Machine Learning-1

Byunggeor Moon

What is Machine Learning?

"Learning is any process by which a system improves performance from experience."

- Herbert Simon

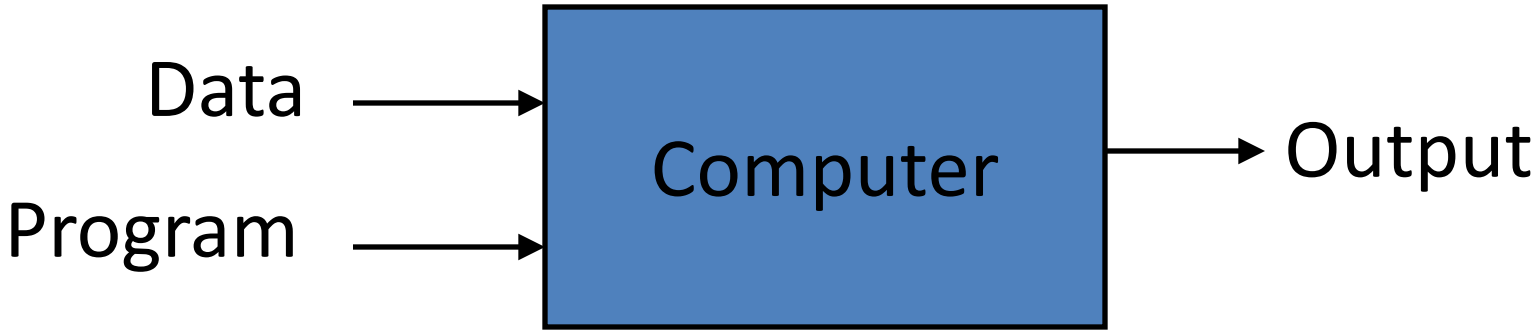
Definition by Tom Mitchell (1998):

Machine Learning is the study of algorithms that

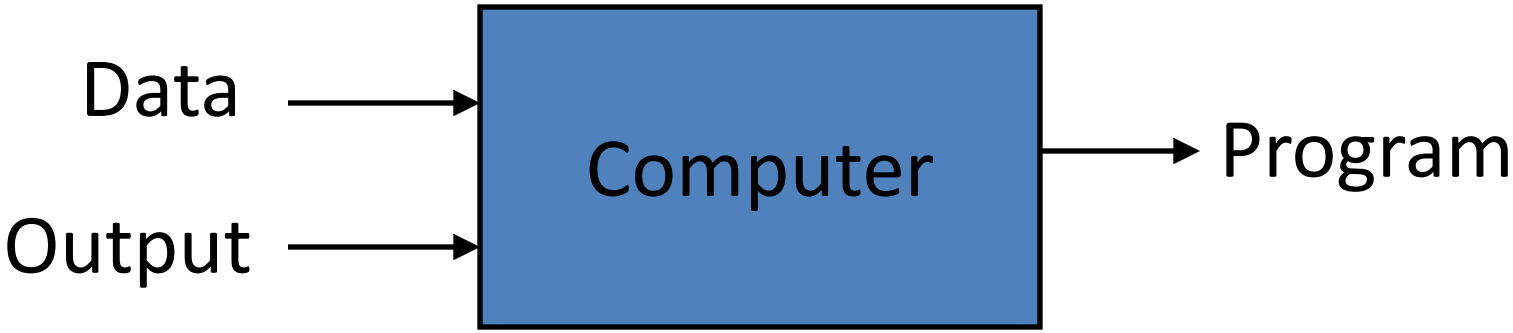
- improve their performance P
- at some task T
- with experience E.

A well-defined learning task is given by <P, T, E>.

Traditional Programming



Machine Learning

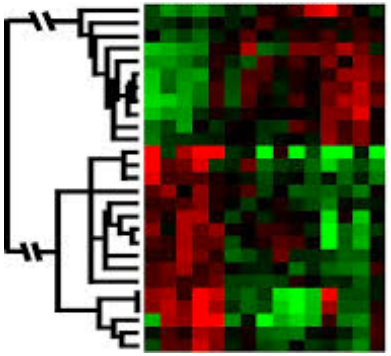


Slide credit: Pedro Domingos

When Do We Use Machine Learning?

ML is used when:

- Human expertise does not exist (navigating on Mars)
- Humans can't explain their expertise (speech recognition)
- Models must be customized (personalized medicine)
- Models are based on huge amounts of data (genomics)



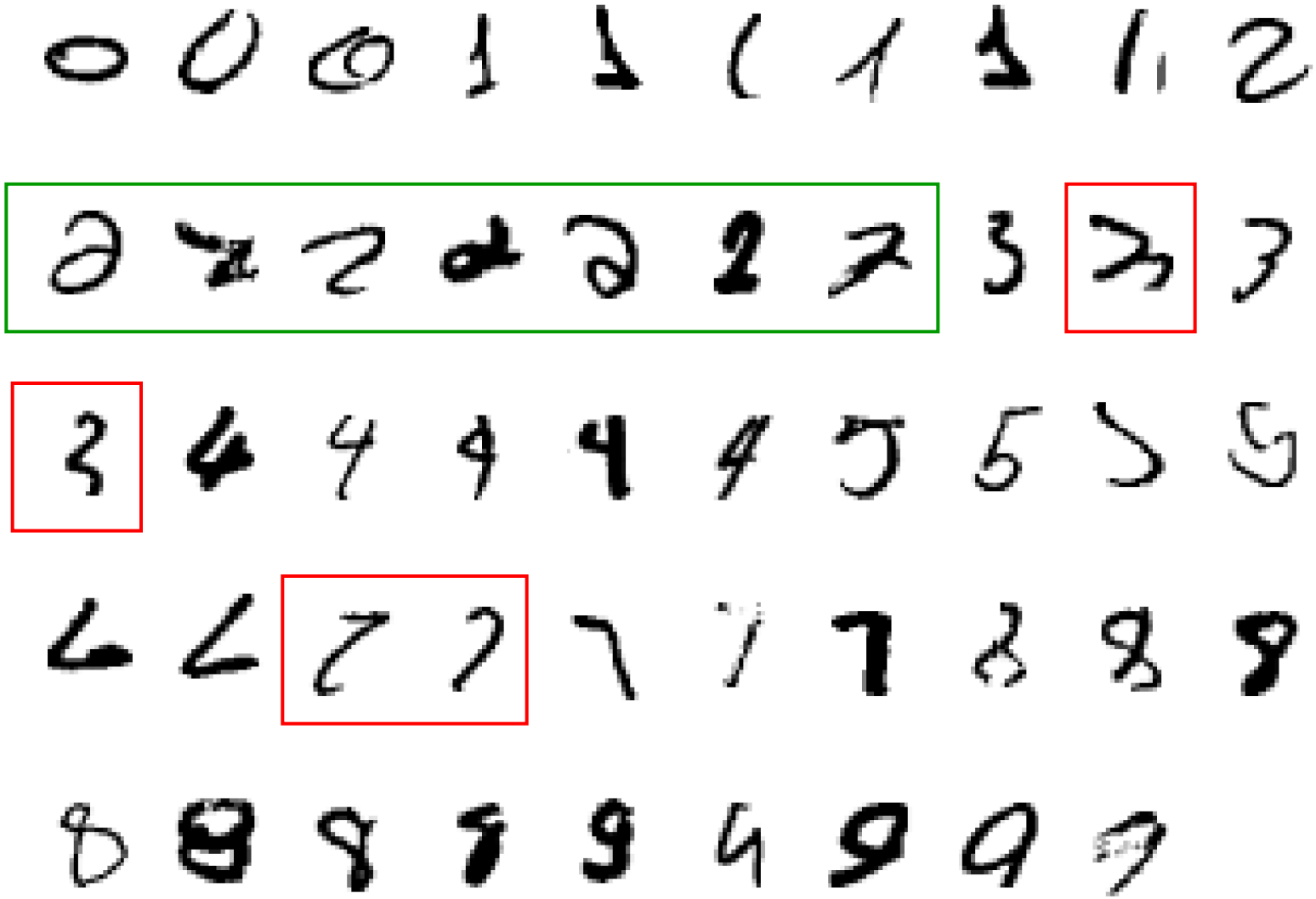
Learning isn't always useful:

- There is no need to "learn" to calculate payroll

Based on slide by E. Alpaydin

A classic example of a task that requires machine learning:

It is very hard to say what makes a 2



Slide credit: Geoffrey Hinton

Some more examples of tasks that are best solved by using a learning algorithm

- Recognizing patterns:
  - Facial identities or facial expressions
  - Handwritten or spoken words
  - Medical images
- Generating patterns:
  - Generating images or motion sequences
- Recognizing anomalies:
  - Unusual credit card transactions
  - Unusual patterns of sensor readings in a nuclear power plant
- Prediction:
  - Future stock prices or currency exchange rates

Slide credit: Geoffrey Hinton

Sample Applications

- Web search
- Computational biology
- Finance
- E-commerce
- Space exploration
- Robotics
- Information extraction
- Social networks
- Debugging software
- [Your favorite area]

Slide credit: Pedro Domingos

Samuel's Checkers-Player

"Machine Learning: Field of study that gives computers the ability to learn without being explicitly programmed." -Arthur Samuel (1959)

Defining the Learning Task

Improve on task T, with respect to
performance metric P, based on experience E

T: Playing checkers
P: Percentage of games won against an arbitrary opponent
E: Playing practice games against itself

T: Recognizing hand-written words
P: Percentage of words correctly classified
E: Database of human-labeled images of handwritten words

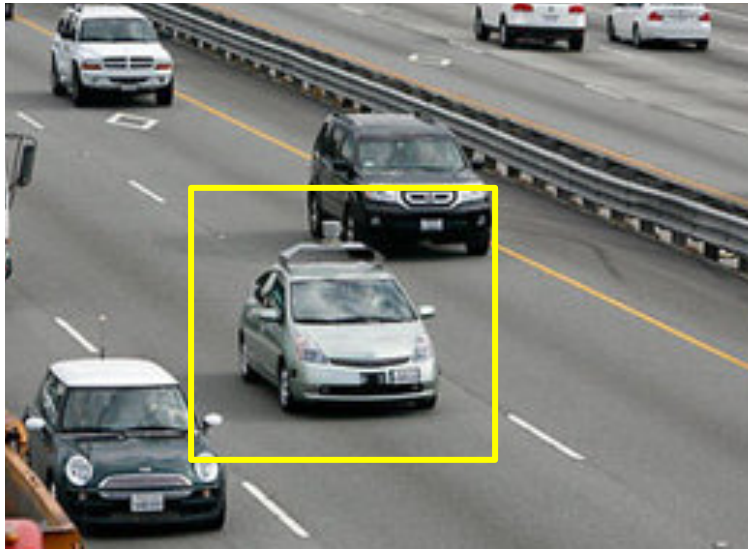
T: Driving on four-lane highways using vision sensors
P: Average distance traveled before a human-judged error
E: A sequence of images and steering commands recorded while observing a human driver.

T: Categorize email messages as spam or legitimate.
P: Percentage of email messages correctly classified.
E: Database of emails, some with human-given labels

Slide credit: Ray Mooney

State of the Art Applications of Machine Learning

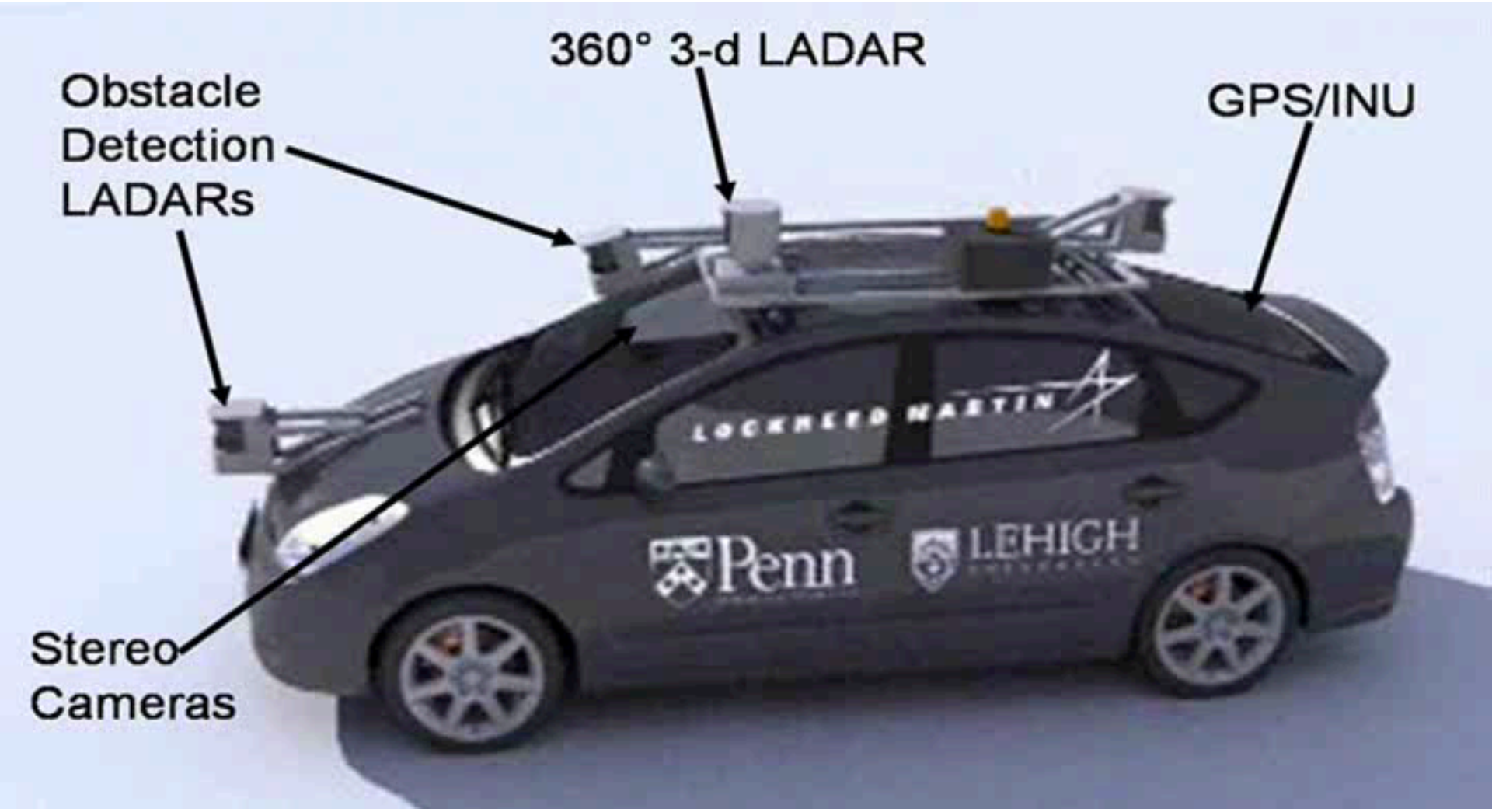
Autonomous Cars



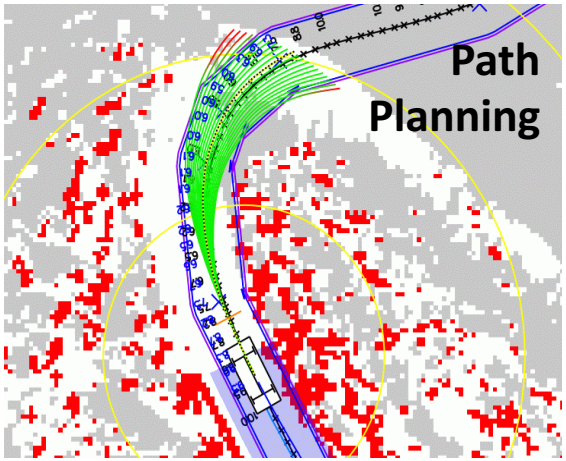
- Nevada made it legal for autonomous cars to drive on roads in June 2011
- As of 2013, four states (Nevada, Florida, California, and Michigan) have legalized autonomous cars

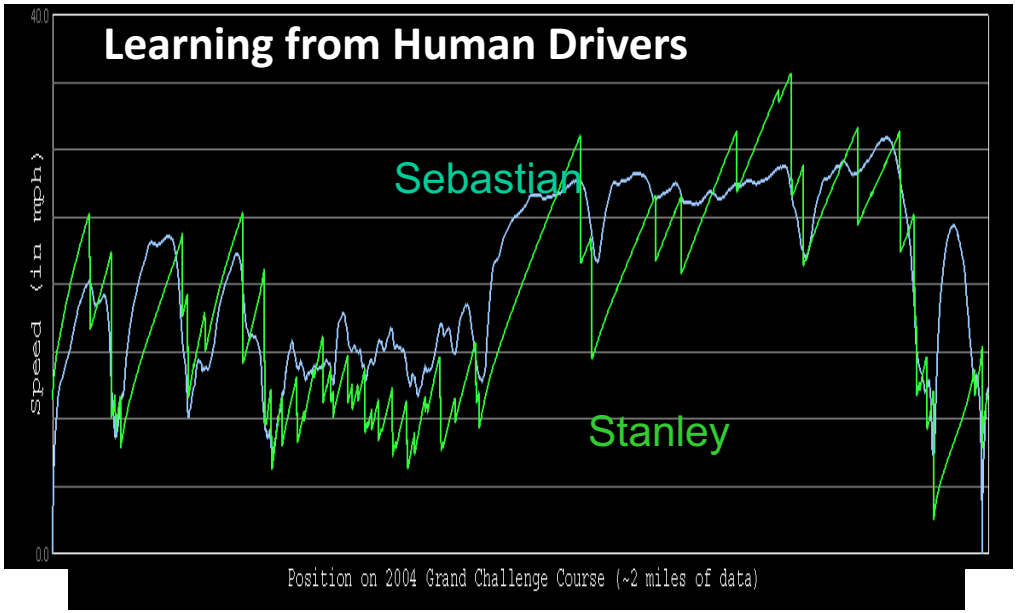
Penn's Autonomous Car →
(Ben Franklin Racing Team)

Autonomous Car Sensors



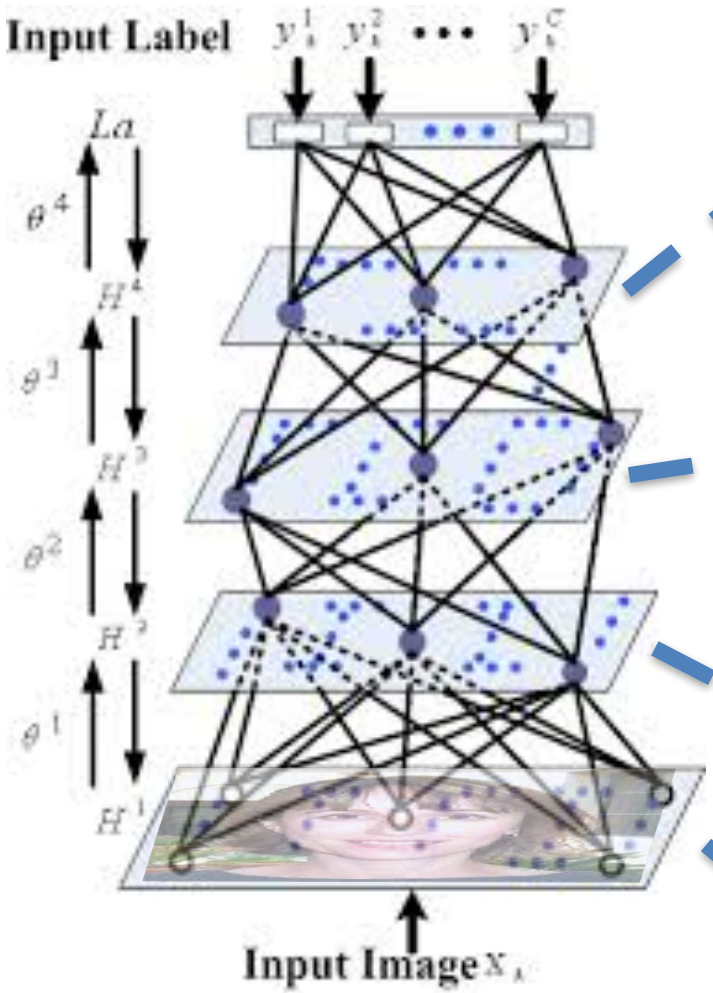
Autonomous Car Technology

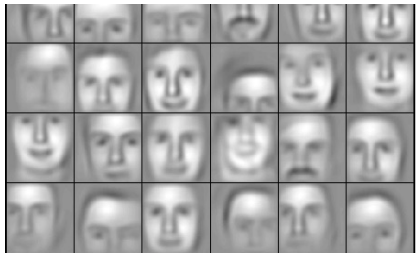




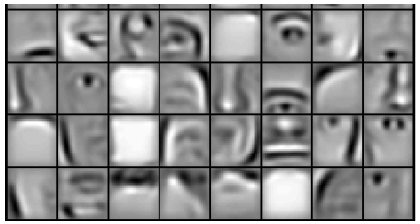
Images and movies taken from Sebastian Thrun's multimedia website.

