AN EFFICIENT CLUSTERING BASED SUBSET SELECTION ALGORITHM OF HIGH DIMENSIONAL DATA

K.Siva Krishna[1]

[1]*Student,CSE, Prakasm Engineering College, A.P., India,**sivakrishna@gmail.com***

Abstract

Data mining, the extraction of hidden predictive information from large databases, is a powerful new technology with great potential to help companies focus on the most important information in their data warehouses. Clustering which tries to group a set of points into clusters such that points in the same cluster are more similar to each other than points in different clusters, under a particular similarity metric. In the generative clustering model, a parametric form of data generation is assumed, and the goal in the maximum likelihood formulation is to find the parameters that maximize the probability (likelihood) of generation of the data given the model. In the most general formulation, the number of clusters k is also considered to be an unknown parameter. Such a clustering formulation is called a "model selection" framework, since it has to choose the best value of k under which the clustering model fits the data. 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, is proposed and experimentally evaluated in this paper. This 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, we adopt the efficient minimum-spanning tree clustering method. The efficiency and effectiveness of the this algorithm are evaluated through an empirical study.

Index Terms: *Data mining, Feature selection, relevant features, redundant features, filter method, feature clustering, graph-based clustering.*

---------- *** ----------

1. INTRODUCTION

Data mining, *the extraction of hidden predictive information from large databases*, is a powerful new technology with great potential to help companies focus on the most important information in their data warehouses. 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.

Feature selection is frequently used as a preprocessing step to machine learning. It is a process of choosing a subset of original features so that the feature space is optimally reduced according to a certain evaluation criterion. Feature selection has been a fertile field of research and development since 1970's and shown very effective in removing irrelevant and redundant features, increasing efficiency in learning tasks, improving learning performance like predictive accuracy, and enhancing comprehensibility of learned results (Blum & Langley, 1997; Dash & Liu, 1997; Kohavi & John, 1997). In recent years, data has become increasingly larger in both rows (i.e., number of instances) and columns (i.e., number of features) in many applications such as genome projects (Xing et al., 2001), text categorization (Yang & Pederson, 1997), image retrieval (Rui et al., 1999), and customer relationship management (Ng & Liu, 2000). This enormity may cause serious problems to many machine learning algorithms with respect to scalability and learning performance. For example, high dimensional data (i.e., data sets with hundreds or thousands of features), can contain high degree of irrelevant and redundant information which may greatly degrade the performance of learning algorithms. Therefore, feature selection becomes very necessary for machine learning tasks when facing high dimensional data nowadays. However, this trend of enormity on both size and dimensionality also poses severe challenges to feature selection algorithms. Some of the

recent research efforts in feature selection have been focused on these challenges from handling a huge number of instances (Liu et al., 2002b) to dealing with high dimensional data (Das, 2001; Xing et al., 2001). This work is concerned about feature selection for high dimensional data. In the following, we first review models of feature selection and explain why a filter solution is suitable for high dimensional data, and then review some recent efforts in feature selection for high dimensional data.

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 [43], [46]. 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.

Feature 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) .

Feature selection algorithms can broadly fall into the filter model or the wrapper model (Das, 2001; Kohavi & John, 1997). The filter model relies on general characteristics of the training data to select some features without involving any learning algorithm, therefore it does not inherit any bias of a learning algorithm. The wrapper model requires one predetermined learning algorithm in feature selection and uses its performance to evaluate and determine which features are selected. As for each new subset of features, the wrapper model needs to learn a hypothesis (or a classifier). It tends to give superior performance as it finds features better suited to the predetermined learning algorithm, but it also tends to be more computationally expensive (Langley, 1994). When the number of features becomes very large, the filter model is usually a choice due to its computational efficiency.

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.

we propose an efficient clustering based subset selection algorithm of high dimensional data. This 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 has a high probability of producing a subset of useful and independent features. The proposed feature subset selection algorithm has been tested upon 5 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.

2. FEATURE SUBSET SELECTION ALGORITHM

2.1 Framework and definitions

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 (shown in Fig.1) 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 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.

