A Study on Feature Subset Selection For High Dimensional Data

A dissertation submitted in the partial fulfillment for the award of Degree of

Master of Technology

In

Software Engineering

By

Ayush Chandel (Roll No. 2K15/SWE/05)

Under the Guidance of

Dr. Rajni Jindal

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

DELHI TECHNOLOGICAL UNIVERSITY

2015-2017

CERTIFICATE

This is to certify that the thesis entitled "**A Study On Feature Subset Selection For High Dimensional Data**" submitted by **Ayush Chandel** in partial fulfillment of the requirements for the award of degree Master of Technology in Software Engineering, is an authentic work carried out by him under my guidance. The content embodied in this thesis has not been submitted by him earlier to any institution or organization for any degree or diploma to the best of my knowledge and belief.

Date:

Dr. Rajni Jindal
Associate Professor and H.O.D.
Department of Computer
Science and Engineering
Delhi Technological University

DECLARATION

I hereby declare that the thesis entitled "**A Study On Feature Subset Selection For High Dimensional Data**" which is being submitted to Delhi Technological University, in partial fulfillment of requirements for the award of degree of Master of Technology (Software Engineering) is an authentic work carried out by me. The material contained in the report has not been submitted to any university or institution for the award of any degree.

Ayush Chandel
2K15/SWE/05

ACKNOWLEDGEMENT

I would like to take this opportunity to express my appreciation and gratitude to all those who have helped me directly or indirectly towards the successful completion of this work.

Firstly, I would like to express my sincere gratitude to my guide **Dr. Rajni Jindal**, **Associate Professor and Head of the Department, Department of Computer Science and Engineering**, **Delhi Technological University, Delhi** whose benevolent guidance, encouragement, constant support and valuable inputs were always there for me throughout the course of my work. Without her continuous support and interest, this thesis would not have been the same as presented here.

Also I would like to extend my thanks to the entire staff in the Department of Software Engineering, DTU for their help during my course of work.

Last but not the least I would like to express my sincere gratitude to my parents and friends for constantly encouraging me during the completion of work.

Ayush Chandel

2K15/SWE/05

ABSTRACT

A high dimensional data is a data consisting thousands of attributes or features. Nowadays for scientific and research applications high dimensional data is used. But as there are thousands of features present in the data, we need to select the features those are non-redundant and most relevant in order to reduce the dimensionality and runtime, and also improve accuracy of the results. In this thesis we provide an overview of some of the methods which are present in literature. A study is done on the existing methods and a HYBRID algorithm for feature selection which incorporates the clustering aspects of FAST feature selection algorithm and similarity measure of a DICE coefficient. The efficiency and accuracy of the results is evaluated by empirical study. In this thesis, we have presented a novel clustering-based feature subset selection algorithm for high dimensional data. The algorithm involves

(i) removing irrelevant features,

(ii) constructing a minimum spanning tree from relative ones, and

(iii) partitioning the MST and selecting representative features.

In the proposed algorithm, a cluster consists of features. Each cluster is treated as a single feature and thus dimensionality is highly reduced. The Proposed System will be Implementation of FAST algorithm along with the DICE coefficient to remove irrelevant and redundant features.

Contents

List of Figures

CHAPTER 1

INTRODUCTION

We live in the data age in which aggregating information is simple and putting it away economical. In 1991 it was asserted that in every two months the amount of stored info doubles. Consequently, as the measure of machine lucid data expands, the capacity to comprehend and make utilization of it doesn't keep pace with its development. With the help of the tools provided by Machine learning expansive amounts of information can be consequently investigated. Machine learning has *feature selection* as one of its most important elements. Feature is determined by distinguishing the most striking components for learning and concentrates our learning calculation on those parts of the information which are most helpful for examination and future forecast. The speculation investigated in this thesis is that feature selection of tasks belonging to supervised classification can be completed based on the clustering between features, and that such a process of feature selection is essential to an array of machine learning algorithms. The component selector is straightforward and quick to execute. It disposes of unessential and excess information and, much of the time, enhances the execution of learning calculations.

1.1 MOTIVATION

The investigation of calculations that naturally enhance their own execution performance with experience comes under machine learning. The most important aspect of performance of any algorithm in machine learning is prediction. An algorithm is said to have learnt, when presented with data that exemplifies a task it itself improves its performance of predicting the key elements in that task. Machine learning calculations can be comprehensively described by the language used to represent the learned information. Research has demonstrated that no single learning algorithm is unmistakably prevalent in all cases, and distinct learning algorithms frequently deliver comparative outcomes. One component that can enormously affect the accomplishment of a learning algorithm is the way the information used to describe the task is learned. The machine learning algorithms exploits the statistical regularities of data and any failure in these regularities will result in the failure of the learning. It is conceivable that new data might be built from the old so as to display measurable statistical regularities and encourage learning, yet a fully automatic method is intractable due to the complexity of the task.

In any case, assuming the information is appropriate for machine learning, if by any way we are able to remove the features which are repetitive or are unessential, then, the task of finding regularities can be made sufficiently easier and this procedure is called 'feature selection'. Since our aim is not towards developing new information, therefore, feature selection is well defined and can be developed to be a fully automatic and computationally tractable process. The advantages of selection of subset of features for learning includes decrease in the amount of data which in turn is used for learning, enhanced accuracy of prediction, learned knowledge that is much more compact and easy to understand, and decreased execution time. Out of all the above mentioned components the decrease in execution time and ease to understanding of the result are much more significant in the sector of business and industrial information mining. A term begat to depict the way toward filtering through extensive databases for intriguing relationships and patterns is Data Mining. Since the expense of disc storage is declining day by day, the extent of numerous corporate and modern databases have developed to the point where examination by anything besides parallelized machine learning calculations running on exceptional simultaneously complex equipment is infeasible. Two methodologies that empower standard machine learning calculations to be connected to vast databases are 'feature selection' and 'sampling'. These two methods help in decreasing the span of the database by first recognizing the most salient components in the data (features) and then sampling by recognizing representative illustrations. Here in this postulation we concentrate on feature selection which is a procedure which makes learning calculations more efficient even if a huge amount of data with many dimensions is available.

1.2 PROBLEM STATEMENT

Since we know that the selection of a subset of features has been a dynamic research zone in pattern recognition, measurements, and data mining groups. The fundamental thought of Feature selection is to pick a subset of info variables by eliminating those features which have very little or no predictive knowledge. Feature selection can essentially enhance the fathomability of the subsequent classifier models and regularly construct a model that sums up better to unseen points. Further, it is generally the case that finding the right subset of predictive features is an essential issue in its own right. The fundamental point is picking a subset of good features concerning the objective ideas, include subset determination is a successful route for lessening dimensionality, evacuating insignificant information, expanding learning precision, and enhancing understandability of the result.

Hence out of the data with thousands of features we need to select only a limited number of features which are relevant and are not redundant. This can cause a reduction in dimensionality and runtime and enable the increase in accuracy. In the thesis a hybrid approach to feature selection is presented with the end goal being the production of a subset of features containing only relevant features. The proposed approach is a combination of FAST feature selection algorithm and DICE coefficient for similarity measure in order to reduce redundancy and irrelevancy of attributes or features.

1.3 REPRESENTATION

A key question in machine learning is the means by which instances can be represented. In most learning models it is accepted that the instances are given as a vector in R^n (where n is any finite measurement or dimension), and the examination begins from that point. There is a general agreement that once we have a decent portrayal, most sensible learning techniques will perform well after a sensible tuning exertion. Then again, on the off chance that we pick a poor portrayal, accomplishing a decent level of execution is hopeless. Be that as it may, how would we pick the most ideal approach for the representation of an abstract object (e.g. image) by a vector of numbers? A decent representation ought to be minimal and in the meantime significant as well. Is there a general technique to discover such a representation? Picking a representation implies picking an arrangement of features to be measured on each case. All things considered, this arrangement of features is normally picked by a human expert in the relevant area who has a decent instinct of what may work. The question is whether it is conceivable to discover algorithms that utilize the given training sample (and perhaps other outside information) to find a decent representation naturally.

On the off chance that the examples are physical entities (e.g. a human patients), picking the features implies picking which physical estimations to perform on them. In other cases the occurrences are given as a vector of numbers (e.g. the gray level of pixels for an image) and after that the task of discovering suitable representation (i.e. suitable set of features) is the assignment of finding a transformation that converts the first portrayal into a superior one. This should be possible by using the labels or in an unsupervised manner without using the labels. If the examples are initially portrayed by a vast arrangement features, one approach to handle this is by utilizing dimensionality reduction. In this method we search for a reduced arrangement of functions (of the initial features) that can represent relevant information. With regards to supervised learning, dimensionality reducing algorithms attempt to find few functions (features) that preserve the label information.

Feature selection is a special type of reduction in dimensions where we confine ourselves to picking just a subset out of the given arrangement of initial array of features. While this may in all accounts seems to be a solid restriction, feature selection and general dimensionality reduction are not that diverse, considering that we can simply first produce numerous conceivable functions of the initially raw features (e.g. numerous sorts of filters and descriptors of an image) and afterward utilize feature selection to pick just some of them. This procedure of producing complex components by applying functions on the crude elements (features) is referred to as feature extraction. In this way, at the end of the day, utilizing feature extraction and feature selection we can get a general reduction in dimensionality.

Feature selection tools are used for four reasons:

- simplification of models so that analysts/users are able to interpret easily,
- training takes much less time,
- to avoid the scourge of dimensionality,
- improved generalization by diminishing overfitting (formally, variance reduction).

The focal assumption when using a feature selection procedure is that many features in the data are either irrelevant or redundant, and can therefore, along these lines be removed without arousing much information loss. irrelevant or redundant features are two different concepts, since a level of correlation between two features can make a particular feature redundant.

Feature selection procedures are very much different from feature extraction. Feature extraction makes new feature components from initial feature functions, whereas, feature selection gives a subset of the feature elements as the output. Feature selection systems are regularly utilized as a part of spaces where there are many features for comparatively few specimens (or points of data). Hence, archetypal cases such as analysis of written texts, where there are thousands of features, and a few tens to hundreds of samples data, apply feature selection method.

Many feature subset determination techniques have been proposed and considered for machine learning applications. They can be separated into four general classes: the Embedded, Wrapper, Filter, and Hybrid which are further discussed about in chapter no. 2.

1.4 SELECTION OF FEATURES IN CASE OF MACHINE LEARNING

A machine learning assignment in order to accomplish itself requires many elements. The portrayal and nature of the example data is an initial matter of importance. Hypothetically, having more attributes should bring about additional segregating power. In any case, pragmatic involvement with many machine learning calculations has demonstrated that it is not the situation generally. Given an array of features the machine learning algorithm attempts to gauge a one-sided likelihood of the class tag but generally the distribution is very much complex and also the data contains many dimensions. Hence, induction is performed frequently on finite data. This makes evaluating the numerous probabilistic parameters troublesome. Keeping in mind the end goal to maintain a strategic distance from over fitting of data, numerous calculations utilize the Occam's Razor inclination to manufacture a straightforward model that still accomplishes some satisfactory level of execution on the data to be trained. This inclination frequently drives a calculation to lean toward few predictive

attributes over countless that, if utilized as a part of the correct mix of attributes, are completely predictive of the class name. In case the data is noisy and unreliable or too much irrelevant or redundant information is present, then the learning becomes difficult during the training phase.

Selection of a subset of features out of a large number of features is the way toward distinguishing and removing as much unimportant and excess information as possible out of the initial data provided. Therefore, lessening the dimensionality of the data may enable learning calculations to be performed quicker and all the more viably. At times, precision on future characterization can be enhanced; in others, the outcome is a more minimal target concept that can be easily interpreted.

Research has demonstrated that normal machine learning algorithms can be unfavourably influenced by redundant and irrelevant data. Irrelevant attributes affect simple nearest neighbour algorithm as its complexity (to reach a given accuracy level the number of training examples needed) grows as the number of irrelevant attributes increases. Decision tree algorithm sample complexity can develop exponentially for even a few concepts (for example, parity). Redundant attributes can adversely affect The naïve Bayes classifier as it assumes that attributes are independent given the class. Decision tree calculations can now and then over fit data to be trained, bringing about large trees. Much of the time, expelling redundant and irrelevant data can bring about delivering smaller trees.

This initial part of the chapter points to some basic connections between statistics and feature selection in machine learning and feature selection in pattern recognition. Critical parts of feature selection calculations are depicted and some basic heuristic search techniques are illustrated.

