Machine Learning Techniques for learning features of any kind of data: A Case Study

P.Anitha, G.Krithka, Mani Deepak Choudhry

***Abstract*— Data mining is a particular data analysis technique that focuses on modeling and knowledge discovery for predictive rather than purely descriptive purposes. Business intelligence covers data analysis that relies heavily on aggregation, focusing on business information. In statistical applications, some people divide data analysis into descriptive statistics, exploratory data analysis (EDA), and confirmatory data analysis (CDA). Machine Learning is a scientific discipline that explores the construction and study of algorithms that can learn from data. Face recognition present a challenging problem in the field of image analysis and computer vision. It can be broadly classified into 3 categories based on face acquisition methodology. In this paper, an overview of some of the well-known methods is provided and some of the benefits and drawbacks of the schemes are examined.**

***Index Terms*—Machine Learning, Face Recognition, Feature Extraction, SVM, Supervised, Semi-Supervised.**

I. Introduction

A. Data Mining

Data mining automates the detection of relevant patterns in a database, using defined approaches and algorithms to look into current and historical data that can then be analyzed to predict future trends. Because data mining tools predict future trends and behaviors by reading through databases for hidden patterns, they allow organizations to make proactive, knowledge-driven decisions and answer questions that were previously too time-consuming to resolve.

Data mining is not all about the tools or database software that being used. The user can perform data mining with comparatively modest database systems and simple tools, including creating and writing user's own, or using off the shelf software packages. Complex data mining benefits from the past experience and algorithms defined with existing software and packages, with certain tools gaining a greater affinity or reputation with different techniques.

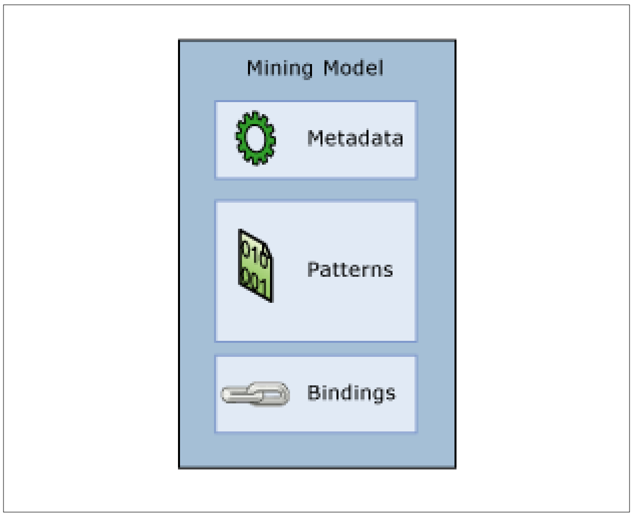

Fig.1. Mining Model

B. Annotation

An **annotation** is metadata (e.g. a comment, explanation, and presentational markup) attached to text, image, or other data. Often annotations refer to a specific part of the original data.

Text annotation is the practice and the result of adding a note or gloss to a text, which may include highlights or underlining, comments, footnotes, tags, and links. Text annotations can include notes written for a reader's private purposes, as well as shared annotations written for the purposes of collaborative writing and editing, commentary, or social reading and sharing. In some fields, text annotation is comparable to metadata insofar as it is added post hoc and provides information about a text without fundamentally altering that original text. Text annotations are sometimes referred to as marginalia, though some reserve this term specifically for hand-written notes made in the margins of books or manuscripts. This article covers both private and socially shared text annotations, including hand-written and information technology-based annotation, as well as Web-based text annotation.

The structural components of any annotation can be roughly divided into three primary elements:

- ➢ A Body,
- ➢ An Anchor and
- ➢ A Marker.

The body of an annotation includes reader-generated symbols and text, such as handwritten commentary or stars in the margin. The anchor is what indicates the extent of the original text to which the body of the annotation refers; it may include circles around sections,

Manuscript received December , 2014.