Deep Belief Net on Face Images

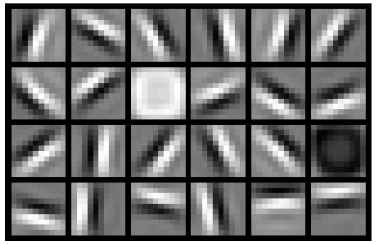




object models



object parts (combination of edges)

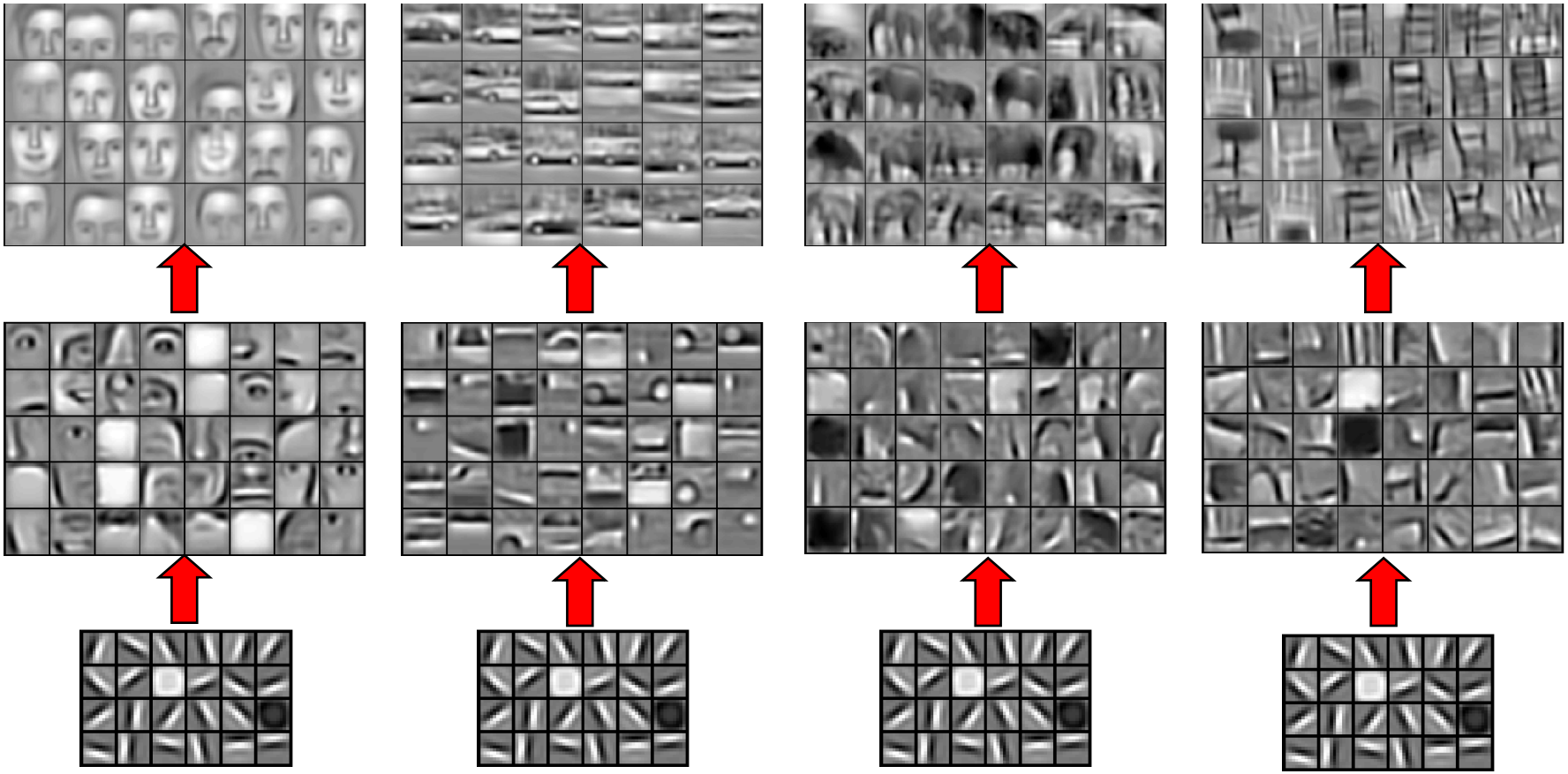


edges

pixels

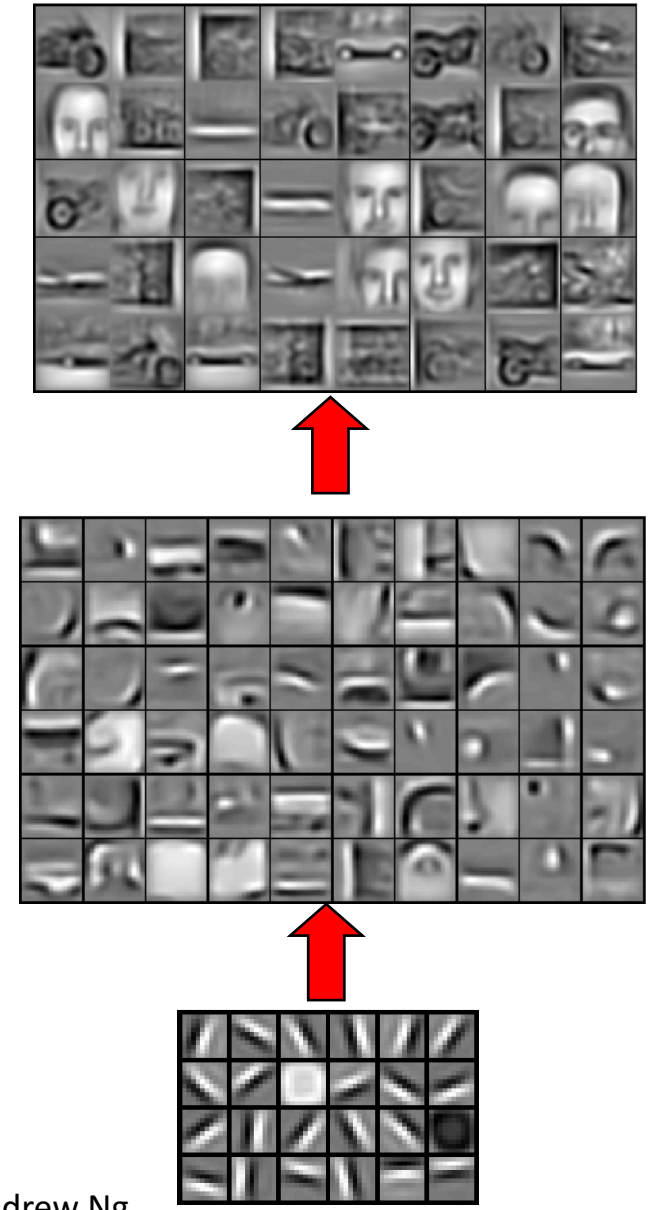
Based on materials by Andrew Ng

Learning of Object Parts



Slide credit: Andrew Ng

Training on Multiple Objects



Trained on 4 classes (cars, faces, motorbikes, airplanes).

Second layer: Shared-features and object-specific features.

Third layer: More specific features.

Slide credit: Andrew Ng

Scene Labeling via Deep Learning

[Farabet et al. ICML 2012, PAMI 2013]

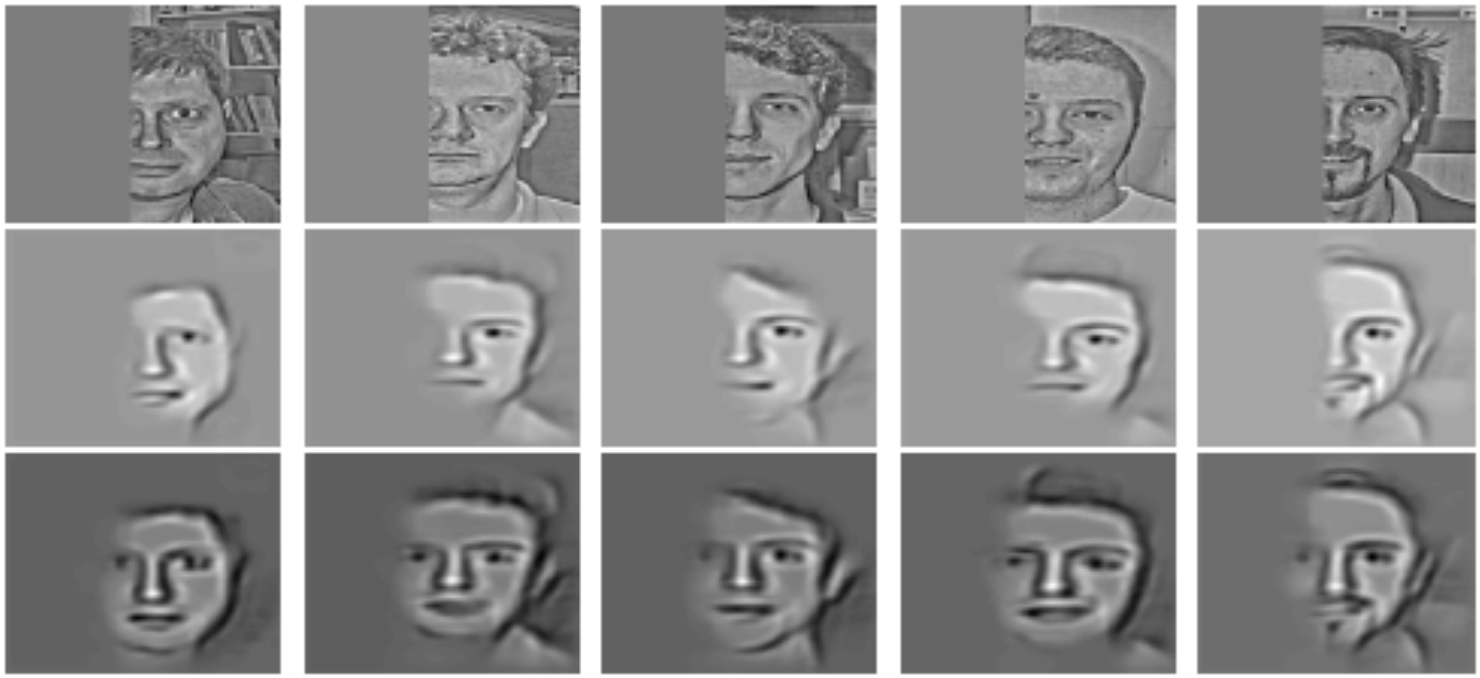
Inference from Deep Learned Models

Generating posterior samples from faces by "filling in" experiments (cf. Lee and Mumford, 2003). Combine bottom-up and top-down inference.

Input images

Samples from feedforward Inference (control)

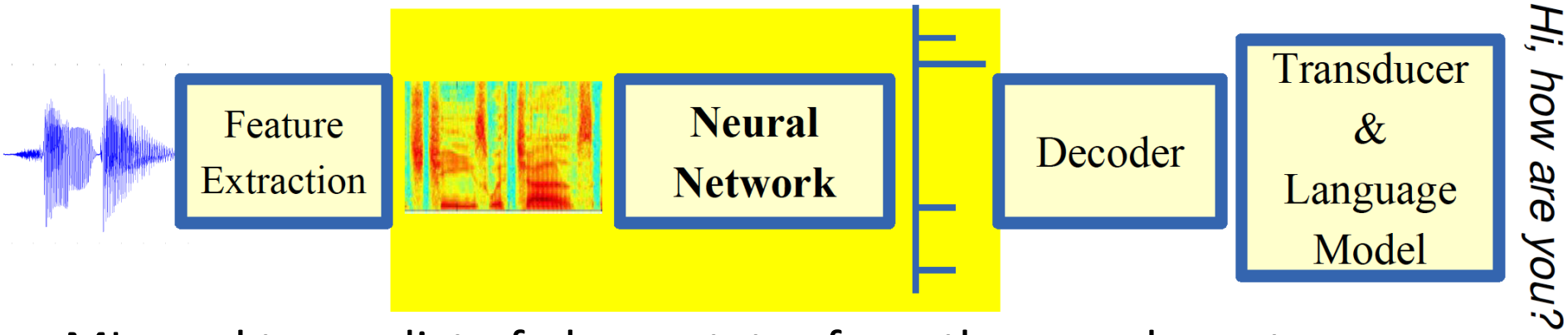
Samples from Full posterior inference



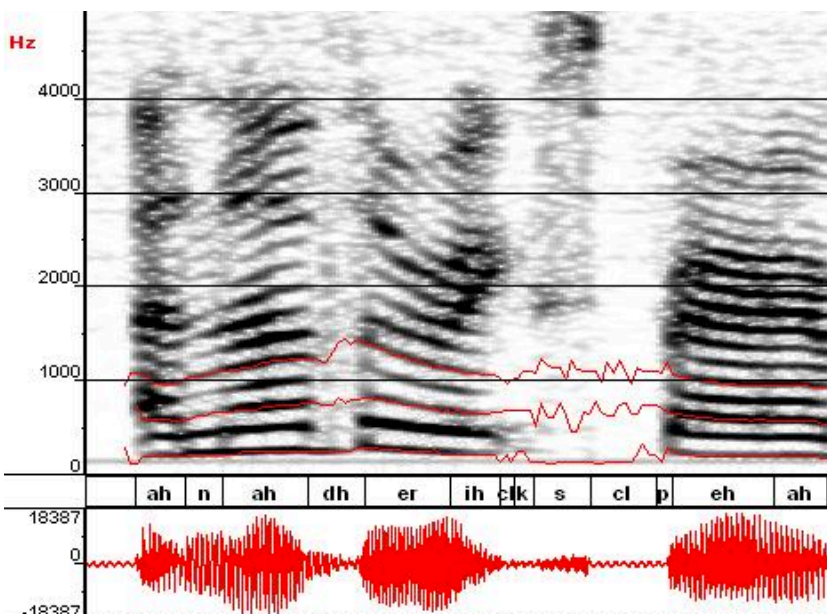
Slide credit: Andrew Ng

Machine Learning in Automatic Speech Recognition

A Typical Speech Recognition System



ML used to predict of phone states from the sound spectrogram



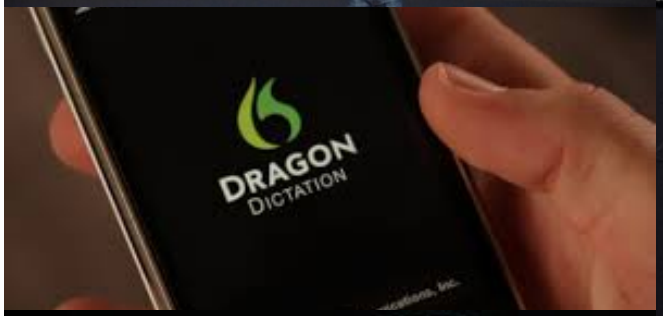
Deep learning has state-of-the-art results

# Hidden Layers	1	2	4	8	10	12
Word Error Rate %	16.0	12.8	11.4	10.9	11.0	11.1

Baseline GMM performance = 15.4%

[Zeiler et al. "On rectified linear units for speech recognition" ICASSP 2013]

Impact of Deep Learning in Speech Technology



Slide credit: Li Deng, MS Research

Types of Learning

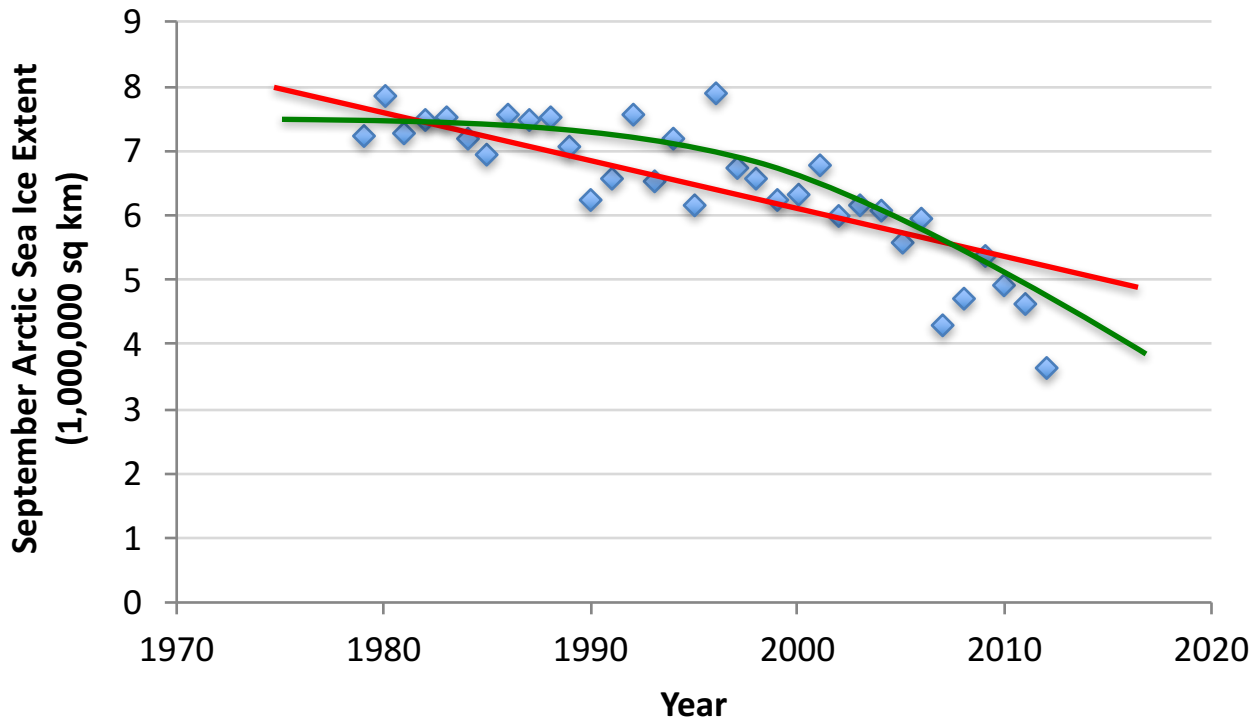
Types of Learning

- **Supervised (inductive) learning**
  - Given: training data + desired outputs (labels)
- **Unsupervised learning**
  - Given: training data (without desired outputs)
- **Semi-supervised learning**
  - Given: training data + a few desired outputs
- **Reinforcement learning**
  - Rewards from sequence of actions

Based on slide by Pedro Domingos

Supervised Learning: Regression

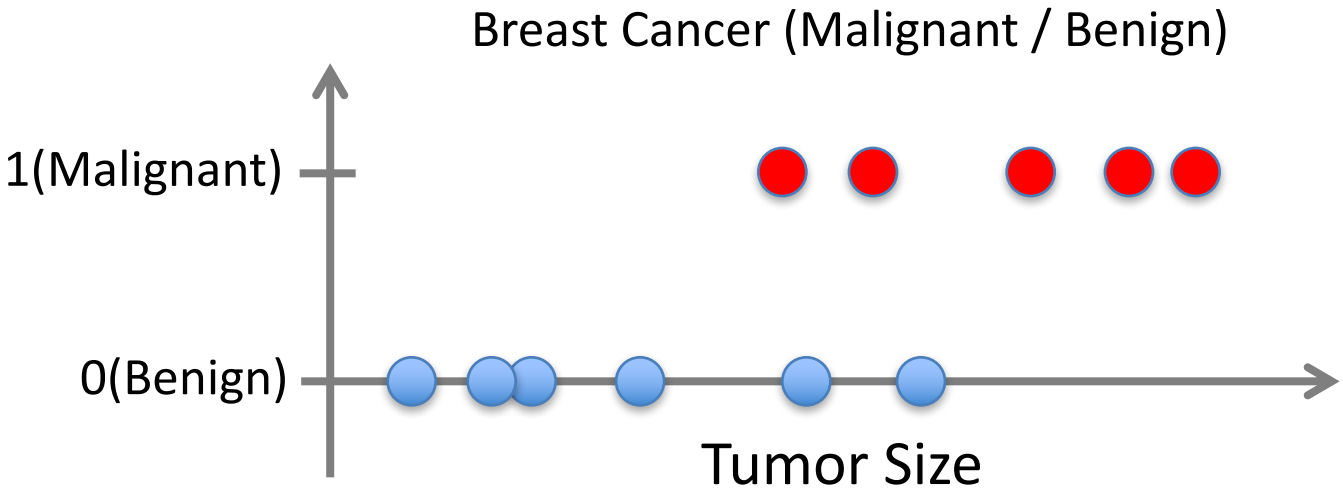
- Given $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$
- Learn a function $f(x)$ to predict y given x
  - y is real-valued == regression



Data from G. Witt. Journal of Statistics Education, Volume 21, Number 1 (2013)

