International Journal of Soft Computing and Engineering (IJSCE)
ISSN: 2231-2307, Volume-2, Issue-4, September 2012

A Comparative Study of Training Algorithms for Supervised Machine Learning

Hetal Bhavsar, Amit Ganatra

Abstract – Classification in data mining has gained a lot of importance in literature and it has a great deal of application areas from medicine to astronomy, from banking to text classification.. It can be described as supervised learning algorithm as it assigns class labels to data objects based on the relationship between the data items with a pre-defined class label. The classification techniques are help to learn a model from a set of training data and to classify a test data well into one of the classes. This research is related to the study of the existing classification algorithm and their comparative in terms of speed, accuracy, scalability and other issues which in turn would help other researchers in studying the existing algorithms as well as developing innovative algorithms for applications or requirements which are not available.

Keywords - classification, decision tree, nearest neighbour, neural network, SVM, Supervised learning

I. INTRODUCTION

The tremendous amount of information stored in databases cannot simply be used for further processing. Data mining involves the use of sophisticated data analysis tools to discover previously unknown, valid patterns and relationships in large data set. These tools can include statistical models, mathematical algorithm and machine learning methods. Consequently, data mining consists of more than collection and managing data, it also includes analysis and prediction [18].

Classification and prediction are two forms of data analysis that can be used to extract models describing important data classes or to predict future data trends. Such analysis can help us to provide with a better understanding of the large data. Classification predicts categorical (discrete, unordered) labels, while prediction models continuous valued functions. Classification technique is capable of processing a wider variety of data than regression and is growing in popularity.

Classification is also called supervised learning, as the instances are given with known labels, contrast to unsupervised learning in which labels are not known. Each instance in the dataset used by supervised or unsupervised learning method is represented by set of features or attributes which may be categorical or continuous [9] [17].

Classification is the process of building the model from the training set made up of database instances and associated class label. The resulting model is then used to predict the class label of the testing instances where the values of the predictor features are known. Supervised classification is one of the tasks most frequently carried out by intelligent techniques. The large number of techniques have been developed.

Manuscript received on September, 2012
Hetal Bhavsar, Information Technology Dept., SVIT, Vasad, Gujarat, India.
Amit Ganatra Computer Engineering Dept., CHARUSAT, Changa, Gujarat, India.

This work concentrated on the comparative study of some very well known classification algorithms like Decision Tree Induction, Baysian Network, Neural Network, K-nearest neighbours and Support Vector Machine. A comparative study would definitely bring out the advantages and disadvantages of one method over the other. This would provide the guideline for interesting research issues which in turn help other researchers in developing innovative algorithms for applications or requirements which are not available.

This paper is organized as follow: Section 2 covers decision tree induction, section 3 describes Bayesian networks, while neural network with backprapogation is discussed in section 4, section 5 covers k-nearest neighbour algorithm whereas support vector machine is describes in section 6. Finally section 7 covers the comparative analysis of these algorithm followed by conclusion.

II. DECISION TREE INDUCTION

Decision tree classifies data into discrete ones using tree structure algorithms[11]. The main purpose of decision trees are to expose the structural information contained in the data. The decision tree method is a supervised machine learning technique that builds a decision tree from a set of class labelled training samples during the machine learning process[9].

The algorithm of Decision trees start with the training samples and their associated class labels. This training set is recursively partitioned base on feature value into subset so that the data in each of the subset is purer that the data in the parent set. Each internal node in a decision tree represent a test on attribute (feature) , each branch represent an outcome of the test and each leaf node represents the class label. As a classifier decision tree is used to identify the class label of an unknown sample, tracing path from root to the leaf node, which holds the class label for that sample[9][17].

The root node of the tree is the feature that best divides the training data. There are several measures for finding the feature that best divides the training data, like Information gain , Gain ratio, Gini index , myopic measures estimate each attribute independently, ReliefF algorithm, Chi square, C-SEP, G-statisticcs, Minimum Description Length (MDL) measure which is least bias toward multivalued attribute, Multivariate split – based on combination of attributes [9][11][17][18].

No one measure is significantly superior than others [9].Decision tree complexity increase with tree height. Therefore, measures that tends to produce tree with multiway and that favour more balanced splits may be preferred, may depend on the dataset.

