Introduction to Machine Learning

Håkan Younes
CS15-381, Fall 2001

What is Machine Learning?

What does it mean for a machine (computer program) to learn?

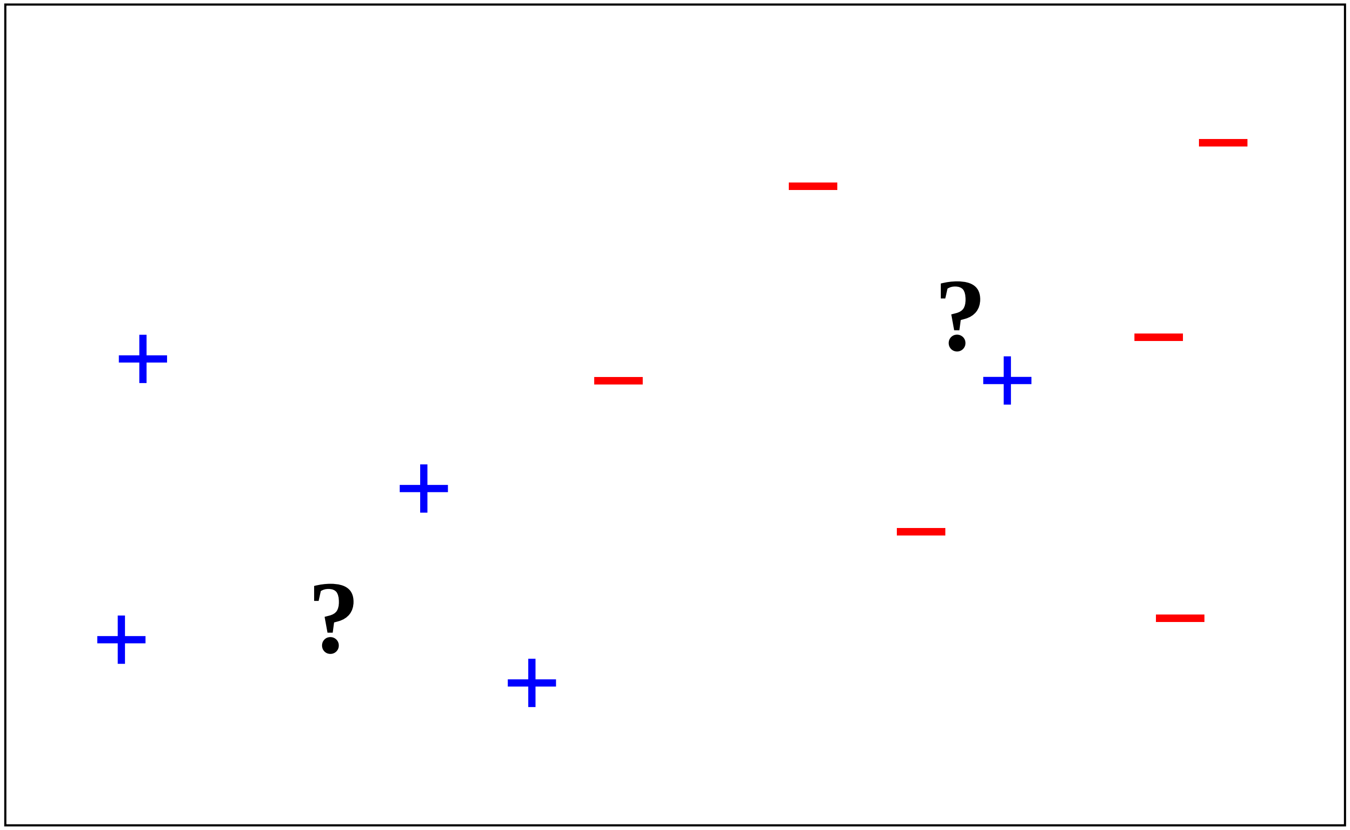
- Machine Learning is the study of computer programs that automatically improve with experience
- Machine Learning is not open-ended discovery

Learning by Caching

- Store instances when they are first encountered
- Retrieve value of instance from memory when queried
- No attempt made to generalize knowledge

