0.918$$

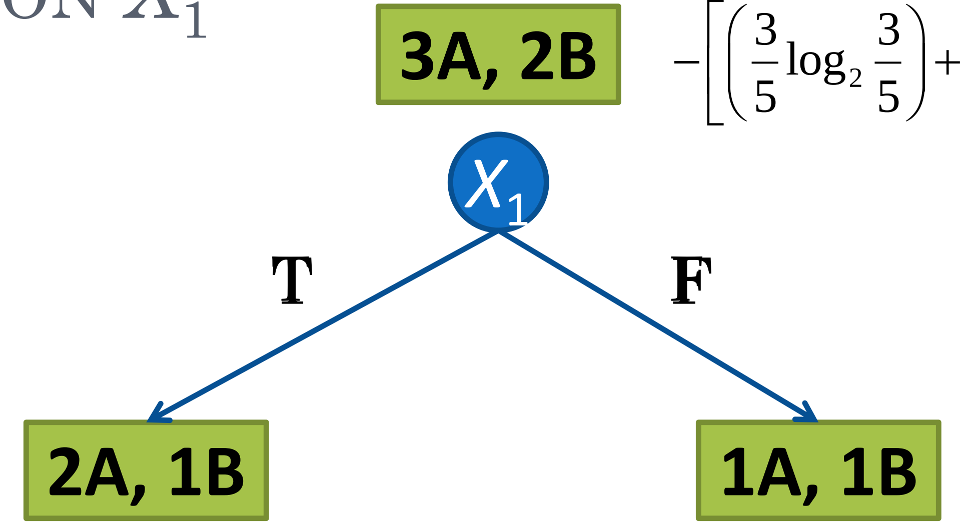
$$-\left[\left(\frac{1}{2}\log_2\frac{1}{2}\right)+\left(\frac{1}{2}\log_2\frac{1}{2}\right)\right]=1$$

After

Split on X_1

Before

$$-\left[\left(\frac{3}{5}\log_2\frac{3}{5}\right)+\left(\frac{2}{5}\log_2\frac{2}{5}\right)\right]=0.971$$



3A, 2B

X_1

T

F

2A, 1B

1A, 1B

$$-\left[\left(\frac{2}{3}\log_2\frac{2}{3}\right)+\left(\frac{1}{3}\log_2\frac{1}{3}\right)\right]=0.918$$

$$-\left[\left(\frac{1}{2}\log_2\frac{1}{2}\right)+\left(\frac{1}{2}\log_2\frac{1}{2}\right)\right]=1$$

After

$$\begin{aligned}
\text{After} &= \text{weighted average}\\
&= prob(X_1=T)\times Entropy(LeftTree)+prob(X_1=F)\times Entropy(RightTree)\\
&= \frac{3}{5}\times 0.918+\frac{2}{5}\times 1\\
&= 0.951
\end{aligned}$$

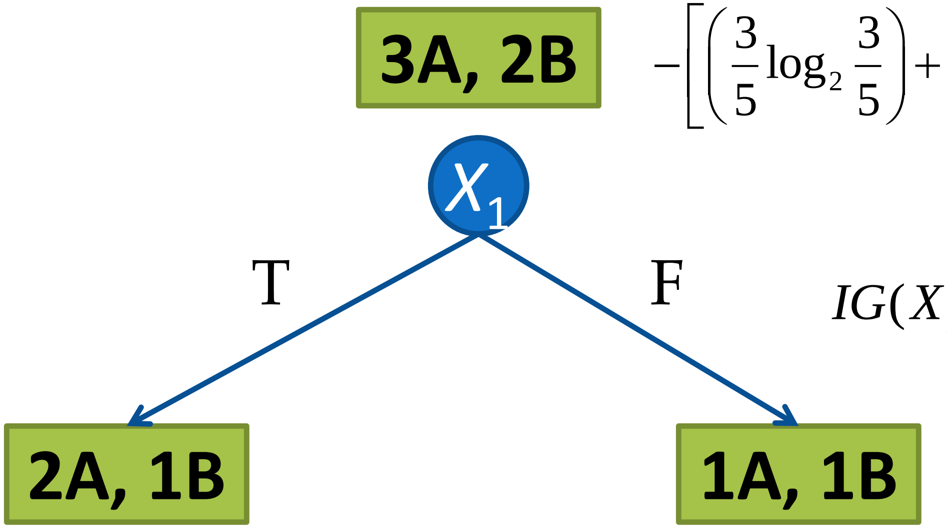
INFORMATION GAIN

$$InformationGain(X_i) = \text{Entropy before - Expected entropy after}$$

SPLIT ON X_1

Before

$$-\left[\left(\frac{3}{5}\log_2\frac{3}{5}\right)+\left(\frac{2}{5}\log_2\frac{2}{5}\right)\right]=0.971$$