***Mrs.P.Anitha**, Assistant Professor, Department of Computer Science & Engineering, Dr.N.G.P.Institute of Technology, Coimbatore, Tamil Nadu, India*

***Mrs.G.Krithika**, Lecturer, Department of Information Technology, Dr.N.G.P.Institute of Technology, Coimbatore, Tamil Nadu, India.*

***Mr.Mani Deepak Choudhry**, Assistant Professor, Department of Information Technology, Dr.N.G.P.Institute of Technology, Coimbatore, Tamil Nadu, India.*

brackets, highlights, underlines, and so on. Annotations may be anchored to very broad stretches of text (such as an entire document) or very narrow sections (such as a specific letter, word, or phrase). The marker is the visual appearance of the anchor, such as whether it is a grey underline or a yellow highlight. An annotation that has a body (such as a comment in the margin) but no specific anchor has no marker.

A **web annotation** is an online annotation associated with a web resource, typically a web page. With a Web annotation system, a user can add, modify or remove information from a Web resource without modifying the resource itself. The annotations can be thought of as a layer on top of the existing resource, and this annotation layer is usually visible to other users who share the same annotation system. In such cases, the web annotation tool is a type of social software tool.

Web annotation can be used for the following purposes:

- ➢ To rate a Web resource, such as by its usefulness, user-friendliness, suitability for viewing by minors.
- ➢ To improve or adapt its contents by adding/removing material, something likes a wiki.
- ➢ As a collaborative tool, e.g. to discuss the contents of a certain resource.
- ➢ As a medium of artistic or social criticism, by allowing Web users to reinterpret, enrich or protest against institution or ideas that appear on the Web.

Automatic image annotation (also known as automatic image tagging or linguistic indexing) is the process by which a computer system automatically assigns metadata in the form of captioning or keywords to a digital image. This application of computer vision techniques is used in image retrieval systems to organize and locate images of interest from a database.

This method can be regarded as a type of multi-class image classification with a very large number of classes - as large as the vocabulary size. Typically, image analysis in the form of extracted feature vectors and the training annotation words are used by machine learning techniques to attempt to automatically apply annotations to new images. The first methods learned the correlations between image features and training annotations, then techniques were developed using machine translation to try to translate the textual vocabulary with the 'visual vocabulary', or clustered regions known as ***blobs***. Work following these efforts has included classification approaches, relevance models and so on.

The advantages of automatic image annotation versus content-based image retrieval (CBIR) are that queries can be more naturally specified by the user. CBIR generally (at present) requires users to search by image concepts such as color and texture, or finding example queries. Certain image features in example images may override the concept that the user is really focusing on. The traditional methods of image retrieval such as those used by libraries have relied on manually annotated images, which is expensive and time-consuming, especially given the large and constantly growing image databases in existence.

C. Machine Learning Techniques

Machine learning is a scientific discipline that explores the construction and study of algorithms that can learn from data. Such algorithms operate by building a model based on inputs and using that to make predictions or decisions, rather than following only explicitly programmed instructions.

Machine learning can be considered a subfield of computer science and statistics. It has strong ties to artificial intelligence and optimization, which deliver methods, theory and application domains to the field.

Machine learning tasks are typically classified into three broad categories, depending on the nature of the learning "signal" or "feedback" available to a learning system. These are:

- ➢ **Supervised learning**. The computer is presented with example inputs and their desired outputs, given by a "teacher", and the goal is to learn a general rule that maps inputs to outputs.
- ➢ **Unsupervised learning**, no labels are given to the learning algorithm, leaving it on its own to find structure in its input. Unsupervised learning can be a goal in itself (discovering hidden patterns in data) or a means towards an end.
- ➢ **In reinforcement learning**, a computer program interacts with a dynamic environment in which it must perform a certain goal (such as driving a vehicle), without a teacher explicitly telling it whether it has come close to its goal or not. Another example is learning to play a game by playing against an opponent.

II. MACHINE LEARNING TECHNIQUES

A. Supervised Learning Techniques

Supervised learning is the machine learning task of inferring a function from labeled training data. The training data consist of a set of *training examples*. In supervised learning, each example is a *pair* consisting of an input object (typically a vector) and a desired output value (also called the *supervisory signal*). A supervised learning algorithm analyzes the training data and produces an inferred function, which can be used for mapping new examples. An optimal scenario will allow for the algorithm to correctly determine the class labels for unseen instances. This requires the learning algorithm to generalize from the training data to unseen situations in a "reasonable" way.