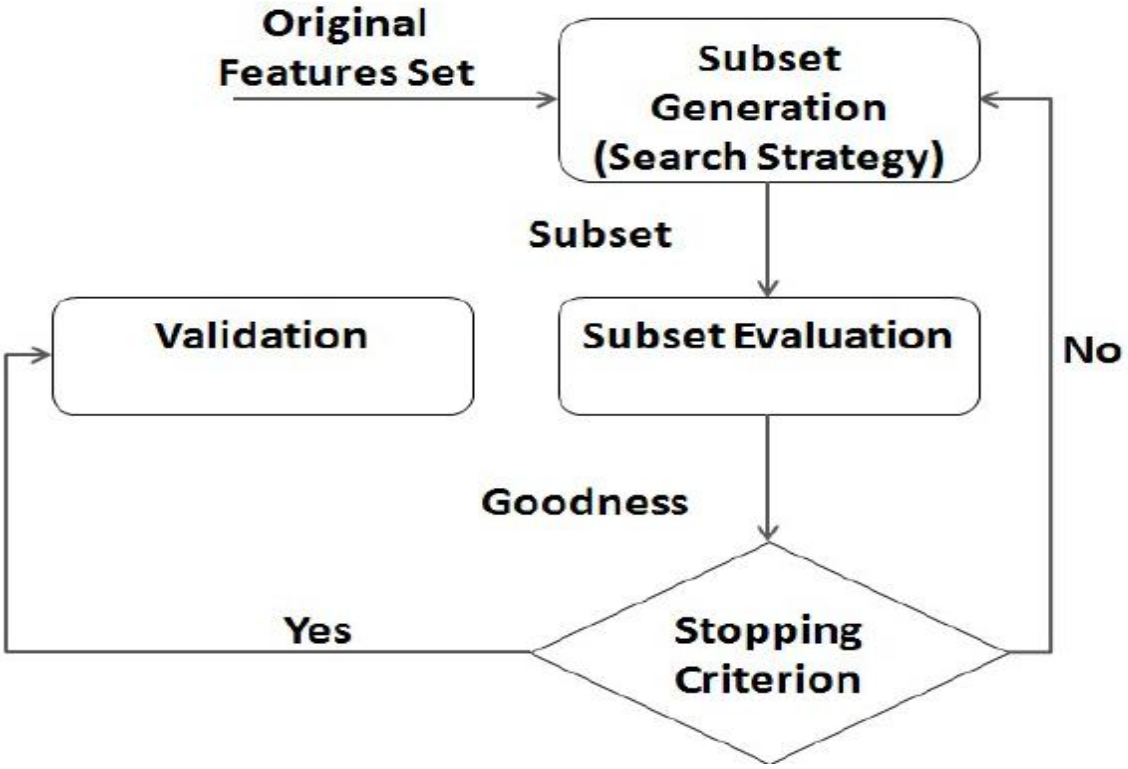

FIG 1: Flowchart depicting the process of feature selection

1.5 FEATURE SELECTION IN PATTERN RECOGNITION AND STATISTICS

The area of selection of subset of features has for quite some time been an exploration territory inside pattern recognition and the field of statistics [9] and hence, not astoundingly it can be said that it is an important issue for both machine learning as well as pattern recognition, as a common task of classification is involved in both the fields. In design acknowledgment, feature selection can affect the financial aspects of information procurement and on the precision and complexity of the classifier. It is likewise valid for machine learning, the only difference being that it has an additional load to refine valuable knowledge from the data. Luckily, the selection of features has been appeared to enhance the fathomability of extracted information.

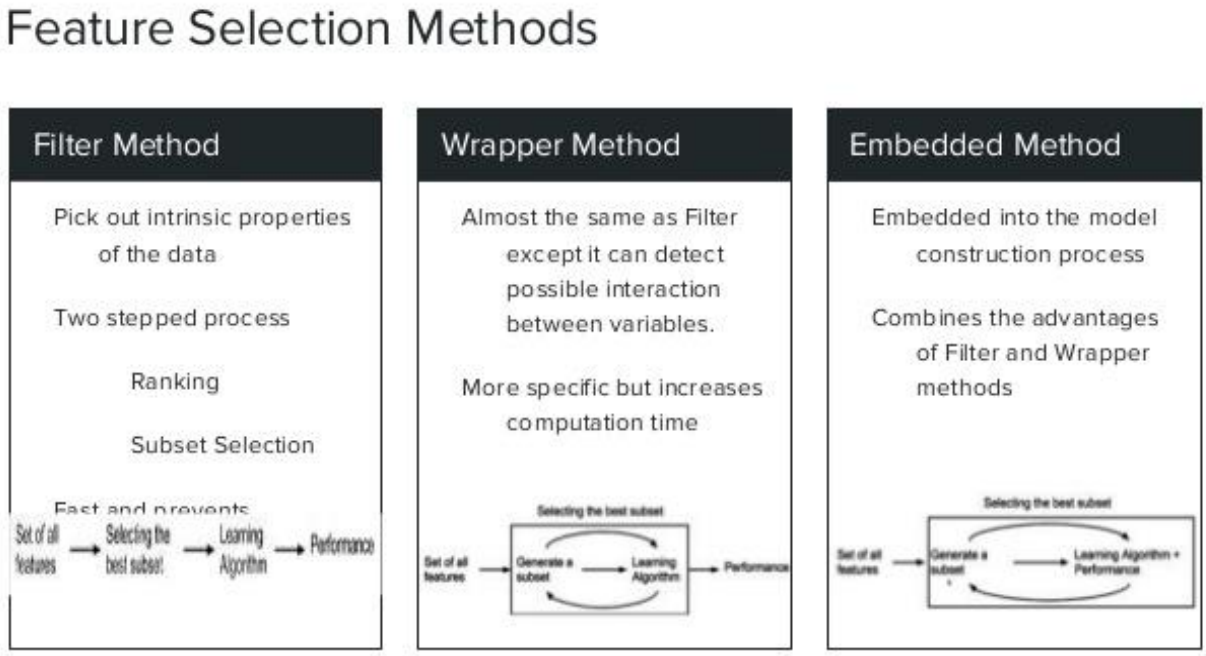

FIG2: The most basic feature selection approaches

Machine learning is motivated and has obtained from both statistics and pattern recognition. Considering an instance of the technique of searching successive backward elimination based on heuristics was first presented by Green and Marill; Similarly, distinctive variations of the technique, including a forward technique and a stepwise strategy were later presented by Kittler. The utilization of cross-validation for evaluating the exactness of a subset was introduced and it has turned into the foundation of the wrapper technique in machine learning. Allen recommended the technique and connected it to an issue in regression which is the issue of choosing predictors.

Numerous measurable methods for assessing the quality of subsets of features in light of properties of the data to be trained are just material to features which are numeric. Moreover, these measures being monotonic (expanding the span of the feature subset can never lead to performance reduction) are different from machine learning as this condition does not hold for basic machine learning algorithms.

Therefore, search algorithms, for example, dynamic programming and branch and bound, which depend on monotonicity for pruning the pursuit space, are not pertinent to calculations that utilize to coordinate the general bias of machine learning calculations in order to select the features.

1.6 CHARACTERISTICS OF FEATURE SELECTION ALGORITHM

Feature selection calculations play out a pursuit throughout the entire space of feature subsets, and, as a result, should focus on four fundamental issues which influences the way of the inquiry:

1. Beginning stage. Choosing a point in the set of feature space from where the inquiry begins. Direction of search can be affected with the initial set of points. An alternative is that at the start no features are selected and then progressively one after the other properties or attributes are included. In the above mentioned case, the pursuit is in forward direction through the search space. On the other hand, the pursuit could begin with all features with progressively discarding them. For this situation, the pursuit continues in reverse through the search space. There is another option in which we start from the middle and then move in outward direction.

2. Search association. A comprehensive inquiry of the whole set of features is restricted for everything except a finite starting number of features. With "n" starting features there exist 2^n conceivable subsets. Heuristic search [6] techniques are much more beneficial than thorough ones and would be able to give better outcomes, despite the fact that they don't ensure finding the ideal subset. Some heuristic search procedures that have been utilized for feature selection are examined in this chapter.

3. Assessment strategy: This step defines how the subsets under examination are separated based on a variable among feature selection algorithms for machine learning. One of the most popular strategy named the filter [8] works does not depend on any particular learning calculation. Even before the actual learning starts it aims to remove undesirable features. In order to access the value of a particular subset these methods utilize heuristics. There is another popular method which depends on a particular algorithm while choosing a contending subset of features. This strategy, called the wrapper, utilizes an enlistment algorithm alongside a measurable re-examining technique, for example, cross-approval to gauge the accuracy of feature subsets. Figure outlines the wrapper and filter ways to deal with feature selection.

4. Stopping measure: A criteria should be defined which enables to stop an algorithm from further looking into the subsets of features. Contingent upon the assessment system, a feature selector can stop looking to add or delete a feature from the subset if further such operations on it do not increase its value with respect to the current set. On the other hand, the algorithm may keep on revising the feature subset for as long as the length of the legitimacy does not corrupt. A different choice could be to keep producing feature subsets until a point where opposite end of the inquiry space is reaches and after that the best among those is taken.

1.7 HEURISTIC SEARCH

A data set generally contains a large number of elements with many dimensions. Hence, to perform a traverse the subset in a reasonable amount of time, some heuristic push is required. A basic pursuit technique which is known as hill climbing [6] works by making neighbourhood changes to the present feature subset. By removing or adding a feature a neighbourhood change can occur producing a subset which is better than the previous. If a change occurs by adding an element (feature) to the subset then it is known as forward selection; while considering just erasures is known as in backward elimination. There is another approach, called stepwise bi-directional search in which both expansion and cancellation occurs at each step and inside each of these variations, the inquiry algorithm may consider all conceivable nearby changes to the present subset and afterward select the best, or may just pick the principal change that enhances the value of the present feature subset. In either case, once a change is acknowledged, it is never reexamined. Figure demonstrates the feature subset space for the golf data. From start to finish, the graph demonstrates every neighbourhood expansion to every node; if filtered from base to the top, the outline demonstrates all conceivable nearby eliminations from every node.

Best first is an AI search technique that permits backtracking along the way of pursuit. Like hill climbing, best first travels through the inquiry space by rolling out neighbourhood improvements to the present feature subset. Be that as it may, dissimilar to slope climbing, if the path being investigated starts to look less encouraging, the best first inquiry can back-track to an all the more encouraging past subset and proceed with the pursuit from that point. If sufficient time is given, a best first hunt will investigate the whole inquiry space, so it is beneficial to utilize a stopping measure. Typically this includes restricting the quantity of completely extended subsets that outcome in no change. Table b demonstrates the best first search algorithm.

Genetic algorithms are versatile search methods which are based on natural selection principles in science. They utilize a populace of contending arrangements—evolving over time—to merge to an ideal arrangement giving an optimized solution. Subsequently, the arrangement space is looked in parallel, which helps in keeping away from local optima. For feature selection, an answer is normally a finite length parallel string mirroring a feature subset—the estimation of each position in the string speaks to the presence of a specific feature. The algorithm is an iterative procedure where each progressive generation is delivered by applying genetic administrators, for example, mutation and crossover to the present generation members. Some of the values of a set are changed randomly by mutation.

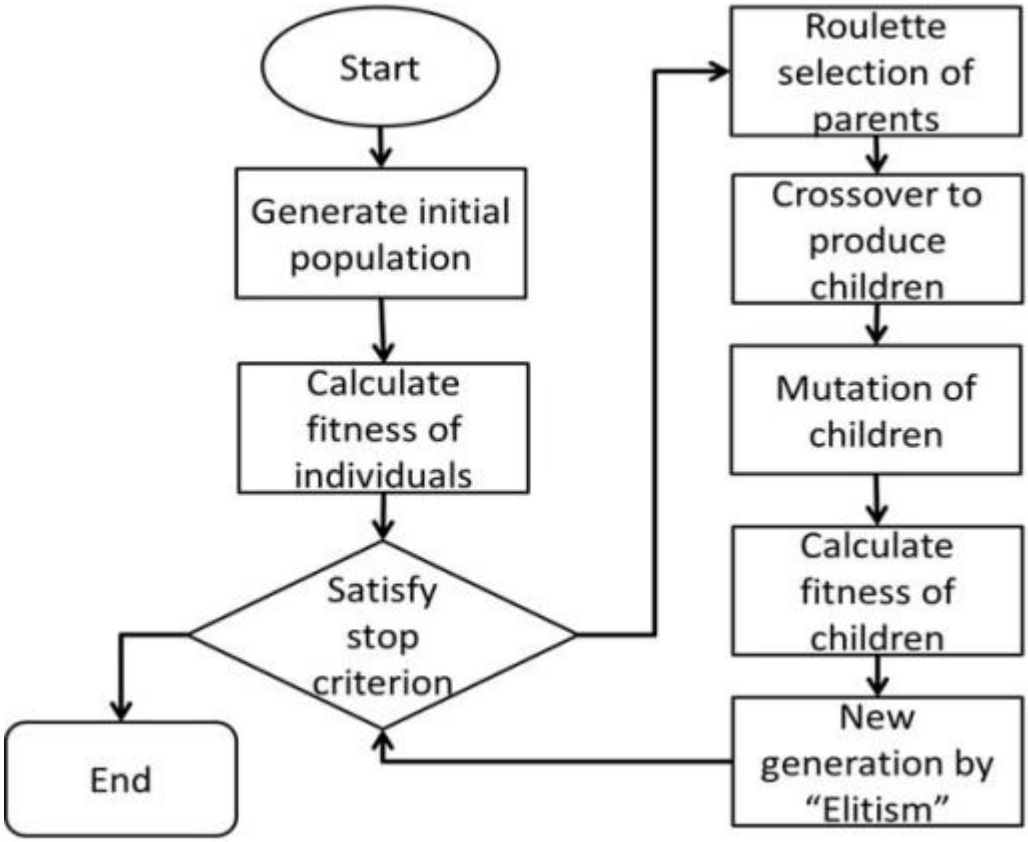

FIG3: A flow diagram for genetic strategy

Crossover joins distinctive features from a couple of subsets into another subset. The use of genetic administrators to populace individuals is controlled by their fitness (how great an element subset is as for an assessment system). Subsets which are better have a more noteworthy possibility of being chosen to frame another subset through crossover or change. In this way, after some time good subsets are developed. Table c demonstrates a straightforward genetic search system.

