MACHINE LEARNING

BEGINNERS GUIDE ALGORITHMS:

Supervised & Unsupervised Learning, Decision Tree & Random Forest Introduction

WILLIAM SULLIVAN

Machine Learning For Beginners Guide Algorithms:

Decision Tree & Random Forest Introduction

Under no circumstances will any legal responsibility or blame be held against the publisher for any reparation, damages, or monetary loss due to the information herein, either directly or indirectly.

Legal Notice:

You cannot amend, distribute, sell, use, quote or paraphrase any part or the content within this book without the consent of the author.

Disclaimer Notice:

Please note the information contained within this document is for educational purposes only. No warranties of any kind are expressed or implied. Readers acknowledge that the author is not engaging in the rendering of legal, financial, medical or professional advice. Please consult a licensed professional before attempting any techniques outlined in this book.

By reading this document, the reader agrees that under no circumstances are is the author responsible for any losses, direct or indirect, which are incurred as a result of the use of information contained within this document, including, but not limited to, —errors, omissions, or inaccuracies

Introduction

I want to thank you for choosing this book, ‘*Machine learning for beginners - Algorithms, Decision Tree & Random Forest Introduction.’*

By choosing this book, you have made the right decision, as you will learn many new, innovative and exciting things about the world of technology and computers. It will help you learn the basics of AI and machine learning in a simple, entertaining and informative way.

Currently one of the most talked about topics in the world of technology, machine learning is a promising concept. But along with the promises and benefits, it is also often associated with controversies and debates. People who are not aware of the nature and advantages of machine learning or have received their information from untrustworthy sources often look down on machine learning and are scared of it as well. However, all the strange and bizarre things that you have heard about machine learning are probably just myths and false apprehensions.

This book will try to do away with such apprehensions by showing how machine learning is perhaps the best thing that could happen to the world of technology right now. You will get answers to all your questions about machine learning and more. So, rather than making assumptions, you will learn and understand what machine learning is all about and make your own decisions.

So let’s read on.

Chapter 1: About Machine Learning

One of the best features of today’s era of technology is its flexibility and adaptableness. A new scientific innovation comes out almost every day. This ever-changing nature of scientific and technological world changes the trajectory of the world every day. Things that were considered dreams and fiction once are now rapidly turning into reality. Human beings are slowly but steadily trying to defeat nature at its own game. However, one field remains to be conquered. We still have not managed to conquer the world of machine learning or AI. However, it has become a buzzword now, and the whole of the world is talking about it. Not everyone is excited about it though. Most people are worried or scared of it. However, there is no need to be afraid of machine learning or AI, as it will help humanity to achieve things that we cannot even currently imagine.

What is Machine Learning?

If you check the search results for the most popular keywords of 2016, you will find that machine learning and AI are leading the figures by a large margin. This steady rise in the fame of machine learning is because of its rising use in our daily lives. It is nowadays being used in various devices and machines as well as gadgets. However, the general population is still are wary of it. So, to do away with such myths, let us have a look at the brief history of machine learning.

As per the 1959 definition of Arthur Samuel, machine learning can be defined as a process of inputting data to the computer systems in a way that the computer will learn the ability to process and perform the activity in the future without being explicitly programmed or being fed with similar or extra data. What this means in simple words is that it will allow computers to develop a ‘mind’ of their own and allow them to “think.” Sounds scary but it isn’t.

If computers are provided with the ability to think, they become smarter and thus easier to use. Their functionality will increase by a large margin, and they become an integral asset for humanity. Machine learning can be used in almost all the fields of epistemology. Right now, it is being used in areas such as cheminformatics, computational anatomy, gaming, adaptive websites, natural language processing, robot movement and locomotion, medical diagnosis, sequence mining, behavior analysis, linguistics, translation, fraud detection, *etc*. The list goes on.

History:

The history of machine learning can be traced to the birth of another related field-AI or artificial intelligence. It is safe to say that both of these fields were born at the same time and then got separated over time. Many scientists studying AI in the beginning slowly shifted towards studying machine learning academically. They started using probabilistic reasoning *etc.* Around the ‘90s the two fields, AI and machine learning, were officially separated, and now both of them are studied individually. In the following chapters, you will learn the basics of machine learning and how it can be used in day-to-day life. You will also learn about the careers that are available in this field as well as certain advanced topics for the experts.

Chapter 2: Machine Learning Basics

What is it that has made machine learning a buzzword in today's era? The simplest answer to the above question would be its unique, feature-rich nature that can change the future of humanity forever. In the words of Bill Gates:

"*A breakthrough in machine learning would be worth ten Microsofts*."

What the above statement roughly means is that scientists and computer experts all over are desperately looking for a breakthrough in machine learning and are looking for a way to make it more accessible, useful and trustworthy. However, such programs are still going, and we still haven't found a way to devise a machine that could think.

In machine learning, computers learn to program themselves. If programming is considered to be automation and an automatic process, then machine learning is the automation of this automatic process, thus making a double automatic process.

Machine learning can make programming more scalable and can help us to produce better results in shorter durations. To prove this, let us see the following comparison:

Differences between Traditional Programming and Machine Learning

Traditional Programming:

The data is fed to the computer, and a program is run. This program then, using the supplied data, presents output.

Machine Learning:

Pre-solved data and the resulting output are fed to the computer. These two inputs are used to create a program. This program then can do the job of traditional programming.

Thus, machine learning can be explained by using the metaphors of agriculture. Algorithms are, in a way seeds while data is nutrients. You are the farmer while the program that grows out of the data is your crop.

Elements of Machine Learning

As machine learning is a complicated and convoluted field, it's hard to understand its basics. It is also an ever-growing field. Hence it is possible to see new development in the area almost every day. For instance, it is believed that every year more than a few hundred, new algorithms are developed all over the world. This brings the number of overall machine learning algorithms to a sum that is larger than ten thousand. Even though a lot of variety is seen in the algorithms of machine learning, all of them are based on three basic concepts that are as follows.

Representation:

This concept deals with the representation of knowledge. It deals with how the knowledge can be represented, what is necessary to represent the knowledge etc. Some examples of representation include sets of rules, including decision trees, support vector machines, instances, neural networks, graphical models, model ensembles, *etc.* Some of these will be discussed in the book later.

Evaluation:

This is the second most important concept of algorithms. It is the way used to evaluate the hypotheses, also known as the candidate program. Some examples are accuracy, prediction and recall, squared error, likelihood, posterior probability, cost, margin, entropy k-L divergence and others.

Optimization:

This is the third and last concept of algorithms. It is the method in which the hypothesis or the candidate program is created. It is also known as the search process. Examples include combinatorial optimization, convex optimization and constrained optimization.

Making various combinations of the above components creates all machine-learning algorithms, and thus they are the basis of machine learning.

