International Journal of Science Engineering and Advance Technology

A Novel Subset Selection Clustering-Based Algorithm for High Dimensional Data

Balineni Bala Krishna[1], Kolavasi Chandra Mouli[2]
M.Tech (CSE), NRI Institute of Technology (NRIIT), A.P., India.
[2]Assistant Professor, Dept. of Master of Computer Application, NRI Institute of Technology (NRIIT), A.P., India.

Abstract — Feature selection, also known as variable selection, attribute selection or variable subset selection, is the process of selecting a subset of relevant features (variables, predictors) for use in model construction. Feature selection techniques are to be distinguished from feature extraction. Feature extraction creates new features from functions of the original features, whereas feature selection returns a subset of the features. Feature selection techniques are often used in domains where there are many features and comparatively few samples (or data points***).*** It involves identifying a subset of the most useful features that produces compatible results as the original entire set of features. A feature selection algorithm may be evaluated from both the efficiency and effectiveness points of view. While the efficiency concerns the time required to find a subset of features, the effectiveness is related to the quality of the subset of features. Based on these criteria, a fast clustering-based feature selection algorithm, FAST, is proposed and experimentally evaluated in this paper. The FAST algorithm works in two steps. In the first step, features are divided into clusters by using graph-theoretic clustering methods. In the second step, the most representative feature that is strongly related to target classes is selected from each cluster to form a subset of features. Features in different clusters are relatively independent; the clustering-based strategy of FAST has a high probability of producing a subset of useful and independent features. To ensure the efficiency of FAST, we adopt the efficient minimum-spanning tree clustering method. The efficiency and effectiveness of the FAST algorithm are evaluated through an empirical study. Extensive experiments are carried out to compare FAST and several representative feature selection algorithms, namely, FCBF, ReliefF, CFS, Consist, and FOCUS-SF, with respect to four types of well-known classifiers, namely, the probability-based Naive Bayes, the tree-based C4.5, the instance-based IB1, and the rule-based RIPPER before and after feature selection. The results, on 35 publicly available real-world high dimensional image, microarray, and text data, demonstrate that FAST not only produces smaller subsets of features but also improves the performances of the four types of classifiers**.**

Keywords — filter method, graph-based clustering Feature subset selection, feature clustering.

I. Introduction

A feature selection algorithm can be seen as the combination of a search technique for proposing new feature subsets, along with an evaluation measure which scores the different feature subsets. The simplest algorithm is to test each possible subset of features finding the one which minimises 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.The aim of choosing a subset of good features with respect to the target concepts, feature subset selection is an effective way for reducing dimensionality, removing irrelevant data, increasing learning accuracy, and improving result comprehensibility [13], [16]. Many feature subset selection methods have been proposed and studied for machine learning applications. They can be divided into four broad categories: the Embedded, Wrapper, Filter, and Hybrid approaches. The embedded methods incorporate feature selection as a part of the training process and are usually specific to given learning algorithms, and therefore may be more efficient than the other three categories [10]. Traditional machine learning algorithms like decision trees or artificial neural networks are examples of embedded approaches [14]. The wrapper methods use the predictive accuracy of a predetermined learning algorithm to determine the goodness of the selected subsets, the accuracy of the learning algorithms is usually high. However, the generality of the selected features is limited and the computational complexity is large. The filter methods are independent of learning algorithms, with good generality.

Their computational complexity is low, but the accuracy of the learning algorithms is not guaranteed [13], [12], [6]. The hybrid methods are a combination of filter and wrapper methods [10], [15], [16] by using a filter method to reduce search space that will be considered by the subsequent wrapper. They mainly focus on combining filter and wrapper methods to achieve the best possible performance with a particular learning algorithm with similar time complexity of the filter methods. The

wrapper methods are computationally expensive and tend to overfit on small training sets [13], [15]. The filter methods, in addition to their generality, are usually a good choice when the number of features is very large. Thus, we will focus on the filter method in this paper. With respect to the filter feature selection methods, the application of cluster analysis has been demonstrated to be more effective than traditional feature selection algorithms. Pereira et al. [12], Baker et al. [4], and Dhillon et al. [10] employed the distributional clustering of words to reduce the dimensionality of text data. In cluster analysis, graph-theoretic methods have been well studied and used in many applications. Their results have, sometimes, the best agreement with human performance [12]. The general graph-theoretic clustering is simple: Compute a neighborhood graph of instances, then delete any edge in the graph that is much longer/shorter (according to some criterion) than its neighbors. The result is a forest and each tree in the forest represents a cluster. In our study, we apply graph theoretic clustering methods to features. In particular, we adopt the minimum spanning tree (MST) based clustering algorithms, because they do not assume that data points are grouped around centers or separated by a regular geometric curve are been widely used.