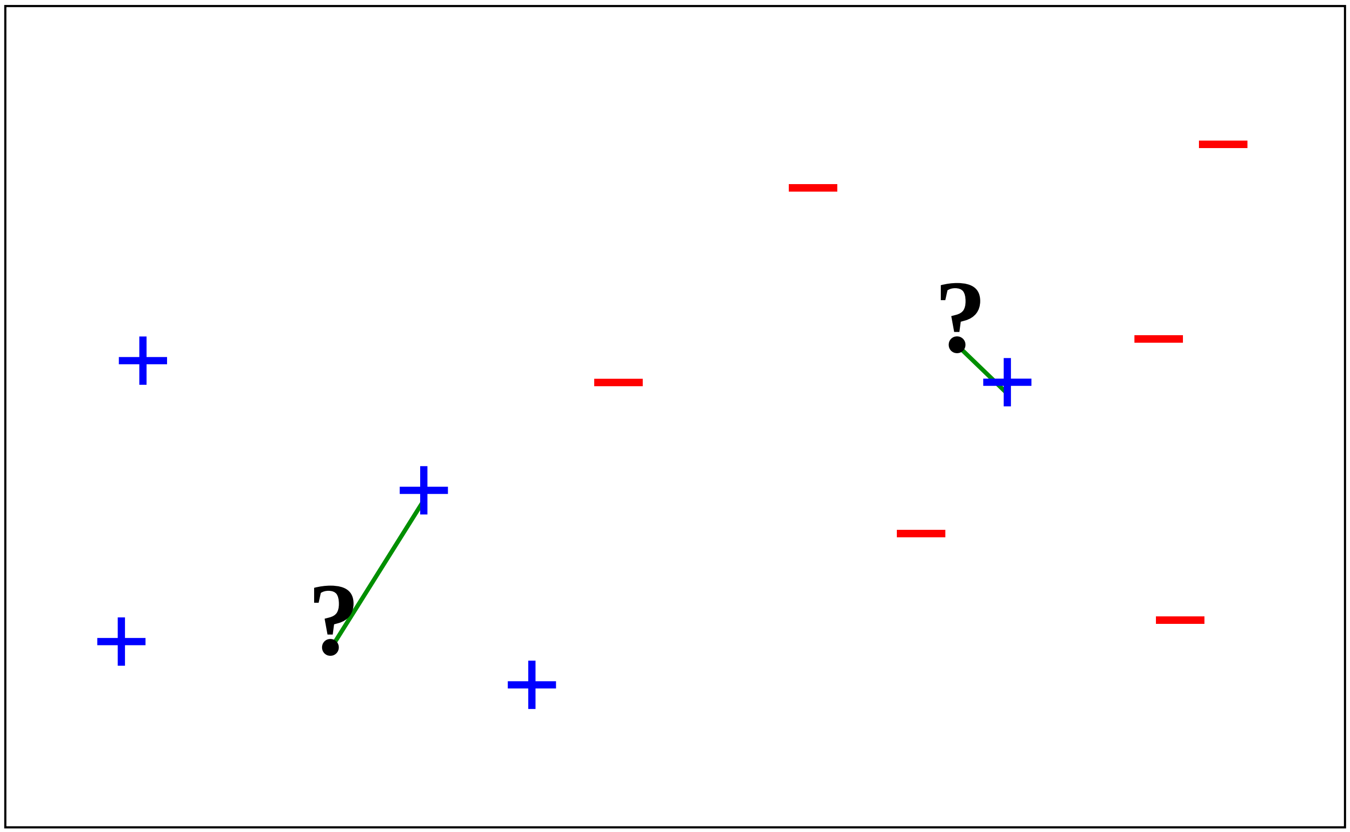
What if query instance is not in memory?

Are the ?'s positive or negative?