The basic decision tree induction algorithm adopt non backtracking, greedy, top-down and recursive divide and conquer strategies.

The algorithm is summarized as follow:

1. Create the root node N.
2. If all the samples belongs to same class C, then return the node N as leaf node with the class labelled C.
3. If no feature is there then return N as leaf node with the most common class in samples.
4. Apply the feature selection measure, to select the best feature.
5. Label node N with the feature found in step 4, called test feature
6. For each value v_i of test feature
7. Partition the samples and grow subtree for each value v_i of test feature
8. Let a_i be the set of tuples for which test feature = v_i
9. If a_i is empty then attach a leaf node with the most common class in samples.
10. Else attach the node returned by Generate_decision_tree (a_i, attribute_list – test_attribute).

Tree pruning is used to improve the prediction and classification accuracy of the algorithm by minimizing over-fitting. Over-fitting in decision tree algorithm results in misclassification error. Tree pruning is done in bottom-up manner and is less tasking compared to the tree growth phase as the training data set is scanned only once.

There are two approaches to tree pruning: 1) Prepruning, which prune the tree by halting its construction early based on the value of prespicified threshold and2) postpruning, which remove the subtree from the fully grown tree. Though, Post pruning required more computation effort but it leads to a more reliable trees. Repetition and replication problem can occurred in pruned trees, which can be solved with multivariant split based on a combination of attribute[9].

Some of the well known decision tree algorithm are ID3, C4.5 and CART.

ID3 algorithm is an expansion of concept learning theory by E. B. Hunt, J. Martin and P.T. Stone[10]. It is a recursive procedure using divide and conquer approach, which supports only nominal attributes. Information gain is used to select a attribute to split. It does not give accurate result when there is too-much noise or details in the training data set, thus a an intensive pre-processing of data is carried out before building a decision tree model with ID3[9].

C4.5 is developed by [12], uses gain ratio for selection of attribute for splitting. It provides an improvement over ID3 as it deals with nominal and numerical attributes as well as able to handle missing and noisy data. Pruning in C4.5 takes place by replacing the internal node with a leaf node thereby reducing the error rate. Classifier generated by C4.5 can be expressed not only in terms of decision tree but also in more comprehensible rule set form. The major disadvantage of rule set form is that it require large amount of CPU time and memory.

CART (Classification and Regression Trees) proposed by Breiman [8], uses Gini index measure for selecting attribute for splitting. Test in CART is always binary. CART prunes trees using a cost-complexity model whose parameters are estimated by cross-validation [20].

The advantages, disadvantages and research issues of Decision Tree Induction are as follows [9][14][17][20].

Advantages:

- Decision Trees are very simple and fast.
- It does not require any domain knowledge or parameter setting and it is able to handle high dimensional data.
- Representation is easy to understand i.e. comprehensible.
- Have good accuracy (may depend on data at hand).
- It Support incremental learning.
- Decision trees are unvaried since they use based on a single feature at each internal node

Disadvantages:

- It has long training time, as it requires one pass over the training tupels in D for each level of tree.
- Lack of available memory, when dealing with large databases.
- The division of the instance space is orthogonal to the axis of one variable and parallel to all other axes. The resulting regions after partitioning are all hyper rectangles.
- Most decision tree algorithms cannot perform well with problems that require diagonal partitioning.
- Decision trees can be significantly more complex representation for some concepts due to the replication problem.
- Orders of attributes in tree nodes have adverse effect on performance.

Research Issues:

- Can a complex decision tree be broken down to a small collection of simple trees that, when voted to gather, give the same result as the complex tree?
- Can we develop a non-trivial tree-construction algorithm that would hardly affected by omitting a single case?

III. BAYESIAN NETWORK

Bayesian Classifiers are statistical classifiers. They predict the class membership probability, that is the probability that a given sample belongs to a particular class. Bayesian belief networks are graphical models, showing the relationship between the subset of attributes. Bayesian classifier have exhibited high accuracy and speed when applied to large databases [7] [13].

Naïve Bayes Classifier is the simple statistical Bayesian Classifier [5]. It is called Naïve as it assume that all variables contribute toward classification and are mutually correlated. This assumption is called class conditional independence [7]. This is an unrealistic assumption for most datasets, however it leads to a simple prediction framework that gives surprisingly good result in many practical cases. The Naïve Bayes Classifier is based on Bayes' Theoram.

