CSE6339 3.0 Introduction to Computational Linguistics
Tuesdays, Thursdays 14:30-16:00 – South Ross 101
Fall Semester, 2011

Machine Learning Introduction

Machine learning is a scientific discipline that is concerned with the design and development of algorithms that allow computers to change behavior based on data, such as from sensor data or databases. A major focus of machine learning research is to automatically learn to recognize complex patterns and make intelligent decisions based on data. Hence, machine learning is closely related to fields such as statistics, probability theory, data mining, pattern recognition, artificial intelligence, adaptive control, and theoretical computer science.

Applications

Applications for machine learning include machine perception, computer vision, natural language processing, syntactic pattern recognition, search engines, medical diagnosis, bioinformatics, brain-machine interfaces and cheminformatics, detecting credit card fraud, stock market analysis, classifying DNA sequences, speech and handwriting recognition, object recognition in computer vision, game playing, software engineering, adaptive websites and robot locomotion.

Human interaction

Some machine learning systems attempt to eliminate the need for human intuition in data analysis, while others adopt a collaborative approach between human and machine. Human intuition cannot, however, be entirely eliminated, since the system's designer must specify how the data is to be represented and what mechanisms will be used to search for a characterization of the data. Machine learning can be viewed as an attempt to automate parts of the scientific method.

Some statistical machine learning researchers create methods within the framework of Bayesian statistics.

Algorithm types

Machine learning algorithms are organized into a taxonomy, based on the desired outcome of the algorithm. Common algorithm types include:

- Supervised learning - Generates a function that maps inputs to desired outputs. For example, in a classification problem, the learner approximates a function mapping a vector into classes by looking at input-output examples of the function.
- Unsupervised learning - Models a set of inputs: like clustering
- Semi-supervised learning - Combines both labeled and unlabeled examples to generate an appropriate function or classifier.
- Reinforcement learning - Learns how to act given an observation of the world. Every action has some impact in the environment, and the environment provides feedback in the form of rewards that guides the learning algorithm.
- Transduction - Tries to predict new outputs based on training inputs, training outputs, and test inputs.
- Learning to learn - Learns its own inductive bias based on previous experience.
- Pareto-based multi-objective learning - a Pareto-based approach to learning that results in a set of learning models, which typically trade off between performance and complexity. See Multi-Objective Machine Learning Website

The computational analysis of machine learning algorithms and their performance is a branch of theoretical computer science known as computational learning theory. Because training sets are finite and the future is uncertain, learning theory usually does not yield absolute guarantees of the performance of algorithms. Instead, probabilistic bounds on the performance are quite common.

In addition to performance bounds, computational learning theorists study the time complexity and feasibility of learning. In computational learning theory, a computation is considered feasible if it can be done in polynomial time. There are two kinds of time complexity results. Positive results show that a certain class of functions can be learned in polynomial time. Negative results show that certain classes cannot be learned in polynomial time.

There are many similarities between Machine Learning theory and Statistics, although they use different terms.

Supervised learning

Supervised learning is a machine learning technique for deducing a function from training data. The training data consist of pairs of input objects (typically vectors), and desired outputs. The output of the function can be a continuous value (called regression), or can predict a class label of the input object (called classification). The task of the supervised learner is to predict the value of the function for any valid input object after having seen a number of training examples (i.e. pairs of input and target output). To achieve this, the learner has to generalize from the presented data to unseen situations in a "reasonable" way (see inductive bias). (Compare with unsupervised learning.) The parallel task in human and animal psychology is often referred to as concept learning.

Supervised learning can generate models of two types. Most commonly, supervised learning generates a global model that maps input objects to desired outputs. In some cases, however, the map is implemented as a set of local models (such as in case-based reasoning or the nearest neighbor algorithm).

In order to solve a given problem of supervised learning (e.g. learning to recognize handwriting) one has to consider various steps:

1. Determine the type of training examples. Before doing anything else, the engineer should decide what kind of data is to be used as an example. For instance, this might be a single handwritten character, an entire handwritten word, or an entire line of handwriting.
2. Gathering a training set. The training set needs to be characteristic of the real-world use of the function. Thus, a set of input objects is gathered and corresponding outputs are also gathered, either from human experts or from measurements.
3. Determine the input feature representation of the learned function. The accuracy of the learned function depends strongly on how the input object is represented. Typically, the input object is transformed into a feature vector, which contains a number of features that are descriptive of the object. The number of features should not be too large, because of the curse of dimensionality; but should be large enough to accurately predict the output.
4. Determine the structure of the learned function and corresponding learning algorithm. For example, the engineer may choose to use artificial neural networks or decision trees.
5. Complete the design. The engineer then runs the learning algorithm on the gathered training set. Parameters of the learning algorithm may be adjusted by optimizing performance on a subset (called a *validation* set) of the training set, or via cross-validation. After parameter adjustment and learning, the performance of the algorithm may be measured on a test set that is separate from the training set.

