Machine Learning

Chapter 18, 21

Some material adopted from notes
by Chuck Dyer

What is learning?

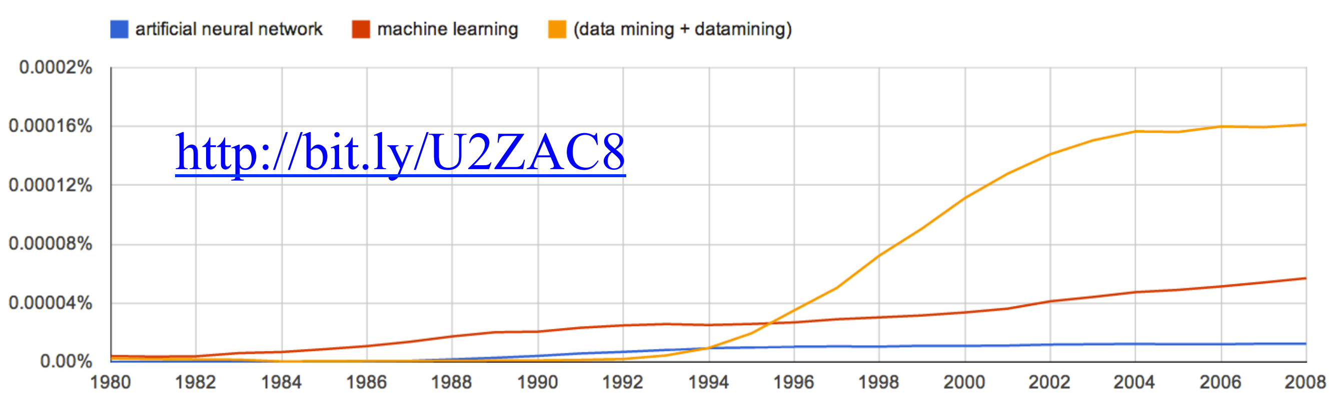
- "Learning denotes changes in a system that ... enable a system to do the same task more efficiently the next time" – Herbert Simon
- "Learning is constructing or modifying representations of what is being experienced" – Ryszard Michalski
- "Learning is making useful changes in our minds" – Marvin Minsky

Why study learning?

- Understand and improve efficiency of **human learning**
  - Use to improve methods for teaching and tutoring people (e.g., better computer-aided instruction)
- **Discover** new things or structure previously unknown
  - Examples: data mining, scientific discovery
- Fill in skeletal or **incomplete specifications in** a domain
  - Large, complex systems can't be completely built by hand & require dynamic updating to incorporate new information
  - Learning new characteristics expands the domain or expertise and lessens the "brittleness" of the system
- Build agents that can **adapt** to users, other agents, and their environment