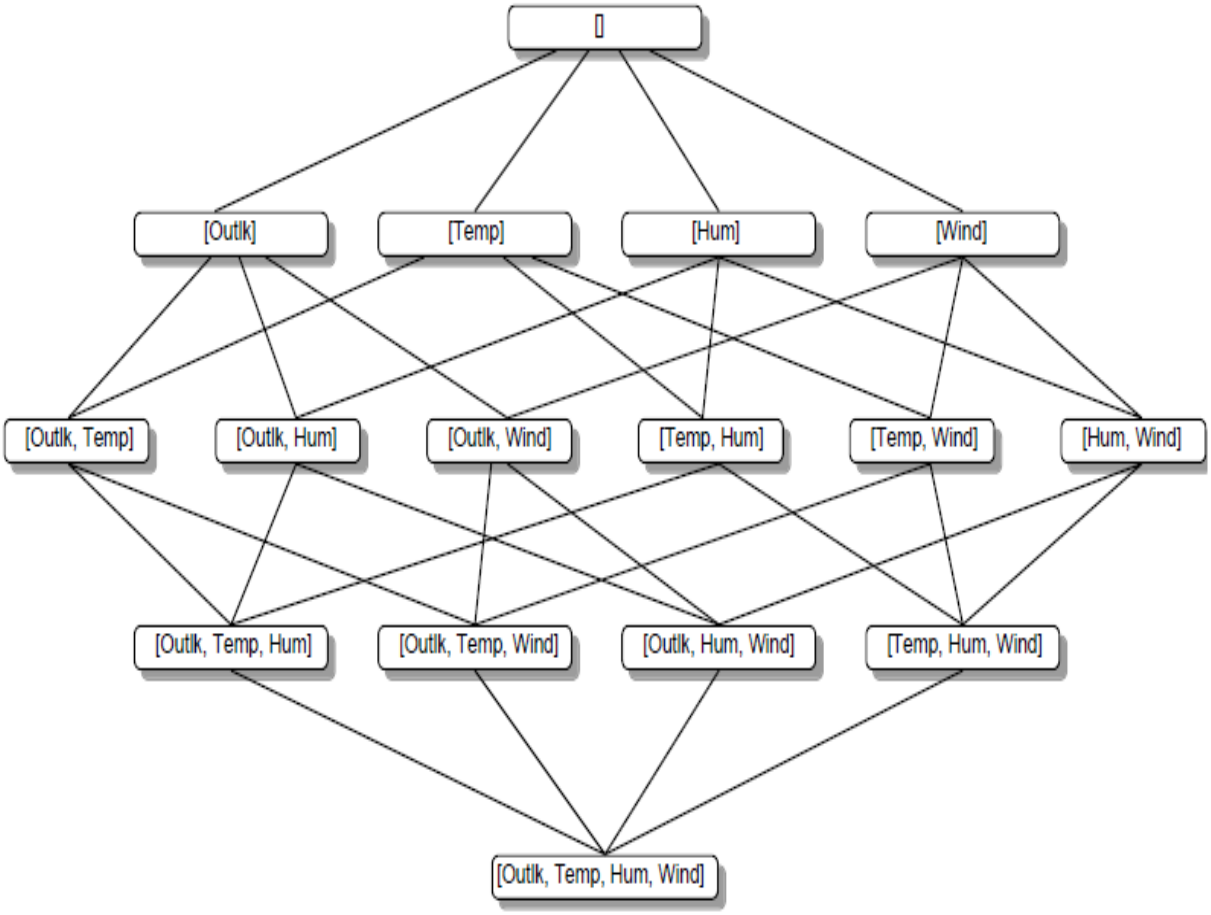

Fig 4: Feature subset space for golf dataset

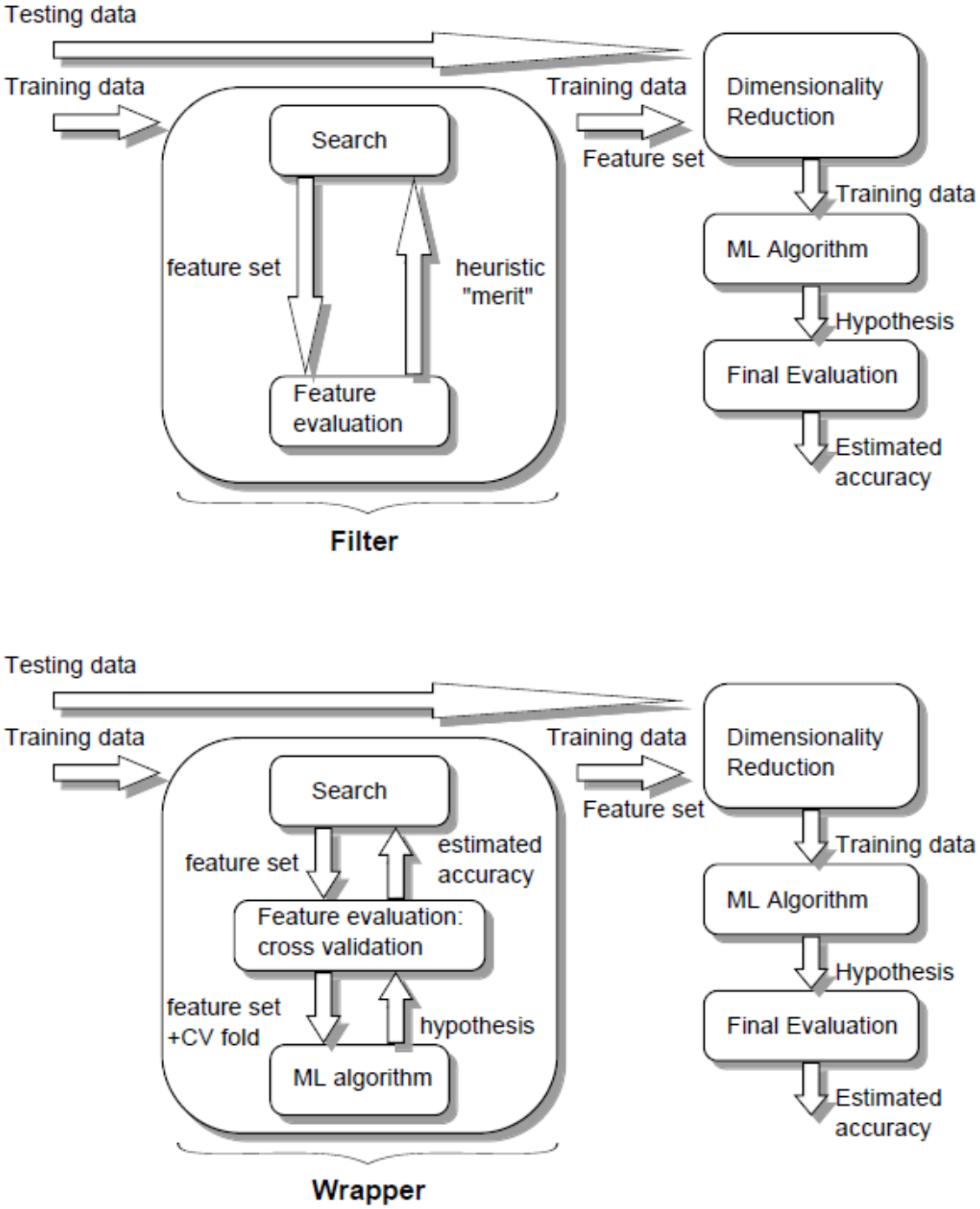

Fig 5: Filter and wrapper methods

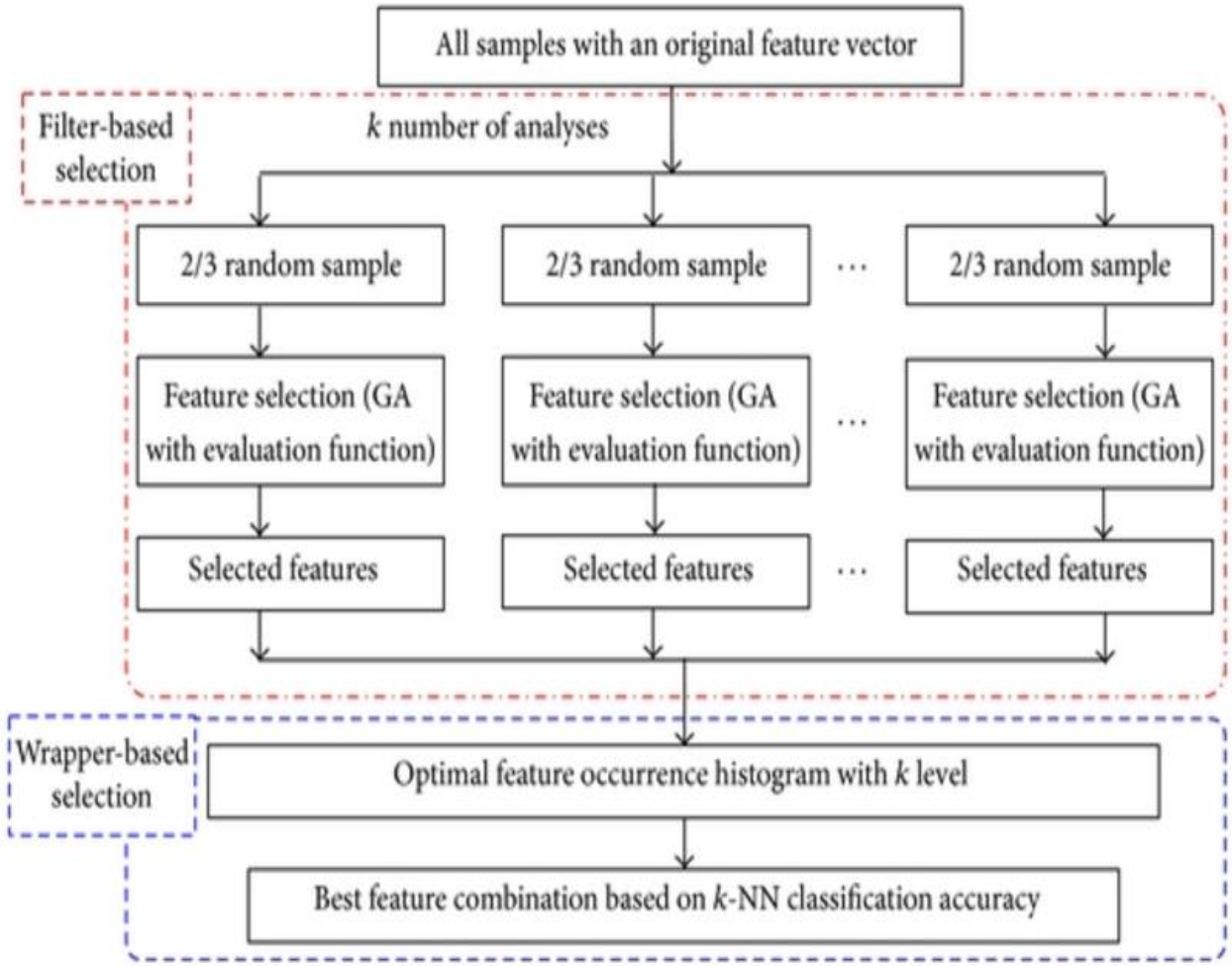

FIG6: A hybrid approach using both filter and wrapper for feature selection

CHAPTER 2

EXISTING METHODS AND RELATED WORK

2.1 FILTER METHOD FOR FEATURE SELECTION

One of the very first ways developed to deal with feature choice inside machine learning were filter techniques. All filter strategies utilize heuristics in light of general attributes of the information instead of a learning calculation to assess the value of each and every subset of features. As a result, filter techniques are for the most part substantially quicker than wrapper strategies, and, therefore, are much more functional for use on data which has a large number of dimensions.

2.2 CONSISTENCE DRIVEN FEATURE FILTER

Almuallim and Dieterich [2] defined a calculation initially intended for Boolean areas called FOCUS. The area comprehensively looks into the space of feature subsets until a point when it finds the base blend of features that partitions the data to be trained into unadulterated classes (that is, feature value combination is related with a solitary class). This is implied to as the "min-features bias". After the whole process of selection of feature a final decision tree is produced by passing the last produced subset containing features to the ID3. Some fundamental challenges are encountered with FOCUS, as indicated by Freitag and Caruanna [6]. One being that FOCUS is headed to accomplish consistency on the data, a comprehensive inquiry might be recalcitrant if many features are expected to achieve consistency. Also, a solid inclination in the direction of consistency may be measurably unjustifiable and shall prompt overfitting of the preparation data and hence, the calculation will keep on adding features to repair a solitary irregularity.

Three calculations—each comprising of method in which individual features are added to the subset combined with a heuristic measure which summarize the min-features inclination—are introduced [3] as strategies to make FOCUS practically possible on areas with many features.

Given below is the primary calculation which assesses features:

$$Entropy(Q) = -\sum_{i=0}^{2^{|Q|}-1} \frac{p_i + n_i}{|Sample|}\left[\frac{p_i}{p_i + N_i}\log_2\frac{p_i}{p_i + N_i} + \frac{n_i}{p_i + n_i}\log_2\frac{n_i}{p_i + n_i}\right].$$

The data to be trained is divided into a set of instances by feature set Q which have a similar truth assignments to the features in Q. Condition given above calculates the general entropy of the class esteems into these subsequent sets. Here, the notations of 'pi' and 'ni' signify the quantity of positive and negative cases in the ith assemble individually. After each subsequent stage the cardinality of the set of features decreases.

The second calculation picks up the most separating feature out of the remaining features and add it to the present subset at each phase of the inquiry. For a given combination of positive and negative illustrations, a feature is segregating if its value contrasts between the two. At each stage, the feature is picked which separates the best number of positive-negative sets of illustrations and which have not yet been segregated by any current feature in the subset.

The third algorithm resembles the second with the exception that every positive-negative illustration constitutes to a weighted addition to the score of each feature that separates it. The quantity to be added relies upon the aggregate number of features that segregate the match.