The Bayes'Theoram is

$$P(H|X) = \frac{P(X|H)P(H)}{P(X)} \quad (1)$$

Where,

H- some hypothesis , such that data tuple X belongs to specified class C

X – some evidence, describe by measure on set of attributes

$P(H|X)$ – the posterior probability that the hypothesis H holds given the evidence X

P(H) – prior probability of H, independent on X

$P(X|H)$ – the posterior probability that of X conditioned on H.

The algorithm calculates the following two probabilities and compares them.

$$R = \frac{P(i|X)}{P(j|X)} = \frac{P(i)P(X|i)}{P(j)P(X|j)} = \frac{P(i)nP(X|i)}{P(j)nP(X|j)} \quad (2)$$

Comparing theses probabilities, predicted class label is the class of higher probability. (i.e. if R>1; Predict i, else predict j). As the naïve bayes classifier uses the product operation to estimate probability p(X/i), what hanppens if probability value for some $P(X_k/i)$ is zero? A zero probability cancels the effects of all of the other probabilities involved in the product. This can be avoided by Laplace correlation or Laplace estimator by adding one to all numerator and adding the number of added ones to the denominator [9].

The advantages, disadvantages and research issues of Naïve Bayesian are as follows [9][14][17][20].

Advantages

- It requires short computational time for training and very easy to construct.
- Model has a form of a product, which can be easily converted into a sum through the use of logarithms – with significant consequent computational advantages.
- Not needing any complicated iterative parameter estimation schemes, so can be applied to large data set.
- Easy interpretation of knowledge representation
- May not be best classifier in any particular application, but it does well and robust.

Disadvantages:

- Theoratically, naïve bayes classifier have minimum error rate comparing to other classifier, but practically it is not always true, because of assumption of class conditional independence and the lack of available probability data.
- Less accurate compare to other classifier.

Research Issues:

- How to add extra edges to include some of the dependencies between the attributes to overcome attribute independence assumption?

IV. BACKPROPAGATION

The perceptron is a simple neural network, proposed in 1962 by Rosenblatt[7]. Neural network is a set of connected input/output units in which each connection has a weight associated with it. During the learning phase, to predict the class label of the input sample, the network learns by adjusting the weight.

Neural network with backpropagation was proposed by [4]. Back propagation algorithm performs learning on a multilayer feed-forward neural network. A multilayer fee-forward network consists of large number of units (neurons) joined to gather in a pattern of connections. These units are : an input layer, one or more hidden layer and an output layer. The input layer, receives the information to be processed, the output layer, shows the result of processing and hidden layer, allow the signals to travel one way only, from input to output[14][17].

The network is trained on as set of paired data to determine input-output mapping. The weight of the connections between neurons are then fixed and the network is used to determine the classification of a new set of data[20].

Back propagation algorithm is most well known and widely used algorithm. They learn by iteratively processing a data set of training samples, comparing the network prediction for each sample with the actual target value. To minimize the mean square error between the network prediction and actual target value the weights are modified. These modifications are done in backward direction from output layer, through each hidden layer down to first hidden layer, and hence the name 'back propagation" [9].

The algorithm is summarizing as follow:

1. Initialize all the weights and bias.
2. Feed the training sample into neural network.
3. Actual target value of the training sample is compared with the network's output. Calculate the error in each output neuron.
4. For each neuron, calculate what the output should have been, and a scaling factor, how much lower or higher the output must be adjusted to match the desired output. This is the local error.
5. The error is propagated backward by updating the weights and biases to reflect the error of the network's prediction.
6. Assign "blame" for the local error to neurons at the previous level, giving greater responsibility to neurons connected by stronger weights.
7. Repeat the steps above on the neurons at the previous level, using each one's "blame" as its error. With more details, the general rule for updating weights is:

   $\Delta W_{ji} = \eta \delta_j O_i$ where:

   - η is a positive number (called learning rate), which determines the step size in the gradient descent search. A large value enables back propagation to move faster to the target weight configuration but it also increases the chance of its never reaching this target.
   - O_i is the output computed by neuron *i*
   - $\delta_j = O_j(1 - O_j)(Tj - O_j)$ for the output neurons, where T_j the wanted output for the neuron *j* and
   - $\delta_j = O_j(1 - O_j)\sum \delta_k W_{kj}$ the internal (hidden) neurons

