International Journal of Science and Research (IJSR)
ISSN: 2319-7064
ResearchGate Impact Factor (2018): 0.28 | SJIF (2018): 7.426

Machine Learning Algorithms - A Review

Batta Mahesh

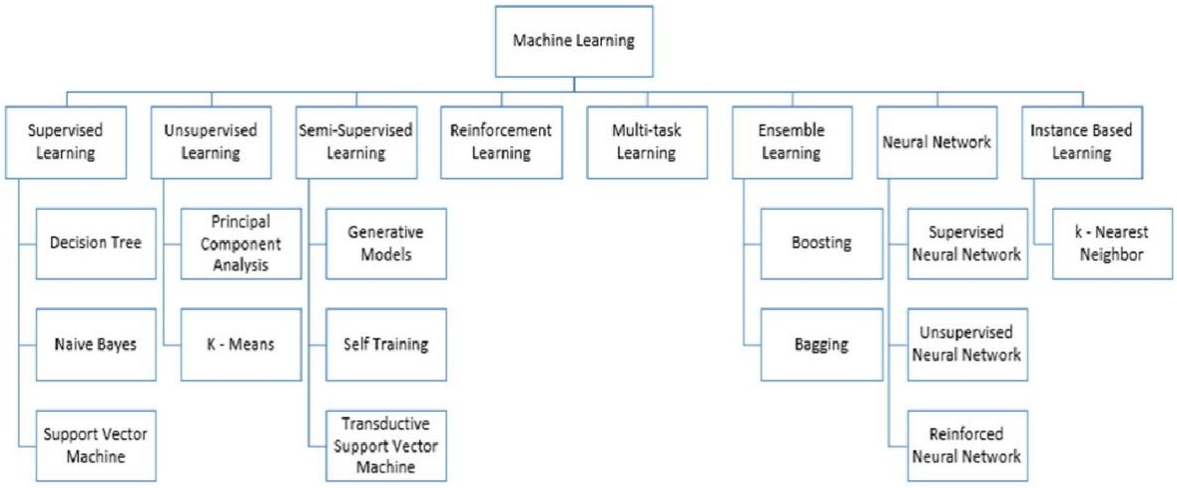
Abstract: ***Machine learning (ML) is the scientific study of algorithms and statistical models that computer systems use to perform a specific task without being explicitly programmed. Learning algorithms in many applications that's we make use of daily. Every time a web search engine like Google is used to search the internet, one of the reasons that work so well is because a learning algorithm that has learned how to rank web pages. These algorithms are used for various purposes like data mining, image processing, predictive analytics, etc. to name a few. The main advantage of using machine learning is that, once an algorithm learns what to do with data, it can do its work automatically. In this paper, a brief review and future prospect of the vast applications of machine learning algorithms has been made.***

Keywords: Algorithm, Machine Learning, Pseudo Code, Supervised learning, Unsupervised learning, Reinforcement learning

1. Introduction

Since their evolution, humans have been using many types of tools to accomplish various tasks in a simpler way. The creativity of the human brain led to the invention of different machines. These machines made the human life easy by enabling people to meet various life needs, including travelling, industries, and computing. And Machine learning is the one among them.

According to Arthur Samuel Machine learning is defined as the field of study that gives computers the ability to learn without being explicitly programmed. Arthur Samuel was famous for his checkers playing program. **Machine learning (ML)** is used to teach machines how to handle the data more efficiently. Sometimes after viewing the data, we cannot interpret the extract information from the data. In that case, we apply machine learning. With the abundance of datasets available, the demand for machine learning is in rise. Many industries apply machine learning to extract relevant data. The purpose of machine learning is to learn from the data. Many studies have been done on how to make machines learn by themselves without being explicitly programmed. Many mathematicians and programmers apply several approaches to find the solution of this problem which are having huge data sets.