Supervised Learning: Classification

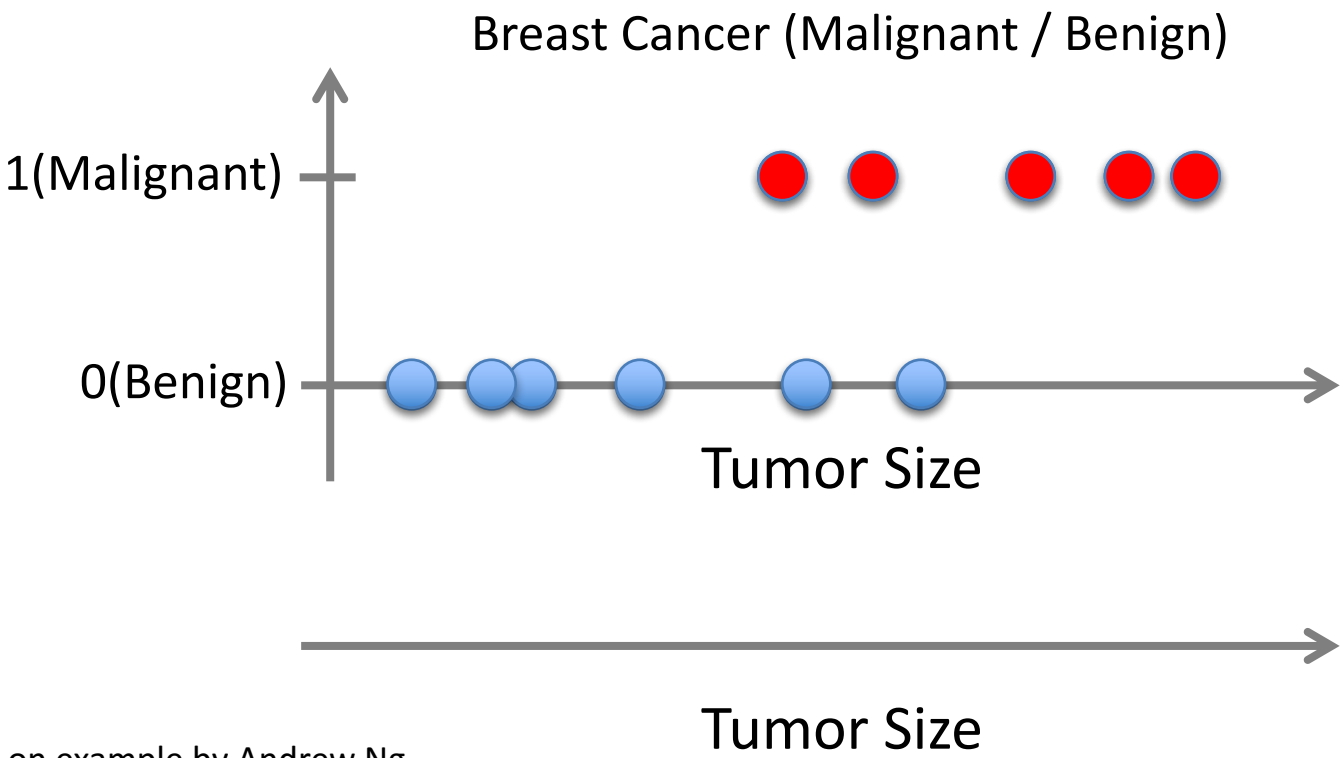
- Given $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$
- Learn a function $f(x)$ to predict y given x
  - y is categorical == classification



Based on example by Andrew Ng

Supervised Learning: Classification

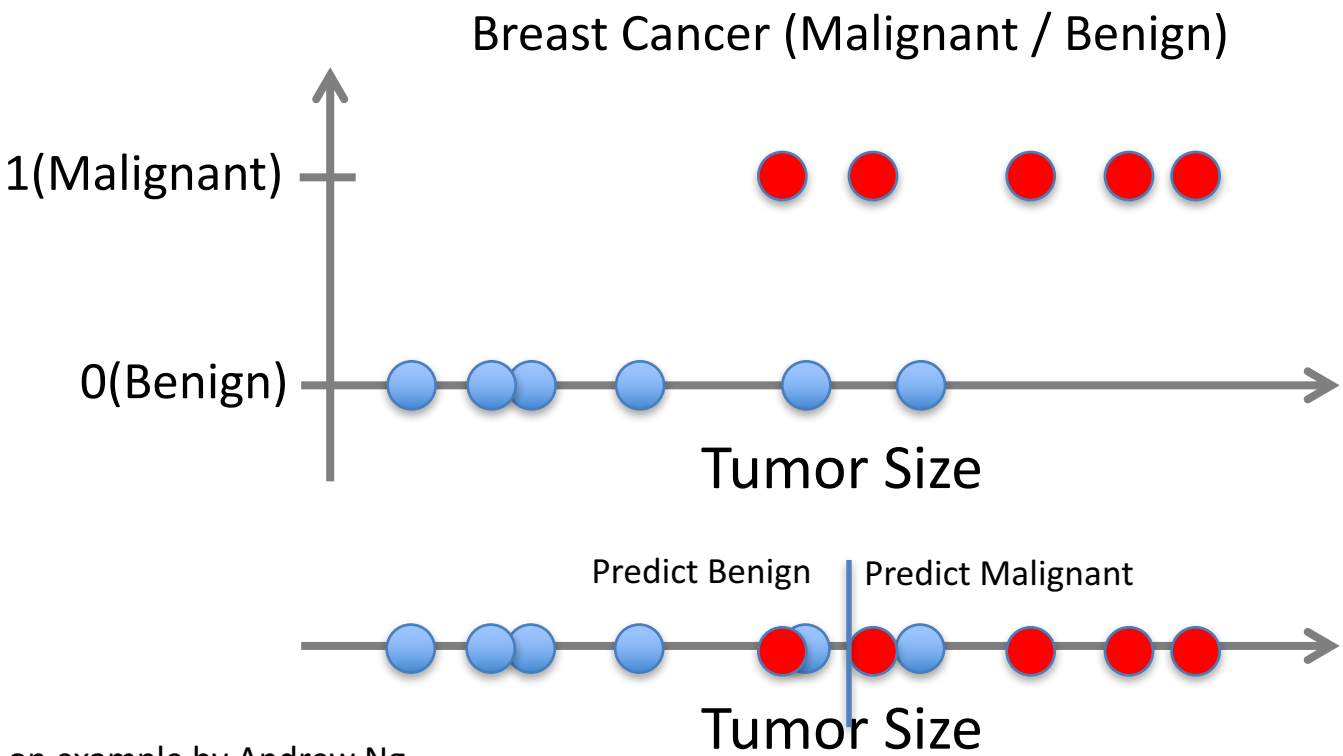
- Given $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$
- Learn a function $f(x)$ to predict y given x
  - y is categorical == classification



Based on example by Andrew Ng

Supervised Learning: Classification

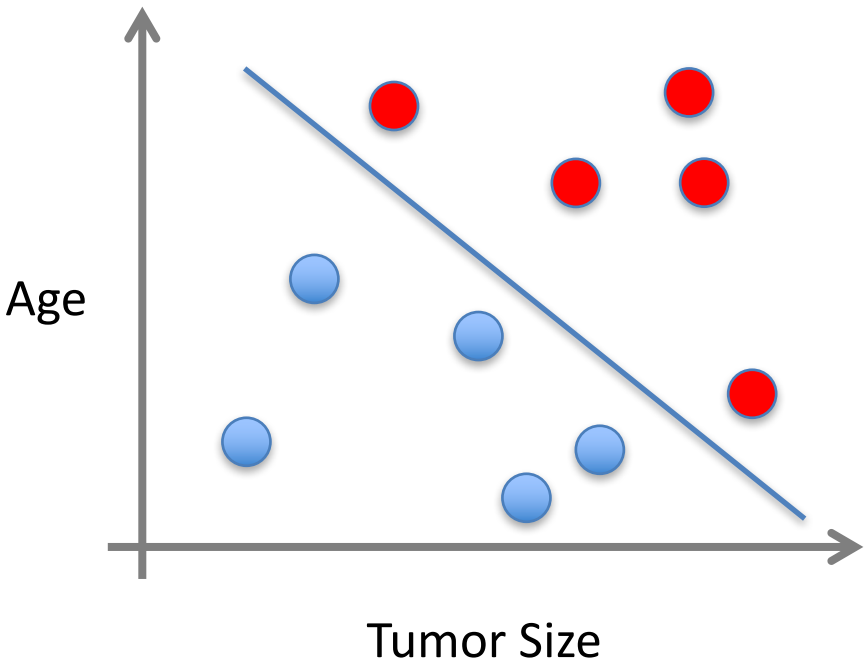
- Given $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$
- Learn a function $f(x)$ to predict y given x
  - y is categorical == classification



Based on example by Andrew Ng

Supervised Learning

- x can be multi-dimensional
  - Each dimension corresponds to an attribute



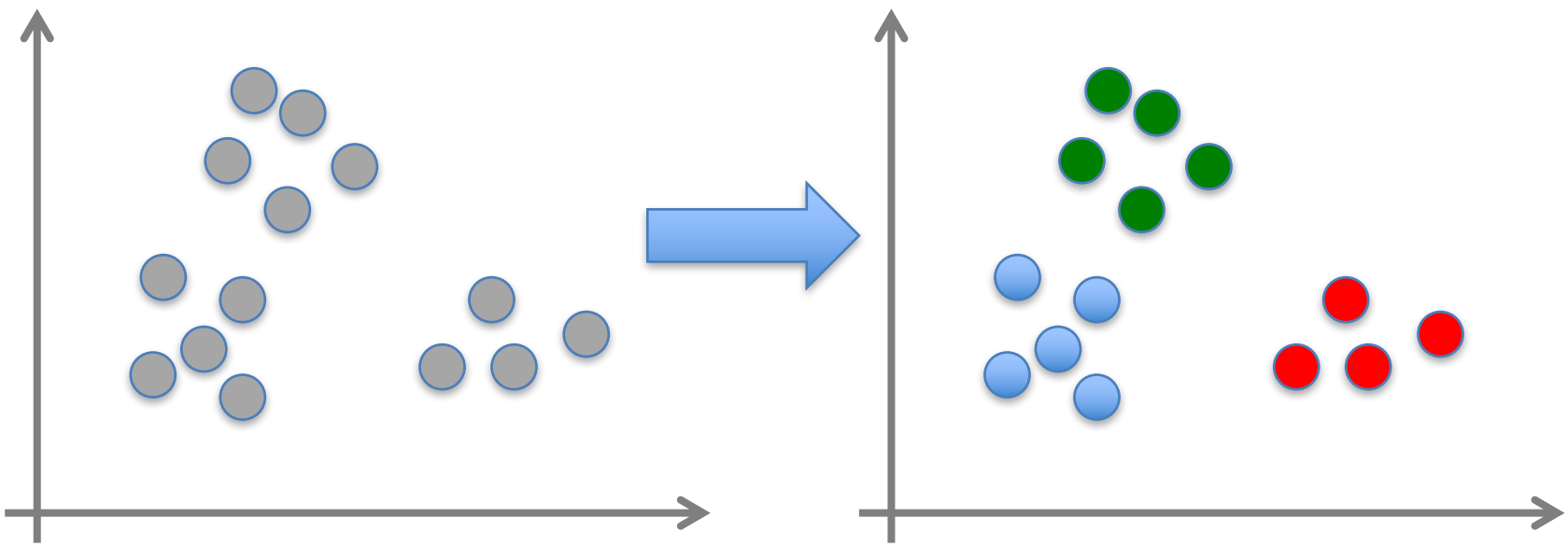
- Clump Thickness
- Uniformity of Cell Size
- Uniformity of Cell Shape

…

Based on example by Andrew Ng

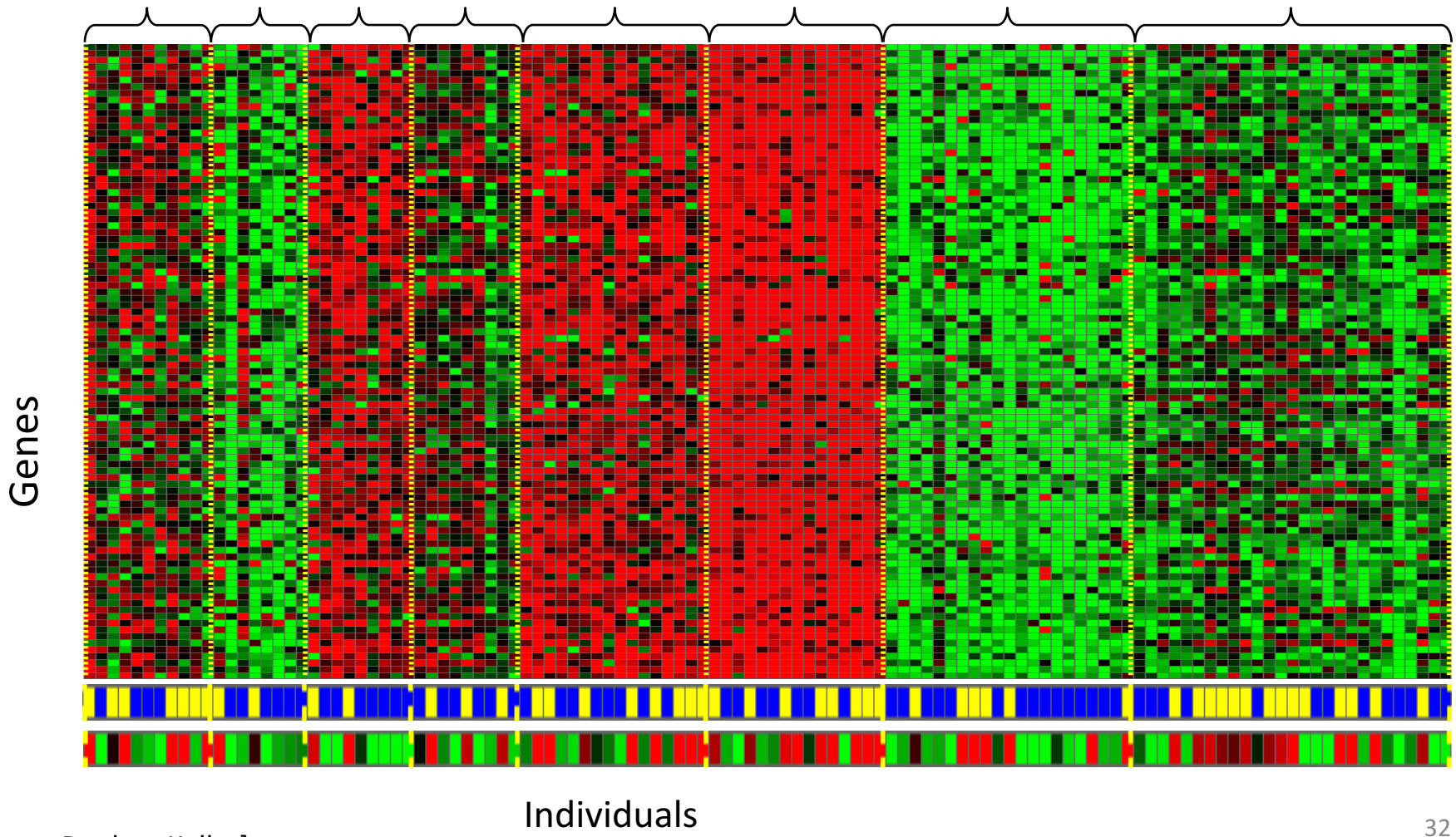
Unsupervised Learning

- Given $x_1, x_2, \ldots, x_n$ (without labels)
- Output hidden structure behind the x's
  - E.g., clustering



Unsupervised Learning

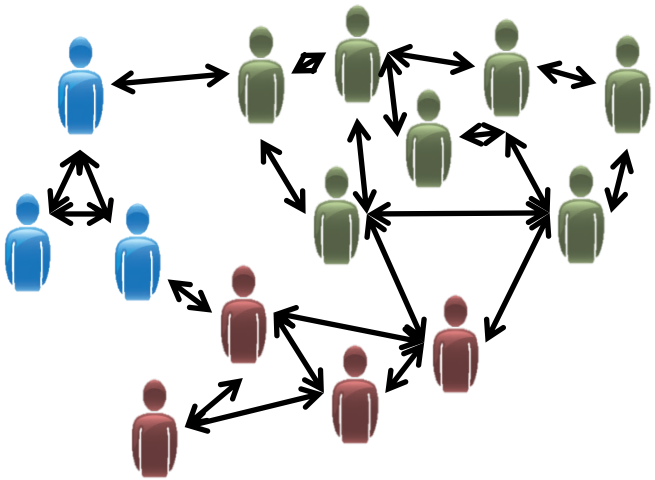
Genomics application: group individuals by genetic similarity



[Source: Daphne Koller]

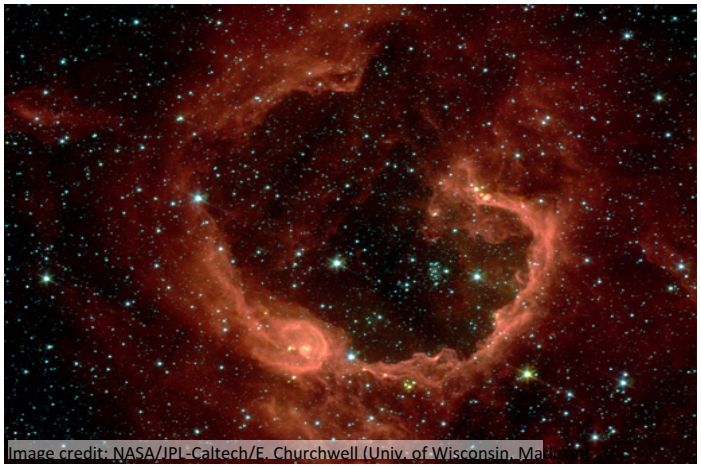
Unsupervised Learning

Organize computing clusters



Social network analysis

Market segmentation



Astronomical data analysis

Slide credit: Andrew Ng

Unsupervised Learning

- Independent component analysis – separate a combined signal into its original sources

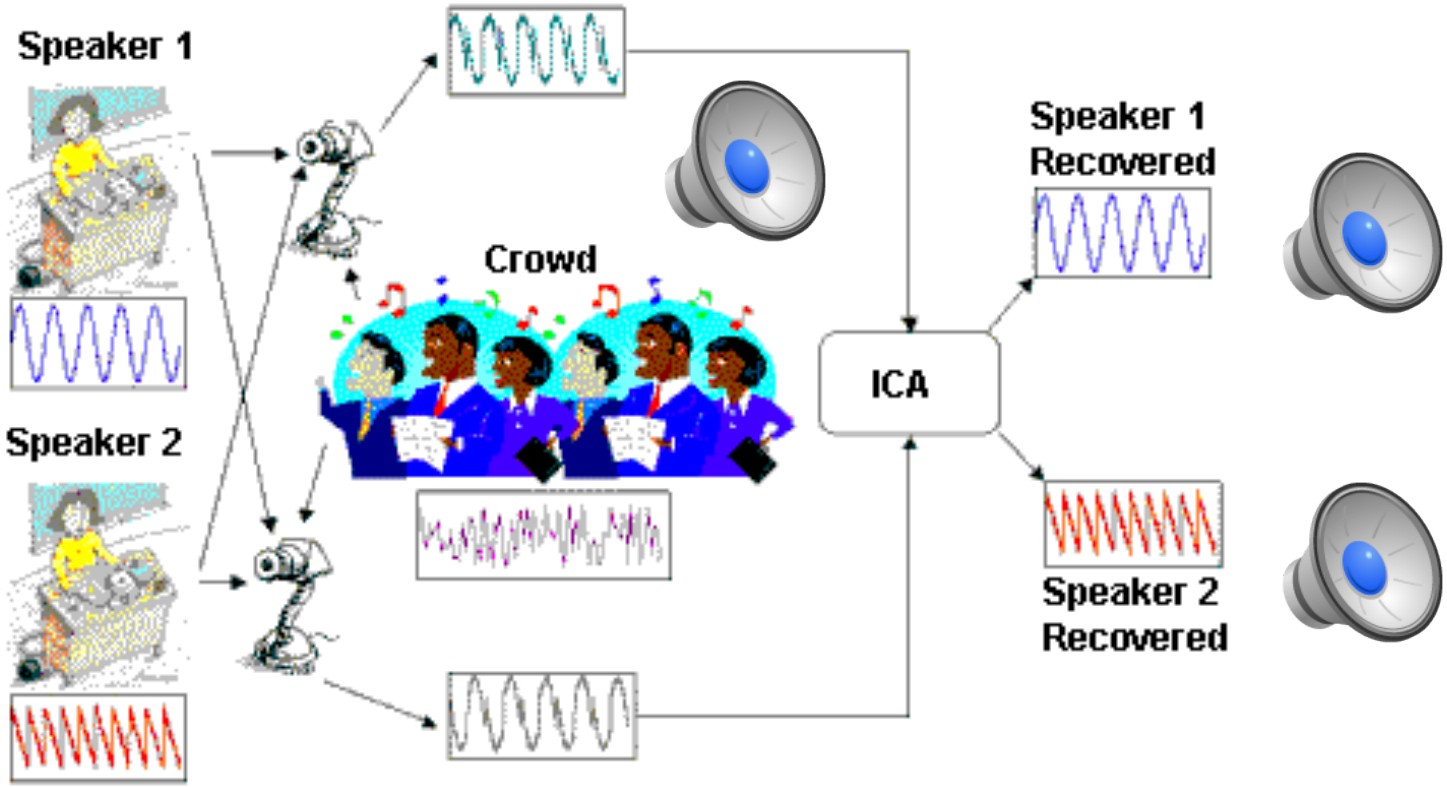
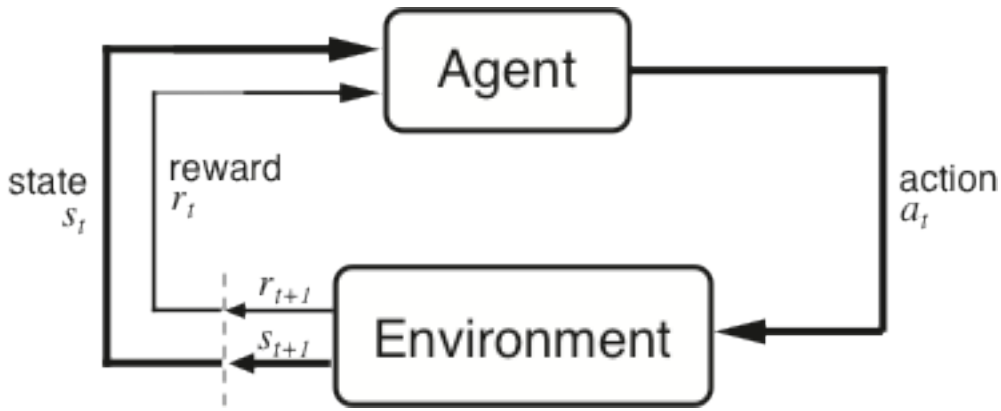