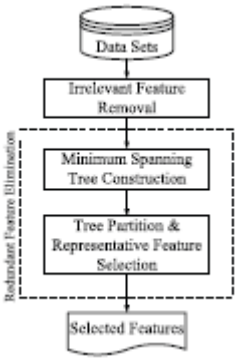

Fig. 1: Framework of the proposed feature subset selection Algorithm

Definition *1: (Relevant feature)* Fi is relevant to the target concept C if and only if there exists some s'_i, f_i and c, such that, for probability $p(S'_i = s'_i, F_i = f_i) > 0$,
$p(C = c \mid S'_i = s'_i, F_i = f_i) \neq p(C = c \mid S'_i = s'_i)$.

Otherwise, feature F_i is an *irrelevant feature*.

Definition 1 indicates that there are two kinds of relevant features due to different S'_i, (i) when $S'_i = S_i$, from the definition we can know that F_i is directly relevant to the target concept; (ii) when $S'_i \subseteq S_i$, from the definition we may obtain that $p(C|S_i, F_i) = p(C|S_i)$. It seems that F_i is irrelevant to the target concept. However, the definition shows that feature F_i is relevant when using $S'_i \cup \{F_i\}$ to describe the target concept. The reason behind is that either Fi is interactive with S'_i or F_i is redundant with $S_i - S'_i$. In this case, we say F_i is indirectly relevant to the target concept.

Most of the information contained in redundant features is already present in other features. As a result, redundant features do not contribute to getting better interpreting ability to the target concept. It is formally defined by Yu and Liu based on Markov blanket. The definitions of Markov blanket and redundant feature are introduced as follows, respectively.

Definition *2: (Markov blanket)* Given a feature $F_i \in F$, let $M_i \subset F$ ($F_i \notin M_i$), M_i is said to be a Markov blanket for F_i if and only if
$p(F - M_i - \{Fi\}, C \mid F_i, M_i) = p(F - M_i - \{F_i\}, C \mid Mi)$.

Definition *3: (Redundant feature)* Let S be a set of features, a feature in S is redundant if and only if it has a Markov Blanket within S. 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.

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 by a number of researchers. Therefore, we choose symmetric uncertainty as the measure of correlation between either two features or a feature and the target concept.

The *symmetric uncertainty* is defined as follows :
$SU(X, Y) = 2 \times Gain(X|Y) / (H(X) + H(Y))$ (1)
.
Where

1) $H(X)$ is the entropy of a discrete random variable X. Suppose $p(x)$ is the prior probabilities for all values of X, $H(X)$ is defined by $H(X) = -\Sigma\, p(x) \log_2 p(x) \quad \forall x \in X$. (2)

2) $Gain(X|Y)$ is the amount by which the entropy of Y decreases. It reflects the additional information about Y provided by X and is called the information gain which is given by $Gain(X|Y) = H(X) - H(X|Y)$
$= H(Y) - H(Y|X)$. (3)

Where $H(X|Y)$ is the conditional entropy which quantifies the remaining entropy (i.e. uncertainty) of a random variable X given that the value of another random variable Y is known. Suppose $p(x)$ is the prior probabilities for all values of X and $p(x \mid y)$ is the posterior probabilities of X given the values of Y, $H(X \mid Y)$ is defined by
$H(X \mid Y) = -\Sigma p(y) \Sigma\, p(x/y) \log_2 p(x/y) \; \forall y \in Y, \; \forall x \in X$. (4)

Information gain is a symmetrical measure. That is the amount of information gained about X after observing Y is equal to the amount of information gained about Y after observing X. This ensures that the order of two variables (e.g.,(X, Y) or (Y,X)) will not affect the value of the measure.

Symmetric uncertainty treats a pair of variables symmetrically, it compensates for information gain's bias toward variables with more values and normalizes its value to the range [0,1]. A

value 1 of $SU(X, Y)$ indicates that knowledge of the value of either one completely predicts the value of the other and the value 0 reveals that X and Y are independent. Although the entropy based measure handles nominal or discrete variables, they can deal with continuous features as well, if the values are discretized properly in advance.