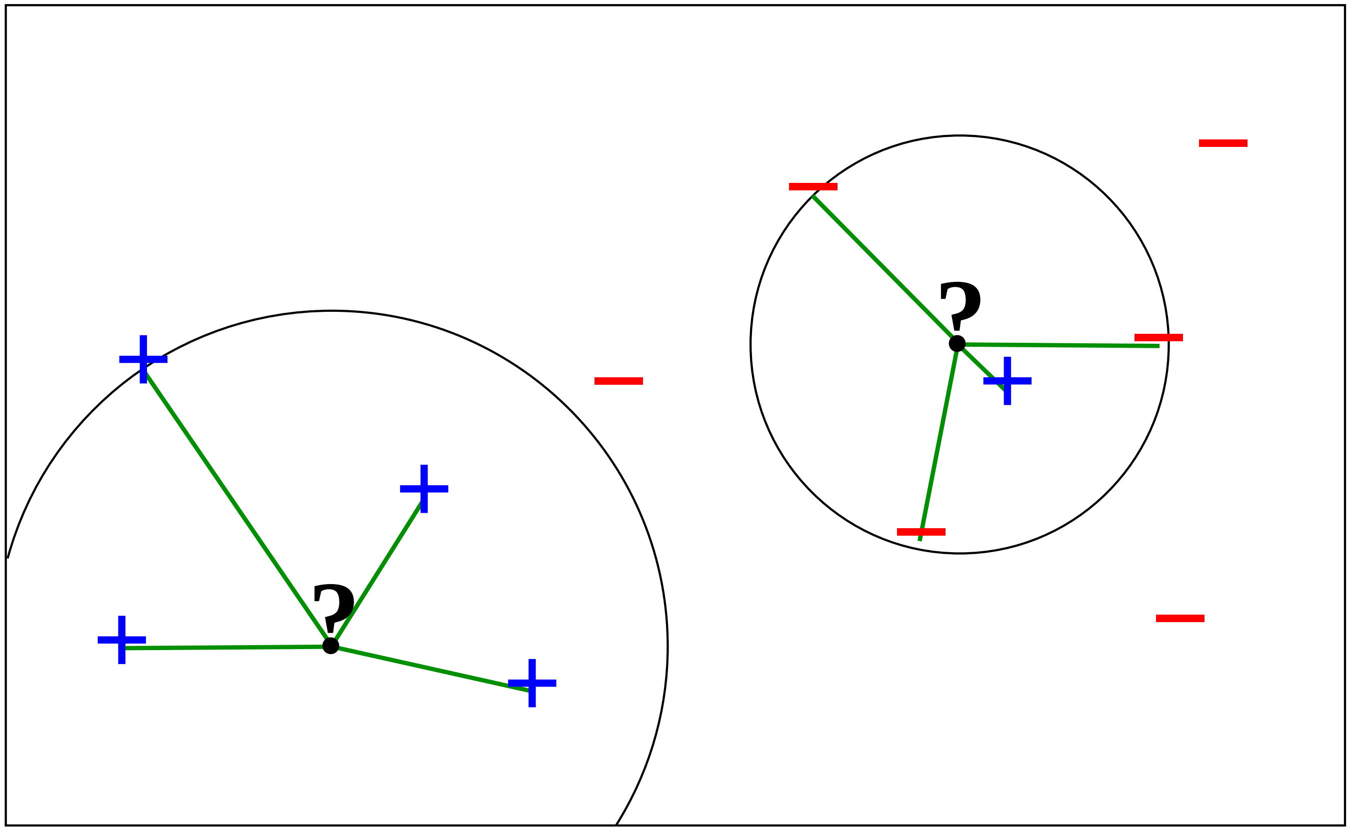
Nearest Neighbor

Assign same value as the nearest neighbor:



k-Nearest Neighbor

Majority of k nearest neighbors (e.g. $k = 4$):



Alternatively: weight influence of neighbors by inverse of square distance

Types of Machine Learning Algorithms

- Classification (learning discrete valued function)
  - e.g. nearest neighbor, perceptrons, decision trees, naive Bayes
- Learning continuous valued function
  - e.g. Locally weighted regression, neural networks
- Reinforcement learning
  - e.g. Q-learning, TD-learning
- Version spaces

Classification

Characteristics of a classification problem:

- Instances are represented by attribute-value pairs
- Target function has discrete output values
  - e.g. boolean classification (*yes* or *no*)