Machine Learning relies on different algorithms to solve data problems. Data scientists like to point out that there's no single one-size-fits-all type of algorithm that is best to solve a problem. The kind of algorithm employed depends on the kind of problem you wish to solve, the number of variables, the kind of model that would suit it best and so on. Here's a quick look at some of the commonly used algorithms in machine learning (ML)

Supervised Learning

Supervised learning is the machine learning task of learning a function that maps an input to an output based on example input-output pairs. It infers a function from labelled training data consisting of a set of training examples. The supervised machine learning algorithms are those algorithms which needs external assistance. The input dataset is divided into train and test dataset. The train dataset has output variable which needs to be predicted or classified. All algorithms learn some kind of patterns from the training dataset and apply them to the test dataset for prediction or classification. The workflow of supervised machine learning algorithms is given in fig below. Most famous supervised machine learning algorithms have been discussed here

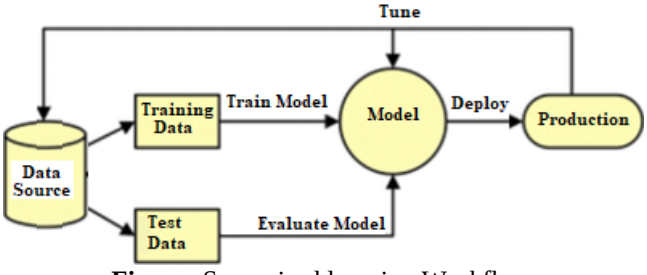


Figure: Supervised learning Workflow

Decision Tree

Decision tree is a graph to represent choices and their results in form of a tree. The nodes in the graph represent an event or choice and the edges of the graph represent the decision rules or conditions. Each tree consists of nodes and branches. Each node represents attributes in a group that is to be classified and each branch represents a value that the node can take**.**

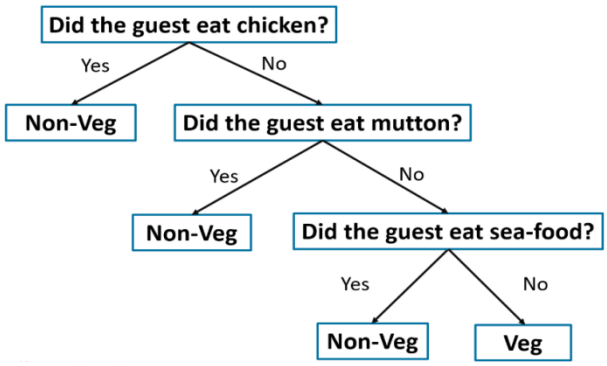


Figure: Decision Tree

Decision Tree Pseudo Code:

```
def
decisionTreeLearning(examples, attributes,
parent_examples):
if len(examples) == 0:
return pluralityValue(parent_examples)
# return most probable answer as there is no training data
left
elif len(attributes) == 0:
return pluralityValue(examples)
elif (all examples classify the same):
return their classification
A = max(attributes, key(a)=importance(a, examples)
# choose the most promissing attribute to condition on
tree = new Tree(root=A)
for value in A.values():
exs = examples[e.A == value]
subtree = decisionTreeLearning(exs, attributes.remove(A),
examples)
# note implementation should probably wrap the trivial case
returns into trees for consistency
tree.addSubtreeAsBranch(subtree, label=(A, value)
return tree
```

Navie Bayes

It is a classification technique based on Bayes Theorem with an assumption of independence among predictors. In simple terms, a Naive Bayes classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature. Naïve Bayes mainly targets the text classification industry. It is mainly used for clustering and classification purpose depends on the conditional probability of happening.