Given $SU(X, Y)$ the symmetric uncertainty of variables X and Y, the relevance *T-Relevance* between a feature and the target concept C, the correlation *FCorrelation* between a pair of features, the feature redundance *F-Redundancy* and the representative feature *R-Feature* of a feature cluster can be defined as follows.

***Definition** 4: (T-Relevance)* The relevance between the feature $Fi \in F$ and the target concept C is referred to as the *T-Relevance* of F_i and C, and denoted by $SU(Fi, C)$. If $SU(F_i, C)$ is greater than a predetermined threshold θ, we say that F_i is a strong *T-Relevance* feature.

***Definition** 5: (F-Correlation)* The correlation between any pair of features F_i and F_j ($F_i, F_j \in F \wedge i \neq j$) is called the *F-Correlation* of F_i and F_j, and denoted by $SU(F_i, F_j)$.

***Definition** 6: (F-Redundancy)* Let $S = \{F_1, F_2, ..., F_i, ..., F_{k<|F|}\}$ be a cluster of features. if $\exists F_j \in S$, $SU(F_j, C) \geq SU(F_i, C) \wedge SU(F_i, F_j) > SU(F_i, C)$ is always corrected for each $F_i \in S$ ($i \neq j$), then F_i are redundant features with respect to the given F_j (i.e. each F_i is a *F-Redundancy*).

***Definition** 7: (R-Feature)* A feature $F_i \in S = \{F_1, F_2, ..., F_k\}$ ($k < |F|$) is a representative feature of the cluster S (i.e. F_i is a *R-Feature*) if and only if, $F_i = \text{argmax}_{Fj \in S} SU(F_j, C)$.

This means the feature, which has the strongest *TRelevance*, can act as a *R-Feature* for all the features in the cluster. According to the above definitions, feature subset selection can be the process that identifies and retains the strong *T-Relevance* features and selects *R-Feature*s from feature clusters. The behind heuristics are that

1) irrelevant features have no/weak correlation with target concept;
2) redundant features are assembled in a cluster and a representative feature can be taken out of the cluster

2.2 Algorithm and Analysis

The proposed algorithm logically consists of tree steps: (i) removing irrelevant features, (ii) constructing a MST from relative ones, and (iii) partitioning the MST and selecting representative features.

For a data set D with m features $F = \{F_1, F_2, ..., F_m\}$ and class C, we compute the *T-Relevance* $SU(F_i, C)$ value for each feature F_i ($1 \leq i \leq m$) in the first step. The features whose $SU(F_i, C)$ values are greater than a predefined threshold θ comprise the target-relevant feature subset $F' = \{F'_1, F'_2, ..., F'_k\}$ ($k \leq m$).

In the second step, we first calculate the *F-Correlation* $SU(F'_i, F'_j)$ value for each pair of features F'_i and F'_j ($F'_i, F'_j \in F' \wedge i \neq j$). Then, viewing features F'_i and F'_j as vertices and $SU(F'_i, F'_j)$ ($i \neq j$) as the weight of the edge between vertices F'_i and F'_j, a weighted complete graph $G = (V,E)$ is constructed where $V = \{F'_i \mid F'_i \in F' \wedge i \in [1, k]\}$ and $E = \{(F'_i, F'_j) \mid (F'_i, F'_j \in F' \wedge i, j \in [1, k] \wedge i \neq j\}$. As symmetric uncertainty is symmetric further the *F-Correlation* $SU(F'_i, F'_j)$ is symmetric as well, thus G is an undirected graph.

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 interweaved. Moreover, the decomposition of complete graph is NP-hard. Thus for graph G, we build a MST, which connects all vertices such that the sum of the weights of the edges is the minimum, using the well-known Prim algorithm.

