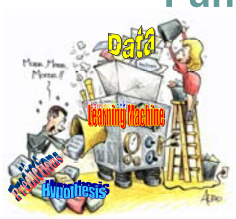
Machine Learning

10-701/15-781, Fall 2011

Introduction to ML and Functional Approximation



Eric Xing

Lecture 1, September 12, 2011

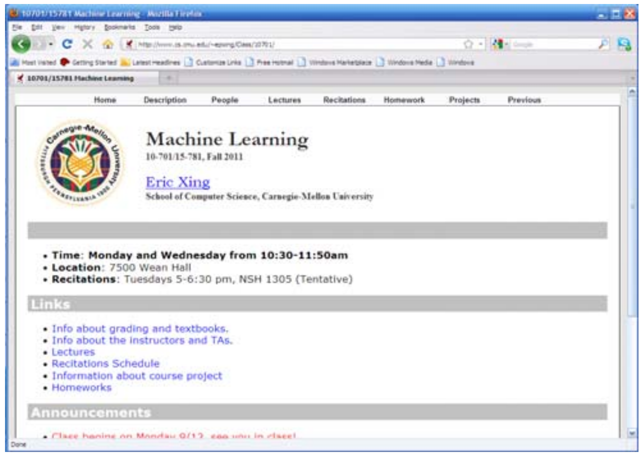
Reading: **Mitchell: Chap 1,3**

Class Registration

- IF YOU ARE ON THE WAITING LIST: This class is now fully subscribed. You may want to consider the following options:
  - Take the class when it is offered again in the Spring semester;
  - Come to the first several lectures and see how the course develops. We will admit as many students from the waitlist as we can, once we see how many registered students drop the course during the first two weeks.

Machine Learning 10-701/15-781

- Class webpage:
  - http://www.cs.cmu.edu/~epxing/Class/10701/



Logistics

- Text book
  - Chris Bishop, **Pattern Recognition and Machine Learning (required)**
  - Tom Mitchell, **Machine Learning**
  - David Mackay, **Information Theory, Inference, and Learning Algorithms**
- Mailing Lists:
  - To contact the instructors: 10701-instr@cs.cmu.edu
  - Class announcements list: 10701-announce@cs.cmu.edu.
- TA:
  - Qirong Ho, GHC 8013, Office hours: TBA
  - Nan Li, GHC 6505, Office hours: 11:00am-12:00pm
  - Suyash Shringarpure, GHC 8013, Office hours: Wednesday 2:00-3:00pm
  - Bin Zhao, GHC 8021, Office hours: Tuesday 3:00-4:00pm
  - Gunhee Kim
- Class Assistant:
  - Michelle Martin, GHC 8001, x8-5527

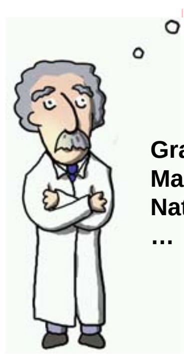
Logistics

- 5 homework assignments: 25% of grade
  - Theory exercises
  - Implementation exercises
- **Final project: 30% of grade**
  - Applying machine learning to your research area
    - NLP, IR,, vision, robotics, computational biology …
  - Outcomes that offer real utility and value
    - Search all the wine bottle labels,
    - An iPhone app for landmark recognition
  - Theoretical and/or algorithmic work
    - a more efficient approximate inference algorithm
    - a new sampling scheme for a non-trivial model …
  - 3-stage reports
- Two exams: 20% and 25% of grade each
  - Theory exercises and/or analysis. Dates already set (no “ticket already booked”, “I am in a conference”, etc. excuse …)
- Policies …

 5

What is Learning

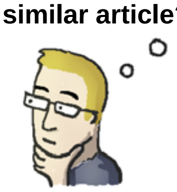
Learning is about seeking a predictive and/or executable understanding of natural/artificial subjects, phenomena, or activities from …



Apoptosis + Medicine

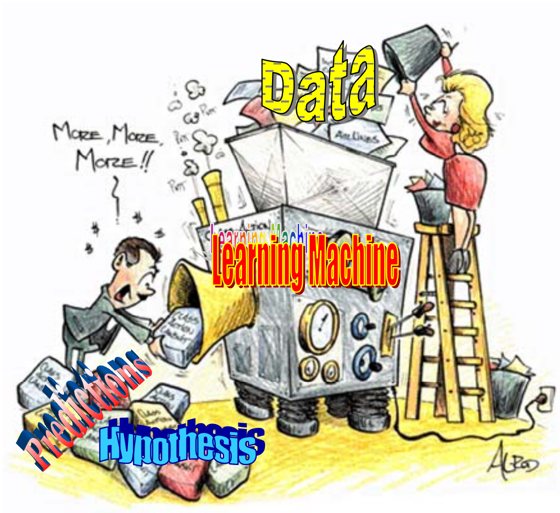
Grammatical rules
Manufacturing procedures
Natural laws
…

Inference:
what does this mean?
Any similar article?
…



 6

Machine Learning

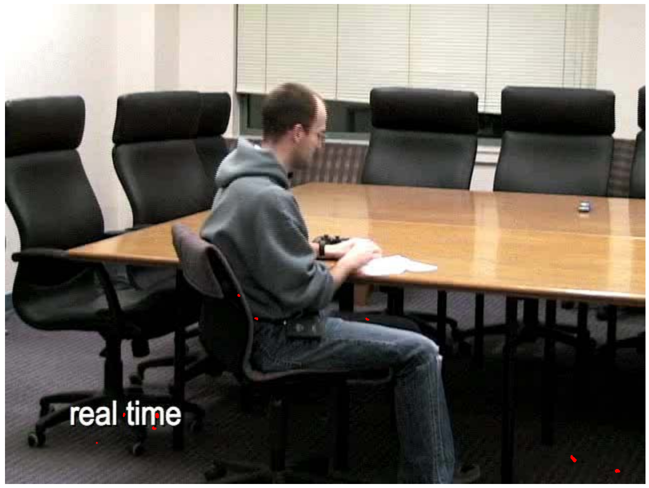


Machine Learning (short)

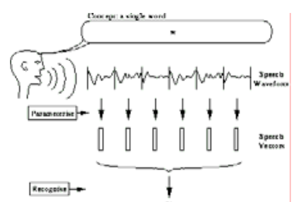
- Study of algorithms that
- improve their performance P
- at some task T
- with experience E

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

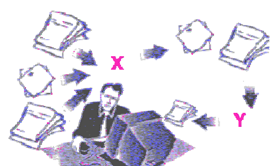
Fetching a stapler from inside an office --- the Stanford STAIR robot



Where Machine Learning is being used or can be useful?

