Machine Learning and ID tree

What is learning?

Marvin Minsky said:

Learning is making useful changes in our minds.

From Wikipedia, the free encyclopedia

Learning is acquiring new, or modifying existing, knowledge, behaviors, skills, values, or preferences and may involve synthesizing different types of information. The ability to learn is possessed by humans, animals and some machines.

Herbert Simon said:

- Learning is any process by which a system improves performance from experience.
- Learning denotes changes in a system that ... enable a system to do the same task more efficiently the next time.

What is machine learning (ML)?

Tom Mitchell (prof. in Carnegie Mellon University) defined

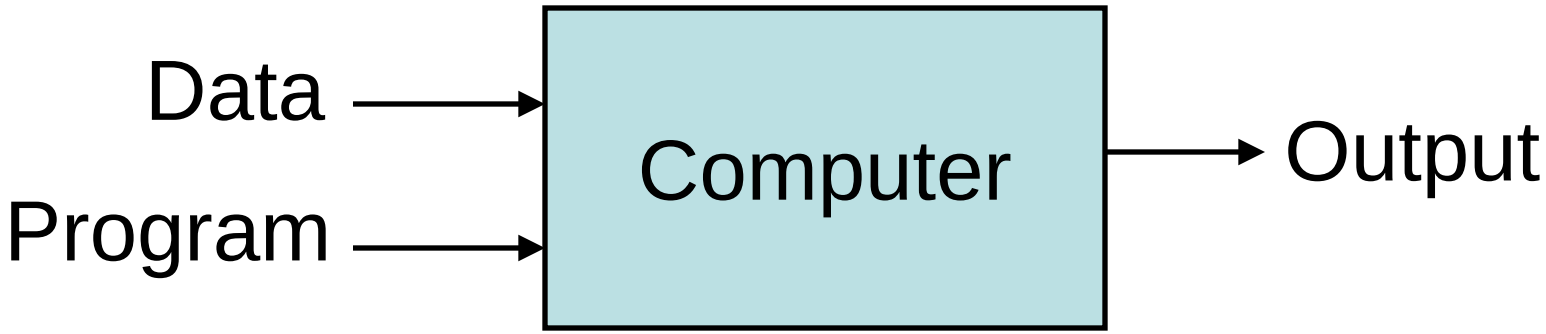
Definition:
A computer program is said to learn from experience *E* with respect to some class of tasks *T* and performance measure *P*, if its performance at tasks *T*, as measured by *P*, improves with experience *E*.

Machine Learning:

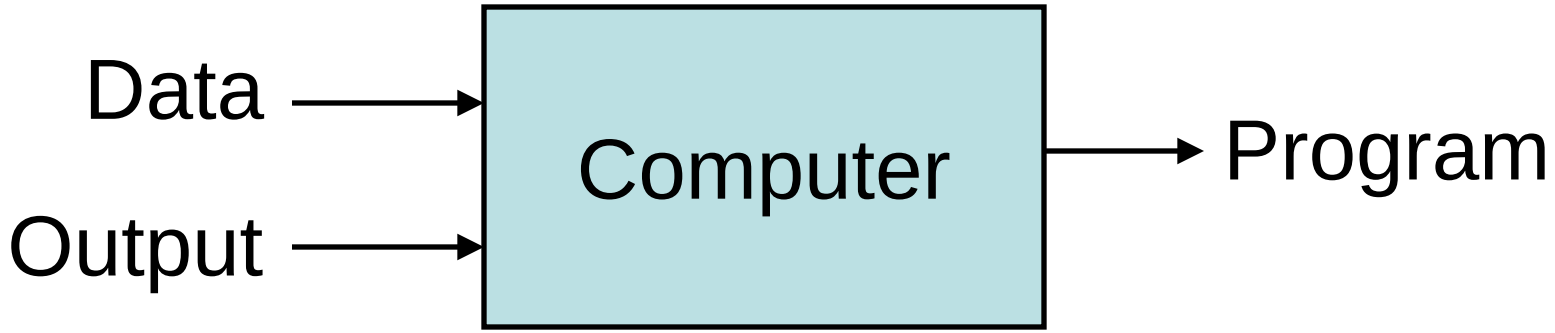
Study of algorithms that
- improve their performance P
- at some task T
- with experience E

