Improving Efficiency and Accuracy of Classification and Clustering of a Text Documents with Feature Selection

[1]Ajay Singh, [2]Yudhvir Singh, [3]Rajeshwar Singh

[1]Ph. D. Research Scholar (PTU)
[2]Associate Professor, UIET, MDU (Rohtak),
[3]Principal &Director, DOABA Group of Colleges, Nawanshahr
[1]ajayjangra@gmail.com, [2]dr.yudhvirs@gmail.com

Abstract— **The quality of the data is one of the most important factors influencing the performance of any classification or clustering algorithm. The attributes defining the feature space of a given data set can often be inadequate, which make it difficult to discover interesting knowledge or desired output. However, even when the original attributes are individually inadequate, it is often possible to combine such attributes in order to construct new ones with greater predictive power. Feature selection, as a preprocessing step to machine learning, has been very effective in reducing dimensionality, removing irrelevant data, and noise from data to improving result comprehensibility. The goal of this thesis is to find out the best feature subset from the given features in order to improve the performance of classification and clustering techniques on complex, real world data. To partition a given document collection into clusters of similar documents a choice of good features along with good clustering algorithms is very important in clustering. The feature selection is an important part in automatic text categorization which can change the entire results of text clusters.**

Keywords: Feature Selection, Classification, Clustering, Machine Learning, Data Mining.

IJERT

I. INTRODUCTION

Data mining tools predict future trends and behaviors, allowing businesses to make proactive, knowledge-driven decisions. The automated, prospective analyses offered by data mining move beyond the analyses of past events provided by retrospective tools typical of decision support systems. Data mining tools can answer business questions that traditionally were too time-consuming to resolve. They scour databases for hidden patterns, finding predictive information that experts may miss because it lies outside their expectations.

As computer and database technologies advance rapidly, data accumulates in a speed unmatchable by human's capacity of data processing. Data mining [5, 22], as a multidisciplinary joint effort from databases, machine learning, and statistics, is championing in turning mountains of data into nuggets. Researchers and practitioners realize that in order to use data mining tools effectively, data preprocessing is essential to successful data mining [4]. Feature selection is one of the important and frequently used techniques in data preprocessing for data mining [11, 12]. It reduces the number of features, removes irrelevant, redundant, or noisy data, and brings the immediate effects for applications: speeding up a data mining algorithm, improving mining performance such as predictive accuracy and result comprehensibility. Feature selection has been a fertile field of research and development since 1970's in statistical pattern recognition [3], machine learning [12], and data mining [20], and widely applied to many fields such as text categorization [6].

Filter algorithms described in the machine learning literature have exhibited a number of drawbacks. Some algorithms do not handle noise in data, and others require that the level of noise be roughly specified by the user a-priori. In some cases, a subset of features is not selected explicitly; instead, features are ranked with the final choice left to the user. Unfortunately, as the amount of machine readable information increases, the ability to understand and make use of it does not keep pace with its growth. Machine learning provides tools by which large quantities of data can be automatically analyzed.

The goal of the feature selection process is, given a dataset that describes a target concept using n attributes, to find the minimum number m of relevant attributes which describe the concept as well as the original set of attributes do. As an example consider a dataset containing information of customers that applied for a credit to a bank. The concept, i.e. the class attribute, is represented by the risk level, low or high, assigned to each customer by a credit manager of the bank. Attributes represent the customer current credit situation, the past credit history and other general information. Thus the data, corresponding to each customer, can be regarded as examples of how the risk level should be assigned to a customer. Effective feature selection, by enabling generalization algorithms to focus on the best subset of useful features, substantially increases the likelihood of obtaining simpler, more understandable and predictive models of the data. There are two common approaches: wrapper and filter. A wrapper uses the intended learning algorithm itself to evaluate the usefulness of features while a filter evaluates features according to heuristics based on general characteristics of the data. The wrapper approach is generally considered to produce better feature subsets but runs much more slowly than a filter. Filters do not require re-execution for different learning algorithms. Filters can provide the same benefits for learning as wrappers do. If improved accuracy for a particular learning algorithm is required, a filter can provide an intelligent starting feature subset for a wrapper, a process that is likely to result in a shorter, and hence faster, search for the wrapper. In a related scenario, a wrapper might be applied to search the filtered feature space that is, the reduced feature space provided by a filter. Both methods help scale the wrapper to larger datasets. For these reasons, a filter approach to feature selection for machine learning is explored.