AI & Learning Today

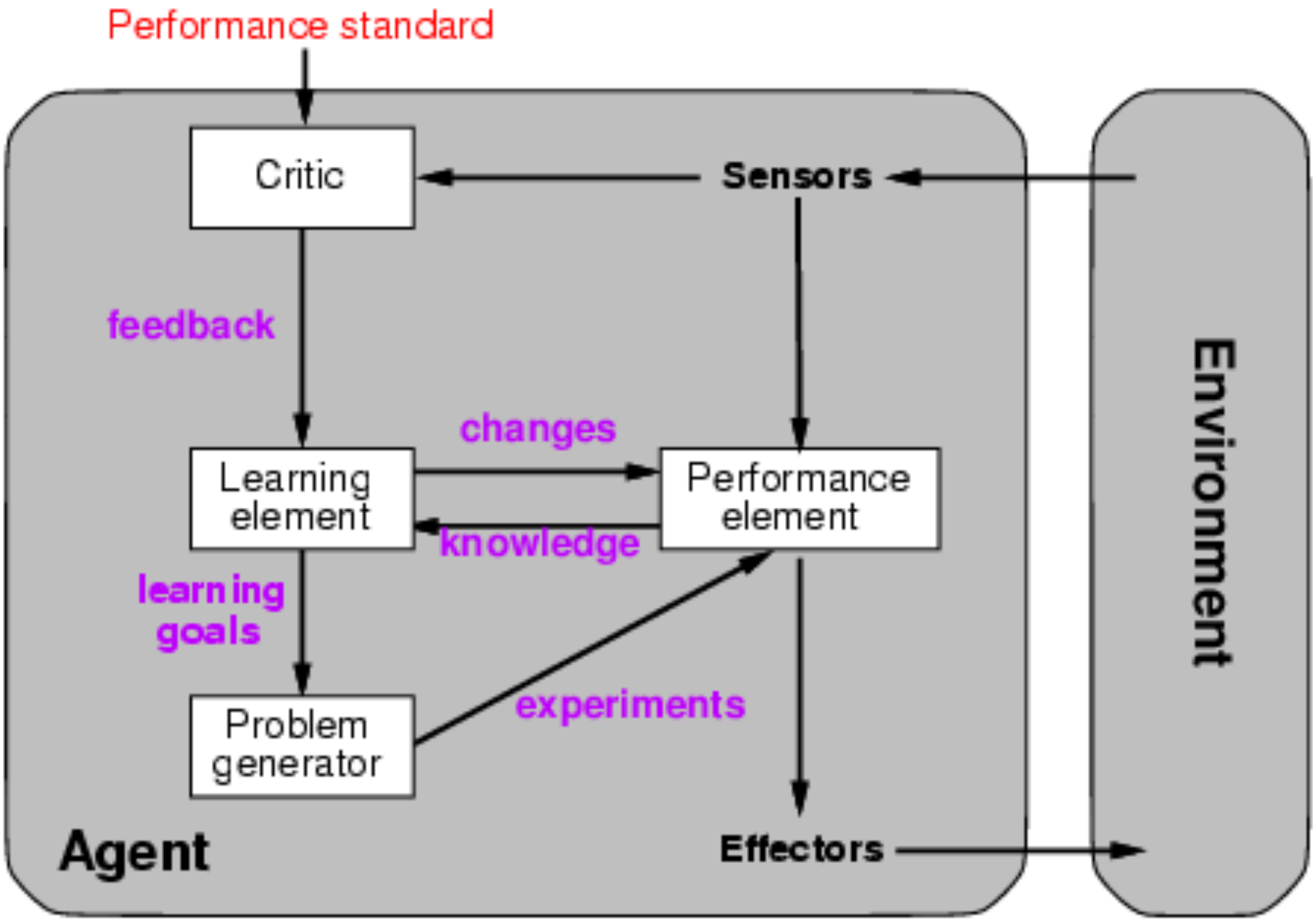
- Neural network learning was popular in the 60s
- In the 70s and 80s it was replaced with a paradigm based on manually encoding and using knowledge
- In the 90s, more data and the Web drove interest in new statistical machine learning (ML) techniques and new data mining applications
- Today, ML techniques and big data are behind almost all successful intelligent systems



Machine Leaning Successes

- Sentiment analysis
- Spam detection
- Machine translation
- Spoken language understanding
- Named entity detection
- Self driving cars
- Motion recognition (Microsoft X-Box)
- Identifying paces in digital images
- Recommender systems (Netflix, Amazon)
- Credit card fraud detection