Example: *PlayTennis*

- Concept: *PlayTennis*
- Known instances:

Outlook	*Humidity*	*Wind*	*PlayTennis*
Overcast	*High*	*Weak*	*Yes*
Sunny	*High*	*Weak*	*Yes*
Sunny	*Normal*	*Strong*	*No*
Rain	*High*	*Strong*	*No*
Rain	*Normal*	*Weak*	*Yes*

- Query instance:

$$\langle \mathit{Outlook} = \mathit{Overcast}, \mathit{Humidity} = \mathit{Normal}, \mathit{Wind} = \mathit{Strong} \rangle$$

Nearest Neighbor for *PlayTennis* Example

What distance function should we use?

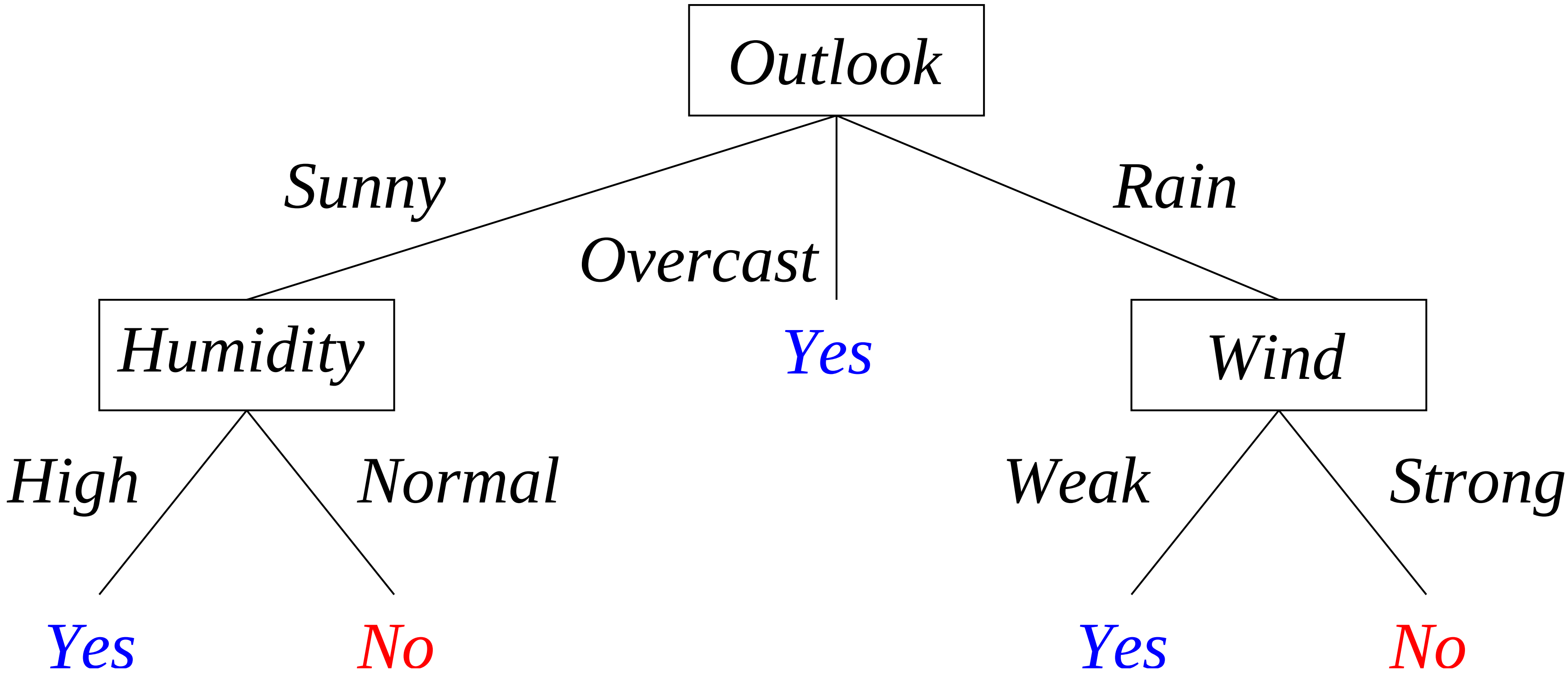
Using number of differing attribute values:

- Nearest neighbor is

  $\langle Outlook = Sunny, Humidity = Normal, Wind = Strong \rangle$

- Value of *PlayTennis*: *No*

Decision Tree for *PlayTennis* Example



Decision Trees

Decision tree representation:

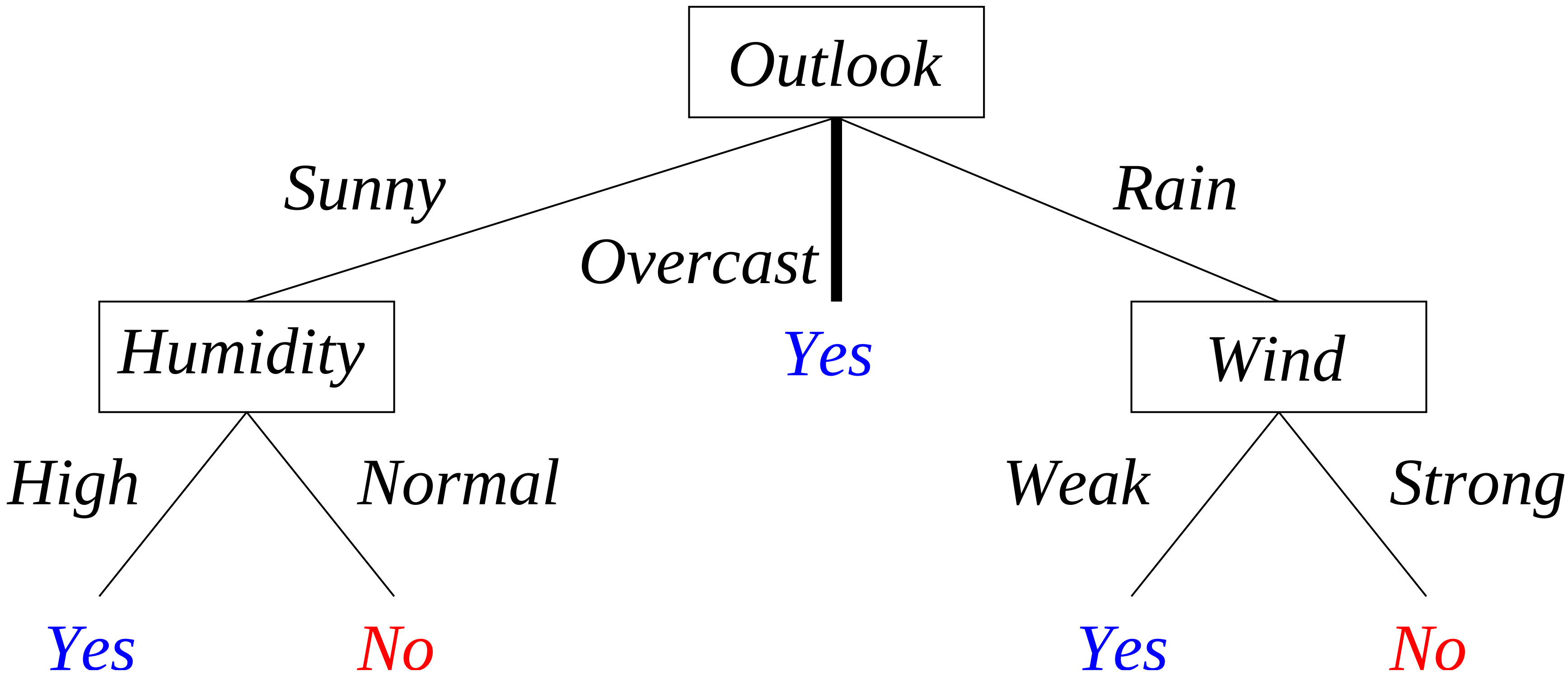
- Internal nodes represent attributes
- Branches out of a node represent possible values
- Terminal nodes represent classifications