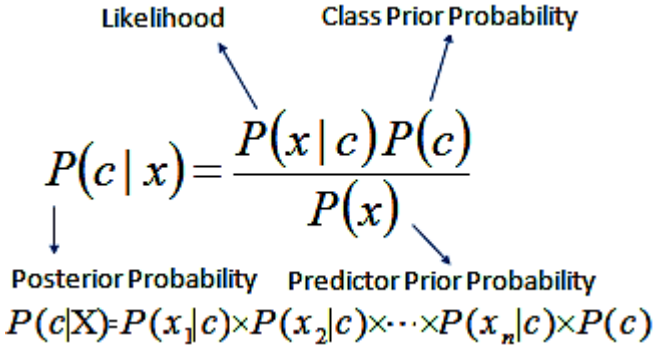


Figure: Navie Bayes

Pseudo Code of Navie Bayes

Input:

Training dataset T,

F= (f1, f2, f3,.., fn) // value of the predictor variable in testing dataset.

Output: A class of testing dataset.

Steps:

1) Read the training dataset T;
2) Calculate the mean and standard deviation of the predictor variables in each class;
3) Repeat Calculate the probability of fi using the gauss density equation in each class; Until the probability of all predictor variables (f1, f2, f3,.., fn) has been calculated.
4) Calculate the likelihood for each class;
5) Get the greatest likelihood

Support Vector Machine

Another most widely used state-of-the-art machine learning technique is Support Vector Machine (SVM). In machine learning, support-vector machines are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis. In addition to performing linear classification, SVMs can efficiently perform a non-linear classification using what is called the kernel trick, implicitly mapping their inputs into high-dimensional feature spaces. It basically, draw margins between the classes. The margins are drawn in such a fashion that the distance between the margin and the classes is maximum and hence, minimizing the classification error.

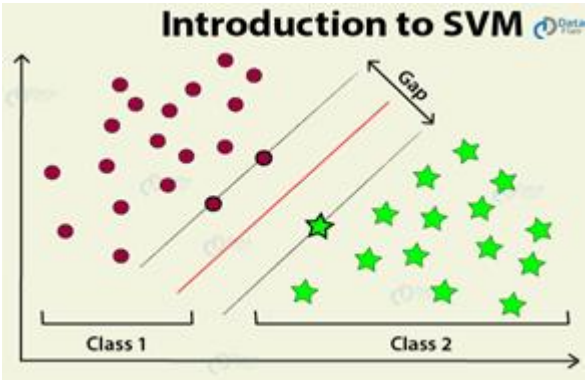


Figure: Support Vector Machine

Pseudo Code of Support Vector Machine

initialize Yi = YI for i ∈ I

repeat

compute svm solution vv , b for data set with imputed labels
compute outputs ii = (vv , xi) + b for all xi in positive bags
set yi = sgn(fi) for every i e i, yi = 1
for (every positive bag bi) end
if (liei(l + yi)/2 == 0)
compute i* = arg maxiei ii
set yi* = 1
end
while (imputed labels have changed)
output (vv, b)

Unsupervised Learning:

These are called unsupervised learning because unlike supervised learning above there is no correct answers and there is no teacher. Algorithms are left to their own devises to discover and present the interesting structure in the data. The unsupervised learning algorithms learn few features from the data. When new data is introduced, it uses the previously learned features to recognize the class of the data. It is mainly used for clustering and feature reduction.

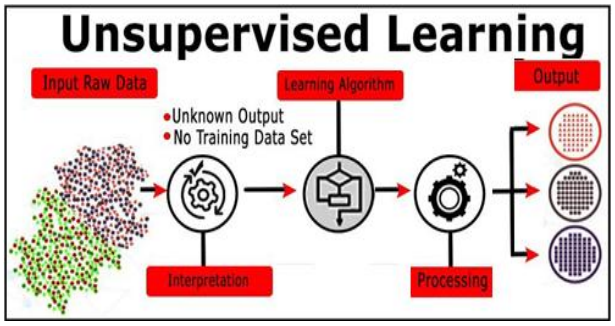


Figure: Unsupervised Learning

Principal Component Analysis

Principal component analysis is a statistical procedure that uses an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called principal components. In this the dimension of the data is reduced to make the computations faster and easier. It is used to explain the variance-covariance structure of a set of variables through linear combinations. It is often used as a dimensionality-reduction technique.

