Classification of Machine Learning Algorithms

Ariruna Dasgupta
Department of Computer Science
St. Xavier's College (Autonomous)
Kolkata, India

Asoke Nath
Department of Computer Science
St. Xavier's College (Autonomous)
Kolkata, India

Abstract— ***The goal of various machine learning algorithms is to device learning algorithms that learns automatically without any human intervention or assistance. The emphasis of machine learning is on automatic methods. Supervised Learning, unsupervised learning and reinforcement learning are discussed in this paper. Machine learning is the core area of Artificial Intelligence. Although a subarea of AI, machine learning also intersects broadly with other fields, especially statistics, but also mathematics, physics, theoretical computer science and more.***

Keywords—Classification;Clustering;Learning;ReinforcementLearning;SupervisedLearning;Unsupervised Learning

I. INTRODUCTION

It is very hard to write programs such as to compute the probability that a credit card transaction is fraudulent. There may not be any rules that are both easy to execute and reliable. We need to combine a very large number of weak rules to build an algorithm to work properly. The program also needs to keep changing since fraud is a moving target. In such cases machine learning algorithms finds a huge application. Machine learning studies how to automatically learn to make accurate predictions based on past observations [3]. It is how to train a machine to perform specific tasks. Some areas where machine learning finds applications are pattern recognition, recognizing anomalies, prediction, neural networks and many more.

Machine learning is categorized into three broad categories:

- Supervised learning: *In this type of learning, the machine is provided with a given set of inputs with their desired outputs. The machine needs to study those given sets of inputs and outputs and find a general function that maps inputs to desired outputs.*
- Unsupervised learning: *Here, the goal is to find a good internal representation of the input. Labeled examples are not available in unsupervised learning.*
- Reinforcement learning: *Here, the algorithm learns a policy of how to act given an observation of the world without knowing whether it has reached the goal or not.*

There are some other types of learning which uses the combinations of the above three algorithms. Between supervised and unsupervised learning is semi-supervised learning, where the machine is provided with a training set with some (often many) of the target outputs missing. Transduction is a special case of this principle where the entire set of problem instances is known at learning time, except that part of the targets is missing.

II. SUPERVISED LEARNING

Supervised learning is inferring a function from a given set of data (inputs with their respective outputs).The training data consists of a set of examples with which the computer is trained [6]. Each example, a pair consisting of an input object (typically a vector) and a desired output value (also called the supervisory signal). A supervised learning algorithm supervises the training data and produces a general rule (function), which can be used for mapping new inputs.

Supervised learning comes in two different flavors:

We consider the each training case consists of an input vector **x** and a target output **t.**

- Regression: *The target output is a real number or a whole vector of real numbers such a price of stock in 6 months' time or the temperature at noon tomorrow [2].*
- Classification: *The target output is a class label like in the simplest case choosing between positive and negative. We can also have multiple alternative levels [2].*

Steps required in order to solve supervised learning are:

- Determination of the type of training data. The user should decide what kind of data is to be used as a training set. In the case of sign language detection, for example, this might be a hand gesture for a single alphabet or hand gesture for an entire word.

- Gathering a training set. The training set must pertain to the real world so that a more generalized function can be made from these. Thus, a set of input objects is gathered and corresponding outputs are also gathered, from real world examples.
- Determination of 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 contain enough information to accurately predict the output.
- Determination of the various structures of the learned function and then devising the learning algorithm accordingly.
- Completion of the design. Execute the learning algorithm using the gathered training set. Some supervised learning algorithms require the user to determine certain control parameters. These parameters may be adjusted by optimizing performance on a subset (called a validation set) of the training set, or via cross-validation.
- Evaluation of the accuracy of the learned function. After parameter adjustment and learning, the performance of the resulting function should be measured on a test set that is separate from the training set.

There are four issues taken into account while dealing with supervised learning:

- Bias-Variance tradeoff: The training set may contain several different but equally good data sets. Now the learning algorithm is said to be biased for a particular input if, when trained on these data sets, it is systematically incorrect while predicting the correct output for that particular output. A learning algorithm has a high variance for a particular input when it provides different outputs when trained on different data sets. Thus, there is a tradeoff between bias and variance and supervised learning approach is able to adjust this tradeoff.

