A Review Of Fast Clustering-Based Feature Subset Selection Algorithm

Pawan Gupta, Susheel Jain, Anurag Jain

Abstract: In this paper we cover some reference paper and compared different algorithm on the basis of their performance and selection of data set. Where the efficiency concerns on the time evaluation of features selection, and the effectiveness is related to the quality of the subset of features selection. We analysis this report based on feature subset selection algorithm from the years of 1997 to 2013 and summaries the result of data.

Keyword: Feature subset selection, filter method, and feature clustering.

I. INTRODUCTION

The feature selection algorithm may be seen as the combination of a search technique and with an evaluation measure which scores the different feature subsets. The simplest algorithm is that algorithm to test each possible subset of features finding and minimizes the error rate. This is an exhaustive search of the space, and is computationally intractable for all but the smallest of feature sets. The choice of evaluation metric heavily influences the algorithm, and it is these evaluation metrics which distinguish between the three main categories of feature selection algorithms: wrappers, filters and embedded methods [1]. The criterion function guiding the search for the best features is usually some kind of separability measure between classes. It can be either classifier independent (i.e., filter approach) or classifier specific (i.e., wrapper approach or embedded method). Wrapper methods use as a predictive model for score feature subsets selection. The wrapper methods train by a new model for each subset, they are very computationally intensive, but this method provides the best performing feature subset for that particular type of model [6]. Filter approaches methods are use a proxy measure instead of the error rate to score a feature subset. The Filter methods are usually less computationally intensive than wrappers, but they produce a feature set which is not tuned to a specific type of predictive model. Many filters methods provide a feature ranking rather than an explicit of best feature subset selection, and the cut - off point in the ranking is chosen by cross-validation [7]. The Embedded methods are catch the all group of these techniques which perform feature selection as part of the model construction process. In information, the most popular form of feature selection is stepwise failure. It is a insatiable algorithm that adds the best feature at each round. In machine learning, this is typically done by cross-validation.

- *Pawan Gupta, Susheel Jain, Anurag Jain*
- *1,2,3Computer Science Department, Radharaman Institute of Technology and Science, Bhopal (M.P.), India*
- *Email: 2007pawan@gmail.com, jain_susheel65@yahoo.co.in, anurag.akjain@gmail.com*

II. RELATED WORK

Generally Feature subset selection can be viewed as the process of identifying and removing as many irrelevant and redundant features as well as possible. This is because firstly irrelevant features that do not contribute predictive accuracy, and secondly redundant features that do not redound to getting a better predictor for that they provide mostly information which is already present in other feature. Traditionally, the feature subset selection research has focused on searching for relevant features. A Relief method is example of this, which weighs each feature according to its ability to discriminate instances under different targets based on distance-based criteria function. However, Relief is ineffective at removing redundant features as two predictive but highly correlated features are likely both to be highly weighted. CFS and FCBF [20], are examples that take into consideration the redundant features. CFS is achieved by the hypothesis that a good feature subset is one that contains features highly correlated with the target, yet uncorrelated with each other. FCBF [20] is a fast filter method which can identify relevant features as well as redundancy among relevant features without pair wise correlation analysis. The FAST algorithm [19] employs the clustering-based method to choose features. Recently, hierarchical clustering has been adopted in word selection in the context of text classification. As distributional clustering of words are agglomerative in nature, and result in suboptimal word clusters and high computational cost. Dhillon et al. [23] proposed a new information-theoretic divisive algorithm for word clustering and applied it to text classification. Butterworth et al. [24] proposed to cluster features using a special metric of Barthelemy-Montjardet distance, and then makes use of the dendrogram of the resulting cluster hierarchy to choose the most relevant attributes. Unfortunately, the cluster evaluation measure based on Barthelemy-Montjardet distance does not identify a feature subset that allows the classifiers to improve their original performance accuracy. Krier et al. [21] presented a methodology combining hierarchical constrained clustering of spectral variables and selection of clusters by mutual information. Their feature clustering method is similar to that of Van Dijck and Van Hulle [22] except that the former forces every cluster to contain consecutive features only. Both methods employed agglomerative hierarchical clustering to remove redundant features. Quite different from these hierarchical clustering-based algorithms and FAST algorithm uses minimum spanning tree-based method to cluster features. While, it does not assume that data points are grouped around centers or separated by a regular geometric curve.