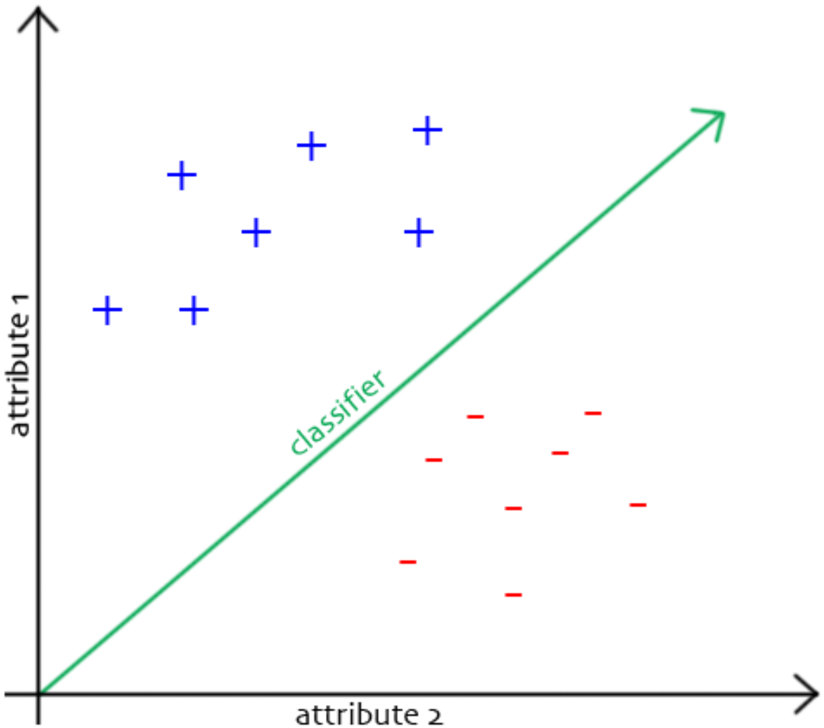
A general model of learning agents



Major paradigms of machine learning

- **Rote learning**: 1-1 mapping from inputs to stored representation, learning by memorization, association-based storage and retrieval
- **Induction:** Use specific examples to reach general conclusions
- **Clustering**: Unsupervised identification of natural groups in data
- **Analogy:** Determine correspondence between two different representations
- **Discovery**: Unsupervised, specific goal not given
- **Genetic algorithms:** *Evolutionary* search techniques, based on an analogy to *survival of the fittest*
- **Reinforcement** – Feedback (positive or negative reward) given at the end of a sequence of steps

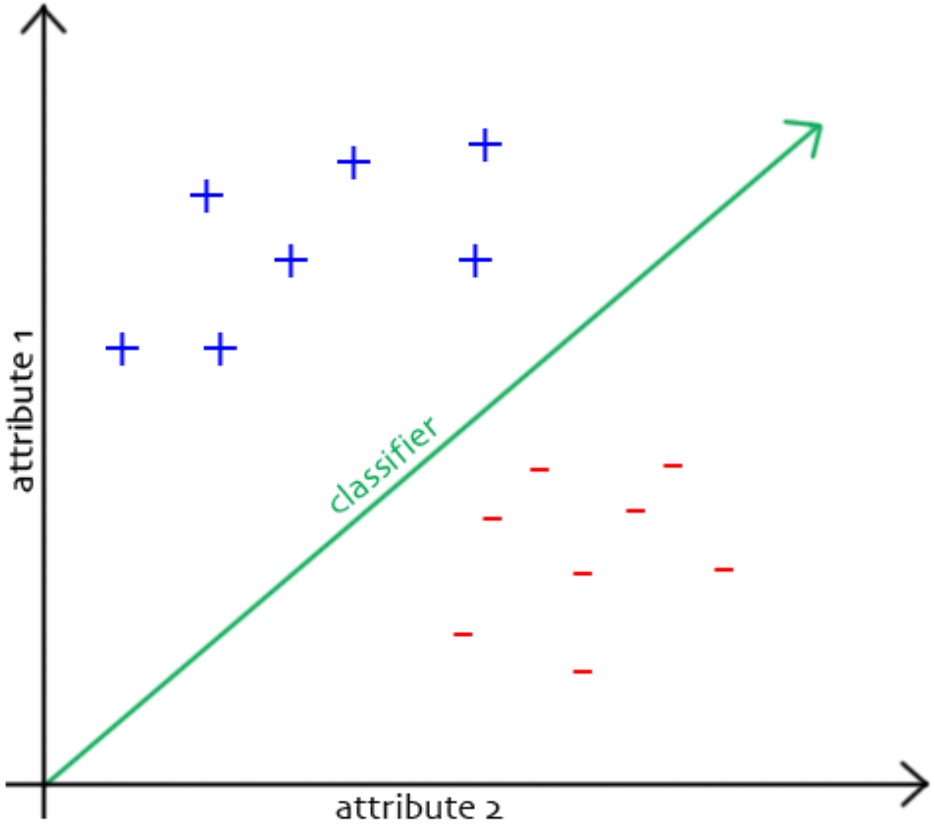
The Classification Problem



- Extrapolate from set of examples to make accurate predictions about future ones
- Supervised versus unsupervised learning
  - Learn unknown function f(X)=Y, where X is an input example and Y is desired output
  - **Supervised learning** implies we're given a **training set** of (X, Y) pairs by a "teacher"
  - **Unsupervised learning** means we are only given the Xs and some (ultimate) feedback function on our performance.