2.3 DISCRETIZED FEATURE SELECTION

It was observed by Setiono and Liu[4] that feature selection could be performed using discretization in case of numerical features. On the off chance of providing a solitary value by discretizing a feature which is numeric, a point can securely be expelled from the training data. There is a joined discretization and feature choice algorithm known as Chi2 examined later on in the chapter which utilizes a chi-square measurement X2 in order to perform discretization. Numeric qualities are at first sorted by putting each watched an incentive into its own interim. The quantities which are numeric are then over and again discretized by utilizing the X2 test to decide when adjoining intervals ought to be consolidated. The degree of the blending procedure is controlled by the utilization set threshold for X2 set. The limit is controlled by endeavouring to keep up the first devotion of the information—irregularity (measured same as in the LVF algorithm portrayed above) controls the procedure.

The paper report comes about on three domains containing a blend of nominal and numerical features before and after discretization. They finish up by reporting that Chi2 [7] is successful at enhancing C4.5's execution and wiping out a few features. Be that as it may, it is uncertain

whether C4.5's change is expected totally to a few features having been expelled or whether discretization assumes a part too.

2.4 COMBINING TWO FILTER ALGORITHMS

A few analysts have investigated the likelihood of utilizing a specific learning algorithm as a pre-processor to find the most appropriate subset of features for a particular learning calculation in focus.

Instanced based learners can use feature subsets which are selected by decision tree algorithms as was described by Cardie [20] in the paper. C4.5 was exercised to three characteristic language data indexes; just the features that showed up in a ultimate conclusion trees were utilized in a k closest neighbour classifier. The utilization of this cross breed framework altogether brought about improved performance over both C4.5 and the k closest neighbour algorithm when used separately.

By using a comparative strategy, Provan and Singh [12] utilized a decision choice tree algorithm to choose features which were in turn used to build a system of Bayesian classifier. Decision trees which are produced by an algorithm are different from oblivious decision trees as in such type of trees, the nodes which are present in a particular level test a particular trait of subset as in case of C4.5. During the experiment, firstly the subsets of features were chosen with the help of oblivious decision tree algorithms (a total of three trees were taken). Here, each of the oblivious tree utilized an alternate data theoretic part model—were assessed with a Bayesian system classifier on a large number of sets of data being used for machine learning. Results demonstrated that Bayesian systems utilizing features chosen by the above tree algorithms beat Bayesian systems without feature determination and Bayesian systems with features chosen by a wrapper.

Nevill-Manning and Homes [13] utilized 1R system described by Holte to gauge the prescient precisions of individual features. Rules are manufactured in 1R in light of a solitary feature. On the off chance that the information is partitioned into sets for preparation and testing, it is conceivable for each rule and each feature to compute the accuracy of classification of datasets. From characterization scores, a positioned rundown of features is achieved. Tests conducted while picking a number of selected and the most elevated positioned features and utilizing them with regular calculations in machine learning demonstrated that, when all things are being considered, the use of main three features are as precise as utilizing the whole set. The above strategy is uncommon because there is no search conducted. Rather, it depends on the client to choose to incorporate the number of features from the positioned array of features in the last subset.

Pfahringer [15] utilized a system which aimed to introduce classifiers based on decision table mainly to choose features. These classifiers also called DTM (Decision Table Majority) are a straightforward kind of closest neighbour classifier in which the comparability work is confined to saved examples which are exactly similar to classified instances. On the off chance that no cases are restored, the most pervasive class in the preparation data is utilized as the anticipated class; generally, every single coordinating example of majority of classes is utilized. Ostensible features makes these DTM to work best to their potential.

Acceptance of a DTM is accomplished by looking greedily into all possible conceivable decision tables. We know that decision tables are characterized by the order of the attributes it incorporates, hence, 51-principles can be viewed as decision trees with only one level.

In the above approach approach, the base portrayal length (MDL) standard works by guiding for a particular subset of features which is not properly classified by algorithm, the cost associated with the encoded decision table. The features which are produced in the ultimate decision table are evaluated and are then utilized with other learning calculations. Tests on a finite number of data sets for learning demonstrated that feature choice by DTMinduction can enhance the exactness of C4.5 now and again. DTMclassifiers instigated by utilizing MDL were additionally differentiated fron those instigated by utilizing cross-validation which is a strategy used in wrapper to evaluate the exactness of tables (and consequently feature sets). This approach discussed in this section was appeared to be more proficient, much easier to execute than cross-validation.

2.5 INFORMATION THEORETIC FEATURE FILTER

Recently, Koller and Sahami [16] presented an algorithm for feature selection in view of the concept of probabilistic reasoning and information theory. Suppose a feature set is given then, then the algorithm works upon the objective that the enlistment algorithm evaluates the likelihood dispersions over the class values such that the choice for a feature subset should endeavour to stay as near these initial distributions as could be allowed. For explaining purposes, let T be an arrangement of classes, F an arrangement of features, A a subset of F, f an array of qualities (f1, . . . , fn) to the features in F , and fx the projection of the qualities in f onto the factors in A. The objective of the feature selector is to pick A so that the values of Prob(T|A = fx) and Prob(T|F = f) converges. To accomplish this objective, the calculation starts by taking all the initial features and at each milestone utilizes a regressive end inquiry to evacuate the feature that causes minimal change between the two circulations. Since it is definitely not dependable to evaluate higher order probability distributions from constrained information, an estimated calculation is provided which utilizes the combinations of features pair wise. Cross entropy measures the contrast between two dispersions and the client must indicate what number of features must be given to the calculation for the purpose of evacuation. given a couple of features the cross entropy of a class can be given as:

$$D(Prob(T/Xi=xi, Xj=xj), Prob(T/Xj=xj)) =$$

$$\sum prob(t/Xi=xi, Xj=xj)log(prob(t/Xi=xi, Xj=xj)/(prob(t/Xj=xj))$$

The purpose of this calculation is to find out a set Mi for each feature Fi, containing K quantities from the remaining such that it is probably going to contain a feature i among all the class esteems. Our main aim is to evaluate the above equation with minimum estimation and calculate out of all the features a selected k contained in set Mi. The normal cross entropy between the dispersion of the class values, given Mi, Xi, and the circulation of class esteems given just Mi, is computed for each feature i. If the above calculated amount is insignificant for any feature, that feature is deleted from that particular set. This procedure continues until the number of features indicated by the user are expelled from the first set.

Investigations on four normal spaces and two simulated areas utilizing C4.5 [12] and naive Bayes as the last induction calculation, demonstrated that when the value of K in the set M is set to 2 the outcome generated is the most optimal. In two areas containing more than 1000 features the number of features produced by the algorithm in last step were less than half of the initial, while at the same time enhancing the exactness by maybe a couple of percent. However, there is an issue involved with the calculation which is that if the feature has two associated values than it must be encoded in parallel keeping in mind the end goal to maintain a strategic distance from the bias of the measures of entropy with multi valued features. This can enormously improve the quantity of features in the initial data, and additionally further dependencies are introduced. Moreover, the importance of the initial attributes is guarded, making the yield of calculations, for example, C4.5 difficult to decipher.

2.6 INSTANCE BASED APPROACH FOR FEATURE SELECTION

An algorithm known as RELIEF which assigned weights to each and every feature with the higher weighed features being desirable which in turn produced instances for learning was proposed by Kira and Rendell [18]. The more the weight of the feature, the more is its capacity to recognize a class among a number of classes. Features are positioned based on their weights whichever weight of the feature surpassed a client determined limit was chosen to frame the last arrangement of features. The calculation works by haphazardly testing occurrences from the preparation data. For each case examined, the closest occurrence of a similar class (closest hit) and inverse class (closest miss) is found. The weight of an a particular feature is refreshed based on the extent by which its value is able to recognize the tested values from its closest hit and closest miss. A high weight of the attribute depends on the off chance that it separates between examples from various classes and has values that are similar.

In case of a similar classes Condition given below demonstrates the updation of weight:

Wc = Wc – difference{(C,A,T)(C,A,T)}/q + difference{(C,A,T')(C,A,T')}/q

where, Wc is the weight for characteristic C, A is an arbitrarily tested occurrence, T is the closest hit, T' is the closest miss, and 'q' is the quantity of occurrences which are measured randomly. The capacity diff figures the distinction between two occurrences for a given characteristic. For attributes which are nominal it is characterized as either 1 (the qualities are distinctive) or 0 (the qualities are the same), while for attributes which are continuous, the distinction is the normalized difference in the space [0, 1] , separating by m ensures that all weights are in the interim [−1, 1].

RELIEF works in two different spaces. Kononenko depicts such improvements in RELIEF that empower it to adapt to multi-class, and even incomplete and noisy domains. Kira and Rendell [18] gave exploratory proof that shows RELIEF to be powerful at recognizing applicable features even withstanding the interaction between features (for instance, in parity issues). Be that as it may, RELIEF does not deal with features that are redundant. According to the authors:

"The flaw of the algorithm is that if the value of relevancy of any feature generally depicted by weight comes out to be more than the threshold then all such features will be taken even though the same amount of information can be given by a selected number of features."

Scherf and Brauer [21] depict a comparable case based approach (EUBAFES) to appoint feature weights autonomously of RELIEF. Like RELIEF, EUBAFES endeavours to strengthen closeness between cases of a similar class while at the same time diminish likenesses between cases of other classes. In order to modify feature weights for this objective a gradient descent advent is taken.

2.7 FEATURE WRAPPERS

Wrapper systems for selection of features utilize an induction calculation to evaluate if a particular array of features is legitimate. The justification given in favour of such strategy is that the induction used in the strategy will at last step give the feature subset which is ought to give a superior gauge of likeliness on comparison with an isolated measure that has a totally unique bias for induction [22]. Feature wrappers regularly accomplish preferable outcomes over the filter methods because of the way that they adjust to a particular cooperation between an induction calculation and its preparation data. Be that as it may, they have a tendency to be much slower than filters since they over and over again call the induction calculation. Also, it is made to run again if an alternate induction calculation is utilized. As we know that the wrapper is a very much characterized procedure, a large portion

of the variety in its application are because of the strategy utilized to assess the off-example exactness of an objective induction calculation, the objective induction calculation itself, and the association of the search. The following part of the chapter looks at the wrappers and the variations in the wrapper method which is aimed at the reduction of the whole cost of the computational process.

2.8 WRAPPERS BASED ON DECISION TREE LEARNERS

John, Kohavi, and Pfleger [23] gave the wrapper technique and presented a first approach of such a generalized technique in the field of machine learning. They wrote formal definitions based on the concepts of feature importance claiming that this technique is able to find the features that are more essential than others. Suppose there is a feature Ai, then, it is emphatically applicable with respect to any objective concept, if, for a given an initial set of features, expulsion of this feature from the feature set causes the value of likelihood distribution of the class to change. In the same way, we can define features which are not emphatically important i.e. features for which the value of likelihood distribution given a subset of initial features, removing a particular features changes the value of distribution of class. If by any chance a feature is not weekly or strongly applicable then it can be assumed to be irrelevant and hence, can be expelled out of the set. Three artificial and three natural spaces were chosen for experimentation by utilizing ID3 and C4.5 as the induction calculations. Exactness was evaluated by utilizing cross approval on the preparation information; a disjoint test set was utilized for detailing concluding exactness. Both forward determination and reverse elimination searches were utilized. Except for one counterfeit space which was artificial, results demonstrated that feature determination did not fundamentally change ID3 or C4.5's execution. Reduction in the size of trees was the primary impact of feature selection.

Following John et al., Caruanna and Freitag [15] tested various heuristic strategies with ID3 on two planning areas. And in addition using backward elimination and forward choice selection they likewise tested two variations of stepwise bi-directional search. The former starts with all the features wherein each step one or more features are removed while in the latter the initial subset is empty which is filled with features at each stage. Results demonstrated that in spite of the fact that the bi-directional searches somewhat outflanked the forward and backward searches, in general there was almost no contrast between the different search systems aside from the calculation time. Feature choice could enhance the execution of ID3 on both these planning areas.

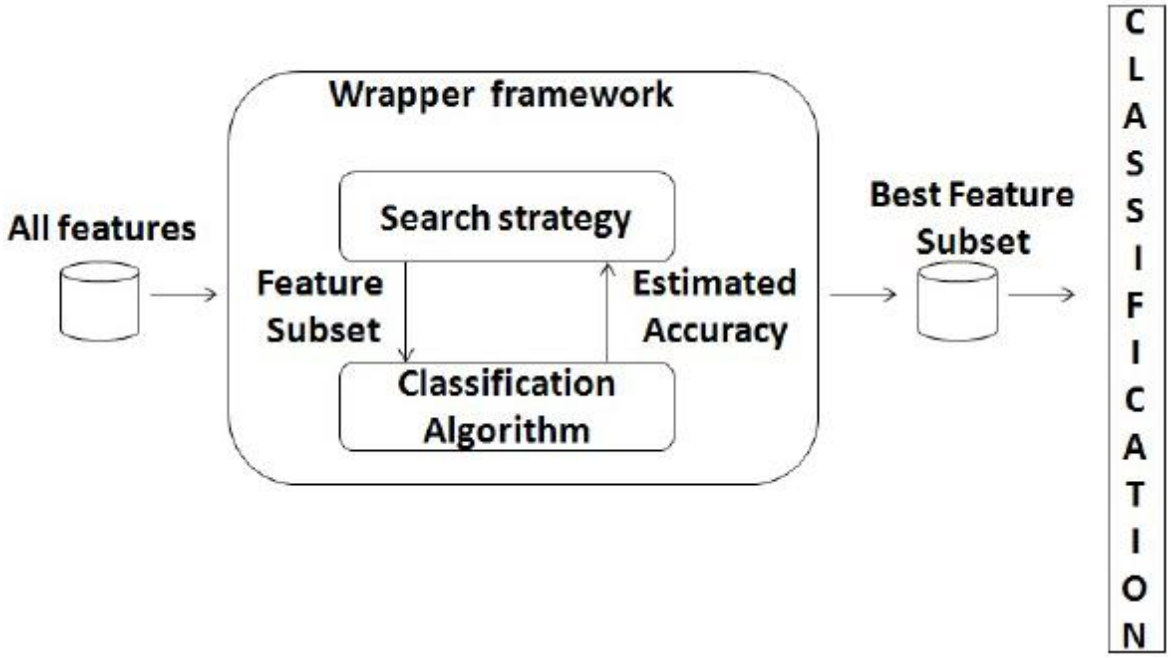

FIG7: Feature selection approach using wrapper

On the other hand, one thing that was common between method proposed by Cherkauer and Shavlik [24] as well as the one given by De Jong and Vafaie was that both connected search techniques based on genetic science in a wrapper structure for enhancing the execution of choice tree learners. Vafaie and De Jong [26] portray a framework which contains two modules working on genetic based calculations. One of the module performs the task of choosing relevant features while the other module is responsible for performing necessary induction calculations. The above two modules could fundamentally enhance the execution of ID3 on a surface characterization issue. Cherkauer and Shavlik [24] exhibit a calculation called SET-Gen which given a choice tree works towards enhancing its comprehensibility and exactness. To accomplish this, SET-Gen's genetic search utilizes a wellness work that is a direct mix of a precision term and an effortlessness term and is given below

Wellness function(W) = ¾(P) + ¼(1-(T+Q)/2)

where W is a feature subset, P is the average of cross-approval precision of C4.5, T is the normal size of the trees created by C4.5 (standardized by the a large quantity of preparing illustrations), and Q is the quantity of features is the subset W (standardized by the aggregate number of accessible features). Condition given above guarantees that the fittest populace individuals are those feature subsets that lead C4.5 to initiate small and precise decision trees.