III. FETURE SUBSET SELECTION ALGORITHM

Feature subset selection is a long existing technique to deal with problems brought by too many features [1]. A feature subset selection method is usually made up of two parts: a feature subset generator and an evaluator. The two parts work together to find the feature subset which meets evaluation criteria best. Feature subset generator can also be seen as a search engine, which can be divided into three categories: exhaustive search engine, heuristic search engine and nondeterministic search engine [1]. These search engines search in the state space using different search strategies. According to supervised feature selection methods, feature selection can be defined as follows:

Definition 1: (Supervised feature selection). Given a set of features F = {f_1... f_i… f_n}, find a subset $F' \subseteq F$ that maximizes the ability of the learning algorithm to classify patterns. Formally, F' should maximize some scoring function Θ, such that $F' = \arg\max_{G \in \Gamma} \{\Theta(G)\} F$, where Γ is the space of all possible feature subsets of F. Throughout the ([25, 26, 27, and 29]), feature subset selection approaches are categorized into three main groups: filter methods, wrapper methods and embedded approaches. Filter methods rely on general characteristics of the training data to estimate and select subsets of features without involving a learning algorithm. Contrary to that, wrapper approaches use a classification algorithm as a black box to assess the prediction accuracy of various subsets. The last group, embedded approaches, performs the feature selection process as an integral part of the machine learning algorithm. In the following, these three techniques are described in detail. The overview of these three approaches is given in Figure 1

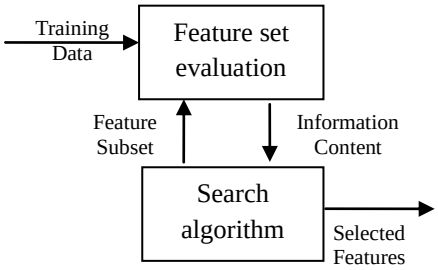

Figure 1.1: Basic feature (Filter) selection principles.

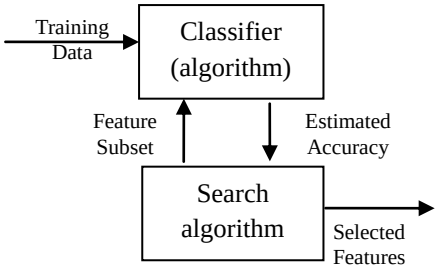

Figure 1.2: Basic feature (Wrapper) selection principles.

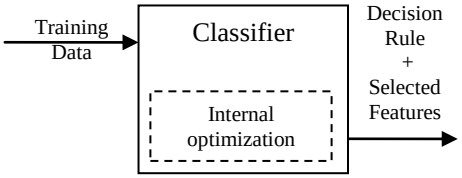

Figure 1.3: Basic (embedded) feature selection principles.

(a) Filter Methods

Filter methods are classifier agnostic, no-feedback, pre-selection methods that are independent of the machine learning algorithm to be applied. Following the classification of [29] (which slightly differs from the classification in [25] and [26), filter methods can further be divided into univariate and multivariate techniques. Univariate filter models consider one feature at a time, while multivariate methods consider subsets of features together, aiming at incorporating feature dependencies. By Guyon and Elissee [26], univariate filter methods are referred to as single variable classifiers, and multivariate filter methods are grouped together with wrapper methods and embedded methods and referred to as variable subset selection methods.

Univariate filter methods. These methods consider features separately and usually make use of some scoring function to assign weights to features individually and rank them based on their relevance to the target concept [27].In the literature, this procedure is commonly known as feature ranking or feature weighting. A feature will be selected if its weight or relevance is greater than some threshold value.

Definition 2:

(Feature ranking). Given a set of features F = {f_1,.. f_i,…. f_n}, order the features by the value of some individual scoring function S(f). If S(fi) is greater than a threshold value t, feature f_i is added to the new feature subset F'. Various univariate filter methods exist, such as the family of instance-based RelieF algorithms, or statistical approaches such as Pearson's correlation, linear regression or statistics. RelieF (recursive elimination of features) algorithms are able to detect conditional dependencies of features between instances. They evaluate the value of an attribute by repeatedly sampling an instance and considering the value of the given attribute for the nearest instance of the same and different class. Most Relief algorithms can operate on both discrete and continuous class data. A theoretical and empirical analysis of several ReliefF algorithms can be found in [30]. Besides instance-based and statistical approaches, there are several univariate filter methods that use information theoretic criteria for variable selection, such as information gain (also called Kullback-Leibler divergence) and gain ratio, both described in [31] and summarized brief in the following.

