CS5201: Advanced Artificial Intelligence

Decision Trees

Arijit Mondal
Dept of Computer Science and Engineering
Indian Institute of Technology Patna
www.iitp.ac.in/~arijit/

Learning

- An agent is learning if it improves its performance on future tasks after making observation about the world
- Why would an agent learn?
  - Designers cannot anticipate all possible situations
  - Designers cannot anticipate all changes over time
  - Sometime, people have no idea how to program a solution
- Inductive learning - Learning a general function or rule from specific input-output pairs
- Analytical / deductive learning - Going from a known general rule to a new rule that is logically entailed

Paradigms of learning

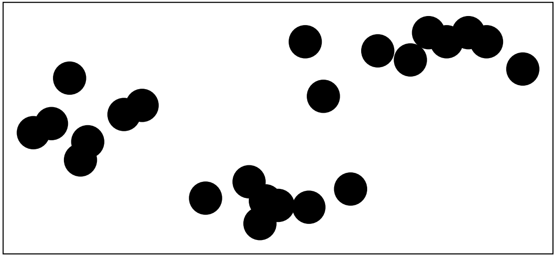
- These are based on the types of feedback
- Supervised learning
  - Both inputs and outputs are given
  - The outputs are typically provided by a friendly teacher
- Reinforcement learning
  - The agent receives some evaluation of its actions (such as a fine for stealing bananas), but is not told the correct action (such as how to buy bananas)
- Unsupervised learning
  - The agent can learn relationships among its percepts, and the trend with time

Supervised learning

- A set of labeled examples $\langle x_1, x_2, \ldots, x_n, y \rangle$
  - x_i are input variables
  - y output variable
- Need to find a function $f : X_1 \times X_2 \times \ldots X_n \rightarrow Y$
- Goal is to minimize error/loss function
  - Like to minimize over all dataset
  - We have limited dataset

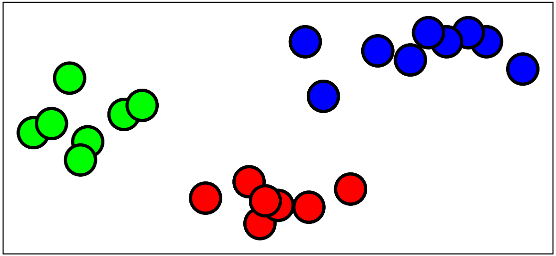
Unsupervised learning

- Learns useful properties of the structure of data set
- Unlabeled data
  - Tries to learn entire probability distribution that generated the dataset
  - Examples
    - Clustering, dimensionality reduction



Unsupervised learning

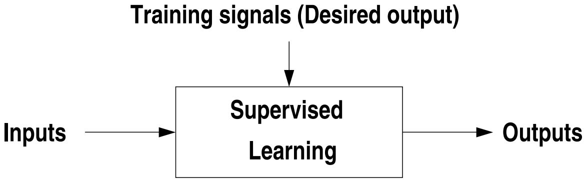
- Learns useful properties of the structure of data set
- Unlabeled data
  - Tries to learn entire probability distribution that generated the dataset
  - Examples
    - Clustering, dimensionality reduction

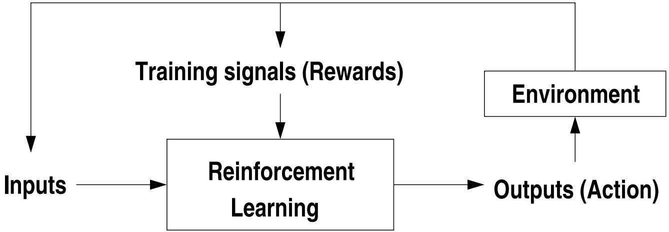


Reinforcement learning

- Set of actions that the learner will make in order to maximize its profit
- Action may not only affect the next situation but also subsequent situation
  - Trial and error search
  - Delayed reward
- A learning agent is interacting with environment to achieve a goal
- Agent needs to have idea of state so that it can take right action
- Three key aspects – observation, action, goal