well-defined learning task: <P,T,E>

Traditional Programming



Machine Learning



A Few Quotes

- “A breakthrough in machine learning would be worth ten Microsofts” (Bill Gates, Chairman, Microsoft)
- “Machine learning is the next Internet”
  (Tony Tether, Director, DARPA)
- Machine learning is the hot new thing”
  (John Hennessy, President, Stanford)
- “Web rankings today are mostly a matter of machine learning” (Prabhakar Raghavan, Dir. Research, Yahoo)
- “Machine learning is going to result in a real revolution”
  (Greg Papadopoulos, CTO, Sun)
- “Machine learning is today’s discontinuity”
  (Jerry Yang, CEO, Yahoo)

Can you find more quotes?

Give you 10 minutes to search for information about machine learning from the Internet

Using keywords: machine learning, learning algorithm, etc

Why is Machine Learning Important?

Why is Machine Learning Important?

- Some tasks cannot be defined well, except by examples (e.g., recognizing people).
- Relationships and correlations can be hidden within large amounts of data. Machine Learning/Data Mining may be able to find these relationships.
- Human designers often produce machines that do not work as well as desired in the environments in which they are used.
- The amount of knowledge available about certain tasks might be too large for explicit encoding by humans (e.g., medical diagnostic).
- Environments change over time.
- New knowledge about tasks is constantly being discovered by humans. It may be difficult to continuously re-design systems “by hand”.

Styles of machine learning

Human have many learning styles
How about machine?

Supervised Learning

- machine performs function (e.g., classification) after training on a data set where inputs and desired outputs are provided
  like decision trees

Unsupervised Learning

- Learning useful structure without labeled classes, optimization criterion, feedback signal, or any other information beyond the raw data
  like clustering

Semi-supervised Learning

??? Getting important in ML
Use unlabeled data to augment a small labeled sample to improve learning?

Supervised versus unsupervised

- Learn an unknown function f(X) = Y, where X is an input example and Y is the desired output.
  -- Supervised learning implies we are given a training set of (X, Y) pairs by a “teacher”
- Unsupervised learning means we are only given the Xs and some (ultimate) feedback functions on our performance.
- Supervised learning – programming by example
- Unsupervised learning
  - recognize similarities between inputs or identify features in the input data.
  - partition the data into group.