The weight of edge (F'_i, F'_j) is *F-Correlation* $SU(F'_i, F'_j)$. After building the MST, in the third step, we first remove the edges $E = \{(F'_i, F'_j) \mid (F'_i, F'_j \in F' \wedge i, j \in [1, k] \wedge i \neq j\}$, whose weights are smaller than both of the *T-Relevance* $SU(F'_i, C)$ and $SU(F'_j, C)$, from the MST. Each deletion results in two disconnected trees T_1 and T_2. Assuming the set of vertices in any one of the final trees to be $V(T)$, we have the property that for each pair of vertices $(F'_i, F'_j \in V(T))$, $SU(F'_i, F'_j) \geq SU(F'_i, C)$ or $SU(F'_i, F'_j) \geq SU(F'_j, C)$ always holds.

From Definition 6 we know that this property guarantees the features in $V(T)$ are redundant. 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 $V(T2)$. Each cluster is a MST as well. Take $V(T_1)$ as an example. From Fig.2 we know that $SU(F_0, F_1) > SU(F_1, C)$, $SU(F_1, F_2) > SU(F_1, C) \wedge SU(F_1, F_2) > SU(F_2, C)$, $SU(F_1, F_3) > SU(F_1, C) \wedge SU(F_1, F_3) > SU(F_3, C)$.

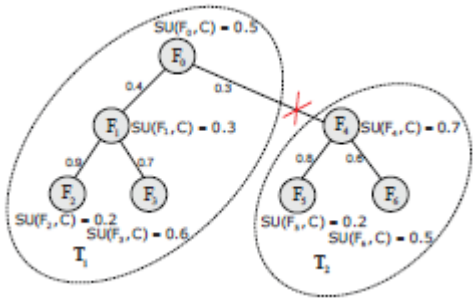

Fig. 2: Example of the clustering step

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, $SU(F_0, F_2) > SU(F_0, C) \Lambda SU(F_0, F_2) > SU(F_2, C)$, $SU(F_0, F_3) > SU(F_0, C) \Lambda SU(F_0, F_3) > SU(F_3, C)$, and $SU(F_2, F_3) > SU(F_2, C) \Lambda SU(F_2, F_3) > SU(F_3, C)$ also hold. As the mutual information between any pair $(F_i, F_j)(i, j = 0, 1, 2, 3 \Lambda i \neq j)$ 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.

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 whose *T-Relevance* $SU(F'_R, C)$ is the greatest. All F'_R (j = 1...|*Forest*|) comprise the final feature subset $\cup F'_R$.

The details of the algorithm is shown in Algorithm 1.

```
input:  S(f_1, f_2, ..., f_N, C)  // a training data se
        δ                         // a predefined thres
output: S_best                    // an optimal subset
1   begin
2     for i = 1 to N do begin
3       calculate SU_{i,c} for f_i;
4       if (SU_{i,c} ≥ δ)
5         append f_i to S'_list;
6     end;
7     order S'_list in descending SU_{i,c} value;
8     f_p = getFirstElement(S'_list);
9     do begin
10      f_q = getNextElement(S'_list, f_p);
11      if (f_q <> NULL)
12        do begin
13          f'_q = f_q;
14          if (SU_{p,q} ≥ SU_{q,c})
15            remove f_q from S'_list;
16            f_q = getNextElement(S'_list, f'_q);
17          else f_q = getNextElement(S'_list, f_q)
18        end until (f_q == NULL);
19      f_p = getNextElement(S'_list, f_p);
20    end until (f_p == NULL);
21  S_best = S'_list;
```

Fig.3 : Algorithm.1 of proposed model.