In order to solve a given problem of supervised learning, one has to perform the following steps:

1. Determine the type of training examples. Before doing anything else, the user should decide what kind of data is to be used as a training set. In the case of handwriting analysis, for example, this might be a single handwritten character, an entire handwritten word, or an entire line of handwriting.
2. Gather a training set. The training set needs to be representative of the real-world use of the function. Thus, a set of input objects is gathered and corresponding outputs are also gathered, either from human experts or from measurements.
3. Determine the input feature representation of the learned function. The accuracy of the learned function depends strongly on how the input object is represented. Typically, the input object is

transformed into a feature vector, which contains a number of features that are descriptive of the object. The number of features should not be too large, because of the curse of dimensionality; but should contain enough information to accurately predict the output.

4. Determine the structure of the learned function and corresponding learning algorithm. For example, the engineer may choose to use support vector machines or decision trees.
5. Complete the design. Run the learning algorithm on the gathered training set. Some supervised learning algorithms require the user to determine certain control parameters. These parameters may be adjusted by optimizing performance on a subset (called a *validation* set) of the training set, or via cross-validation.
6. Evaluate the accuracy of the learned function. After parameter adjustment and learning, the performance of the resulting function should be measured on a test set that is separate from the training set.

A wide range of supervised learning algorithms is available, each with its strengths and weaknesses. There is no single learning algorithm that works best on all supervised learning problems

i. Support Vector Machine

Support Vector Machine - In machine learning, **support vector machines** (**SVMs**, also **support vector networks**) are supervised learning models with associated learning algorithms that analyze data and recognize patterns, used for classification and regression analysis. Given a set of training examples, each marked as belonging to one of two categories, an SVM training algorithm builds a model that assigns new examples into one category or the other, making it a non-probabilistic binary linear classifier. An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall on.

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.

A support vector machine constructs a **hyperplane** or set of hyperplanes in a high- or infinite-dimensional space, which can be used for classification, regression, or other tasks. Intuitively, a good separation is achieved by the hyperplane that has the largest distance to the nearest training data point of any class (**so-called functional margin**), since in general the larger the margin the lower the generalization error of the classifier.

The original problem may be stated in a finite dimensional space, it often happens that the sets to discriminate are not linearly separable in that space. For this reason, it was proposed that the original finite-dimensional space be mapped into a much higher-dimensional space, presumably making the separation easier in that space. To keep the computational load reasonable, the mappings used by SVM schemes are designed to ensure that dot products may be computed easily in terms of the variables in the original space, by defining them in terms of a kernel function $k(x, y)$ selected to suit the problem. The hyperplanes in the higher-dimensional space are defined as the set of points whose dot product with a vector in that space is constant. The vectors defining the hyperplanes can be chosen to be linear combinations with parameters α_i of images of feature vectors that occur in the data base. With this choice of a hyperplane, the points x in the feature space that are mapped into the hyperplane are defined by the relation: $\sum_i \alpha_i k(x_i, x) = \mathrm{constant.}$ Note that if $k(x, y)$ becomes small as y grows further away from x, each term in the sum measures the degree of closeness of the test point x to the corresponding data base point x_i. In this way, the sum of kernels above can be used to measure the relative nearness of each test point to the data points originating in one or the other of the sets to be discriminated. Note the fact that the set of points x mapped into any hyperplane can be quite convoluted as a result, allowing much more complex discrimination between sets which are not convex at all in the original space.

Applications

SVMs can be used to solve various real world problems:

- SVMs are helpful in text and hypertext categorization as their application can significantly reduce the need for labeled training instances in both the standard inductive and transductive settings.
- Classification of images can also be performed using SVMs. Experimental results show that SVMs achieve significantly higher search accuracy than traditional query refinement schemes after just three to four rounds of relevance feedback.
- SVMs are also useful in medical science to classify proteins with up to 90% of the compounds classified correctly.
- Hand-written characters can be recognized using SVM

ii. Decision Tree Learning

Decision tree learning uses a decision tree as a predictive model which maps observations about an item to conclusions about the item's target value. It is one of the predictive modeling approaches used in statistics, data mining and machine learning. Tree models where the target variable can take a finite set of values are called **classification trees**. In these tree structures, leaves represent class labels and branches represent conjunctions of features that lead to those class labels. Decision trees where the target variable can take continuous values (typically real numbers) are called **regression trees**.