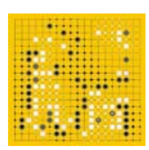


Speech recognition



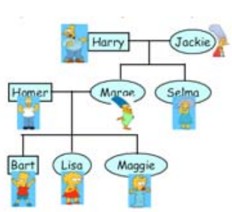
Information retrieval

Computer vision



Games

Robotic control



Pedigree

Evolution

Planning

Natural language processing and speech recognition

- Now most pocket **Speech Recognizers** or **Translators** are running on some sort of learning device --- the more you play/use them, the smarter they become!

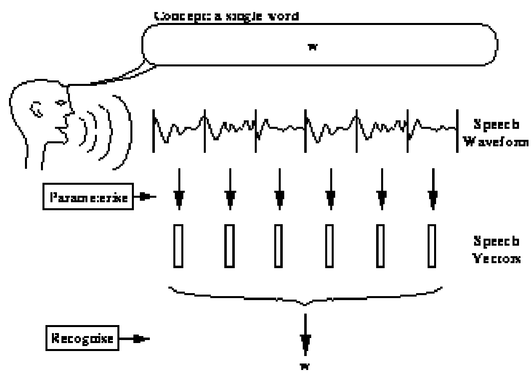


Fig. 1.2 Isolated Word Problem

Object Recognition

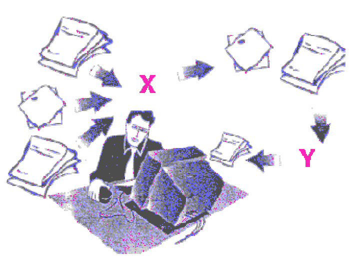
- Behind a security camera, most likely there is a computer that is learning and/or checking!

Robotic Control

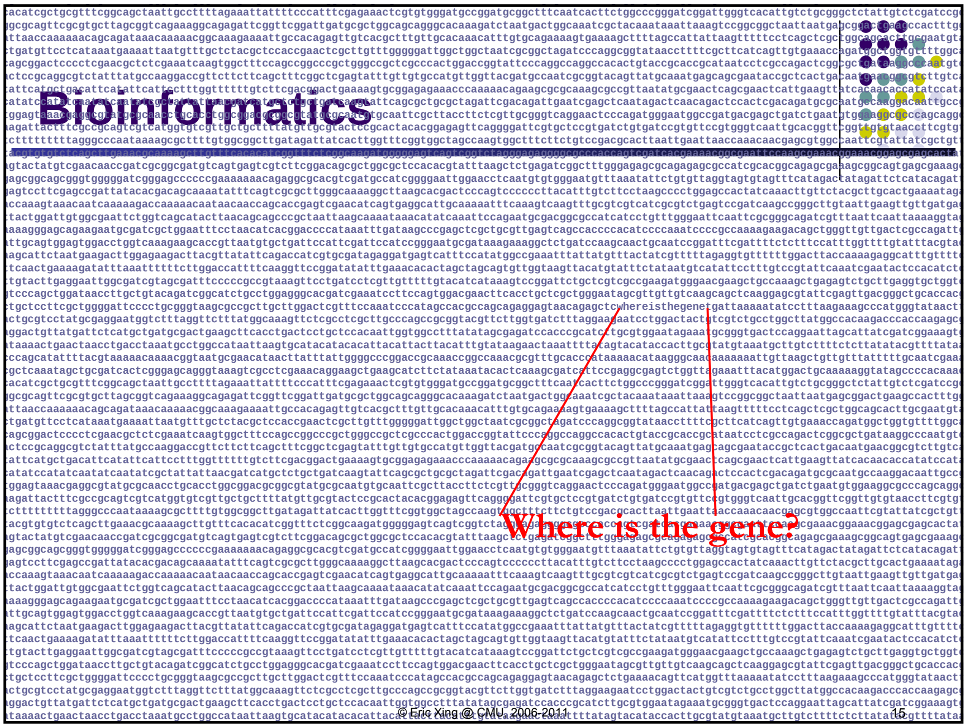
- The **best** helicopter pilot is now a computer!
  - it runs a program that learns how to fly and make acrobatic maneuvers by itself!
  - no taped instructions, joysticks, or things like …

Text Mining

- Reading, digesting, and categorizing a vast text database is too much for human!



- **We want:**



Paradigms of Machine Learning

- Supervised Learning
  - Given $D = \{\mathbf{X}_i, \mathbf{Y}_i\}$, learn $f(\cdot): \mathbf{Y}_i = f(\mathbf{X}_i)$, s.t. $D^{new} = \{\mathbf{X}_j\} \Rightarrow \{\mathbf{Y}_j\}$
- Unsupervised Learning
  - Given $D = \{\mathbf{X}_i\}$, learn $f(\cdot): \mathbf{Y}_i = f(\mathbf{X}_i)$, s.t. $D^{new} = \{\mathbf{X}_j\} \Rightarrow \{\mathbf{Y}_j\}$
- Semi-supervised Learning
- Reinforcement Learning
  - Given $D = \{\text{env, actions, rewards, simulator/trace/real game}\}$
    learn $\begin{matrix}\text{policy}: e, r \to a \\ \text{utility}: a, e \to r\end{matrix}$, s.t. $\{\text{env, new real game}\} \Rightarrow a_1, a_2, a_3 \ldots$
- Active Learning
  - Given $D \sim G(\cdot)$, learn $D^{new} \sim G'(\cdot)$ and $f(\cdot)$, s.t. $D^{all} \Rightarrow G'(\cdot)$, policy, $\{\mathbf{Y}_j\}$

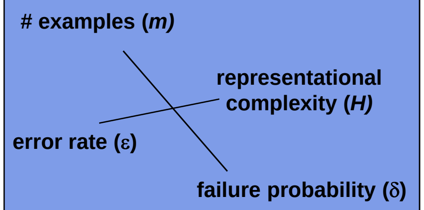
Machine Learning - Theory

For the learned F(; θ)

- Consistency (value, pattern, …)
- Bias versus variance
- Sample complexity
- Learning rate
- Convergence
- Error bound
- Confidence
- Stability
- …

PAC Learning Theory

(supervised concept learning)



$$m \geq \frac{1}{\epsilon}(\ln |H| + \ln(1/\delta))$$

Why machine learning?