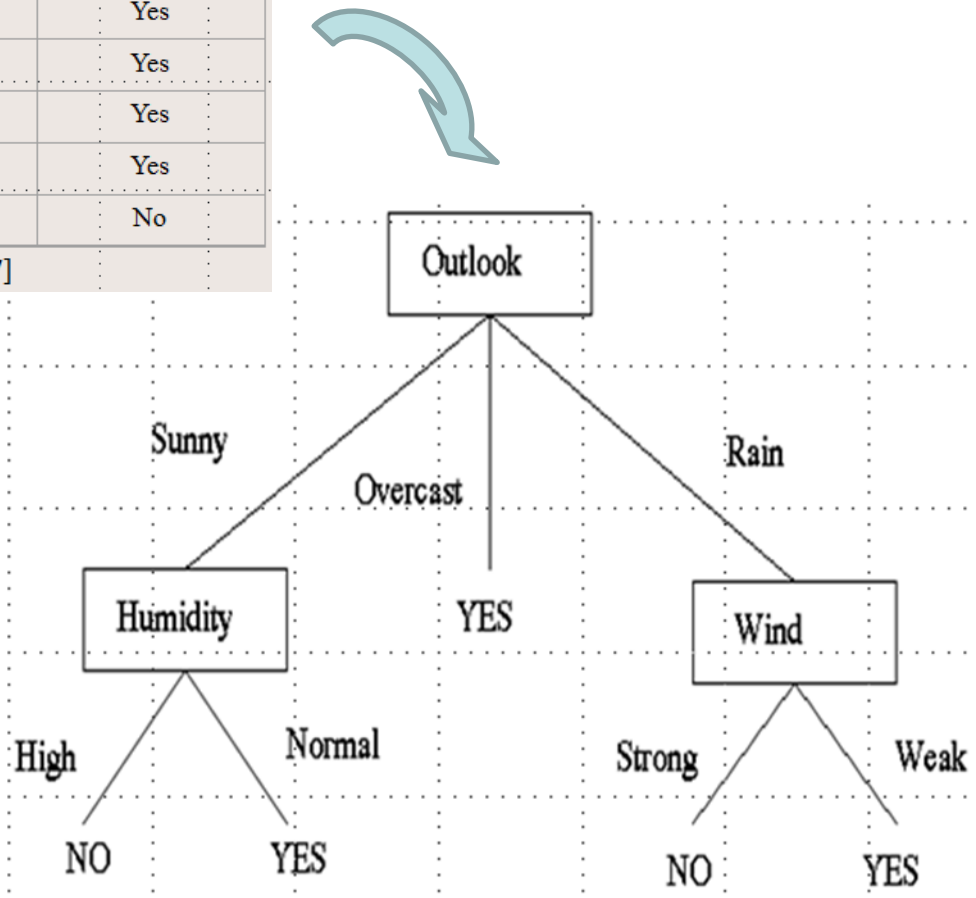
Decision Tree Learning

- **Learning Decision Trees**
  - Decision tree induction is a simple but powerful learning paradigm. In this method a set of training examples is broken down into smaller and smaller subsets while at the same time an associated decision tree get incrementally developed. At the end of the learning process, a decision tree covering the training set is returned.
  - The decision tree can be thought of as a set sentences (in Disjunctive Normal Form) written propositional logic.
  - Some characteristics of problems that are well suited to Decision Tree Learning are:
    - Attribute-value paired elements
    - Discrete target function
    - Disjunctive descriptions (of target function)
    - Works well with missing or erroneous training data

An 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

[See: Tom M. Mitchell, *Machine Learning*, McGraw-Hill, 1997]



Building a Decision Tree

1. First test all attributes and select the on that would function as the best root;
2. Break-up the training set into subsets based on the branches of the root node;
3. Test the remaining attributes to see which ones fit best underneath the branches of the root node;
4. Continue this process for all other branches until
   a. all examples of a subset are of one type
   b. there are no examples left (return majority classification of the parent)
   c. there are no more attributes left (default value should be majority classification)

Determining which attribute is best (Entropy & Gain)

- Entropy (E) is the minimum number of bits needed in order to classify an arbitrary example as yes or no
- $E(S) = \sum_{i=1}^{c} -p_i \log_2 p_i$,
  - Where S is a set of training examples,
  - c is the number of classes, and
  - p_i is the proportion of the training set that is of class i
- For our entropy equation $0 \log_2 0 = 0$
- The information gain G(S,A) where A is an attribute
- $G(S,A) \equiv E(S) - \sum_{v \text{ in Values}(A)} (|S_v| / |S|) * E(Sv)$

Decision Trees

example data sets

By calculating information entropy

classifiers and prediction models

apply information theory

By Shanon and Weaver (1949)

The unit of information is a bit, and the amount of information in a single binary answer is $log_2 P(v)$, where $P(v)$ is the probability of event v occurring.

Information needed for a correct answer,

$E(S) = I(p/(p+n), n/(p+n)) = - (p/(p+n)\ log_2\ p/(p+n)\ ) - n/(p+n) log_2\ n/(p+n)\ )$

Information contained in the remained sub-trees,

$Remainder(A) = \Sigma(p_i + n_i)\ /(p+n)\ I(p_i/(p_i + n_i), n_i/(p_i + n_i))$

$Gain(A) = I(p/(p+n), n/(p+n)) - Remainder(A)$

disorder

Outlook	Play Tennis
Sunny	No
Sunny	No
Overcast	Yes
Rain	Yes
Rain	Yes
Rain	No
Overcast	Yes
Sunny	No
Sunny	Yes
Rain	Yes
Sunny	Yes
Overcast	Yes
Overcast	Yes
Rain	No