Information gain. One option for ranking features of a given dataset according to how well they differentiate the classes of objects is to use their information gain (IG), which is also used to compute splitting criteria for some decision tree algorithms (C4.5 algorithm). Information gain is based on entropy, an information-theoretic measure of the “uncertainty” contained in a training set, due to more than one possible classification [31]. Given a discrete explanatory attribute x with respect to

the (also discrete) class (or target) attribute y, the uncertainty about the value of y is defined as the overall entropy

$$H(y) = -\sum_{i=1}^{k} P(y = y_i) \log_2 (P(y = y_i))$$

The conditional entropy of y given x is then defined as

$$H(y/x) = -\sum_{j=1}^{l} P(x = x_i) H(y/x = x_i)$$

The reduction in entropy (“uncertainty”) or the gain in information" of each attribute x is then computed as

$$I\,G(y;x) = H(y) - H(y/x)$$

Gain ratio. Information gain favors features which assume many different values. Since this property of a feature is not necessarily connected with the splitting information of a feature, features can also be ranked based on their information gain ratio (GR), which normalizes the information gain and is defined as

$$G\,R(y;x) = I\,G(y;x) / splitin\ fo(x)$$

Where

$$splitin\ fo(x) = -\sum_{i=1}^{l} P(x = x_i) \log_2 P(x = x_i)$$

Although feature ranking may not be optimal, it may be preferable to other feature decrease methods because of its computational and numerical scalability. Feature ranking is computationally efficient, because it requires only the computation of scores and sorting the scores, and statistically robust, because it is known to avoid over fitting due to its low variance compared to multivariate feature selection methods.

Multivariate filter methods. Members of this second group of filter methods (also referred to as subset search evaluation) search through applicant feature subsets guided by a certain evaluation measure which captures the quality of each subset (not only the individual predictive power of single features). Contrary to wrapper methods, multivariate filter methods do not rely on a specific learning algorithm. Instead, consistency measures or correlation measures are often used to find a minimum number of features that are able to separate classes as good as the full set of features. An overview over several subset search evaluation methods can be found in [33]. Subset search evaluation can be defined as follows:

Definition 3: (Subset search evaluation). Given a set of features F = {f_1,.. f_i,…. f_n}, find a feature subset F' such that the classification accuracy based on the feature subset F' is maximized. The “best” subset search evaluation approach to find the optimal subset would be to try all possible feature subsets as input for the classification algorithm and choose that subset which leads to the best results (in terms of classification accuracy).Since the number of possible attribute subsets grows exponentially with the number of attributes this exhaustive search method is impractical for all but simple toy problems. To illustrate this, 'Figure 2' shows all 16 (24) possible feature subsets for a very simple four-feature problem. Generally, the size of the search space for n features is 2n. Hence, most multivariate filter methods as well as most wrapper methods use heuristic search methods to balance the quality of the subset and the cost of finding it. Famous examples of multivariate filter methods are correlation-based feature selection (CBFS) and its extension, fast CBFS [34]. Generally

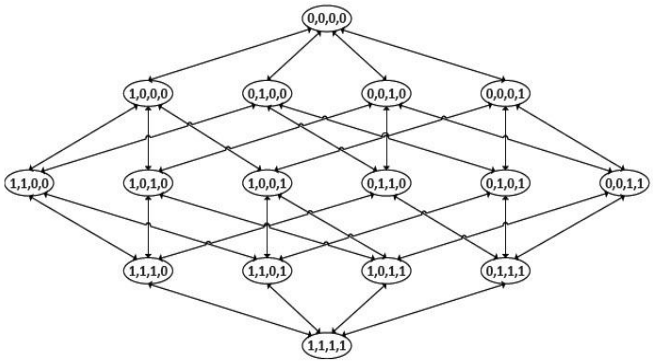

Figure 2: All 2n possible feature subsets for a simple four-feature problem [25], \1" and \0" denotes the appearance/absence of a feature in a subset, respectively.

From figure each node is connected to nodes that is it have one feature deleted or added. Speaking, the idea behind CBFS is that a feature is important and relevant to the class concept but it is not redundant to any of the other relevant features. Applied with correlation, the goodness of a feature is measured, i.e., it is measured whether a feature is highly correlated with the class but not highly correlated with any of the other features. In Fleuret's approach, conditional mutual information is used to catch dependencies among features. The author states that by selection features which maximize their mutual information with the class to predict conditional to any feature which is already picked. Compared to feature ranking methods, multivariate filter algorithms are able to identify and remove redundancies between features. Concerning the computational cost, since multivariate filter algorithms need to measure the quality of a possibly large number of candidate feature subsets instead of the quality of single features, they are less scalable and slower than univariate techniques ([26], [29]).However, they still have a better computational complexity than wrapper methods.