2.9 INSTANCE BASED LEARNING WRAPPERS

The wrapper approach was proposed at roughly a similar time and was not related to John et al. by Langley and Sage's algorithm [28] amid their examination of the basic closest neighbour calculation's affectability to unessential qualities. Scaling tests demonstrated that the closest neighbour's specimen complexity (the quantity of preparing cases expected to achieve a given exactness) increments exponentially with the quantity of attributes which are irrelevant in the information. A calculation called OBLIVION is introduced which performs backward disposal of features utilizing an oblivious choice tree as the acceptance calculation. Tests with OBLIVION utilizing k-overlay cross approval on a few counterfeit areas demonstrated that it could expel excess features furthermore, learn quicker than C4.5 on spaces where features communicate.

Moore and Lee [31] adopted a comparable strategy to improve closest neighbour calculations, be that as it may, their framework does not use k-overlap cross-approval and concentrates on enhancing the expectation of numeric as opposed to discrete classes. Aha and Blankert additionally utilize leaving one attribute out cross validation, however combine it with a beam search. Their outcomes demonstrate that feature determination can enhance the execution of IB1 (a closest neighbour classifier) on an inadequate (not very many occurrences) design area with many features. Moore, Hill, and Johnson incorporate not just feature determination in the wrapper procedure, but additionally the quantity of closest neighbors utilized as a part of expectation and the space of mix capacities. Utilizing leaving one cross approval, they accomplish huge change on a few control issues including the forecast of ceaseless classes. In the same way, Skalak joined feature determination with model determination into a solitary wrapper handle utilizing irregular transformation hill climbing as the search technique. Exploratory outcomes indicated noteworthy change in precision for closest neighbour on two characteristic spaces and a radical diminishment in the storage demand of algorithm (number of occurrences held amid training).

Domingos [3] portrayed a wrapper method to deal with feature determination for learners using instantiation. The inspiration for such a method is that there might be attributes which are either applicable to just a confined zone of the example domain and unessential somewhere else, or pertinent given just certain number of esteems (which reacts weekly) of different features and are otherwise not essential. In any of the above cases, when all around accessing of feature set is done (by taking the whole array of instances), the superfluous parts of features in the set may overpower their viewpoints which are required for learning based on instantiation. This is also the case when backward search techniques are utilized with the wrapper. Domingos [3] presented a calculation called RC with the ability to recognize and make use of important features. The functioning of RC starts by choosing a conceivably diverse arrangement of features for each occurrence in the preparation set. It does this by utilizing a backward search system and cross approval to assess exactness. For each example in the preparing set, RC discovers its closest neighbour of a similar class and evacuates those features in which the two contrast. The exactness of the whole preparing dataset is then

assessed by cross approval. In the event that the precision has not corrupted, the new instance is acknowledged; generally the occurrence is re-established to its unique state and deactivated (no further feature determination is endeavoured). The feature choice process proceeds until all examples are inert.

Analyses of a number of machine learning datasets demonstrated that RC beat standard wrapper feature selectors utilizing forward and in reverse selection techniques on a number of instance based learners. The efficiency context delicate approach moreover appeared on manufactured areas designed to display confined feature dependency. At the point when features are all around significant or insignificant, RC has no preferred standpoint over standard wrapper feature determination. Besides, when couple of illustrations are accessible, or the information is boisterous, standard wrapper methodologies can identify universally unessential features more effectively than RC. Domingos [3] additionally noticed that wrappers that utilize instance based learners (counting RC) are unsatisfactory for use on databases containing many examples since they are quadratic in N (the quantity of examples).

Kohavi [5] utilized wrapper feature determination to investigate the capability of DTM classifiers. Convenient data structures permit the utilization of quick incremental cross-approval with DTM classifiers. Investigations demonstrated that DTM classifiers utilizing suitable feature subsets contrasted positively for complex calculations as in case of C 4.5

2.10 WRAPPERS FOR BAYES CLASSIFIER

Because of the naive Bayes classifier's presumption that, inside each class, likelihood dispersions for properties are free of each other, Langley and Sage note that its execution on spaces with excess features can be enhanced by expelling such features. A forward search system is employed to choose features for use with naive

Bayes, instead of the regressive methodologies that are utilized regularly with choice tree calculations and subsequent instance based learners. The basis for a forward search is that it ought to instantly distinguish conditions when destructive repetitive features are included. Experimentations demonstrated general change and expanded learning rate on three out of six normal areas, with no change on the three which remained.

Pazzani [6] later combined feature determination and basic inductance system in a wrapper system for enhancing the execution of naive Bayes. Forward and reverse hill climbing search methodologies are thought about. In the previous case, the calculation considers not just the augumentation of single features to the present subset but also additionally making another property by going along with one features which has not yet been selected with each of the chose features in the subset. In the last case, the calculation considers both erasing respective features and supplanting sets of features with a joined feature. Results of experimentation on

a number of machine learning datasets demonstrate that both methodologies enhance the execution of naive Bayes. The forward system is much more superior in expelling repetitive features than the reverse procedure. Since it begins with the full arrangement of features, and considers all conceivable pairwise joined features, the backward technique is more viable at recognizing attribute connections than the forward technique.

Much better version of naive Bayes utilizing wrapper-based feature choice is additionally detailed by Kohavi and Sommerfield and Kohavi and John.

Provan and Singh [8] have connected the wrapper to choose features which subsequently are used to built Bayesian systems. Their outcomes demonstrated that while feature choice did not make strides precision over systems built from the full arrangement of features, the systems developed after feature determination were extensively smaller and quicker to understand.

2.11 IMPROVING THE WRAPPER TECHNIQUES

Many critique of the wrapper way to deal with feature choice are worried with its computational cost. For each feature subset analyzed, an induction calculation is conjured k times in a k-fold cross validation. This can make the wrapper restrictively moderate for use on substantially bigger data sets with many features. This downside of the method has driven a few specialists to research methods for alleviating the cost of the assessment procedure.

Caruanna and Freitag [6] came up with a plan that reserves decision trees. This can significantly diminish the quantity of trees developed amid feature determination and permit bigger spaces to be sought.

Moore and Lee [10] present a strategy to "race" contending models or feature subsets. In the event that eventually amid leave one out cross-validation, a subset regarded to be far-fetched to have the most reduced assessed error, its assessment is ended. This has the impact of decreasing the rate of preparing cases utilized amid assessment and diminishes the computational cost of completely assessing every subset. The calculation likewise "obstructs" all close indistinguishable feature subsets—aside from one—in the race. This avoids running feature subsets with almost indistinguishable predictions to the end. Both dashing and blocking utilize Bayesian insights to keep up a likelihood appropriation on the gauge of the mean cross validation error for each contending subset. The calculation utilizes forward choice, yet rather than successively attempting every neighbourhood change to the best subset, these changes are dashed. The race completes when just a single contending subset remains or on the ending of cross validation.

Kohavi and John [23] present the idea of "compound" inquiry space administrators in an endeavor to make in best first search and backward techniques computationally plausible. At the point when every single neighborhood change (augmentations or cancellations of single features) to a given feature subset have been assessed, the primary compound administrator is made, joining the two best neighborhood changes. This administrator is then connected to the feature subset, making another subset facilitate away in the pursuit space. In the event that the main compound administrator prompts a subset with an enhanced estimate, a compound administrator is built that joins the best three nearby changes, and so on. The utilization of compound administrators pushes the hunt more rapidly toward the firmly pertinent features. Experimentations utilizing compound administrators with a forward best first search demonstrated no huge change in the precision for ID3 or naive Bayes. At the point when compound administrators were consolidated with a best first look, accuracy decreased marginally for ID3 yet enhanced for C4.5. The poor outcomes with ID3 recommend that the best first search can even now stall out in some nearby maxima. The change with C4.5 is because of C4.5's pruning (again a type of feature determination) because of which the algorithm is not stuck on a local maxima.

2.12 FEATURE WEIGHING ALGORITHMS

Giving weight to features can be seen as a speculation of feature determination. In feature choice, feature weights are limited to 0 or 1 (a feature is utilized or it is definitely not). Feature weighting permits better separation between features by doling out each a constant esteemed weight. Calculations, for example, closest neighbour (that typically treat each feature similarly) can be effortlessly altered to incorporate feature weighting while figuring closeness between cases. One thing to note is that, as a rule, feature weighting calculations does not diminish the dimensionality of the information. Unless features with low weight are evacuated from the training data at first, it is accepted that each feature is valuable for induction; the extent of its weight reflects the degree of its usefulness. Utilizing weights for features also includes seeking a considerably bigger space and a more prominent shot of overfitting.

Salzberg [14] consolidated incremental feature weighting in an instance based learner referred to as EACH. If the classification made is right, the weight for each matching feature is increased by f (the worldwide feature change rate). Crisscrossing features have their weights decremented by this same sum. For inaccurate arrangements, the inverse happens—features that are mismatched are increased while the weights of coordinating features are decremented. Salzberg noted that the estimation of f should be tuned for various informational sets to give best outcomes.

Wettschereck and Aha [17] took noted that EACH's weighting plan is not applicable to skewed depictions of concepts. IB4 is an expansion of the k closest neighbour calculation that addresses this issue by computing a different arrangement of feature weights for each and every concept. The weight for feature 'i' is processed utilizing

Aggregate Weight is relied upon to close on to one portion of Weight Normaliser for attributes which are clearly unimportant. Both Cumulative Weight and Weight Normaliser are incrementally refreshed on learning. Given 'h' be the highest of the watched frequencies among the classes of two cases X (the case to be arranged) and Y (its most comparative neighbour in the description). Total Weight i is increased by

$$1 - \text{diff}(x_i, y_i) \times (1 - h)$$

if X and Y have the same class

$$\text{diff}(x_i, y_i) \times (1 - h)$$ otherwise

where, increment in weight normalizer is (1-h). Experimentations showed its performance to be better than k nearest neighbour algorithm for irrelevant attributes.

RELIEF is a calculation that uses an occasion based way to give weights to features. Wettschereck and Aha utilized RELIEF algorithm to compute weights for a k nearest neighbour calculation and they reported critical change over standard k nearest neighbour in seven out of ten specified areas.

Kohavi, Langley, and Yun [19] further depicted a way to deal with feature weights that considered a little arrangement of discrete weights as opposed weights which are continuous. Their approach utilizes the wrapper combined with basic nearest neighbour to appraise the exactness of feature weights and a best search to investigate the weight space. In tests that broaden the quantity of discrete weights considered by the calculation demonstrated that there is no preferred standpoint in expanding the quantity of non-zero discrete weights over two; except for some precisely created artificial space domains, utilizing one non-zero weight (proportional to feature determination) was hard to outflank. Fundamental Relief can assess the nature of numerical and discrete features, which are fully associated. For instance, parity problems, where the learning examples are depicted with an extra number of insignificant features, Relief can recognize a subset of important features. A more practical variation of Relief is its augmentation, called ReliefF. The first Relief was intended for two-class issues without missing esteems and is very delicate to noise.

The above strategies for feature weighting all utilize feedback from a nearest neighbour calculation (either incrementally amid learning or in a unique stage before induction) to

change weights. Some non-feedback techniques for setting weights includes: the per classification feature significance which sets the weight for a feature to the restrictive likelihood of the class given the feature, the cross-classification feature significance, which is like feature significance per category however over classes it averages, and the common data between the feature and the class. These methodologies require numeric features to be discretized.

Reasonable machine learning calculations frequently make presumptions or apply heuristics that exchange some exactness of the model for speed of execution, and fathomability of the outcome. While these presumptions and heuristics are sensible and regularly yield great outcomes about, the existence of redundant and irrelevant data can regularly trick them, bringing decreased exactness and less justifiable outcomes. Feature subset determination can offer assistance on concentrating the learning calculation on the critical features for a specific issue. It can likewise lessen the dimensionality of the information, enabling learning calculations to work quicker and all the more successfully.