13 million Wikipedia pages

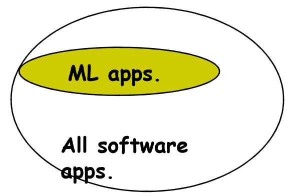
500 million users

3.6 billion photos

24 hours videos uploaded per minute

Growth of Machine Learning

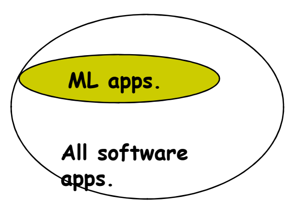
- Machine learning already the preferred approach to
  - Speech recognition, Natural language processing
  - Computer vision
  - Medical outcomes analysis
  - Robot control
  - …



- This ML niche is growing (why?)

Growth of Machine Learning

- Machine learning already the preferred approach to
  - Speech recognition, Natural language processing
  - Computer vision
  - Medical outcomes analysis
  - Robot control
  - …



- This ML niche is growing
  - Improved machine learning algorithms
  - Increased data capture, networking
  - Software too complex to write by hand
  - New sensors / IO devices
  - Demand for **self-customization to user, environment**

Summary:
What is Machine Learning

Machine Learning seeks to develop theories and computer systems for

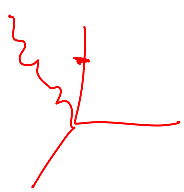
- **representing;**
- **classifying, clustering, recognizing, organizing;**
- **reasoning under uncertainty;**
- **predicting;**
- **and reacting to**
- **…**

complex, real world data, based on the system's own experience with data, and (hopefully) under a unified model or mathematical framework, that

- **can be formally characterized and analyzed**
- **can take into account human prior knowledge**
- **can generalize and adapt across data and domains**
- **can operate automatically and autonomously**
- **and can be interpreted and perceived by human.**

Elements of Machine Learning

- **Here are some important elements to consider before you start:**
  - Task:
    - Embedding? Classification? Clustering? Topic extraction? …
  - Data and other info:
    - Input and output (e.g., continuous, binary, counts, …)
    - Supervised or unsupervised, of a blend of everything?
    - Prior knowledge? Bias?
  - Models and paradigms:
    - BN? MRF? Regression? SVM?
    - Bayesian/Frequents ? Parametric/Nonparametric?
  - Objective/Loss function:
    - MLE? MCLE? Max margin?
    - Log loss, hinge loss, square loss? …
  - Tractability and exactness trade off:
    - Exact inference? MCMC? Variational? Gradient? Greedy search?
    - Online? Batch? Distributed?
  - Evaluation:
    - Visualization? Human interpretability? Perperlexity? Predictive accuracy?
- **It is better to consider one element at a time!**

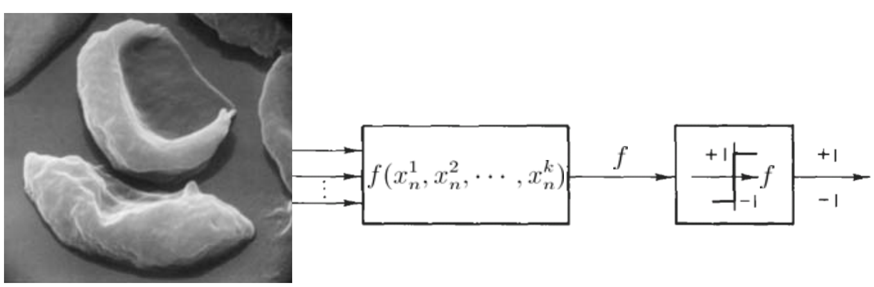


Inference
Prediction
Decision-Making under uncertainty
…

→ **Statistical Machine Learning**
→ **Function Approximation:** $F(\ |\theta)$?

Classification

- sickle-cell anemia



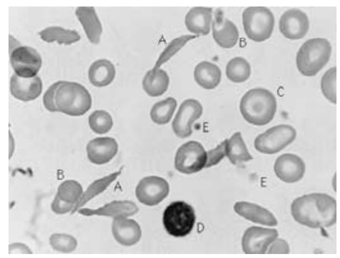
Function Approximation

- **Setting**:
  - Set of possible instances X = $\{x_1, x_2 \cdots x_n\}$
  - Unknown target function $f: X \rightarrow Y$
  - Set of function hypotheses $H=\{ h \mid h: X \rightarrow Y \}$

$x = \begin{bmatrix} \vdots \\ \vdots \end{bmatrix}$

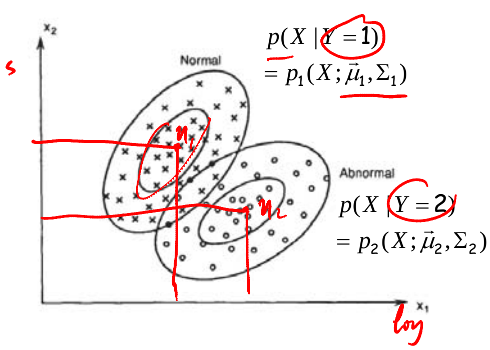
$y \in \{+1, -1\}$

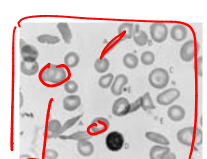
- **Given**:
  - Training examples $\{<x_i, y_i>\}$ of unknown target function f
- **Determine**:
  - Hypothesis $h \in H$ that best approximates f