Image credit: statsoft.com Audio from http://www.ism.ac.jp/~shiro/research/blindsep.html

Reinforcement Learning

- Given a sequence of states and actions with (delayed) rewards, output a policy
  - Policy is a mapping from states → actions that tells you what to do in a given state
- Examples:
  - Credit assignment problem
  - Game playing
  - Robot in a maze
  - Balance a pole on your hand

The Agent-Environment Interface



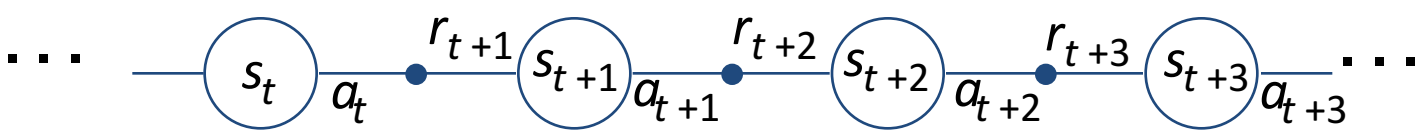
Agent and environment interact at discrete time steps : $t = 0, 1, 2, \ldots$

Agent observes state at step t: $s_t \in S$

produces action at step t: $a_t \in A(s_t)$

gets resulting reward : $r_{t+1} \in \Re$

and resulting next state : s_{t+1}



Slide credit: Sutton & Barto

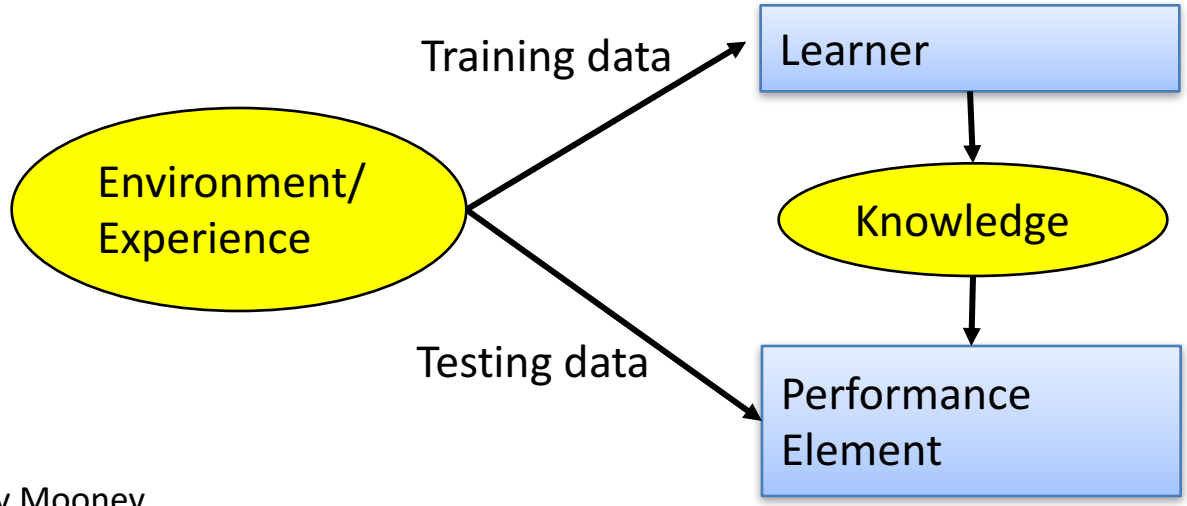
Reinforcement Learning

https://www.youtube.com/watch?v=4cgWya-wjgY

Framing a Learning Problem

Designing a Learning System

- Choose the training experience
- Choose exactly what is to be learned
  - i.e. the ***target function***
- Choose how to represent the target function
- Choose a learning algorithm to infer the target function from the experience



Based on slide by Ray Mooney

ML in a Nutshell

- Tens of thousands of machine learning algorithms
  - Hundreds new every year

- Every ML algorithm has three components:
  - **Representation**
  - **Optimization**
  - **Evaluation**

Slide credit: Pedro Domingos

Various Function Representations

- Numerical functions
  - Linear regression
  - Neural networks
  - Support vector machines
- Symbolic functions
  - Decision trees
  - Rules in propositional logic
  - Rules in first-order predicate logic
- Instance-based functions
  - Nearest-neighbor
  - Case-based
- Probabilistic Graphical Models
  - Naïve Bayes
  - Bayesian networks
  - Hidden-Markov Models (HMMs)
  - Probabilistic Context Free Grammars (PCFGs)
  - Markov networks

Slide credit: Ray Mooney

Various Search/Optimization Algorithms

- Gradient descent
  - Perceptron
  - Backpropagation
- Dynamic Programming
  - HMM Learning
  - PCFG Learning
- Divide and Conquer
  - Decision tree induction
  - Rule learning
- Evolutionary Computation
  - Genetic Algorithms (GAs)
  - Genetic Programming (GP)
  - Neuro-evolution

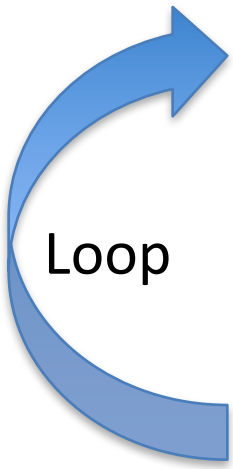
Slide credit: Ray Mooney

Evaluation

- Accuracy
- Precision and recall
- Squared error
- Likelihood
- Posterior probability
- Cost / Utility
- Margin
- Entropy
- K-L divergence
- etc.

Slide credit: Pedro Domingos

ML in Practice



- Understand domain, prior knowledge, and goals
- Data integration, selection, cleaning, pre-processing, etc.
- Learn models
- Interpret results
- Consolidate and deploy discovered knowledge

Based on a slide by Pedro Domingos

Lessons Learned about Learning

- Learning can be viewed as using direct or indirect experience to approximate a chosen target function.
- Function approximation can be viewed as a search through a space of hypotheses (representations of functions) for one that best fits a set of training data.
- Different learning methods assume different hypothesis spaces (representation languages) and/or employ different search techniques.

Slide credit: Ray Mooney

- **Supervised learning**
  - Decision tree induction
  - Linear regression
  - Logistic regression
  - Support vector machines & kernel methods
  - Model ensembles
  - Bayesian learning
  - Neural networks & deep learning
  - Learning theory
- **Unsupervised learning**
  - Clustering
  - Dimensionality reduction
- **Reinforcement learning**
  - Temporal difference learning
  - Q learning
- **Evaluation**
- **Applications**

A Brief History of Machine Learning

History of Machine Learning

- 1950s
  - Samuel's checker player
  - Selfridge's Pandemonium
- 1960s:
  - Neural networks: Perceptron
  - Pattern recognition
  - Learning in the limit theory
  - Minsky and Papert prove limitations of Perceptron
- 1970s:
  - Symbolic concept induction
  - Winston's arch learner
  - Expert systems and the knowledge acquisition bottleneck
  - Quinlan's ID3
  - Michalski's AQ and soybean diagnosis
  - Scientific discovery with BACON
  - Mathematical discovery with AM

Slide credit: Ray Mooney

History of Machine Learning (cont.)

- 1980s:
  - Advanced decision tree and rule learning
  - Explanation-based Learning (EBL)
  - Learning and planning and problem solving
  - Utility problem
  - Analogy
  - Cognitive architectures
  - Resurgence of neural networks (connectionism, backpropagation)
  - Valiant's PAC Learning Theory
  - Focus on experimental methodology
- 1990s
  - Data mining
  - Adaptive software agents and web applications
  - Text learning
  - Reinforcement learning (RL)
  - Inductive Logic Programming (ILP)
  - Ensembles: Bagging, Boosting, and Stacking
  - Bayes Net learning

Slide credit: Ray Mooney

History of Machine Learning (cont.)

- 2000s
  - Support vector machines & kernel methods
  - Graphical models
  - Statistical relational learning
  - Transfer learning
  - Sequence labeling
  - Collective classification and structured outputs
  - Computer Systems Applications (Compilers, Debugging, Graphics, Security)
  - E-mail management
  - Personalized assistants that learn
  - Learning in robotics and vision
- 2010s
  - Deep learning systems
  - Learning for big data
  - Bayesian methods
  - Multi-task & lifelong learning
  - Applications to vision, speech, social networks, learning to read, etc.
  - ???

Based on slide by Ray Mooney

Decision Trees
-Classification and Regression

Function Approximation

Problem Setting

- Set of possible instances $\mathcal{X}$
- Set of possible labels $\mathcal{Y}$
- Unknown target function $f : \mathcal{X} \to \mathcal{Y}$
- Set of function hypotheses $H = \{h \mid h : \mathcal{X} \to \mathcal{Y}\}$

Input: Training examples of unknown target function f

$$\{\langle \boldsymbol{x}_i, y_i \rangle\}_{i=1}^{n} = \{\langle \boldsymbol{x}_1, y_1 \rangle, \ldots, \langle \boldsymbol{x}_n, y_n \rangle\}$$

Output: Hypothesis $h \in H$ that best approximates f

Based on slide by Tom Mitchell

Sample Dataset

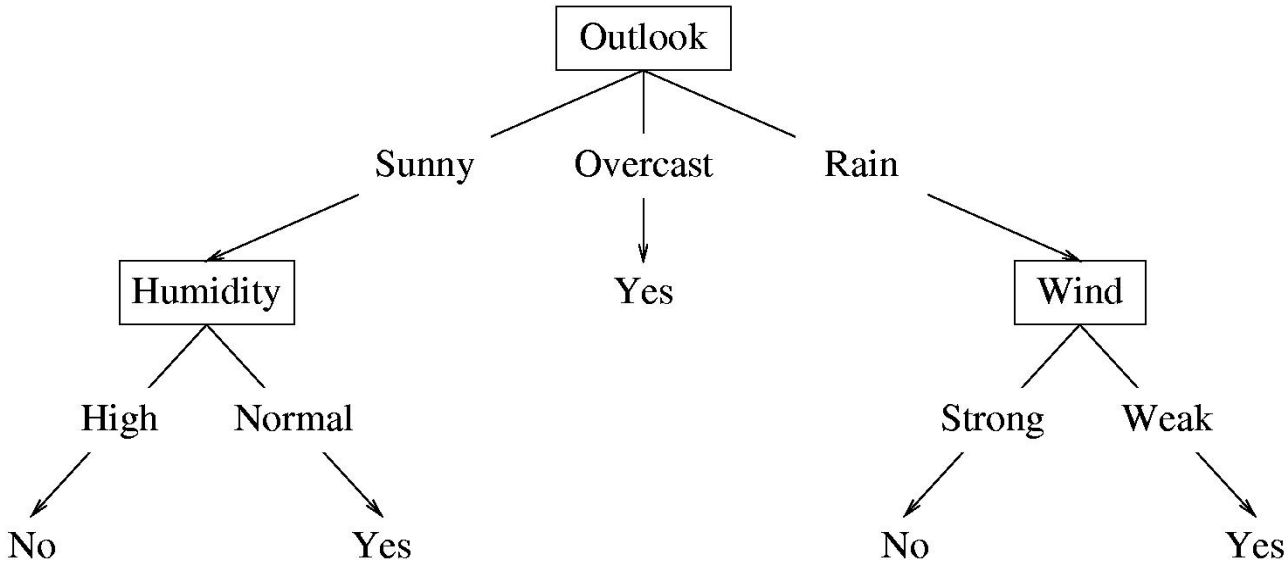
- Columns denote features X_i
- Rows denote labeled instances $\langle \boldsymbol{x}_i, y_i \rangle$
- Class label denotes whether a tennis game was played

$\langle \boldsymbol{x}_i, y_i \rangle$

Predictors				Response
Outlook	**Temperature**	**Humidity**	**Wind**	**Class**
Sunny	Hot	High	Weak	No
Sunny	Hot	High	Strong	No
Overcast	Hot	High	Weak	Yes
Rain	Mild	High	Weak	Yes
Rain	Cool	Normal	Weak	Yes
Rain	Cool	Normal	Strong	No
Overcast	Cool	Normal	Strong	Yes
Sunny	Mild	High	Weak	No
Sunny	Cool	Normal	Weak	Yes
Rain	Mild	Normal	Weak	Yes
Sunny	Mild	Normal	Strong	Yes
Overcast	Mild	High	Strong	Yes
Overcast	Hot	Normal	Weak	Yes
Rain	Mild	High	Strong	No

Decision Tree

- A possible decision tree for the data:

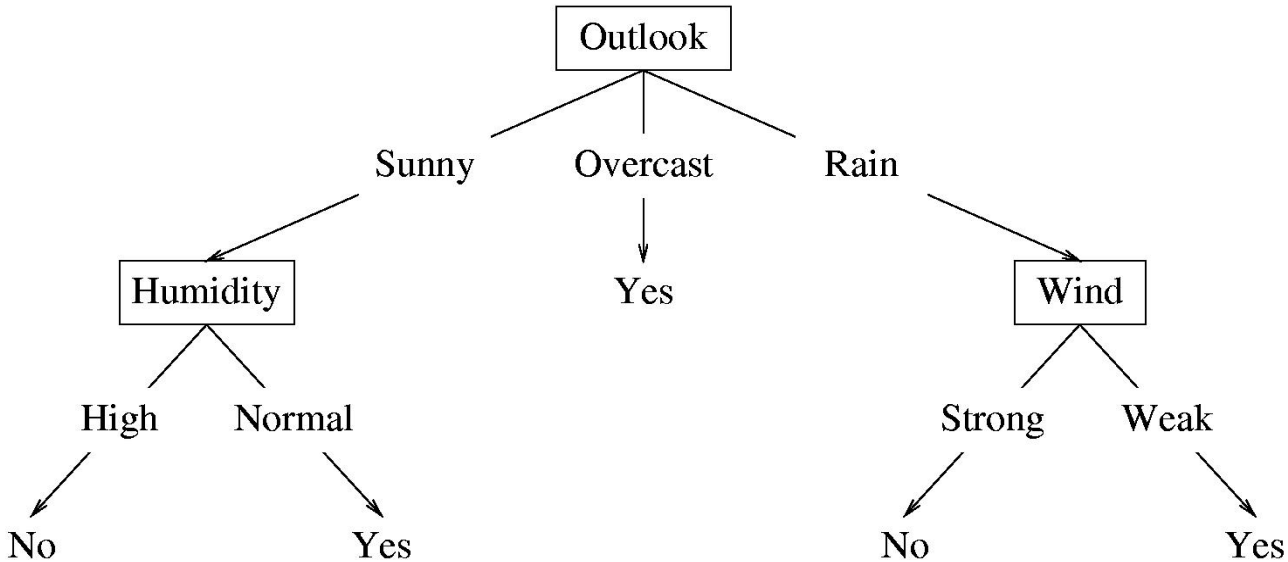


- Each internal node: test one attribute X_i
- Each branch from a node: selects one value for X_i
- Each leaf node: predict Y (or $p(Y \mid \boldsymbol{x} \in \text{leaf})$)

Based on slide by Tom Mitchell

Decision Tree

- A possible decision tree for the data:



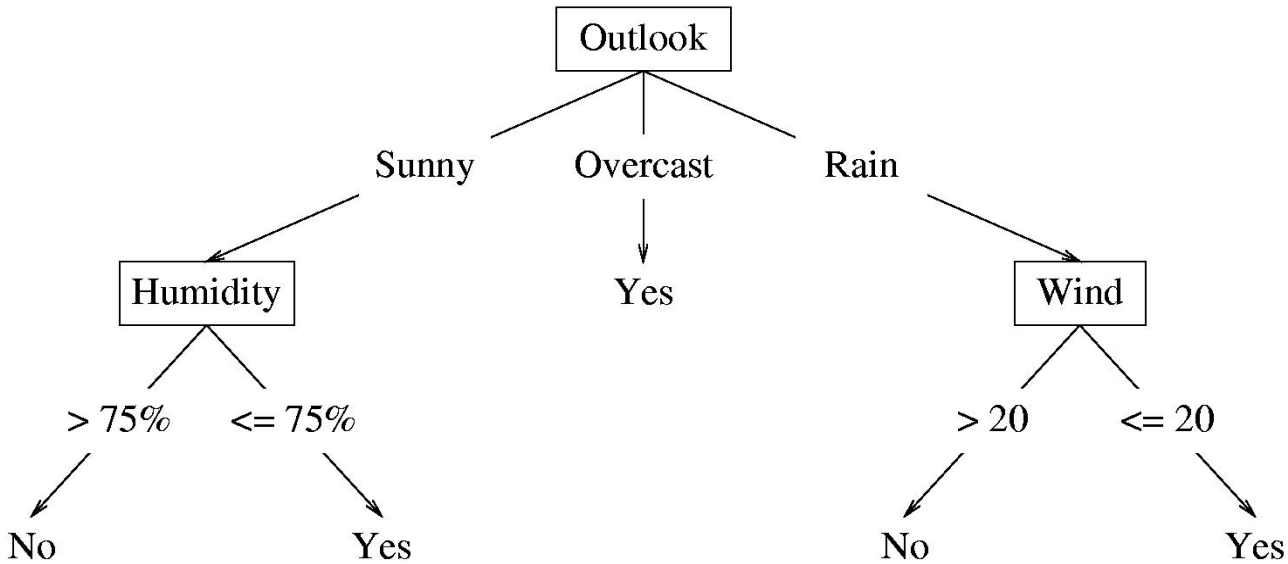
- What prediction would we make for

  <outlook=sunny, temperature=hot, humidity=high, wind=weak> ?