- Amount of training Data and Function Complexity: The amount of data required to provide during the training period depend on the complexity of the function required to map from the training data set. So, for a simple function with low complexity, the learning algorithm can learn from a small amount of data. Whereas, on the other hand, for high complexity functions, the learning algorithm needs large amount of data.

- Dimensionality of the input space: If the input feature vectors have high dimension then the learning algorithm can be difficult even it depends on a small number of features. This is because the many "extra" dimensions can confuse the learning algorithm and cause it to have high variance. Hence, high input dimensionality typically requires tuning the classifier to have low variance and high bias.

- Noise in the input values: A fourth issue is the degree of noise in the desired output values. If the desired output values are often incorrect (because of human error or sensor errors), then the learning algorithm should not attempt to find a function that exactly matches the training examples. Attempting to fit the data too carefully leads to over fitting. You can over fit even when there are no measurement errors (stochastic noise) if the function you are trying to learn are too complex for your learning model. In such a situation that part of the target function that cannot be modeled "corrupts" your training data - this phenomenon has been called deterministic noise. When either type of noise is present, it is better to go with a higher bias, lower variance estimator.

Some other factors:

- Data Integrity: Various kinds of inputs must be provided in the training set so as to consider all the possible cases. This will provide better learning. But this may not work in all the cases. For example, in case of neural nets, the inputs must be in the same range.
- Data Redundancy: The input vectors should not contain much redundant data because it may confuse the learning algorithm. Hence, the performance of the algorithm will degrade.
- Presence of interactions and non-linearities: If each feature contribute independently to the outputs the simple as well as complex algorithms work well.

We start by choosing a model class. A model class, ***f***, is a way of using some numerical parameters, ***w***, to map each input vector *x*, to the predicted output *y*.

$$y=f(x;W)$$

Learning usually means adjusting the parameter to reduce the discrepancy between the target output t, on the each training case and the actual output y produced by the model.

For regression, ½ $(y\text{-}t)^2$ is often a sensible measure of discrepancy. For classification there other measures more sensible which works better.

There are various methods used to implement Supervised Learning algorithms. One such technique is Naïve Bayes Classifier.

Naïve Bayes Classifier is based on Bayes rule of conditional probability. It makes use of all the attributes contained in the data and analyzes them individually independent of each other. In most of the cases data available is too little and instances with a particular combination of attributes may not be available. Hence it becomes difficult to predict the classification of new instance using Bayes rule. To overcome this issue Naïve Bayes classifier considers each attributes separately for creating new instances [4]. This technique is simple and works well in many classification problems. The reason behind this is may be when the data available is less; the assumption that the attributes are independent of each other tends to provide the classifier with more information that could have been obtained by taking all the attributes together.

III. UNSUPERVISED LEARNING

Unsupervised learning is much harder because here the computer have to learn to perform specified tasks without telling it how to perform. Thus it becomes difficult to say what the goal of this learning is [8]. There are two approaches to unsupervised learning.

- The first approach is to teach the agent not by giving explicit categorizations, but by giving some sort of reward system to indicate success. This type of learning fits the decision making problems because the goal is to make a decision and not to categorize the problem. This type of learning closely associates with a real world where agents are rewarded for performing certain tasks and punished for doing other. Learning about the actual ways of doing the works is unnecessary. The reward system can be learned from previous rewards and punishments [7]. This approach can be used where sufficient data are not available and where learning is time consuming.
- The second approach is known as clustering. Here there is no reward system instead it finds similarities in the training data. The assumption is often that the cluster will match reasonably well with an intuitive classification. For example, in online shopping the store itself recommends things based on the clustering algorithms.

Unsupervised learning algorithms are designed to extract structure from data samples. The quality of the structure is measured by a cost function which is usually minimized to infer optimal parameters characterizing the hidden structure in data. Reliable and robust inference requires a guarantee that extracted structures are difficult for data source that is similar structures have to be extracted from a second sample set of the second data source.

One technique used to implement unsupervised learning is Repeated Bisections Clustering Algorithm.