Decision-making as dividing a high-dimensional space

- Classification-specific Dist.: P(X|Y)





$h(y \leftarrow x)$
$= P(y|x)$
posterior dist

- Class prior (i.e., "weight"): P(Y)

The Bayes Rule

- What we have just did leads to the following general expression:

$$h(y) = P(Y \mid X) = \frac{P(X \mid Y)\,p(Y)}{P(X)}$$

This is Bayes Rule

Bayes, Thomas (1763) An essay towards solving a problem in the doctrine of chances. *Philosophical Transactions of the Royal Society of London,* **53:370-418**

Example of a learned decision rule

- When each class is a normal …

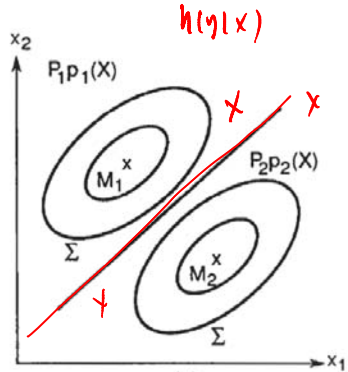
x^{new}

$h(y=1 \mid x^{new})$

$\vee \wedge \parallel$

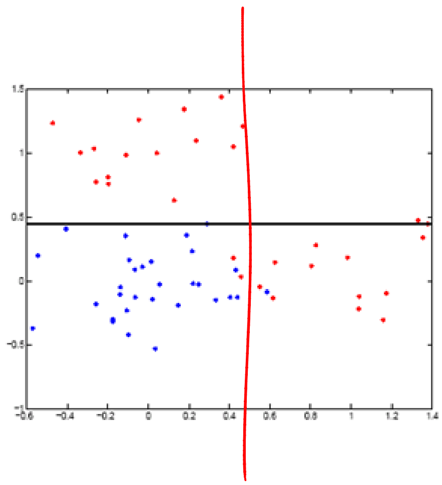
$h(y=-1 \mid x^{new})$

$h(y \mid x)$



- We can write the decision boundary analytically in some cases … homework!!

Complex decision boundary



A Tax-Fraud detection problem:

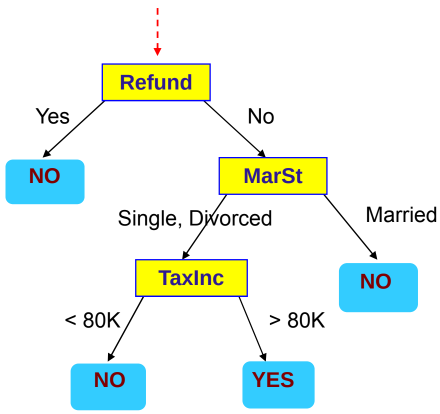
- What F to use?
  - Hypothesis
- How to use?

Query Data

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?

Apply a Decision Tree to the Query

Start from the root of tree.



Query Data

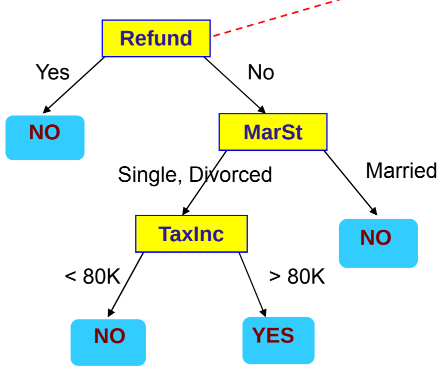
Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?

h( )

Apply Model to Test Data

Query Data

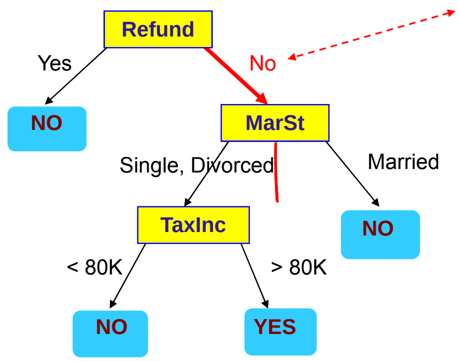
Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?



Apply Model to Test Data

Query Data

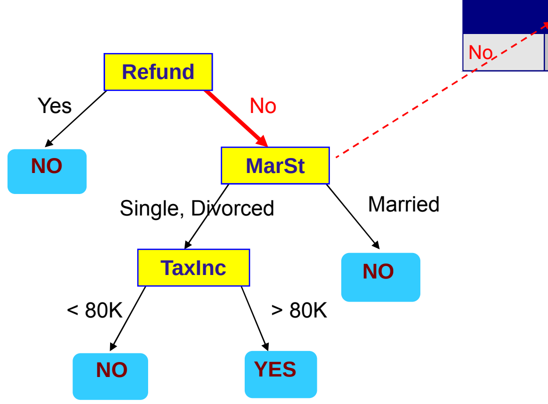
Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?



Apply Model to Test Data

Query Data

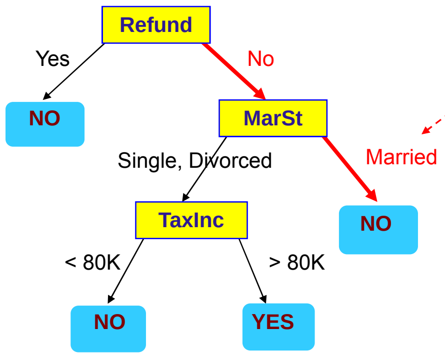
Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?



Apply Model to Test Data

Query Data

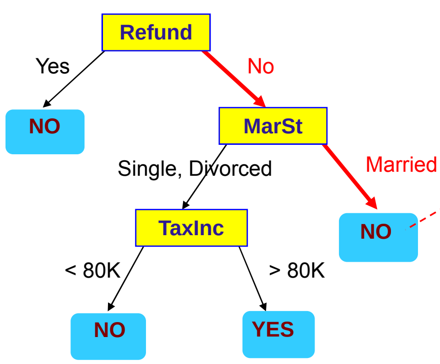
Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?



