CSDS 440: Machine Learning

Soumya Ray (he/him, sray@case.edu)

Olin 516

Office hours T, Th 11:15-11:45 or by appointment

Zoom link

Today

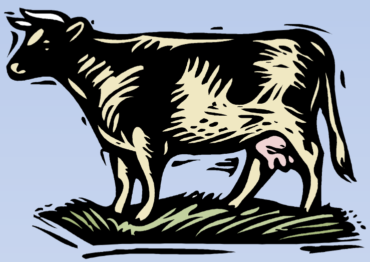
- Foundations of machine Learning

Supervised Learning

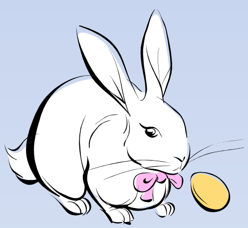
- Examples *E* are annotated with target concept's output by a teacher/oracle
- Learning system must find a concept that matches annotations (*P*)
- Example: learn to recognize animals

Supervised Learning

tiger



cow



elephant

starfish

Note: Annotation received by learner does not need to be correct!!

Other Learning Paradigms

- Unsupervised Learning
- Semi-supervised Learning
- Active Learning
- Transductive Learning
- Transfer Learning
- Structured Prediction
- Reinforcement Learning
- Preference Learning (Ranking)
- “Few-shot” learning

Example Representation

- What is the *internal representation* of an example in a learning system?

- Representation choice affects reasoning and the choice of hypothesis space, and the cost of learning

Feature Vector Representation

- Examples are attribute-value pairs (note “feature”==“attribute”)
- Number of attributes are fixed
- Can be written as an n-by-m matrix

	Attribute$_1$	**Attribute$_2$**	**Attribute$_3$**
Example$_1$	Value$_{11}$	Value$_{12}$	Value$_{13}$
Example$_2$	Value$_{21}$	Value$_{22}$	Value$_{23}$
Example$_3$	Value$_{31}$	Value$_{32}$	Value$_{33}$

Feature Vectors

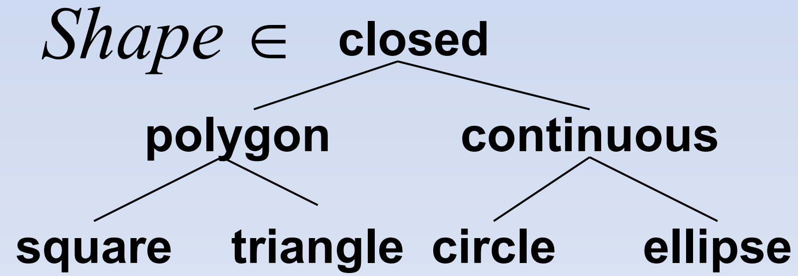
Example

	Has-fur?	Long-Teeth?	Scary?
Animal_1	Yes	No	No
Animal_2	No	Yes	Yes
Animal_3	Yes	Yes	Yes

Types of Features

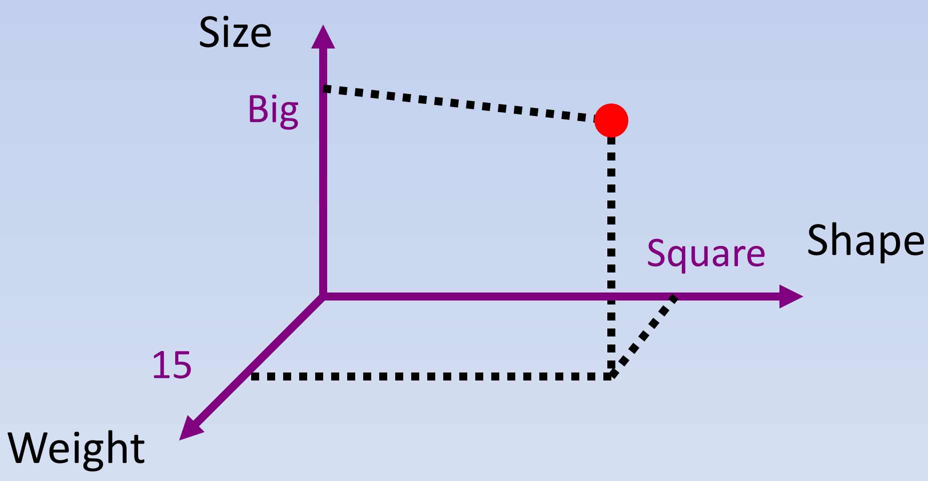
- Discrete, Nominal
- Continuous
- Discrete, Ordered
- Hierarchical