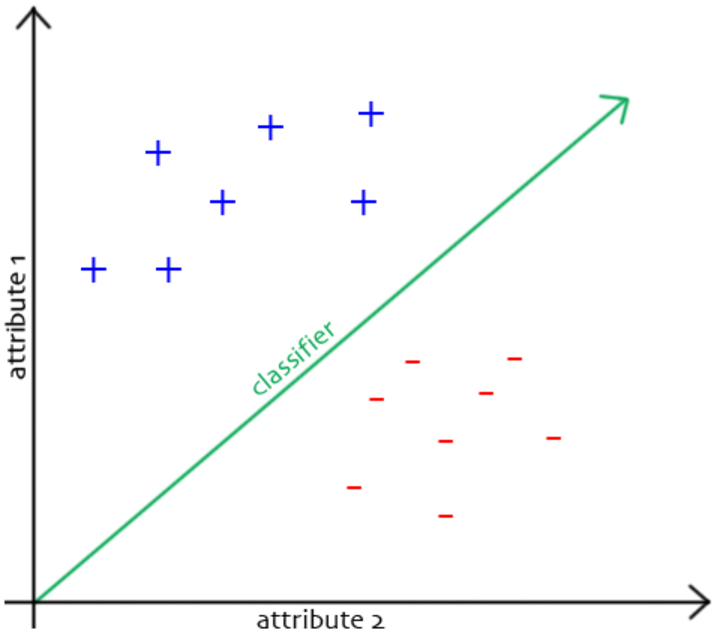
- Concept learning or classification (aka "induction")
  - Given a set of examples of some concept/class/category, determine if a given example is an instance of the concept or not
  - If it is an instance, we call it a positive example
  - If it is not, it is called a negative example
  - Or we can make a probabilistic prediction (e.g., using a Bayes net)

Supervised Concept Learning



- Given a training set of positive and negative examples of a concept
- Construct a description that will accurately classify whether future examples are positive or negative
- That is, learn some good estimate of function f given a training set $\{(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)\}$, where each y_i is either + (positive) or - (negative), or a probability distribution over +/-

Inductive Learning Framework



- Raw input data from sensors are typically preprocessed to obtain a **feature vector**, X, that adequately describes all of the relevant features for classifying examples
- Each x is a list of (attribute, value) pairs. For example,

  X = [Person:Sue, EyeColor:Brown, Age:Young, Sex:Female]
- The number of attributes (a.k.a. features) is fixed (positive, finite)
- Each attribute has a fixed, finite number of possible values (or could be continuous)

• Each example can be interpreted as a point in an n-dimensional **feature space**, where n is the number of attributes

Measuring Model Quality

- How good is a model?
  - Predictive accuracy
  - False positives / false negatives for a given cutoff threshold
    - Loss function (accounts for cost of different types of errors)
  - Area under the (ROC) curve
  - Minimizing loss can lead to problems with overfitting
- Training error
  - Train on all data; measure error on all data
  - Subject to overfitting (of course we'll make good predictions on the data on which we trained!)
- Regularization
  - Attempt to avoid overfitting
  - Explicitly minimize the complexity of the function while minimizing loss. Tradeoff is modeled with a *regularization parameter*

Cross-Validation

- Divide data into training set and test set
- Train on training set; measure error on test set
- Better than training error, since we are measuring *generalization to new data*
- To get a good estimate, we need a reasonably large test set
- But this gives less data to train on, reducing our model quality!

Cross-Validation, cont.

- k-fold cross-validation:
  - Divide data into *k* folds
  - Train on *k-1* folds, use *k*th fold to measure error
  - Repeat *k* times; use average error to measure generalization accuracy
  - Statistically valid; gives good accuracy estimates
- Leave-one-out cross-validation (LOOCV)
  - *k*-fold where k=N (test data = 1 instance!)
  - Accurate but expensive; requires building *N* models