P(Outlook = Rain and Play Tennis = yes) = 3/5
P(Outlook = Rain and Play Tennis = no) = 2/5

$$Entropy(S_{rain}) = -\frac{3}{5}\log_2\left(\frac{3}{5}\right) - \frac{2}{5}\log_2\left(\frac{2}{5}\right) = .971$$

$$Entropy(S_{overcast}) = -\frac{4}{4}\log_2\left(\frac{4}{4}\right) - 0\log_2(0) = 0$$

$$Entropy(S_{sunny}) = -\frac{2}{5}\log_2\left(\frac{2}{5}\right) - \frac{3}{5}\log_2\left(\frac{3}{5}\right) = .971$$

P(rain) = 5/14 P(overcast) = 4/14 P(sunny) = 5/14

$$\text{Entropy (Play Tennis | Outlook)} = -\frac{5}{14}(.971) - \frac{4}{14}(0) - \frac{5}{14}(.971) = .694$$

By knowing Outlook, how much information have I gained?

Entropy (Play Tennis) - Entropy (Play Tennis | Outlook) = .940 - .694 = .246

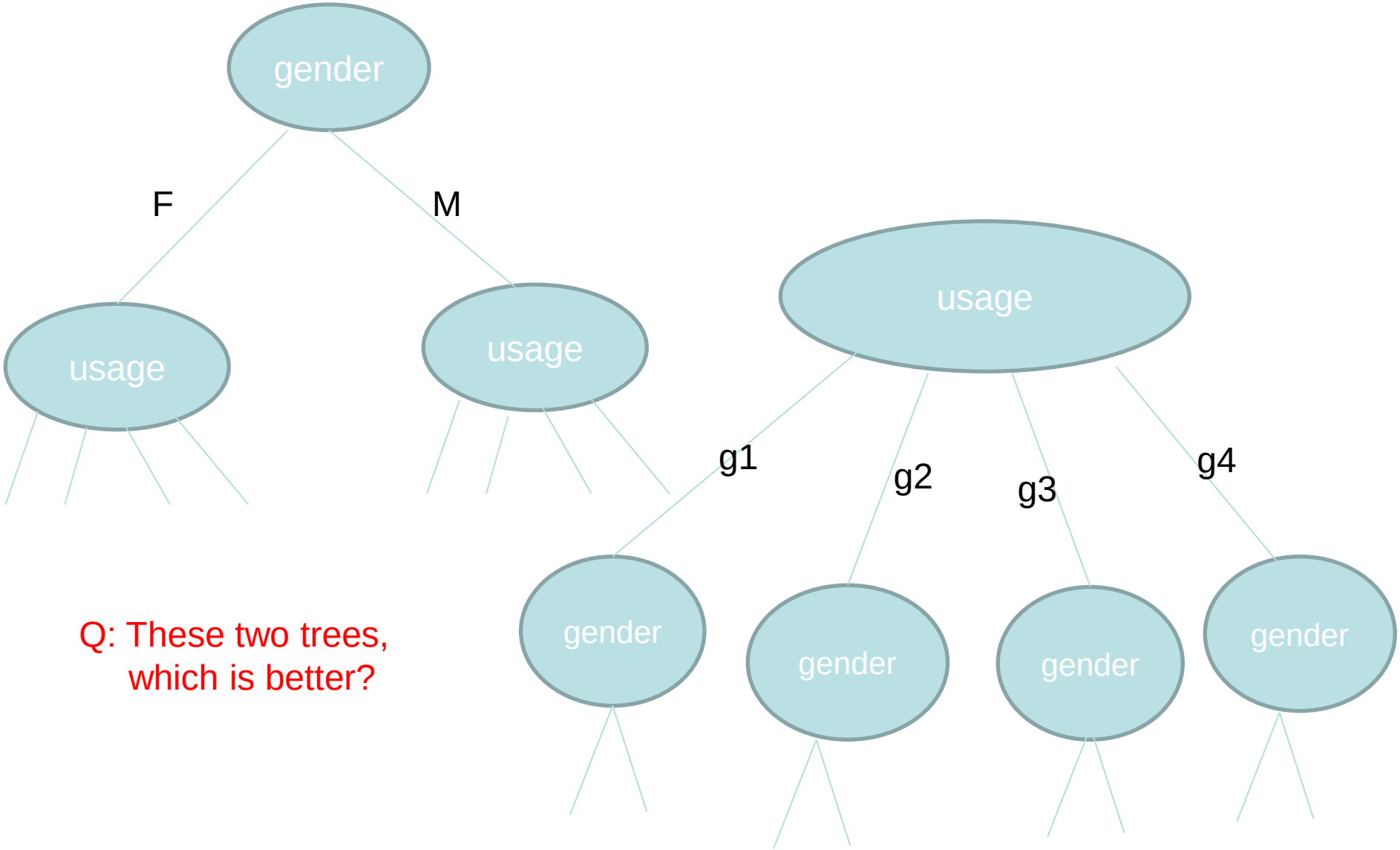
P(Play Tennis = Yes) = 9/14
P(Play Tennis = No) = 5/14