In decision analysis, a decision tree can be used to visually and explicitly represent decisions and decision making. In data mining, a decision tree describes data but not decisions; rather the resulting classification tree can be an input for decision making.

Decision tree learning is a method commonly used in data mining. The goal is to create a model that predicts the value of a target variable based on several input variables. An example is shown on the right. Each interior node corresponds to one of the input variables; there are edges to children for each of the possible values of that input variable. Each leaf represents a value of the target variable given the values of the input variables represented by the path from the root to the leaf.

A decision tree is a simple representation for classifying examples. Decision tree learning is one of the most successful techniques for supervised classification learning. For this section, assume that all of the features have finite discrete domains, and there is a single target feature called the classification. Each element of the domain of the classification is called a class. A decision tree or a classification tree is a tree in which each internal (non-leaf) node is labeled with an input feature. The arcs coming from a node labeled with a feature are labeled with each of the possible values of the feature. Each leaf of the tree is labeled with a class or a probability distribution over the classes.

A tree can be "learned" by splitting the source set into subsets based on an attribute value test. This process is repeated on each derived subset in a recursive manner called recursive partitioning. The recursion is completed when the subset at a node has all the same value of the target variable, or when splitting no longer adds value to the predictions. This process of *top-down induction of decision trees* (TDIDT) is an example of a greedy algorithm, and it is by far the most common strategy for learning decision trees from data.

In data mining, decision trees can be described also as the combination of mathematical and computational techniques to aid the description, categorization and generalization of a given set of data.

Decision trees used in data mining are of two main types:

- **Classification tree** analysis is when the predicted outcome is the class to which the data belongs.
- **Regression tree** analysis is when the predicted outcome can be considered a real number (e.g. the price of a house, or a patient’s length of stay in a hospital).

The term **Classification and Regression Tree (CART)** analysis is an umbrella term used to refer to both of the above procedures, first introduced by Breiman et al. Trees used for regression and trees used for classification have some similarities - but also some differences, such as the procedure used to determine where to split.

Some techniques, often called *ensemble* methods, construct more than one decision tree:

- **Bagging** decision trees, an early ensemble method, builds multiple decision trees by repeatedly resampling training data with replacement, and voting the trees for a consensus prediction.
- A **Random Forest** classifier uses a number of decision trees, in order to improve the classification rate.
- **Boosted Trees** can be used for regression-type and classification-type problems.
- **Rotation forest** - in which every decision tree is trained by first applying principal component analysis (PCA) on a random subset of the input features.

Decision tree learning is the construction of a decision tree from class-labeled training tuples. A decision tree is a flow-chart-like structure, where each internal (non-leaf) node denotes a test on an attribute, each branch represents the outcome of a test, and each leaf (or terminal) node holds a class label. The topmost node in a tree is the root node.

There are many specific decision-tree algorithms. Notable ones include:

- ID3 (Iterative Dichotomiser 3)
- C4.5 (successor of ID3)
- CART (Classification And Regression Tree)
- CHAID (CHi-squared Automatic Interaction Detector). Performs multi-level splits when computing classification trees.
- MARS: extends decision trees to handle numerical data better.
- Conditional Inference Trees. Statistics-based approach that uses non-parametric tests as splitting criteria, corrected for multiple testing to avoid over fitting. This approach results in unbiased predictor selection and does not require pruning.

Amongst other data mining methods, decision trees have various advantages:

- **Simple to understand and interpret.** People are able to understand decision tree models after a brief explanation.
- **Requires little data preparation.** Other techniques often require data normalization, dummy variables need to be created and blank values to be removed.
- **Able to handle both numerical and categorical data.** Other techniques are usually specialized in analyzing datasets that have only one type of variable. (For example, relation rules can be used only with nominal variables while neural networks can be used only with numerical variables.)
- **Uses a white box model.** If a given situation is observable in a model the explanation for the condition is easily explained by Boolean logic. (An example of a black box model is an artificial neural network since the explanation for the results is difficult to understand.)
- **Possible to validate a model using statistical tests.** That makes it possible to account for the reliability of the model.
- **Robust.** Performs well even if its assumptions are somewhat violated by the true model from which the data were generated.
- **Performs well with large datasets.** Large amounts of data can be analyzed using standard computing resources in reasonable time.