Clustering means dividing the available data instances into sub-groups based on the similarities between the instances in a certain group. These sub-groups are called clusters and hence the process is known as clustering.

This algorithm is one way of carrying out K-means clustering algorithm. K-means clustering involves clustering the data instances into **k** clusters, where **k** is a positive number. In this the number of clusters is given by the user. K-means clustering algorithm starts with placing **k** centroids far away from each other as possible within the available domain [10]. Each of the available instances is assigned a particular centroid based on metrics such as Euclidean distance, Manhattan distance, etc. The position of the centroid is calculated every time a new instance is added to the clusters. This process continues until there is no more change in the clusters. The initial position of the centroid is important since it affects the further clustering. Hence the centroids are kept as far away from each other as possible within the available space.

In repeated bisection method, a k-way partitioning via repeated bisections is obtained by recursively applying k-means clustering to compute two way clustering. Initially there are two clusters. One cluster is selected at a time and is bisected further and so on. This process continues **(k-1)** times leading to k clusters.

It has been observed that semi-supervised learning works as well as supervised learning. In some cases it outperforms supervised methods.

IV. REINFORCEMENT LEARNING

In Reinforcement learning (RL) the output is an action or a sequence of actions and the only supervisory signal is an occasional scalar reward. The basic reinforcement method consists of: a set of environment sets S; a set of actions A; rules of transitioning between states; rules that determine the scalar immediate reward of a transition; rules that describes what the agent observes [5].

The observation depends upon the scalar immediate reward associated with the last transition. In many works, the agent is also assumed to observe the current environmental state that is interaction with the real world. Sometimes restrictions are imposed on the actions of an agent.

A reinforcement learning agent interacts with its environment in discrete time steps. At each time t, the agent receives an observation o_t, which typically includes the reward r_t. It then chooses an action a_t from the set of actions available, which is subsequently sent to the environment. The environment moves to a new state s_{t+1} and the reward r_{t+1} associated with the transition (s_t, a_t, s_{t+1}) is determined. The goal of a reinforcement learning agent is to collect as much reward as possible. The agent can choose any action as a function of the history and it can even randomize its action selection.

When the agent's performance is compared to that of an agent which acts optimally from the beginning, the difference in performance gives rise to the notion of regret. Note that in order to act near optimally, the agent must reason about the long term consequences of its actions: In order to maximize my future income I had better go to school now, although the immediate monetary reward associated with this might be negative.

Two components make reinforcement learning powerful: The use of samples to optimize performance and the use of function approximation to deal with large environments. Thanks to these two key components, reinforcement learning can be used in large environments in any of the following situations:

A model of the environment is known, but an analytic solution is not available; only a simulation model of the environment is given (the subject of simulation-based optimization).

The only way to collect information about the environment is by interacting with it.

The first two of these problems could be considered planning problems (since some form of the model is available), while the last one could be considered as a genuine learning problem. However, under reinforcement learning methodology both planning problems would be converted to machine learning problems.

Reinforcement learning generally provides poor performance because the functions are selected randomly without estimating the possible outcomes.

The main approaches to learning in this context are as follows:

- Direct Learning: The optimal control policy is learned without first learning an explicit model.
- Indirect Learning: Estimate an explicit model of the environment and compute an optimal policy for the estimated model.

RL initially referred to the latter (value-based) methods, although today the name applies more broadly. Our focus will be on the following class of algorithms.

Policy-Iteration based schemes (“actor-critic” learning):

The “policy evaluation” block essentially computes the value function under the current policy (assuming a fixed, stationary policy).

The “actor” block performs some form of policy improvement, based on the policy iteration idea:

$\bar{\pi} \in \text{argmax}\{r + pV\}$. In addition, it is responsible for implementing some “exploration” process.

Value-Iteration based Schemes: These schemes are based on some on-line version of the value-iteration recursions:

$$V^*_{t+1} = \max_\pi[r^\pi + P^\pi V^*_t].$$

A technique to solve reinforcement learning is Deterministic Q-Learning.