Reinforcement vs supervised learning





Decision trees

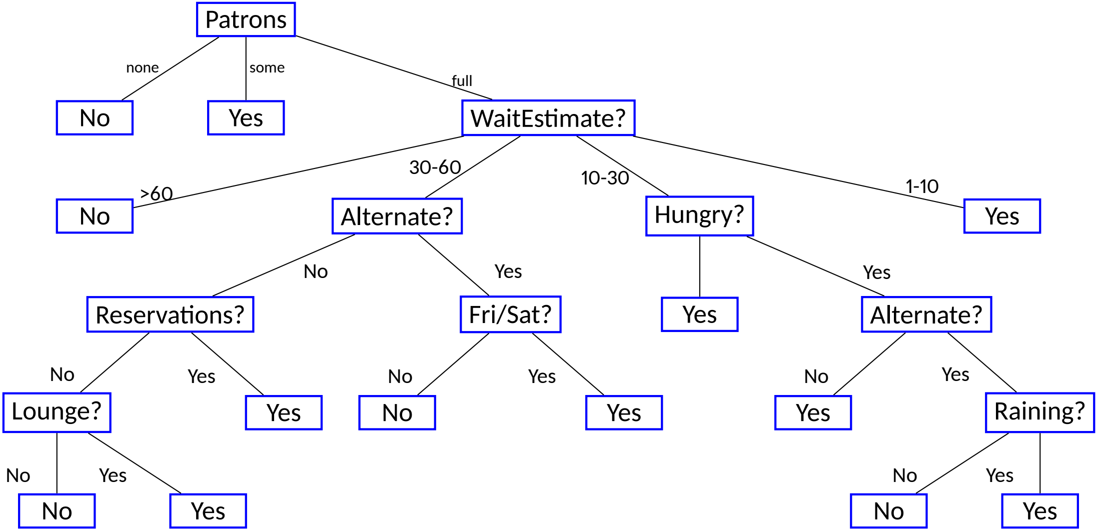
- A decision tree takes as input an object or situation described by a set of properties, and outputs a yes/no ”decision”
- A list of variables which potentially affect the decision on whether to wait for a table at a restaurant.
  - Alternate: whether there is a suitable alternative restaurant
  - Lounge: whether the restaurant has a lounge for waiting customers
  - Fri/Sat: true on Fridays and Saturdays
  - Hungry: whether we are hungry
  - Patrons: how many people are in it (None, Some, Full)
  - Price: the restaurant’s rating (*, **, ***)
  - Raining: whether it is raining outside
  - Reservation: whether we made a reservation
  - Type: the kind of restaurant (Indian, Chinese, Thai, Fastfood)
  - WaitEstimate: 0-10 mins, 10-30, 30-60, >60.

Observations

Example	Input Attributes										Goal
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*WillWait*
$\mathbf{x}_1$	*Yes*	*No*	*No*	*Yes*	*Some*	*$$$*	*No*	*Yes*	*French*	*0–10*	$_1 =$ *Yes*
$\mathbf{x}_2$	*Yes*	*No*	*No*	*Yes*	*Full*	*$*	*No*	*No*	*Thai*	*30–60*	$_2 =$ *No*
$\mathbf{x}_3$	*No*	*Yes*	*No*	*No*	*Some*	*$*	*No*	*No*	*Burger*	*0–10*	$_3 =$ *Yes*
$\mathbf{x}_4$	*Yes*	*No*	*Yes*	*Yes*	*Full*	*$*	*Yes*	*No*	*Thai*	*10–30*	$_4 =$ *Yes*
$\mathbf{x}_5$	*Yes*	*No*	*Yes*	*No*	*Full*	*$$$*	*No*	*Yes*	*French*	*60*	$_5 =$ *No*
$\mathbf{x}_6$	*No*	*Yes*	*No*	*Yes*	*Some*	*$$*	*Yes*	*Yes*	*Italian*	*0–10*	$_6 =$ *Yes*
$\mathbf{x}_7$	*No*	*Yes*	*No*	*No*	*None*	*$*	*Yes*	*No*	*Burger*	*0–10*	$_7 =$ *No*
$\mathbf{x}_8$	*No*	*No*	*No*	*Yes*	*Some*	*$$*	*Yes*	*Yes*	*Thai*	*0–10*	$_8 =$ *Yes*
$\mathbf{x}_9$	*No*	*Yes*	*Yes*	*No*	*Full*	*$*	*Yes*	*No*	*Burger*	*60*	$_9 =$ *No*
$\mathbf{x}_{10}$	*Yes*	*Yes*	*Yes*	*Yes*	*Full*	*$$$*	*No*	*Yes*	*Italian*	*10–30*	$_{10} =$ *No*
$\mathbf{x}_{11}$	*No*	*No*	*No*	*No*	*None*	*$*	*No*	*No*	*Thai*	*0–10*	$_{11} =$ *No*
$\mathbf{x}_{12}$	*Yes*	*Yes*	*Yes*	*Yes*	*Full*	*$*	*No*	*No*	*Burger*	*30–60*	$_{12} =$ *Yes*

