RESEARCH ARTICLE OPEN ACCESS

High Dimensional Data Clustering Using Fast Cluster Based Feature Selection

Karthikeyan.P[1], Saravanan.P[2], Vanitha.E[3]
[1]PG Scholar, Department of Computer & Communication Engineering, PTR CET, Madurai, Tamil Nadu
[2]Assistant Professor, Department of Computer Science Engineering, PTR CET, Madurai, Tamil Nadu
[3]Assistant Professor, Department of Computer Science Engineering, PTR CET, Madurai, Tamil Nadu

ABSTRACT
Feature selection 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 (MST) using the Kruskal's Algorithm clustering method. The efficiency and effectiveness of the FAST algorithm are evaluated through an empirical study.

Index Terms—Feature subset selection, filter method, feature clustering, graph-based clustering

I. INTRODUCTION CLASSIFICATION

Data mining refers to "using a variety of techniques to identify nuggets of information or decision-making knowledge in bodies of data, and extracting these in such a way that they can be put to use in the areas such as decision support, prediction, forecasting and estimation. The data is often voluminous, but as it stands of low value as no direct use can be made of it; it is the hidden information in the data that is useful". Data mine tools have to infer a model from the database, and in the case of supervised learning this requires the user to define one or more classes.

The database contains one or more attributes that denote the class of a tuple and these are known as predicted attributes whereas the remaining attributes are called predicting attributes. A combination of values for the predicted attributes defines a class. When learning classification rules the system has to find the rules that predict the class from the predicting attributes so firstly the user has to define conditions for each class, the data mine system then constructs descriptions for the classes. Basically the system should given a case or tuple with certain known attribute values be able to predict what class this case belongs to, once classes are defined the system should infer rules that govern the classification therefore the system should be able to find the description of each class.

With 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.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 selections a part of the training process and are usually specific to given learning algorithms, and therefore maybe more efficient than the other three categories. Traditional machine learning algorithms like decision trees or artificial neural networks are examples of embedded approaches.

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 the hybrid

methods area combination of filter and wrapper methods 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. 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.

1.1 System Architecture

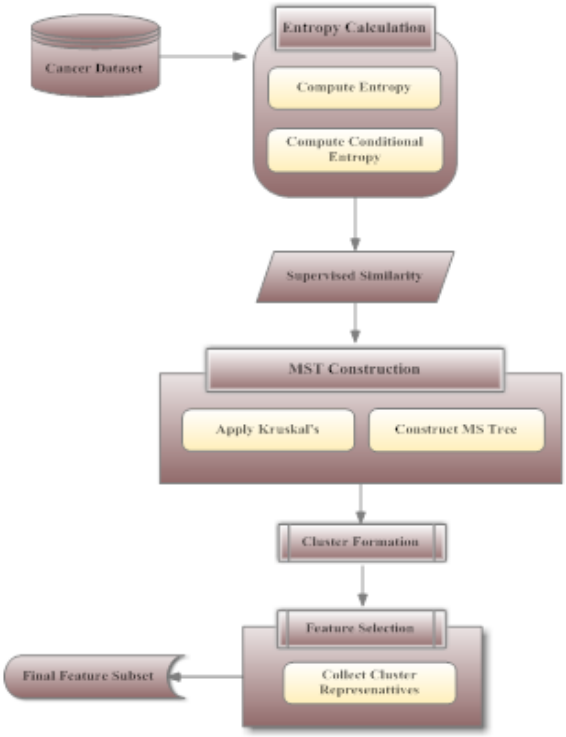

Fig 1Architecture of Proposed Method

II. CLUSTERING

Clustering and segmentation are the processes of creating a partition so that all the members of each set of the partition are similar according to some metric. A cluster is a set of objects grouped together because of their similarity or proximity. Objects are often decomposed into an exhaustive and/or mutually exclusive set of clusters. Clustering according to similarity is a very powerful technique, the key to it being to translate some intuitive measure of similarity into a quantitative measure. When learning is unsupervised then the system has to discover its own classes i.e. the system clusters the data in the database. The system has to discover subsets of related objects in the training set and then it has to find descriptions that describe each of these subsets. There are a number of approaches for forming clusters. One approach is to form rules which dictate membership in the same group based on the level of similarity between members. Another approach is to build set functions that measure some property of partitions as functions of some parameter of the partition.

