# Decision Tree

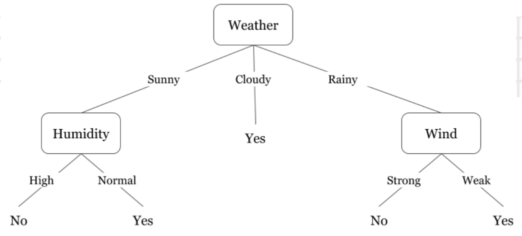
INSTRUCTOR: HONGJIE CHEN

MAY 23RD 2022



Acknowledgement: many slides are derived from those of Tom Mitchell, Pascal Poupart, Pieter Abbeel, Erir Eaton, Carlos Guestrin, William Cohen, and Andrew Moore

# Decision Tree (ID3, C4.5, etc.)




- Top-down induction of decision trees
  - Set $A =$ the “best” attribute to split current node
  - For each attribute value of $A$, create a branch
  - Divide current node into children node through branches
  - If children node are perfect or some criteria are met, stop, otherwise recursively repeat on children nodes

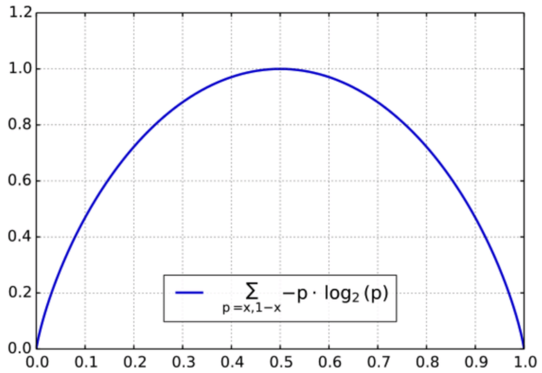
- Intuition: top-down **greedy** growth of decision tree using “best” attribute until all samples are perfectly classified.

- Question: How to pick “best” attribute?



VIRGINIA TECH

# Sample Entropy




| Outlook | Temperature | Humidity | Windy | PlayTennis |
|---|---|---|---|---|
| Sunny | Hot | High | False | No |
| Sunny | Hot | High | True | No |
| Overcast | Hot | High | False | Yes |
| Rainy | Mild | High | False | Yes |
| Rainy | Cool | Normal | False | Yes |
| Rainy | Cool | Normal | True | No |
| Overcast | Cool | Normal | True | Yes |
| Sunny | Mild | High | False | No |
| Sunny | Cool | Normal | False | Yes |
| Rainy | Mild | Normal | False | Yes |
| Sunny | Mild | Normal | True | Yes |
| Overcast | Mild | High | True | Yes |
| Overcast | Hot | Normal | False | Yes |
| Rainy | Mild | High | True | No |

- $S$ is a (sub)set of training samples
- $p_\oplus$ is the porportion of positive samples in $S$
- $p_\ominus$ is the porportion of negative samples in $S$
- Entropy measures the *impurity* of $S$

$$H(S) := -p_\oplus \log_2 p_\oplus - p_\ominus \log_2 p_\ominus$$

# Entropy as a Measure of Impurity

- $H(S) := -p_{\oplus}\log_2 p_{\oplus} - p_{\ominus}\log_2 p_{\ominus}$

- Impure case:

$$\begin{cases} p_{\oplus} = 0.5 \\ p_{\ominus} = 0.5 \end{cases} \Rightarrow H(S) = -\frac{1}{2}\log_2\frac{1}{2} - \frac{1}{2}\log_2\frac{1}{2} = 1$$

- Pure case: $\begin{cases} p_{\oplus} = 1 \\ p_{\ominus} = 0 \end{cases} \Rightarrow H(S) = -\log_2 1 = 0$



VIRGINIA TECH.