Image source: AI by Russel & Norvig

Sample decision tree



Decision Tree Learning

- Aim: find a small tree consistent with the training examples
- Idea: (recursively) choose ”most significant” attribute as root of (sub) tree

1. pick an attribute to split at a non-terminal node
2. split examples into groups based on attribute value
3. for each group:
   A. if no examples - return majority from parent
   B. else if all examples in same class - return class
   C. else loop to step 1

Choosing an attribute

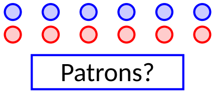
Idea: A good attribute splits the examples into subsets that are (ideally) ”all positive” or ”all negative”

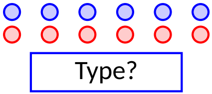
Patrons?

Type?

Choosing an attribute

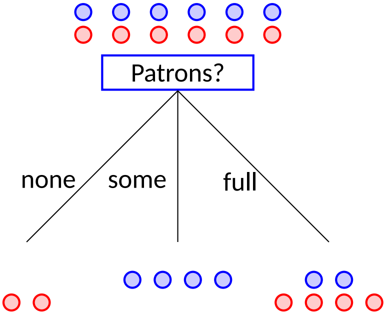
Idea: A good attribute splits the examples into subsets that are (ideally) ”all positive” or ”all negative”

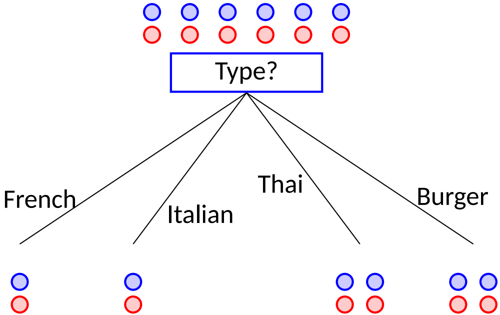




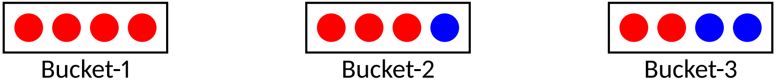
Choosing an attribute

Idea: A good attribute splits the examples into subsets that are (ideally) ”all positive” or ”all negative”



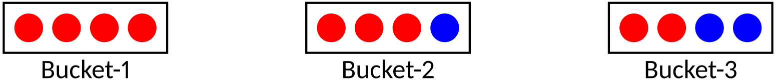


Choosing an attribute



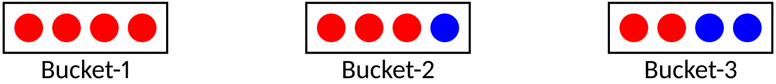
- **How much information do we have on the color of a ball drawn at random?**

Choosing an attribute



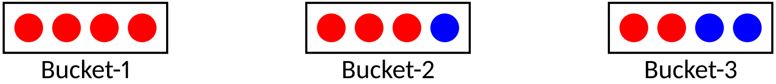
- **How much information do we have on the color of a ball drawn at random?**
  - **In the first bucket we are sure that the ball will be red**
  - **In the second bucket we know with 75% certainty that the ball will be red**
  - **In the third bucket we know with 50% certainty that the ball will be red**

Choosing an attribute



- How much information do we have on the color of a ball drawn at random?
  - In the first bucket we are sure that the ball will be red
  - In the second bucket we know with 75% certainty that the ball will be red
  - In the third bucket we know with 50% certainty that the ball will be red
- Bucket-1 gives us the most amount of knowledge about the color of the ball
- Entropy is the opposite of knowledge
  - Bucket-1 has the least amount of entropy and Bucket-3 has the highest entropy

Entropy and Probability



- **How many distinct arrangements of the balls are possible?**
  - **For the first bucket we have only one arrangement: RRRR**
  - **For the second bucket we have four arrangements: RRRB, RRBR, RBRR, BRRR**
  - **For the third bucket we have six arrangements: RRBB, RBBR, BBRR, RBRB, BRBR, BRRB**
- **The probability of finding a specific arrangement in four draws of balls is less for the third bucket because the number of possible arrangements is larger**

A game to understand entropy

- We are given, again, the three buckets to choose. The rules go as follows:
  - We choose one of the three buckets.
  - We are shown the balls in the bucket, in some order. Then, the balls go back in the bucket.
  - We then pick one ball out of the bucket, at a time, record the color, and return the ball back to the bucket.
  - If the colors recorded make the same sequence than the sequence of balls that we were shown at the beginning, then we win. If not, then we lose.