Each iteration through the training set for weight modification is called epoch. The back propagation algorithm will have to perform a number of epoch for weight modifications before it reaches a good weight configuration. For this reason, a number of different stopping rules are used by neural network to decide when training ends. The four most common stopping rules are:

i) Stop after a specified number of epochs, ii) Stop when an error measure reaches a threshold, iii) Stop when the error measure has seen no improvement over a certain number of

epochs, iv) Stop when the error measure on some of the data that has been sampled from the training data (hold-out set, validation set) is more than a certain amount than the error measure on the training set (overfitting).

Complexity:

Given |D| samples and w weights, each ehoch requires O(|D|*w) time. In worst case, the number of epochs can be exponential in n, the number of inputs.

The advantages, disadvantages and research issues of neural network with backpropagation are as follows [9][14][17][20].

Advantages:

- Neural networks are able to tolerate noisy data as well as able to classify patterns on which they are not been trained.
- They can be used when we have the little knowledge of the relationship between attributes and classes.
- Well suited for continuous valued inputs and outputs.
- Inherently parallel, so parallelization techniques can be used to speed up the computational process.
- Successful on several real world application like handwritten character recognition, pathology and laboratory medicine, and many more.

Disadvantage:

- Involves long learning time, therefore more suitable for application where this is feasible.
- Poor interpretability as knowledge is represented in a form of a network of units connected by weighted links.
- Require number of parameters that are to be determined empirically, e.g. network topology or structure, number of hidden layers, number of units in each hidden layer and in output layer.

Research Issues:

- Extracting the knowledge embedded in trained neural networks and representing that knowledge symbolically is the challenging issue of neural network.

V. K- NEAREST NEIGHBOR CLASSIFICATION

K-nearest neighbor is non-parametric, instance based leaning method. Instance based classifiers are also called lazy learners as they store all of the training samples and do not build a classifier until a new, unlabeled sample needs to be classified. Lazy-learning algorithms require less computation time during the training phase than eager-learning algorithms (such as decision trees, neural networks and Bayes networks) but more computation time during the classification process[9][14][17].

The *k-nearest neighbors' algorithm* is amongst the simplest of all machine learning algorithms. It is based on the principal that the samples that are similar are lies in close proximity [3]. Given an unlabeled sample, K-nearest neighbor classifier searches the pattern space for the k-objects that are closest to it and assigned the class by identifying the most frequent class label. If the value of k=1 then assign the class of the training sample that is the closest to the unknown sample in the pattern space.

The instance based learning algorithm is summarized as follow:

Procedure Instance Base Learner (TestingInstances) for each testing instance

{

1. Given a testing instance find the k most nearest neighbour from the training set according to a distance metric
2. The most frequent class label of the k nearest neighbour is the class label of the testing instance.

}

Generally, n dimensional attributes are used to represent training samples. Each training sample is represented by a point in an n-dimensional space. Main elements of this process are: 1) set of stored samples, 2) similarity or distance measure to compute distance between two samples and 3) value of k, the number of nearest neighbors.There are several distance measures presented as shown in the table 1. Ideally, the measure should be choosen in such a way that minimize the distance between the samples that are similar and maximize the distance between the samples that are dissimilar.

The measures listed in Table I assumes numerical attributes only. If the attributes are categorical then difference between two values is taken to be 1 if they are different, and value is taken to be 0 if they are identical.

The choice of k also affects the performance of k-nearest neighbour algorithm. If value of k is small, and noise is present in the pattern space, then noisy samples may win the majority votes, which results into misclassification error. This can be solved with larger value of k. If value of k is large, and if the portion of the class is small, then instances of other class may win the majority votes, results into misclassification error. A smaller value of k can solve this problem.