(b) Wrapper Methods

Wrapper methods [25] are feedback methods which incorporate the machine learning algorithm in the feature selection process, i.e., they rely on the performance of a specific classifier. Here, the classification algorithm is used as a black box [25]. Wrapper methods search through the space of feature subsets using a learning algorithm to guide the search. To search the space of different feature subsets, a search algorithm is “wrapped” around the classification model. In a search procedure the space of possible feature subsets is defined and generated various subsets of features, and estimated classification accuracy of the learning algorithm for each feature subset. Generally, wrapper methods can be divided into two groups, deterministic and randomized methods.

Deterministic wrapper methods. These methods search through the space of available feature either forwards or backwards. In the forward selection process, single attributes are added to an initially empty set of attributes. At each step,

variables that are not already in the model are tested for inclusion in the model and the most significant variables (e. g., the variable that increases the classification accuracy based on the feature subset the most) are added to the set of attributes. Li and Yang [28] have compared multiple classifiers with FS in recursive and non-recursive settings and showed that the ability of a classifier for penalizing redundant features and promoting independent features in the recursive process has a strong inuence on its success.

Randomized (non-deterministic) wrapper methods. Compared to deterministic wrapper methods, randomized wrapper algorithms search the next feature subset partly at random (i.e., the current subset does not directly grow or shrink from any previous set following a deterministic rule). Single features or several features at once can be added, removed, or replaced from the previous feature set. Famous representatives of randomized wrapper methods are genetic algorithms, stochastic search methods which are inspired by evolutionary biology and use techniques encouraged from evolutionary processes such as mutation, crossover, selection and inheritance to select a feature subset. The interaction of wrapper methods with the classification algorithm often results in better classification accuracy of the selected subsets than the accuracy achieved with filter methods ([25, 26, 29]). Nevertheless, the selected subsets are classifier dependent and have a high risk of over fitting (29). Like multivariate filter models, wrapper methods model feature dependencies, but are computationally more expensive. Furthermore, deterministic models are prone to getting stuck in a local optimum (greedy search). Compared to deterministic wrapper methods, randomized methods are less prone to getting stuck in a local optimum, but have a higher risk of over fitting [29]. While some studies report that non-deterministic wrapper approaches are usually faster than deterministic approaches, the survey in [29] states that randomized algorithms are sometimes slower than deterministic algorithms.

(c) Embedded Approaches

Embedded approaches [35], sometimes also referred to as nested subset methods [26], act as an integral part of the machine learning algorithm itself. During the operation of the classification process, the algorithm itself decides which attributes to use and which to ignore. Just like wrapper methods, embedded approaches thus depend on a specific learning algorithm, but may be more efficient in several aspects. Since they avoid retraining of the classifier from scratch for every feature subset investigated, they are usually faster than wrapper methods. Moreover, they make better use of the available data by not needing to split the training data into a training and test/validation set. Decision trees are famous examples that use embedded feature selection by selecting the attribute that achieves the “best” split in terms of class distribution at each leave. This procedure is repeated recursively on the feature subsets until some stopping criterion is satisfied.

(d) Comparison of Filter, Wrapper and Embedded Approaches

Wrapper methods usually have the highest computational complexity, followed by embedded approaches. Multivariate filter methods are usually faster than embedded approaches and wrapper methods, but slower than univariate filter methods (feature ranking), which are often a good choice for datasets with a very large numbers of instances. Since wrapper and embedded approaches select the feature subsets based on a specific learning algorithm, they are more prone to over fitting than filter methods which are independent of the later applied classifier. Moreover, deterministic wrapper algorithms are also prone to getting stuck in local optima. On the other hand, wrapper and embedded methods have often proven to achieve very good classification accuracies for specific problems. Due to the wide variety of variables, data distributions, classifiers and objectives, there is no feature selection method that is universally better than others. A general comparison of filter and wrapper methods as well as embedded approaches is, for example, given in [29], where methods are compared in terms of classification performance, general and computational complexity, scalability, model independence, and behavior (are methods prone to getting stuck in local optima or have high risk of over fitting).

IV. RESULT ANALYSIS

The proportion of selected feature is the ratio of number of features selected by a feature selection algorithm to original number of features of a data set. According to definition authors taking some common five text ,five image and five microarray data set like as chess, coil 2000, elephant, colon and breast cancer. Evaluate proportion of this data generally according to all literature algorithms achieve significant reduction of dimensionality by selecting only a small portion of the original features, figure:1 represent the algorithm vs. proportion of selected feature where 1(FAST) 2(FCBF) 3(CFS) 4(RelifF) represent the algorithm of feature sub selection.