B. Semi-supervised Learning Techniques

Semi-supervised learning is a class of supervised learning tasks and techniques that also make use of unlabeled data for training - typically a small amount of labeled data with a large amount of unlabeled data. Semi-supervised learning falls between unsupervised learning (without any labeled training data) and supervised learning (with completely labeled training data). Many machine-learning researchers have found that unlabeled data, when used in conjunction with a small amount of labeled data, can produce considerable improvement in learning accuracy. The acquisition of labeled data for a learning problem often requires a skilled human agent (e.g. to transcribe an audio segment) or a physical experiment (e.g. determining the 3D structure of a protein or determining whether there is oil at a particular location). The cost associated with the labeling process thus may render a fully labeled training set infeasible, whereas acquisition of unlabeled data is relatively inexpensive. In such situations, semi-supervised learning can be of great practical value. Semi-supervised learning is also of theoretical interest in machine learning and as a model for human learning.

In order to make any use of unlabeled data, we must assume some structure to the underlying distribution of data. Semi-supervised learning algorithms make use of at least one of the following assumptions.

1) Smoothness assumption

Points which are close to each other are more likely to share a label. This is also generally assumed in supervised learning and yields a preference for geometrically simple decision boundaries. In the case of semi-supervised learning, the smoothness assumption additionally yields a preference for decision boundaries in low-density regions, so that there are fewer points close to each other but in different classes.

2) Cluster assumption

The data tend to form discrete clusters, and points in the same cluster are more likely to share a label (although data sharing a label may be spread across multiple clusters). This is a special case of the smoothness assumption and gives rise to feature learning with clustering algorithms.

3) Manifold assumption

The data lie approximately on a manifold of much lower dimension than the input space. In this case we can attempt to learn the manifold using both the labeled and unlabeled data to avoid the curse of dimensionality. Then learning can proceed using distances and densities defined on the manifold.

The manifold assumption is practical when high-dimensional data are being generated by some process that may be hard to model directly, but which only has a few degrees of freedom. For instance, human voice is controlled by a few vocal folds, and images of various facial expressions are controlled by a few muscles. We would like in these cases to use distances and smoothness in the natural space of the generating problem, rather than in the space of all possible acoustic waves or images respectively.

Methods of semi-supervised learning

- Generative Models
- Low-Density Separation
- Graph-Based Methods
- Heuristics Approaches

C. Unsupervised Learning Techniques

In machine learning, the problem of **unsupervised learning** is that of trying to find hidden structure in unlabeled data. Since the examples given to the learner are unlabeled, there is no error or reward signal to evaluate a potential solution. This distinguishes unsupervised learning from supervised learning and reinforcement learning.

Unsupervised learning is closely related to the problem of density estimation in statistics. However unsupervised learning also encompasses many other techniques that seek to summarize and explain key features of the data. Many methods employed in unsupervised learning are based on data mining methods used to preprocess data.

Approaches to unsupervised learning include:

- Clustering (e.g., k-means, mixture models, hierarchical clustering),
- Hidden Markov Models,
- Blind signal separation using feature extraction techniques for dimensionality reduction, e.g.:
  - Principal Component Analysis,
  - Independent Component Analysis,
  - Non-Negative Matrix Factorization,
  - Singular Value Decomposition.

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

i. Clustering

Cluster analysis or **clustering** is the task of grouping a set of objects in such a way that objects in the same group (called a **cluster**) are more similar (in some sense or another) to each other than to those in other groups (clusters). It is a main task of exploratory data mining, and a common technique for statistical data analysis, used in many fields, including machine learning, pattern recognition, image analysis, information retrieval, and bioinformatics.

Cluster analysis itself is not one specific algorithm, but the general task to be solved. It can be achieved by various algorithms that differ significantly in their notion of what constitutes a cluster and how to efficiently find them. Popular notions of clusters include groups with small distances among the cluster members, dense areas of the data space, intervals or particular statistical distributions. Clustering can therefore be formulated as a multi-objective optimization problem. The appropriate clustering algorithm and parameter settings (including values such as the distance function to use, a density threshold or the number of expected clusters) depend on the individual data set and intended use of the results.

Cluster analysis as such is not an automatic task, but an iterative process of knowledge discovery or interactive multi-objective optimization that involves trial and failure. It will often be necessary to modify data preprocessing and model parameters until the result achieves the desired properties.

