Machine Learning Part 1
Introduction to Machine Learning

With some elements from: Tom Mitchel, Gheorghe Tecuci

Philippe Pasquier
Office 565 (floor 14)
pasquier@sfu.ca

SFU

Philippe Pasquier, January 2008

What is learning?

Learning is a very general term denoting the way in which agents:

- **Acquire and organize knowledge (by building, modifying and organizing internal representations of some external reality);**
- **Discover new knowledge and theories (by creating hypotheses that explain some data or phenomena);**
- **Acquire skills (by gradually improving their motor or cognitive skills through repeated practice, sometimes involving little or no conscious thought).**

⇨ **Learning results in changes in the agent that improve its competence and/or efficiency.**

What is learning?

- **Two complementary dimensions of learning:**
  - **Competence: A system is improving its competence if it learns to solve a broader class of problems, and to make fewer mistakes in problem solving.**
  - **Efficiency: A system is improving its efficiency, if it learns to solve the problems from its area of competence faster or by using fewer resources.**

⇨ **Can we do that for artificial agents?**

What is machine learning?

- **Artificial Intelligence is the Science and Engineering that is concerned with the theory and practice of developing systems that exhibit the characteristics we associate with intelligence in human/animal behavior**
- **Machine Learning is the domain of Artificial Intelligence which is concerned with building adaptive computer systems that are able to improve their competence and/or efficiency through learning from input data or from their own problem solving experience.**
- **Machine learning theories and techniques draw inspiration from many disciplines: Artificial Intelligence, Data Mining, Probability and Statistics, Information theory, Numerical optimization, Computational complexity theory, Control theory (adaptive), Psychology (developmental, cognitive), Neurobiology, Linguistics, Philosophy, ...**

Machine Learning

- **Definition: "Learning is any process by which a system improves performance from experience." (Herbert Simon)**
- **More formally: Improve on task, T, with respect to performance metric, P, based on experience, E**
- **What is the task?**
  - Classification: Assign object/event to one of a given finite set of categories.
  - Problem solving / planning / control: Performing actions in an environment in order to achieve a goal.

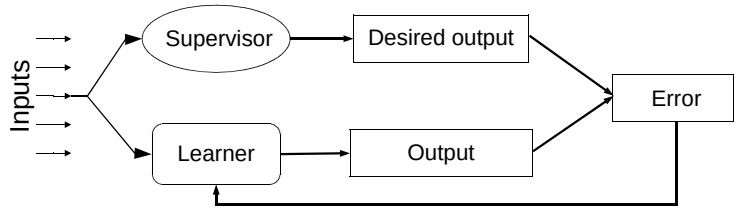
  } **Suppervised Learning**

  - Clustering: Partition unlabeled examples into disjoint subsets of *clusters*, such that:
    - Examples within a cluster are very similar
    - Examples in different clusters are very different

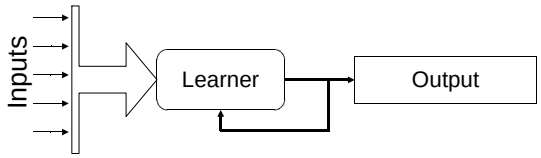
  } **Unsuppervised Learning**

Supervised vs. Unsupervised Learning

Supervised learning:



Unsupervised learning:



Examples

- **Classification:**
  - **The task: assign object/event to one of a given finite set of categories.**
  - **Examples: Medical diagnosis, Fraud detection in e-commerce, Worm detection in network packets, Spam filtering in email, Recommended articles in a newspaper, Recommended books, movies, music, or jokes, Financial investments, DNA sequences, Spoken words, Handwritten letters, Astronomical images, Rhythm selection, Human trained filters, ...**
- **Problem Solving / Planning / Control**
  - **The task: performing actions in an environment in order to achieve a goal.**
  - **Examples: Playing checkers, chess, backgammon, ..., Balancing a pole, controlling a mobile robot, ..., Driving a car or a jeep, ..., Flying a plane, helicopter, or rocket, ..., Controlling a character in a video game, Composing music in the style of Bach, ..., Generating rythmic patterns, ...**
- **Clustering:**
  - **The task: creating classes for data and classifying them (Reducing the number of dimensions)**
  - **Examples: Clustering news articles, Clustering sounds, Clustering images, ...**

Defining a Learning System

T: Playing checkers
P: Percentages 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

Designing a Learning System

- **The design process involves:**
  - **Choosing exactly what is too be learned, i.e. the *target function*: learning can be viewed as using direct or indirect experience to approximate a chosen target function.**
  - **Choosing how to represent the target function.**
  - **Choosing the training experience (in the supervised case) or the input data (in the unsupervised case)**
  - **Choosing a learning algorithm to infer the target function from the experience: 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.**

Various Function Representations

- **Numerical functions**
  - **Linear regression**
  - **Neural networks**
  - **Support vector machines**
- **Symbolic functions**
  - **Decision trees**
  - **Rules in propositional logic or 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**
- **Agent(s) as a function:**
  - **Reinforcement learning**
  - **Learning to communicate**
  - **Communicate to learn: distributed learning**

Various Search Algorithms

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

Evaluation of Learning Systems

- Assessing solutions' correctness or quality: classification accuracy, goal achievement, number and relevance of clusters, ...
- Experimental evaluation:
  - Conduct controlled cross-validation experiments to compare various methods on a variety of benchmark datasets.
  - Gather data on their performance, e.g. test accuracy, training-time, testing-time.
  - Analyze differences for statistical significance.
- Theoretical evaluation:
  - Analyze algorithms mathematically and prove theorems about their:
    - Computational complexity
    - Ability to fit training data
    - Sample complexity (number of training examples needed to learn an accurate function)

"Nothing that is worth knowing can be taught."

Oscar Wilde