Classify an instance by following a path from the root to a terminal node

Decision Tree for *PlayTennis* Example

Query instance:

$$\langle Outlook = Overcast, Humidity = Normal, Wind = Strong \rangle$$



Value of *PlayTennis*: *Yes*

Constructing Decision Trees

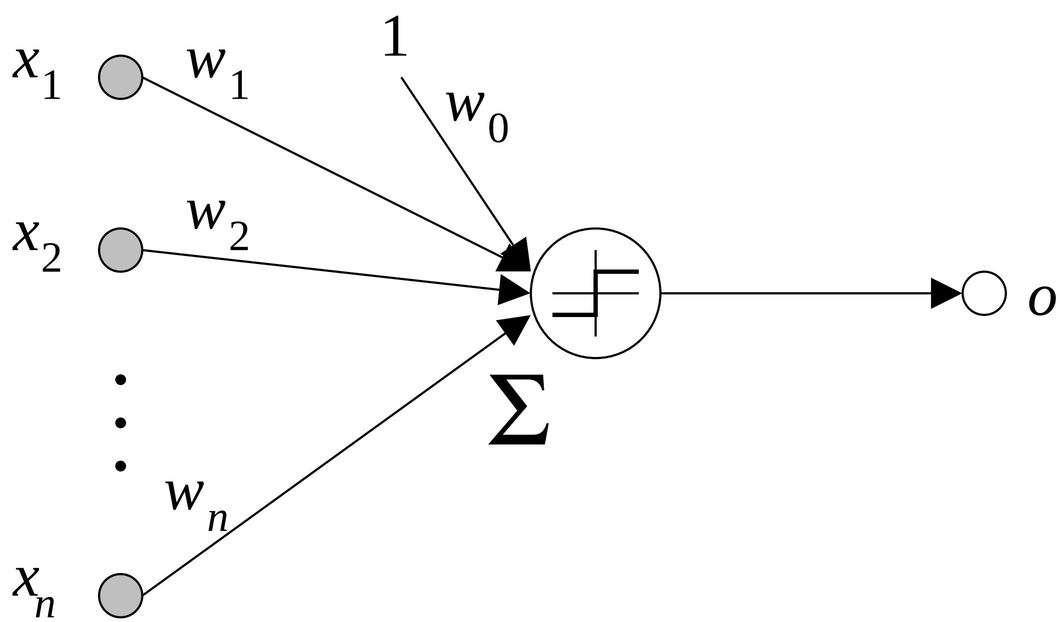
- If all instances have same classification, create terminal node
- Otherwise, choose an attribute a for root node
- For each value v, create decision tree recursively for instances such that value of a is v

Decision Trees and Information Theory

- Decision tree will look different depending on which attribute is chosen for each node
- Small decision tree desirable (Occam's razor)
- NP-hard to find minimal decision tree
- Greedy heuristic: choose attribute that reduces entropy the most