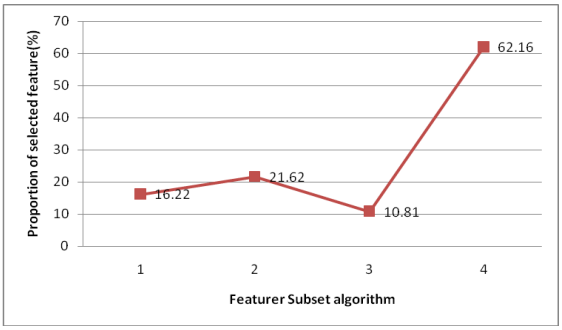

Figure1: Feature subset algorithm vs proportion of selected feature

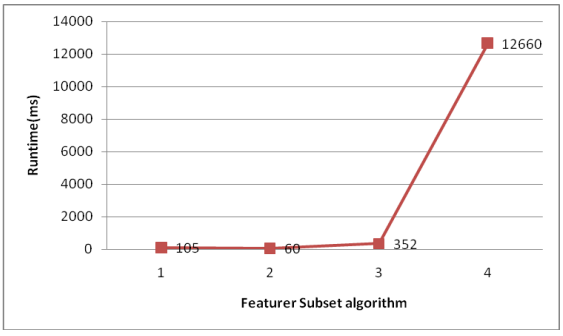

Figure 2: Feature subset algorithm vs proportion of selected feature.

From the analysis the advantages of filter approaches to feature selection outweigh their disadvantages. Generally, filters execute times is faster than wrappers, so it better chance of scaling databases with a large number of features than other algorithm. Filters method do not require re-execution for different learning algorithms [13] and it provide the same benefits for learning as wrappers method. But improvement of accuracy for a particular learning algorithm is required.

V. CONCLUSION

Feature Selection techniques are studies and classified and define the better response of feature subset selection which is the search algorithm. We observe selection bias in the context of classification based on the published algorithm in years 1997 to 2013.

REFERENCES

[1] Guyon I. and Elisseeff A., "An introduction to variable and feature selection," The Journal of Machine Learning Research, vol. 3, pp. 1157-1182, 2003.

[2] Yu L. and Liu H., "Efficient feature selection via analysis of relevance and redundancy," The Journal of Machine Learning Research, vol. 25, pp. 1205-1224, 2004.

[3] Peng H. C., Long F. H., and Ding C., "Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 27, no. 8, pp. 1226-1238, 2005.

[4] Banks A., Vincent J., and Anyakoha C., "A review of particle swarm optimization. Part i: background and development," Natural Computing, vol. 6, no. 4, pp. 467-484, 2007.

[5] Azevedo G. L. F. B. G., Cavalcanti G. D. C., and Filho E. C. B. C., "An approach to feature selection for keystroke dynamics systems based on pso and feature weighting," in Proc. IEEE Congress on Evolutionary Computation (CEC'97), pp. 3577-3584, 2007.

[6] Wang X., Yang J., Teng X., Xia W., and Jensen R., "Feature selection based on rough sets and article swarm optimization," Pattern Recognition Letters, vol. 28, no. 4, pp. 459-471, 2007.

[7] Chakraborty B., "Feature subset selection by particle swarm optimization with fuzzy fitness function," in Proceedings3rd International Conference on Intelligent System and Knowledge Engineering (ISKE'08)), vol. 1, pp. 1038-1042, 2008.

[8] Chuang L., Chang H., Tu C., and Yang C., "Improved binary pso for feature selection using gene expression data," Computational Biology and Chemistry, vol. 32, no. 1, pp. 29-38, 2008.

[9] Li A. and Wang B., "Feature subset selection based on binary particle swarm optimization and overlap information entropy," in International Conference on Computational Intelligence and Software Engineering (CiSE'09), pp. 1-4, 2009.

[10] Hall M., Frand E., Holms G., Pfahringer B., Reutemann P., and Witten I. H., "The weka data mining software: An update," SIGKDD Explorations, vol. 11, no. 1, pp. 10-18, 2009.

[11] Clerc M., Particle swarm optimization, Wiley-ISTE, 2010.

[12] Frank A. and Asuncion A., "UCI machine learning repository," 2010.