# Entropy

- Entropy $H(X)$ of a random variable $X$ is defined as:

$$H(X) = -\sum_i P(X=i)\log_2 P(X=i)$$

**Multi-class**

**$Y$ in this page is a R.V.**

- Specific conditional entropy $H(X \mid Y=v)$

$$H(X \mid Y=v) = -\sum_i P(X=i \mid Y=v)\log_2 P(X=i \mid Y=v)$$

- Conditional entropy $H(X \mid Y)$

$$H(X \mid Y) = \sum_{v \in \text{values(Y)}} P(Y=v)\, H(X \mid Y=v)$$

- Mutual information (a.k.a. information gain) of $X$ and $Y$

$$I(X,Y) = H(X) - H(X \mid Y) = H(Y) - H(Y \mid X)$$

# Information Gain

- Mutual information (a.k.a. information gain) of $X$ and $Y$
  - $I(X, Y) = H(X) - H(X|Y) = H(Y) - H(Y|X)$

- Information gain is the expected reduction in entropy of target variable $Y$ for data sample $S$, in observance of variable $A$
  - $\text{Gain}(S, A) = I_S(A, Y) = H_S(Y) - H_S(Y|A)$

- Question: How to pick the best attribute?
- Answer: One that obtains the highest information gain, i.e., reducing entropy the most.

# *Highest Information Gain*

- Gain$(S, A) = I_S(A, Y) = H_S(Y) - H_S(Y|A)$ **Highest**
- $H_S(Y)$ is fixed
- $-H_S(Y|A)$ **Highest**
- $H_S(Y|A)$ **Lowest**

- Low impurity, i.e., High purity
  - Split data that results the most skewed distribution.

# More than Entropy

- Given a (subset of) dataset $D$ containing $C$ classes
- Entropy

  Proportion for class $c$: $\hat{\pi}_c = \frac{1}{|D|} \sum_{l \in D} \mathbb{I}(y^l = c)$

  Entropy error: $-\sum_{c=1}^{C} \hat{\pi}_c \log \hat{\pi}_c$
- Misclassification rate

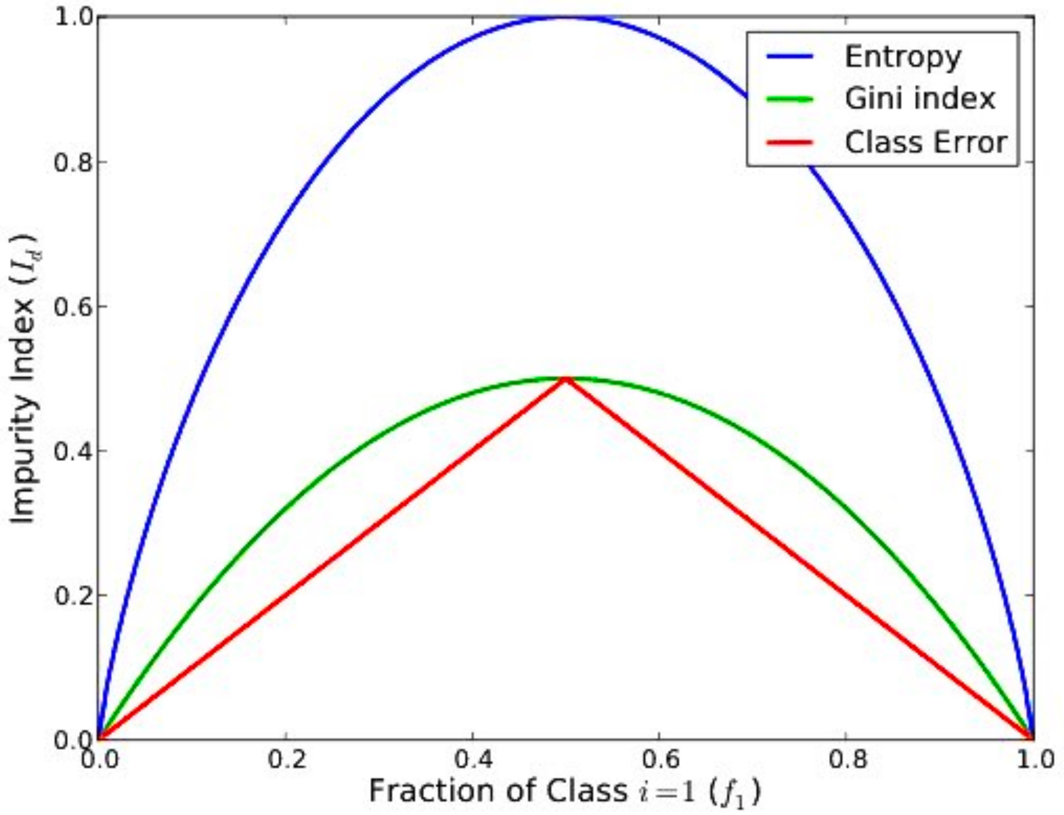
  Assign prediction by the major class $\hat{y} = \text{argmax}_c \hat{\pi}_c$

