# CS412 Office Hours

April 8, 2019

# Covered Materials

1. Gini index calculation
2. Rainforest details
3. Bayesian network reasoning

# Gini Index

- Gini index: Used in CART, and also in IBM IntelligentMiner
- If a data set $D$ contains examples from $n$ classes, gini index, $gini(D)$ is defined as
  - $gini(D) = 1 - \sum_{j=1}^{n} p_j^2$
    - $p_j$ is the relative frequency of class $j$ in $D$
- If a data set $D$ is split on $A$ into two subsets $D_1$ and $D_2$, the $gini$ index $gini(D)$ is defined as
  - $gini_A(D) = \frac{|D_1|}{|D|} gini(D_1) + \frac{|D_2|}{|D|} gini(D_2)$
- Reduction in Impurity:
  - $\Delta gini(A) = gini(D) - gini_A(D)$
- The attribute which provides the smallest $gini_{split}(D)$ (or the largest reduction in impurity) is chosen to split the node (*need to enumerate all the possible splitting points for each attribute*)

# Computation of Gini Index: Binary Split

- Example: D has 9 tuples in buys_computer = "yes" and 5 in "no"

$$gini(D) = 1 - \left(\frac{9}{14}\right)^2 - \left(\frac{5}{14}\right)^2 = 0.459$$

- Suppose the attribute income partitions D into 10 in $D_1$: {low, medium} and 4 in $D_2$
  - $gini_{income \in \{low, medium\}}(D) = \frac{10}{14} gini(D_1) + \frac{4}{14} gini(D_2)$

$$= \frac{10}{14}\left(1 - \left(\frac{7}{10}\right)^2 - \left(\frac{3}{10}\right)^2\right) + \frac{4}{14}\left(1 - \left(\frac{2}{4}\right)^2 - \left(\frac{2}{4}\right)^2\right) = 0.443$$

$$= Gini_{income \in \{high\}}(D)$$

  - $Gini_{\{low,\ high\}}$ is 0.458; $Gini_{\{medium,\ high\}}$ is 0.450
  - Thus, split on $income \in$ {low, medium} (i.e., also $\{high\}$) has the lowest Gini index

| age | income | student | credit_rating | buys_computer |
|---|---|---|---|---|
| <=30 | high | no | fair | no |
| <=30 | high | no | excellent | no |
| 31…40 | high | no | fair | yes |
| >40 | medium | no | fair | yes |
| >40 | low | yes | fair | yes |
| >40 | low | yes | excellent | no |
| 31…40 | low | yes | excellent | yes |
| <=30 | medium | no | fair | no |
| <=30 | low | yes | fair | yes |
| >40 | medium | yes | fair | yes |
| <=30 | medium | yes | excellent | yes |
| 31…40 | medium | no | excellent | yes |
| 31…40 | high | yes | fair | yes |
| >40 | medium | no | excellent | no |

# Computation of Gini Index: Full Split

In the quiz questions, full split is used instead of binary split. If an attribute has K categories, we split the dataset into K partitions. In practice, binary split is used more often as it generally performs better.

Following the previous example, if we choose to split on income, we will have 3 partitions: D1 Low (4), D2 Medium (6) and D3 High (4).

| age | income | student | credit_rating | buys_computer |
|---|---|---|---|---|
| <=30 | high | no | fair | no |
| <=30 | high | no | excellent | no |
| 31…40 | high | no | fair | yes |
| >40 | medium | no | fair | yes |
| >40 | low | yes | fair | yes |
| >40 | low | yes | excellent | no |
| 31…40 | low | yes | excellent | yes |
| <=30 | medium | no | fair | no |
| <=30 | low | yes | fair | yes |
| >40 | medium | yes | fair | yes |
| <=30 | medium | yes | excellent | yes |
| 31…40 | medium | no | excellent | yes |
| 31…40 | high | yes | fair | yes |
| >40 | medium | no | excellent | no |

The original gini index:

$$gini(D) = 1 - (\frac{9}{14})^2 - (\frac{5}{14})^2 = 0.459$$

Gini index after splitting:

$$\begin{aligned} gini_{income}(D) &= \frac{4}{14} gini_{income=low} + \frac{6}{14} gini_{income=med} + \frac{4}{14} gini_{income=high} \\ &= \frac{4}{14}(1 - (\frac{3}{4})^2 - (\frac{1}{4})^2) + \frac{6}{14}(1 - (\frac{4}{6})^2 - (\frac{2}{6})^2) + \frac{4}{14}(1 - (\frac{2}{4})^2 - (\frac{2}{4})^2) \end{aligned}$$

# Comments on the Programming Assignment

1. You can use either binary split or full split, both will pass the test. Full split is simplier to implement.
2. If your decision tree is overfitting, try to prune your tree using a validation set
   a. Some simple ways to prune the tree include stop splitting when the number of datapoints at the node is smaller than k or stop growing the tree when the number of levels exceeds L.
3. If your decision tree is running too slow
   a. Are you copying the dataset around?
   b. Are you iterating through the entire dataset when you only need a partition?
   c. Are you using libraries that are slow? (For example pandas)
   d. It’s ok if your code is not optimal. We will grade for output, not efficiency.