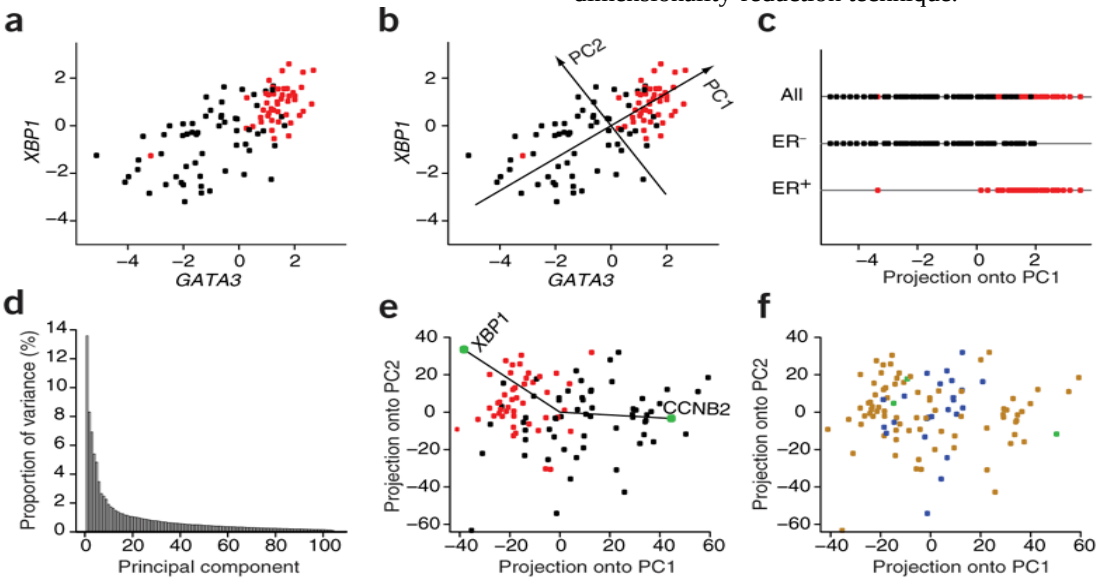


Figure: Principal Component Analysis

K-Means Clustering

K-means is one of the simplest unsupervised learning algorithms that solve the well known clustering problem. The procedure follows a simple and easy way to classify a given data set through a certain number of clusters. The main idea is to define k centers, one for each cluster. These centers should be placed in a cunning way because of different location causes different result. So, the better choice is to place them is much as possible far away from each other.

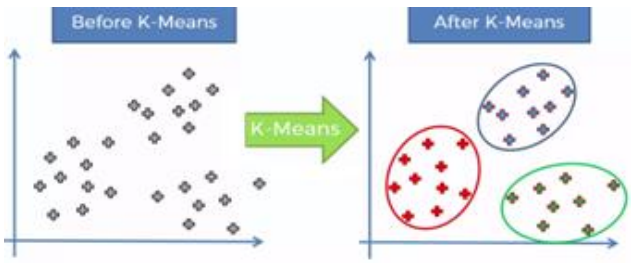


Figure: K-Means Clustering

The next step is to take each point belonging to a given data set and associate it to the nearest center. When no point is pending, the first step is completed and an early group age is done. At this point we need to re-calculate k new centroids as bary center of the clusters resulting from the previous step.

Input:
D= {t1, t2, …. Tn } // Set of elements
K // Number of desired clusters
Output:
K // Set of clusters
K-Means algorithm:
Assign initial values for *m1, m2,…. mk*
repeat
assign each item tj to the clusters which has the closest mean; calculate new mean for each cluster;
until convergence criteria is met;

Figure: Pseudo Code of K-Means Clustering

Semi Supervise Learning:
Semi-supervised machine learning is a combination of supervised and unsupervised machine learning methods. It can be fruit-full in those areas of machine learning and data mining where the unlabeled data is already present and getting the labeled data is a tedious process. With more common supervised machine learning methods, you train a machine learning algorithm on a "labeled" dataset in which each record includes the outcome information. The some of Semi Supervise learning algorithms are discussed below

Transductive SVM
Transductive support vector machines (TSVM) has been widely used as a means of treating partially labeled data in semisupervised learning. Around it, there has been mystery because of lack of understanding its foundation in generalization. It is used to label the unlabeled data in such a way that the margin is maximum between the labeled and unlabeled data. Finding an exact solution by TSVM is a NP-hard problem.