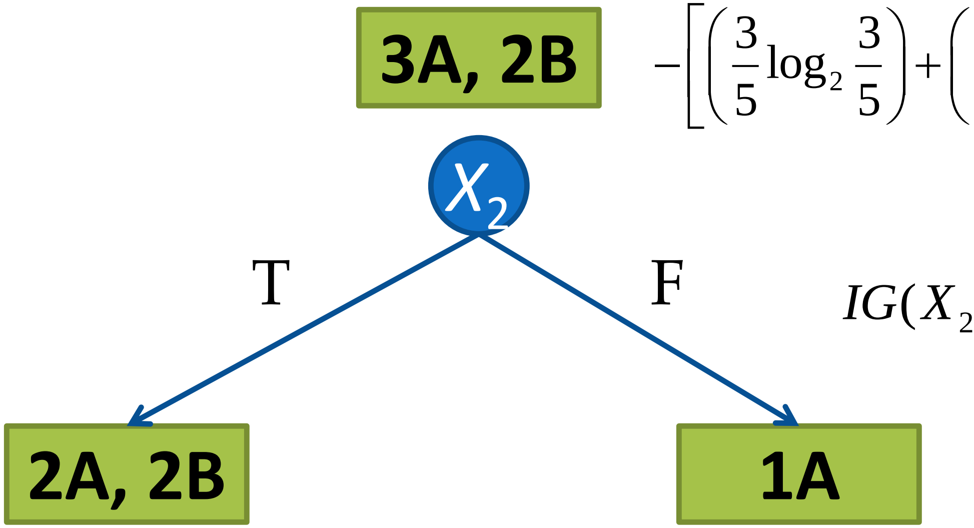
$$IG(X_1)=0.971-0.951=0.02$$

$$-\left[\left(\frac{2}{3}\log_2\frac{2}{3}\right)+\left(\frac{1}{3}\log_2\frac{1}{3}\right)\right]=0.918$$

$$-\left[\left(\frac{1}{2}\log_2\frac{1}{2}\right)+\left(\frac{1}{2}\log_2\frac{1}{2}\right)\right]=1$$

$$\begin{aligned}
\text{After} &= \text{weighted average}\\
&= prob(X_1=T)\times Entropy(LeftTree)+prob(X_1=F)\times Entropy(RightTree)\\
&= \frac{3}{5}\times 0.918+\frac{2}{5}\times 1\\
&= 0.951
\end{aligned}$$

Split on X_2



Before ↓

$$-\left[\left(\frac{3}{5}\log_2\frac{3}{5}\right)+\left(\frac{2}{5}\log_2\frac{2}{5}\right)\right]=0.971$$

$$IG(X_2)=0.971-0.8=0.171$$

$$-\left[\left(\frac{2}{4}\log_2\frac{2}{4}\right)+\left(\frac{2}{4}\log_2\frac{2}{4}\right)\right]=1$$

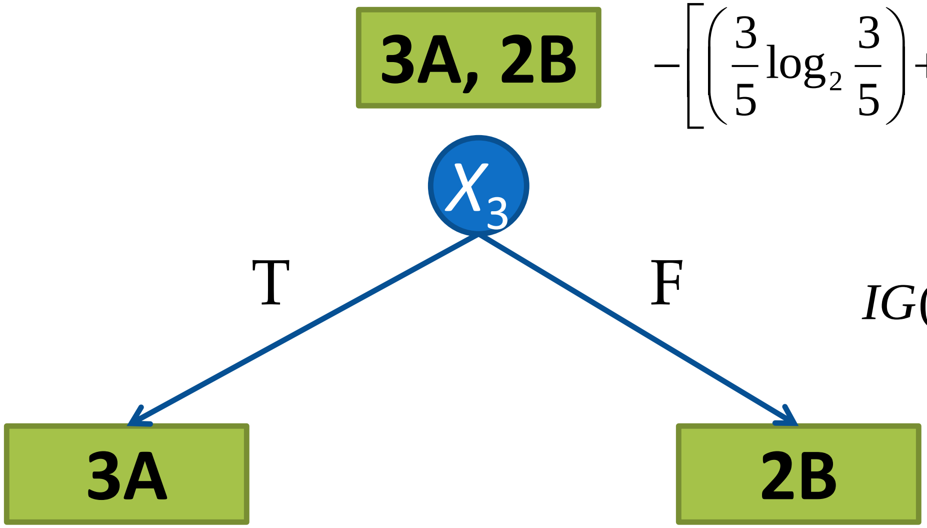
$$-\left[\left(\frac{1}{1}\log_2\frac{1}{1}\right)+\left(\frac{0}{1}\log_2\frac{0}{1}\right)\right]=0$$

$$\begin{aligned}\text{After} &= \text{weighted average}\\ &= prob(X_2=T)\times Entropy(LeftTree)+prob(X_2=F)\times Entropy(RightTree)\\ &= \frac{4}{5}\times 1+\frac{1}{5}\times 0\\ &= 0.8\end{aligned}$$