$$\text{Entropy(Play Tennis)} = -\,9/14 \log_2 (9/14) - 5/14 \log_2 (5/14) = .940$$

$$E(S) = \Sigma^{c}_{i=1} -p_i \log_2 p_i ,$$

How to make prediction: who is going to renew his/her video rental card, e.g. Tsutaya, T-card ?

If you have some training data.



Q: These two trees, which is better?

Information Gain

(an example)

Suppose that there are the total of 1000 customers, men renew 90 percent of the time, women renew 70 percent , and the customer set is made up half of men and half of women.

Information gain by testing whether a customer is a male or female?

remainder(gender) = -[(500/1000)I(450/500, 50/500)+(500/1000)I(350/500, 140/500)]

= (0.5)I(0.9, 0.1) - (0.5)I(0.7, 0.3) = 1-0.5x0.468996-0.5x0.881291

= 0.675143

Suppose that we had grouped the customers' usage habits into 3 groups: under 4 hours a month, from 4 to 10 hours, and over 10. The customers are evenly split among three groups. The first group renews at 50 percent, the second at 90 percent, and the third at 100 percent.

Information gain by testing on the attribute, usage?

remainder(usage) = - [(1/3)I(1/2, 1/2)+(1/3)I(9/10, 1/10)+(1/3)I(1, 0)]

= -0.333x1.0-0.333x0.466133-0.333x1.0

= 0.821222

Conclusion: **In building a decision tree, it is better to first split the data based on whether the customer was male or female, and then on how much connect-time they used.**

Information Gain

- The information gain of a feature F is the expected reduction in entropy resulting from splitting on this feature.

$$Gain(S,F) = Entropy(S) - \sum_{v \in Values(F)} \frac{|S_v|}{|S|} Entropy(S_v)$$

where S_v is the subset of S having value v for feature F.

- Entropy of each resulting subset weighted by its relative size.
- Example:

Ball	Size	Color	Weight	Rubber?	Result (Bounces?)
1	Small	green	Light	yes	yes
2	Small	blue	Medium	no	no
3	Medium	red	Medium	no	no
4	Small	red	Medium	yes	yes
5	Large	green	Heavy	yes	yes
6	Medium	blue	Heavy	yes	no
7	Medium	green	Heavy	yes	no
8	Small	red	Light	no	no

Figure C1: Identification Tree Training Data

S= Result (bounces?)
F = Size
|S|=8
V=1: Small
2: Large
3: Medium
$|S_1| = 4$
$|S_2| = 1$
$|S_3| = 3$

E(S)= I(p/(p+n), n/(p+n)) = - (p/(p+n) log2 p/(p+n)) - n/(p+n)log2 n/(p+n))

|S|=8

E(S) = - 3/8*log2(3/8) – 5/8*log2(5/8) = 0.954434

$$Gain(S,F) = Entropy(S) - \sum_{v \in Values(F)} \frac{|S_v|}{|S|} Entropy(S_v)$$

Gain(S, Size) =?
Gain(S, Color) =?
Gain(S, Weight) =?
Gain(S, Rubber) =?