Generative Models
A Generative model is the one that can generate data. It models both the features and the class (i.e. the complete data). If we model P(x,y): I can use this probability distribution to generate data points - and hence all algorithms modeling P(x,y) are generative. One labeled example per component is enough to confirm the mixture distribution.

Self-Training
In self-training, a classifier is trained with a portion of labeled data. The classifier is then fed with unlabeled data. The unlabeled points and the predicted labels are added together in the training set. This procedure is then repeated further. Since the classifier is learning itself, hence the name self-training.

Reinforcement Learning
Reinforcement learning is an area of machine learning concerned with how software agents ought to take actions in an environment in order to maximize some notion of cumulative reward. Reinforcement learning is one of three basic machine learning paradigms, alongside supervised learning and unsupervised learning.

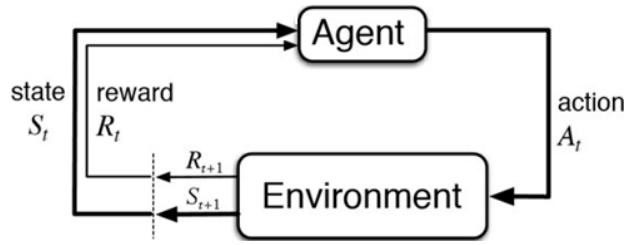


Figure: Reinforcement Learning

Multitask Learning
Multi-Task learning is a sub-field of Machine Learning that aims to solve multiple different tasks at the same time, by taking advantage of the similarities between different tasks. This can improve the learning efficiency and also act as a regularize. Formally, if there are **n** tasks (conventional deep learning approaches aim to solve just 1 task using 1 particular model), where these **n** tasks or a subset of them are related to each other but not exactly identical, Multi-Task Learning **(MTL)** will help in improving the learning of a particular model by using the knowledge contained in all the n tasks.

Ensemble Learning
Ensemble learning is the process by which multiple models, such as classifiers or experts, are strategically generated and combined to solve a particular computational intelligence problem. Ensemble learning is primarily used to improve the performance of a model, or reduce the likelihood of an unfortunate selection of a poor one. Other applications of ensemble learning include assigning a confidence to the decision made by the model, selecting optimal features, data fusion, incremental learning, non-stationary learning and error-correcting.

Boosting:
The term 'Boosting' refers to a family of algorithms which converts weak learner to strong learners. Boosting is a technique in ensemble learning which is used to decrease bias and variance. Boosting is based on the question posed by Kearns and Valiant **"Can a set of weak learners create a single strong learner?"** A weak learner is defined to be a classifier, a strong learner is a classifier that is arbitrarily well-correlated with the true classification.

Boosting Algorithm under Loss Function L
Inputs: Joint distribution Q on $\mathcal{Z} = \mathcal{X} \times \{1, -1\}$ and initial predictor $H^{(0)} \in \mathcal{S}[\mathcal{H}]$. In practice, Q is the empirical probability of training samples.
Loop For $m = 1, \ldots, M$
1. Find a hypothesis $h^{(m)} \in \mathcal{H}$ such that
$$h^{(m)} = \arg\min_{h \in \mathcal{H}} \int_{\mathcal{Z}} Q(dz) L'(-yH^{(m-1)}(x)) I(y \neq h(x))$$
The minimization does not need to be exact.
2. Find a coefficient $\alpha^{(m)} \in \mathbb{R}$ such that
$\alpha^{(m)} = \arg\min R_L(Q, H^{(m-1)} + \alpha h^{(m)})$.
Line search methods or Newton methods can be applied to solve the one-dimensional optimization problem.
3. Update the predictor: $H^{(m)} = H^{(m-1)} + \alpha^{(m)} h^{(m)}$.
Output: $H^{(M)}$ as an estimated predictor.
Figure: Boosting Pseudo code