Based on slide by Tom Mitchell

Decision Tree

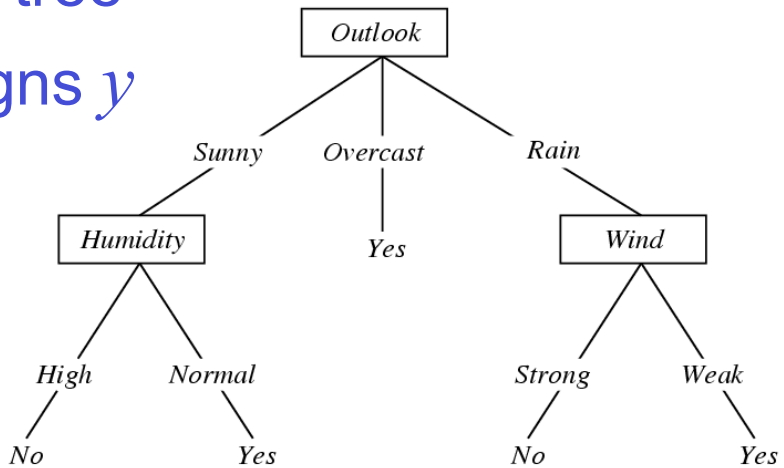
- If features are continuous, internal nodes can test the value of a feature against a threshold



Decision Tree Learning

Problem Setting:

- Set of possible instances X
  - each instance x in X is a feature vector
  - e.g., <*Humidity=low, Wind=weak, Outlook=rain, Temp=hot*>
- Unknown target function $f : X \rightarrow Y$
  - Y is discrete valued
- Set of function hypotheses $H = \{ h \mid h : X \rightarrow Y \}$
  - each hypothesis h is a decision tree
  - trees sorts x to leaf, which assigns y



Slide by Tom Mitchell

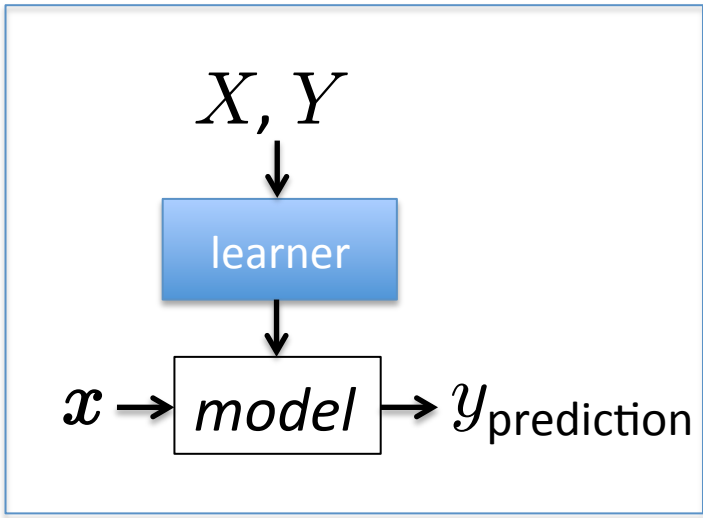
Stages of (Batch) Machine Learning

Given: labeled training data $X, Y = \{\langle \boldsymbol{x}_i, y_i \rangle\}_{i=1}^{n}$

- Assumes each $\boldsymbol{x}_i \sim \mathcal{D}(\mathcal{X})$ with $y_i = f_{target}(\boldsymbol{x}_i)$

Train the model:

model ← *classifier*.train(X, Y)

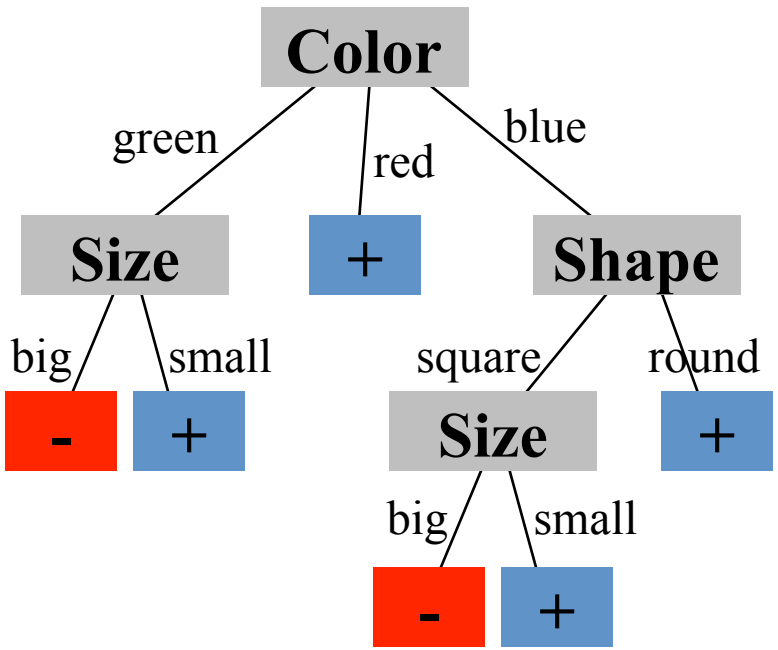


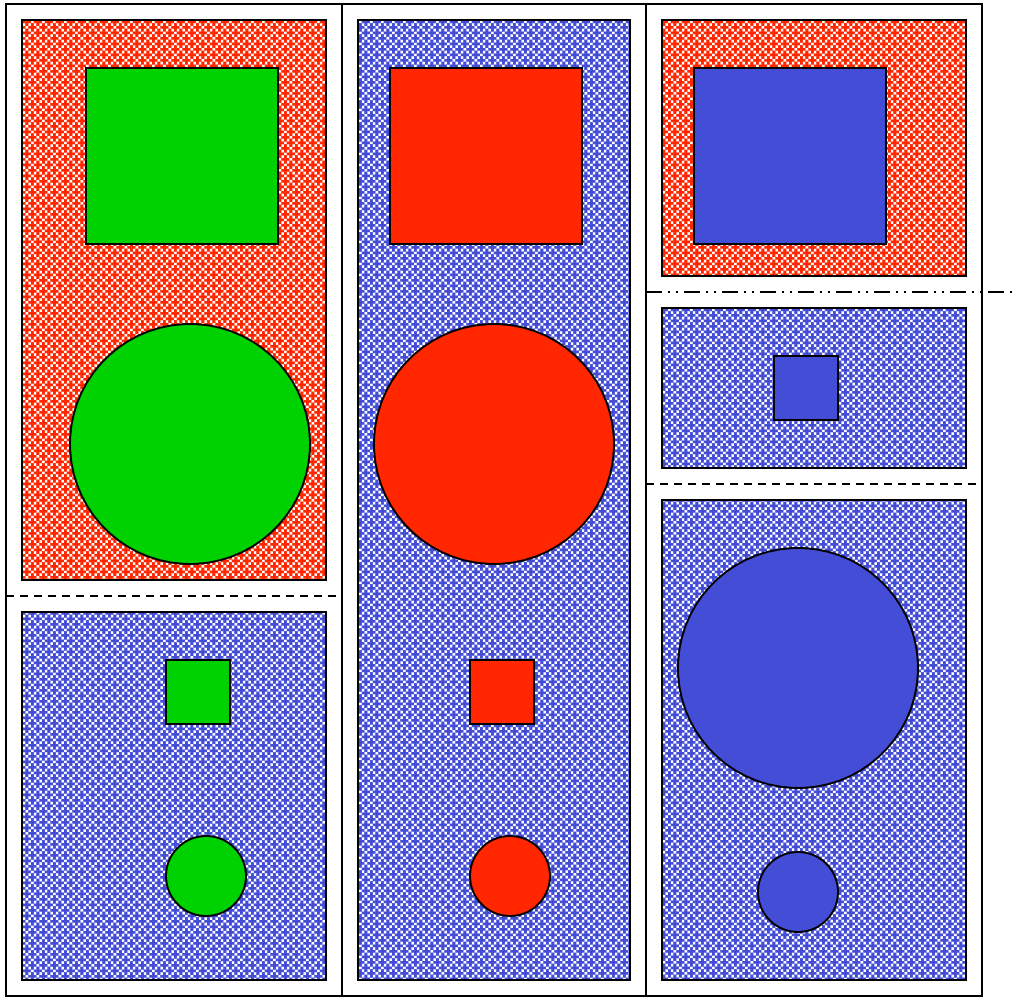
Apply the model to new data:

- Given: new unlabeled instance $\boldsymbol{x} \sim \mathcal{D}(\mathcal{X})$

$y_{\text{prediction}}$ ← *model*.predict($\boldsymbol{x}$)

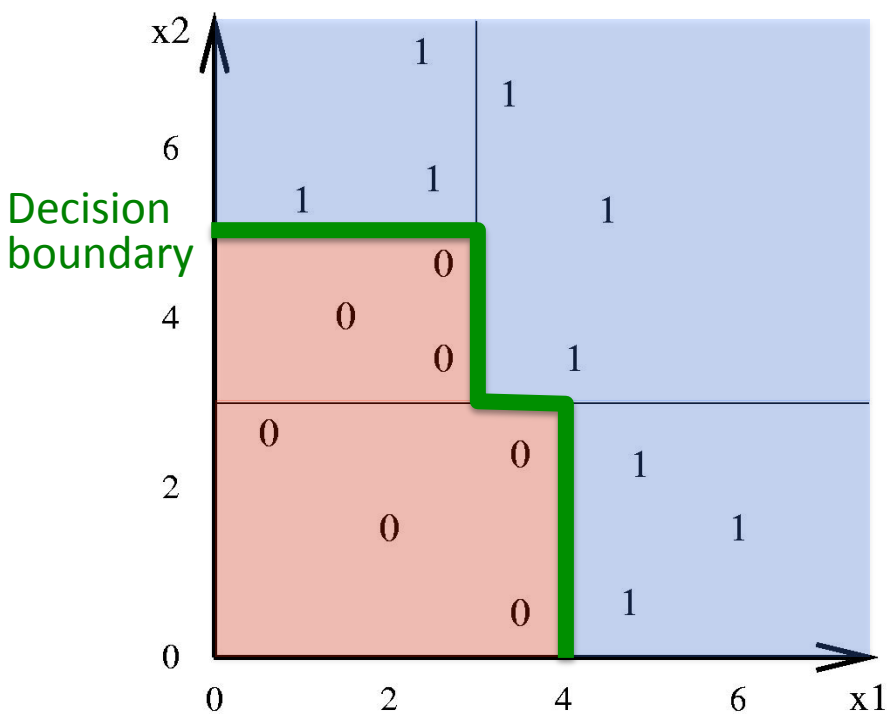
Decision Tree Induced Partition

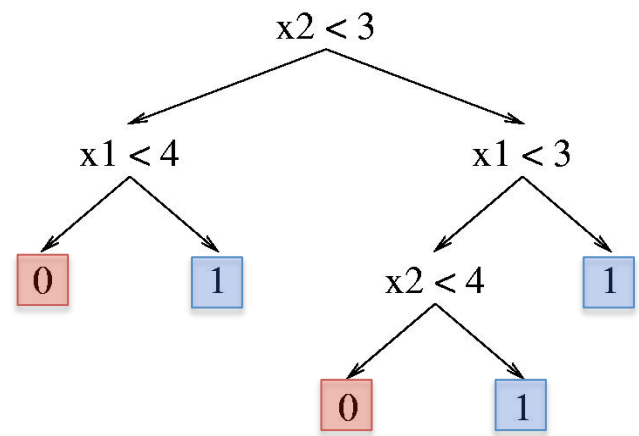




Decision Tree – Decision Boundary

- Decision trees divide the feature space into axis-parallel (hyper-)rectangles
- Each rectangular region is labeled with one label
  - or a probability distribution over labels





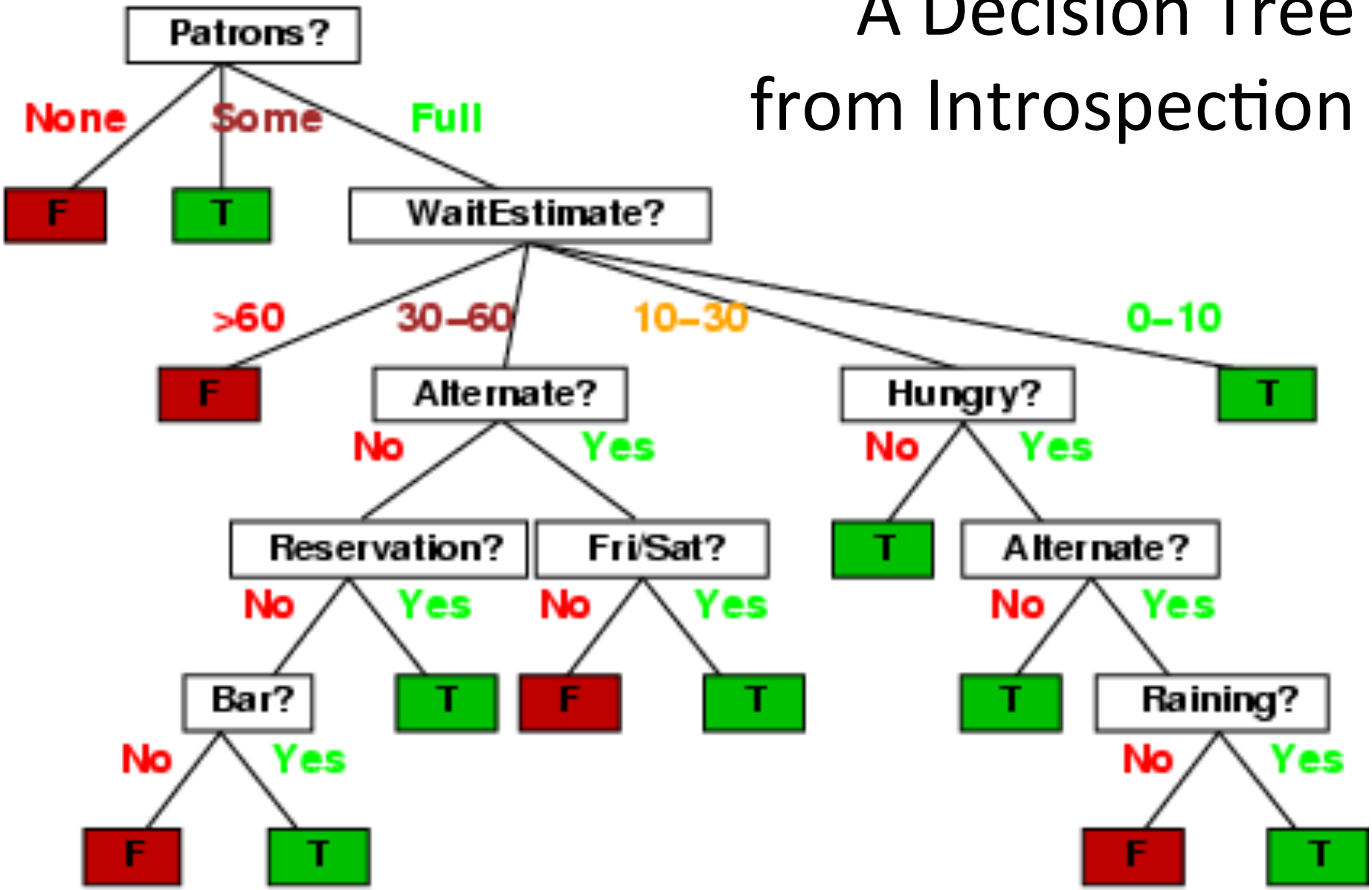
Another Example: Restaurant Domain (Russell & Norvig)

Model a patron's decision of whether to wait for a table at a restaurant

Example	Attributes										Target
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*Wait*
X_1	T	F	F	T	Some	$$$	F	T	French	0–10	T
X_2	T	F	F	T	Full	$	F	F	Thai	30–60	F
X_3	F	T	F	F	Some	$	F	F	Burger	0–10	T
X_4	T	F	T	T	Full	$	F	F	Thai	10–30	T
X_5	T	F	T	F	Full	$$$	F	T	French	>60	F
X_6	F	T	F	T	Some	$$	T	T	Italian	0–10	T
X_7	F	T	F	F	None	$	T	F	Burger	0–10	F
X_8	F	F	F	T	Some	$$	T	T	Thai	0–10	T
X_9	F	T	T	F	Full	$	T	F	Burger	>60	F
X_{10}	T	T	T	T	Full	$$$	F	T	Italian	10–30	F
X_{11}	F	F	F	F	None	$	F	F	Thai	0–10	F
X_{12}	T	T	T	T	Full	$	F	F	Burger	30–60	T

~7,000 possible cases

A Decision Tree from Introspection



Is this the best decision tree?

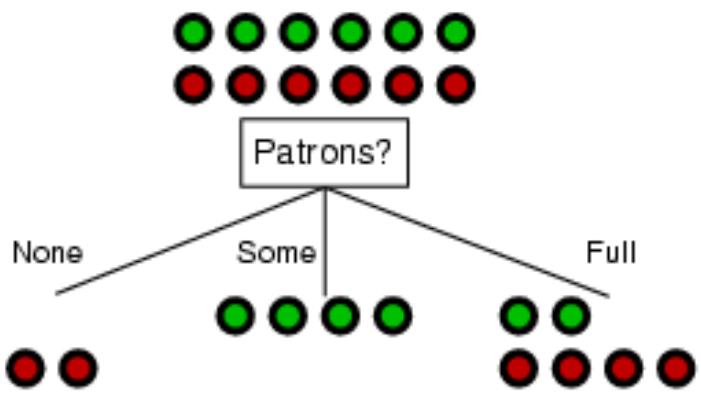
Choosing the Best Attribute

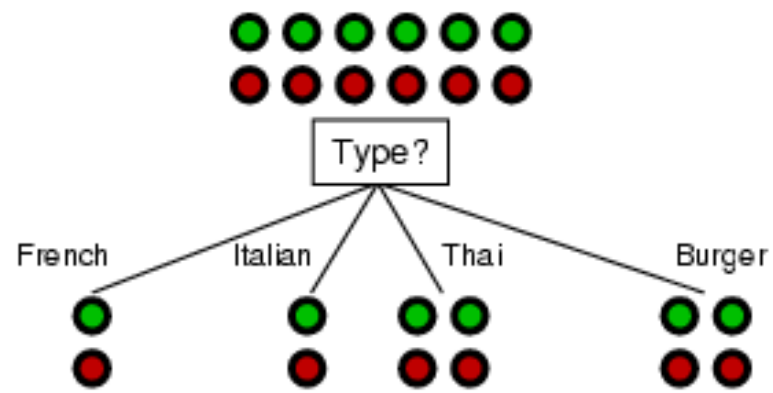
Key problem: choosing which attribute to split a given set of examples

- Some possibilities are:
  - **Random:** Select any attribute at random
  - **Least-Values:** Choose the attribute with the smallest number of possible values
  - **Most-Values:** Choose the attribute with the largest number of possible values
  - **Max-Gain:** Choose the attribute that has the largest expected *information gain*
    - i.e., attribute that results in smallest expected size of subtrees rooted at its children
- The ID3 algorithm uses the Max-Gain method of selecting the best attribute

Choosing an Attribute

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

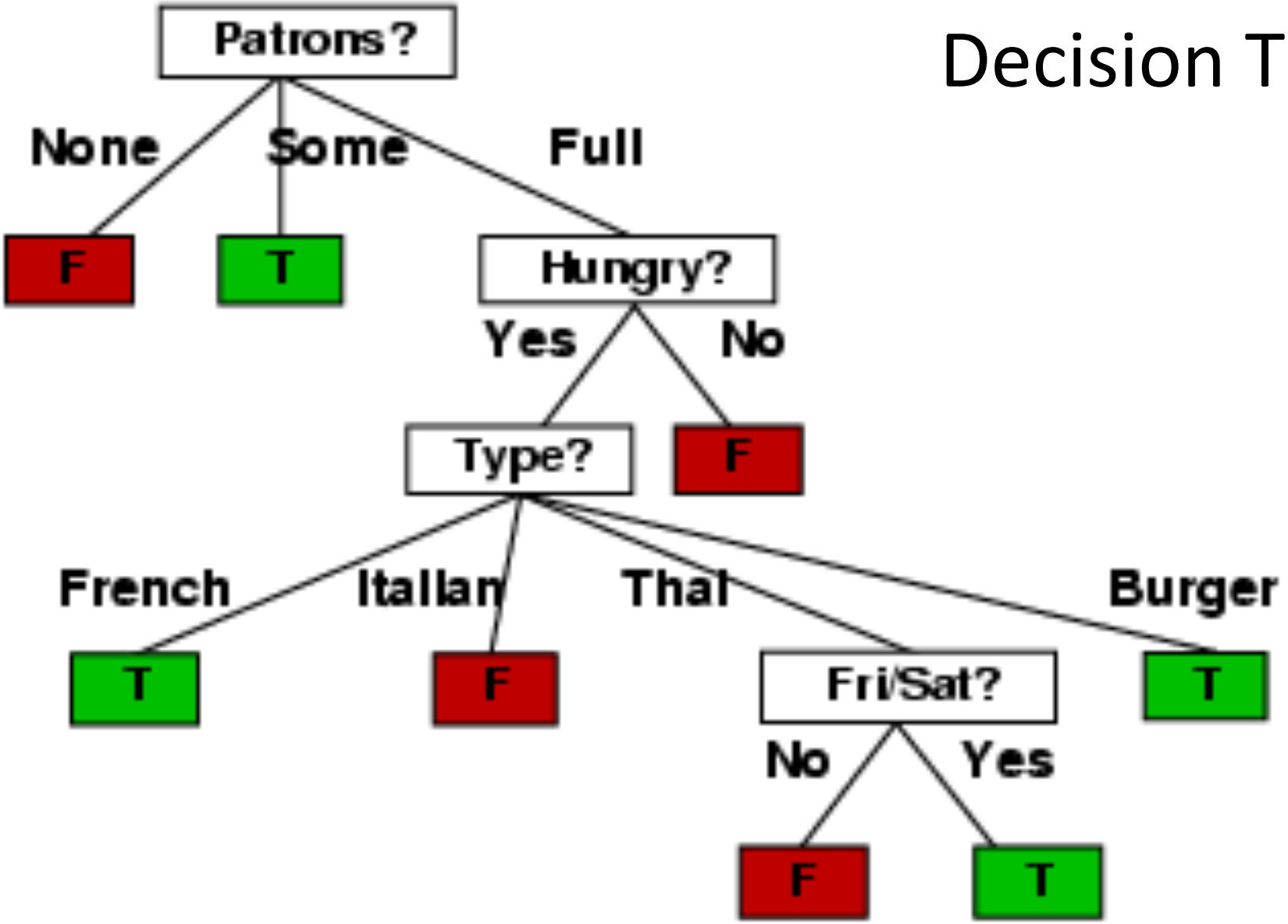




Which split is more informative: *Patrons?* or *Type?*

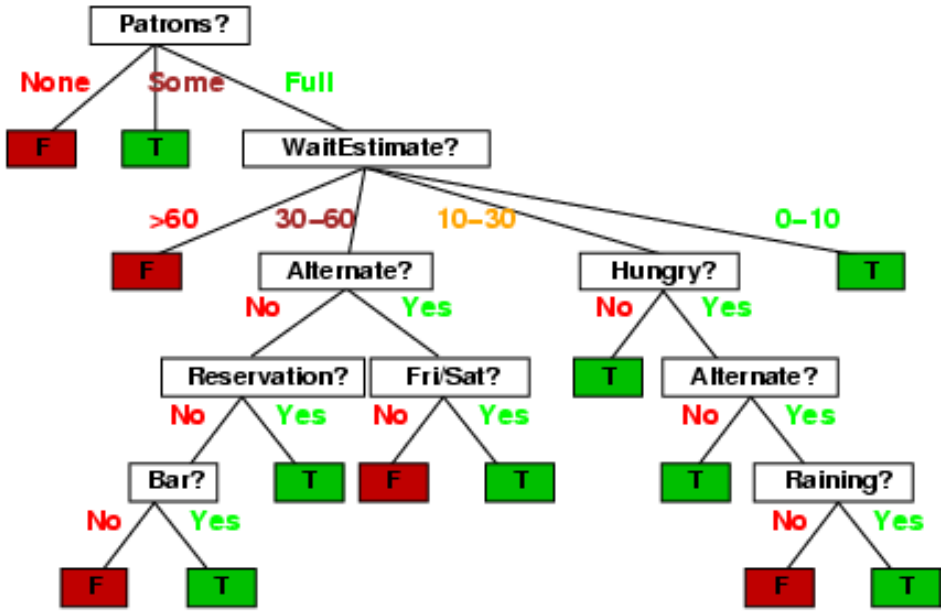
Based on Slide from M. desJardins & T. Finin