The subtle differences are often in the usage of the results: while in data mining, the resulting groups are the matter of interest, in automatic classification the resulting discriminative power is of interest. This often leads to misunderstandings between researchers coming from the fields of data mining and machine learning, since they use the same terms and often the same algorithms, but have different goals.

Typical cluster models include:

- **Connectivity models:** for example hierarchical clustering builds models based on distance connectivity.
- **Centroid models:** for example the k-means algorithm represents each cluster by a single mean vector.
- **Distribution models:** clusters are modeled using statistical distributions, such as multivariate normal distributions used by the Expectation-maximization algorithm.
- **Density models:** for example DBSCAN and OPTICS defines clusters as connected dense regions in the data space.
- **Subspace models:** in Biclustering (also known as **Co-clustering or two-mode-clustering**), clusters are modeled with both cluster members and relevant attributes.
- **Group models:** some algorithms do not provide a refined model for their results and just provide the grouping information.
- **Graph-based models:** a clique, i.e., a subset of nodes in a graph such that every two nodes in the subset are connected by an edge can be considered as a prototypical form of cluster. Relaxations of the complete connectivity requirement (a fraction of the edges can be missing) are known as quasi-cliques.

A "clustering" is essentially a set of such clusters, usually containing all objects in the data set. Additionally, it may specify the relationship of the clusters to each other, for example a hierarchy of clusters embedded in each other. Clusterings can be roughly distinguished as:

- **Hard Clustering:** each object belongs to a cluster or not
- **Soft Clustering** (also: fuzzy clustering): each object belongs to each cluster to a certain degree (e.g. a likelihood of belonging to the cluster)

There are also finer distinctions possible, for example:

- **Strict Partitioning Clustering:** here each object belongs to exactly one cluster
- **Strict Partitioning Clustering with outliers:** objects can also belong to no cluster, and are considered outliers.
- **Overlapping Clustering** (also: alternative clustering, multi-view clustering): while usually a hard clustering, objects may belong to more than one cluster.
- **Hierarchical Clustering:** objects that belong to a child cluster also belong to the parent cluster
- **Subspace clustering:** while an overlapping clustering, within a uniquely defined subspace, clusters are not expected to overlap.

Clustering Algorithms

Clustering algorithms can be categorized based on their cluster model, as listed above. The following overview will only list the most prominent examples of clustering algorithms, as there are possibly over 100 published clustering algorithms. Not all provide models for their clusters and can thus not easily be categorized. An overview of algorithms explained in Wikipedia can be found in the list of statistics algorithms.

There is no objectively "correct" clustering algorithm, but as it was noted, "clustering is in the eye of the beholder. The most appropriate clustering algorithm for a particular problem often needs to be chosen experimentally, unless there is a mathematical reason to prefer one cluster model over another. It should be noted that an algorithm that is designed for one kind of model has no chance on a data set that contains a radically different kind of model. For example, k-means cannot find non-convex clusters.

TABLE 1
DIFFERENT TYPES OF CLUSTERING ALGORITHMS

S.No	Clustering Algorithms	
	Algorithm	***Description***
1	Connectivity Based Clustering	Connectivity based clustering, also known as *hierarchical clustering*, is based on the core idea of objects being more related to nearby objects than to objects farther away. These algorithms connect "objects" to form "clusters" based on their distance.
2	Centroid Based Clustering	In centroid-based clustering, clusters are represented by a central vector, which may not necessarily be a member of the data set. When the number of clusters is fixed to k, *k*-means clustering gives a formal definition as an optimization problem: find the k cluster centers and assign the objects to the nearest cluster center, such that the squared distances from the cluster are minimized.
3	Distribution Based Clustering	The clustering model most closely related to statistics is based on distribution models. Clusters can then easily be defined as objects belonging most likely to the same distribution. A convenient property of this approach is that this closely resembles the way artificial data sets are generated: by sampling random objects from a distribution.
4	Density Based Clustering	In density-based clustering, clusters are defined as areas of higher density than the remainder of the data set. Objects in these sparse areas - that are required to separate clusters - are usually considered to be noise and border points.

ii. Hidden Markov Model

A **hidden Markov model** (**HMM**) is a statistical Markov model in which the system being modeled is assumed to be a Markov process with unobserved (*hidden*) states. An HMM can be presented as the simplest dynamic Bayesian network.