Apply Model to Test Data

Query Data

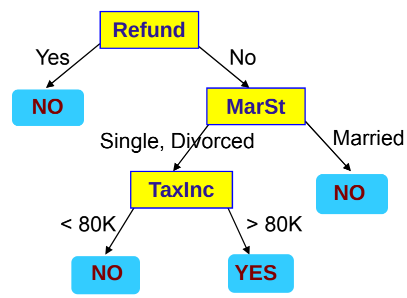
Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?



Assign Cheat to "No"

A hypothesis for *TaxFraud*

- Input: a vector of attributes
  - X=[Refund,MarSt,TaxInc]
- Output:
  - Y= Cheating or Not
- *H* as a procedure:



- Each internal node: test one attribute X_i
- Each branch from a node: selects one value for X_i
- Each leaf node: predict Y

A Tree to Predict C-Section Risk

- Learned from medical records of 1000 wonman

  Negative examples are C-sections

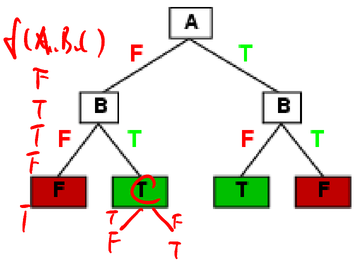
```
[833+,167-] .83+ .17-
Fetal_Presentation = 1: [822+,116-] .88+ .12-
| Previous_Csection = 0: [767+,81-] .90+ .10-
| | Primiparous = 0: [399+,13-] .97+ .03-
| | Primiparous = 1: [368+,68-] .84+ .16-
| | | Fetal_Distress = 0: [334+,47-] .88+ .12-
| | | | Birth_Weight < 3349: [201+,10.6-] .95+
| | | | Birth_Weight >= 3349: [133+,36.4-] .78+
| | | Fetal_Distress = 1: [34+,21-] .62+ .38-
| Previous_Csection = 1: [55+,35-] .61+ .39-
Fetal_Presentation = 2: [3+,29-] .11+ .89-
Fetal_Presentation = 3: [8+,22-] .27+ .73-
```

Expressiveness

- Decision trees can express any function of the input attributes.
- E.g., for Boolean functions, truth table row → path to leaf:

C	A	B	A xor B	f(A,B,C)
F	F	F	F	F
F	F	T	T	T
F	T	F	T	T
F	T	T	F	F
F	T	T	T	
F	T	F	T	

$x^{new} =$ F T F T

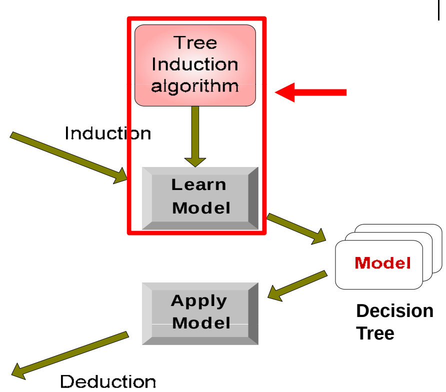


- Trivially, there is a consistent decision tree for any training set with one path to leaf for each example (unless *f* nondeterministic in *x*) but it probably won't generalize to new examples
- Prefer to find more compact decision trees

Learning a Decision

Tid	Attrib1	Attrib2	Attrib3	Class
1	Yes	Large	125K	No
2	No	Medium	100K	No
3	No	Small	70K	No
4	Yes	Medium	120K	No
5	No	Large	95K	Yes
6	No	Medium	60K	No
7	Yes	Large	220K	No
8	No	Small	85K	Yes
9	No	Medium	75K	No
10	No	Small	90K	Yes

Training Set



Tid	Attrib1	Attrib2	Attrib3	Class
11	No	Small	55K	?
12	Yes	Medium	80K	?
13	Yes	Large	110K	?
14	No	Small	95K	?
15	No	Large	67K	?

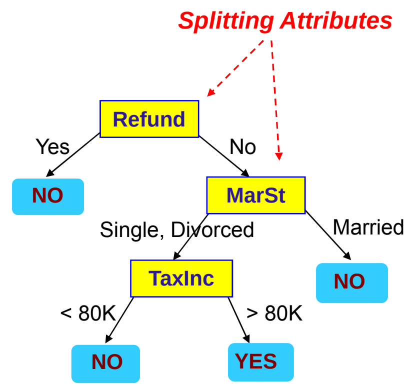
Test Set

Example of a Decision Tree

categorical categorical continuous class

Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

Training Data



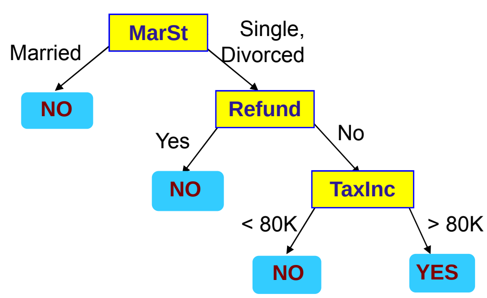
Model: Decision Tree

Another Example of Decision Tree

categorical categorical continuous class

Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

Training Data



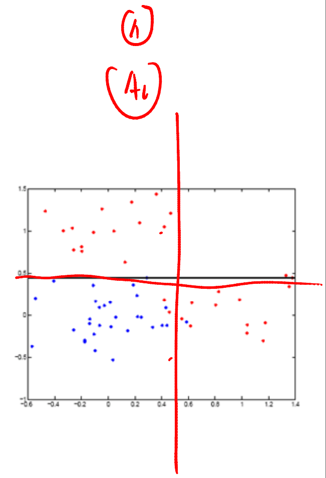
There could be more than one tree that fits the same data!

Top-Down Induction of DT

Main loop:

1. $A \leftarrow$ the "best" decision attribute for next *node*
2. Assign A as decision attribute for *node*
3. For each value of A, create new descendant of *node*
4. Sort training examples to leaf nodes
5. If training examples perfectly classified, Then STOP, Else iterate over new leaf nodes

Which attribute is best?



Tree Induction

- Greedy strategy.
  - Split the records based on an attribute test that optimizes certain criterion.
- Issues
  - Determine how to split the records
    - How to specify the attribute test condition?
    - How to determine the best split?
  - Determine when to stop splitting

Tree Induction

- Greedy strategy.
  - Split the records based on an attribute test that optimizes certain criterion.