- *Color* $\in$ *(red, blue, green)*
- *Height*
- *Size* $\in$ *(small, medium, large)*
- *Shape* $\in$



Feature Space

- We can think of examples embedded in an n dimensional vector space



Other Example Representations

- Relational representation
- Multiple-instance representation
- Sequential representation
- Multi-view representation

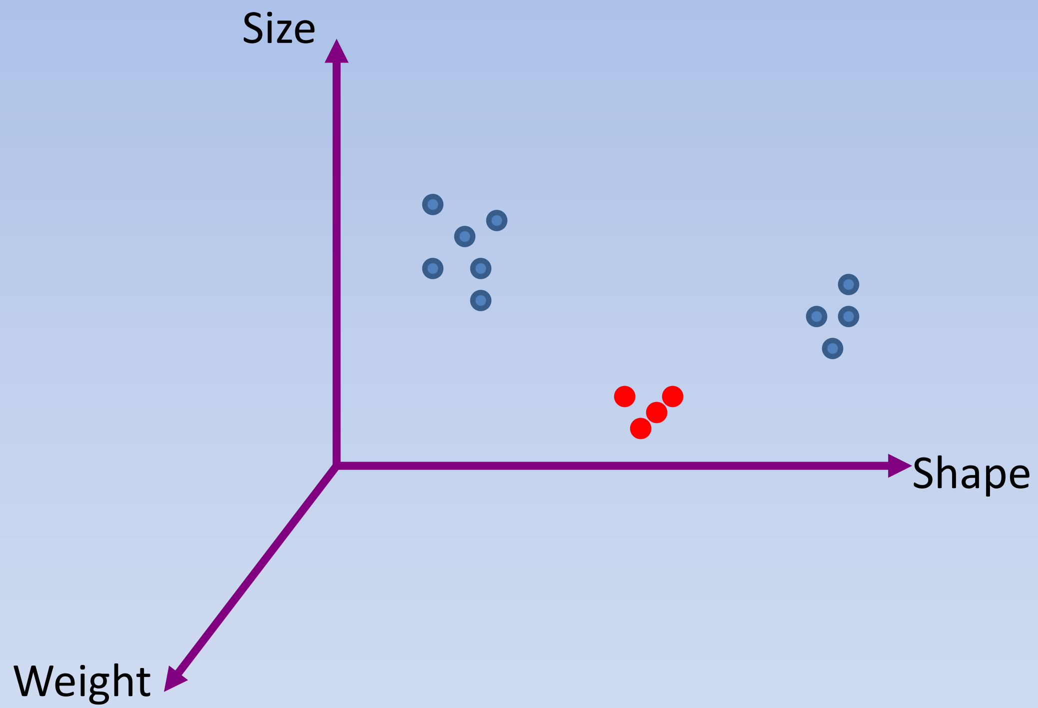
The Binary Classification Problem

- Simplest supervised learning problem
- Target concept assigns one of two labels ("*positive*" or "*negative*") to all examples---the class label
- Can extend to "multiclass", "regression", "multi-label" problems

Example

	X			Y	
	Has-fur?	Long-Teeth?	Scary?	*Lion?*	
Animal$_1$	Yes	No (x_{ij})	No	No	$(\boldsymbol{x}_i, y_i)$
Animal$_2$	No	Yes	Yes	No	
Animal$_3$	Yes	Yes	Yes	Yes	