Table I: Distance Measures

Minkowsky: $D(x,y)=\left(\sum_{i=1}^{m} \mid x_i - y_i \mid^r\right)^{1/r}$
Manhattan: $D(x,y)=\sum \mid x_i - y_i \mid$
Chebychev: $D(x,y)=\max_{i=1}^{m} \mid x_i - y_i \mid$
Euclidean: $D(x,y)=\left(\sum_{i=1}^{m} \mid x_i - y_i \mid^2\right)^{1/2}$
Camberra: $D(x,y)=\sum_{i=1}^{m} \frac{\mid x_i - y_i \mid}{\mid x_i - y_i \mid}$
Kendall's Rank Correlation: $D(x,y)=1-\frac{2}{m(m-1)}\sum_{i=j}^{m}\sum^{i-1} sign(x_i - y_i)sign(y_i - y_j)$

Complexity:

Nearest neighbor classifier are very slow in classifying a new sample. Given a training data set of |D| tuples and k=1, then O(|D|) comparisons are required in order to classify a given test tuple. This can be further reduce by storing the training samples in search trees, which reduce the number of comparison to O(log|D|). Parallel implementation can reduce the running time to a constant, that is O(1), which is independent of |D|.

The advantages, disadvantages and research issues of KNN are as follows [9][14][17][20].

Advantages:

- Easy to understand and easy to implement classification technique.
- An expected lazy learning methods are faster at a training than eager methods.
- Perform well on application in which a sample can have many class labels.

Disadvantages:

- Lazy learners incur expensive computational costs when the number of potential neighbors which to compare a given unlabeled sample is large.
- Slower at classification since all computation is delayed to that time.
- Nearest neighbor classifiers assign equal weight to each attribute. This may cause confusion when there are many irrelevant attributes in the data and results into poor accuracy.
  Solution: Assign weight to the attributes and prune noisy data samples.
- Sensitive to the local structure of the data.
- They have large storage requirements.
- They are sensitive to the choice of the similarity function that is used to compare instances.
- They lack a principled way to choose *k*, except through cross-validation or similar, computationally-expensive technique.

Research Issues:

- Retaining the classification accuracy of the *k*NN classifier by eliminating many of the stored data objects. This is known as 'condensing' and can greatly speed up the classification of new objects.
- Large amount of work on the application of proximity graphs to the KNN problem.

VI. SUPPORT VECTOR MACHINE

SVM have attracted a great deal of attention in the last decade and actively applied to various domains applications. SVMs are typically used for learning classification, regression or ranking function. SVM are based on statistical learning theory and structural risk minimization principal and have the aim of determining the location of decision boundaries also known as hyperplane that produce the optimal separation of classes [2][9][19].

Support Vector Classification (SVC) is the algorithm that revolve around the notion of a "margin"—either side of a hyperplane that separates two data classes. Maximizing the margin and thereby creating the largest possible distance between the separating hyperplane and the instances on either side of it has been proven to reduce an upper bound on the expected generalisation error.

SVM is outlined first for the linearly separable case. It then uses kernel functions for nonlinear mapping to transform the original training data into a higher dimension, within which it searches for linear optimal separating hyperplane. Finally, slack variables are introduced for noisy data to allow training errors [1].

A. Linearly Separable data:

If training data is linearly separable, then a pair (w,b) exists such that

$$\mathbf{w}\cdot\mathbf{x}_i+b >= 1 \text{ for } y_i = 1, \text{and} \quad (3)$$

$$\mathbf{w}\cdot\mathbf{x}_i+b <= -1 \text{ for } y_i = -1 \quad (4)$$

With the decision rule given by

$$f(X) = \text{sgn}(w.x + b) \quad (5)$$

Where w is the weight vector and b is the bias (or –b is termed as threshold).

SVM searches for the optimal separating hyperplane that correctly classifies the data as shown in fig. 1. This is equivalent to maximizing the distance, normal to the hyperplane, between the convex hull of two classes and this distance is called the margin.

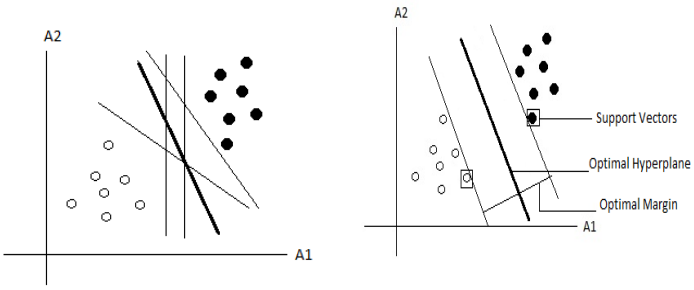

Figure 1. Optimal Separating Hyperplane

Hence, the hyperplane that optimally separates the data is the one that minimizes

$$\Phi(w)=\frac{1}{2}\|w\|^2 \quad (6)$$

Subject to $y_i(\mathbf{w}\cdot\mathbf{x}_i+b) >= 1, \forall(\mathbf{x}_i,y_i) \in D$.