II .PROBLEM STATEMENT

A feature selection algorithm may be evaluated from both the efficiency and effectiveness points of view. While the efficiency concerns the time required to find a subset of features, the effectiveness is related to the quality of the subset of features. Based on these criteria, a fast clustering-based feature selection algorithm, FAST, is proposed and experimentally evaluated in this paper .Based on the MST method, we propose a Fast clustering-bAsed feature Selection algoriThm (FAST).The FAST algorithm works in two steps. In the first step, features are divided into clusters by using graph-theoretic clustering methods. In the second step, the most representative feature that is strongly related to target classes is selected from each cluster to form the final subset of features. Features in different clusters are relatively independent, the clustering-based strategy of FAST has a high probability of producing a subset of useful and independent features. The proposed feature subset selection algorithm FAST was tested upon 35 publicly available image, microarray, and text data sets. The experimental results show that, compared with other five different types of feature subset selection algorithms, the proposed algorithm not only reduces the number of features, but also improves the performances of the four well-known different types of classifiers.

III .RELATED WORK

Feature subset selection can be viewed as the process of identifying and removing as many irrelevant and redundant features as possible. This is because: (i) irrelevant features do not contribute to the predictive accuracy [3], and (ii) redundant features do not redound to getting a better predictor for that they provide mostly information which is already present in other feature(s). Of the many feature subset selection algorithms, some can effectively eliminate irrelevant features but fail to handle redundant features [13], [1], [7], [14], [10], yet some of others can eliminate the irrelevant while taking care of the redundant features [5], [6], [12], [10]. Our proposed FAST algorithm falls into the second group. Traditionally, feature subset selection research has focused on searching for relevant features. A well known example is Relief [14], 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 [16]. Relief-F [11] extends Relief, enabling this method to work with noisy and incomplete data sets and to deal with multi-class problems, but still cannot identify redundant features. However, along with irrelevant features, redundant features also affect the speed and accuracy of learning algorithms, and thus should be eliminated as well [16], [15], [11]. CFS [6], FCBF [10] and CMIM [12] are examples that take into consideration the redundant features. CFS [9] 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. CMIM [12] iteratively picks features which maximize their mutual information with the class to predict, conditionally to the response of any feature already picked. Different from these algorithms, our proposed FAST algorithm employs clustering based method to choose features. Recently, hierarchical clustering has been adopted in word selection in the context of text classification (e.g., [2], [4], and [10]). Distributional clustering has been used to cluster words into groups based either on their participation in particular grammatical relations with other words by Pereira et al. [12] or on the distribution of class labels associated with each word by Baker and McCallum [4]. As distributional clustering of words are agglomerative in nature, and result in sub-optimal word clusters and high computational cost, Dhillon et al. [10] proposed a new information-theoretic divisive algorithm for word clustering and applied it to text classification. Butterworth et al. [8] 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. Furthermore, even compared with other feature selection methods, the obtained accuracy is lower. Hierarchical clustering also has been used to select features on spectral data. Van

Dijk and Van Hullefor [14] proposed a hybrid filter/wrapper feature subset selection algorithm for regression. Krier et al. [8] 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 Dijk and Van Hullefor [4] 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, our proposed FAST algorithm uses minimum spanning tree based method to cluster features. Meanwhile, it does not assume that data points are grouped around centers or separated by a regular geometric curve.