In simpler Markov models (like a Markov chain), the state is directly visible to the observer, and therefore the state transition probabilities are the only parameters. In a *hidden* Markov model, the state is not directly visible, but output, dependent on the state, is visible. Each state has a probability distribution over the possible output tokens. Therefore the sequence of tokens generated by an HMM gives some information about the sequence of states. Note that the adjective 'hidden' refers to the state sequence through which the model passes, not to the parameters of the model; the model is still referred to as a 'hidden' Markov model even if these parameters are known exactly.

Hidden Markov models are especially known for their application in temporal pattern recognition such as speech, handwriting, gesture, part-of-speech tagging, musical score following, partial discharges and bioinformatics.

A hidden Markov model can be considered a generalization of a mixture model where the hidden variables (or latent variables), which control the mixture component to be selected for each observation, are related through a Markov process rather than independent of each other. Recently, hidden Markov models have been generalized to pairwise Markov models and triplet Markov models which allow consideration of more complex data structures and the modeling of non-stationary data.

III. Comparison of Machine Learning Techniques

In supervised learning, the output datasets are provided which are used to train the machine and get the desired outputs whereas in unsupervised learning no datasets are provided, instead the data is clustered into different classes.

Example: Face recognition

1. **Supervised learning**: Learn by examples as to what a face is in terms of structure, color, etc so that after several iterations it learns to define a face. Supervised learning is when the data you feed your algorithm is "tagged" to help your logic make decisions. Applications in which the training data comprises examples of the input vectors along with their corresponding target vectors are known as supervised learning problems. The data as presented to a machine learning algorithm is fully labeled. That means: all examples are presented with a classification that the machine is meant to reproduce. For this, a classifier is learned from the data, the process of assigning labels to yet unseen instances is called classification.
2. In **semi-supervised** systems, the machine is allowed to additionally take unlabeled data into account. Due to a larger data basis, semi-supervised systems often outperform their supervised counterparts using the same labeled examples. The reason for this improvement is that more unlabeled data enables the system to model the inherent structure of the data more accurately. Bootstrapping, also called self-training, is a form of learning that is designed to use even less training examples, therefore sometimes called **weakly-supervised**. Bootstrapping starts with a few training examples, trains a classifier, and uses thought-to-be positive examples as yielded by this classifier for retraining. As the set of training examples grows, the classifier improves, provided that not too many negative examples are misclassified as positive, which could lead to deterioration of performance.
3. **Unsupervised learning**: since there is no desired output in this case that is provided therefore categorization is done so that the algorithm differentiates correctly between the face of a horse, cat or human (clustering of data). Unsupervised learning are types of algorithms that try to find correlations without any external inputs other than the raw data. In other pattern recognition problems, the training data consists of a set of input vectors x without any corresponding target values. The goal in such unsupervised learning problems may be to discover groups of similar examples within the data, where it is called clustering. **Unsupervised** systems are not provided any training examples at all and conduct clustering. This is the division of data instances into several groups. The results of clustering algorithms are data driven, hence more 'natural' and better suited to the underlying structure of the data. This advantage is also its major drawback: without a possibility to tell the machine what to do (like in classification), it is difficult to judge the quality of clustering results in a conclusive way. But the absence of training example preparation makes the unsupervised paradigm very appealing.

Issues in Machine Learning Techniques:

- Bias-variance trade-off
- Function complexity and amount of training data
- Dimensionality of input space
- Noise in the output values
- Heterogeneity of Data
- Redundancy in data
- Presence of non-linearity's

IV. Conclusion

This paper gives an idea about the machine learning techniques involved in learning scheme. The analysis or learning of data is an important factor in annotating any kind of data. The Supervised Learning Techniques (SLT) will lead to minimize error with respect to the given inputs. The Semi-Supervised Learning Techniques far better than SLT as it inherits the structure of learning scheme accurately for any kind of data with less training examples. The Unsupervised Learning Techniques results show that they are natural and better suited to the underlying structure of the data.