- Issues
  - Determine how to split the records
    - How to specify the attribute test condition?
    - How to determine the best split?
  - Determine when to stop splitting

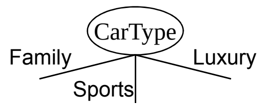
How to Specify Test Condition?

- Depends on attribute types
  - Nominal
  - Ordinal
  - Continuous

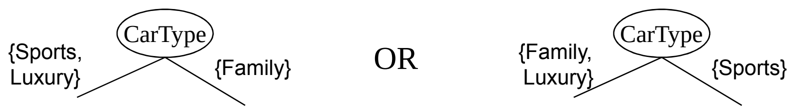
- Depends on number of ways to split
  - 2-way split
  - Multi-way split

Splitting Based on Nominal Attributes

- Multi-way split: Use as many partitions as distinct values.

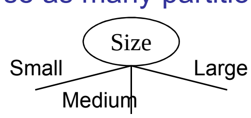


- Binary split: Divides values into two subsets.
  Need to find optimal partitioning.

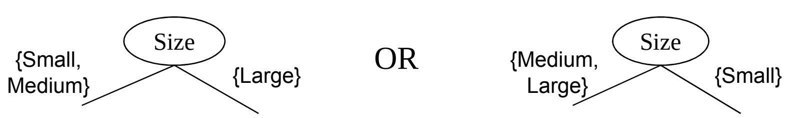


Splitting Based on Ordinal Attributes

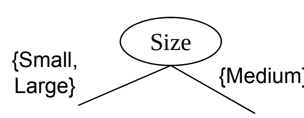
- Multi-way split: Use as many partitions as distinct values.



- Binary split: Divides values into two subsets.
  Need to find optimal partitioning.



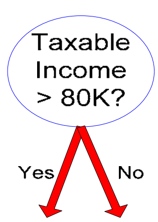
- What about this split?



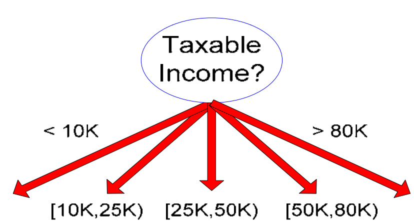
Splitting Based on Continuous Attributes

- Different ways of handling
  - Discretization to form an ordinal categorical attribute
    - Static – discretize once at the beginning
    - Dynamic – ranges can be found by equal interval bucketing, equal frequency bucketing (percentiles), or clustering.
  - Binary Decision: $(A < v)$ or $(A \geq v)$
    - consider all possible splits and finds the best cut
    - can be more compute intensive

Splitting Based on Continuous Attributes



(i) Binary split



(ii) Multi-way split

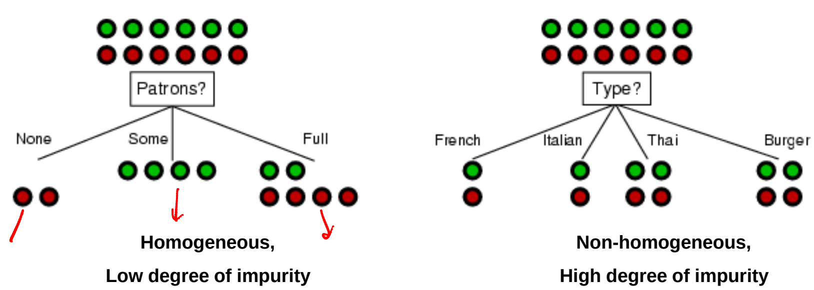
Tree Induction

- Greedy strategy.
  - Split the records based on an attribute test that optimizes certain criterion.

- Issues
  - Determine how to split the records
    - How to specify the attribute test condition?
    - How to determine the best split?
  - Determine when to stop splitting

How to determine the Best Split

- Idea: a good attribute splits the examples into subsets that are (ideally) "all positive" or "all negative"



- Greedy approach:
  - Nodes with homogeneous class distribution are preferred
- Need a measure of node impurity:

How to compare attribute?

- Entropy
  - Entropy *H(X)* of a random variable *X*

$$\boxed{H(X) = -\sum_{i=1}^{N} P(x=i)\log_2 P(x=i)}$$

  - *H(X)* is the expected number of bits needed to encode a randomly drawn value of *X* (under most efficient code)
  - Why?

    Information theory:

    Most efficient code assigns $-\log_2 P(X=i)$ bits to encode the message *X=I*, So, expected number of bits to code one random *X* is:

$$-\sum_{i=1}^{N} P(x=i)\log_2 P(x=i)$$

How to compare attribute?

- Conditional Entropy
  - Specific conditional entropy *H(X|Y=v)* of *X* given *Y=v* :

$$H(X|y=j) = -\sum_{i=1}^{N} P(x=i|y=j)\log_2 P(x=i|y=j)$$

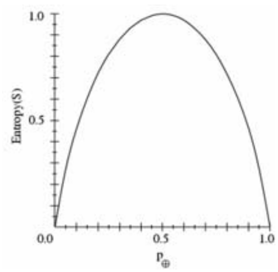
  - Conditional entropy *H(X|Y)* of *X* given *Y* :

$$H(X|Y) = -\sum_{j\in Val(y)} P(y=j)\log_2 H(X|y=j)$$

  - Mututal information (aka information gain) of *X* and *Y* :

$$\begin{aligned} I(X;Y) &= H(X) - H(X|Y) = H(Y) - H(Y|X) \\ &= H(X) + H(Y) - H(X,Y) \end{aligned}$$