# Using Decision Trees for Large Datasets: RainForest

- Our goal is to reduce the number of times we read/write from the disk where the database resides.
- RainForest decouples the “counting” process and the split criteria computation process.
- Database access only happens during the construction of the AVC-groups
- After we get the AVC-groups, we do not need to refer to the original data points anymore
- Any split criteria that can be computed with the **AVC-groups alone** can be used
    a. For example, the MDL (minimum description length) cannot be directly applied as it can only be computed once the tree is built.

# RainForest: A Scalable Classification Framework

- The criteria that determine the quality of the tree can be computed separately
  - Builds an AVC-list: **AVC (Attribute, Value, Class_label)**
- **AVC-set** (of an attribute *X* )
  - Projection of training dataset onto the attribute *X* and class label where counts of individual class label are aggregated
- **AVC-group** (of a node *n* )
  - Set of AVC-sets of all predictor attributes at node *n*
- Then read the data again & do it similarly to generate the next level of the tree

| age | income | student | credit rating | comp |
|---|---|---|---|---|
| <=30 | high | no | fair | no |
| <=30 | high | no | excellent | no |
| 31…40 | high | no | fair | yes |
| >40 | medium | no | fair | yes |
| >40 | low | yes | fair | yes |
| >40 | low | yes | excellent | no |
| 31…40 | low | yes | excellent | yes |
| <=30 | medium | no | fair | no |
| <=30 | low | yes | fair | yes |
| >40 | medium | yes | fair | yes |
| <=30 | medium | yes | excellent | yes |
| 31…40 | medium | no | excellent | yes |
| 31…40 | high | yes | fair | yes |
| >40 | medium | no | excellent | no |

The Training Data

AVC-set on *Age*

| Age | Buy_Computer | |
|---|---|---|
| | yes | no |
| <=30 | 2 | 3 |
| 31..40 | 4 | 0 |
| >40 | 3 | 2 |

AVC-set on *Income*

| income | Buy_Computer | |
|---|---|---|
| | yes | no |
| high | 2 | 2 |
| medium | 4 | 2 |
| low | 3 | 1 |

AVC-set on *Student*

| student | Buy_Computer | |
|---|---|---|
| | yes | no |
| yes | 6 | 1 |
| no | 3 | 4 |

AVC-set on *Credit_Rating*

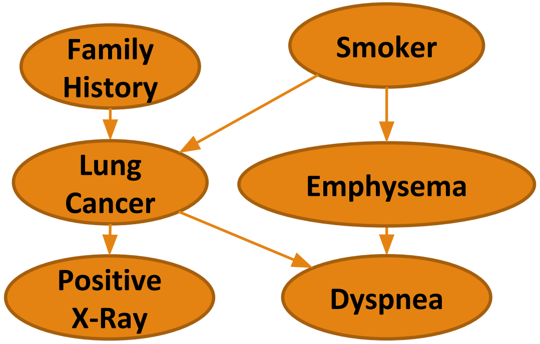
| Credit rating | Buy_Computer | |
|---|---|---|
| | yes | no |
| fair | 6 | 2 |
| excellent | 3 | 3 |

Its AVC Sets

# Bayesian Network

- **Bayesian network** (or **Bayesian belief network, probabilistic network**):
  - Allows *class conditional independencies* between *subsets* of variables
  - Represents the joint distribution compactly in a factorized way
- Two components:
  - A *directed acyclic graph* (called a structure)
    - The **nodes** represent random variables, obser
    - The **edges** represent direct dependency between variables
  - A set of *conditional probability tables* (CPTs)
    - CPTs are attached to the nodes

Nodes: random variables Links: dependency



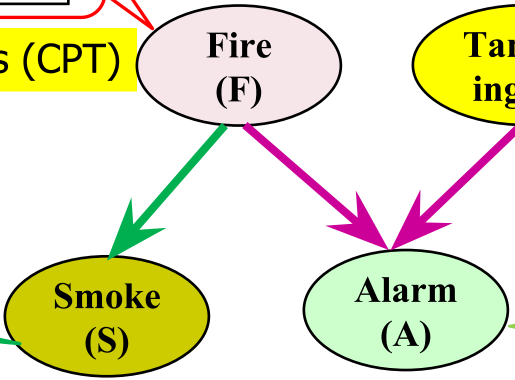

| | *FH, S* | *FH, ~S* | *~FH, S* | *~FH, ~S* |
|---|---|---|---|---|
| *LC* | 0.8 | 0.5 | 0.7 | 0.1 |
| *~LC* | 0.2 | 0.5 | 0.3 | 0.9 |