Example in Feature Space

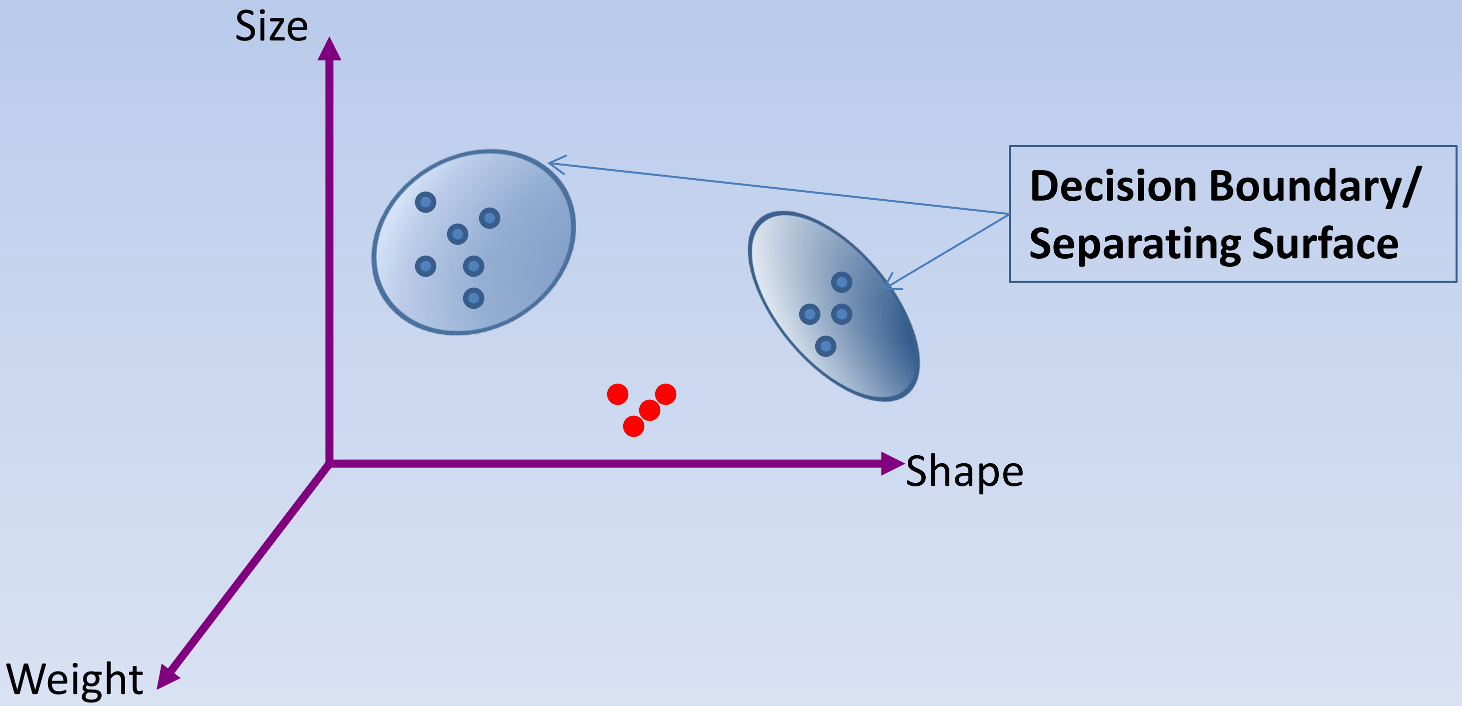


The Learning Problem

- Given: A binary classification problem
- Do: Produce a "classifier" (concept) that assigns a label to a new example

Binary Classifier Concept Geometry

- (Union of) N-dimensional volume(s) in feature space (possibly a disjoint collection)



Decision Tree Induction (Ch 3, Mitchell)