Types and Kinds of Machine Learning

As said earlier, machine learning is complex and vast field hence it can be divided into many sections and classes. However, on a superficial level, it can be split into four parts, and they are as follows:

1. Supervised Learning
2. Unsupervised Learning
3. Semi-supervised Learning
4. Reinforcement Learning

Supervised learning:

Supervised Learning is also known as inductive learning in the technological circles. It is considered to be the most advanced and mature of all the forms of learning. This is why it is the most studied as well as most used learning as well. It is easy to used Learning type as it is much easier to learn under supervision than without supervision. In Inductive Learning, we are presented with an example of a function in the form of data (x), and the output of the function is (f(x)). The mission of inductive learning here is to understand and learn the function for the new data (x).

In this learning, the program is 'trained' with the help of some already defined set of 'examples.' This training helps the program to learn the ability to formulate a new and accurate result using the newly fed data with ease and without any interference.

Supervised learning is the most used and most favorite of all forms of learning, this chapter as well as this book will try to focus on it. Other types will be discussed briefly.

In most of the supervised learning applications, the final mission is to create a proper and well-set predictor function h(x). It is also known as the hypothesis. The 'learning' contains many mathematical algorithms that are necessary to optimize the function. When it is optimized, it can correctly predict the value of h(x) if data X is fed to the computer related to a particular domain. For instance, if the data being fed is the square footage of agricultural land, the program should be able to return the estimated price for the piece of land.

However, it is seen that x always represents more than one data point. For instance, if we are to continue the above example, then the program may take the number of wells (x2), number of trees (x3), number of greenhouses (x4), number of electric poles (x5), number compost holes (x6) and many other variables along with the first one that is the square footage of the land (x1). Then the determination of the correct input is the input that will come out with the correct result. This is one of the major parts of machine learning design. However, as the topic might get too complex and complicated, this example will only assume a single input value.

Example Let us assume that the program or predictor is using this form: $H (x) = \theta 0 + \theta 1x$ Here $\theta 0$ and $\theta 1$ are constants. The mission here is to find the perfect values for the above two constants to create and make our predictor work properly.

To optimize the predictor h (x), training examples are used. In each of these examples a value of x train is added and corresponding to this value an output value-y is already known. For instance, the difference between the known *i.e.* correct value y, and the predicted value h (x train) is found. When enough training examples are fed, the differences can be studied and checked to determine and measure our faults of h (x). Using our findings, we can change and manipulate the h (x) by manipulating the values of $\theta 0$ and $\theta 1$ to make it more accurate. This process then is repeated until the best values of $\theta 0$ and $\theta 1$ are found. This is how the predictor is trained. This trained predictor can now read real life data and predict perfectly to almost perfect results.

Unsupervised learning:

The data that is fed to the system *i.e.* the training data does not include any desired output. Thus, the data that is to be fed is without output. It is difficult to understand and proclaim that whether this is good and recommended method of learning or not. Examples include clustering *etc.*

Semi-supervised learning:

This is a mixture of both the above kinds of learning where training data contains some but not all desired outputs.

Reinforcement learning:

Considered to be the most ambitious of all the types of learning. Here rewards are given from a sequence of actions. AI often prefers this.

Machine Learning in Practice

Although an important part of the overall machine learning process, machine algorithms are a tiny part of the complete process. In reality, the process is much more complicated. An example is as follows:

Start Loop

Here the domain is to be understood. The current data on hand, available knowledge and goals are analyzed. This often includes communicating with the domain experts. The goals are often unclear, and you have to attempt and try many things before implementing anything.

Data integration, selection, cleaning and pre-processing

This is the most time-consuming part in the overall process. It takes almost half or more than half of the overall required processing time. Procuring high-quality data is critical. However, the quality and quantity of data are often reversely proportional as the more the data, the more it will be dirty. Sorting out good and usable data from the rest is why this process takes a very long time.

Learning models

This part, though being the most mature part of the process, is also the most fun part of it. General tools are used for this.

Interpreting results

In many cases, it is not necessary to understand how the model works, the only focus being the results. Often human experts can challenge you on this.

Consolidating and deploying discovered knowledge.

Though many projects succeed in the lab and are remarkable, yet the chances of them being used in real life are quite rare. Most of the projects are discarded and not used in real life at all.

End Loop

There is an end cycle at the end, and it is not a one-shot process. It is necessary to run the loop until a desired and usable result is achieved. The data can also change midway, affecting the overall process, as you need a new loop to replace the earlier one.

Sample applications of machine learning

This is a small list of fields in which machine learning can be applied to achieve great results:

- Web search: It can rank web pages and search entries according to your previous clicks and likes and will prominently show new, similar results.
- Computational biology: Can rationally create drugs in the system or computer using old and past experiments.
- Finance: Can evaluate how much risk is present on a credit card. Can also be used to send tailored offers according to the likings of a customer. Can also help you in choosing where to invest money.
- E-commerce: Identifying a transaction's nature, whether it is a true and correct transaction or whether it is a fraudulent one. It can also help you in predicting the number of customers.
- Space exploration: Can help with radio astronomy and space probes.
- Robotics: Can help to create self-driving cars and autonomous robots. Can contribute to building robots that can handle uncertain situations and new environments.
- Information extraction: Can extract information from databases on the web.
- Social networks: Can help you get data on preferences and relationships.
- Debugging: Can be used for debugging.

Chapter 3: Machine Learning: Algorithms

While discussing the basics of machine learning in the last chapter, we talked briefly about machine learning in algorithms. Let us take a deeper look at the algorithms, their types, the most popular ones and everything else about them in this chapter.

As said in the last chapter, the sheer number of algorithms that are available can overwhelm any beginner to machine learning. Therefore, it is necessary to categorize them in two large sections for our convenience.

There are two ways of classifying algorithms; the first one is categorizing them by learning style while the second is the grouping of algorithms by the similarity of function or form. Both of these techniques or methods of categorizing are useful, however, let us algorithms by similarity category and check out its nuances.

Ensemble Learning Method

There are a variety of ways in which an algorithm can base a problem that relies on the interactions that take place with the experience or environment or any other form of input data.

It is highly popular in artificial intelligence as well as machine learning textbooks to consider the learning styles of an algorithm first followed by everything else later.

An algorithm can have only a few relevant learning models or learning styles. These will be explained in the sections below.

The categorization, or to use the scientific term, taxonomy/organization of machine learning algorithms is highly desirable as it makes you think of the roles of input data as well as the model preparation process. You can then choose one that is the best for your problem to get the best result.

Although covered briefly in the first chapter, these are the three different but basic learning styles:

Supervised Learning

In this, the input data is known as training data. It features a known result or a label, for instance, spam/not-spam or stock price, *etc.* A proper prediction model is constructed using the training process. It is needed to make predictions, and these predictions get corrected if they are wrong. The training process continues to repeat itself until perfection, or the desired level of accuracy is achieved.

Examples: Logistic Regression and the Back Propagation Neural Network.

Unsupervised Learning

The data under this category does not have a known output or result neither does it have a label.

The prediction model is constructed by guessing the number of structures present in the input data, often to take out general rules.

Example problems include dimensionality reduction, clustering, and association rule learning.

Example algorithms include various algorithms such as the Apriori algorithm and k-Means.

Semi-Supervised Learning

In this, the input data has no solid form and is a jumbled a mixture of labeled as well as unlabeled examples.