  Error rate: $\frac{1}{|D|} \sum_{l \in D} \mathbb{I}(y^l \neq \hat{y}) = 1 - \hat{\pi}_{\hat{y}}$
- Gini index

  Gini error: $\sum_{c=1}^{C} \hat{\pi}_c (1 - \hat{\pi}_c) = 1 - \sum_{c} \hat{\pi}_c{}^2$

# *Comparing Metrics*

- Assume boolean labels (two classes)

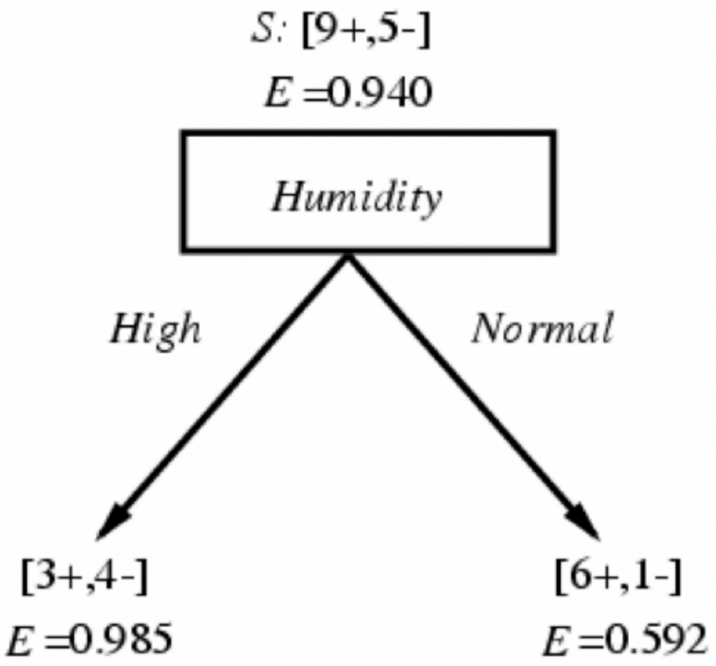


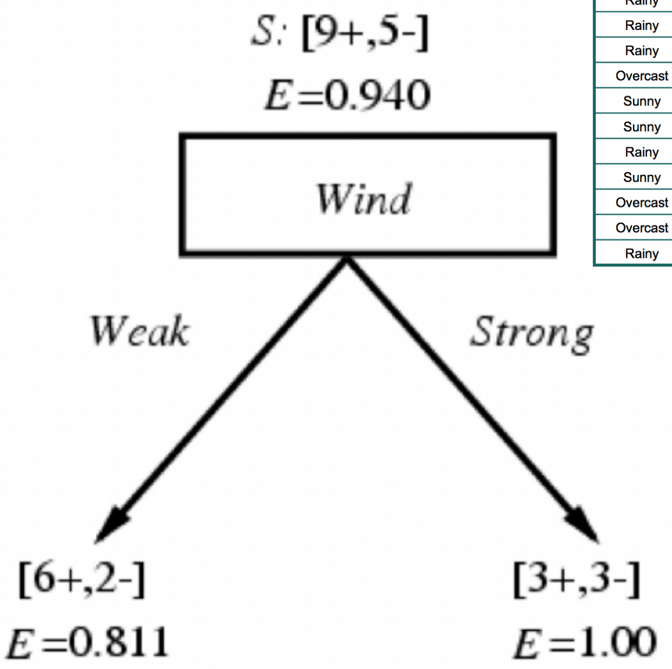
# Example Data

| Outlook | Temperature | Humidity | Windy | PlayTennis |
|---|---|---|---|---|
| Sunny | Hot | High | False | No |
| Sunny | Hot | High | True | No |
| Overcast | Hot | High | False | Yes |
| Rainy | Mild | High | False | Yes |
| Rainy | Cool | Normal | False | Yes |
| Rainy | Cool | Normal | True | No |
| Overcast | Cool | Normal | True | Yes |
| Sunny | Mild | High | False | No |
| Sunny | Cool | Normal | False | Yes |
| Rainy | Mild | Normal | False | Yes |
| Sunny | Mild | Normal | True | Yes |
| Overcast | Mild | High | True | Yes |
| Overcast | Hot | Normal | False | Yes |
| Rainy | Mild | High | True | No |

# Select the Best Attribute

| Outlook | Temperature | Humidity | Windy | PlayTennis |
|---|---|---|---|---|
| Sunny | Hot | High | False | No |
| Sunny | Hot | High | True | No |
| Overcast | Hot | High | False | Yes |
| Rainy | Mild | High | False | Yes |
| Rainy | Cool | Normal | False | Yes |
| Rainy | Cool | Normal | True | No |
| Overcast | Cool | Normal | True | Yes |
| Sunny | Mild | High | False | No |
| Sunny | Cool | Normal | False | Yes |