The rest of this paper is organized as follow: section 2 describes the feature selection and machine learning. Section 3 describes the subset generation, Subset evaluation, stopping criteria and result validation. Section 4 describes a categorizing framework for feature selection algorithms. Section 5 describes the software tool and results. Section 6 concludes the paper.

IJERT

II. FEATURE SELECTION AND MACHINE LEARNING

Feature selection is a process that selects a subset of original features. The optimality of a feature subset is measured by an evaluation criterion. As the dimensionality of a domain expands, the number of features N increases. Finding an optimal feature subset is usually intractable [13] and many problems related to feature selection have been shown to be NP-hard [7]. A typical feature selection process consists of four basic steps, namely, subset generation, subset evaluation, stopping criterion, and result validation. Subset generation is a search procedure that produces candidate feature subsets for evaluation based on a certain search strategy. Feature selection can be found in many areas of data mining such as classification, clustering, association rules, regression. For example, feature selection is called subset or variable selection in Statistics [14]. A number of approaches to variable selection and coefficient shrinkage for regression are summarized in [21].

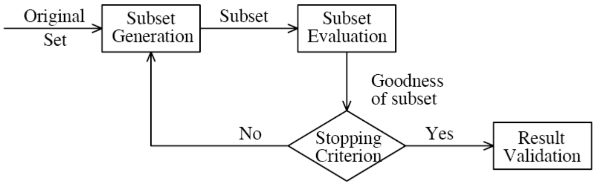


Figure 1: Four key steps of feature selection

Feature selection algorithms designed with different evaluation criteria broadly fall into three categories: the filter model [20], the wrapper model [13], and the hybrid model [15]. The filter model relies on general characteristics of the data to evaluate and select feature subsets without involving any mining algorithm. The wrapper model requires one predetermined mining algorithm and uses its performance as the evaluation criterion. It searches for features better suited to the mining algorithm aiming to improve mining performance, but it also tends to be more computationally expensive than the filter model [13]. The hybrid model attempts to take advantage of the two models by exploiting their different evaluation criteria in different search stages.

Machine learning is the study of algorithms that automatically improve their performance with experience. At the heart of performance is prediction. Machine learning algorithms can be broadly characterized by the language used to represent learned knowledge. No single learning approach is clearly superior in all cases, and in fact, different learning algorithms often produce similar results. One factor that can have an enormous impact on the success of a learning algorithm is the nature of the data used to characterize the task to be learned. If the data fails to exhibit the statistical regularity that machine learning algorithms exploit, then learning will fail. It is possible that new data may be constructed from the old in such a way as to exhibit statistical regularity and facilitate learning, but the complexity of this task is such that a fully automatic method is intractable.

III. SUBSET GENERATION, SUBSET EVALUATION, STOPPING CRITERIA AND RESULT VALIDATION

Subset generation - If variable elimination has not been sorted out after two decades of work assisted by high-speed computing, then perhaps the time has come to move on to other problems [2].

Subset generation is essentially a process of heuristic search, with each state in the search space specifying a candidate subset for evaluation. The nature of this process is determined by two basic issues. First, one must decide the search starting point (or points) which in turn influences the search direction. Search may start with an empty set and successively add features (i.e., forward), or start with a full set and successively remove features (i.e., backward), or start with both ends and add and remove features simultaneously (i.e., bi-directional). Search may also start with a randomly selected subset in order to avoid being trapped into local optima [18]. Second, one must decide a search strategy. For a data set with N features, there exist 2N candidate subsets. This search space is exponentially prohibitive for exhaustive search with even a moderate N. Therefore, different strategies have been explored: complete, sequential, and random search.