FEATURE SUBSET SELECTION ALGORITHM

3.1 Framework and definitions irrelevant features, along with redundant features, severely affect the accuracy of the learning machines [1], [15]. Thus, feature subset selection should be able to identify and remove as much of the irrelevant and redundant information as possible. Moreover, "good feature subsets contain features highly correlated with (predictive of) the class, yet uncorrelated with (not predictive of) each other." [10] Fig. 2: Framework of the proposed feature subset selection algorithm keeping these in mind, we develop a novel algorithm which can efficiently and effectively deal with both irrelevant and redundant features, and obtain a good feature subset. We achieve this through a new feature selection framework (shown in Fig.2) which composed of the two connected components of irrelevant feature removal and redundant feature elimination. The former obtains features relevant to the target concept by eliminating irrelevant ones, and the latter removes redundant features from relevant ones via choosing representatives from different feature clusters, and thus produces the final subset. The irrelevant feature removal is straightforward once the right relevance measure is defined or selected, while the redundant feature elimination is a bit of sophisticated. In our proposed FAST algorithm, it involves (i) the construction of the minimum spanning tree (MST) from a weighted complete graph; (ii) the partitioning of the MST into a forest with each tree representing a cluster; and (iii) the selection of representative features from the clusters.

Four different types of classification algorithms are employed to classify data sets before and after feature selection. They are (i) the probability-based Naive Bayes (NB), (ii) the tree-based C4.5, (iii) the instance-based lazy learning algorithm IB1, and (iv) the rule-based RIPPER, respectively. Naive Bayes utilizes a probabilistic method for classification by multiplying the individual probabilities of every feature-value pair. This algorithm assumes independence among the features and even then provides excellent classification results. Decision tree learning algorithm C4.5 is an extension of ID3 that accounts for unavailable values, continuous attribute value ranges, pruning of decision trees, rule derivation, and so on. The tree comprises of nodes (features) that are selected by information entropy. Instance-based learner IB1 is a single-nearest neighbor algorithm, and it classifies entities taking the class of the closest associated vectors in the training set via distance metrics. It is the simplest among the algorithms used in our study. Inductive rule learner RIPPER (Repeated Incremental Pruning to Produce Error Reduction) [12] is a propositional rule learner that defines a rule based detection model and seeks to improve it iteratively by using different heuristic techniques. The constructed rule set is then used to classify new instances. 3) When evaluating the performance of the feature subset selection algorithms, four metrics, (i) the proportion of selected features (ii) the time to obtain the feature subset, (iii) the classification accuracy, and (iv) the Win/Draw/Loss record, are used. The proportion of selected features is the ratio of the number of features selected by a feature selection algorithm to the original number of features of a data set. The Win/Draw/Loss record presents three values on a given measure, i.e. the numbers of data sets for which our proposed algorithm FAST obtains better, equal, and worse performance than other five feature selection algorithms, respectively.

This can be illustrated by an example. Suppose the MST shown in Fig.2 is generated from a complete graph G. In order to cluster the features, we first traverse all the six edges, and then decide to remove the edge (F_0, F_4) because its weight $SU(F_0, F_4) = 0.3$ is smaller than both $SU(F_0, C) = 0.5$ and $SU(F_4, C) = 0.7$. This makes the MST is clustered into two clusters denoted as $V(T_1)$ and $V(T_2)$. Each cluster is a MST as well. Take $V(T_1)$ as an example. From Fig.2 we know that $SU(F_0, F_1) >$

$SU\{F_1, C\}$, $SU\{F_1, F_2\} > SU(F_1, C) \wedge SU\{F_1, F_2\} > SU\{F_2, C\}$, $SU\{F_1, F_3\} > SU(F_1, C) \wedge SU\{F_1, F_3\} > SU\{F_3, C\}$. We also observed that there is no edge exists between F_0 and F_2, F_0 and F_3, and F_2 and F_3. Considering that T_1 is a MST, so the $SU(F_0, F_2)$ is greater than $SU\{F_0, F_1\}$ and $SU(F_1, F_2)$, $SU\{F_0, F_3\}$ is greater than $SU\{F_0, F_1\}$ and $SU\{F_1, F_3\}$, and $SU(F_2, F_3)$ is greater than $SU(F_1, F_2)$ and $SU(F_2, F_3)$. Thus, $SU\{F_0, F_2\} > SU\{F_0, C\} \wedge SU(F_0, F_2) > SU(F_2, C)$, $SU\{F_0, F_3\} > SU\{F_0, C\} \wedge SU(F_0, F_3) > SU\{F_3, C\}$, and $SU(F_2, F_3) > SU\{F_2, C\} \wedge SU(F_2, F_3) > SU\{F_3, C\}$ also hold. As the mutual information between any pair $(F_i, F_j)(i, j = 0, 1, 2, 3 \wedge i \neq j)$ of F_0, F_1, F_2, and F_3 is greater than the mutual information between class C and F_i or F_j, features F_0, F_1, F_2, and F_3 are redundant.