References

[1] Tapas Ranjan Baitharu, Subhendu Kumar Pani, "A Survey on Application of Machine Learning Algorithms on Data Mining", International Journal of Innovative Technology and Exploring Engineering, ISSN: 2278-3075, Volume-3, Issue-7, December 2013.

[2] Provost, F., & Fawcett, T., Robust Classification for Imprecise Environments. Machine Learning, Vol. 42, No.3, pp.203-231, 2001.

[3] Robert L Grossman, Michael Sabala, Yunhong Gu, Anushka Anand, Matt Handley, Rajmonda Sulo, & Lee Wilkinson. Distributed discovery in e-science: Lessons from the angle project. In Next Generation Data Mining (NGDM '07), 2008.

[4] Tang, H. (2008). A simple approach of data mining in Excel. *4th International Conference on WirelessCommunication, Networking and Mobile Computing*, 1-4.

[5] Jayaveer Singh and Manisha J.Nene, "A Survey on Machine Learning Techniques for Intrusion Detection System", *International Journal of Advanced Research in Computer and Communication Engineering*, Vol. 2, Issue 11, November 2013.

[6] J. Burges, "A tutorial on support vector machines for pattern recognition" Data Mining and Knowledge Discovery, vol. 2, pp. 12 1-167,1998.

[7] S. B. Kotsiantis, "Supervised Machine Learning: A Review of Classification Techniques", Informatica 31 (2007) 249–268.

[8] Barto, A. G. & Sutton, R. (1997). *Introduction to Reinforcement Learning.* MIT Press.

[9] Breslow, L. A. & Aha, D. W. (1997). "Simplifying decision trees: A survey". *Knowledge Engineering Review 12:* 1–40.

[10] Bkassiny, M., Yang Li ; Jayaweera, S.K., "A survey on Machine Learning Techniques in Cognitive Radios", ISSN:1553-877X, pp. 1136-1159, 22 October 2012.

[11] Aziz Nasridinov, Young-Ho Park, "A Survey on Machine Learning Techniques in Wireless Sensor Netowrks", ICCA 2013, Vol.30.

[12] Wernick, Yang, Brankov, Yourganov and Strothar, "Machine Learning in Medical Imaging", IEEE Signal Processing Magazine, Vol.27, No.4, July 2010, pp.25-38.

[13] Mehryar Mohri, Afshin Rostamizadeh, Ameet Talwalkar (2012) "Foundations of Mahcine Learning", MIT Press ISBN 9780262018258.

[14] C.M.Bishop (2006). "Pattern Recognition and Machine Learning", Springer ISBN 0-387-31073-8.

[15] Ian H. Witten and Eibe Frank (2011). "Data Mining: Practical Machine Learning Tools and Techniques" Morgan Kaufmann, 664 pp., ISBN 978-0123748560.

[16] Huang T. – M., Kecman V., Kopriva I. (2006), "Kernel Based Algorithms for Mining huge Data Sets, Supervised, Semi-Supervised, and Unsupervised Learning", Springer – Verlag, Berlin, Heidelberg, 260 pp. 96 illus., Hardcover, ISBN 3-540-31681-7.

Mrs.P. Anitha received her B.E. degree in Computer Science and Engineering from Avinashilingam University in 2006. She also received M.Tech degree from Anna University of Technology, Coimbatore in 2010. She is now pursuing Ph.D in the field of Network Security and Data Mining. She is now working as Assistant Professor in Dr.N.G.P.Institute of Technology, Coimbatore, India. She has published papers in National and International Conferences. She is the lifetime member of ISTE.

Mrs.G.Krithika has received her Diploma in Computer Technology from PSG Polytechnic College in 2003. She received her B.E. degree in Computer Science and Engineering from Sri Ramakrishna Institute of Technology in 2006. She is currently pursuing M.E. Computer Science Engineering from Dr.N.G.P.Institute of Technology. She is currently working as Lecturer in Dr.N.G.P.Institute of Technology, Coimbatore, India. She is the lifetime member of ISTE.

Mani Deepak Choudhry has received his B.Tech degree in Information Technology from SNS College of Technology in 2010. He also received M.E degree in Software Engineering from Anna University, Regional centre-Coimbatore in 2014. He is currently working as Assistant Professor in the Department of Information Technology, Dr.N.G.P Institute of Technology, Coimbatore, India. He is the lifetime member of ISTE.