III. FEATURE SELECTION

It is widely recognized that a large number of features can adversely affect the performance of inductive learning algorithms, and clustering is not an exception. However, while there exists a large body of literature devoted to this problem for supervised learning task, feature selection for clustering has been rarely addressed. The problem appears to be a difficult one given that it inherits all the uncertainties that surround this type of inductive learning. Particularly, that there is not a single performance measure widely accepted for this task and the lack of supervision available.

In machine learning and statistics, feature selection, also known as variable selection, attribute selection or variable subset selection, is the process of selecting a subset of relevant features for use in model construction. The central assumption when using a feature selection technique is that the data contains many redundant or irrelevant features. Redundant features are those which provide no more information than the currently selected features, and irrelevant features provide no useful information in any context. Feature selection techniques are a subset of the more general field of 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). The archetypal case is the use of feature selection in analyzing DNA microarrays, where there are many thousands of features, and a few tens to hundreds of samples. Feature selection techniques provide three main benefits when constructing predictive models

- Improved model interpretability,
- Shorter training times,
- Enhanced generalization by reducing over fitting.

Feature selection is also useful as part of the data analysis process, as shows which features are important for prediction, and how these features are related. With such an 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. Irrelevant features, along with

redundant features, severely affect the accuracy of the learning machines. 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."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

3.1 Wrapper Filter

Wrapper methods are widely recognized as a superior alternative in supervised learning problems, since by employing the inductive algorithm to evaluate alternatives they have into account the particular biases of the algorithm. How- ever, even for algorithms that exhibit a moderate complexity, the number of executions that the search process requires results in a high computational cost, especially as we shift to more exhaustive search strategies. 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

3.2 Hybrid Approach

The hybrid methods are a combination of filter and wrapper methods 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.

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. 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 and have been widely used in practice.

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 various numerical 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 classification accuracy.

3.3 Using Mutual Information for Selecting Features in Supervised Neural Net Learning

Investigates the application of the mutual in for "criterion to evaluate a set of candidate features and to select an informative subset to be used as input data for a neural network classifier. Because the mutual information measures arbitrary dependencies between random variables, it is suitable for assessing the "information content" of features in complex classification tasks, where methods bases on linear relations (like the correlation) are prone to mistakes.

The fact that the mutual information is independent of the coordinates chosen permits a robust estimation. Nonetheless, the use of the mutual information for tasks characterized by high input dimensionality requires suitable approximations because of the prohibitive demands on computation and samples. An algorithm is proposed that is based on a "greedy" selection of the features and that takes both the mutual information with respect to the output class and with respect to the already-selected features into account. Finally the results of a series of experiments are discussed.

During "preprocessing" stage, where an appropriate number of relevant features are extracted from the raw data, has a crucial impact both on the complexity of the learning phase and on the achievable generalization performance. While it is essential that the information contained in the input vector is sufficient to determine the output class, the presence of too many input features can burden the training process and can produce a neural network with more connection weights that those required by the problem

A major weakness of these methods is that they are not invariant under a transformation of the variables. For example a linear scaling of the input variables (that may be caused by a change of units for the measurements) is sufficient to modify the PCA results. Feature selection methods that are sufficient for simple distributions of the patterns belonging to different classes can fail in classification tasks with complex decision boundaries. In addition, methods based on a linear dependence (like the correlation) cannot take care of arbitrary relations between the pattern coordinates and the different classes. On the contrary, the mutual information can measure arbitrary relations between variables and it does not depend on transformations acting on the different variables.

Our objective was less ambitious, because only the first of the above options was considered (leaving the second for the capabilities of the neural net to build complex features from simple ones). We assumed that a set of candidate features with globally sufficient information is available and that the problem is that of extracting from this set a suitable subset that is sufficient for the task, thereby reducing the processing times in the operational phase and, possibly, the training times and the cardinality of the example set needed for a good generalization.

In particular we were interested in the applicability of the mutual information measure. For this reason we considered the estimation of the MI from a finite set of samples, showing that the MI for different features is over-estimated in approximately the same way. This estimation is the building block of the MIFS algorithm, where the features are selected in a "greedy" manner, ranking them according to their MI with respect to the class discounted by a term that takes the mutual dependencies into account.

3.4 On Feature Selection through Clustering

The algorithm for feature selection that clusters attributes using a special metric and then makes use of the dendrogram of the resulting cluster hierarchy to choose the most relevant attributes. The main interest of our technique resides in the improved understanding of the structure of the analyzed data and of the relative importance of the at-tributes for the selection process.