| Rainy | Mild | Normal | False | Yes |
| Sunny | Mild | Normal | True | Yes |
| Overcast | Mild | High | True | Yes |
| Overcast | Hot | Normal | False | Yes |
| Rainy | Mild | High | True | No |




**More Skewed**

*Gain (S, Humidity)*

$= .940 - (7/14).985 - (7/14).592$
$= .151$

*Gain (S, Wind)*

$= .940 - (8/14).811 - (6/14)1.0$
$= .048$

**More Gain**

# *What if Features are Continuous*

- Test the splitting of each unique feature value

- HW1

# *Open-ended Questions*

- Is there more than one decision tree that perfectly infers the data labels?
- If so, which one do you choose and why?
  - Deep v.s. Shallow
  - More branches v.s. Less branches
- Do we always want a perfect decision tree? Why?
  - Think about what a perfect decision tree implies

# *Thoughts*

- There are more than one possible perfect decision trees.
- We typically favor a shallow and simple decision tree rather than a more complicated one, if the latter achieves the same performance or even performs slightly better.
  - Generalization
  - Time and space complexity
- A perfect decision tree can (sometimes) indicate overfitting.
  - Imagine memorizing all training samples but failing to infer a new testing sample that is never met before.

# *Decesion Tree Learning*

- Recall the set of function hypotheses: $H : \{h \,|\, h : X \to Y\}$
- Now we want to obtain a good $h$, a good decision tree.

- Which tree in the hypotheses should we obtain?