Complete search

It guarantees to find the optimal result according to the evaluation criterion used. Exhaustive search is complete (i.e., no optimal subset is missed). However, search is complete does not necessarily means that it must be exhaustive. Different heuristic functions can be used to reduce the search space without jeopardizing the chances of finding the optimal result. Hence, although the order of the search space is $O(2^N)$, a smaller number of subsets are evaluated. Some examples are branch and bound [10], and beam search [18].

Sequential search

It gives up completeness and thus risks losing optimal subsets. There are many variations to the greedy hill-climbing approach, such as sequential forward selection, sequential backward elimination, and bi-directional selection [4]. All these approaches add or remove features one at a time. Another alternative is to add (or remove) p features in one step and remove (or add) q features in the next step ($p > q$) [18]. Algorithms with sequential search are simple to implement and fast in producing results as the order of the search space is usually $O(N^2)$ or less.

Random search

It starts with a randomly selected subset and proceeds in two different ways. One is to follow sequential search, which injects randomness into the above classical sequential approaches. Examples are random-start hill-climbing and simulated annealing [18]. The other is to generate the next subset in a completely random manner (i.e., a current subset does not grow or shrink from any previous subset following a deterministic rule), also known as the Las Vegas algorithm [9]. For all these approaches, the use of randomness helps to escape local optima in the search space, and optimality of the selected subset depends on the resources available.

Subset evaluation - As we mentioned earlier, each newly generated subset needs to be evaluated by an evaluation criterion. The goodness of a subset is always determined by a certain criterion (i.e., an optimal subset selected using one criterion may not be optimal according to another criterion). Evaluation criteria can be broadly categorized into two groups based on their dependency on mining algorithms that will finally be applied on the selected feature subset. We discuss the two groups of evaluation criteria below.

Independent criteria

Dependency criteria

1. Independent criteria - Typically, an independent criterion is used in algorithms of the filter model. It tries to evaluate the goodness of a feature or feature subset by exploiting the intrinsic characteristics of the training data without involving any mining algorithm. Some popular independent criteria are distance measures, information measures, dependency measures, and consistency measures [3, 4].

Distance measures are also known as separability, divergence, or discrimination measures. For a two-class problem, a feature X is preferred to another feature Y if X induces a greater difference between the two-class conditional probabilities than Y, because we try to find the feature that can separate the two classes as far as possible. X and Y are indistinguishable if the difference is zero.

Information measures typically determine the information gain from a feature. The information gain from a feature X is defined as the difference between the prior uncertainty and expected posterior uncertainty using X. Feature X is preferred to feature Y if the information gain from X is greater than that from Y.

Dependency measures are also known as correlation measures or similarity measures. They measure the ability to predict the value of one variable from the value of another. In feature selection for classification, we look for how strongly a feature is associated with the class. A feature X is preferred to another feature Y if the association between feature X and class C is higher than the association between Y and C. In feature selection for clustering, the association between two random features measures the similarity between the two.

Consistency measures are characteristically different from the above measures because of their heavy reliance on the class information and the use of the Min-Features bias [19] in selecting a subset of features. These measures attempt to find a minimum number of features that separate classes as consistently as the full set of features can. An inconsistency is defined as two instances having the same feature values but different class labels.

2. Dependency criteria - A dependency criterion used in the wrapper model requires a predetermined mining algorithm in feature selection and uses the performance of the mining algorithm applied on the selected subset to determine which features are selected. It usually gives superior performance as it finds features better suited to the predetermined mining algorithm, but it also tends to be more computationally expensive, and may not be suitable for other mining algorithms [12]. For example, in a task of classification, predictive accuracy is widely used as the primary measure. It can be used as a dependent criterion for feature selection. As features are selected by the classifier that later on uses these selected features in predicting the class labels of unseen instances, accuracy is normally high, but it is computationally rather costly to estimate accuracy for every feature subset [16].