- Probability for win
  - Bucket-1:
  - Bucket-2:
  - Bucket-3:

Issues with probability

- **Products of many probability terms will make the metric very small and create precision problems**
- **Instead, we can take the logarithm of P(win), which will convert the product into a sum. Since probability terms are fractional, the logarithm will be negative and hence we take its negation**
- **For example, for Bucket-2 we compute:**

  $$-\log_2(0.75) - \log_2(0.75) - \log_2(0.75) - \log_2(0.25) = 3.245$$

- **Finally we take the average in order to normalize:**

  $$\frac{1}{4}(-\log_2(0.75) - \log_2(0.75) - \log_2(0.75) - \log_2(0.25)) = 0.81125$$

Entropy

- **Entropy =** $\frac{-m}{m+n}\log_2\left(\frac{m}{m+n}\right)+\frac{-n}{m+n}\log_2\left(\frac{n}{m+n}\right)$
  - **Bucket-1:**

Entropy

- **Entropy =** $\frac{-m}{m+n}\log_2\left(\frac{m}{m+n}\right)+\frac{-n}{m+n}\log_2\left(\frac{n}{m+n}\right)$
  - **Bucket-1:** $\frac{-4}{4+0}\log_2\left(\frac{4}{4+0}\right)+\frac{-0}{4+0}\log_2\left(\frac{0}{4+0}\right)=0+0=0$
  - **Bucket-2:**

Entropy

- **Entropy =** $\frac{-m}{m+n}\log_2\left(\frac{m}{m+n}\right)+\frac{-n}{m+n}\log_2\left(\frac{n}{m+n}\right)$
  - **Bucket-1:** $\frac{-4}{4+0}\log_2\left(\frac{4}{4+0}\right)+\frac{-0}{4+0}\log_2\left(\frac{0}{4+0}\right)=0+0=0$
  - **Bucket-2:** $\frac{-3}{3+1}\log_2\left(\frac{3}{3+1}\right)+\frac{-1}{3+1}\log_2\left(\frac{1}{3+1}\right)=0.81125$
  - **Bucket-3:**

Entropy

- **Entropy =** $\frac{-m}{m+n}\log_2\left(\frac{m}{m+n}\right)+\frac{-n}{m+n}\log_2\left(\frac{n}{m+n}\right)$
  - **Bucket-1:** $\frac{-4}{4+0}\log_2\left(\frac{4}{4+0}\right)+\frac{-0}{4+0}\log_2\left(\frac{0}{4+0}\right)=0+0=0$
  - **Bucket-2:** $\frac{-3}{3+1}\log_2\left(\frac{3}{3+1}\right)+\frac{-1}{3+1}\log_2\left(\frac{1}{3+1}\right)=0.81125$
  - **Bucket-3:** $\frac{-2}{2+2}\log_2\left(\frac{2}{2+2}\right)+\frac{-2}{2+2}\log_2\left(\frac{2}{2+2}\right)=\frac{1}{2}+\frac{1}{2}=1$

Attribute selection

- **Information content (Entropy):** $I(P(v_1), \ldots, P(v_n)) = \sum_{j=1}^{n} -P(v_j) \log_2 P(v_j)$

Attribute selection

- **Information content (Entropy):** $I(P(v_1), \ldots, P(v_n)) = \sum_{j=1}^{n} -P(v_j) \log_2 P(v_j)$
- **For a training set containing p positive examples and n negative examples:**

$$I\left(\frac{p}{p+n}, \frac{n}{p+n}\right) = -\frac{p}{p+n} \log_2 \frac{p}{p+n} - \frac{n}{p+n} \log_2 \frac{n}{p+n}$$

Attribute selection

- **Information content (Entropy):** $I(P(v_1), \ldots, P(v_n)) = \sum_{j=1}^{n} -P(v_j) \log_2 P(v_j)$
- **For a training set containing p positive examples and n negative examples:**
$$I\left(\frac{p}{p+n}, \frac{n}{p+n}\right) = -\frac{p}{p+n} \log_2 \frac{p}{p+n} - \frac{n}{p+n} \log_2 \frac{n}{p+n}$$
- **A chosen attribute A divides the training set E into subsets $E_1, \ldots, E_v$ according to their values for A_i where A has v distinct values**
$$\mathbf{R}(A) = \sum_{i=1}^{v} \frac{p_i + n_i}{p+n} I\left(\frac{p_i}{p_i+n_i}, \frac{n_i}{p_i+n_i}\right)$$