Ball	Size	Color	Weight	Rubber?	Result (Bounces?)
1	Small	green	Light	yes	yes
2	Small	blue	Medium	no	no
3	Medium	red	Medium	no	no
4	Small	red	Medium	yes	yes
5	Large	green	Heavy	yes	yes
6	Medium	blue	Heavy	yes	no
7	Medium	green	Heavy	yes	no
8	Small	red	Light	no	no

Figure C1: Identification Tree Training Data

Four possible splitting:

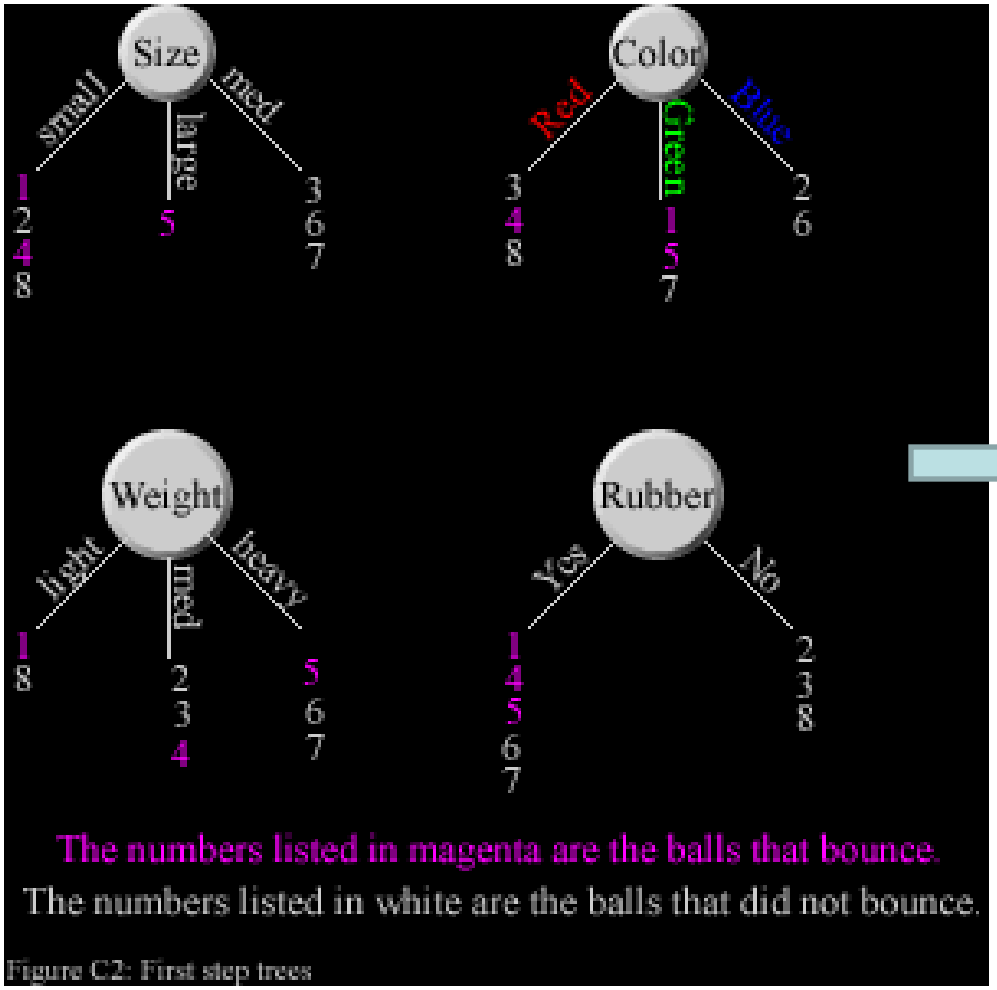


Figure C2: First step trees

Qs:
Which is better?
Which is the best?