In a task of clustering, the wrapper model of feature selection tries to evaluate the goodness of a feature subset by the quality of the clusters resulted from applying the clustering algorithm on the selected subset. There exist a number of heuristic criteria for estimating the quality of clustering results, such as cluster compactness, scatter

IJERT

separability, and maximum likelihood. Recent work on developing dependent criteria in feature selection for clustering can been found in [16].

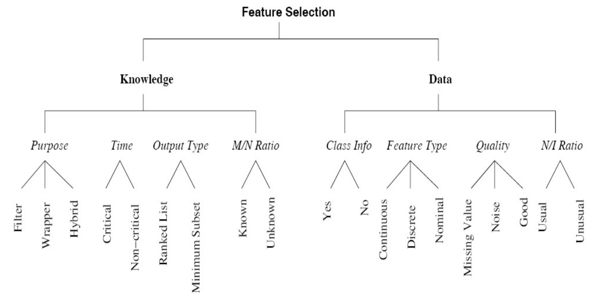
Stopping criteria - A stopping criterion determines when the feature selection process should stop. Some frequently used stopping criteria are: (a) the search completes; (b) some given bound is reached, where a bound can be a specified number (minimum number of features or maximum number of iterations); (c) subsequent addition (or deletion) of any feature does not produce a better subset; and (d) a sufficiently good subset is selected (e.g., a subset may be sufficiently good if its classification error rate is less than the allowable error rate for a given task).

Result validation - A straightforward way for result validation is to directly measure the result using prior knowledge about the data. If we know the relevant features beforehand as in the case of synthetic data, we can compare this known set of features with the selected features. Knowledge on the irrelevant or redundant features can also help. We do not expect them to be selected. In real-world applications, however, we usually do not have such prior knowledge. Hence, we have to rely on some indirect methods by monitoring the change of mining performance with the change of features. For example, if we use classification error rate as a performance indicator for a mining task, for a selected feature subset, we can simply conduct the "before-and-after" experiment to compare the error rate of the classifier learned on the full set of features and that learned on the selected subset [4].

IV. A CATEGORIZING FRAMEWORK FOR FEATURE SELECTION ALGORITHMS

There exists a vast body of available feature selection algorithms. In order to better understand the inner instrument of each algorithm and the commonalities and differences among them, we develop a three-dimensional categorizing framework (shown in Table 1) based on the previous discussions. We understand that search strategies and evaluation criteria are two dominating factors in designing a feature selection algorithm, so they are chosen as two dimensions in the framework. In Table 1, under Search Strategies, algorithms are categorized into Complete, Sequential, and Random. Under Evaluation Criteria, algorithms are categorized into Filter, Wrapper, and Hybrid.

We consider Data Mining Tasks as a third dimension because the availability of class information in Classification or Clustering tasks affects evaluation criteria used in feature selection algorithms. In addition to these three basic dimensions, algorithms within the Filter category are further distinguished by specific evaluation criteria including Distance, Information, Dependency, and Consistency. Within the Wrapper category, Predictive Accuracy is used for Classification, and Cluster Goodness for Clustering.



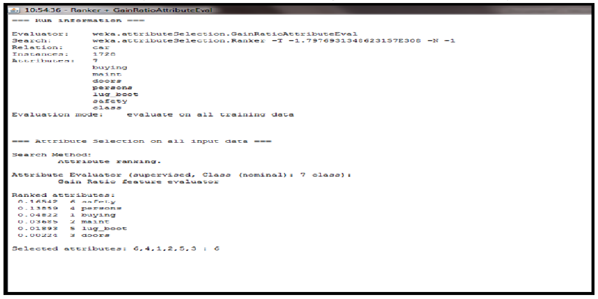
Filter Algorithm -Algorithms within the filter model are illustrated through a generalized filter algorithm. For a given data set D, the algorithm starts the search from a given subset S0 (an empty set, a full set, or any randomly selected subset) and searches through the feature space by a particular search strategy. Each generated subset S is evaluated by an independent measure M and compared with the previous best one. If it is found to be better, it is regarded as the current best subset. The search iterates until a predefined stopping criterion δ is reached. The algorithm outputs the last current best subset S_{best} as the final result. By varying the search strategies and evaluation measures used in steps 5 and 6 in the algorithm, we can design different individual algorithms within the filter model. Since the filter model applies independent evaluation criteria without involving any mining algorithm, it does not inherit any bias of a mining algorithm and it is also computationally efficient.