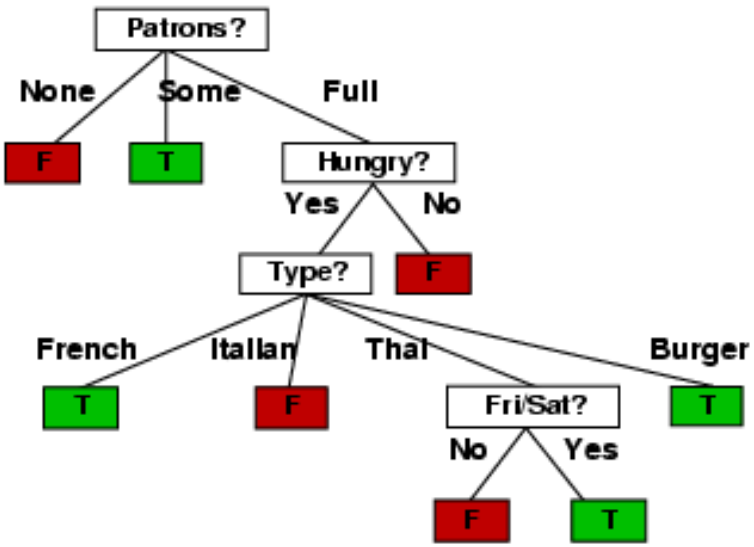
ID3-induced Decision Tree



Based on Slide from M. desJardins & T. Finin

Compare the Two Decision Trees



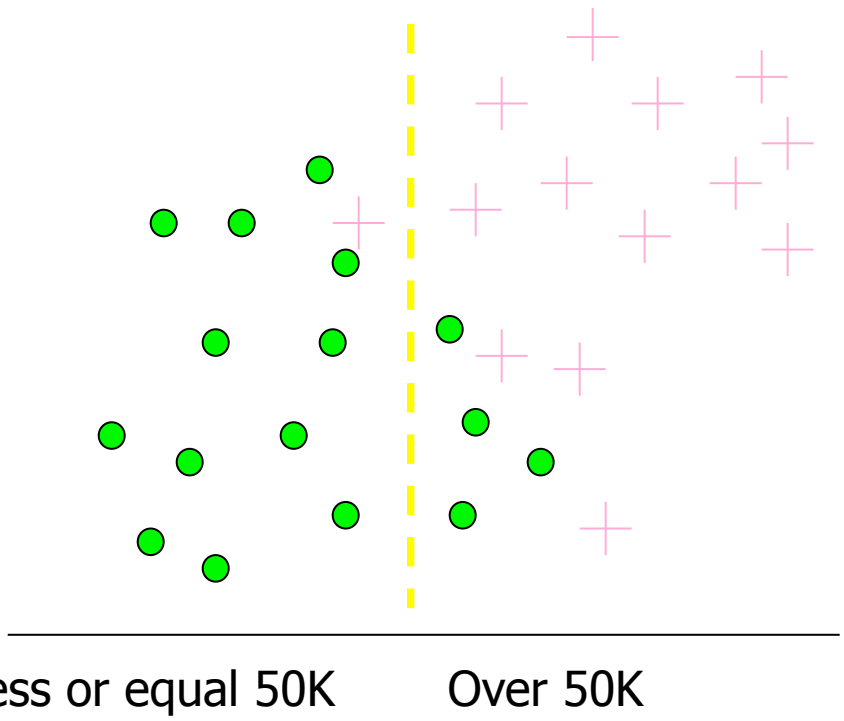


Based on Slide from M. desJardins & T. Finin

Information Gain

Which test is more informative?

Split over whether Balance exceeds 50K



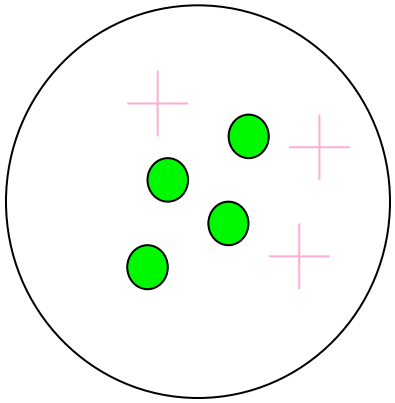
Split over whether applicant is employed

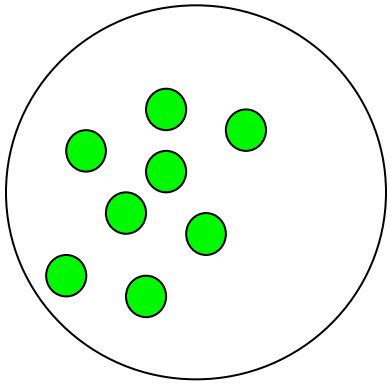
Based on slide by Pedro Domingos

Information Gain

Impurity/Entropy (informal)

– Measures the level of **impurity** in a group of examples

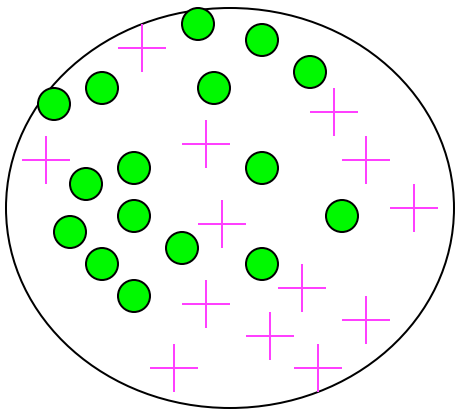




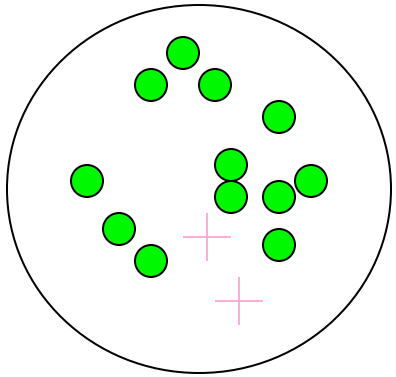
Based on slide by Pedro Domingos

Impurity

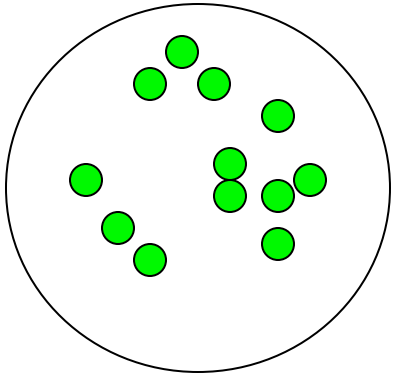
Very impure group



Less impure



Minimum impurity



Based on slide by Pedro Domingos

Entropy: a common way to measure impurity

of possible values for X

Entropy $H(X)$ of a random variable X

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

$H(X)$ 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:

$$\sum_{i=1}^{n} P(X=i)(-\log_2 P(X=i))$$

Slide by Tom Mitchell

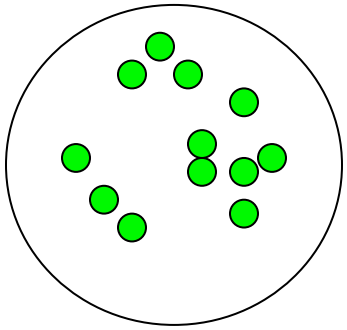
2-Class Cases:

Entropy $$H(x) = -\sum_{i=1}^{n} P(x=i)\log_2 P(x=i)$$

- What is the entropy of a group in which all examples belong to the same class?
  - entropy = - 1 $\log_2 1$ = 0

not a good training set for learning

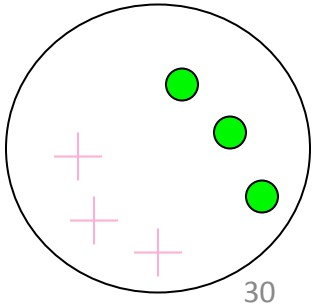
Minimum impurity



- What is the entropy of a group with 50% in either class?
  - entropy = -0.5 $\log_2 0.5$ – 0.5 $\log_2 0.5$ =1

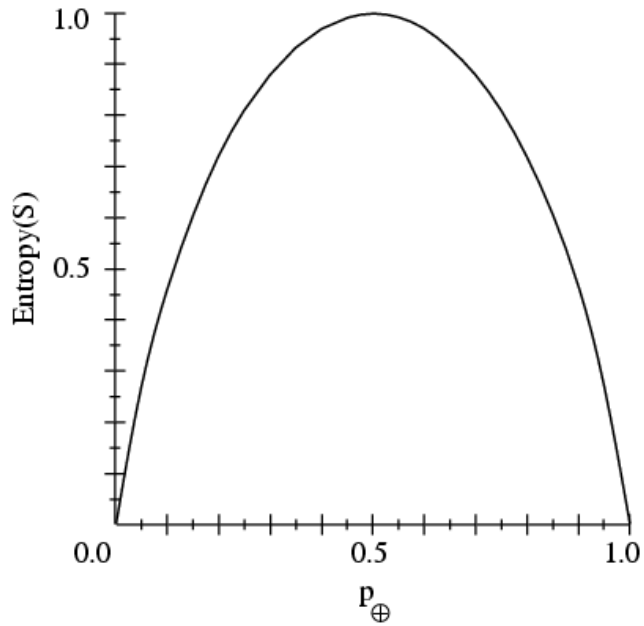
good training set for learning

Maximum impurity



Based on slide by Pedro Domingos

Sample Entropy



- S is a sample of training examples
- $p_\oplus$ is the proportion of positive examples in S
- $p_\ominus$ is the proportion of negative examples in S
- Entropy measures the impurity of S

$$H(S) \equiv -p_\oplus \log_2 p_\oplus - p_\ominus \log_2 p_\ominus$$

Slide by Tom Mitchell

Information Gain

- We want to determine which attribute in a given set of training feature vectors is most useful for discriminating between the classes to be learned.
- Information gain tells us how important a given attribute of the feature vectors is.
- We will use it to decide the ordering of attributes in the nodes of a decision tree.

Based on slide by Pedro Domingos

From Entropy to Information Gain

Entropy $H(X)$ of a random variable X

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

Specific conditional entropy $H(X|Y=v)$ of X given $Y=v$:

$$H(X|Y=v) = -\sum_{i=1}^{n} P(X=i|Y=v)\log_2 P(X=i|Y=v)$$

Conditional entropy $H(X|Y)$ of X given Y :

$$H(X|Y) = \sum_{v \in values(Y)} P(Y=v)H(X|Y=v)$$

Mututal information (aka Information Gain) of X and Y :

$$I(X,Y) = H(X) - H(X|Y) = H(Y) - H(Y|X)$$

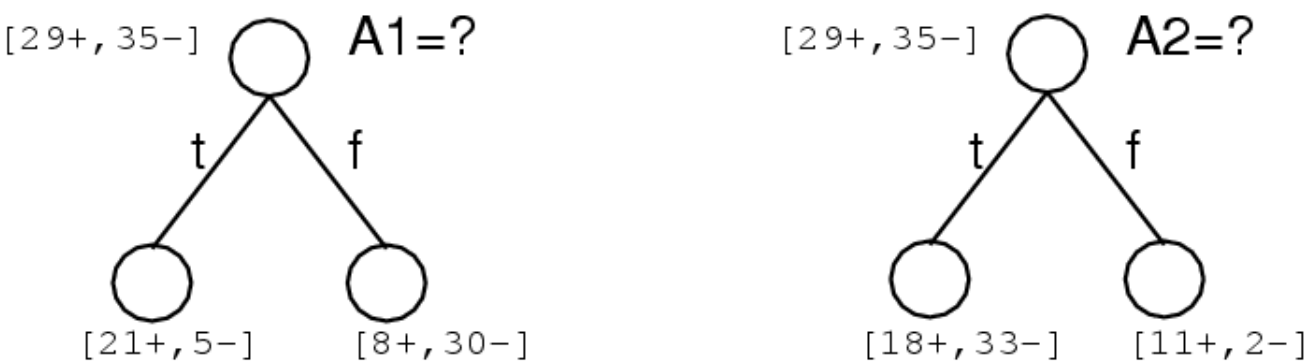
Slide by Tom Mitchell

Information Gain

Information Gain is the mutual information between input attribute A and target variable Y

Information Gain is the expected reduction in entropy of target variable Y for data sample S, due to sorting on variable A

$$Gain(S, A) = I_S(A, Y) = H_S(Y) - H_S(Y|A)$$



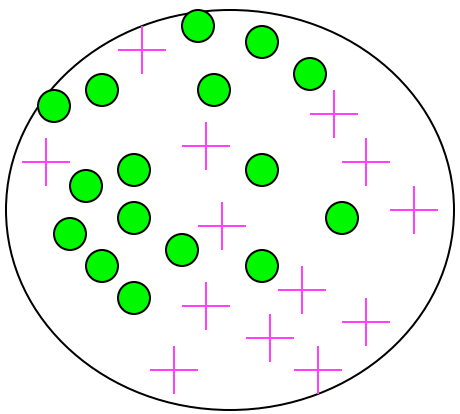
Slide by Tom Mitchell

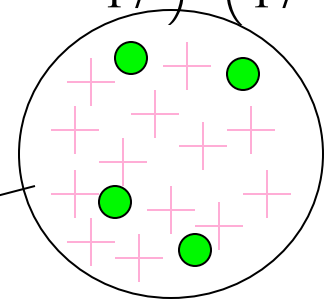
Calculating Information Gain

Information Gain = entropy(parent) – [average entropy(children)]

child entropy $-\left(\frac{13}{17}\cdot\log_2\frac{13}{17}\right)-\left(\frac{4}{17}\cdot\log_2\frac{4}{17}\right)=0.787$

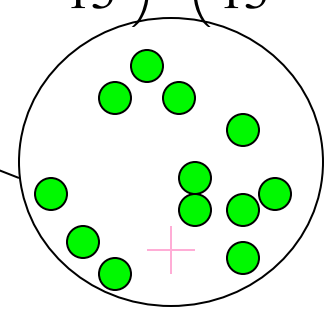
Entire population (30 instances)





17 instances

child entropy $-\left(\frac{1}{13}\cdot\log_2\frac{1}{13}\right)-\left(\frac{12}{13}\cdot\log_2\frac{12}{13}\right)=0.391$



13 instances

parent entropy $-\left(\frac{14}{30}\cdot\log_2\frac{14}{30}\right)-\left(\frac{16}{30}\cdot\log_2\frac{16}{30}\right)=0.996$

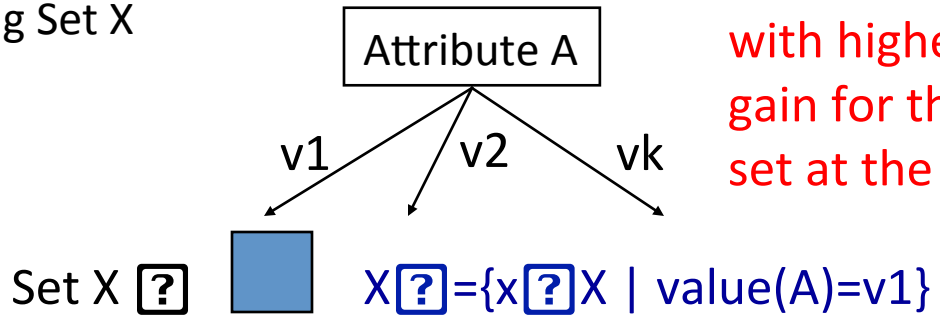
(Weighted) Average Entropy of Children = $\left(\frac{17}{30}\cdot 0.787\right)+\left(\frac{13}{30}\cdot 0.391\right)=0.615$

Information Gain= 0.996 - 0.615 = 0.38

Based on slide by Pedro Domingos

Using Information Gain to Construct a Decision Tree

Full Training Set X



Choose the attribute A with highest information gain for the full training set at the root of the tree.

Construct child nodes for each value of A. Each has an associated subset of vectors in which A has a particular value.

repeat recursively till when?

Disadvantage of information gain:

- It prefers attributes with large number of values that split the data into small, pure subsets
- Quinlan's gain ratio uses normalization to improve this

Based on slide by Pedro Domingos

How well does it work?

Many case studies have shown that decision trees are at least as accurate as human experts.

- A study for diagnosing breast cancer had humans correctly classifying the examples 65% of the time; the decision tree classified 72% correct
- British Petroleum designed a decision tree for gas-oil separation for offshore oil platforms that replaced an earlier rule-based expert system
- Cessna designed an airplane flight controller using 90,000 examples and 20 attributes per example

Based on Slide from M. desJardins & T. Finin

The Naïve Bayes Classifier

Bayes' Rule

- Recall Baye's Rule:

$$P(\text{hypothesis} \mid \text{evidence}) = \frac{P(\text{evidence} \mid \text{hypothesis}) \times P(\text{hypothesis})}{P(\text{evidence})}$$

- Equivalently, we can write:

$$P(Y = y_k \mid X = \mathbf{x}_i) = \frac{P(Y = y_k)P(X = \mathbf{x}_i \mid Y = y_k)}{P(X = \mathbf{x}_i)}$$

where X is a random variable representing the evidence and Y is a random variable for the label

- This is actually short for:

$$P(Y = y_k \mid X = \mathbf{x}_i) = \frac{P(Y = y_k)P(X_1 = x_{i,1} \wedge \ldots \wedge X_d = x_{i,d} \mid Y = y_k)}{P(X_1 = x_{i,1} \wedge \ldots \wedge X_d = x_{i,d})}$$

where X_j denotes the random variable for the jth feature

Naïve Bayes Classifier

Idea: Use the training data to estimate

$$P(X \mid Y) \quad \text{and} \quad P(Y) \ .$$

Then, use Bayes rule to infer $P(Y|X_{\text{new}})$ for new data

---

Easy to estimate from data

Impractical, but necessary

$$P(Y = y_k \mid X = \mathbf{x}_i) = \frac{\boxed{P(Y = y_k)}\,\boxed{P(X_1 = x_{i,1} \wedge \ldots \wedge X_d = x_{i,d} \mid Y = y_k)}}{\boxed{P(X_1 = x_{i,1} \wedge \ldots \wedge X_d = x_{i,d})}}$$

Unnecessary, as it turns out

- Recall that estimating the joint probability distribution $P(X_1, X_2, \ldots, X_d \mid Y)$ is not practical

Naïve Bayes Classifier

Problem: estimating the joint PD or CPD isn't practical

– Severely overfits,

However, if we make the assumption that the attributes are independent given the class label, estimation is easy!

$$P(X_1, X_2, \ldots, X_d \mid Y) = \prod_{j=1}^{d} P(X_j \mid Y)$$

- In other words, we assume all attributes are *conditionally independent* given Y
- Often this assumption is violated in practice, but more on that later…

Training Naïve Bayes

Estimate $P(X_j \mid Y)$ and $P(Y)$ directly from the training data by counting!

Sky	Temp	Humid	Wind	Water	Forecast	Play?
sunny	warm	normal	strong	warm	same	*yes*
sunny	warm	high	strong	warm	same	*yes*
rainy	cold	high	strong	warm	change	*no*
sunny	warm	high	strong	cool	change	*yes*

$\mathrm{P}(\mathrm{play}) = ?$

$\mathrm{P}(\mathrm{Sky} = \mathrm{sunny} \mid \mathrm{play}) = ?$

$\mathrm{P}(\mathrm{Humid} = \mathrm{high} \mid \mathrm{play}) = ?$

$\ldots$

$\mathrm{P}(\neg\mathrm{play}) = ?$

$\mathrm{P}(\mathrm{Sky} = \mathrm{sunny} \mid \neg\mathrm{play}) = ?$

$\mathrm{P}(\mathrm{Humid} = \mathrm{high} \mid \neg\mathrm{play}) = ?$

$\ldots$

Training Naïve Bayes

Estimate $P(X_j \mid Y)$ and $P(Y)$ directly from the training data by counting!

Sky	Temp	Humid	Wind	Water	Forecast	Play?
sunny	warm	normal	strong	warm	same	*yes*
sunny	warm	high	strong	warm	same	*yes*
rainy	cold	high	strong	warm	change	*no*
sunny	warm	high	strong	cool	change	*yes*

$\mathrm{P(play)} = ?$

$\mathrm{P(Sky = sunny \mid play)} = ?$

$\mathrm{P(Humid = high \mid play)} = ?$

$\ldots$

$\mathrm{P(\neg play)} = ?$

$\mathrm{P(Sky = sunny \mid \neg play)} = ?$

$\mathrm{P(Humid = high \mid \neg play)} = ?$

$\ldots$

Training Naïve Bayes

Estimate $P(X_j \mid Y)$ and $P(Y)$ directly from the training data by counting!