Bagging
Bagging or bootstrap aggregating is applied where the accuracy and stability of a machine learning algorithm needs to be increased. It is applicable in classification and regression. Bagging also decreases variance and helps in handling overfitting.

```
1. Initialize feature vector bg feature= [0, 0, .....0]
2. for token in text.tokenize() do
3.       if    token in dict then
4.              token idx=getindex(dict, token)
5.             bg feature[token idx]++
6.       else
7.             continue
8.       end if
9.   end for
10. return bg feature
```

Figure: Pseudo code of Bagging

Neural Networks

A neural network is a series of algorithms that endeavors to recognize underlying relationships in a set of data through a process that mimics the way the human brain operates. In this sense, neural networks refer to systems of neurons, either organic or artificial in nature. Neural networks can adapt to changing input; so the network generates the best possible result without needing to redesign the output criteria. The concept of neural networks, which has its roots in artificial intelligence, is swiftly gaining popularity in the development of trading systems.

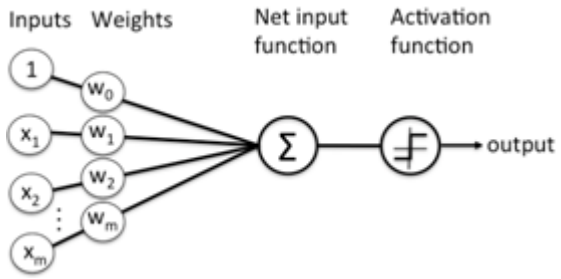


Figure: Neural Networks

An artificial neural network behaves the same way. It works on three layers. The input layer takes input. The hidden layer processes the input. Finally, the output layer sends the calculated output.

Supervised Neural Network

In the supervised neural network, the output of the input is already known. The predicted output of the neural network is compared with the actual output. Based on the error, the parameters are changed, and then fed into the neural network again. Supervised neural network is used in feed forward neural network.

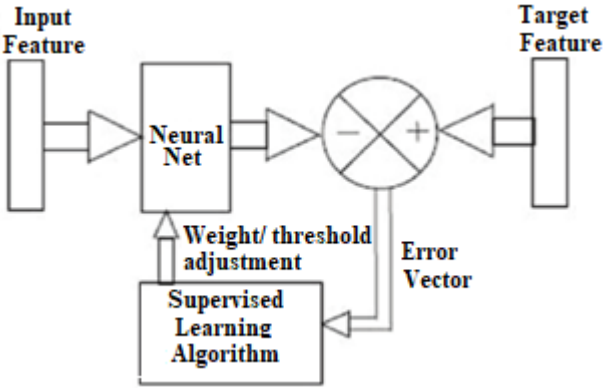


Figure: Supervised Neural Network

Unsupervised Neural Network

The neural network has no prior clue about the output the input. The main job of the network is to categorize the data according to some similarities. The neural network checks the correlation between various inputs and groups them.

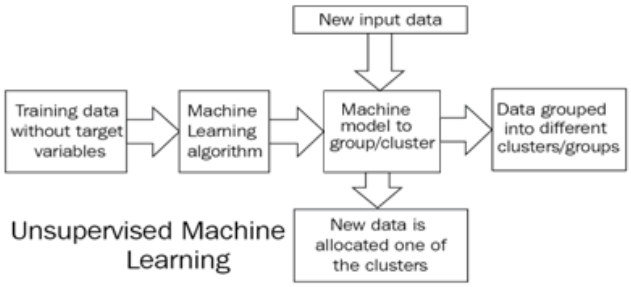


Figure: Unsupervised Neural Network

Reinforced Neural Network

Reinforcement learning refers to goal-oriented algorithms, which learn how to attain a complex objective (goal) or maximize along a particular dimension over many steps; for example, maximize the points won in a game over many moves. They can start from a blank slate, and under the right conditions they achieve superhuman performance. Like a child incentivized by spankings and candy, these algorithms are penalized when they make the wrong decisions and rewarded when they make the right ones – this is reinforcement.