There are two primary ways to deal with feature subset choice depicted in the chapter. The wrapper—which is tuned to a particular interplay between an acceptance calculation and its preparation information—has been appeared to give great outcomes, yet practically speaking might be too sluggish to be of viable use on expansive genuine spaces containing many features. On the other hand, Filter strategies are significantly speedier as they does not include over and again invoking of a learning calculation. Existing filter arrangements show various drawbacks. A few calculations can't deal with noise (or depend on the client to determine the level of commotion for a specific issue). In a few cases, a subset of features is not chosen unequivocally; rather, features are ranked with the last decision left to the client. A few calculations don't deal with both excess and superfluous features. Different calculations increase features to be changed such that really builds the underlying number of features and henceforth the pursuit space. This last case can result in lost significance from the first portrayal, which thus can have an effect on the understanding of actuated models.

Feature weights are effectively fused into learning calculations, for example, nearest neighbour, and in any case, the benefit of feature weighting over feature choice is insignificant, due to the expanded shot of overfitting the information. Finally, feature weighting does not diminish the dimensionality of the first information.

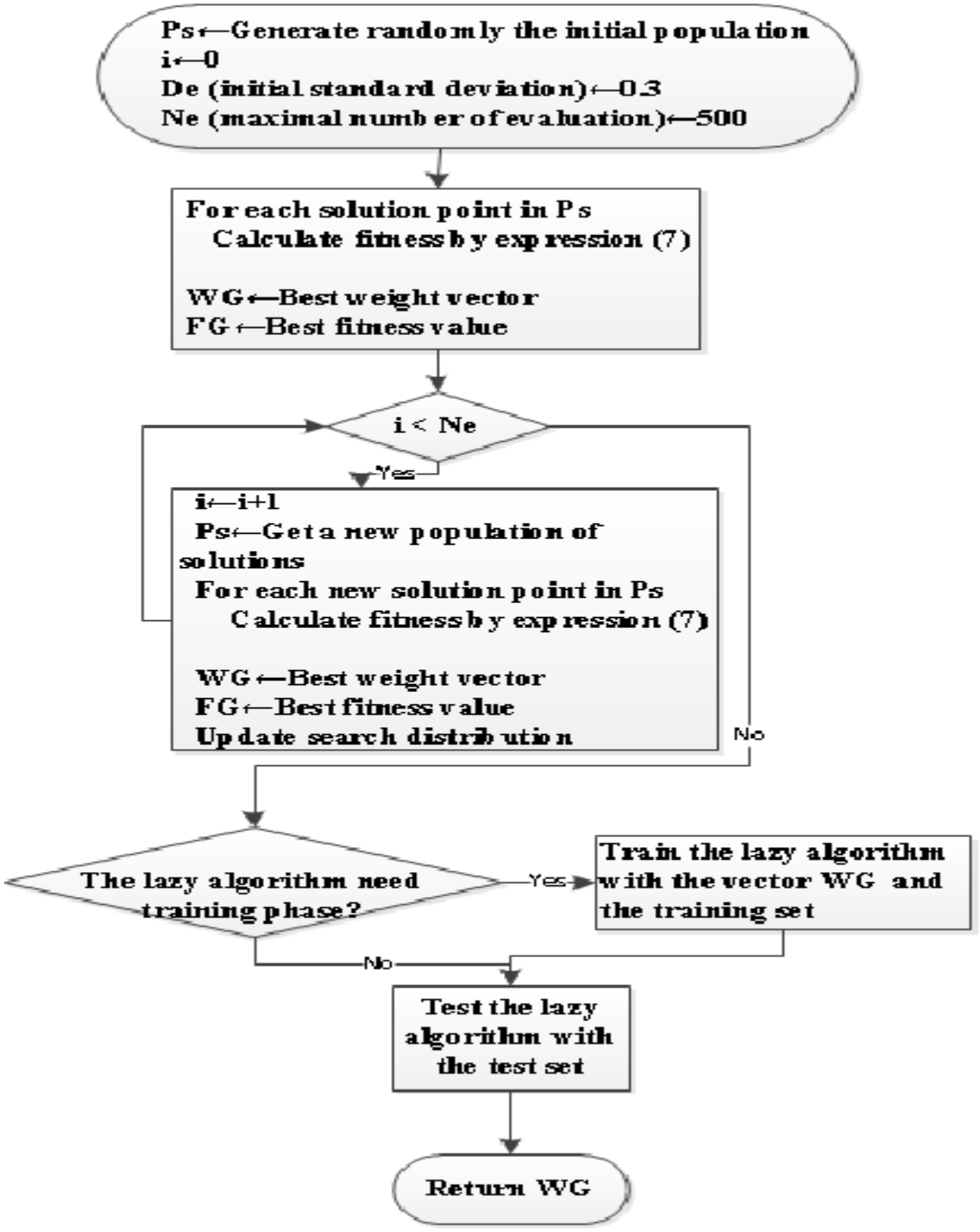

FIG8: The flowchart above shows the evolutionary weight updation technique

CHAPTER 3

A CLUSTERING BASED FAST FEATURE SELECTION

3.1 SCHEMA AND IMPORTANT TERMS

Features which are insignificant, alongside features which are redundant, seriously influence the precision of the learning machines. In this manner, feature subset selection ought to have the capacity to recognize and expel as much as superfluous and excess data as could be expected. Considering these points, a novel calculation which can productively and viably manage both insignificant and repetitive features, and get a decent feature subset is required. We accomplish this through another feature selection structure which is composed of two associated parts of irrelevant feature expulsion and redundant feature expulsion. The previous gets features significant to the objective idea by taking out immaterial ones, and the last expels repetitive features from important ones by means of picking delegates from various feature groups, and in this manner creates the last subset. The expulsion of features which are not relevant is clear once the correct measure is characterized or chosen, while the excess feature end is a touch more complex. In FAST algorithm calculation, it includes

1) the development of the minimum weight tree(MST) from a complete weighted graph;

2) the dividing of the MST into forests with different clusters represented by different trees; and

3) the selection of features to represent each group or forest.

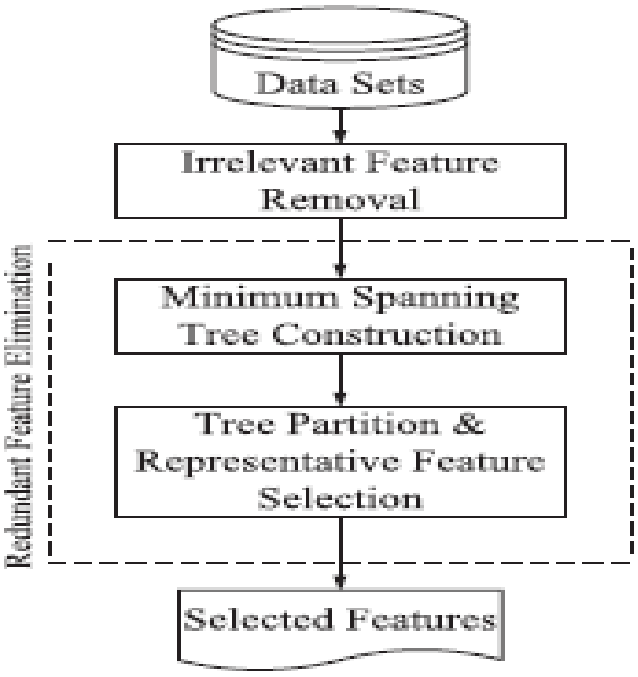

FIG9: Framework for FAST algorithm

Keeping in mind the end goal of more decisively presenting the calculation, and on the grounds that feature subset selection system includes irrelevant feature evacuation and redundant feature removal, the conventional definitions of redundant and irrelevant features are presented [25], and then definitions based on variable dependencies are also provided below

Let there be a set 'F' representing an array of features, F_i be a feature belonging to the set F,

S_i = F-{F_i } is a set and S_i' is proper subset of $S_i$and C being the value of target concept which is to be achieved,

*1st annotation: Relevant Feature: "*Fi is relevant to the target concept C if and only if there exists any s_i, fi, and c, such that, for probability P($S_i' = s_i', F_i = f_i$) >0,

P(C=c/$S_i = s_i, F_i = f_i$) is not equal to P(C=c/$S_i = s_i$)"

Otherwise the features are said to be irrelevant. The other features contain most of the information which a redundant feature provides.

2nd annotation: Markov Blanket: "If a feature Fi belonging to feature set F is given and let M_i is a proper subset of F, then Markov blanket of feature Fi is denoted by Mi and

P(F- M_i –{ Fi }, C/ Fi, M_i) = P(F- M_i –{ Fi }, C/ M_i) "

3rd annotation: Redundant features: "Considering a set of features S, a feature in Fi in S is said to be redundant if within S it has a markov blanket."

Important features which have solid relationship with concept C (target) are constantly essential for a best subset, while excess features with their esteems totally associated with each other are not essential. In this way, thoughts of feature relevance and feature redundancy are regularly based on relationship between features and concept connection of feature with its target. Shared information is a quantification of difference of the feature esteems and target classes from statistical independence. This is a varied assessment of connection of esteems of the features and the target.

The value of SU [25] is evaluated with the help of shared data in which it is normalized to a value based on the esteem value or the value of esteem and target class, and then utilized to assess the integrity of features used for characterization by various specialists. In this manner, we pick SU as the measure to show relationship between a feature with another feature in feature set or the concept C which is the target.

Equation of symmetric uncertainty is given below

$$SU(A, B) = \frac{2*GAIN(u/v)}{H(u)+H(v)}$$

Here,

H(A) is the entropy of a random discrete variable A. Assume that all esteems of A has probabilities p(A) and H(A) is characterized by

$$H(A) = - \sum_A p(a)\log p(a)$$

INFOG(A/B) [25] is the sum by which there is a reduction in value of B. It mirrors the extra data about any random variable A and is known as the information gain and is calculated as below

$$INFOG(A/B) = H(A) – H(A/B)$$
$$= H(A) – H(A/B)$$

Where H(A/B) is the conditional entropy which measures the rest of the entropy (or uncertainty) of an arbitrary variable A given that the estimation of another arbitrary variable B is known. Assume, p(a) is the initial liklihood for all estimations of A and p(a/b) is the likelihood presence of 'a' when 'b' is given.

$$H(A/B) = - \sum_b p(b) \sum_a p(a/b)\log p(a/b)$$

The quantity of INFOG() is symmetrical in nature which means the measure of increase in the about variable X on watching Y and the measure of information increase about variable Y on watching X are equal. Hence the estimation of measure will not be affected by the two factors which implies that both (A, B) and (B, A) are equivalent.

The way in which the quantity SU works is that it handles multiple factors simultaneously and adjusts for value of the quantity INFOG() towards the direction of factors which have more number of values inclination toward factors and standardize it in the range of [0,1]. SU(X/Y) value of 1 shows that learning of the estimation of it totally predicts the estimation of the other and the esteem 0 uncovers that X and Y are not at all dependent on each other. This type of measure which uses entropy for its calculation generally works for discrete quantities, however, the continuous factors can be discretized initially and can managed by these measures.

Suppose SU(A,B) is the symmetric vulnerability of factors A and B and pertinence is T-Relevance between a feature and the C is the objective idea. The correlation F-Correlation between a match of features, F-Redundancy is the feature redundancy and RFeature is the delegate feature of the feature group (cluster). The definitions if all the above terms are given below:

*4th annotation: T-Relevance: "*It is the extent to which a feature Fi of feature set F is relevant to a target concept C and is represented as SU(Fi, C). Generally a minimum threshold 't' is kept. If the value of SU(Fi, C) comes out to be greater than the defined threshold we say that the feature has strong T-Relevance."

5th annotation: F-Correlation: "It is the extent to which a feature Fi is related to a feature Fj and is denoted by SU(Fi, Fj).Here, both the features Fi and Fj belong to the feature set F and Fi is not equal to Fj."

6th annotation: F-Redundancy: "If S = {F1; F2; . . . ; Fi; . . . } is a set of features belonging to a cluster, then there exist a feature Fi in the set S such that the relation given below

$$SU(Fj/C) \geq SU(Fi/C) \text{ and } SU(Fi/Fj) > SU(Fi/C)$$

is true for all the features in the set S with i and j being different. Then, the feature Fi is redundant with respect to Fj and hence is not essential can be removed."

7th annotation: R-Feature: "R-Feature is a feature Fi belonging to the cluster S if it has the maximum value of SU(Fj/C) among all the values present in the cluster. This feature is then used to represent the whole cluster"

The above statement implies that the feature Fi which has the maximum value of T-Relevance can be used as R-Feature of each and every feature in the cluster S.

By the above definitions it can be understood that the features which have the a high TRelevance values are chosen out of which the representing feature RFeatures are found out from the clusters. Following two points can be observed from the theory and are given below:

1. If correlation between features and target concept C is feebly related. It indicates presence of irrelevant features ;

2. Features which are redundant are able to be clustered and the representing feature of the cluster can be removed.

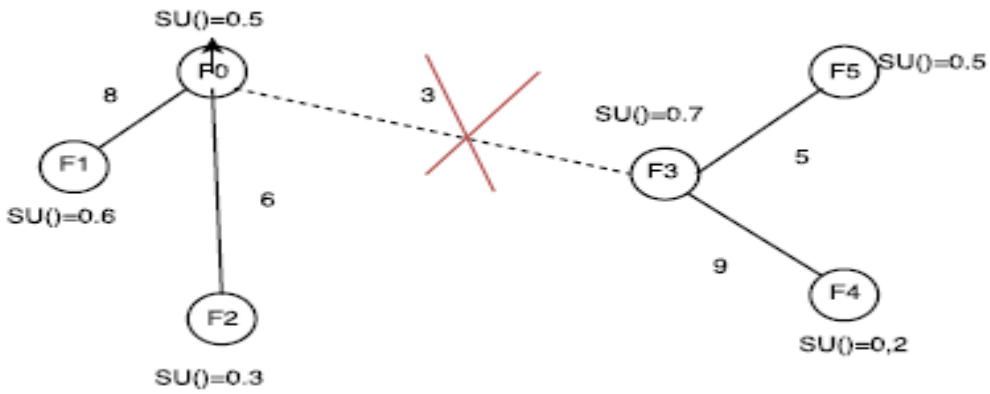

FIG10: Clustering step involved as the process in the algorithm

3.2 ANALYSIS OF THE ALGORITHM

The above clustering based algorithm legitimately comprises of the following steps:

1) features which are irrelevant are removed,

2) a minimum spanning tree is constructed from the remaining features

3) based on the measure of TRelevance and FCorrelation the MST can be broken down into forests representing different clusters each having a representative feature value.