The model needs to learn the structures that are necessary to not only organize the data but make predictions as well.

Example problems include classification and regression.

Example algorithms are extensions.

Algorithms Grouped By Similarity

As said earlier, algorithms can be classified on various bases. They are often classified by the similarities that are seen in their functioning. For instance, neural network inspired methods and tree-based methods.

It is considered to be one of the best ways of grouping algorithms and is one of the most used methods. However, it is not perfect as many algorithms exist that cannot be classified in watertight compartments. For instance, the Learning Vector Quantization is a method that is an instance-based method and a neural network inspired method. Thus, it is not possible to classify the algorithms on a deeper level with just two criteria. Hence, people often use a nested approach while classifying machine learning algorithms.

There are many algorithms and groups of algorithms. Some of the major and frequently used are listed below.

Regression Algorithms

These deal with the modeling of the relationship among the variables. This relationship is iteratively refined with the help of a measure of error in the predictions or probability that is achieved by the model. They often work on the base of statistics and are now used in statistical machine learning. People find this slightly confusing as regression can be used to refer to the class of the problem as well as the class of the algorithm itself, however, basically, regression is said to be a process.

Some of the highly popular regression algorithms include: ● Ordinary Least Squares Regression (OLSR) ● Linear Regression ● Logistic Regression ● Stepwise Regression ● Multivariate Adaptive Regression Splines (MARS) ● Locally Estimated Scatterplot Smoothing (LOESS)

Instance-based Algorithms

Instance-based learning is a model that is a decision problem with examples of training data. These are considered to be important or necessary for the model. These methods often build up a database of example data. This is then compared with the new data to the database with the help of similarity measure. This is done to locate the best match and throw out a prediction. This is the reason why instance-based methods are often known as winner-take-all methods as well as memory-based learning. The primary focus in this method is the representation of the stored instances as well as the similarity measures used between them.

Following are the highly instance-based algorithms: ● K-Nearest Neighbor (kNN) ● Learning Vector Quantization (LVQ) ● Self-Organizing Map (SOM) ● Locally Weighted Learning (LWL)

Regularization Algorithms

These serve as an extension or a secondary method to another method such as regression method. They are used to punish the models by their complicatedness and favor the simpler models that are good at generalizing.

Highly used regularization algorithms are: ● Ridge Regression ● Least Absolute Shrinkage and Selection Operator (LASSO) ● Elastic Net ● Least-Angle Regression (LARS)

Decision Tree Algorithms

Decision tree methods manufacture a model of decisions that have been made based on the real values of attributes in the available data. A fork is formed in the decision tree until a prediction is made for the provided record. They are often trained for the classification of data as well as regression problems. They are quite fast as well as accurate, making them highly popular and a favorite in the world of machine learning.

Standard decision tree algorithms are as follows: ● Classification and Regression Tree (CART) ● Iterative Dichotomiser 3 (ID3) ● C4.5 and C5.0

(different versions of a robust approach) ● Chi-squared Automatic Interaction Detection (CHAID) ● Decision Stump ● M5

- Conditional Decision Trees

Bayesian Algorithms

These methods are known as Bayesian methods as they explicitly apply Bayes' Theorem for various problems that include classification and regression.

The most common Bayesian algorithms are: ● Naive Bayes ● Gaussian Naive Bayes ● Multinomial Naive Bayes ● Averaged One-Dependence Estimators (AODE) ● Bayesian Belief Network (BBN) ● Bayesian Network (BN)

Clustering Algorithms

These are like regression as they describe the class of problem as well as the class of methods. Different modeling approaches organize them, for instance, centroid-based and hierarchal. All of these use the inbuilt structures in the data for a better organization of the data into small groups of maximum commonality.

Following are the highly popular clustering algorithms: ● K-Means ● K-Medians ● Expectation Maximization (EM) ● Hierarchical Clustering

Association Rule Learning Algorithms

It identifies the rules that evaluate the relationships between data and variables. These rules can find many commercially useful and relevant associations in massive and multidimensional data sets. These can be used with organizations.

Some examples include: ● Apriori algorithm ● Eclat algorithm

Artificial Neural Network Algorithms

These are networks that have been modeled and inspired by the functioning and structures of the biological nerves. They are often used for classification problems as well as regression, but it may encompass various other things, as it is a huge field with various epistemological fields and sides.

Highly popular artificial neural network algorithms include: ● Perceptron ● Back-Propagation ● Hopfield Network ● Radial Basis Function Network (RBFN) ● Deep Learning Algorithms

Deep Learning Algorithms

These are modern and highly up to date artificial neural networks that can use cheap and abundant computation. You will find more about these in the following chapters. These concern with the construction of large as well as highly complex networks of neurons.

Some of the most popular deep learning algorithms include: ● Deep Boltzmann Machine (DBM) ● Deep Belief Networks (DBN) ● Convolutional Neural Network (CNN) ● Stacked Auto-Encoders ● Dimensionality Reduction Algorithms

Dimensional Reduction Algorithms

These are like clustering methods, and they try to find and use the inbuilt structure of the data. In these cases, this is done in an unsupervised manner.

This can be useful to visualize dimensional data or to simplify data that can then be used in a supervised learning method.

Ensemble Algorithms

These models are made of various weaker models that have been trained independently to work on specific tasks. The predictions of all these models are then combined to form an ensemble prediction. A lot of effort is put into the types of weak learners as well as how they can be combined. It is supposed to be a highly powerful technique.

Chapter 4: Decision Tree and Random Forests: Part One

This and the following two chapters will deal with the primary subject matter of this books *i.e.* decision trees and random forests. As decision trees and random forests is a vast field, it has been broken down into three chapters for the ease of the reader, so that even if the reader is a beginner or an amateur, they would be able to follow the information in the book. Let us now begin with the basics of decision trees.

What is a Decision Tree? How exactly does it work?

Before beginning any topic, it is necessary to understand the basics of it and what it entitles. Thus, this section will deal with general information regarding Decision Trees.

In simple language a Decision Tree can be defined as a kind or type of learning algorithm that is supervised and has a pre-defined target variable that is commonly used in classification problems. Classification problems are a forte of Decision Trees as they fit perfectly with classification problems.

Decision trees can be used in both continuous input output variables as well as categorical output variables. In this method, the sample is split into two or more homogeneous sets by most significant splitter or differentiator in input variables.

Example:

Let's take a look at a sample of 30 children with three different variables *viz.* Gender (M or F), Town (A or B) and Weight (50 to 60 kg). 15 of these 30 play piano in their free time. Now, the condition is that I want to create a model that will predict who plays piano in their free time. For this problem, it is necessary to differentiate the children and add them in categories such as children who play the piano. This should be based on highly significant input variable in between all the three.

In such a case a decision tree can help a lot as it differentiates and categorizes children using all the all the three variables and will form the most homogenous groups of students that will be heterogeneous to each other.

Decision Tree, Algorithms

Decision trees, as mentioned above finds and identifies the most significant variable as well as its value. This finds the best homogeneous set of data. However, the question that comes out of this is how does it do this, *i.e.* how does it identify the variable as well as the split. The decision tree uses a variety of algorithms to perform this action. These algorithms will be discussed in the next few sections of this chapter.