Sample Entropy



- S is a sample of training examples
- p_+ is the proportion of positive examples in S
- p_- is the proportion of negative examples in S
- Entropy measure the impurity of S

$$H(S) \equiv -p_+ \log_2 p_+ - p_- \log_2 p_-$$

Examples for computing Entropy

$$\boxed{H(X) = -\sum_{i=1}^{N} P(x=i) \log_2 P(x=i)}$$

C1	**0**
C2	**6**

P(C1) = 0/6 = 0   P(C2) = 6/6 = 1

Entropy = – 0 log 0 – 1 log 1 = – 0 – 0 = 0

C1	**1**
C2	**5**

P(C1) = 1/6   P(C2) = 5/6

Entropy = – (1/6) $\log_2$ (1/6) – (5/6) $\log_2$ (1/6) = 0.65

C1	**2**
C2	**4**

P(C1) = 2/6   P(C2) = 4/6

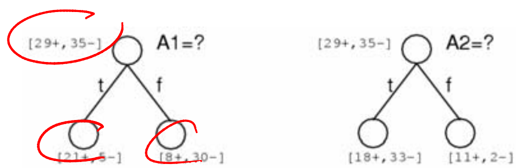
Entropy = – (2/6) $\log_2$ (2/6) – (4/6) $\log_2$ (4/6) = 0.92

Information Gain

- Information Gain:

$$GAIN_{split} = Entropy(p) - \left(\sum_{i=1}^{k} \frac{n_i}{n} Entropy(i) \right)$$

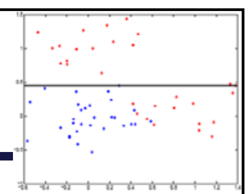
Parent Node, p is split into k partitions; n_i is number of records in partition i



Gain(S,A) = mutual information between A and target class variable over sample S

- Measures Reduction in Entropy achieved because of the split. Choose the split that achieves most reduction (maximizes GAIN)
- Used in ID3 and C4.5
- Disadvantage: Tends to prefer splits that result in large #of partitions, each being small but pure.

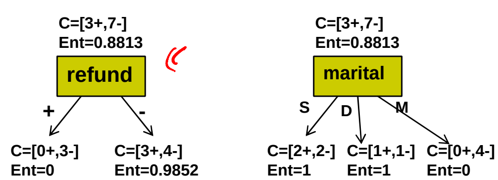
Exercise



categorical categorical continuous class

Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

Training Data



Gain(C,Refund)
=.8813-(.3x0+.7x.9852)
= .1906

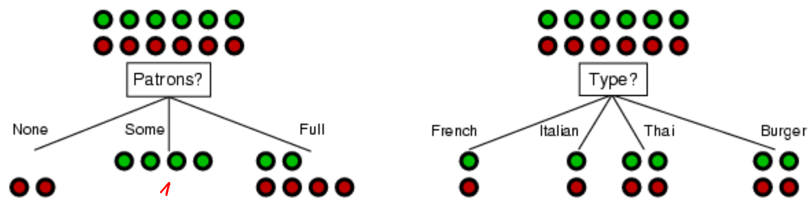
Gain(C,M)
=.8813-(.4x1+.2x1+.4x0)
= .2813

Which one should be at the root?

- **Choose the best classifier!**

Stopping Criteria for Tree Induction

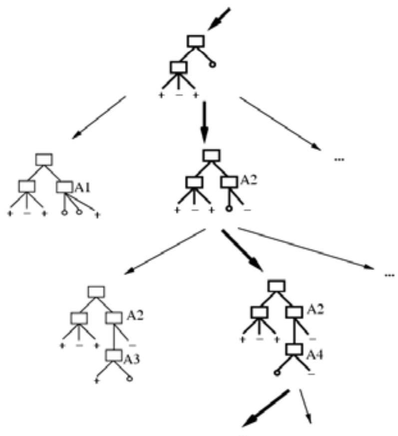
- Stop expanding a node when all the records belong to the same class
- Stop expanding a node when all the records have similar attribute values
- Early termination (to be discussed later)



Decision Tree Based Classification

- Advantages:
  - Inexpensive to construct
  - Extremely fast at classifying unknown records
  - Easy to interpret for small-sized trees
  - Accuracy is comparable to other classification techniques for many simple data sets
- Example: C4.5
  - Simple depth-first construction.
  - Uses Information Gain
  - Sorts Continuous Attributes at each node.
  - Needs entire data to fit in memory.
  - Unsuitable for Large Datasets.
    - Needs out-of-core sorting.
  - You can download the software from: http://www.cse.unsw.edu.au/~quinlan/c4.5r8.tar.gz

Which Tree Should We Output?



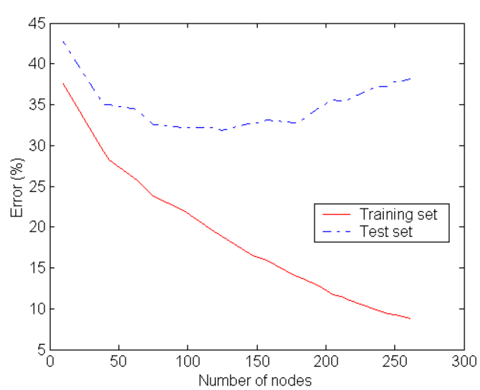
- ID3 performs heuristic search through space of decision trees
- It stops at smallest acceptable tree. Why?

Occam's razor: prefer the simplest hypothesis that fits the data

Practical Issues of DT

- Underfitting and Overfitting
- Missing Values

Will be covered in recitation!



Summary: what you should know:

- **Machine Learning is Cool and Useful!!**
  - Paradigms of Machine Learning.
  - Design elements learning
  - Theories on learning
- Well posed function approximation problems:
  - Instance space, X
  - Sample of labeled training data { <x_i, y_i>}
  - Hypothesis space, H = { f: X→Y }
- Learning is a search/optimization problem over H
  - Various objective functions
    - minimize training error (0-1 loss)
    - among hypotheses that minimize training error, select smallest (?)
- Decision tree learning
  - Greedy top-down learning of decision trees (ID3, C4.5, ...)
  - Overfitting and tree/rule post-pruning
  - Extensions…

Questions to think about (1)

- ID3 and C4.5 are heuristic algorithms that search through the space of decision trees. Why not just do an exhaustive search?

Questions to think about (2)

- Consider target function f: <x1,x2> → y, where x1 and x2 are real-valued, y is boolean. What is the set of decision surfaces describable with decision trees that use each attribute at most once?