# *Inductive Inference*

- Data-driven
- From specific observations to general rules

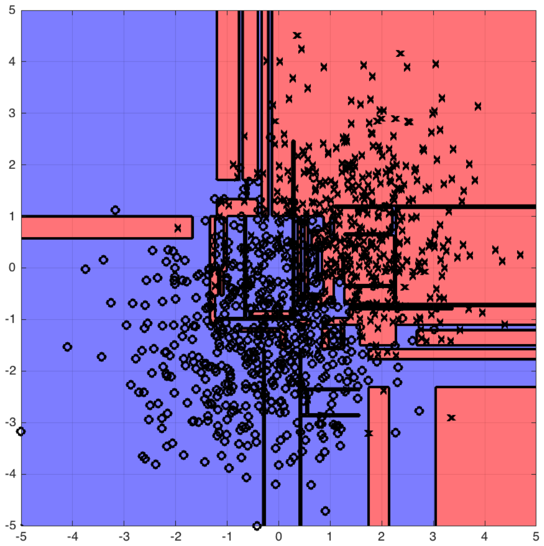
- Generalization cannot go beyond the training data
  - Which decision tree $h$ to pick depends on the training data

# *Occam's Razor*

- Choose the simplest hypothesis that fits the data
- Stop the top-down greedy growth of decision tree at smallest acceptable tree.
  - How to know we could stop at a node?
  - When the samples in the node are …
- What if we don't stop?

# Overfitting in Decision Tree Learning

- Another example on real valued data



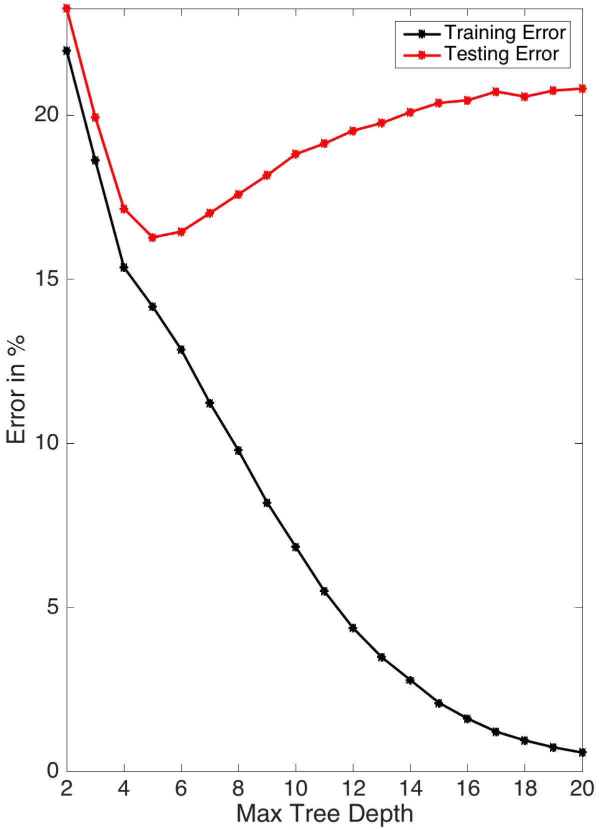




Figure credit: [link]

# *Strategies to avoid overfitting*

- Do not pursue perfect decision tree with the training data
  - Stop growing the tree when information gain is trivial
  - Grow full tree, then post-prune
  - …

- How to select the "best" tree?
  - Measure performance on training dataset
  - Measuer performance on standalone validation dataset

- **Validation data**: Data that are not used to train the model, but are evaluated to see how well the model is trained, and usually serve the purpose hyperparameters tuning.

# *Reduce-error Pruning with Validation Data*

- Split data into training and validation sets
- Create a perfect decision tree on the training set
- Repeat until further pruning is harmful:
  - For each possible node, evaluate the impact on the validation set after pruning it. By pruning, it means removing the subtree at that node, make it a leaf and assign the most common class at that node
  - Greedily remove the one that results most accuracy improvement on the validation set.

**Q: What if two nodes results the same best accuracy improvement?**

***Occam's Razor***

# *Random Forests*

- Boostrap Aggregation (Bagging)
  - Train multiple decision trees (thus, forests)
    - Randomly make a subset of datasets
    - Train a decision tree on the subset

For a new sample
  - Gain a prediction from each decision tree
  - Use average or majority vote as the final prediction

  - Also randomly subsample features
  - Reduce variance without changing bias*