This optimisation is known as a convex quadratic programming (QP) problem.

Any training tuples that fall on margin are called support vectors. The support vectors are the most difficult tuples to classify and give the most information regarding classification. Other data points are ignored. Since the complexity of SVM is depends only on support vectors, which are very less in numbers, they are well suited for the data sets where the number of features is large compared to number of training instances[2][9][16][19].

A general pseudo-code for SVMs is illustrated as follows:

1. Introduce the Langrangian Multiplier to (1) for each of inequlity constraints, so that it can also proceed to non-separable and non-linear cases. The lagrangian for this problem is

$$\Phi(w, b, \alpha) = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{l} \alpha_i(y^i[(w,x_i)+ b]-1) \quad (7)$$

2. Minimize (7) with respect to w and b and maximize with respect α ≥0. This is a convex quadratic programming problem.
3. In the solution, those points for which $\alpha_i > 0$ and $y_i(\mathbf{w}\cdot\mathbf{x}_i+b) = 1$ are called "support vectors".

B. Linear SVM classifier: Nonseparable

If the two classes are not linearly separable, the SVM tries to find the hyperplane that maximises the margin while, at the same time, minimising a quantity proportional to the number of misclassification errors using the concept of soft margin[2][19]. The trade-off between margin and misclassification error is controlled by a user-defined constant called slack variables ξ_i, i=1,2,….N in the constraints, which then become:

$$y_i(\mathbf{w}\cdot\mathbf{x}_i+b) >= 1-\xi_i, \text{ i=1,2,..l} \quad (8)$$

where $\xi_i >=0$

Thus, the value of ξ_i must exceed unity for an error to occur. The $\sum_i \xi i$ is an upper bound on the number of training errors. The langrangian for this case is

$$\Phi(w, b, \alpha,\xi,\beta) = \frac{1}{2}||w||^2 + C \sum_i \xi i - \sum_{i=1}^{l} \alpha_i(y^i[(w,x_i)+ b]-1+\xi i) - \sum_{j=1}^{l} \beta i \xi i \quad (9)$$

Where α and β are the Lagrange multipliers. The Lagrangian has to be minimised with respect to w, b, x and maximised with respect to α and β.

C. Nonlinear SVM Classifier (Kernel SVM)

SVM can also be extended to learn non-linear decision functions by first projecting the input data onto a high-dimensional feature space using kernel functions and formulating a linear classification problem in that feature space[1][2].

The mapping of data to some other (possibly infinite space) Hilbert space H is denoted by,

$$\Phi : R^d \rightarrow H$$

The decision functions of equ. 5 become

$$f(x) = sgn(w.\Phi(x)+b) \quad (10)$$

Mapping the data to H is time consuming and storing it may be impossible, e.g. if H is infinite dimensional. Since the data only appear in inner products we require a computable function that gives the value of the inner product in H without explicitly performing the mapping[16]. Hence, introduce a kernel function,

$$k(x_i, x_j) = \Phi(x_i) \cdot \Phi(x_j) \quad (11)$$

The kernel function allows us to construct an optimal separating hyperplane in the space H without explicitly performing calculation in this space.

One of the advantages of support vector machine is improvement of generalization performance by appropriate selection of kernels, so selection of kernels to specific application is more significant. The common kernels that are used in SVM are given below[15].

- Linear Kernel: $k(x_i, x_j) = x_i \cdot x_j$
- Polynomial Kernels: $k(x_i, x_j)=(\gamma(x_i, x_j) + r)^d$, $r \geq 0$, $\gamma>0$
- Radial Basis Function Kernels (RBF): $k(x_i, x_j)=exp(-||x_i, x_j||^2 / 2\sigma^2)$ where $\sigma > 0$
- Hyperbolic Tangent Kernel: $k(x_i, x_j) =\tanh(\alpha (x_i, x_j) + r)$, $r \geq 0$

The advantages, disadvantages and research issues of SVM are as follows [9][14][17][20].