# Representing Joint Distribution with Bayesian Network

| Fire=1 | Fire=0 |
|---|---|
| 0.02 | 0.98 |

| Tampering=1 | Tampering=0 |
|---|---|
| 0.1 | 0.9 |

**C**onditional **P**robability **T**ables (CPT)




| | Smoke=1 | Smoke=0 |
|---|---|---|
| Fire=1 | 0.9 | 0.1 |
| Fire=0 | 0.01 | 0.99 |

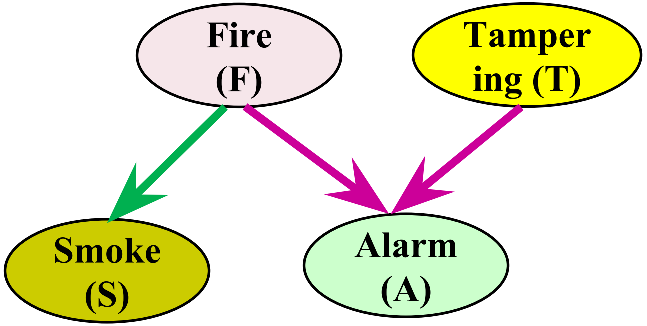
| | Alarm=1 | Alarm=0 |
|---|---|---|
| F=1,T=1 | 0.5 | 0.5 |
| F=1,T=0 | 0.99 | 0.01 |
| F=0,T=1 | 0.85 | 0.15 |
| F=0,T=0 | 0.0001 | 0.9999 |

From the Bayesian network we can read the factorization of the joint distribution: $p(F, S, A, T) = p(F) \cdot p(T) \cdot p(S|F) \cdot p(A|F, T)$

In general, we have the chain rule for Bayesian networks: $p(X) = \prod_k p(x_k|\text{Parents}(x_k))$

# Bayesian Network: An Example

| | Smoke=1 | Smoke=0 |
|---|---|---|
| Fire=1 | 0.9 | 0.1 |
| Fire=0 | 0.01 | 0.99 |

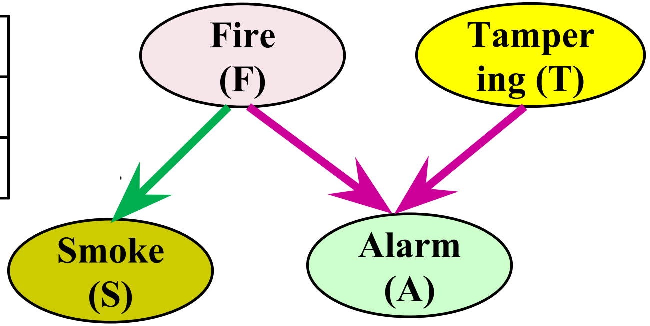



- Causal Reasoning:
  - The value of variable Fire influences variable Smoke
    - If F=1 → $p(S=1|F=1) = 0.9$; if F=0 → $p(S=1|F=0) = 0.01$

# Bayesian Network: An Example

| Fire=1 | Fire=0 |
| --- | --- |
| 0.02 | 0.98 |

| | Smoke=1 | Smoke=0 |
| --- | --- | --- |
| Fire=1 | 0.9 | 0.1 |
| Fire=0 | 0.01 | 0.99 |



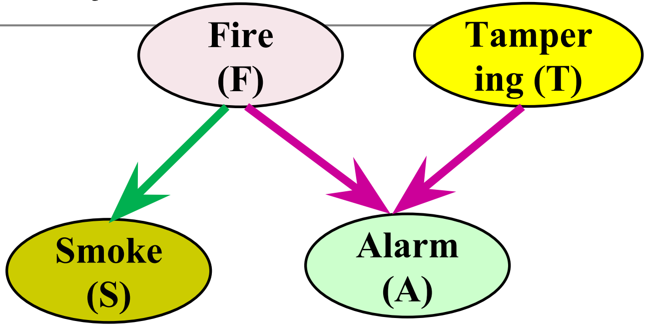

- Evidential Reasoning:
  - The value of variable Smoke also influences Fire
    - If S=1 → p(F=1|S=1) = $\frac{p(S=1|F=1)*p(F=1)}{p(S=1)}$ = 0.647; if S=0, p(F=1|S=0)=0.002

$$
\begin{aligned}
s(F=1|S=1) &= \frac{p(S=1,F=1)}{p(S=1)} \\
&= \frac{p(S=1|F=1)\times p(F=1)}{\sum_f p(S=1,F)} \\
&= \frac{p(S=1|F=1)\times p(F=1)}{p(S=1|F=1)p(F=1)+p(S=1|F=0)p(F=0)} \\
&= \frac{0.9\times 0.02}{0.9\times 0.02+0.01\times 0.98} \\
&= 0.647
\end{aligned}
$$