Perceptrons

- Modelled after neurons
- Binary classifier



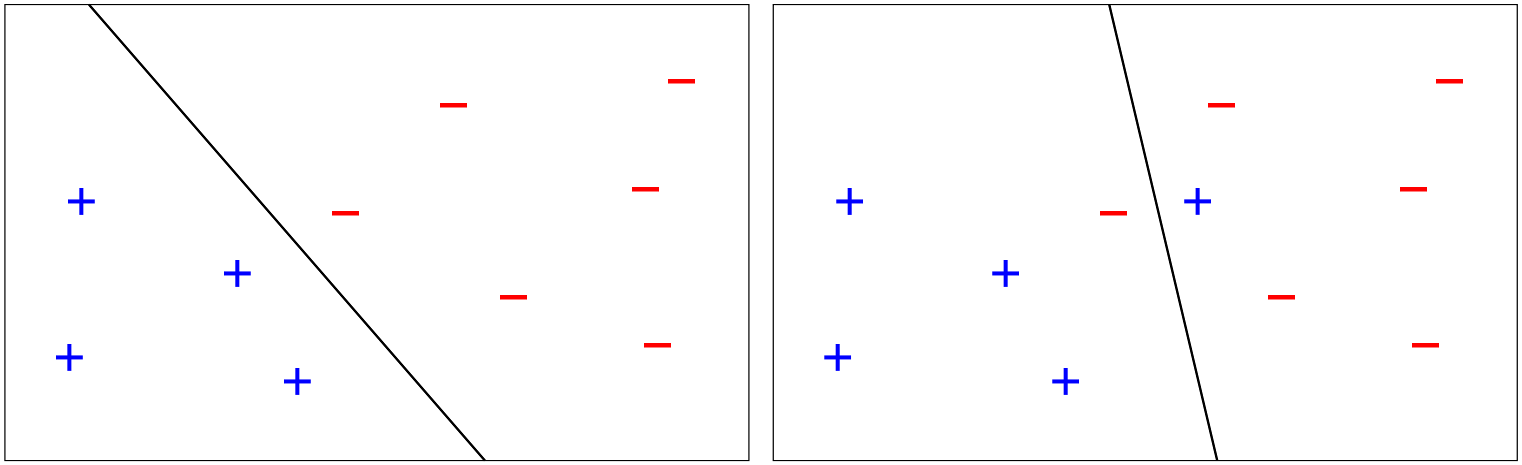
Training a Perceptron

Select weights $w_0, \ldots, w_n$ minimizing the square error summed over the training instances:

$$E = \frac{1}{2} \sum_{x \in D} (o(x) - t(x))^2$$

Linear Separability

Possible hypotheses learned by perceptron for two different training sets:



(cf. linear regression)

Locally Weighted Regression

- Perceptron learning constructs a global hypothesis
- Nearest neighbor learning constructs hypothesis for each query instance
- Distance weighted nearest neighbor can be viewed as locally weighted regression

Reinforcement Learning

Reinforcement learning is used to learn policies

- Robot controller
- Game playing (e.g. checkers, backgammon)

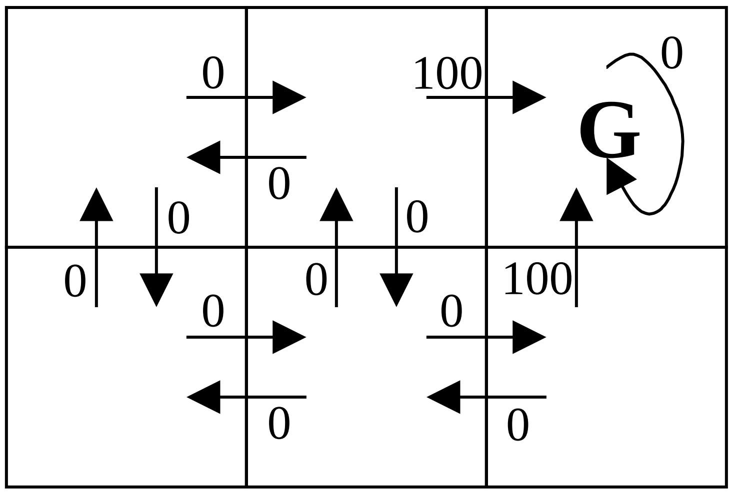
Policies and Reward

- A policy is a function from states to actions

$$\pi : S \rightarrow A$$

- Reward is given when when an action leads to a desirable state

Reinforcement Learning: Example



Issues in Reinforcement Learning

- Delayed reward
- Exploration vs. exploitation
- Partially observable states

Version Spaces

The version space consists of all plausible hypotheses

Question: Can we find a compact representation?

Most General and Most Specific Hypotheses

Positive instance:

$$\langle \mathit{Overcast}, \mathit{High}, \mathit{Weak} \rangle$$

Most general hypothesis:

$$\langle ?, ?, ? \rangle$$

Most specific hypothesis:

$$\langle \mathit{Overcast}, \mathit{High}, \mathit{Weak} \rangle$$

Evaluating Machine Learning Algorithms

- How representative are the training instances?
- Danger of overfitting

Validation Sets

- Split instances into training set and validation set
- Construct hypothesis from training set
- Evaluate performance on validation set

Cross Validation

- Sometimes training instances are hard to come by
- Split instances into k disjoint sets
- Perform k validations with a different validation set each time