Ball	Size	Color	Weight	Rubber?	Result (Bounces?)
1	Small	green	Light	yes	yes
2	Small	blue	Medium	no	no
3	Medium	red	Medium	no	no
4	Small	red	Medium	yes	yes
5	Large	green	Heavy	yes	yes
6	Medium	blue	Heavy	yes	no
7	Medium	green	Heavy	yes	no
8	Small	red	Light	no	no

Figure C1: Identification Tree Training Data

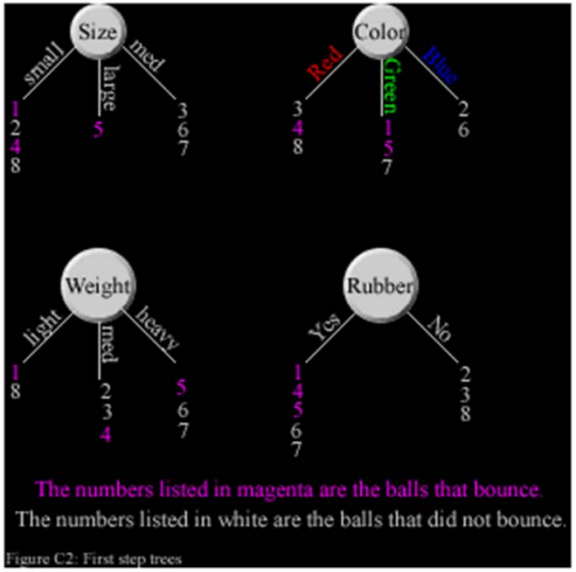


Figure C2: First step trees

(0.954434)

$$Gain(S,F) = Entropy(S) - \sum_{v \in Values(F)} \frac{|S_v|}{|S|} Entropy(S_v)$$

$Size\_Disorder = \Sigma_b\ (nb/nt) * (\Sigma_c\ -(nbc/nb)\log_2(nbc/nb))$

$= (4/8) * ((-(2/4)\ \log_2\ (2/4))$

$+ (-(2/4)\ \log_2\ (2/4))) + ((1/8) * 0) + ((3/8) * 0)$

$= 0.5$

How about color?

weight?

rubber?

Please write down their formulae.

Color: 0.69
Weight: 0.94
Rubber: 0.61

S:Size |S|=8
V=1: Small
2: Large
3: Medium
$|S_1| = 4$
$|S_2| = 1$
$|S_3| = 3$

Ball	Size	Color	Weight	Rubber?	Result (Bounces?)
1	Small	green	Light	yes	yes
2	Small	blue	Medium	no	no
3	Medium	red	Medium	no	no
4	Small	red	Medium	yes	yes
5	Large	green	Heavy	yes	yes
6	Medium	blue	Heavy	yes	no
7	Medium	green	Heavy	yes	no
8	Small	red	Light	no	no

Figure C1: Identification Tree Training Data

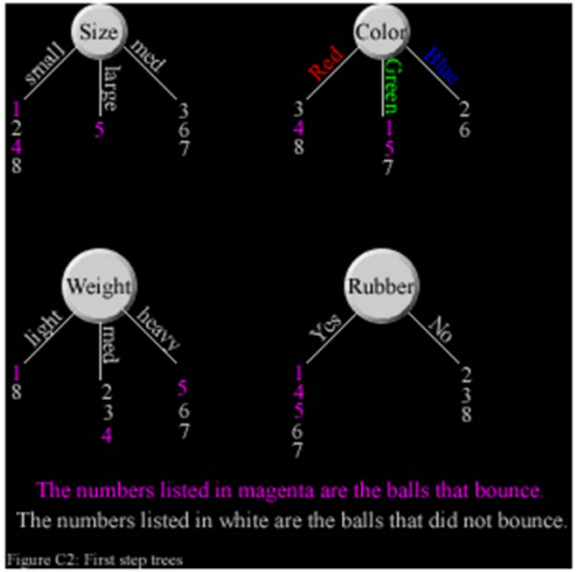


Figure C2: First step trees

$$Gain(S,F) = Entropy(S) - \sum_{v \in Values(F)} \frac{|S_v|}{|S|} Entropy(S_v)$$