Types of Decision Trees

The categorization of the decision tree can be based on the kind of target variable that is available. It is found that it can be of two types:

Categorical Variable Decision Tree:

A Decision Tree that has categorical target variable is known as a categorical variable decision tree. For instance: If we are to continue the above example. The target variable in it will be "Children play piano or not." *i.e.* YES or NO.

Continuous Variable Decision Tree:

This is the second type of decision tree. If the tree has an ongoing target variable, it is known as Continuous Variable Decision Tree.

Example: Instead of continuing the above example, let us try something different. Let's assume that a person will pay his tax to the tax department (Yes/No). Here it is seen that the income of the individual is the important variable. However, the tax company may or may not have the details of the person. Now, as we already are aware of the importance of the variable, a decision tree can be constructed to predict the income using variables like the product, occupation, *etc*. In such a case the predictions are happening for a continuous variable.

Terminology and Jargon related to Decision Trees

As machine learning is a complicated and complex topic, it has its language that is full of jargon. Similarly, the field of Decision Tree too has its jargon that is often full of tree metaphors. Let us have a look at the most important idioms.

- Root Node: It stands for the total sample or population. It is then further divided into two or more homogeneous sets.
- Splitting: This is the process that is used to classify nodes into two or more subnodes.
- Decision Node: The further division of subnodes into more subnodes is also known as splitting. However, the node created in this process is known as a decision node.
- Leaf/ Terminal Node: The Nodes that cannot be split are known as Leaf or Terminal node.
- Pruning: Continuing the metaphors of tree and plants, the process in which the subnodes of a decision node are removed, is known as pruning. It is thus an anti-splitting process.
- Branch / Sub-Tree: As the name suggests, the sub-section of a complete tree is known as a sub-tree or branch.
- Parent and Child Node: In the division of the nodes, the node that gets divided is referred as a parent node, while the nodes that are formed due to this division, *i.e.* the subnodes are known as child nodes.

These are the most commonly used jargons in the Decision Tree terminology. Now let us have a look at the advantages and disadvantages of the algorithm.

There exists no perfect machine-learning algorithm, and all algorithms have one problem or the other. Hence, instead of criticizing an algorithm, it is necessary to study its advantages and disadvantages and use it according to your requirements and needs. Below you will find the pros and cons of decision tree algorithm discussed in brief.

Advantages

Easy to understand:

Even for the people who do not have an analytical background, decision tree algorithm is very easy to comprehend. A user does not need to have any statistical knowledge or information to study, read and interpret the trees. Users can easily read the data in decision trees, as the graphical representation is highly intuitive and user-friendly.

Useful in Data exploration:

It is believed that decision tree, if not the fastest, then definitely one of the fastest method of identification of the most significant variable as well as the relation between two or more than two variables. A decision tree can help users to create new variables as well as features. These new features will have more strength for predicting the target variable. It is also usable in the data exploration stage. For instance, let's say the user is working on a project where the data is available in the form of multiple variables. Here the decision tree will work and predict the most significant variable with ease.

Less data cleaning required:

As said in the first chapter, in machine learning, a user has to spend most of their time in data cleaning and separating the good data from the bad data. However, in the case of the decision tree, this process is considerably easy and does not take a lot of time. It remains uninfluenced by the missing values as well as outliers, and thus the cleaning process becomes straightforward.

The data type is not a constraint:

Decision tree is a versatile algorithm, and it can handle categorical as well as numerical data variables with ease.

NonParametric Method:

A nonparametric method means a method that has no assumptions regarding classifier structures or spatial distribution. A decision tree is a nonparametric method.

Disadvantages

Over fitting:

Decision tree models have many difficulties out of which over fitting is the most common and practical. However, this problem can be solved with using constraints on model parameters as well as pruning. An example is discussed in the following section.

Not fit for continuous variables:

Although it can work with continuous variables, it is not at all suitable for it. Decision tree starts losing information when it starts categorizing variables in more and more categories.

Regression Trees vs. Classification Trees

The last stage in a decision tree is the terminal nodes that are also known as the leaves. What this means is that the decision trees are normally constructed upside down, where the leaves are at the bottom while the roots are at the top.

The above-mentioned trees, both work in similar or almost similar ways, however there exist some minor differences that are quite important. Let us go through them one by one.

When the dependent variable is continuous, the regression trees are used whereas when the dependent variable is categorical; the classification trees are used.

The value gained by the terminal nodes in the sample data is the mean of response observed in the case of a regression tree. Therefore, in the case of data observation, its prediction is the mean value.

Whereas, the class gained by the terminal node in the sample data is the mode of observation in the case of the classification tree. Therefore, if some unseen data comes under that region the prediction that takes place the mode value.

Both the types of trees separate the predictor space in different, non-overlapping spaces. They are in a way, high-dimensional boxes.

Both trees use a top-down greedy method. This approach is known as recursive binary splitting. It is a top-down method as it starts from the top of the tree, where the observations are present in a single region and then divides the predictor space into two new branches down the tree. It is said to be ‘greedy’ as, the algorithm that looks for a best available variable in only the current split. It does not concern about the future splits that lead to a superior tree. The splitting or dividing process goes on until the stopping criterion defined by the user is achieved. For instance: the user can order the algorithm to stop as soon as the number of observations in each node goes below than 50.

In such cases, the dividing process or the splitting process gives an output of fully constructed or grown trees until the criterion is reached. However, it is most likely that the data will over-fit the tree. This leads to a bad accuracy in the form of unseen data. Due to this, pruning is necessary. It is normally used for solving the problem of over-fitting. More on this in the next section!

Where does the tree get split?

Where should the tree get split is an important question that all the users ask. This decision often affects the trees' accuracy by a large margin. However, the criteria that are used to make this decision are different in the case of classification and regression trees.

Multiple algorithms are used by the decision trees to make decisions where to split and how much to split. There is an increase in the homogeneity when the sub-nodes are created. This means that the purity of the nodes is enhanced when the target variable increases.

The selection of algorithm is also based on the kind of target variables. The four, most commonly used algorithms in such decision trees are:

Gini Index

In Gini Index, if two items are selected from a population, randomly, then they need to be of the same class. The probability for this will be one of the population will be pure.

- It will work with the categorical target variable such as "Success" or "Failure."
- It can only do Binary splits.
- The more the value of Gini, the more the value of homogeneity.
- Gini method is used to do binary splits by the CART (Classification and Regression Tree).

Let us look at the steps that can be used to calculate by Gini for a split: The following formula is used to calculate the Gini for the subnodes Sum of square of probability for success and failure i.e. (p^2+q^2) Now calculate the Gini for the division for weighted Gini of each node of the created split.

For instance: In our first example, where we need to segregate the children on the basis of target variable *i.e.* playing piano or not.