Suppose D is an informal index with and F is a feature set with ‘m’ features such that F = {F1; F2; . . . ; Fm}. Considering a concept C, the first step involves calculation of T-Relevance for each feature Fi (1 < i < m) i.e. SU(Fi, C). The feature which have the value of SU(Fi, C) greater than a predefined threshold ‘t’ are taken in the target value subset of the algorithm.

Considering an initial feature set F, in the second step the value of measure SU(Fi, Fj) (FCorrelation where i is not equal to j) is found. The graph G mirrors the connections among all the objective significant features. But, the graph G has k vertices and k(k - 1)/2 edges which for high-dimensional information, is intensely thick and the edges with various weights are firmly interlaced. Decomposing a complete graph is a NP hard problem which makes situation difficult in case of high dimensional data. Therefore for G, we construct a MST, which associates all vertices utilizing the notable Prim's calculation such that the aggregate of the weights of the edges is the least and thereby helping in achieving the end goal. The value associated with edge (F'i; F'j) is FCorrelation SU(F'i; F'j). This is the weight relation in the above calculation.

In the wake of building the MST, in the third step, we initially evacuate the edges E = {(F'i; F'j)/(F'i; F'j having a place with F' and i, j has a place with [1, k] with i not being equivalent to j, whose weights are littler than both of the T-Relevance SU(F'i, C) and SU(F'j, C), from the MST. Every cancellation brings about two separated trees T1 and T2.

Accepting the arrangement of vertices in any of the last trees to be V (T), we have the property that for each combination of vertices (F'i, F'j) has a place in V(T), SU(F'i, F'j) >= SU(F'i, C) and SU(F'i, F'j) >= SU(F'j, C) essentially holds. This property results in features that are redundant remaining in V (T).

CHAPTER 4

DICE COEFFICIENT

The dice coefficient which is also called a **Sørensen–Dice index** is a measure which is used in the comparison of two samples based on some measure of similarity. The idea of such a similarity measure was given by Lee Raymond Rice. The other names of this index include "similarity coefficient" or "index".

The original intention of this index was to show the presence or absence of any data in two samples A and B where the number of elements in each set are given by $|A|$ and $|B|$ and THE similarity quotient is given below:

$$\frac{2\,|A \cap B\,|}{|A\,| + \,|B\,|}$$

And based on the above it can be written as

$$\text{DSC} = \frac{2TP}{2TP+FP+FN}$$

The difference of the dice coefficient with another similarity measure which is known as Jaccard Index is that the latter counts only the truth values of the numerator and denominator. The value of the Similarity Coefficient of the Dice coefficient lies in the range [0, 1] and can be seen on the same light as measure of similarity on sets.

Vector operations define the set operations over the set X and Y

$$\text{s} = \frac{2AB}{A^2+B^2}$$

The above formula works not only for binary vectors but vectors in general.

When similarity measure is to be calculated in terms of a string it is done by using bigrams as given below

$$s = \frac{2n_t}{n_a + n_b}$$

Here, n_t is the number of bigrams found in both the strings. Also, the number of bigrams in first and second string are n_a, n_b respectively. For example, suppose there are two strings

Night and nacht

First and foremost each string is divided into bigrams i.e. {ni, ig, gh, ht} and {na, ac, ch, ht}.

Here the cardinality of both the set is 4.

In the above example the element which is common in both the sets is {ht} and therefore the number of bigrams which are common are 1.

By using the above formula for similarity measure we get

s = 2*1/4+4 = 0.25

CHAPTER 5

PROPOSED SYSTEM

5.1 ALGORITHM

In the following thesis a hybrid algorithm for feature selection in proposed. In the system, we turn our focus to two factors, one being the relevance between any two features of the set and the second being the relevance of the feature with a target concept. In order to calculate the relevance between the features the quantity which has been taken is the Dice coefficient. A representative feature is extracted from each cluster of features to remove any redundancy and this removal is based on the relevance between different features and between the target concept and each feature. A lot of techniques have been proposed earlier for the task of feature selection but most of these techniques aim to remove irrelevant features and do not pay attention to redundant features. The system which is being proposed in this thesis aims to remove both redundant as well as irrelevant features thereby making the subset produced in the end much more optimal. The system utilizes the concept of first dividing the whole initial array of features into clusters using the concept of MST (minimum spanning tree) as FAST clustering based algorithm does and then choosing a feature from each cluster to represent that particular cluster. The proposed system is the hybrid of clustering based FAST feature selection algorithm and the Dice coefficient of similarity measure.

Another advantage of feature selection is its usefulness in data analysis process. This is due to the fact that it shows which of these features are important and thus can be used for prediction. The use of clustering in above system is that the features which are similar can be grouped together and hence irrelevant features can be removed easily. Elimination of irrelevant features is easier and most importantly redundancy can be removed. Selected datasets are thereby obtained from clustering.

A minimum spanning tree is a tree whose combined weight of edges for a given number of vertices is smaller than any other tree with the same number of vertices. Basically every uni-directed graph which may not be connected forms a minimum spanning forest which is made up of many minimum spanning trees. When the clustering analysis of data is done the data generally can comprise of dimensions ranging from a few dozens to even thousands. This type of high dimensional data is generally encountered in areas related to medicine where a

large number of dimensions are produced at a single time technologies involving DNA microarray and also in the area of text documents clustering.

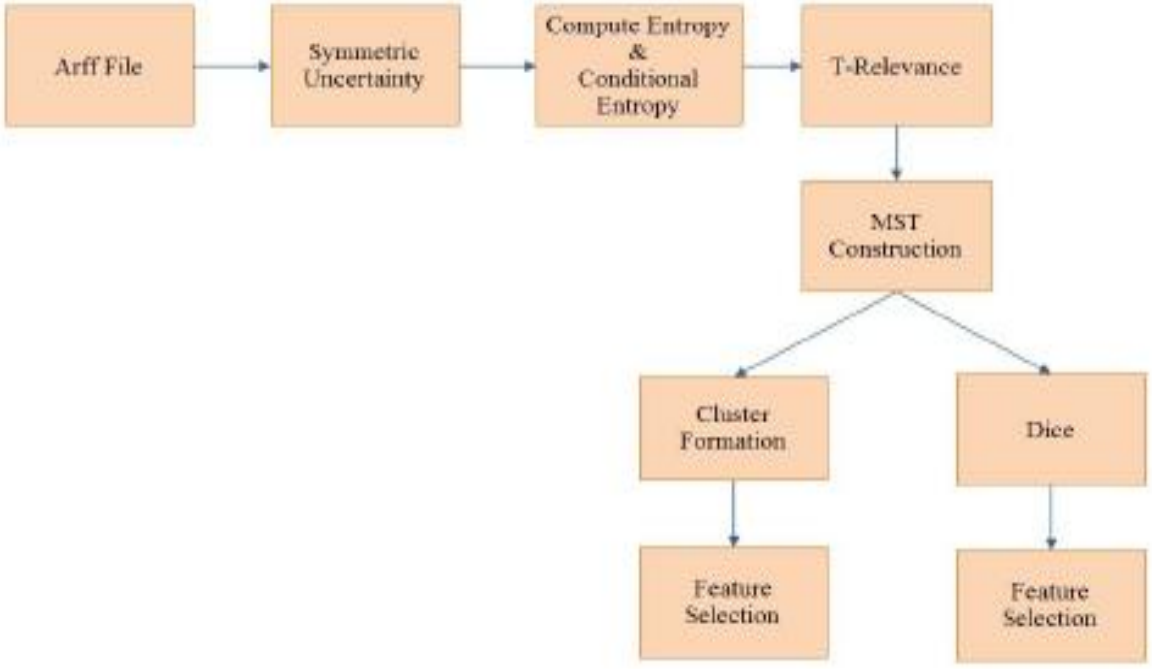

FIG11: Figure representing the hybrid algorithm for feature selection

5.2 FLOW CHART DIPICTING THE HYBRID ALGORITHM

The diagram which is given below shows the implementation of feature selection based on clustering. The basic idea of feature selection (or variable selection) is the selection of features for the usage in model construction in case of machine learning and statistics.

The following benefits are provided by such algorithms:

1 Interoperability of the model is improved

1 It takes much less time to train the model

3 Over fitting is reduced and thus there is an increase in generalization

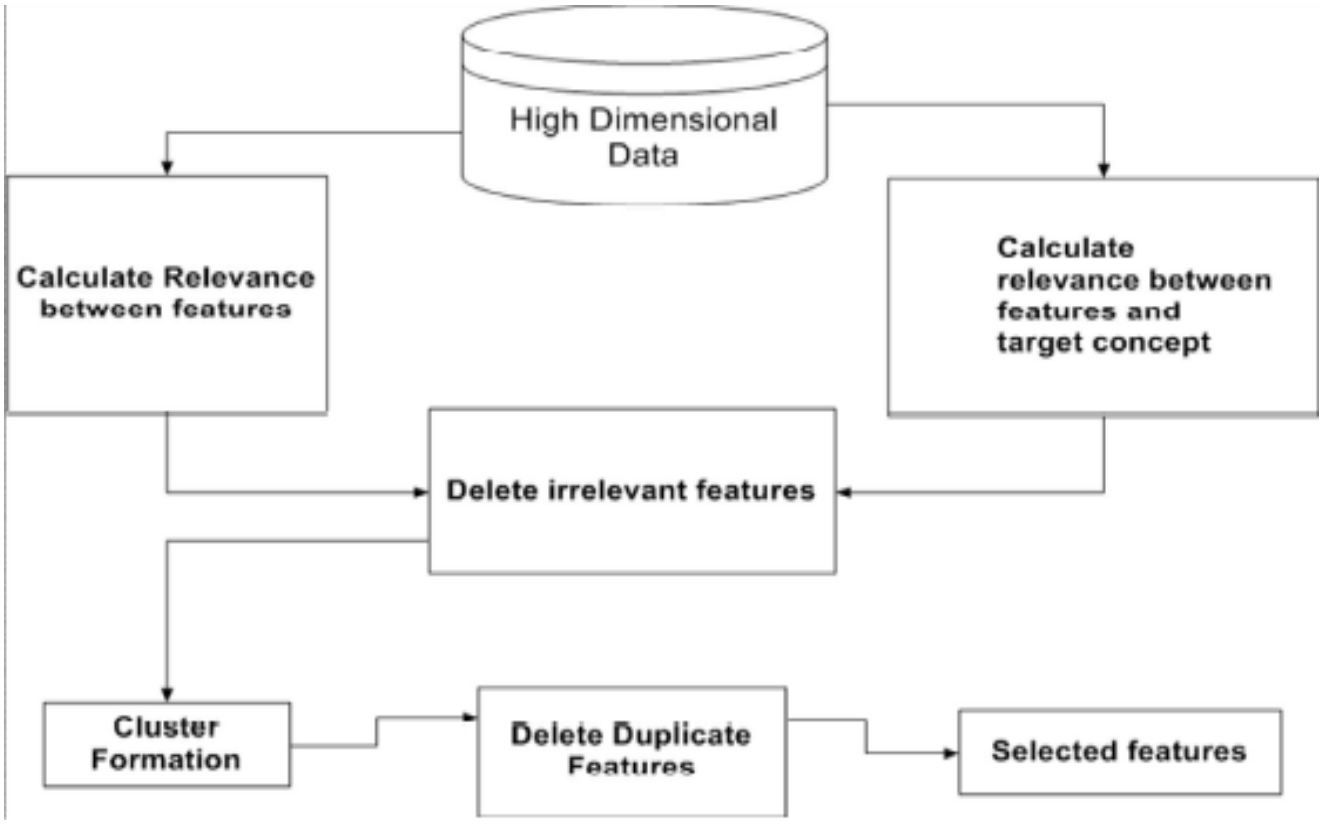

FIG12: Flowchart depicting the hybrid aproach

CHAPTER 6

IMPLEMENTATION

The following chapter deals with the research work with respect to its experimental setup. The chapter is divided into three sections. Section 6.1 deals with the software and tools used, section 6.2 deals with the system requirements which is used to implement the proposed system and section 6.3 deals with the output with respect to each dataset.

6.1 THE TOOL AND SOFTWARE USED

The scheme which is proposed in the thesis is implemented using SCIKIT- learn which is a machine learning library containing tools for classification, regression and clustering such as Support Vector Machines and Random Forests. The language which is used for the implementation is PYTHON.