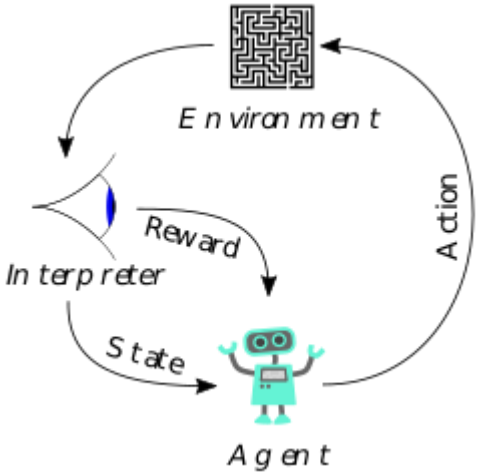


Figure: Reinforced Neural Network

Instance-Based Learning

Instance-based learning refers to a family of techniques for classification and regression, which produce a class label/predication based on the similarity of the query to its nearest neighbor(s) in the training set. In explicit contrast to other methods such as decision trees and neural networks, instance-based learning algorithms do not create an abstraction from specific instances. Rather, they simply store all the data, and at query time derive an answer from an examination of the queries nearest neighbour (s).

K-Nearest Neighbor

The k-nearest neighbors (KNN) algorithm is a simple, supervised machine learning algorithm that can be used to solve both classification and regression problems. It's easy to implement and understand, but has a major drawback of

becoming significantly slows as the size of that data in use grows.

```
k-Nearest Neighbor
Classify (X, Y, x) // X: training data, Y: class labels of X, x: unknown sample
for i = 1 to m do
  Compute distance d(X_i, x)
end for
Compute set I containing indices for the k smallest distances d(X_i, x).
return majority label for {Y_i where i ∈ I}
```

Figure: Pseudo code of KNN

2. Conclusion

Machine Learning can be a Supervised or Unsupervised. If you have lesser amount of data and clearly labelled data for training, opt for Supervised Learning. Unsupervised Learning would generally give better performance and results for large data sets. If you have a huge data set easily available, go for deep learning techniques. You also have learned Reinforcement Learning and Deep Reinforcement Learning. You now know what Neural Networks are, their applications and limitations. This paper surveys various machine learning algorithms. Today each and every person is using machine learning knowingly or unknowingly. From getting a recommended product in online shopping to updating photos in social networking sites. This paper gives an introduction to most of the popular machine learning algorithms.

References

[1] W. Richert, L. P. Coelho, “Building Machine Learning Systems with Python”, Packt Publishing Ltd., ISBN 978-1-78216-140-0
[2] J. M. Keller, M. R. Gray, J. A. Givens Jr., “A Fuzzy K-Nearest Neighbor Algorithm”, IEEE Transactions on Systems, Man and Cybernetics, Vol. SMC-15, No. 4, August 1985
[3] https://www.geeksforgeeks.org/machine-learning/
[4]] S. Marsland, Machine learning: an algorithmic perspective. CRC press, 2015.
[5] M. Bkassiny, Y. Li, and S. K. Jayaweera, “A survey on machine learning techniques in cognitive radios,” IEEE Communications Surveys & Tutorials, vol. 15, no. 3, pp. 1136–1159, Oct. 2012.
[6] https://en.wikipedia.org/wiki/Instance-based_learning
[7] R. S. Sutton, “Introduction: The Challenge of Reinforcement Learning”, Machine Learning, 8, Page 225-227, Kluwer Academic Publishers, Boston, 1992
[8] P. Harrington, “Machine Learning in action”, Manning Publications Co., Shelter Island, New York, 2012