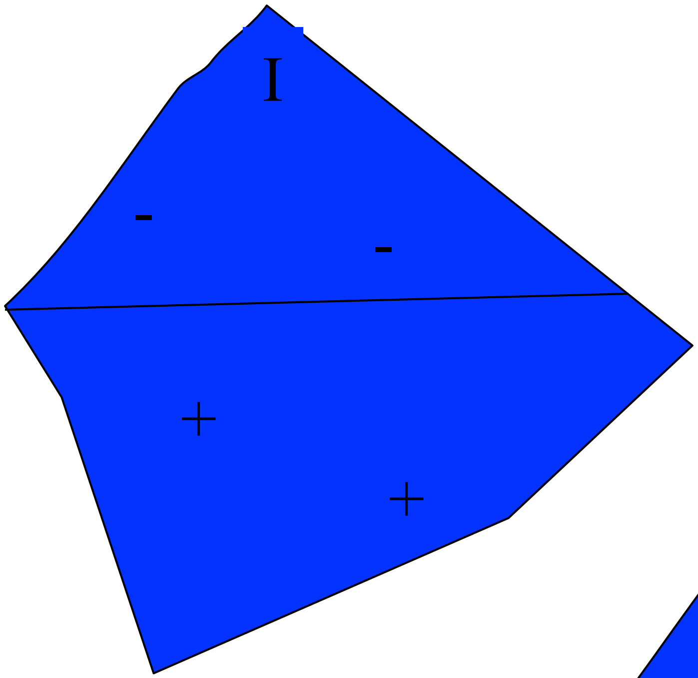
Inductive learning as search

- Instance space I defines the language for the training and test instances
  - Typically, but not always, each instance i∈I is a feature vector
  - Features are sometimes called attributes or variables
  - I: $V_1 \times V_2 \times \ldots \times V_k$, $i = (v_1, v_2, \ldots, v_k)$
- Class variable C gives an instance's class (to be predicted)
- Model space M defines the possible classifiers
  - M: $I \rightarrow C$, M = {m1, … mn} (possibly infinite)
  - Model space is sometimes, but not always, defined in terms of the same features as the instance space
- Training data can be used to direct the search for a good (consistent, complete, simple) hypothesis in the model space

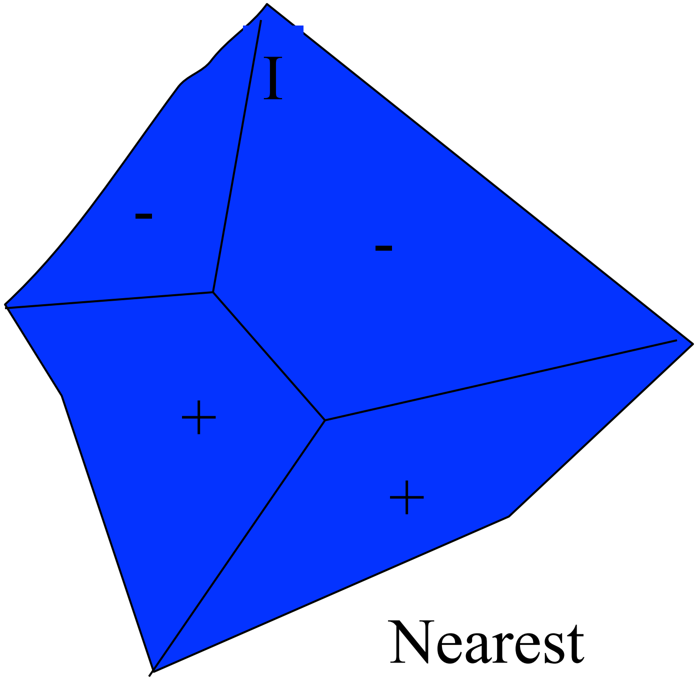
Model spaces

- **Decision trees**
  - Partition the instance space into axis-parallel regions, labeled with class value
- **Version spaces**
  - Search for necessary (lower-bound) and sufficient (upper-bound) partial instance descriptions for an instance to be in the class
- Nearest-neighbor classifiers
  - Partition the instance space into regions defined by the centroid instances (or cluster of k instances)
- Associative rules (feature values → class)
- First-order logical rules
- Bayesian networks (probabilistic dependencies of class on attributes)
- Neural networks

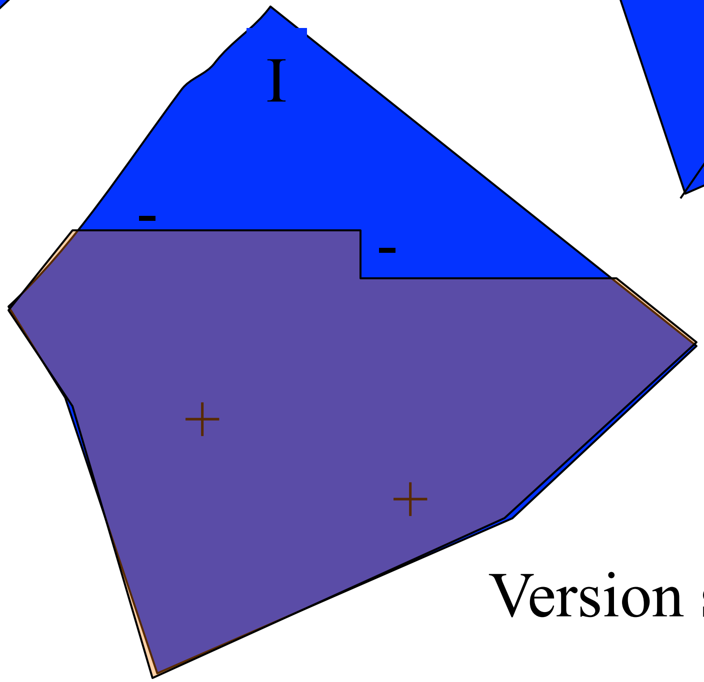
Model spaces



Decision tree



Nearest neighbor



Version space