SPLIT ON X_3

Before

$$-\left[\left(\frac{3}{5}\log_2\frac{3}{5}\right)+\left(\frac{2}{5}\log_2\frac{2}{5}\right)\right]=0.971$$



$$IG(X_3)=0.971-0=0.971$$

$$-\left[\left(\frac{3}{3}\log_2\frac{3}{3}\right)+\left(\frac{0}{3}\log_2\frac{0}{3}\right)\right]=0$$

$$-\left[\left(\frac{0}{2}\log_2\frac{0}{2}\right)+\left(\frac{2}{2}\log_2\frac{2}{2}\right)\right]=0$$

$$\begin{aligned}\text{After} &= \text{weighted average}\\ &= prob(X_3=T)\times Entropy(LeftTree)+prob(X_3=F)\times Entropy(RightTree)\\ &= \frac{3}{5}\times 0+\frac{2}{5}\times 0\\ &= 0\end{aligned}$$

Information Gain on Data2

$$IG(X_1) = 0.971 - 0.951 = 0.02$$

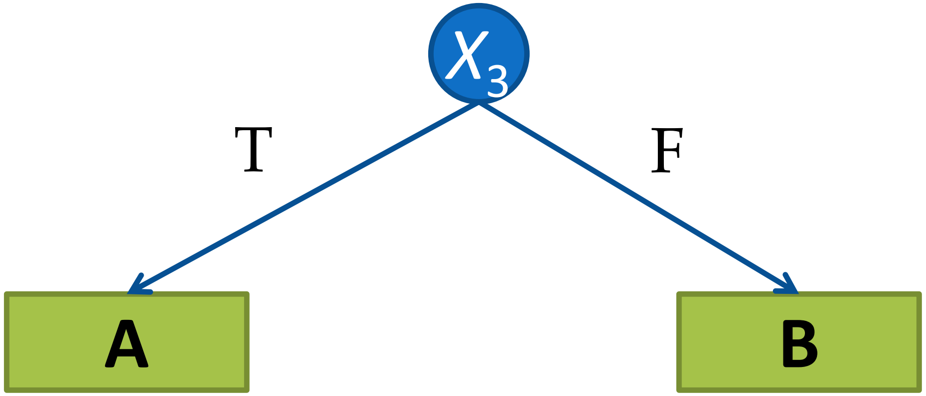
$$IG(X_2) = 0.971 - 0.8 = 0.171$$

$$IG(X_3) = 0.971 - 0 = 0.971$$

A DT Algorithm (ID3)

- Start with the empty tree
- At each iteration
  - Pick the locally optimal feature and split on it
  - Stop when all leaves are pure (or no more features are left to split)

Let's Apply it: Data2-Tree3



- Empirical error?
- Prediction power?
- Size?
- Is Occam happy now?

Attributes with many values

- If an attribute has many values, it will have high information gain probably by chance
- See an example
- Solution: instead of information gain, use gain ratio
    - Gain Ratio = Information Gain / Entropy of the Split

How about numeric attributes?

- How many possible split points?
- Let’s see an example

Another Example

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

Hypotheses represented by trees?

- Conjunctions?
- Disjunctions?
- Negations?
- What is the inductive bias of the ID3 algorithm?

OVERFITTING

- Given a hypothesis h
  - $error_{train}(h)$: Error of h on the train set
  - $error_{distribution}(h)$: Error of h on the entire distribution of the data
- $h \in H$ **overfits** the train set if there is an $h' \in H$ such that:
  - $error_{train}(h) < error_{train}(h')$ and
  - $error_{distribution}(h) > error_{distribution}(h')$

WHEN TO STOP GROWING THE TREE?

- Technically
  - Stop when the leaf is pure
  - Otherwise, stop when no more attributes are left to test
- If there are errors in the training data
  - The tree can end up being larger than it needs to be
- Remember that we want a small tree; larger trees tend to overfit the training data
- Two solutions:
  - Stop early based on a criteria
  - Post-prune the tree

EARLY STOPPING (PRE-PRUNING)

- Stop growing a branch based on some fixed criteria
- Example criteria:
  - Stop when the number of instances in a leaf gets below a threshold
  - Stop when the information gain of the remaining attributes gets below a threshold
  - Stop when the entropy at a leaf is below a threshold
  - Stop when the depth of the tree reaches a threshold
  - (and so on)

POST-PRUNING USING VALIDATION DATA

- Keep a separate data for validation
- First, grow the full tree using the training data
- Then, prune a node (and those below it) as long as pruning improves performance on the validation data

Scikit-Learn – Decision Trees

- http://scikit-learn.org/stable/modules/tree.html
- http://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html

Other Popular Topics

- Random forests
  - Fit several decision trees on subsamples of the training data and use averaging for predictions
  - http://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html
- Decision trees for regression
  - We've seen decision trees for classification, but trees can be used for classification
  - http://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeRegressor.html
- Gradient-boosted regression trees
  - Highly accurate
  - http://scikit-learn.org/stable/modules/ensemble.html#gradient-boosting
- Note: The data mining class (CS422) goes into details of these topics and more