Advantages:

- One of the most robust and accurate methods among all well-known algorithms.
- It has a sound theoretical foundation, requires only a dozen examples for training, insensitive to the number of dimensions.
- Find the best classification function to distinguish between members of the two classes in the training data
- SVM is less prone to overfitting than other methods.

Disadvantages:

- It is computationally expensive, as solving QP methods require large matrix operations as well as time consuming numerical computations.
- SVMs are extremely slow in learning, requiring large amount of training time.
- The memory requirement grows with the square of the number of training examples.
- Poor interpretability of results

Research Issues:

- The underlying model implemented in SVMs is determined by the choice of the kernel. Deciding which kernel is the most suitable for a given application is obviously an important (and open) issue.
- The statistical learning theory developed by Vapnik and Chervonenkis provides necessary and sufficient conditions in terms of the Vapnik–Chervonenkis (VC) dimension (a capacity measure for functions). However, the estimation of the VC dimension for SVMs is often not possible and the relationship between both approaches is still an open issue.
- From a statistical point of view an important subject remains open: the interpretability of the SVM outputs.
- Regarding the finite sample performance of SVMs, where bias and variability computations for linear inversion algorithms (a particular case of regularization methods) are studied. The way to extend these ideas to the SVM nonlinear case is an interesting open problem.
- Expansion to very large database includes a large proportion of the training data, which leads to a model that is expensive both to store and to evaluate. Alleviating this problem is one area of ongoing research in SVMs.

VII. COMPARATIVE ANALYSIS

Supervised classification is one of the tasks most frequently carried out by intelligent techniques. The large number of techniques have been developed, some of which have been discussed in the previous sections. The table II shows the comparative studies of some commonly used classification techniques from the existing evidence and theoretical studies [9] [3] [4]. This comparison shows that not a single learning algorihtm outperform other algorithm all over the other datasets.

Table II: Comparative study of commonly used Classification Techniques

	Decision Trees	Neural Network	Naïve Bayes	K-Nearest Neighbor	Support Vector Machine
Proposed By	Quinlan	Rosenblatt	Duda and Hurt	Cover and Hart	Vapnik
Accuracy in general	Good	V. Good	Average	Good	Excellent
Speed of learning	V. Good	Average	Excellent	Excellent	Average
Speed of classification	Excellent	Excellent	Excellent	Average	Excellent
Tolerance to missing values	V. Good	Average	Excellent	Average	Good
Tolerance to irrelevant attributes	V. Good	Average	Good	Good	Excellent
Tolerance to redundant attributes	Good	Good	Average	Good	V. Good
Tolerance to highly interdependent attributes	Good	V. Good	Average	Average	V. Good
Dealing with discrete/ binary/continuous attributes	All	Not discrete	Not continuous	All	Not discrete
Tolerance to noise	Good	Good	V. Good	Average	Good
Dealing with danger of overfitting	Good	Average	V. Good	V. Good	Good
Attempts for incremental learning	Good	V. Good	Excellent	Excellent	Good
Explanation ability/ transparency of knowledge/ classification	Excellent	Average	Excellent	Good	Average
Support Multiclassification	Excellent	Naturally extended	Naturally extended	Excellent	Binary Classifier

VIII. CONCLUSION

In this paper the comparison of the most well known classification algorithms like decision trees, neural network, Bayesian network, nearest neighbour and support vector machine has been done in detail. The aim behind this study was to learn their key ideas and find the current research issues, which can help other researchers as well as students who are doing an advanced course on classification. The comparative study had shown that each algorithm has its own set of advantages and disadvantages as well as its own area of implementation. None of the algorithm can satisfy all the criteria. One can investigate a classifier which can be built by an integration of two or more classifier by combining their strength.