2.3 Time Complexity

Based on the methodology presented before, we develop an efficient algorithm, as given in Figure 3, given a data set with N features and a class C, the algorithm finds a set of predominant features S_{best} for the class concept. It consists of two major parts. In the first part (line 2-7), it calculates the SU value for each feature, selects relevant features into S_0 list based on the predefined threshold δ, and orders them in descending order according to their SU values. In the second part (line 8-20), it further processes the ordered list S_0 list to remove redundant features and only keeps predominant ones among all the selected relevant features. According to Heuristic 1, a feature f_p that has already been determined to be a predominant feature can always be used to filter out other features that are ranked lower than f_p and have f_p as one of its redundant peers. The iteration starts from the first element (Heuristic 3) in S_0 list (line 8) and continues as follows. For all the remaining features (from the one right next to f_p to the last one in S_0 list), if f_p happens to be a redundant peer to a feature f_q, f_q will be removed from S_0 list (Heuristic 2). After one round of filtering features based on f_p, the algorithm will take the currently remaining feature right next to f_p as the new reference (line 19) to repeat the filtering process. The

algorithm stops until there is no more feature to be removed from S_0 list.

The first part of the above algorithm has a linear time complexity in terms of the number of features N. As to the second part, in each iteration, using the predominant feature f_p identified in the previous round, the algorithm can remove a large number of features that are redundant peers to f_p in the current iteration. The best case could be that all of the remaining features following f_p in the ranked list will be removed; the worst case could be none of them. On average, we can assume that half of the remaining features will be removed in each iteration. Therefore, the time complexity for the second part is O(N logN) in terms of N. Since the calculation of SU for a pair of features is linear in term of the number of instances M in a data set, the overall complexity of the alogirhtm is O(MN logN).

3. RESULTS AND ANALYSIS

In this section we present the experimental results in terms of the proportion of selected features, the time to obtain the feature subset, the classification accuracy, and the Win/Draw/Loss record. For the purpose of exploring the statistical significance of the results, we performed a nonparametric Friedman test followed by Nemenyi post-hoc test, as advised by Demsar and Garcia and Herrerato to statistically compare algorithms on multiple data sets. Thus the Friedman and the Nemenyi test results are reported as well.

3.1 Proportion of selected features

Table 2 records the proportion of selected features of the six feature selection algorithms for each data set. From it we observe that

1) Generally all the six algorithms achieve significant reduction of dimensionality by selecting only a small portion of the original features. This algorithm on average obtains the best proportion of selected features of 1.82%. The Win/Draw/Loss records show this algorithm wins other algorithms as well.

2) For image data, the proportion of selected features of each algorithm has an increment compared with the corresponding average proportion of selected features on the given data sets except Consist has an improvement. This reveals that the five algorithms are not very suitable to choose features for image data compared with for microarray and text data.

Data set	This	Proportion of selected features (%) of				
		FCBF	CFS	ReliefF	Consist	FOCUS-SF
chess	16.22	21.62	10.81	62.16	81.08	18.92
mfeat-fourier	19.48	49.35	24.68	98.70	15.58	15.58
coil2000	3.49	8.14	11.63	50.00	37.21	1.16
elephant	0.86	3.88	5.60	6.03	0.86	0.86
arrhythmia	2.50	4.64	9.29	50.00	8.93	8.93
fqs-nowe	0.31	2.19	5.63	26.56	4.69	4.69
colon	0.30	0.75	1.35	39.13	0.30	0.30
fbis.wc	0.80	1.45	2.30	0.95	1.75	1.75
AR10P	0.21	1.04	2.12	62.89	0.29	0.29
PIE10P	1.07	1.98	2.52	91.00	0.25	0.25
oh0.wc	0.38	0.88	1.10	0.38	1.82	1.82
oh10.wc	0.34	0.80	0.56	0.40	1.61	1.61
B-cell1	0.52	1.61	1.07	30.49	0.10	0.10
B-cell2	1.66	6.13	3.85	96.87	0.15	0.15
B-cell3	2.06	7.95	4.20	98.24	0.12	0.12
base-hock	0.58	1.27	0.82	0.12	1.19	1.19
TOX-171	0.28	1.41	2.09	64.60	0.19	0.19
tr12.wc	0.16	0.28	0.26	0.59	0.28	0.28
tr23.wc	0.15	0.27	0.19	1.46	0.21	0.21
tr11.wc	0.16	0.25	0.40	0.37	0.31	0.31
embryonal-tumours	0.14	0.03	0.03	13.96	0.03	0.03
leukemia1	0.07	0.03	0.03	41.35	0.03	0.03
leukemia2	0.01	0.41	0.52	60.63	0.08	0.08
tr21.wc	0.10	0.22	0.37	2.04	0.20	0.20
wap.wc	0.20	0.53	0.65	1.10	0.41	0.41
PIX10P	0.15	3.04	2.35	100.00	0.03	0.03
ORL10P	0.30	2.61	2.76	99.97	0.04	0.04
CLL-SUB-111	0.04	0.78	1.23	54.35	0.08	0.08
ohscal.wc	0.34	0.44	0.18	0.03	NA	NA
la2s.wc	0.15	0.33	0.54	0.09	0.37	NA
la1s.wc	0.17	0.35	0.51	0.06	0.34	NA
GCM	0.13	0.42	0.68	79.41	0.06	0.06
SMK-CAN-187	0.13	0.25	NA	14.23	0.06	0.06
new3s.wc	0.10	0.15	NA	0.03	NA	NA
GLA-BRA-180	0.03	0.35	NA	53.06	0.02	0.02
Average(Image)	3.59	10.04	6.68	79.85	3.48	3.48
Average(Microarry)	0.71	2.34	2.50	52.92	0.91	0.91
Average(Text)	2.05	3.25	2.64	10.87	11.46	2.53
Average	1.82	4.27	3.42	42.54	5.44	2.06
Win/Draw/Loss	-	33/0/2	31/0/4	29/1/5	20/2/13	19/2/14