[13] Wang J., Zhao Y., and Liu P., "Effective feature selection with particle swarm optimization based one-dimension searching," in 3rd International Symposium on Systems and Control in Aeronautics and Astronautics (ISSCAA), pp. 702-705. 2010.

[14] Dejaeger K., Verbeke W., Martens D. and Baesens B., "Data Mining Techniques for Software Effort Estimation: A Comparative Study," IEEE Transactions on Software Engineering - TSE , vol. 38, no. 2, pp. 375-397, 2012.

[15] Javed K., Babri H. A., Saeed M., "Feature Selection Based on Class-Dependent Densities for High-Dimensional Binary Data," IEEE Transactions on Knowledge and Data Engineering - TKDE , vol. 24, no. 3, pp. 465-477, 2012.

[16] Kohavi R. and John G. H., "A survey on particle swarm optimization in feature selection," in Global Trends in Information Systems and Software Applications, Springer, pp. 192-201, 2012.

[17] Xue B., Zhang M., and Browne W. N., "Multi-objective particle swarm optimization (pso) for feature selection," in Proceedings of the fourteenth international conference on Genetic and evolutionary computation conference, pp. 81-88, 2012.

[18] Cervante L., Xue B., Zhang M., and Shang L., "Binary particle swarm optimization for feature selection: A filter based approach," in Proc. IEEE Congress on Evolutionary Computation (CEC'12), pp. 1-8, 2012.

[19] Qinbao Song, Jingjie Ni, and Guangtao Wang, "A Fast Clustering-Based Feature Subset Selection Algorithm for High-Dimensional Data," IEEE Transactions On Knowledge And Data Engineering, Vol. 25, No. 1, pp. 1-14, January 2013.

[20] L. Yu and H. Liu, "Feature Selection for High-Dimensional Data: A Fast Correlation-Based Filter Solution," Proc. 20th Int'l Conf. Machine Leaning, vol. 20, no. 2, pp. 856-863, 2003.

[21] 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.

[22] G. Van Dijck and M.M. Van Hulle, "Speeding Up the Wrapper Feature Subset Selection in Regression by Mutual Information Relevance and Redundancy Analysis," Proc. Int'l Conf. Artificial Neural Networks, 2006.

[23] I.S. Dhillon, S. Mallela, and R. Kumar, "A Divisive Information Theoretic Feature Clustering Algorithm for Text Classification," J. Machine Learning Research, vol. 3, pp. 1265-1287, 2003.

[24] R. Butterworth, G. Piatetsky-Shapiro, and D.A. Simovici, "On Feature Selection through Clustering," Proc. IEEE Fifth Int'l Conf. Data Mining, pp. 581-584, 2005.

[25] Ron Kohavi and George H. John, "Wrapper for Feature Subset Selection."Artificial Intelligence, 97(1-2) pp.273-324, 1997.

[26] I. Guyon and A. Elissee, "An Introduction to Variable and Feature Selection," J. Mach. Learn.Res.,3, pp. 1157-1182, 2003.

[27] Lei Yu and Huan Liu, "Feature Selection for High Dimensional Data: A Fast Correlation-based filter solution," ICML 03: Proceedings of the 20th International Conference Machine learning, pp. 856-863, 2003.

[28] Fan Li and Yiming Yang, "Using recursive classification to discover predictive features," proceeding of the 2005 ACM Symposium applied computing, pp 1054-1058, 2005.

[29] Y. Saeys, I Inza, and P Larranaga."A Review of Feature Selection Techniques in Bioinformatics,"Bioinformatic, 2007.

[30] M. Robnik-Sikonja and I. Kononenko, "Theoretical and Empirical Analysis of ReliefF and RReliefF," Mach.Learn.,53(1-2), pp. 32-69,2003.

[31] Max Bramer. Principles of Data Mining. Springer, 2007.

[32] M. Dash, H. Liu and H, Motoda, "Consistency based Feature Selection," Proceedings of the 4th Pacic Asia Conference on Knowledge Discovery and Data Mining, Current Issues and New Applications, pp.98-109, Springer, 2000.

[33] M. Dash, H Liu, "Consistency based searching Feature Selection," Artificial Intelligence, 151(1-2), pp. 155-176, 2003.

[34] L Yu and H Liu, "Efficient Feature Selection via Analysis of Relevance and Redundancy,"J.Mach. Learn Res. Vol.5,pp. 1205-1224, 2004.

[35] AL Blum and P Langley, "Selection of Relevant Features and Examples in Machine Learning," Artificial Intelligence, Vol.97, pp. 245-271, 1997.