Decision Tree, Algorithm, Gini IndexSplit on Gender:

Calculate, Gini for subnode Female = (0.2)*(0.2) + (0.8)*(0.8) =0.68

Gini for subnode Male = (0.65)*(0.65) + (0.35)*(0.35) =0.55

Calculate weighted Gini for Split Gender = (10/30)*0.68 + (20/30)*0.55 = 0.59

Similar for Split on Class:

Gini for subnode Class IX = (0.43)*(0.43) + (0.57)*(0.57) =0.51

Gini for subnode Class X = (0.56)*(0.56) + (0.44)*(0.44) =0.51

Calculate weighted Gini for Split Class = (14/30)*0.51 + (16/30)*0.51 = 0.51

Above, you can see that Gini score for Split on Gender is higher than Split on

Class, hence, the node split will take place on Gender.

Chi-Square

This algorithm is used to find the statistical significance of the differences between the parent node and the sub-node. We can calculate it by the sum of all the squares of all the differences between the expected frequencies of target variable and observed frequencies of the target variable.

Categorical target variable “Success” or “Failure” are used.

It can perform two or more than two splits.

The more the value of Chi-Square the higher is the value of the statistical significance of differences between the parent node and the sub-node.

The following formula is used to calculate the Chi-Square of each node: Chi-square = ((Actual – Expected)^2 *Expected)^1*2

The formula forms a tree that is known as CHAID (Chi-square Automatic Interaction Detector) Necessary steps to undertake to calculate Chi-square for a split:

The Chi-square for an individual node is to be calculated by calculating the deviation for ‘Success’ as well as ‘Failure’ both.

The resultant Chi-square are then added and summed to calculate the ultimate Chi-square.

Information Gain, Decision Tree

Now it is possible to construct a conclusion that a low amount of impure nodes need less information to describe them. This information theory is used to measure and define the degree of disorganization in a system. This is known as Entropy. For instance, if a sample is 100% homogeneous then the entropy level is zero, however, if the sample is divided equally then the entropy level is 1 or one.

The lesser the entropy, the better it is.

How to calculate entropy for a split:

Calculate and formulate the entropy of parent node.

Formulate the entropy of each separate node in the split and then calculate the average weighted of all the other nodes that are present in the split.

Reduction in Variance

Until now we have seen the algorithms for target variables. When the variance is reduced, it is an algorithm that can be used in continuous target variables. This algorithm utilizes standard formula of variance that can be used to choose the best split.

How to calculate Variance:

Formulate the variance for each node.

Formulate the variance for every split as a weighted average of every node variance.

Till now we have seen the basics of decision trees as well as the basics of decision-making process that chooses the best splits for a tree model. This tree can be used and applied to both classification problems and regression problems.

How to avoid over-fitting in trees and what are the main parameters in tree modeling?

Over-fitting is one of the major problems that researchers face when constructing decision trees. If a limit does not exist in a decision tree, then it presents you 100% accuracy. However, in the worse case scenario, it will construct one leaf for every observation. Thus, it becomes extremely important to avoid over-fitting while constructing a decision tree. This can be done using the following two methods: ● Setting constraints on tree size ● Tree pruning Let us discuss the first one in brief.

How to set Constraints on Tree Size This can be performed using a variety of parameters that can be used to define a tree.

These parameters are explained below. These are irrespective of the tool used. These parameters are available in R as well as Python.

- Minimum Samples for a node split.
- It can define a number of observations that are needed for a required node for splitting.
- It is also used to limit over-fitting. A large number of values can prevent learning relations.
- A large number of values can also lead to under-fitting.
- It can define the minimum sample for a leaf.
- It can be used to control over-fitting.
- Lower values are appreciated for the imbalanced class problems, as the regions in which minority class turns out to be majority class are minuscule.

Maximum Depth of Tree:

- This can be used to check the depth of a tree.
- It is used to control over-fitting.
- It needs to be tuned using CV.
- It has a large number of terminal nodes.

Chapter 5: Decision Trees: Part 2

In this chapter certain topics such as tree pruning, etc. will be handled.

Tree Pruning

As mentioned in the last chapter, the method of setting a constraint is considered to be a greedy-approach, and thus, it is not recommended. What this means is that the method will just check for the best split and more on forward until one of the many given stopping conditions are achieved. For, instance if we are to use the metaphors of driving then:

Let us assume that there are two lanes.

In lane one, cars are moving at 90km/h.

In lane two, trucks are moving at 40km/h.

You are driving a red car, and now you have two options. You can either take a left turn and overtake the two cars that are driving in front of you, as soon as possible, or continue to be in your lane.

Let us now go through these choices one by one. In the choice number one, you will take over the cars immediately and then move into a lane where the trucks are driving. Now you will have to drive in the lane with the trucks at the speed of 40km/h until a spot opens up in the old lane. Meanwhile, the cars in the original lane, which were behind you, have moved on and gone ahead of you. Your choice would be the best choice if your mission were to cover the most distance in around 20 seconds. However, in the latter choice, you will continue with your speed, cross the trucks and then get a chance to take over and go ahead.

This is the difference between pruning and normal decision tree. A decision tree that has constraints will not be able to see the ‘truck’ or the obstruction in front of it and will try to take a greedy approach. However, if pruning is used, the method will take a few steps back and will get a chance to think upon the condition before acting on it.

This means that pruning is better than the other option. So how to use it?

For utilizing pruning, it is necessary to make decision tree with a large depth.

Then we need to begin at the bottom and then slowly and gradually start cutting off the leaves that are presenting negative results as compared from the top.

For instance, if a split is giving a result of -10 *i.e.* the loss of 10, then in the subsequent split should give us a gain of 20.

A simple decision tree will halt its working at step one, however, after pruning, we will begin to see the overall gain rising to +10 all the while keeping both the leaves.

Linear models or tree based models?

The use of the algorithm is dependent on the type of problem that you are trying to solve. Let us have a look at some factors that you need to consider and that can help you decide the algorithm that is to be used when solving a problem: If the relationship between independent and dependent variable is approximated using a linear model, the linear regression will outperform the model.

If there exists a high nonlinearity as well as the complex relationship between the independent & dependent variables, a tree model will serve better than a regular method.

A decision table is a much easy to understand and explain model as compared to a linear model.

Ensemble methods:

The dictionary meaning of the word ensemble is a group. Thus, ensemble methods are made of a group of predictive models that can gain better stability and accuracy. They are also known to boost the tree-based models.

Like all other models that we have seen by now, a decision tree based model too has certain problems. These problems include bias and variance. Here bias stands for the difference between the average values that predicted than the actual values. Variance refers to the variety of the predictions of models at the same point if the samples are taken from one same population.

When the complexity of the model is increased, it is possible to see the reduction in the prediction error thanks to lower bias in the model. However, when you gradually continue to build a more complicated model you can over-fit your model, and thus your model may suffer from the high variance.

A strong and highly powerful model should be able to maintain a proper balance between the above-mentioned types of errors. This is also known as the tradeoff management of bias-variance errors. One of the best ways to do the tradeoff analysis is to perform ensemble learning.

The most common methods of ensemble learning are as follows: ● Bagging ● Boosting ● Stacking In this chapter, we’ll focus on Bagging in detail while Boosting will be handled in the next chapter.

What is bagging? How does it work?