To demonstrate some key ideas, we start with a simplified learning algorithm that is suitable for a deterministic model, namely:

$s_{t+1} = f(s_t, a_t)$
$r_t = r(s_t, a_t)$

We consider the discounted return criterion:

$V^{\pi}(s) = \sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)$, given $s_0 = s$, $a_t = \pi(s_t)$
$V*(s) = \max_{\pi} V^{\pi}(s)$

Recall our definition of the Q-function (or state-action value function), specialized to the present deterministic setting:

$Q(s, a) = r(s, a) + \gamma V*(f(s, a))$
The optimality equation is then
$V*(s) = \max_a Q(s, a)$
or, in terms of Q only:
$Q(s, a) = r(s, a) + \gamma \max_{a'} Q(f(s, a), a')$
Our learning algorithm runs as follows:
• Initialize: Set $\hat{Q}(s, a) = Q_0(s, a)$, for all s, a.
• At each stage n = 0, 1, . . . :
– Observe s_n, a_n, r_n, s_{n+1}.
– Update $\hat{Q}(s_n, a_n)$: $\hat{Q}(s_n, a_n) := r_n + \gamma \max_{a'} \hat{Q}(s_{n+1}, a')$ [1]

We note that this algorithm does not tell us how to choose the actions a_n.

V. COMPARATIVE STUDY

The similarities of the above algorithms are:

- A machine learning algorithm learns from past experiences and produces an output based on the experiences.
- The algorithms have strong relations to mathematical optimization.
- The algorithms are related to statistical computation.

The differences are shown in Table 1.

Table 1. Differences

Dissimilarities		
Supervised Learning	Unsupervised Learning	Reinforcement Learning
The output is based on the training data set. Classification is used here.	The output is based on the clustering of data.	The output is based on the agent's interaction with the environment. It used deterministic or non-deterministic way of learning.
Priori is necessary	Priori is not necessary.	Priori is required.
It will always produce same output for a specific input.	It will produce different outputs on each run for a specific input.	The output changes if the environment does not remain same for a specific input.

VI. CONCLUSION

Machine Learning Research spans almost four decades. Much of the research has been to define various types of learning, establish the relationships among them, and elaborate the algorithms that characterize them [9]. But, much less effort has been devoted to bring machine learning to bear on real world applications. But recently researchers have found broader applications of machine learning to real world problems. Some of these are:

- Adaptive websites
- Affective computing

- Bioinformatics
- Brain-machine interfaces
- Cheminformatics
- Classifying DNA sequences
- Computational finance
- Computer vision, including object recognition
- Detecting credit card fraud
- Game playing
- Natural language processing
- Data mining
- Speech and handwriting recognition
-

Apart from these there are many more fields where machine learning is used now-a-days.

REFERENCES

[1]. Nahum Shimkin, "Learning in Complex System", Lecture Notes, Spring 2011.
[2]. Thomas G. Dietterich, "Machine-Learning Research", AI Magazine Volume 18 Number 4 (1997).
[3]. Rob Schapire, "Machine Learning Algorithms for Classification", Princeton University.
[4]. S. B. Kotsiantis, "Supervised Machine Learning: A Review of Classification Techniques", Informatica 31 (2007) 249-268.
[5]. Leslie Pack Kaelbling, Michael L. Littman, Andrew W. Moore "Reinforcement Learning: A Survey", Journal of Artificial Intelligence, Research 4 (1996) 237-285, May 1996.
[6]. R. Sathya, Annamma Abraham, "Comparison of Supervised and Unsupervised Learning Algorithms for Pattern Classification", (IJARAI) International Journal of Advanced Research in Artificial Intelligence, Vol. 2, No. 2, 2013.
[7]. R. Sathya and A. Abraham, "Unsupervised Control Paradigm for Performance Evaluation", International Journal of Computer Application, Vol 44, No. 20, pp. 27-31, 2012.
[8]. Taiwo Oladipupo Ayodele, "Types of Machine Learning Algorithms", University of Portsmouth, United Kingdom.
[9]. Alberto Maria Segre, "Applications of Machine Learning", Cornell University.
[10]. J.A. Hartigan and M.A. Wong, "A K-Means Clustering Algorithm", Journal of the Royal Statistical Society. Series C (Applied Statistics), Vol. 28, No. 1 (1979), pp. 100-108.