- A “classical” (1980s) family of machine learning algorithms for classification

- Widely used and extremely popular, available in nearly all ML toolkits

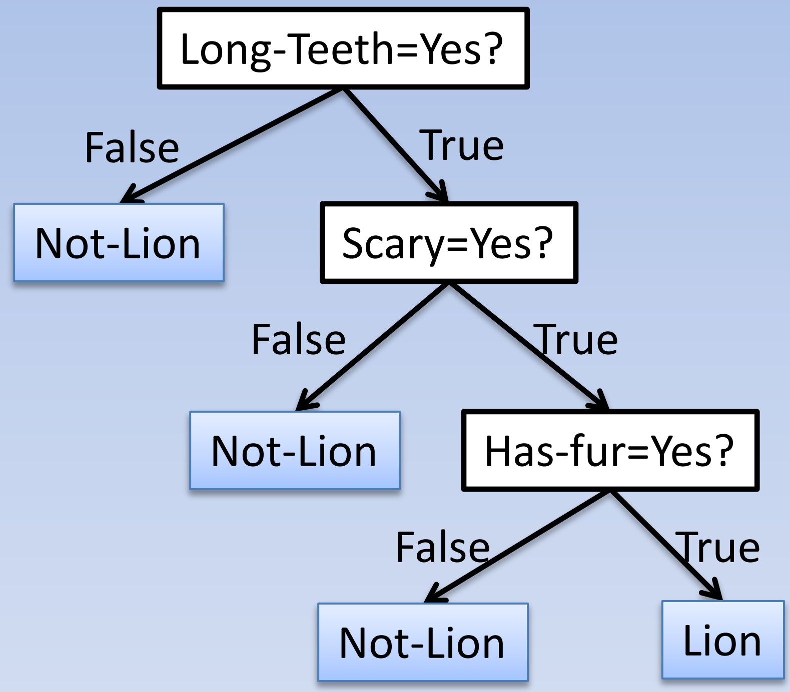
What is a Decision Tree?

- Tree: directed acyclic graph, each node has at most one parent
- Internal nodes: Tests on attributes
- Leaves: Class labels

Example

	Has-fur?	Long-Teeth?	Scary?	*Lion?*
Animal_1	Yes	No	No	No
Animal_2	No	Yes	Yes	No
Animal_3	Yes	Yes	Yes	Yes

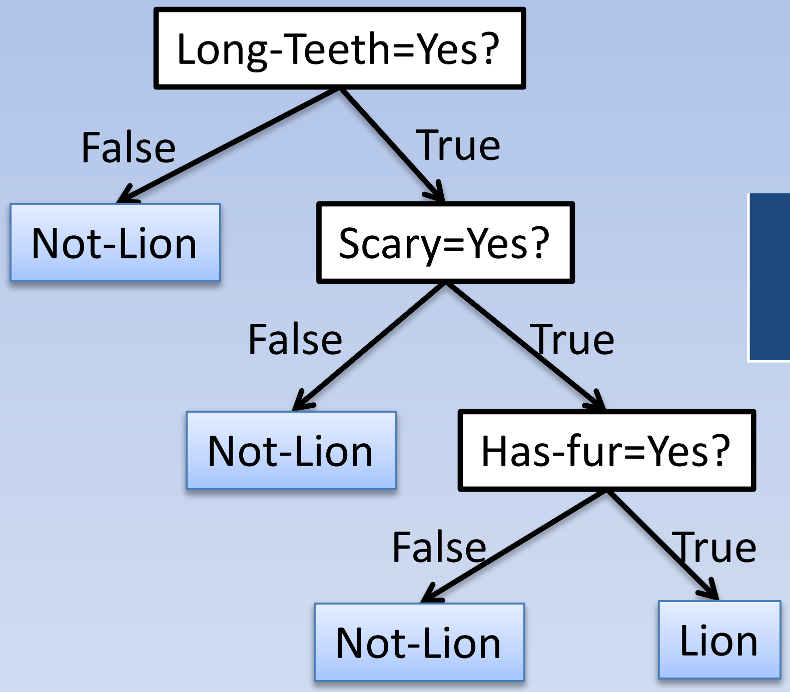
Example


Long-Teeth=Yes?
False
True
Not-Lion
Scary=Yes?
False
True
Not-Lion
Has-fur=Yes?
False
True
Not-Lion
Lion