REFERENCES

[1] A. M. Javier ,M. Moguerza, "Support Vector Machines with Applications," *Statistical Science* , vol. 21, no. 3, pp. 322-336, 2006.
[2] C. J. Burges, "A Tutorial on Support Vector Machines for Pattern Recognition," *Data Mining and Knowledge Discovery*, vol. 2, 1998.
[3] Cover, T. , Hart, "Nearest Neighbor Pattern Classification," IEEE Transactions on Information Theory, vol. 13, no. 1, pp. 21-27, 1967.
[4] D. E. Rumelhart, G. E. Hinton and R. I. Williams, "Learning internal representation by error propagation," Parrallel Distrubuted Processing, 1986.
[5] Duda R, Hart P, "Pattern Classification and Scene Analysis," John Wiley and Sons, New York, 1973.
[6] F. Rosenblatt, "The perceptron: A probabilisticc model for informtaion storage and organization in the brain," *Psychological Review*, vol. 65, pp. 386-498, 1958.

[7] Friedman, N., Geiger, D., Goldazmidt, "Bayesian Network Classifiers," *Machine Learning*, vol. 29, pp. 131-163, 1997.
[8] Gao, Jiawei Hen and Jing, "Classification and regression trees," Wadsworth, Belmont, 1984.
[9] J. Han and M. Kamber, Data Mining Concepts and Techniques, Elevier, 2011.
[10] J. R. Quinlan, "Discovering rules by induction from large collections of examples," *Expert Systems in the Microelectronic age*, pp. 168-201, 1979.
[11] J. R. Quinlen, "Introduction of Decision Trees," *Machine Learning*, vol. 1, pp. 81-106, 1986.
[12] J. R.Quinlan, "C4.5: Programs for machien learning," Morgan Kaufmann, San Fransisco, 1993.
[13] Jensen, "An Introduction to Bayesian Networks," *Springer*, 1996.
[14] K. P. Soman, Insight into Data Miining Theory and Practice, New Delhi: PHI, 2006.
[15] Klaus_Robert Muller, Sebastian Mika, Gunnar Ratsch, Koji Tsuda, Bernhard Scholkopf, "An Introduction to Kernel Based Learning Algorithms," CRC Press, 2002.
[16] Robert Burbodge, Bernard Buxton, "An introduction to Support Vector Machines for Data Mining," Computer Science Dept., UCL, UK.
[17] S. B. Kotsiantis, "Supervised Machine Learning: A Riview of Classification Techniques," *Informatica*, vol. 31, pp. 249-268, 2007.
[18] Thair N. Phyu, "Survey of Classification echniques in Data Mining," in *International Multiconference of Engineers and Computer Scientists*, Hong Kong, 2009.
[19] Vapnik, Corinna Cortes and Vladimir, "Support Vector Network," *Machine Learning*, vol. 20, pp. 273-297, 1995.
[20] XindongWu, Vipin Kumar, J. Ross Quinlan, Joydeep Ghosh, Qiang Yang, Hiroshi Motoda, Geoffrey J. McLachlan, Angus Ng, Bing Liu, Philip S. Yu, Zhi-Hua Zhou, Michael Steinbach, David J. Hand, Dan Steinberg, "Top 10 algorithms in data mining," *Knowledge Information system*, vol. 14, pp. 1-37, 2008.

Ms. Hetal Bharat Bhavsar completed her M.E. Computer Science from D. D. University, Gujarat, India. She has more than 12 years of teaching experience at SVIT, Gujarat Technological University. Presently she is pursuing her Ph. D. in Computer Science and Engineering, affiliated to CHARUSAT University, Gujarat. She has published a number of papers in the proceedings and journals of national and international levels. She is a life member of CSI and ISTE professional bodies.

Mr. Amit Ganatra has received his M.Tech. degree 2004from Dept. of Computer Engineering, and Dharmsinh Desai University, Gujarat and he is pursuing Ph.D. in Information Fusion Techniques in Data Mining from KSV University, Gandhinagar, Gujarat, India and working closely with Dr.Y.P.Kosta (Guide). He is a member of IEEE and CSI. His areas of interest include Database and Data Mining, Artificial Intelligence, System software, soft computing and software engineering. He has 11 years of teaching experience at UG level and concurrently 7 years of teaching and research experience at PG level, having good teaching and research interests. In addition he has been involved in various consultancy projects for various industries. His general research includes Data Warehousing, Data Mining and Business Intelligence, Artificial Intelligence and Soft Computing. In these areas, he is having good research record and published and contributed over 70 papers (Author and Co-author) published in referred journals and presented in various international conferences