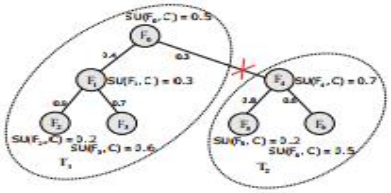

Fig. 2: Example of the clustering step

After removing all the unnecessary edges, a forest *Forest* is obtained. Each tree $T_j \in Forest$ represents a cluster that is denoted as $V(T_j)$, which is the vertex set of T_j as well. As illustrated above, the features in each cluster are redundant, so for each cluster $V(T_j)$ we choose a representative feature F_R^j whose *T-Relevance* $SU(F_R^j, C)$ is the greatest. All F_R^j $(j = 1...|Forest|)$ comprise the final feature subset $\cup F_R^j$.

The details of the FAST algorithm is shown in Algorithm 1.

Algorithm 1: FAST

```
inputs: D(F_1, F_2, ..., F_m, C) - the given data set
        θ - the T-Relevance threshold.
output: S - selected feature subset .
//==== Part 1 : Irrelevant Feature Removal ====
for i = 1 to m do
    T-Relevance = SU (F_i, C)
    if T-Relevance > θ then
        S = S ∪ {F_i};
//==== Part 2 : Minimum Spanning Tree Construction ====
G = NULL; //G is a complete graph
for each pair of features {F'_i, F'_j} ⊂ S do
    F-Correlation = SU (F'_i, F'_j)
    Add F'_i and/or F'_j to G with F-Correlation as the weight of
    the corresponding edge;
minSpanTree = Prim (G); //Using Prim Algorithm to generate the
    minimum spanning tree
//==== Part 3 : Tree Partition and Representative Feature Selection ====
Forest = minSpanTree
for each edge E_ij ∈ Forest do
    if SU(F'_i, F'_j) < SU(F'_i, C) ∧ SU(F'_i, F'_j) < SU(F'_j, C) then
        Forest = Forest − E_ij
S = ϕ
for each tree T_i ∈ Forest do
    F_R^j = argmax_{F'_k ∈ T_i} SU(F'_k, C)
    S = S ∪ {F_R^j};
return S
```

1) The proposed algorithm is compared with five different types of representative feature selection algorithms. They are (i) FCBF [10], [7], (ii) ReliefF [11], (iii) CFS [6], (iv) Consist [14], and (v) FOCUSSF [2], respectively. FCBF and ReliefF evaluate features individually. For FCBF, in the experiments, we set the relevance threshold to be the SUvalue of the m/logm ranked feature for each data set (m is the number of features in a given data set) as suggested by Yu and Liu [10], [11]. ReliefF searches for nearest neighbors of instances of different classes and weights features according to how well they differentiate instances of different classes. The other three feature selection algorithms are based on subset evaluation. CFS exploits best-first search based on the evaluation of a subset that contains features highly correlated with the target concept, yet uncorrelated with each other. The Consist method searches for the minimal subset that separates classes as consistently as the full set can under best-first search strategy. FOCUS-SF is a variation of FOCUS [2]. FOCUS has the same evaluation strategy as Consist, but it examines all subsets of features. Considering the time efficiency, FOUCS-SF replaces exhaustive search in FOCUS with sequential forward selection

In order to make the best use of the data and obtain stable results, a (M = 5)×(N = 10)-cross-validation strategy is used. That is, for each data set, each feature subset selection algorithm and each classification algorithm, the 10-fold cross-validation is repeated M = 5 times, with each time the order of the instances of the data set being randomized. This is because many of the algorithms exhibit order effects, in that certain orderings dramatically improve or degrade performance [5]. Randomizing the order of the inputs can help diminish the order effects.

In the experiment, for each feature subset selection algorithm, we obtain M×N feature subsets Subset and the corresponding runtime Time with each data set. Average Subset and Time, we obtain the number of selected features further the proportion of selected features and the corresponding runtime for each feature selection algorithm on each data set. For each classification algorithm, we obtain M×N classification Accuracy for each feature selection algorithm and each data set. Average these Accuracy, we obtain mean accuracy of each classification algorithm under each feature selection.