Classification with a decision tree

- Suppose we are given a tree and a new example
- Starting at the root, check each attribute test
- This identifies a path through the tree, follow this until we reach a leaf
- Assign the class label in the leaf

Example



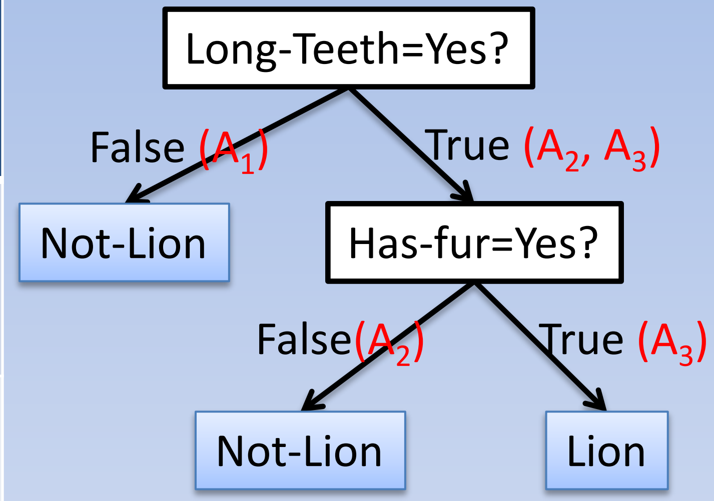
	Has-fur?	Long-Teeth?	Scary?
$Animal_1$	Yes	Yes	No

Decision Tree Induction

- Given a set of examples, produce a decision tree
- Decision tree induction works using the idea of recursive partitioning
  - At each step, the algorithm will choose an attribute test
    - If no attribute looks good, return
  - The chosen test will partition the examples into disjoint partitions
  - The algorithm will then recursively call itself on each partition until
    - a partition only has data from one class (pure node) OR
    - it runs out of attributes

Example

	Has-fur?	Long-Teeth?	Scary?	*Lion?*
Animal$_1$	Yes	No	No	No
Animal$_2$	No	Yes	Yes	No
Animal$_3$	Yes	Yes	Yes	Yes



Choosing an Attribute

- Which attribute should we choose to test first?
  - Ideally, the one that is "most predictive" of the class label
    - i.e., the one that gives us the "most information" about what the label should be

- This idea is captured by the "(Shannon) entropy" of a random variable

Entropy of a Random Variable

- Suppose a random variable X has density $p(x)$. Its (Shannon) "entropy" is defined by:

$$H(X) = E(-\log_2(p(X)))$$

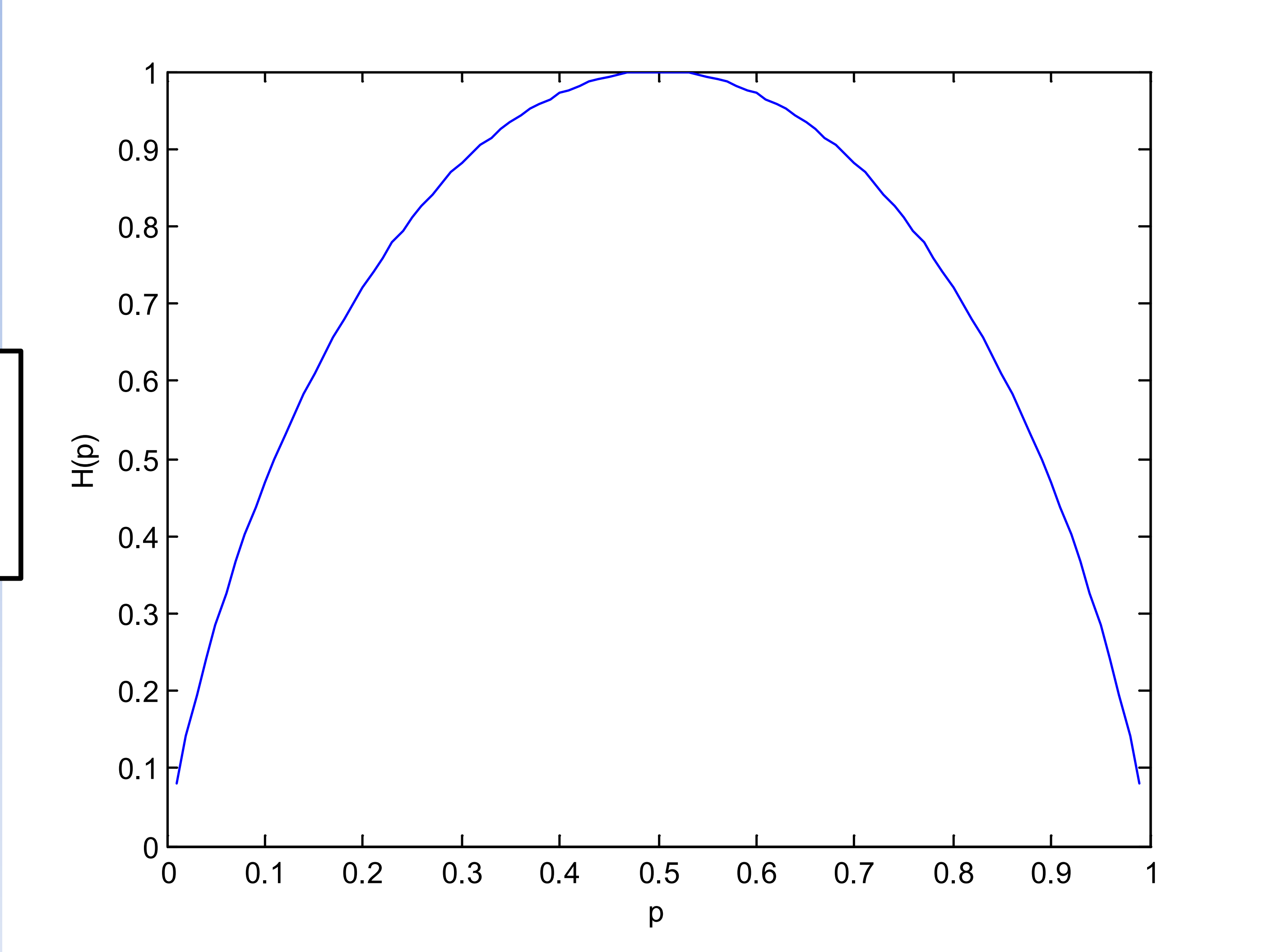
$$= -\sum_x p(X = x)\log_2(p(X = x))$$

- Note: 0log(0) = 0 .

Example

- Suppose X has two values, 0 and 1, and pdf $p(0)=0.5, p(1)=0.5$
  - Then $H(X)=$?
- Suppose X has two values, 0 and 1, and pdf $p(0)=0.99, p(1)=0.01$
  - Then $H(X)=$? 0.081
- Suppose X has two values, 0 and 1, and pdf $p(0)=0.01, p(1)=0.99$
  - Then $H(X)=$?

Entropy of a Bernoulli r.v.



Entropy is typically denoted by $H(.)$

What is entropy?

- Measure of "information content" in a distribution
- Suppose we wanted to describe an r.v. X with n values and distribution $p(X=x)$
  - Shortest lossless description takes $-log_2(p(x))$ bits for each x
  - So entropy is the expected length of the shortest lossless description of the r.v.

Source Coding Theorem, Claude Shannon 1948

What's the connection?

- Entropy measures the *information content* of a random variable

- Suppose we treat the class variable, Y, as a random variable and measure its entropy

- Then we measure its entropy after partitioning the examples with an attribute X

The Entropy Connection

- The difference will be a measure of the "information gained" about Y by partitioning the examples with X

- So if we can choose the attribute X that maximizes this "information gain", we have found what we needed