Questions to think about (3)

- Why use Information Gain to select attributes in decision trees? What other criteria seem reasonable, and what are the tradeoffs in making this choice?

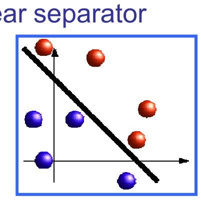
Additional material:

Learning non-linear functions

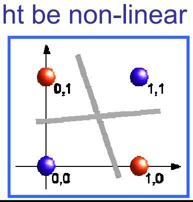
f: X → Y

- X (vector of) continuous and/or discrete vars
- Y discrete vars

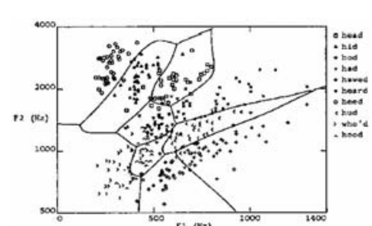
- Linear separator



- f might be non-linear function



The XOR gate



Speech recognition

Hypothesis spaces

How many distinct decision trees with *n* Boolean attributes?

= number of Boolean functions

= number of distinct truth tables with 2^n rows = 2^{2^n}

- E.g., with 6 Boolean attributes, there are 18,446,744,073,709,551,616 trees

Notes on Overfitting

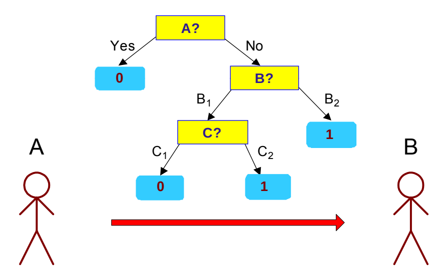
- Overfitting results in decision trees that are more complex than necessary
- Training error no longer provides a good estimate of how well the tree will perform on previously unseen records
- **Which Tree Should We Output?**
  - Occam's razor: prefer the simplest hypothesis that fits the data

Occam's Razor

- Given two models of similar generalization errors, one should prefer the simpler model over the more complex model
- For complex models, there is a greater chance that it was fitted accidentally by errors in data
- Therefore, one should include model complexity when evaluating a model

Minimum Description Length (MDL)

X	y
X_1	1
X_2	0
X_3	0
X_4	1
…	…
X_n	1



X	y
X_1	?
X_2	?
X_3	?
X_4	?
…	…
X_n	?

- Cost(Model,Data) = Cost(Data|Model) + Cost(Model)
  - Cost is the number of bits needed for encoding.
  - Search for the least costly model.
- Cost(Data|Model) encodes the misclassification errors.
- Cost(Model) uses node encoding (number of children) plus splitting condition encoding.

How to Address Overfitting

- Pre-Pruning (Early Stopping Rule)
  - Stop the algorithm before it becomes a fully-grown tree
  - Typical stopping conditions for a node:
    - Stop if all instances belong to the same class
    - Stop if all the attribute values are the same
  - More restrictive conditions:
    - Stop if number of instances is less than some user-specified threshold
    - Stop if class distribution of instances are independent of the available features (e.g., using χ^2 test)
    - Stop if expanding the current node does not improve impurity measures (e.g., Gini or information gain).

How to Address Overfitting…

- Post-pruning
  - Grow decision tree to its entirety
  - Trim the nodes of the decision tree in a bottom-up fashion
  - If generalization error improves after trimming, replace sub-tree by a leaf node.
  - Class label of leaf node is determined from majority class of instances in the sub-tree
  - Can use MDL for post-pruning

Handling Missing Attribute Values

- Missing values affect decision tree construction in three different ways:
  - Affects how impurity measures are computed
  - Affects how to distribute instance with missing value to child nodes
  - Affects how a test instance with missing value is classified

Computing Impurity Measure

Tid	Refund	Marital Status	Taxable Income	Class
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	?	Single	90K	Yes

Missing value

Before Splitting:

Entropy(Parent)
= -0.3 log(0.3)-(0.7)log(0.7) = 0.8813

	Class = Yes	Class = No
Refund=Yes	0	3
Refund=No	2	4
Refund=?	1	0

Split on Refund:

Entropy(Refund=Yes) = 0

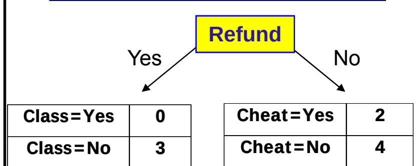
Entropy(Refund=No)
= -(2/6)log(2/6) – (4/6)log(4/6) = 0.9183

Entropy(Children)
= 0.3 (0) + 0.6 (0.9183) = 0.551

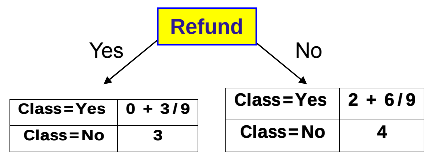
Gain = 0.9 × (0.8813 – 0.551) = 0.3303

Distribute Instances

Tid	Refund	Marital Status	Taxable Income	Class
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No



Tid	Refund	Marital Status	Taxable Income	Class
10	?	Single	90K	Yes



Probability that Refund=Yes is 3/9

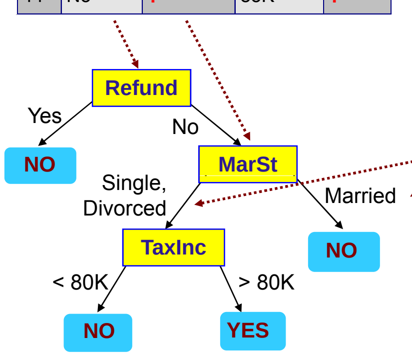
Probability that Refund=No is 6/9

Assign record to the left child with weight = 3/9 and to the right child with weight = 6/9

Classify Instances

New record:

Tid	Refund	Marital Status	Taxable Income	Class
11	No	?	85K	?



	Married	Single	Divorced	Total
Class=No	3	1	0	4
Class=Yes	6/9	1	1	2.67
Total	3.67	2	1	6.67

Probability that Marital Status = Married is 3.67/6.67

Probability that Marital Status ={Single,Divorced} is 3/6.67