IJERT

V. SOFTWARE TOOL AND RESULT

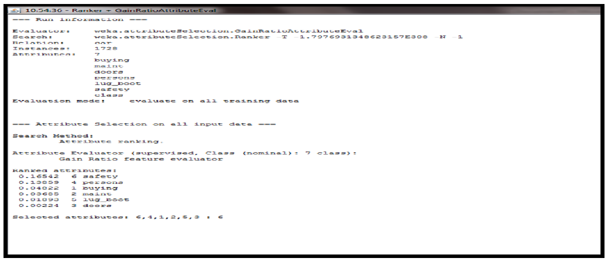
Weka is a collection of machine learning algorithms for data mining tasks. The algorithms can either be applied directly to a dataset or called from your own Java code. Weka contains tools for data pre-processing, classification, regression, clustering, association rules, and visualization. It is also well-suited for developing new machine learning schemes [1].

The essence of these successful applications lies at the recognition of a need for effective data preprocessing: data mining can be effectively accomplished with the aid of feature selection. Data is often collected for many reasons other than data mining (e.g., required by law, easy to collect, or simply for the purpose of book-keeping). In real-world applications, one often encounters problems such as too many features, individual features unable to independently capture significant characteristics of data, high dependency among the individual features, and emergent behaviors of combined features. Humans are ineffective at formulating and understanding hypotheses when data sets have large numbers of variables (possibly thousands in cases involving demographics and hundreds of thousands in cases involving Web browsing, microarray data analysis, or text document analysis), and people would find it easy to understand aspects of the problem in lower-dimensional subspaces. Feature selection can reduce the dimensionality to enable many data mining algorithms to work effectively on data with large dimensionality.



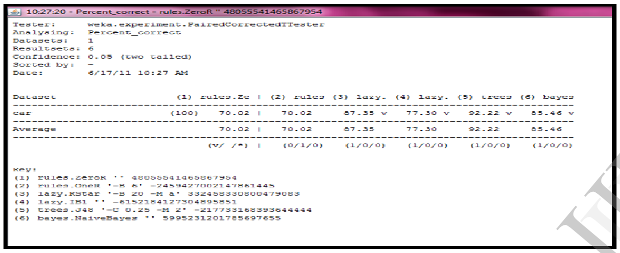
```
=== Run information ===

Evaluator:    weka.attributeSelection.GainRatioAttributeEval
Search:       weka.attributeSelection.Ranker -T -1.7976931348623157E308 -N -1
Relation:     car
Instances:    1728
Attributes:   7
              buying
              maint
              doors
              persons
              lug_boot
              safety
              class
Evaluation mode:    evaluate on all training data


=== Attribute Selection on all input data ===

Search Method:
        Attribute ranking.

Attribute Evaluator (supervised, Class (nominal): 7 class):
        Gain Ratio feature evaluator

Ranked attributes:
 0.16542   6 safety
 0.13859   4 persons
 0.04822   1 buying
 0.03685   2 maint
 0.01893   5 lug_boot
 0.00224   3 doors

Selected attributes: 6,4,1,2,5,3 : 6
```