Procedure *Experimental Process*

```
1  M = 5, N = 10
2  DATA = {D1, D2, ..., D35}
3  Learners = {NB, C4.5, IB1, RIPPER}
4  FeatureSelectors = {FAST, FCBF, ReliefF, CFS, Consist, FOCUS-SF}
5  for each data ∈ DATA do
6      for each times ∈ [1, M] do
7          randomize instance-order for data
8          generate N bins from the randomized data
9          for each fold ∈ [1, N] do
10             TestData = bin[fold]
11             TrainingData = data - TestData
12             for each selector ∈ FeatureSelectors do
13                 (Subset, Time) = selector(TrainingData)
14                 TrainingData' = select Subset from TrainingData
15                 TestData' = select Subset from TestData
16                 for each learner ∈ Learners do
17                     classifier = learner(TrainingData')
18                     Accuracy = apply classifier to TestData'
```

IV. Conclusion

Filter methods use a proxy measure instead of the error rate to score a feature subset. This measure is chosen to be fast to compute, whilst still capturing the usefulness of the feature set. In this paper, 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 drastically reduced. We have compared the performance of the proposed algorithm with those of the five well-known feature selection algorithms FCBF, ReliefF, CFS, Consist, and FOCUS-SF on the 35 publicly available image, microarray, and text data from the four different aspects of the proportion of selected features, runtime, classification accuracy of a given classifier, and the Win/Draw/Loss record. Generally, the proposed algorithm obtained the best proportion of selected features, the best runtime, and the best classification accuracy for Naive Bayes, C4.5, and RIPPER, and the second best classification accuracy for IB1. The Win/Draw/Loss records confirmed the conclusions. We also found that FAST obtains the rank of 1 for microarray data, the rank of 2 for text data, and the rank of 3 for image data in terms of classification accuracy of the four different types of classifiers, and CFS is a good alternative. At the same time, FCBF is a good alternative for image and text data. Moreover, Consist and FOCUSSF are alternatives for text data. For the future work, we plan to explore different types of correlation measures, and study some formal properties of feature space.

References

[1] Almuallim H. and Dietterich T.G., Algorithms for Identifying Relevant Features, In Proceedings of the 9th Canadian Conference on AI, pp 38-45, 1992.

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

[3] Arauzo-Azofra A., Benitez J.M. and Castro J.L., A feature set measure based on relief, In Proceedings of the fifth international conference on Recent Advances in Soft Computing, pp 104-109, 2004.

[4] Baker L.D. and McCallum A.K., Distributional clustering of words for text classification, In Proceedings of the 21st Annual international ACM SIGIR Conference on Research and Development in information Retrieval, pp 96- 103, 1998.

[5] Fayyad U. and Irani K., Multi-interval discretization of continuous-valued attributes for classification learning, In Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence, pp 1022-1027, 1993

[6] Dougherty, E. R., Small sample issues for microarray-based classification. Comparative and Functional Genomics, 2(1), pp 28-34, 2001.

[7] Biesiada J. and Duch W., Features election for high-dimensionaldatała Pearson redundancy based filter, AdvancesinSoftComputing, 45, pp 242C249, 2008.

[8] Butterworth R., Piatetsky-Shapiro G. and Simovici D.A., On Feature Selection through Clustering, In Proceedings of the Fifth IEEE international Conference on Data Mining, pp 581-584, 2005.

[9] Cardie, C., Using decision trees to improve case-based learning, In Proceedings of Tenth International Conference on Machine Learning, pp 25-32, 1993.

[10] 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.

[11] Chikhi S. and Benhammada S., ReliefMSS: a variation on a feature ranking ReliefF algorithm. Int. J. Bus. Intell. Data Min. 4(3/4), pp 375-390, 2009.

[12] Cohen W., Fast Effective Rule Induction, In Proc. 12th international Conf. Machine Learning (ICML'95), pp 115-123, 1995.

[13] Dash M. and Liu H., Feature Selection for Classification, Intelligent Data Analysis, 1(3), pp 131-156, 1997.

[14] Dash M., Liu H. and Motoda H., Consistency based feature Selection, In Proceedings of the Fourth Pacific

Asia Conference on Knowledge Discovery and Data Mining, pp 98-109, 2000.

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

[16] Dash M. and Liu H., Consistency-based search in feature selection. Artificial Intelligence, 151(1-2), pp 155-176, 2003.