Sky	Temp	Humid	Wind	Water	Forecast	Play?
sunny	warm	normal	strong	warm	same	*yes*
sunny	warm	high	strong	warm	same	*yes*
rainy	cold	high	strong	warm	change	*no*
sunny	warm	high	strong	cool	change	*yes*

$\mathrm{P(play)} = 3/4$

$\mathrm{P(Sky = sunny \mid play)} = ?$

$\mathrm{P(Humid = high \mid play)} = ?$

$\ldots$

$\mathrm{P(\neg play)} = 1/4$

$\mathrm{P(Sky = sunny \mid \neg play)} = ?$

$\mathrm{P(Humid = high \mid \neg play)} = ?$

$\ldots$

Training Naïve Bayes

Estimate $P(X_j \mid Y)$ and $P(Y)$ directly from the training data by counting!

Sky	Temp	Humid	Wind	Water	Forecast	Play?
sunny						*yes*
sunny						*yes*
rainy	cold	high	strong	warm	change	*no*
sunny						*yes*

$\mathrm{P(play)} = 3/4$

$\mathrm{P(Sky = sunny \mid play)} = ?$

$\mathrm{P(Humid = high \mid play)} = ?$

$\ldots$

$\mathrm{P(\neg play)} = 1/4$

$\mathrm{P(Sky = sunny \mid \neg play)} = ?$

$\mathrm{P(Humid = high \mid \neg play)} = ?$

$\ldots$

Training Naïve Bayes

Estimate $P(X_j \mid Y)$ and $P(Y)$ directly from the training data by counting!

Sky	Temp	Humid	Wind	Water	Forecast	Play?
sunny						*yes*
sunny						*yes*
rainy	cold	high	strong	warm	change	*no*
sunny						*yes*

$\mathrm{P(play)} = 3/4$

$\mathrm{P(Sky = sunny \mid play)} = 1$

$\mathrm{P(Humid = high \mid play)} = ?$

$\ldots$

$\mathrm{P(\neg play)} = 1/4$

$\mathrm{P(Sky = sunny \mid \neg play)} = ?$

$\mathrm{P(Humid = high \mid \neg play)} = ?$

$\ldots$

Training Naïve Bayes

Estimate $P(X_j \mid Y)$ and $P(Y)$ directly from the training data by counting!

Sky	Temp	Humid	Wind	Water	Forecast	Play?
sunny	warm	normal	strong	warm	same	*yes*
sunny	warm	high	strong	warm	same	*yes*
rainy						*no*
sunny	warm	high	strong	cool	change	*yes*

$\mathrm{P}(\mathrm{play}) = 3/4$

$\mathrm{P}(\mathrm{Sky} = \mathrm{sunny} \mid \mathrm{play}) = 1$

$\mathrm{P}(\mathrm{Humid} = \mathrm{high} \mid \mathrm{play}) = ?$

$\ldots$

$\mathrm{P}(\neg\mathrm{play}) = 1/4$

$\mathrm{P}(\mathrm{Sky} = \mathrm{sunny} \mid \neg\mathrm{play}) = ?$

$\mathrm{P}(\mathrm{Humid} = \mathrm{high} \mid \neg\mathrm{play}) = ?$

$\ldots$

Training Naïve Bayes

Estimate $P(X_j \mid Y)$ and $P(Y)$ directly from the training data by counting!

Sky	Temp	Humid	Wind	Water	Forecast	Play?
sunny	warm	normal	strong	warm	same	*yes*
sunny	warm	high	strong	warm	same	*yes*
rainy						*no*
sunny	warm	high	strong	cool	change	*yes*

$\mathrm{P}(\mathrm{play}) = 3/4$

$\mathrm{P}(\mathrm{Sky} = \mathrm{sunny} \mid \mathrm{play}) = 1$

$\mathrm{P}(\mathrm{Humid} = \mathrm{high} \mid \mathrm{play}) = ?$

$\ldots$

$\mathrm{P}(\neg\mathrm{play}) = 1/4$

$\mathrm{P}(\mathrm{Sky} = \mathrm{sunny} \mid \neg\mathrm{play}) = 0$

$\mathrm{P}(\mathrm{Humid} = \mathrm{high} \mid \neg\mathrm{play}) = ?$

$\ldots$

Training Naïve Bayes

Estimate $P(X_j \mid Y)$ and $P(Y)$ directly from the training data by counting!

Sky	Temp	Humid	Wind	Water	Forecast	Play?
		normal				*yes*
		high				*yes*
rainy	cold	high	strong	warm	change	*no*
		high				*yes*

$\mathrm{P(play)} = 3/4$

$\mathrm{P(Sky = sunny \mid play)} = 1$

$\mathrm{P(Humid = high \mid play)} = ?$

$\ldots$

$\mathrm{P(\neg play)} = 1/4$

$\mathrm{P(Sky = sunny \mid \neg play)} = 0$

$\mathrm{P(Humid = high \mid \neg play)} = ?$

$\ldots$

Training Naïve Bayes

Estimate $P(X_j \mid Y)$ and $P(Y)$ directly from the training data by counting!

Sky	Temp	Humid	Wind	Water	Forecast	Play?
		normal				*yes*
		high				*yes*
rainy	cold	high	strong	warm	change	*no*
		high				*yes*

$\mathrm{P(play)} = 3/4$

$\mathrm{P(Sky = sunny \mid play)} = 1$

$\mathrm{P(Humid = high \mid play)} = 2/3$

$\ldots$

$\mathrm{P(\neg play)} = 1/4$

$\mathrm{P(Sky = sunny \mid \neg play)} = 0$

$\mathrm{P(Humid = high \mid \neg play)} = ?$

$\ldots$

Training Naïve Bayes

Estimate $P(X_j \mid Y)$ and $P(Y)$ directly from the training data by counting!

Sky	Temp	Humid	Wind	Water	Forecast	Play?
sunny	warm	normal	strong	warm	same	*yes*
sunny	warm	high	strong	warm	same	*yes*
		high				*no*
sunny	warm	high	strong	cool	change	*yes*

$\mathrm{P(play)} = 3/4$

$\mathrm{P(Sky = sunny \mid play)} = 1$

$\mathrm{P(Humid = high \mid play)} = 2/3$

$\ldots$

$\mathrm{P(\neg play)} = 1/4$

$\mathrm{P(Sky = sunny \mid \neg play)} = 0$

$\mathrm{P(Humid = high \mid \neg play)} = ?$

$\ldots$

Training Naïve Bayes

Estimate $P(X_j \mid Y)$ and $P(Y)$ directly from the training data by counting!

Sky	Temp	Humid	Wind	Water	Forecast	Play?
sunny	warm	normal	strong	warm	same	*yes*
sunny	warm	high	strong	warm	same	*yes*
		high				*no*
sunny	warm	high	strong	cool	change	*yes*

$\mathrm{P}(\mathrm{play}) = 3/4$

$\mathrm{P}(\mathrm{Sky} = \mathrm{sunny} \mid \mathrm{play}) = 1$

$\mathrm{P}(\mathrm{Humid} = \mathrm{high} \mid \mathrm{play}) = 2/3$

$\ldots$

$\mathrm{P}(\neg\mathrm{play}) = 1/4$

$\mathrm{P}(\mathrm{Sky} = \mathrm{sunny} \mid \neg\mathrm{play}) = 0$

$\mathrm{P}(\mathrm{Humid} = \mathrm{high} \mid \neg\mathrm{play}) = 1$

$\ldots$

Training Naïve Bayes

Estimate $P(X_j \mid Y)$ and $P(Y)$ directly from the training data by counting!

Sky	Temp	Humid	Wind	Water	Forecast	Play?
sunny	warm	normal	strong	warm	same	*yes*
sunny	warm	high	strong	warm	same	*yes*
rainy	cold	high	strong	warm	change	*no*
sunny	warm	high	strong	cool	change	*yes*

$\mathrm{P}(\mathrm{play}) = 3/4$

$\mathrm{P}(\mathrm{Sky} = \mathrm{sunny} \mid \mathrm{play}) = 1$

$\mathrm{P}(\mathrm{Humid} = \mathrm{high} \mid \mathrm{play}) = 2/3$

$\ldots$

$\mathrm{P}(\neg\mathrm{play}) = 1/4$

$\mathrm{P}(\mathrm{Sky} = \mathrm{sunny} \mid \neg\mathrm{play}) = 0$

$\mathrm{P}(\mathrm{Humid} = \mathrm{high} \mid \neg\mathrm{play}) = 1$

$\ldots$

Laplace Smoothing

- Notice that some probabilities estimated by counting might be zero
  - Possible overfitting!

- Fix by using Laplace smoothing:
  - Adds 1 to each count

$$P(X_j = v \mid Y = y_k) = \frac{\mathrm{c}_v + 1}{\sum_{v' \in \mathrm{values}(X_j)} \mathrm{c}_{v'} + |\mathrm{values}(X_j)|}$$

where

- c_v is the count of training instances with a value of v for attribute j and class label y_k
- $|\mathrm{values}(X_j)|$ is the number of values X_j can take on

Training Naïve Bayes with Laplace Smoothing

Estimate $P(X_j \mid Y)$ and $P(Y)$ directly from the training data by counting with Laplace smoothing:

Sky	Temp	Humid	Wind	Water	Forecast	Play?
sunny						*yes*
sunny						*yes*
rainy	cold	high	strong	warm	change	*no*
sunny						*yes*

$\mathrm{P}(\mathrm{play}) = 3/4$

$\mathrm{P}(\mathrm{Sky} = \mathrm{sunny} \mid \mathrm{play}) = 4/5$

$\mathrm{P}(\mathrm{Humid} = \mathrm{high} \mid \mathrm{play}) = ?$

$\ldots$

$\mathrm{P}(\neg\mathrm{play}) = 1/4$

$\mathrm{P}(\mathrm{Sky} = \mathrm{sunny} \mid \neg\mathrm{play}) = ?$

$\mathrm{P}(\mathrm{Humid} = \mathrm{high} \mid \neg\mathrm{play}) = ?$

$\ldots$

Training Naïve Bayes with Laplace Smoothing

Estimate $P(X_j \mid Y)$ and $P(Y)$ directly from the training data by counting with Laplace smoothing:

Sky	Temp	Humid	Wind	Water	Forecast	Play?
sunny	warm	normal	strong	warm	same	*yes*
sunny	warm	high	strong	warm	same	*yes*
rainy						*no*
sunny	warm	high	strong	cool	change	*yes*

$\mathrm{P}(\mathrm{play}) = 3/4$

$\mathrm{P}(\mathrm{Sky} = \mathrm{sunny} \mid \mathrm{play}) = 4/5$

$\mathrm{P}(\mathrm{Humid} = \mathrm{high} \mid \mathrm{play}) = ?$

$\ldots$

$\mathrm{P}(\neg\mathrm{play}) = 1/4$

$\mathrm{P}(\mathrm{Sky} = \mathrm{sunny} \mid \neg\mathrm{play}) = 1/3$

$\mathrm{P}(\mathrm{Humid} = \mathrm{high} \mid \neg\mathrm{play}) = ?$

$\ldots$

Training Naïve Bayes with Laplace Smoothing

Estimate $P(X_j \mid Y)$ and $P(Y)$ directly from the training data by counting with Laplace smoothing:

Sky	Temp	Humid	Wind	Water	Forecast	Play?
		normal				*yes*
		high				*yes*
rainy	cold	high	strong	warm	change	*no*
		high				*yes*

$\mathrm{P(play)} = 3/4$

$\mathrm{P(Sky = sunny \mid play)} = 4/5$

$\mathrm{P(Humid = high \mid play)} = 3/5$

$\ldots$

$\mathrm{P(\neg play)} = 1/4$

$\mathrm{P(Sky = sunny \mid \neg play)} = 1/3$

$\mathrm{P(Humid = high \mid \neg play)} = ?$

$\ldots$

Training Naïve Bayes with Laplace Smoothing

Estimate $P(X_j \mid Y)$ and $P(Y)$ directly from the training data by counting with Laplace smoothing:

Sky	Temp	Humid	Wind	Water	Forecast	Play?
sunny	warm	normal	strong	warm	same	*yes*
sunny	warm	high	strong	warm	same	*yes*
		high				*no*
sunny	warm	high	strong	cool	change	*yes*

$\mathrm{P}(\mathrm{play}) = 3/4$

$\mathrm{P}(\mathrm{Sky} = \mathrm{sunny} \mid \mathrm{play}) = 4/5$

$\mathrm{P}(\mathrm{Humid} = \mathrm{high} \mid \mathrm{play}) = 3/5$

$\ldots$

$\mathrm{P}(\neg\mathrm{play}) = 1/4$

$\mathrm{P}(\mathrm{Sky} = \mathrm{sunny} \mid \neg\mathrm{play}) = 1/3$

$\mathrm{P}(\mathrm{Humid} = \mathrm{high} \mid \neg\mathrm{play}) = 2/3$

$\ldots$

Using the Naïve Bayes Classifier

- Now, we have

$$P(Y = y_k \mid X = \mathbf{x}_i) = \frac{P(Y = y_k) \prod_{j=1}^{d} P(X_j = x_{i,j} \mid Y = y_k)}{\boxed{P(X = \mathbf{x}_i)}}$$

This is constant for a given instance, and so irrelevant to our prediction

  – In practice, we use log-probabilities to prevent underflow

- To classify a new point $\mathbf{x}$,

$$h(\mathbf{x}) = \arg\max_{y_k} \; P(Y = y_k) \prod_{j=1}^{d} P(X_j = \underbrace{x_j}_{j^{\text{th}} \text{ attribute value of } \mathbf{x}} \mid Y = y_k)$$

$$= \arg\max_{y_k} \; \log P(Y = y_k) + \sum_{j=1}^{d} \log P(X_j = x_j \mid Y = y_k)$$

The Naïve Bayes Classifier Algorithm

- For each class label y_k
  - Estimate $\mathrm{P}(\mathrm{Y} = y_k)$ from the data
  - For each value $x_{i,j}$ of each attribute X_i
    - Estimate $\mathrm{P}(\mathrm{X}_i = x_{i,j} \mid \mathrm{Y} = y_k)$

- Classify a new point via:

$$h(\mathbf{x}) = \arg\max_{y_k} \ \log P(Y = y_k) + \sum_{j=1}^{d} \log P(X_j = x_j \mid Y = y_k)$$

- In practice, the independence assumption doesn't often hold true, but Naïve Bayes performs very well despite it

Computing Probabilities (Not Just Predicting Labels)

- NB classifier gives predictions, not probabilities, because we ignore $\mathrm{P(X)}$ (the denominator in Bayes rule)
- Can produce probabilities by:
  - For each possible class label y_k, compute

$$\underbrace{\tilde{P}(Y = y_k \mid X = \mathbf{x})}_{} = P(Y = y_k) \prod_{j=1}^{d} P(X_j = x_j \mid Y = y_k)$$

This is the numerator of Bayes rule, and is therefore off the true probability by a factor of α that makes probabilities sum to 1

  - α is given by $\alpha = \dfrac{1}{\sum_{k=1}^{\#classes} \tilde{P}(Y = y_k \mid X = \mathbf{x})}$
  - Class probability is given by

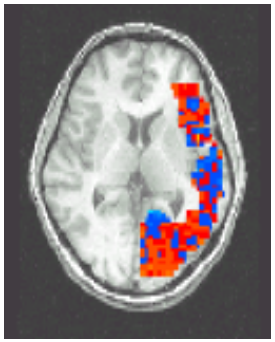
$$P(Y = y_k \mid X = \mathbf{x}) = \alpha \tilde{P}(Y = y_k \mid X = \mathbf{x})$$

Naïve Bayes Applications

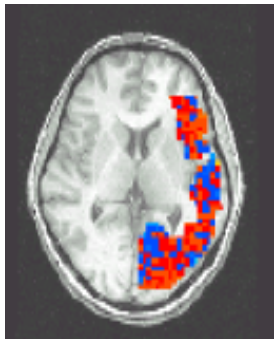
- Text classification
  - Which e-mails are spam?
  - Which e-mails are meeting notices?
  - Which author wrote a document?

- Classifying mental states

Learning $\mathrm{P}(\mathrm{BrainActivity} \mid \mathrm{WordCategory})$



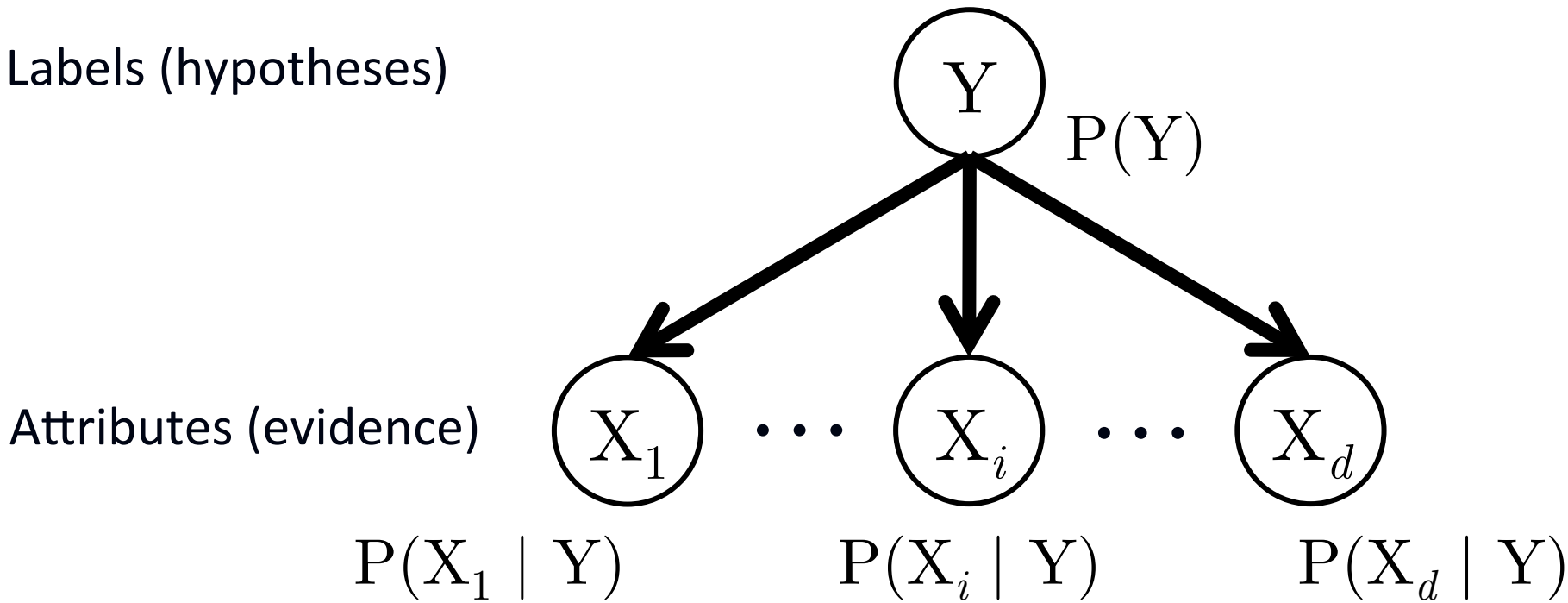
People Words



Animal Words

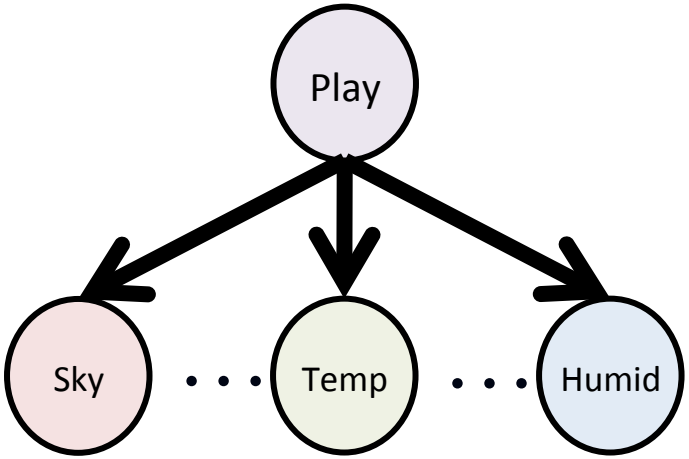
Pairwise Classification
Accuracy: 85%

The Naïve Bayes Graphical Model



- Nodes denote random variables
- Edges denote dependency
- Each node has an associated conditional probability table (CPT), conditioned upon its parents

Example Using NB for Classification



Play?	P(Play)
yes	3/4
no	1/4

Temp	Play?	P(Temp | Play)
warm	*yes*	4/5
cold	*yes*	1/5
warm	*no*	1/3
cold	*no*	2/3

Sky	Play?	P(Sky | Play)
sunny	*yes*	4/5
rainy	*yes*	1/5
sunny	*no*	1/3
rainy	*no*	2/3

Humid	Play?	P(Humid | Play)
high	*yes*	3/5
norm	*yes*	2/5
high	*no*	2/3
norm	*no*	1/3

Predict label for:

$\mathbf{x}$ = (rainy, warm, normal)

$$P(\text{play} \mid \mathbf{x}) \propto \log P(\text{play}) + \log P(\text{rainy} \mid \text{play}) + \log P(\text{warm} \mid \text{play}) + \log P(\text{normal} \mid \text{play})$$

$$\propto \log 3/4 + \log 1/5 + \log 4/5 + \log 2/5 = \boxed{-1.319 \quad \text{predict PLAY}}$$

$$P(\neg\text{play} \mid \mathbf{x}) \propto \log P(\neg\text{play}) + \log P(\text{rainy} \mid \neg\text{play}) + \log P(\text{warm} \mid \neg\text{play}) + \log P(\text{normal} \mid \neg\text{play})$$

$$\propto \log 1/4 + \log 2/3 + \log 1/3 + \log 1/3 = -1.732$$

Naïve Bayes Summary

Advantages:

- Fast to train (single scan through data)
- Fast to classify
- Not sensitive to irrelevant features
- Handles real and discrete data
- Handles streaming data well

Disadvantages:

- Assumes independence of features

Document Classification

Document Classification

PROBLEM SETTING

Given:

- Representation of a document
- Set of classes 1,...,K

Classes:

(AI) (Programming) (HCI)

ML | Planning | Semantics | Garb.Coll. | Multimedia | GUI

Training Data:						
	learning	planning	programming	garbage	...	...
	intelligence	temporal	semantics	collection		
	algorithm	reasoning	language	memory		
	reinforcement	plan	proof...	optimization		
	network...	language...		region...		