Bagging is a method that is used to decrease the variance of the predictions by summing the output of two or more classifiers that are modeled on different samples in the same data set.

The following steps are followed in bagging:

Create Multiple Data sets:

The original data is replaced in sampling with the formation of new data sets.

The new data sets may contain fractions of the rows and columns. These are normally hyper parameters in a bagging model.

If the columns and rows are taken in the fraction that is less than one, it can help to make the model strong and robust. It can also reduce over-fitting.

Build Multiple Classifiers:

These are constructed with each new data set.

Normally the same classifier is constructed on every data set, and the predictions are made later.

Combine Classifiers:

All the predictions of every classifier are combined with the help of a mode, a mean or a median. This depends on the problem on the hand.

The combined values are normally stronger than a single value.

It is important to notice that the number of models that are to be constructed here is not supposed to be a hyper parameter. A greater number of models are almost always better than lower numbers. However, in certain cases, they can also perform on similar planes and thus can give a similar performance to the lower numbers. It is possible to show this theoretically that the variance of all the combined predictions are reduced to 1/n where n is the number of classifiers, of the normal variance under certain assumptions.

A variety of implementations of bagging models exist. Random forest is one of such models. It will be discussed it in the next and the last chapter of the decision tree series.

Chapter 6: Decision Trees: Part Three (Random Forests)

In this chapter let us focus on the basics of Random Forests and how it performs its tasks. Random Forest can be called as a universal solution as it is said if you do not know which algorithm to use, you should use a random forest.

The random forest can be defined as a versatile and smart machine learning method that can perform both classifications as well as regression tasks. It can also perform dimensional reduction methods, outlier values, treat missing values and other steps of data exploration as well. It is an expert solution for most of the problems. It is known as an ensemble way of learning as a group of weak models are combined to form this, powerful model.

Workings of Random Forest:

Multiple decision trees are grown in a random forest. This is opposite to the CART model. For the classification of new object based attributes, every one of the trees presents a classification. This presentation is also known as ‘voting’ for the class. The forest then is given a choice to choose the classification with the most votes.

For instance:

Let's say that the number of cases in one training set is N. Then the sample of the N cases can be taken at random. However, it must be replaced. The sample will serve as the training set for the growing tree.

If there exist M input variables then, m<M is specified in a way that at each node, the m variables will be selected at random out of the M. The split will happen on M and m is used to split node.

All the trees are grown as much as they can grow, and no pruning is done.

The prediction of new data takes place by aggregating the other predictions of the ntree.

Advantages of Random Forest

- Random forest algorithm can be used in both sorts of problems. It can be used in regression as well as classification.
- It can handle a large amount of data set in high dimensionality. It can handle more than a few thousands of variables and can very well identify the significant ones among them. It is therefore considered to be an important dimensionality reduction method.
- It can effectively estimate the missing data and can easily maintain accuracy even if it is fed a large amount of data.
- It has various methods that can be used to balance errors in the data set.
- The above features can be used with unlabeled data as well. Thus, it can work unsupervised.
- It samples input data with replacement. This process is known as bootstrap sampling.

Disadvantages of Random Forest

- Like advantages, Random Forest has certain disadvantages too. However, as compared to the disadvantages, the number of advantages is large, thus making Random forests a far better option than other options.
- It is not as good at regression as it is with classification. It often does not come out with precise, continuous nature predictions. It cannot make predictions beyond the range of the provided training data in the case of regression.
- The data may become over-fit if the sample data is too noisy.
- It can act as a black box approach for statistical modelers as you cannot control the performance of the model. You can only try random seeds and different parameters.

What is Boosting? How does it work?

To define in simple terms, boosting is a family of algorithms that transform weak learners to strong or stronger learners.

The definition can be further understood by the use of the following example that deals with spam email identification:

If you are asked to identify Spam and regular mail, what will be the criteria that you will follow to perform your task? Normally, the following steps will be followed:

- If the email contains only one, promotional image: SPAM
- It only has link or links: SPAM
- The body of the email contains only sentences such as “You won a prize money of ---”: SPAM
- Email received from an official domain: Not a SPAM
- Email received from a known sender: Not a SPAM

Thus, these are the rules or the criteria that we use to classify whether an email is a spam or not spam. However, do you firmly believe that these criteria are sufficient enough to classify emails in spam and not spam? The answer to the above question is no.

These rules cannot be said to be powerful enough to classify email as spam or not spam on their own. Hence, these rules are known as a weak learner. However, these weak learners can be converted into strong ones by combining the predictions made by the weak learners with the help of various methods such as:

By utilizing average or weighted average

Choosing the prediction that has a higher vote rate.

For instance: In the above example we have five weak learners. If three out of these five are said to be SPAM, then the remaining two will be voted as Not Spam. Therefore, we will continue to consider an email SPAM because the votes are higher in the case.

How does it work?

As we have seen that boosting actually sums the weak learners to form a strong rule. However, the question that should arise in your mind is, how does it identify the weak rules?

To find the weak rule it is necessary to apply the machine-learning algorithm with a varied distribution first. Every time this algorithm is applied, a new weak prediction rule is formed. This is a repetitive process and is repeated many times. The boosting algorithm combines these rules into one single strong prediction rule.

How do we choose a different distribution for each round?

To choose the correct distribution, the following steps can be followed:

1. All the distributions are taken and assigned by the base learner to equal weight and attention.
2. If any prediction error happens due to the first base learning algorithm, more attention is then paid to the observation that has caused the error. It is then applied to the next base-learning algorithm.
3. The second step is repeated until the base learning algorithm limit is reached or a high level of accuracy is gained.

In the last step, it sums up all the results received from the weak learners and makes a strong learner. This learner can improve the prediction power of the model eventually. Boosting is highly focused on mis-classified and high error examples that are preceded by weak rules.

Many boosting algorithms exist that can enhance the accuracy of boosting models even more. Following is a tutorial where you can learn even more about the two most commonly used boosting methods: Gradient Boosting (GBM) and XGBoost.

GBM or XGBoost: Which is more powerful?

Many researchers have admired the boosting capabilities of XGBoost. It often gives out a better output as compared to GBM however, in many cases it is seen that the benefits are less, often inconsequential. XGBoost is better than GBM for the following reasons:

Regularization:

XGBoost has regularization, unlike Standard GBM implementation that has no regularization. Regularization can control the amount of over-fitting. This is why the other name for XGBoost is ‘regularized boosting technique.'

Parallel Processing:

Unlike GBM XGBoost has parallel processing, which makes it considerably faster than regular boosting.

A question that might arise in your mind is that how boosting, a serialized or sequential process can be parallelized? This can be explained by the following simple clarification. A decision tree can only be constructed on the base of an older one or a previous one. Hence, it is possible to create or grow a new tree from each core. Thus, it is a versatile method.

XGBoost can also work with Hadoop.

High Flexibility

XGBoost is far more flexible than GBM as it allows the user to optimize and personalize whatever evaluation criteria ass well as optimization objectives that the user wants to change. This makes it highly powerful as it practically removes the limitations from the field and it possible to do whatever you want.