Another term for supervised learning is classification. A wide range of classifiers are available, each with its strengths and weaknesses. Classifier performance depends greatly on the characteristics of the data to be classified. There is no single classifier that works best on all given problems; this is also referred to as the No free lunch theorem. Various empirical tests have been performed to compare classifier

performance and to find the characteristics of data that determine classifier performance. Determining a suitable classifier for a given problem is however still more an art than a science.

The most widely used classifiers are the Neural Network (Multilayer perceptron), Support Vector Machines, k-nearest neighbor algorithm, Gaussian mixture model, Gaussian, naive Bayes, decision tree and radial basis function classifiers.

Unsupervised learning

In machine learning, unsupervised learning is a class of problems in which one seeks to determine how the data are organized. It is distinguished from supervised learning (and reinforcement learning) in that the learner is given only unlabeled examples.

Unsupervised learning is closely related to the problem of density estimation in statistics. However unsupervised learning also encompasses many other techniques that seek to summarize and explain key features of the data.

One form of unsupervised learning is clustering. Another example is blind source separation based on Independent Component Analysis (ICA).

Among neural network models, the Self-organizing map (SOM) and Adaptive resonance theory (ART) are commonly used unsupervised learning algorithms. The SOM is a topographic organization in which nearby locations in the map represent inputs with similar properties. The ART model allows the number of clusters to vary with problem size and lets the user control the degree of similarity between members of the same clusters by means of a user-defined constant called the vigilance parameter. ART networks are also used for many pattern recognition tasks, such as automatic target recognition and seismic signal processing.

Semi-supervised learning

In computer science, semi-supervised learning is a class of machine learning techniques that make use of both labeled and unlabeled data for training - typically a small amount of labeled data with a large amount of unlabeled data. Semi-supervised learning falls between unsupervised learning (without any labeled training data) and supervised learning (with completely labeled training data). Many machine-learning researchers have found that unlabeled data, when used in conjunction with a small amount of labeled data, can produce considerable improvement in learning accuracy. The acquisition of labeled data for a learning problem often requires a skilled human agent to manually classify training examples. The cost associated with the labeling process thus may render a fully labeled training set infeasible, whereas acquisition of unlabeled data is relatively inexpensive. In such situations, semi-supervised learning can be of great practical value.

One example of a semi-supervised learning technique is co-training, in which two or possibly more learners are each trained on a set of examples, but with each learner using a different, and ideally independent, set of features for each example.

An alternative approach is to model the joint probability distribution of the features and the labels. For the unlabelled data the labels can then be treated as 'missing data'. It is common to use the EM algorithm to maximize the likelihood of the model

Reinforcement learning

Inspired by related psychological theory, in computer science, reinforcement learning is a sub-area of machine learning concerned with how an *agent* ought to take *actions* in an *environment* so as to maximize some notion of long-term *reward*. Reinforcement learning algorithms attempt to find a *policy* that maps *states* of the world to the actions the agent ought to take in those states. In economics and game theory, reinforcement learning is considered as a boundedly rational interpretation of how equilibrium may arise.

The environment is typically formulated as a finite-state Markov decision process (MDP), and reinforcement learning algorithms for this context are highly related to dynamic programming techniques. State transition probabilities and reward probabilities in the MDP are typically stochastic but stationary over the course of the problem.
Reinforcement learning differs from the supervised learning problem in that correct input/output pairs are never presented, nor sub-optimal actions explicitly corrected. Further, there is a focus on on-line performance, which involves finding a balance between exploration (of uncharted territory) and exploitation (of current knowledge).

Formally, the basic reinforcement learning model, as applied to MDPs, consists of:

1. a set of environment states S;
2. a set of actions A; and
3. a set of scalar "rewards" in .

At each time t, the agent perceives its state and the set of possible actions $A(s_t)$. It chooses an action and receives from the environment the new state s_{t+1} and a reward r_t. Based on these interactions, the reinforcement learning agent must develop a policy which maximizes the quantity for MDPs which have a terminal state, or the quantity

$$R = \sum_t \gamma^t r_t$$

for MDPs without terminal states (where is some "future reward" discounting factor).

Thus, reinforcement learning is particularly well suited to problems which include a long-term versus short-term reward trade-off. It has been applied successfully to various problems, including robot control, elevator scheduling, telecommunications, backgammon and chess.