This algorithm ranks 3 with the proportion of selected features of 3.59% that has a tiny margin of 0.11% to the first and second best proportion of selected features 3.48% of Consist and FOCUS-SF, and a margin of 76.59% to the worst proportion of selected features 79.85% of ReliefF.

3) For microarray data, the proportion of selected features has been improved by each of the six algorithms compared with that on the given data sets. This indicates that the six algorithms work well with microarray data. This algorhtm ranks 1 again with the proportion of selected features of 0.71%. Of the six algorithms, only CFS cannot choose features for two data sets whose dimensionalities are 19994 and 49152, respectively.

4) For text data, FAST ranks 1 again with a margin of 0.48% to the second best algorithm FOCUS-SF.

4. CONCLUSIONS AND FUTURE WORK

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 this algorithm tobtains 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.

ACKNOWLEDGEMENT

We would like to express our sincere thanks to Sri. Dr. Kancharla Ramaiah Secretary and Correspondent, Prakasam Engineering College, Kandukur, A.P. India for his support with providing research environment . We are extremely thankful to our collogues , friends and family members who are cooperated in this work.

REFERENCES

[1] Bay, S. D. (1999). The UCI KDD Archive. http://kdd.ics.uci.edu.

[2] Blake, C., & Merz, C. (1998). UCI repository of machine learning databases. http://www.ics.uci.edu/_mlearn/MLRepository.html.

[3] Blum, A., & Langley, P. (1997). Selection of relevant features and examples in machine learning. Artificial Intelligence, 97, 245–271.

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

[5] Das, S. K. (1971). Feature selection with a linear dependence measure. IEEE Transactions on Computers.

[6] Dash, M., & Liu, H. (1997). Feature selection for classifications. Intelligent Data Analysis: An International Journal, 1, 131–156.

[7] Dash, M., Liu, H., & Motoda, H. (2000). Consistency based feature selection. Proceedings of the Fourth Pacific Asia Conference on Knowledge Discovery and Data Mining (pp. 98–109). Springer-Verlag.

[8] Fayyad, U., & Irani, K. (1993). Multi-interval discretization of continuous-valued attributes for classification learning. Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence (pp. 1022–1027). Morgan Kaufmann.

[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.

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

[18] Dhillon I.S., Mallela S. and Kumar R., A divisive information theoretic feature clustering algorithm for text classification, J. Mach. Learn. Res., 3, pp 1265-1287, 2003.

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

[20] 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.