Output of attribute selection using Gain Ratio AttributeEval and Ranker



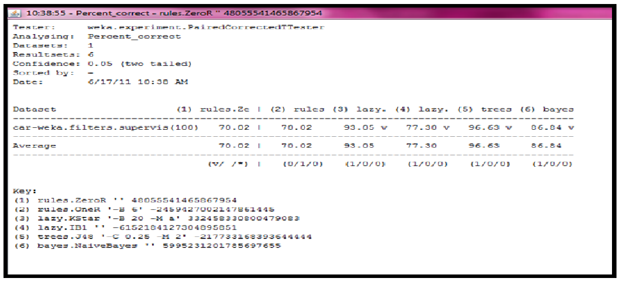
```
=== Run information ===

Evaluator:    weka.attributeSelection.GainRatioAttributeEval
Search:       weka.attributeSelection.Ranker -T -1.7976931348623157E308 -N -1
Relation:     car
Instances:    1728
Attributes:   7
              buying
              maint
              doors
              persons
              lug_boot
              safety
              class
Evaluation mode:    evaluate on all training data


=== Attribute Selection on all input data ===

Search Method:
        Attribute ranking.

Attribute Evaluator (supervised, Class (nominal): 7 class):
        Gain Ratio feature evaluator

Ranked attributes:
 0.16542   6 safety
 0.13859   4 persons
 0.04822   1 buying
 0.03685   2 maint
 0.01893   5 lug_boot
 0.00224   3 doors

Selected attributes: 6,4,1,2,5,3 : 6
```

Output of attribute selection using Greedystepwise & Filtered Subset Eval


10:27:20 - Percent_correct - rules.ZeroR '' 48055541465867954

```
Tester:     weka.experiment.PairedCorrectedTTester
Analysing:  Percent_correct
Datasets:   1
Resultsets: 6
Confidence: 0.05 (two tailed)
Sorted by:  -
Date:       6/17/11 10:27 AM

Dataset                  (1) rules.Ze | (2) rules (3) lazy. (4) lazy. (5) trees (6) bayes
-----------------------------------------------------------------------------------------
car                     (100)   70.02 |   70.02     87.35 v   77.30 v   92.22 v   85.46 v
-----------------------------------------------------------------------------------------
Average                         70.02 |   70.02     87.35     77.30     92.22     85.46
-----------------------------------------------------------------------------------------
                              (v/ /*) |   (0/1/0)   (1/0/0)   (1/0/0)   (1/0/0)   (1/0/0)


Key:
(1) rules.ZeroR '' 48055541465867954
(2) rules.OneR '-B 6' -2459427002147861445
(3) lazy.KStar '-B 20 -M a' 332458330800479083
(4) lazy.IB1 '' -6152184127304895851
(5) trees.J48 '-C 0.25 -M 2' -217733168393644444
(6) bayes.NaiveBayes '' 5995231201785697655
```

Results without Feature Selection


10:38:55 - Percent_correct - rules.ZeroR '' 48055541465867954

```
Tester:     weka.experiment.PairedCorrectedTTester
Analysing:  Percent_correct
Datasets:   1
Resultsets: 6
Confidence: 0.05 (two tailed)
Sorted by:  -
Date:       6/17/11 10:38 AM

Dataset                  (1) rules.Ze | (2) rules (3) lazy. (4) lazy. (5) trees (6) bayes
-----------------------------------------------------------------------------------------
car-weka.filters.supervis(100)  70.02 |   70.02     93.05 v   77.30 v   96.63 v   86.84 v
-----------------------------------------------------------------------------------------
Average                         70.02 |   70.02     93.05     77.30     96.63     86.84
-----------------------------------------------------------------------------------------
                              (v/ /*) |   (0/1/0)   (1/0/0)   (1/0/0)   (1/0/0)   (1/0/0)


Key:
(1) rules.ZeroR '' 48055541465867954
(2) rules.OneR '-B 6' -2459427002147861445
(3) lazy.KStar '-B 20 -M a' 332458330800479083
(4) lazy.IB1 '' -6152184127304895851
(5) trees.J48 '-C 0.25 -M 2' -217733168393644444
(6) bayes.NaiveBayes '' 5995231201785697655
```

Results with Feature Selection

VI. CONCLUSION

The recent developments in various methods used for feature selection have addressed the problem from the pragmatic point of view of improving the performance of textual data. There is a challenge in operating an input spaces of several thousand variables. In this thesis an analysis on the feature selection methods is carried, and implementation of an arrf file in Weka is done. The major conclusions after going through the literature review, analysis and experimentation is that choice of a good feature can contribute a lot to the classification and clustering the text documents. A comparative study of various classification methods is also done, through calculating the accuracy of all methods using Weka. Classification of data is done in two ways, without using feature selection and with using feature selection and comparative results have been studied. Through these results it can be conclude that the accuracy of classification is degraded if the appropriate features are removed by the feature selection methods.