Disorder

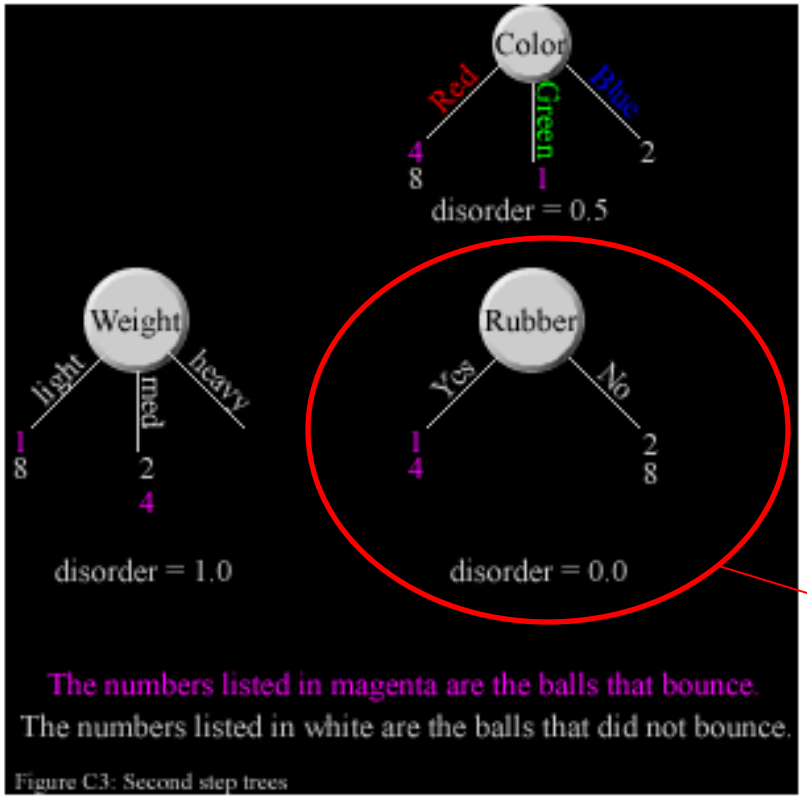
$$Size\_Disorder = \Sigma_b\ (nb/nt) * (\Sigma_c\ -(nbc/nb)\log_2(nbc/nb))$$

$$= (4/8) * ((-(2/4)\ \log_2\ (2/4)) + (-(2/4)\ \log_2\ (2/4))) + ((1/8) * 0) + ((3/8) * 0)$$

$$= 0.5$$

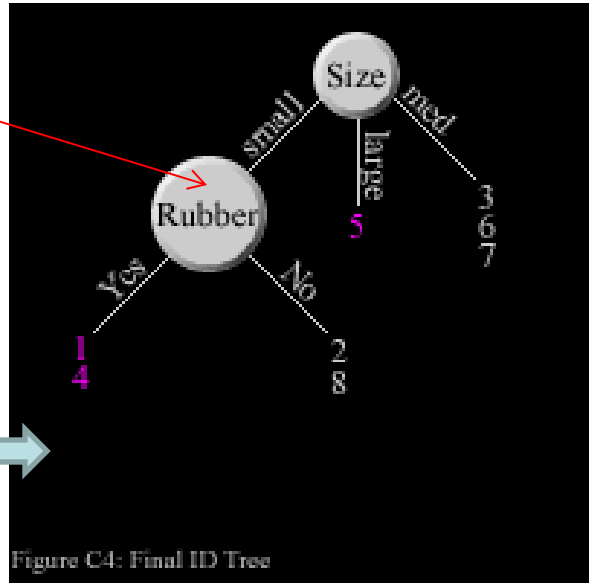
Color_Disorder = 0.69
Weight_Disorder = 0.94
Rubber_Disorder = 0.61

For the case of Size = small, continue to split this note



How about other two cases?
Split or not? Why?
- medium?
- large?

Finish splitting?
Why?



Home Work

Read the following site:

http://ai-depot.com/Tutorial/RuleBased.html