The performance, robustness, and usefulness of classification algorithms are improved when relatively few features are involved in the classification. Thus, selecting relevant features for the construction of classifiers has received a great deal of attention. The central idea of this work is to introduce an algorithm for feature selection that clusters attributes using a special metric and, then uses a hierarchical clustering for feature selection. Hierarchical algorithms generate clusters that are placed in a cluster tree, which i s commonly known as a dendrogram. Clustering's are obtained by extracting t hose clusters that are situated at a given height in this tree. It shows that good classifiers can be built by using a small number of attributes located at the centers of the clusters identified in the dendrogram. This type of data compression can be achieved with little or no penalty in terms of the accuracy of the classifier produced and highlights the relative importance of attributes.

Clustering's were extracted from the tree produced by the algorithm by cutting the tree at various heights starting with the maximum height of the tree created above (corresponding t o a single cluster) and working down t o a height of 0 (which consists of single-attribute clusters). A 'representative' attribute was created for each cluster as the attribute that has the minimum total distance to the other members of the cluster, again using the Barth ´elemyMontjardet distance. A similar study was undertaken f or the zoo database, after eliminating the attribute animal which determines uniquely the type of the animal. These results suggest that this method has comparable accuracy to the wrapper method and CSF. However, the tree of attributes helps to understand the relationships between attributes and their relative importance.

Attribute clustering help to build classifiers in a semi-supervised manner allowing analysts a certain degree of choice in the s election of the features that may be considered by classifiers, and illuminating relationships between attributes and their relative importance for classification. With the increased interest of data miners in n bio-computing in n general, and in microarray data in particular, classification problems that involve thousands of features and relatively few examples came t o t he fore. We intend to apply our techniques to this type of data.

IV. IRRELEVANT FEATURES REMOVAL

Irrelevant features, along with redundant features, severely affect the accuracy of the learning machines. 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." 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 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 1) the construction of the minimum spanning tree from a weighted complete graph; 2) the partitioning of the MST into a forest with each tree representing a cluster; and 3) the selection of representative features from the clusters.

4.1 Load Data and Classify

Load the data into the process. The data has to be preprocessed for removing missing values, noise and outliers. Then the given dataset must be converted into the arff format which is the standard format for WEKA toolkit. From the arff format, only the attributes and the values are extracted and stored into the database. By considering the last column of the dataset as the class attribute and select the distinct class labels from that and classify the entire dataset with respect to class labels.

4.2 Information Gain Computation

Relevant features have strong correlation with target concept so are always necessary for a best subset, while redundant features are not because their values are completely correlated with each other. Thus, notions of feature redundancy and feature relevance are normally in terms of feature correlation and feature-target concept correlation.

To find the relevance of each attribute with the class label, Information gain is computed in this module. This is also said to be Mutual Information measure. Mutual information measures how much the distribution of the feature values and target classes differ from statistical independence. This is a nonlinear estimation of correlation between feature values or feature values and target classes. The symmetric uncertainty (SU) is derived from the mutual information by normalizing it to the entropies of feature values or feature values and target classes, and has been used to evaluate the goodness of features for classification

The symmetric uncertainty is defined as follows:

$$Gain(X|Y) = H(X) - H(X|Y)$$
$$= H(Y) - H(Y\big|X)$$

To calculate gain, we need to find the entropy and conditional entropy values. The equations for that are given below:

$$H(X) = -\sum_{x \in X} p(x) log_2 p(x)$$

$$H(X|Y) = -\sum_{y \epsilon Y} p(y) \sum_{x \epsilon X} p(x|y)\, log_2 p(x|y)$$

Where p(x) is the probability density function and p (x|y) is the conditional probability density function.

4.3 T-Relevance Calculation

The relevance between the feature Fi € F and the target concept C is referred to as the T-Relevance of Fi and C, and denoted by SU(Fi,C). If SU(Fi,C) is greater than a predetermined threshold, we say that Fi is a strong T-Relevance feature.

$$SU(X,Y) = \frac{2 \times Gain(X|Y)}{H(X) + H(Y)}$$

After finding the relevance value, the redundant attributes will be removed with respect to the threshold value.

4.4 F-Correlation Calculation

The correlation between any pair of features Fi and Fj (Fi,Fj € ^ F ^ i ≠ j) is called the F-Correlation of Fi and Fj, and denoted by SU(Fi, Fj). The equation symmetric uncertainty which is used for finding the relevance between the attribute and the class is again applied to find the similarity between two attributes with respect to each label.