VII REFERENCES

[1] Weka 3: Data Mining Software in Java http://www.cs.waikato.ac.nz/ml/weka/

[2] A.J. Miller, Selection of subsets of regression variables, J. Roy. Statist. Sot. A 147 (1984) 389-425.

[3] M. Ben-Bassat. Pattern recognition and reduction of dimensionality. In P. R. Krishnaiah and L. N. Kanal, editors, Handbook of statistics-II, 1982, pages 773–791. North Holland.

[4] H. Liu and H. Motoda. Feature Selection for Knowledge Discovery and Data Mining. Boston: Kluwer Academic Publishers, 1998.

[5] R. Agrawal, T. Imielinski, and A. Swami. Database mining: A performance perspective. IEEE Transactions on Knowledge and Data Engineering, 1993, 5(6):914–925.

[6] E. Leopold and Kindermann J. Text categorization with support vector machines. how to represent texts in input space? Machine Learning, 2002, 46:423–444.

[7] A.L. Blum and R.L. Rivest. Training a 3-node neural networks is NP-complete. Neural Networks, 1992, 5:117 – 127.

[8] U.M. Fayyad, G. Piatetsky-Shapiro, and P. Smyth. From data mining to knowledge iscovery: An overview. In U.M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, editors, Advances in Knowledge Discovery and Data Mining, 1996, pages 495–515. AAAI Press / The MIT Press.

[9] G. Brassard and P. Bratley. Fundamentals of Algorithms. Prentice Hall, New Jersey, 1996.

[10] P.M. Narendra and K. Fukunaga. A branch and bound algorithm for feature subset selection. IEEE Trans. on Computer, 1977, C-26(9):917–922.

[11] H. Liu and H. Motoda, editors. Feature Extraction, Construction and Selection: A Data Mining Perspective. Boston: Kluwer Academic Publishers, 1998. 2nd Printing, 2001.

[12] A.L. Blum and P. Langley. Selection of relevant features and examples in machine learning. Artificial Intelligence, 1997, 97:245–271.

[13] R. Kohavi and G.H. John. Wrappers for feature subset selection. Artificial Intelligence, 1997, 97(1-2):273–324.

[14] A. Miller. Subset Selection in Regression. Chapman & Hall/CRC, 2 edition, 2002.

[15] S. Das. Filters, wrappers and a boosting-based hybrid for feature selection. In Proceedings of the Eighteenth International Conference on Machine Learning, 2001, pages 74–81.

[16] M. Dash and H. Liu. Feature selection for clustering. In Proceedings of the Fourth Pacific Asia Conference on Knowledge Discovery and Data Mining, (PAKDD-2000), 2000, pages 110–121.

[17] G.H. John, R. Kohavi, and K. Pfleger. Irrelevant feature and the subset selection problem. In Proceedings of the Eleventh International Conference on Machine Learning, 1994, pages 121–129.

[18] J. Doak. An evaluation of feature selection methods and their application to computer security. Technical report, Davis CA: University of California, Department of Computer Science, 1992.

[19] H. Almuallim and T.G. Dietterich. Learning boolean concepts in the presence of many irrelevant features. Artificial Intelligence, 1994, 69(1-2):279–305.

IJERT

[20] M. Dash, K. Choi, P. Scheuermann, and H. Liu. Feature selection for clustering – a filter solution. In Proceedings of the Second International Conference on Data Mining, 2002, pages 115–122.

[21] T. Hastie, R. Tibshirani, and J. Friedman. The Elements of Statistical Learning. Springer, 2001.

[22] H. Almuallim and T.G. Dietterich. Learning with many irrelevant features. In Proceedings of the Ninth National Conference on Artificial Intelligence, 1991, pages 547–552.

IJERT