XGBoost is a better option than GBM because it can handle missing values better than GMB. The user can supply a different value that is not the observation and can pass it as a parameter. XGBoost will try to perform different tasks as it can locate a missing value on every node and can learn the path as well.

Tree Pruning:

A GBM often stops splitting nodes when it locates a negative loss in the split. Therefore, it is considered to be a greedier algorithm. Compared to GBM, XGBoost can split till the max_depth of the value is achieved and then it starts the pruning mechanism where it removes the splits, going backward when it cannot find a positive gain.

Built-in Cross-Validation

XGBoost has a built in mechanism of cross-validation that is run at the each repetition. Thus, it becomes extremely easy to derive the perfect number of boosting repetitions in a single run. This is not seen in the case of GBM where you have to run a grid-search, and only some specific values get tested in the mechanism.

Continuing the Existing Model

It possible to start the XGBoost model from any last iteration in the last run. This is highly important in certain applications.

How to work with GBM in R and Python?

It is necessary to understand the necessary parameters before we have a look at the algorithm itself. These can be used in R as well as Python.

Following is the pseudo-code of GBM algorithm for two classes:

Start the outcome.

1. Repeat from 1 to the actual number of trees.
2. Add the weights for the targets after each run. These will be based on the previous run itself.
    - 2.1. Fix the model on the selected data subsample.
    - 2.2. Perform predictions till a full set of observation is done.
    - 2.3. Update the results with the current results. Identify the learning rate.
3. Return the last output.

Although this is a basic and very naïve explanation of GBM, yet it will help the beginners to understand the workings of the algorithm.

Now let us have a look at the parameters that are used in Python:

learning_rate

This identifies the impact of every single tree on the final result. GBM begins the work by initializing the starting estimate. This is then updated with the help of result of all the subsequent trees. The learning parameter is used to control the magnitude of the change that is seen in the estimates.

In this example, lower values are more appreciated as they can make the model stronger to certain characteristics of the tree. This allows them to generalize well.

n_estimators

This stands for the number of sequential trees that need to be modeled. GBM is quite robust at a large number of trees. However, it may over-fit at a certain point. Hence, this parameter is used to tune the learning rate.

Subsample

This is the fraction of observations that are to be chosen for every tree. This is performed using random sampling.

Values that are less than 1 can make the model strong by slowing down the variance. Generally, values of ~0.8 used and work great.

Apart from the above mentioned, there are certainly other parameters that can be used to enhance the performance of the method.

Loss

It is in reference to the loss function. This is to be minimized with each split.

This parameter can have many values for both, regression and classification. Normally the default values are used. Other values are to be used only if you understand their role.

Init

This parameter affects the initialization of the output. It can be used if another model whose result is to be used as the initial estimates for GBM has been constructed.

random state

This is the random number seed due to which the same random numbers get generated whenever they are used.

This parameter is necessary for tuning. If a random number is not fixed, then we cannot have different results for all the following runs on the same parameters.

This makes it difficult to compare the models.

Verbose

This refers to the type of output that is to be printed when the model fits well. The values can be varied, and they are as follows:

1. 0: No sort of result is generated. This is the default value.
2. 1: Result is generated at only certain intervals for the trees.
3. >1: Result is generated for all the trees.

warm_start

This is an interesting parameter, and it can help the user in a significant way if it is used judicially and properly. This parameter can be used to fit extra trees on the previous models with ease. This will allow you to save a lot of time and thus allow you to conduct various advanced applications.

Presort

This is used when you need to choose whether to presort the data for quicker splits. It creates an automatic selection by default. However, the values can be changed if necessary.

For R users who use caret package these are the main tuning parameters: ● N.trees – This is the number of repetitions and the tree that will be used to grow the subsequent trees.

- Interaction.depth – It identifies the complexity of the tree or the total number of splits that the tree has to perform.
- Shrinkage – This is used in the reference of learning rate. It similar to the learning rate as presented in python.
- N.minobsinnode – It is the minimum number of training samples that are needed for a node to continue splitting.

Where to practice?

To master any skill or concept, it is necessary to practice regularly. Similarly, you also need to practice regularly if you want to master the above-mentioned algorithms.

Practicing is easy for the above-mentioned algorithms as there are many online services where you can find practice games and tests. These tests are updated frequently, and you can also participate in a variety of competitions that can help you master algorithms. These competitions often feature a global level leaderboard, so not only can you practice your algorithms, but also can show off your skills and progress to the whole world. You can also find offline practice sets that can be used to practice when you cannot connect to the Internet.

Thus, here ends the three part series of Decision Tree learning series in this book. In the next chapter, let us have a look at some other forms of machine learning that becoming highly popular in today's world.

Chapter 7: Deep Learning

Till now we have discussed the basics of machine learning and the basics and working so f decision trees and random forests as well. However, it is clear that the field of machine learning does not end here; rather it grows in size as well as complications hereafter. One of the major concepts that are becoming highly popular nowadays is Deep learning. This chapter will serve as a minor introduction to this method of learning.

Deep learning or hierarchical learning is said to be the use of non-biological neural networks to study and learn the aspects that have one or more than one hidden layers. It is situated in the learning data representations. In the method, the learning can be totally supervised, partially supervised or not supervised at all.

The architectures of deep belief system, deep learning system, recurrent neural networks as well as deep neural networks all are used and applied in many fields including computer vision, speech recognition, machine translation, social network filtering, natural language processing, bioinformatics, sound recognition, *etc.* Here these methods give out outputs that can easily compete with the results that are given out by human users.

Deep learning thus is a family or class of machine learning algorithm that uses a cascade of more than two layers of nonlinear processing units for feature extraction and transformation. These layers are cascading and nested therefore the output, or the result of the last layer becomes the input or the sample for the next or following layer. These can be supervised or unsupervised, ex-classification and pattern analysis. The unsupervised ones use multifaceted levels of data. They also use the representations of the data. The method is nested here where the higher level features are based on, the lower level features.

The World School Council of London first designed deep learning.

The difference between Machine Learning, Deep Learning, and AI:

The main difference or rather the main characteristic difference between the above-mentioned three concepts is that Deep learning is a type of machine learning whereas machine learning is a form of AI or artificial intelligence. Machine learning is considered to be the most popular approaches or forms of artificial intelligence. However, not all of the systems based on AI use machine learning, for instance, self-driven or automatic cars use rule-based systems.

It has been prophesized that in the coming future machine learning will be the most prominent form of AI.

Deep learning can be defined as a type or kind of approach towards machine learning that is extremely popular and advanced.

Though the future of machine learning is bright, however, it is not concrete. If another technique is invented that turns out to be better than machine learning, it will be replaced as soon as possible, without any issue.

Chapter 8: Digital Neural Network and Computer Science

The modern computer and computing were invented in the 20th century. People everywhere have guessed and tried to speculate the achievements and innovations that the computers might create in the future. Some of them include talking to humans, recognizing their faces as well as gestures, playing games such as chess against humans, driving cars, *etc.* It is to be noted that computers have and are still performing all the above-mentioned activities. Most of these are due to the development and growth of AI and machine learning, especially in the case of Digital Neural Network.