Attribute selection

- **Information content (Entropy):** $I(P(v_1), \ldots, P(v_n)) = \sum_{j=1}^{n} -P(v_j) \log_2 P(v_j)$
- **For a training set containing p positive examples and n negative examples:**

$$I\left(\frac{p}{p+n}, \frac{n}{p+n}\right) = -\frac{p}{p+n}\log_2\frac{p}{p+n} - \frac{n}{p+n}\log_2\frac{n}{p+n}$$

- **A chosen attribute A divides the training set E into subsets $E_1, \ldots, E_v$ according to their values for A_i where A has v distinct values**

$$\mathbf{R}(A) = \sum_{i=1}^{v} \frac{p_i + n_i}{p+n} I\left(\frac{p_i}{p_i+n_i}, \frac{n_i}{p_i+n_i}\right)$$

- **Information gain (IG) or reduction in entropy** $IG(A) = I\left(\frac{p}{p+n}, \frac{n}{p+n}\right) - \mathbf{R}(A)$
- **Choose the attribute with the largest IG**

Information gain: example

- **For the training set** $p = n = 6$, $I(\frac{6}{12}, \frac{6}{12}) = 1$ **bit**

$IG(Patrons) =$

Information gain: example

- **For the training set** $p = n = 6$**,** $I(\frac{6}{12}, \frac{6}{12}) = 1$ **bit**

$$IG(Patrons) = 1 - [\frac{2}{12}I(0,1) + \frac{4}{12}I(1,0) + \frac{6}{12}I(\frac{2}{6}, \frac{4}{6})] = 0.0541$$

$$IG(Type) =$$

Information gain: example

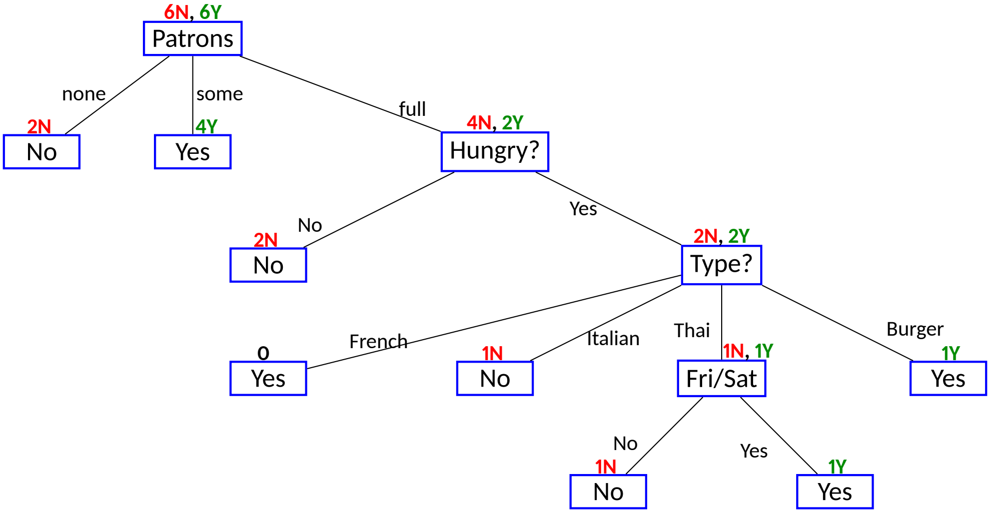
- **For the training set** $p = n = 6$, $I(\frac{6}{12}, \frac{6}{12}) = 1$ **bit**

$$IG(Patrons) = 1 - [\frac{2}{12}I(0,1) + \frac{4}{12}I(1,0) + \frac{6}{12}I(\frac{2}{6}, \frac{4}{6})] = 0.0541$$

$$IG(Type) = 1 - [\frac{2}{12}I(\frac{1}{2}, \frac{1}{2}) + \frac{2}{12}I(\frac{1}{2}, \frac{1}{2}) + \frac{4}{12}I(\frac{2}{4}, \frac{2}{4}) + \frac{4}{12}I(\frac{2}{4}, \frac{2}{4})] = 0$$

- **Patrons will be selected**

Final decision tree



A good tree

- Not too small: need to handle important but possibly subtle distinctions in data
- Not too big:
  - Computational efficiency (avoid redundant, spurious attributes)
  - Avoid over-fitting training examples

Exercise problem

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

Thank you!