Based on slide by Chris Manning

Document Classification

Test Data:

"planning
language
proof
intelligence"

PROBLEM SETTING

Given:

- Representation of a document
- Set of classes 1,...,K

Determine:

- Class to which document d belongs

Classes:

(AI) (Programming) (HCI)

ML | Planning | Semantics | Garb.Coll. | Multimedia | GUI

Training Data:

ML	Planning	Semantics	Garb.Coll.	Multimedia	GUI
learning	planning	programming	garbage	...	...
intelligence	temporal	semantics	collection		
algorithm	reasoning	language	memory		
reinforcement	plan	proof...	optimization		
network...	language...		region...		

Based on slide by Chris Manning

Text Classification: Examples

- Classify news stories as *World, US, Business, SciTech, Sports, etc.*
- Add terms to Medline abstracts (e.g. "Conscious Sedation" [E03.250])
- Classify business names by industry
- Classify student essays as *A/B/C/D/F*
- Classify email as *Spam/Other*
- Classify email to tech staff as *Mac/Windows/ ...*
- Classify pdf files as *ResearchPaper/Other*
- Determine authorship of documents
- Classify movie reviews as *Favorable/Unfavorable/Neutral*
- Classify technical papers as *Interesting/Uninteresting*
- Classify jokes as *Funny/NotFunny*
- Classify websites of companies by Standard Industrial Classification (SIC) code

Based on slide by William Cohen

Text Classification: Examples

- Best-studied benchmark: *Reuters-21578* newswire stories
  - 9603 train, 3299 test documents, 80-100 words each, 93 classes

ARGENTINE 1986/87 GRAIN/OILSEED REGISTRATIONS

BUENOS AIRES, Feb 26

Argentine grain board figures show crop registrations of grains, oilseeds and their products to February 11, in thousands of tonnes, showing those for future shipments month, 1986/87 total and 1985/86 total to February 12, 1986, in brackets:

- Bread wheat prev 1,655.8, Feb 872.0, March 164.6, total 2,692.4 (4,161.0).
- Maize Mar 48.0, total 48.0 (nil).
- Sorghum nil (nil)
- Oilseed export registrations were:
- Sunflowerseed total 15.0 (7.9)
- Soybean May 20.0, total 20.0 (nil)

The board also detailed export registrations for subproducts, as follows....

→ Categories: grain, wheat (of 93 binary choices)

Slide by William Cohen

Spam Filtering

From: "" <takworlld@hotmail.com>
Subject: real estate is the only way... gem oalvgkay

Anyone can buy real estate with no money down

Stop paying rent TODAY !

There is no need to spend hundreds or even thousands for similar courses

I am 22 years old and I have already purchased 6 properties using the methods outlined in this truly INCREDIBLE ebook.

Change your life NOW !

==========
Click Below to order:
http://www.wholesaledaily.com/sales/nmd.htm
==========

Bag of Words Representation

What is the ~~best~~ representation for documents?

simplest, yet useful

Idea: Treat each document as a sequence of words

- Assume that word positions are generated *independently*

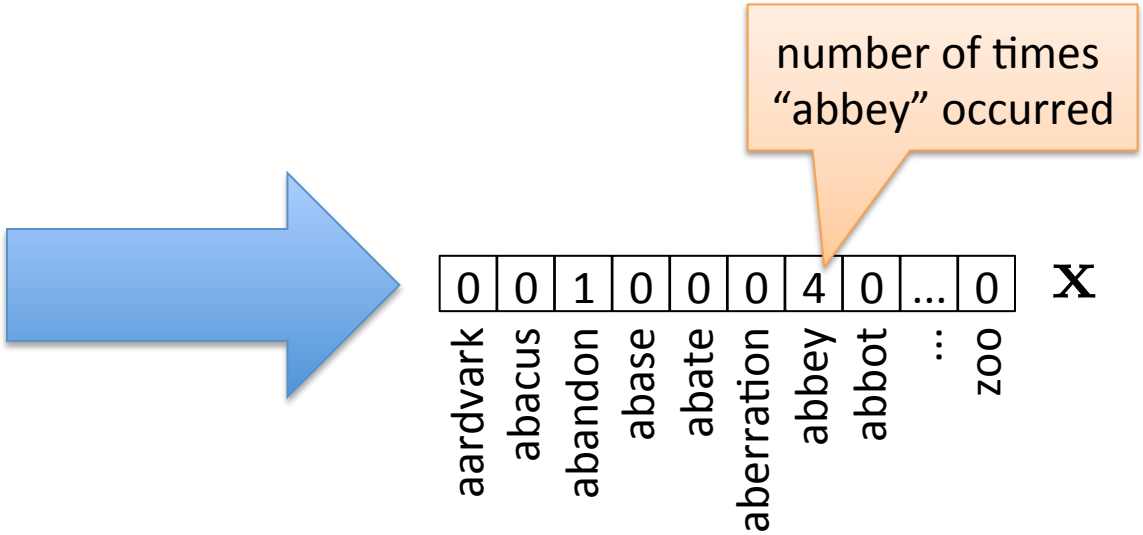
<u>Dictionary</u>: set of all possible words

- Compute over set of documents
- Use Webster's dictionary, etc.

Bag of Words Representation

Represent document d as a vector of word counts $\mathbf{x}$

- x_j represents the count of word j in the document
  - $\mathbf{x}$ is sparse (few non-zero entries)



What are Some Issues with the Bag of Words Representation?

- Documents have different lengths
- Some words aren't meaningful to represent the content of a document
  - e.g., "the", "a", etc.
- Rare words may be more meaningful than common words

Need a better representation for the documents...

Eliminate Stop Words

Common, “less-meaningful” words are called <u>stop words</u>

- Delete stop words before doing any document processing

Example stop words:

a	because	does	haven't	i	more	our	some	they'll	we'll	why
about	been	doesn't	having	i'd	most	ours	such	they're	we're	why's
above	before	doing	he	i'll	mustn't	ourselves	than	they've	we've	with
after	being	don't	he'd	i'm	my	out	that	this	were	won't
again	below	down	he'll	i've	myself	over	that's	those	weren't	would
against	between	during	he's	if	no	own	the	through	what	wouldn't
all	both	each	her	in	nor	same	their	to	what's	you
am	but	few	here	into	not	shan't	theirs	too	when	you'd
an	by	for	here's	is	of	she	them	under	when's	you'll
and	can't	from	hers	isn't	off	she'd	themselves	until	where	you're
any	cannot	further	herself	it	on	she'll	then	up	where's	you've
are	could	had	him	it's	once	she's	there	very	which	your
aren't	couldn't	hadn't	himself	its	only	should	there's	was	while	yours
as	did	has	his	itself	or	shouldn't	these	wasn't	who	yourself
at	didn't	hasn't	how	let's	other	so	they	we	who's	yourselves
be	do	have	how's	me	ought		they'd	we'd	whom	

Term Frequency

Term frequency $tf_{t,d}$ is some measure of importance of term t to document d

Boolean: $tf_{t,d} = 1$ if t occurs in d, 0 otherwise

Raw Counts: $tf_{t,d} = c_{t,d}$

- $c_{t,d}$ is the number of times t occurs in d

Log-Scaled Counts: $tf_{t,d} = \begin{cases} 1 + \log c_{t,d} & \text{if } c_{t,d} > 0 \\ 0 & \text{otherwise} \end{cases}$

- Reduces relative impact of frequent terms

Normalized Counts: $tf_{t,d} = c_{t,d} / |d|$

- Normalize raw counts by length of document $|d|$

Words → Counts → Weight Matrix

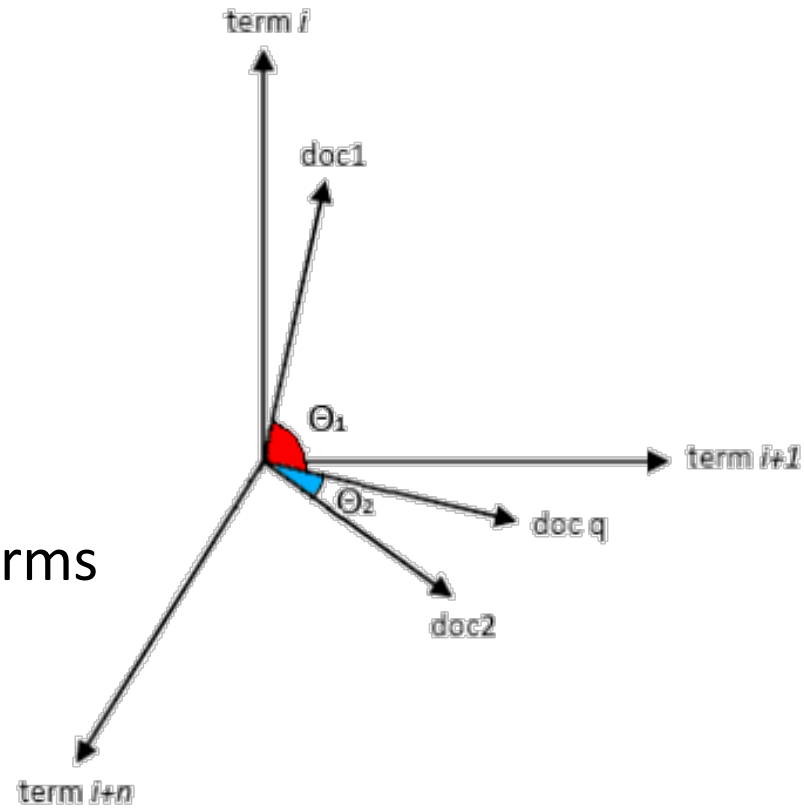
	Antony and Cleopatra	Julius Caesar	The Tempest	Hamlet	Othello	Macbeth
Antony	5.25	3.18	0	0	0	0.35
Brutus	1.21	6.1	0	1	0	0
Caesar	8.59	2.54	0	1.51	0.25	0
Calpurnia	0	1.54	0	0	0	0
Cleopatra	2.85	0	0	0	0	0
mercy	1.51	0	1.9	0.12	5.25	0.88
worser	1.37	0	0.11	4.15	0.25	1.95

Each document is now represented by a real-valued vector of $|D|$ TF-IDF weights

Based on slide by P. Nayak and P. Raghavan

Documents as Vectors

- So we have a $|D|$-dimensional vector space
  - Terms are axes of the space
  - Documents are points or vectors in this space
- Very high-dimensional:
  - Over 1M words in english
  - More if we allow non-word terms
- Very sparse vectors

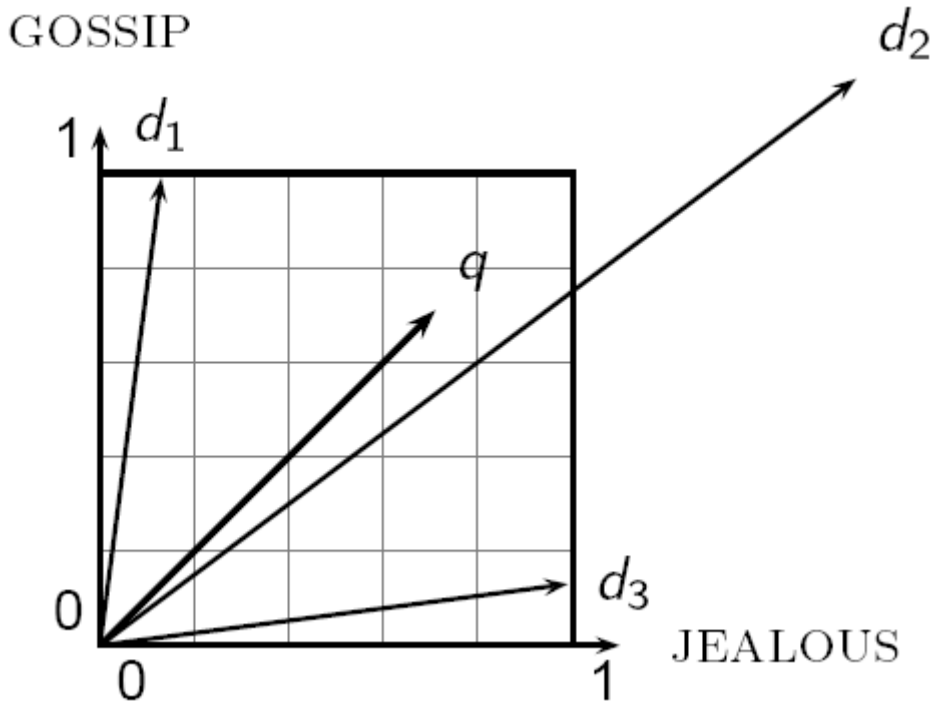


- **Idea:** Measure similarity of documents via proximity in the vector space

Based on slide by P. Nayak and P. Raghavan

Why Euclidean Distance is a Bad Idea

- Because Euclidean distance is large for vectors of different lengths



$||\mathbf{q} - \mathbf{d}_2||_2$ is large, even though the distribution of terms in the query $\mathbf{q}$ and the distribution of terms in the document $\mathbf{d}_2$ are very similar

Based on slide by P. Nayak and P. Raghavan

Use Angle Instead of Distance

Thought experiment:

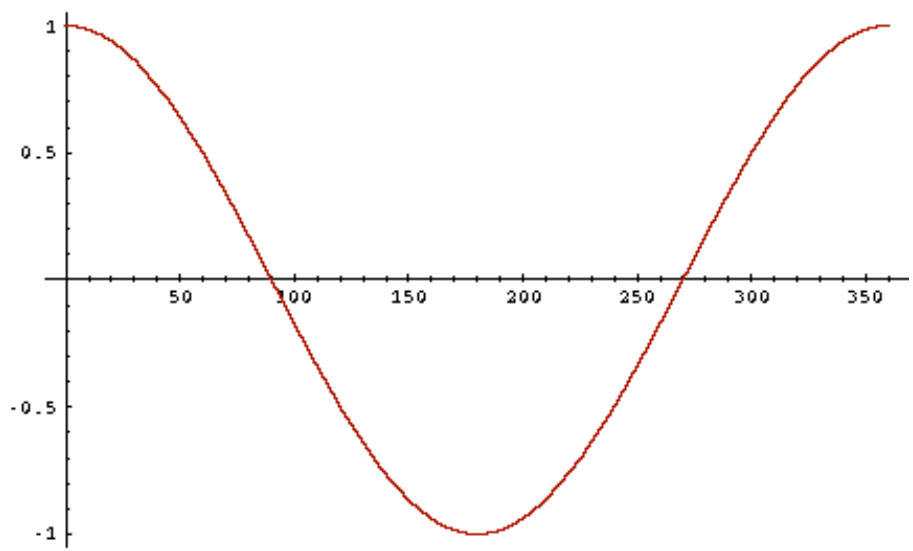
- Take a document d and append it to itself, creating a new document d'
- Semantically, d and d' have the same content
- But, the Euclidean distance between the two documents can be quite large
- However, note that the angle between the two documents is 0, corresponding to maximal similarity

Key Idea: Measure similarity based on angle of vector

Based on slide by P. Nayak and P. Raghavan

From Angles to Cosines

- The following two notions are equivalent:
  - Measure similarity between documents d_i and d_j via <u>decreasing</u> order of the angle between $\mathbf{x}_i$ and $\mathbf{x}_j$
  - Measure similarity in <u>increasing</u> order of cosine($\mathbf{x}_i$, $\mathbf{x}_j$)
- Cosine is a monotonically decreasing function for the interval [0°, 180°]



Based on slide by P. Nayak and P. Raghavan

Length Normalization

- A vector can be (length-) normalized by dividing each of its components by its length (the L_2 norm)

$$\mathbf{x} = \mathbf{x} \,/\, ||\mathbf{x}||_2$$

- Dividing a vector by its L_2 norm makes it a unit (length) vector (on surface of unit hypersphere)

- Effect on the two documents d and d' (d appended to itself) from earlier slide: they have identical vectors after length-normalization
  - Long and short documents now have comparable weights

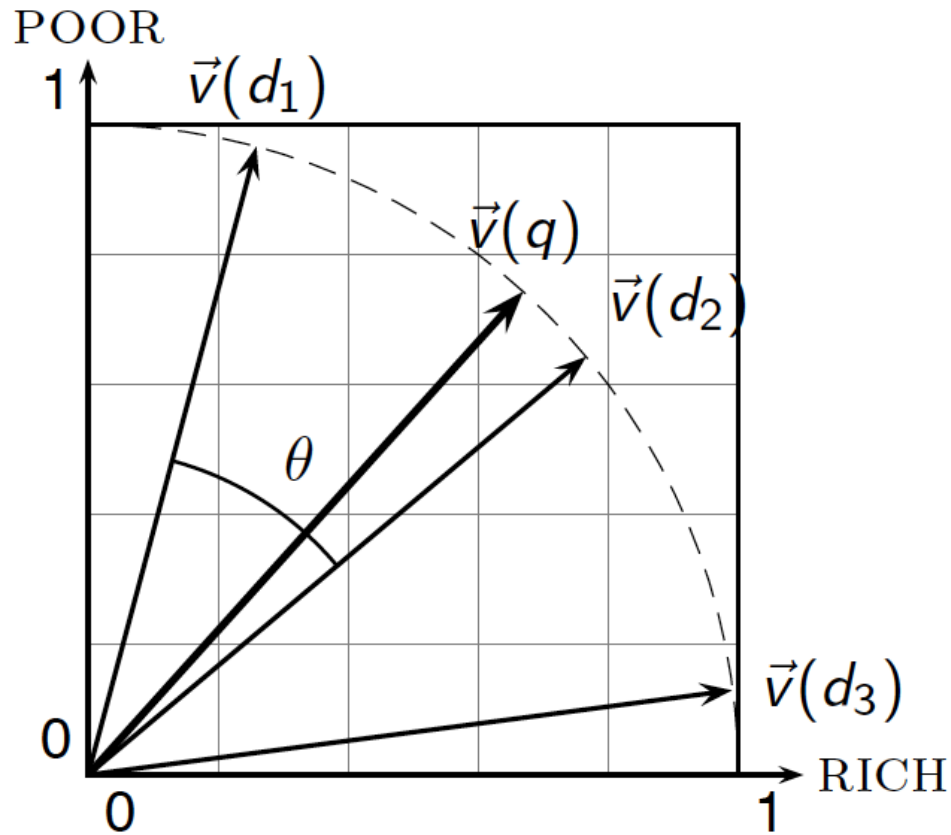
Based on slide by P. Nayak and P. Raghavan

Cosine Similarity

$\mathbf{x}_i$ and $\mathbf{x}_j$ are TF-IDF weight vectors

$$\cos(\mathbf{x}_i, \mathbf{x}_j) = \frac{\mathbf{x}_i \cdot \mathbf{x}_j}{|\mathbf{x}_i||\mathbf{x}_j|} = \underbrace{\frac{\mathbf{x}_i}{|\mathbf{x}_i|}}_{} \cdot \underbrace{\frac{\mathbf{x}_j}{|\mathbf{x}_j|}}_{}$$

unit-length vectors



$\cos(\mathbf{x}_i, \mathbf{x}_j)$ is the cosine similarity of $\mathbf{x}_i$ and $\mathbf{x}_j$

- Equivalently, the cosine of the angle between $\mathbf{x}_i$ and $\mathbf{x}_j$
- For unit vectors, cosine similarity is simply the dot product

Based on slide by P. Nayak and P. Raghavan

Example: Cosine Similarity Amongst Three Documents

Term Frequencies (counts)

term	SaS	PaP	WH
affection	115	58	20
jealous	10	7	11
gossip	2	0	6
wuthering	0	0	38

SaS: *Sense and Sensibility*

PaP: *Pride and Prejudice*

WH: *Wuthering Heights*?

Note: To simplify this example, we don't do IDF weighting

Example: Cosine Similarity Amongst Three Documents

Log-Scaled Counts

term	SaS	PaP	WH
affection	3.06	2.76	2.30
jealous	2.00	1.85	2.04
gossip	1.30	0	1.78
wuthering	0	0	2.58

After Length Normalization

term	SaS	PaP	WH
affection	0.789	0.832	0.524
jealous	0.515	0.555	0.465
gossip	0.335	0	0.405
wuthering	0	0	0.588

cos(SaS,PaP) $\approx 0.789 \times 0.832 + 0.515 \times 0.555 + 0.335 \times 0.0 + 0.0 \times 0.0$
≈ 0.94

cos(SaS,WH) ≈ 0.79

cos(PaP,WH) ≈ 0.69

Based on slide by P. Nayak and P. Raghavan