**P(A/B) = P(A,B)/P(B)**

**P(A/B) = P(B/A) * P(A)/$\sum_a$P(B,a)**

# Bayesian Network: An Example

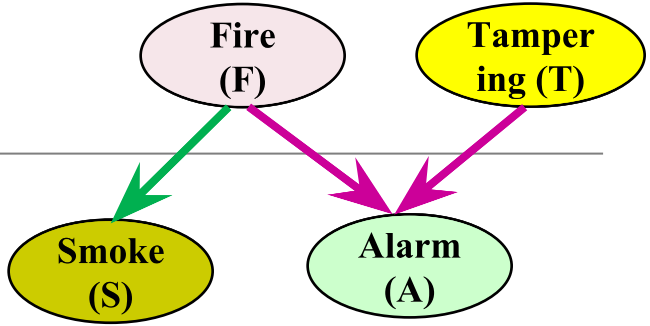



- Intercausal Reasoning:
  - The value of variable Fire does not influence Tampering
    - p(T|F) = p(T), F and T are independent
  - However, observing Alarm makes Fire and Tampering coupled
    - p(T|F, A) = $\frac{p(T,A,F)}{p(A|F)p(F)} = \frac{p(A|F,T)p(F)p(T)}{p(F)\sum_T p(A,T|F)} = \frac{p(A|F,T)p(T)}{\sum_T p(A|F,T)p(T)}$

P(A/B) = P(A,B)/P(B)

P(A/B) = P(B/A) * P(A)/$\sum_a$P(B,a)

# Calculation




- $p(T|F,A) = \frac{p(T,A,F)}{p(A|F)p(F)} = \frac{p(A|F,T)p(F)p(T)}{p(F)\sum_T p(A,T|F)} = \frac{p(A|F,T)p(T)}{\sum_T p(A|F,T)p(T)}$

| | Alarm=1 | Alarm=0 |
|---|---|---|
| F=1,T=1 | 0.5 | 0.5 |
| F=1,T=0 | 0.99 | 0.01 |
| F=0,T=1 | 0.85 | 0.15 |
| F=0,T=0 | 0.0001 | 0.9999 |

| Tampering=1 | Tampering=0 |
|---|---|
| 0.1 | 0.9 |

$$
\begin{aligned}
p(T=1|F=1,A=1) &= \frac{p(T=1,F=1,A=1)}{p(A=1|F=1)p(F=1)} \\
&= \frac{p(A=1|F=1,T=1)p(F=1)p(T=1)}{p(A=1|F=1)p(F=1)} \\
&= \frac{p(A=1|F=1,T=1)p(T=1)}{\sum_T p(A=1,T|F=1)} \\
&= \frac{0.5\times 0.1}{0.5\times 0.1 + 0.99\times 0.9} \\
&= 0.053
\end{aligned}
$$

$$
\begin{aligned}
p(T=1|F=0,A=1) &= \frac{p(A=1|T=1,F=0)p(T=1)}{\sum_T p(A=1,T|F=0)} \\
&= \frac{0.85\times 0.1}{0.85\times 0.1 + 0.001\times 0.9} \\
&= 0.9989 \sim 1.000
\end{aligned}
$$

$$
\begin{aligned}
s(F=1|S=1) &= \frac{p(S=1,F=1)}{p(S=1)} \\
&= \frac{p(S=1|F=1)\times p(F=1)}{\sum_f p(S=1,F)} \\
&= \frac{p(S=1|F=1)\times p(F=1)}{p(S=1|F=1)p(F=1)+p(S=1|F=0)p(F=0)} \\
&= \frac{0.9\times 0.02}{0.9\times 0.02+0.01\times 0.98} \\
&= 0.647
\end{aligned}
$$

$$
\begin{aligned}
p(T=1|F=1,A=1) &= \frac{p(T=1,F=1,A=1)}{p(A=1|F=1)p(F=1)} \\
&= \frac{p(A=1|F=1,T=1)p(F=1)p(T=1)}{p(A=1|F=1)p(F=1)} \\
&= \frac{p(A=1|F=1,T=1)p(T=1)}{\sum_T p(A=1,T|F=1)} \\
&= \frac{0.5\times 0.1}{0.5\times 0.1+0.99\times 0.9} \\
&= 0.053
\end{aligned}
$$

$$
\begin{aligned}
p(T=1|F=0,A=1) &= \frac{p(A=1|T=1,F=0)p(T=1)}{\sum_T p(A=1,T|F=0)} \\
&= \frac{0.85\times 0.1}{0.85\times 0.1+0.001\times 0.9} \\
&= 0.9989 \sim 1.000
\end{aligned}
$$