There are a total of the datasets used which are as follows:

1 IRIS dataset

2 Breast Cancer (Diagnostic) dataset

3 Digit dataset

6.2 SPECIFICATION OF THE SYSTEM

The following are the specifications on which the proposed system works and is evaluated against the other algorithms:

OS: Windows 7

Processor: Intel Core i5, 2.1 GHz

Memory: 4GB

6.3 OUTPUT

The following contains the output of the system with respect to the iris dataset when

1 The classifier used is Decision Tree

```
No Feature Selection
Classifer: DecisionTree
Best score: 0.96

with FAST
Classifer: DecisionTree
Best score: 0.953333333333
Elapsed Time: 0.015841960907

hydbrid
Classifer: DecisionTree
Best score: 0.953333333333
Elapsed Time: 0.0188829898834

with k best
Classifer: DecisionTree
Best score: 0.7
Elapsed Time: 0.0170140266418

with RFE recursive features selection
Classifer: DecisionTree
Best score: 0.706666666667
Elapsed Time: 0.0129289627075

with chi2
Classifer: DecisionTree
Best score: 0.693333333333
Elapsed Time: 0.0173509120941
```

2 The classifier used is Logistic Regression

```
No Feature Selection
Classifer: LogisticRegression
Best score: 0.966666666667

with FAST
Classifer: LogisticRegression
Best score: 0.946666666667
Elapsed Time: 0.0190830230713
```

```
hydbrid
Classifer: LogisticRegression
Best score: 0.946666666667
Elapsed Time: 0.0173478126526

with k best
Classifer: LogisticRegression
Best score: 0.673333333333
Elapsed Time: 0.0170369148254

with RFE recursive features selection
Classifer: LogisticRegression
Best score: 0.686666666667
Elapsed Time: 0.014349937439

with chi2
Classifer: LogisticRegression
Best score: 0.686666666667
Elapsed Time: 0.0170550346375
```

The following contains the output of the system with respect to the breast cancer dataset when

1 The classifier used is Decision Tree

```
No Feature Selection
Classifer: DecisionTree
Best score: 0.915641476274

with FAST
Classifer: DecisionTree
Best score: 0.892794376098
Elapsed Time: 1.99794507027

hydbrid
Classifer: DecisionTree
Best score: 0.893394376098
Elapsed Time: 1.88068580627

with k best
Classifer: DecisionTree
Best score: 0.892794376098
Elapsed Time: 10.1985478401

with RFE recursive features selection
Classifer: DecisionTree
Best score: 0.882249560633
Elapsed Time: 10.2871899605
```

```
with chi2
Classifer: DecisionTree
Best score: 0.898066783831
Elapsed Time: 11.087665081
```

2 The classifier used is Logistic Regression

```
No Feature Selection
Classifer: LogisticRegression
Best score: 0.95079086116

with FAST
Classifer: LogisticRegression
Best score: 0.901704745167
Elapsed Time: 2.26395487785

hydbrid
Classifer: LogisticRegression
Best score: 0.916704745167
Elapsed Time: 2.05792212486

with k best
Classifer: LogisticRegression
Best score: 0.910369068541
Elapsed Time: 11.3869140148

with RFE recursive features selection
Classifer: LogisticRegression
Best score: 0.91388400703
Elapsed Time: 12.0738861561

with chi2
Classifer: LogisticRegression
Best score: 0.915641476274
Elapsed Time: 11.3438799381
```

The following contains the output of the system with respect to the DIGIT dataset when

1 The classifier used is Decision Tree

```
Using the decision tree
No Feature Selection
Classifer: DecisionTree
Best score: 0.835837506956

with FAST
Classifer: DecisionTree
Best score: 0.81393377852
Elapsed Time: 2.39760494232

hydbrid
Classifer: DecisionTree
Best score: 0.826377295492
Elapsed Time: 2.4536819458


with k best
Classifer: DecisionTree
Best score: 0.822481914302
Elapsed Time: 3.62280988693

with RFE recursive features selection
Classifer: DecisionTree
Best score: 0.800779076238
Elapsed Time: 1.97197508812

with chi2
Classifer: DecisionTree
Best score: 0.819699499165
Elapsed Time: 1.02919387817
```

2 The classifier used is Logistic Regression

```
No Feature Selection
Classifer: LogisticRegression
Best score: 0.936004451864


with FAST
Classifer: LogisticRegression
Best score: 0.903171953255
Elapsed Time: 2.39633107185

hydbrid
Classifer: LogisticRegression
Best score: 0.903171954252
Elapsed Time: 2.38589406013
```

```
with k best
Classifer: LogisticRegression
Best score: 0.875904284919
Elapsed Time: 3.54621601105

with RFE recursive features selection
Classifer: LogisticRegression
Best score: 0.894268224819
Elapsed Time: 1.9013299942

with chi2
Classifer: LogisticRegression
Best score: 0.884251530328
Elapsed Time: 0.968654155731
```

CHAPTER 7

RESULTS AND ANALYSIS

This chapter deals with the analysis of outputs which were observed in the last chapter. The parameter which is used to differentiate between the different algorithms is the 'score' which here is equivalent to the accuracy to which the subset of features selected by the algorithm is able to classify the given dataset.

The following table represents the score with respect to each algorithm used in the system for IRIS dataset

Algorithm	Score using Decision Tree as a Classifier	Score using Logistic Regression as a Classifier
Hybrid	0.9533	0.9466
k best	0.7	0.6733
RFE recursive features selection	0.706	0.6866
chi2	0.6933	0.6866
FAST	0.9533	0.9466

The score or accuracy of the algorithm is as follows

1 For Decision Tree as a classifier

Score(Hybrid) = Score(FAST) > Score(Chi2) > Score(RFE) > Score(K best)

2 For Logistic Regression as a classifier

Score(Hybrid) = Score(FAST) > Score(RFE) = Score(Chi2) > Score(K best)

The following table represents the score with respect to each algorithm used in the system for Breast Cancer dataset

Algorithm	Score using Decision Tree as a Classifier	Score using Logistic Regression as a Classifier
Hybrid	0.8933	0.916
k best	O.8927	0.916
RFE recursive features selection	0.88	0.913
chi2	0.89	0.915
FAST	0.8927	0.901

The score or accuracy of the algorithm is as follows

1 For Decision Tree as a classifier

Score(Hybrid) > Score(FAST) > Score(K best) > Score(chi2) > Score(RFE)

2 For Logistic Regression as a classifier

Score(Hybrid) = Score(k best) > Score(chi2) > Score(RFE) > Score(FAST)

The following table represents the score with respect to each algorithm used in the system for DIGIT dataset

Algorithm	Score using Decision Tree as a Classifier	Score using Logistic Regression as a Classifier
Hybrid	0.826	0.90
k best	0.822	0.875
RFE recursive features selection	0.800	0.894
chi2	0.819	0.88

FAST	0.813	0.90

The score or accuracy of the algorithm is as follows

1 For Decision Tree as a classifier

Score(Hybrid) > Score(chi2) > Score(FAST) > Score(k best) > Score(RFE)

2 For Logistic Regression as a classifier

Score(Hybrid) > Score(FAST) > Score(RFE) > Score(chi2) > Score(kbest)

CHAPTER 8

CONCLUSION AND FUTURE WORK

The main motive of the thesis is to decrease the dimensionality of the feature set in order to decrease runtime of the system and increase accuracy. Here, a clustering based method for feature selection is proposed which incorporates a similarity measure along with it. In this method, a cluster is composed of features and instead of using all the features in a particular cluster, a representative feature is chosen. This in turn reduced the size of the feature set. DICE coefficient further increases the efficiency of the system. The algorithm is composed of two steps:

1) Developing a minimum spanning tree from the initial set of feature using a similarity measure
2) Dividing the MST repetitively to produce clusters or forests

The proposed method presented a better score than many of the existing methods like FAST, chi2, etc and also helped in the reduction of the runtime of the whole process.

For future work, we suggest to work out the practical analysis and testing of the system in order to incorporate the possibilities of further improvement.

REFERENCES

[1] D.W. Aha and R. L. Blankert. Feature selection for case-based classification of cloud types. In *Working Notes of th AAAI-94 Workshop on Case-Based Reasoning*, pages 106–112, 1994.

[2] H. Almuallim and T. G. Dietterich. Learning with many irrelevant features. In *Proceedings of the Ninth National Conference on Artificial Intelligence*, pages 547–542. MIT Press, 1991.

[3] P. Domingos and M. Pazzani. On the optimality of the simple Bayesian classifier under zero-one loss. *Machine Learning*, 29:103–130, 1997.

[4] R. Setiono and H. Liu. Chi2: Feature selection and discretization of numeric attributes. In *Proceedings of the Seventh IEEE International Conference on Tools with Artificial Intelligence*, 1995.

[5] M. Pazzani. Searching for dependencies in Bayesian classifiers. In *Proceedings of the Fifth International Workshop on AI and Statistics*, 1995.

[6] R. Caruana and D. Freitag. Greedy attribute selection. In *Machine Learning: Proceedings of the Eleventh International Conference*. Morgan Kaufmann, 1994.

[7] R. R. Quinlan. Induction of decision trees. *Machine Learning*, 1:81–106, 1986.

[8] G. M. Provan and M. Singh. Learning Bayesian networks using feature selection. In D. Fisher and H. Lenz, editors, *Learning from Data, Lecture Notes in Statistics*, pages 291–300. Springer-Verlag, New York, 1996.

[9] A. J. Miller. *Subset Selection in Regression*. Chapman and Hall, New York, 1990.

[10] A. W. Moore and M. S. Lee. Efficient algorithms for minimizing cross validation error. In *Machine Learning: Proceedings of the Eleventh International Conference*. Morgan Kaufmann, 1994.

[11] C. Cardie. Using decision trees to improve cased-based learning. In *Proceedings of the First International Conference on Knowledge Discovery and Data Mining*. AAAI Press, 1995.

[12] G. M. Provan and M. Singh. Learning Bayesian networks using feature selection. In D. Fisher and H. Lenz, editors, *Learning from Data, Lecture Notes in Statistics*, pages 291–300. Springer-Verlag, New York, 1996.
.

[13] G. Holmes and C. G. Nevill-Manning. Feature selection via the discovery of

simple classification rules. In *Proceedings of the Symposium on Intelligent Data Analysis*, Baden-Baden, Germany, 1995.

[14] S. Salzberg. A nearest hyperrectangle learning method. *Machine Learning*, 6:251–276, 1991.

[15] B. Pfahringer. Compression-based feature subset selection. In *Proceedings of the IJCAI-95 Workshop on Data Engineering for Inductive Learning*, pages 109–119, 1995.

[16] D. Koller and M. Sahami. Towards optimal feature selection. In *Machine Learning: Proceedings of the Thirteenth International Conference on Machine Learning*. Morgan Kaufmann, 1996.

[17] D. Wettschereck and D. W. Aha. Weighting features. In *First International Conference on Cased-Based Reasoning*.

[18] K. Kira and L. A. Rendell. A practical approach to feature selection. In *Machine Learning: Proceedings of the Ninth International Conference*, 1992.

[19] R. Kohavi, P. Langley, and Y. Yun. The utility of feature weighting in nearest-neighbor algorithms. In *Proceedings of the Ninth European Conference on Machine Learning*, Prague, 1997. Springer-Verlag.

[20] C. Cardie. Using decision trees to improve cased-based learning. In *Proceedings of the First International Conference on Knowledge Discovery and Data Mining*. AAAI Press, 1995.

[21] M. Scherf and W. Brauer. Feature selection by means of a feature weighting approach. Technical Report FKI-221-97, Technische Universit¨at M¨unchen, 1997.

[22] P. Langley. Selection of relevant features in machine learning. In *Proceedings of the AAAI Fall Symposium on Relevance*. AAAI Press, 1994.

[23] G. H. John, R. Kohavi, and P. Pfleger. Irrelevant features and the subset selection problem. In *Machine Learning: Proceedings of the Eleventh International Conference*. Morgan Kaufmann, 1994.

[24] K. J. Cherkauer and J. W. Shavlik. Growing simpler decision trees to facilitate knowledge discovery. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*. AAAI Press, 1996.
[25] QinBao , “A Fast Clustering-Based Feature Subset Selection Algorithm for High Dimensional Data -in "ieee transactions on knowledge and data engineering” vol:25 no:1 year 2013.

[26] H. Vafaie and K. De Jong. Genetic algorithms as a tool for restructuring feature space representations. In *Proceedings of the International Conference on Tools with A.I.* IEEE Computer Society Press, 1995.

[27] Yu L. and Liu H., "Feature selection for high-dimensional data: a fast correlation-based filter) solution", in Proceedings of 20th International Conference on Machine Leaning, 20(2), pp 856-863, 2003.

[28] P. Langley and S. Sage. Scaling to domains with irrelevant features. In R. Greiner, editor, *Computational Learning Theory and Natural Learning Systems*, volume 4. MIT Press, 1994.

[29] Yu L. and Liu H.,Efficient feature selection via analysis of relevance and redundancy,journal of machine learning research,10(5),pp 1205-1224,2004

[30] Van Dijk G. and Van Hulle M.M., "Speeding Up the Wrapper Feature Subset Selection in Regression by Mutual Information Relevance and Redundancy Analysis", International Conference on Artificial Neural Networks, 2006

[31] A. W. Moore, D. J. Hill, and M. P. Johnson. An empirical investigation of brute force to choose features, smoothers and function approximators. In S. Hanson, S. Judd, and T. Petsche, editors, *Computational Learning Theory and Natural Learning Systems*, volume 3. MIT Press, 1992.

[32] Chanda P., Cho Y., Zhang A. and Ramanathan M., Mining of Attribute Interactions Using Information Theoretic Metrics, In Proceedings of IEEE international Conference on Data Mining Workshops, pp 350-355, 2009.

[33] Forman G., An extensive empirical study of feature selection metrics for text classification, Journal of Machine Learning Research, 3, pp 1289-1305,2003.

[34] Fleuret F., Fast binary feature selection with conditional mutual Information, Journal of Machine Learning Research, 5, pp 1531- 1555, 2004.

[35] C.Krier, D.Francois, F. Rossi, and M. Verleysen, "Feature Clustering and Mutual Information for the Selection of Variables in Spectral Data," Proc. European Symp. Artificial Neural Networks Advances in Computational Intelligence and Learning, pp. 157-162, 2007.