Deep learning is a form of sophisticated and highly developed form of neural networks. These were first developed and used about 70 years ago. Two scientists, Walter Pitts and Warren McCullough in 1944, first thought of the idea. However, the research soon changed its trajectory, and it stopped developing. However, in the 90s it rose again, and now it is considered to be the most important concept in the AI.

Artificial neural networks are also known as ANN are computing systems that are based on biological neural networks commonly seen in the brains of animals, especially animals. Thus, it is an imitation of the intricate system of neurons that have been developed by nature.

The best feature of this system is that it learns and performs tasks by inspecting examples without the need of task-specific programming. This thus decreases the time as well as the resources that are required.

An ANN group is constructed of multiple artificial neurons. These are connected with each other where the connections are used to transmit and transfer signals from one neuron to the other. They are also organized in multiple layers. Each of these layers performs a different and varied function. The signals normally travel many times through the whole set.

The biggest goal of ANN is the imitation of the human brain and the methods that are used by the human brain to solve these problems. However, nowadays the focus of ANN is changing and now it is more concentrated on the problem solving itself and not on imitating the neural networks. This is why scientists are coming up with many techniques and networks that are quite different than the natural ones but are far more powerful than the regular ones.

Applications of ANN

ANN is used in applications such as ● Solar energy to model and design better gadgets and appliances.

- ● Also used in system modeling.
- ● Can easily handle complicated and incomplete data.
- ● Can be used to solve nonlinear problems.
- ● Air conditioning systems ● Ventilation ● Refrigeration ● Heating ● Power Generation ● Load forecasting ● Estimation of heating loads in buildings and construction.
- ● Robotics ● Forecasting ● Pattern Recognition ● Medicine ● Manufacturing ● Social and psychological sciences ● Signal processing ● Optimization.
- ● Prediction of air flow ● Prediction of consumption of energy in a solar building.

Thus, it can be seen that artificial neural network can work wonders in the field of computers.

Advantages of ANN

- The neural network is able to perform tasks that cannot be performed by the linear network.
- As it is a nonlinear network, even if one or more elements fail, the network still continues to process and work.
- It need not be reprogrammed again and again.
- It is easy to use.
- It is an adaptive method of learning. It is highly robust and can solve a variety of difficult problems with relative ease. Can be used in a variety of applications.
- The advantages of ANN are far more than the disadvantages of risks that are associated with the technology.
- It uses simple samples to come up with results and responses.
- It is extremely flexible and adaptable. It can learn easily with very fewer variables.
- It is more forgiving to the user generated errors and thus is much more suitable than a linear network.
- Based on adaptive learning.

Risks associated with ANN

There are certain risks involved while using ANN. However, it can be seen that the list of advantages is much longer the list of risks:

- A user needs a lot of training to learn how to use ANN properly. However, once learned, it is quite easy to use.
- It is a time-consuming process.
- The architecture is different than microprocessor hence proper emulation is needed before using ANN.

Types of Artificial Neural Networks

Multiple types of ANN or artificial neural exists, out these, the following three are important.

- Feedback ANN: The result goes into the network itself, and the process gets repeated until the perfect result or output is achieved.
- Feed Forward ANN: It is a simple network. It contains an output layer, an input layer and one or more layer of neurons
- Classification Prediction ANN: It is a subset of the feed forward artificial neural network.

Summary:

The goal of AI (Artificial intelligence) is to mimic human capabilities, such as knowledge application, abstract thinking, reasoning, *etc.* On the contrary machine learning operates on the principles of using writing software based on past experiences.

To our surprise machine learning in actual fact is very similarly related to data mining and statistics as oppose to AI. Why is that? In essence when a computer program performs a particular task based on experiences from the past, than we can conclude the machine has "learnt". This is much different from a program performing a task as its' programmers have comprehensively defined its capacities already.

Machine learning can be placed into 3 distinct categories:

***Supervised Learning* -** Teach and train machines with data that is already labeled with the correct answers and outcomes. The more data available the more the machine will learn from the given topic or subject matters. For instance, an apple is red, banana yellow , and broccoli is green. After the machine is trained it is given new unseen data and the learning algorithm uses its past experiences to determine an outcome.

Unsupervised Learning- when a machine is trained with data sets that do not have labels. The learning algorithm is not told what the data represents. So for example, here is a letter or an apple, but no further information is given on its details. Or perhaps describing characteristic of a certain object, like a tree, but not giving the label of its name.

So how does this work? Imagine listening to a podcast of a foreign language like Mandarin or Hindi. You listen to hundreds upon hundreds of podcasts and start to establish patterns, form models, and recognize certain sounds. -Unsupervised learning operates much in the same way.

Reinforcement Learning**—**similar to unsupervised learning, the data is not labeled. But when asked a question of the data, the outcome will be evaluated. So for instance, imagine playing against a computer in the game of chess.

When the computer wins it than evaluates the result to validate how it won and thus reinforces the action. Ergo, when the computer engages in playing thousands of games it collectively establishes a strategy and framework in winning.

There is this buzz word "Neural Net" we hear of in the world of machine learning, but what is it exactly? Its modeled after our brain neural networks and operates synonymously to a neuron. When a given number of inputs are given the neural will propagate a signal depending on how it interprets it.

The Above mentioned in this summary sub chapter is the fundamental concepts of machine learning. This is how machine learning functions and operates, and much of what we utilize today without even knowing it!

Conclusion

This is the era of technology, and thus it is changing and evolving every day. Every day, it is becoming extremely difficult to keep up with the new innovations, latest trends, discoveries and inventions in almost every field. This is especially true in the case of science and technology. Science and technology are the two fields that are developing the most rapidly in today's world. Hence, to keep up with such a never seen before pace, a person needs to be extremely up to date and techno-savvy. However, this not possible for everyone. Hence, books like this one help general population to understand the basics of the current buzzwords and concepts, so that they can be relevant. This book is an attempt to serve such population by helping them understand the basics of machine learning, AI, and decision trees in simple and lucid manner.

The book is best suited for beginners in who are interested in machine learning. However, you can also find information that is suitable for adepts as well. For the ease of the general reader, the language of the book has been kept simple and relatively jargon free. The book is divided into multiple sections so that you can only read the sections that are important for you.

The main concern of this book, tree based algorithms is highly important for all sort of scientists that deal with data. They are known to provide the best performing model as compared to other models in the machine learning family. In this book, different aspects of decision trees and random forests have been discussed. Tree based modeling has been discussed from scratch. I am sure that you must have learned the importance of decision tree and how a biological concept is being used to upgrade the technology.

I am sure that this book will act as a basic guide for everyone who is interested in AI. As stated earlier, it is written in a simple and lucid manner, so it suitable for every member of your family. As an added bonus, a few chapters have been dedicated to deep learning and neural networking. These will help as a guide if you want to study advanced machine learning. I also hope that this book has done away with any myths that you had about machine learning and AI.

Once again thank you for buying this book and good luck!

If you enjoyed reading this book please leave a quality review on Amazon. It would be very helpful and appreciated.

Thank you.