4.5 MST Construction

With the F-Correlation value computed above, the Minimum Spanning tree is constructed. For that, we use Kruskal's algorithm which form MST effectively.

Kruskal's algorithm is a greedy algorithm in graph theory that finds a minimum spanning tree for a connected weighted graph. This means it finds a subset of the edges that forms a tree that includes every vertex, where the total weight of all the edges in the tree is minimized. If the graph is not connected, then it finds a minimum spanning forest (a minimum spanning tree for each connected component).

Description:

1. Create a forest F (a set of trees), where each vertex in the graph is a separate tree.
2. Create a set S containing all the edges in the graph
3. While S is nonempty and F is not yet spanning

- Remove an edge with minimum weight from S
- If that edge connects two different trees, then add it to the forest, combining two trees into a single tree

- Otherwise discard that edge.

At the termination of the algorithm, the forest forms a minimum spanning forest of the graph. If the graph is connected, the forest has a single component and forms a minimum spanning tree. The sample tree is as follows,

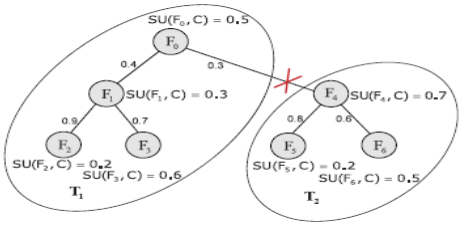

Fig 2. Correlations

ALGORITHM

---

inputs: *D*(*F*1, *F*2, ..., *Fm*, *C*) - the given data set
θ- the T-Relevance threshold.
output: *S* - selected feature subset .
//==== Part 1 : Irrelevant Feature Removal ====
1 for *i = 1 to m* **do**
2 T-Relevance = SU (*Fi*, *C*)
3 **if** T-Relevance > *θ* **then**
4 *S = S ∪ {Fi}*;
//==== Part 2: Minimum Spanning Tree Construction ====
5 *G* = NULL; //G is a complete graph
6 for *each pair of features {F'i, F'j} ⊂ S* **do**
7 F-Correlation = SU (*F'*,*F'j*)
8 *AddF'iand/orF'jtoGwith*F-Correlation *astheweightofthecorrespondingedge*;
9 minSpanTree = KRUSKALS(*G*); //Using KRUSKALS Algorithm to generate theminimum spanning tree
//==== Part 3: Tree Partition and Representative Feature Selection ====
10 Forest = minSpanTree
11 for *each edge* ∈Forest **do**
12 if
SU(*F'i, F'j*) <SU(*F'i, C*) ∧SU(*F'i, F'j*) <SU(*F'j,C*) **then**
13 Forest = Forest − *Eij*
14 *S* = *ϕ*
15 for *each tree* ∈Forest **do**
16 *Fj*$_R$= argmax*F'k*∈*Tt*SU(*F'k,C*)
17 *S = S ∪ {FjR}*;
18 return*S*

In this tree, the vertices represent the relevance value and the edges represent the F-Correlation value. The complete graph G reflects the correlations among all the target-relevant features. Unfortunately, graph G has k vertices and k(k-1)/2 edges. For high-dimensional data, it is heavily dense and the edges with different weights are strongly interwoven. Moreover, the decomposition of complete graph is NP-hard. Thus for graph G, we build an MST, which connects all vertices such that the sum of the weights of the edges is the minimum, using the well knownKruskal algorithm. The weight of edge (Fi`,Fj`) is F-Correlation SU(Fi`,Fj`).

4.6 Cluster Formation

After building the MST, in the third step, we first remove the edges whose weights are smaller than both of the T-Relevance SU(Fi`, C) and SU(Fj`, C), from the MST. After removing all the unnecessary edges, a forest Forest is obtained. Each tree Tj € Forest represents a cluster that is denoted as V (Tj), which is the vertex set of Tj as well. As illustrated above, the features in each cluster are redundant, so for each cluster V (Tj) we choose a representative feature Fj R who's T-Relevance SU(Fj R,C) is the greatest.

V. EXPERIMENTAL RESULTS

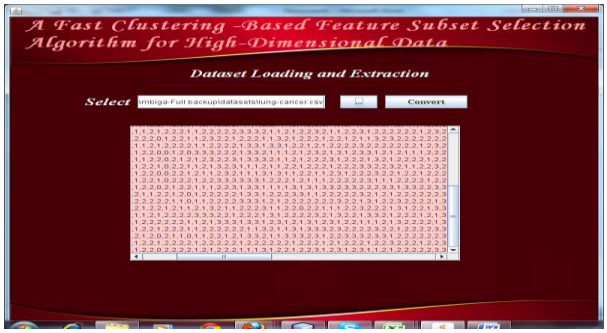

Fig 3.Dataset Loading

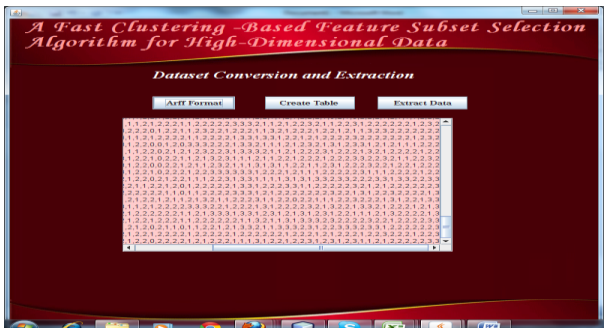

Fig 4. Dataset Conversion

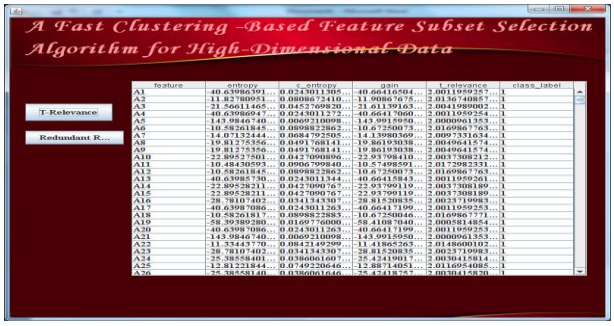

Fig 5. Entropy & Gain Values

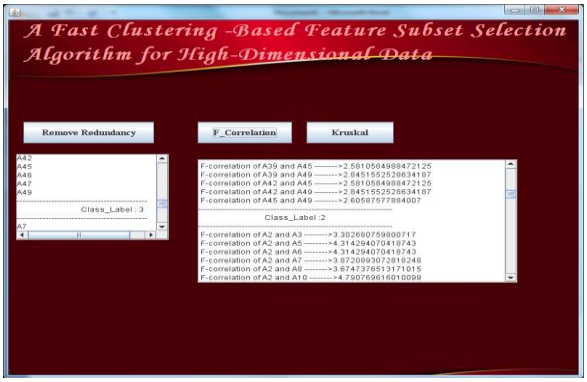

Fig 6. F- Correlation & Relevance

VI. CONCLUSION AND FUTURE WORK

In this Project present a FAST clustering-based feature subset selection algorithm for high dimensional data. The algorithm involves 1) removing irrelevant features, 2) constructing a minimum spanning tree from relative ones, and 3) 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. The text data from the four different aspects of the proportion of selected features, run time, classification accuracy of a given classifier.Clustering-based feature subset selection algorithm for high dimensional data. For the future work, we plan to explore different types of correlation measures, and study some formal properties of feature space. In feature we are going to classify the high dimensional data.

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] Bell D.A. and Wang, H., A formalism for relevance and its application infeature subset selection, Machine Learning, 41(2), pp 175-195, 2000.

[3] Biesiada J. and Duch W., Features election for high-dimensionaldatałaPearsonredundancy based filter, Advances inSoftComputing, 45, pp 242C249,2008.

[4] Dash M., Liu H. and Motoda H., Consistency based feature Selection, InProceedings of the Fourth Pacific Asia Conference on Knowledge Discoveryand Data Mining, pp 98-109, 2000.

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

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

[7] Demsar J., Statistical comparison of classifiers over multiple data sets, J.Mach. Learn. Res., 7, pp 1-30, 2006.

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

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

[10] Garcia S and Herrera F., An extension on "Statistical Comparisons ofClassifiers over Multiple Data Sets" for all pairwise comparisons, J. Mach.Learn. Res., 9, pp 2677-2694, 2008.

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

[12] Hall M.A., Correlation-Based Feature Selection for Discrete and NumericClass Machine Learning, In Proceedings of 17th International Conferenceon Machine Learning, pp 359-366, 2000.