Pattern Recognition Letters 40 (2014) 104–112

Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

Integration of dense subgraph finding with feature clustering for unsupervised feature selection ☆

Sanghamitra Bandyopadhyay [a,*], Tapas Bhadra [a], Pabitra Mitra [b], Ujjwal Maulik [c]

[a] Machine Intelligence Unit, Indian Statistical Institute, Kolkata 700108, India
[b] Department of Computer Science and Engineering, Indian Institute of Technology, Kharagpur 721302, India
[c] Department of Computer Science and Engineering, Jadavpur University, Kolkata 700032, India

ARTICLE INFO

Article history:
Received 15 May 2013
Available online 15 December 2013

Keywords:
Pattern recognition
Unsupervised feature selection
Mutual information
Normalized mutual information

ABSTRACT

In this article a dense subgraph finding approach is adopted for the unsupervised feature selection problem. The feature set of a data is mapped to a graph representation with individual features constituting the vertex set and inter-feature mutual information denoting the edge weights. Feature selection is performed in a two-phase approach where the densest subgraph is first obtained so that the features are maximally non-redundant among each other. Finally, in the second stage, feature clustering around the non-redundant features is performed to produce the reduced feature set. An approximation algorithm is used for the densest subgraph finding. Empirically, the proposed approach is found to be competitive with several state of art unsupervised feature selection algorithms.

© 2013 Elsevier B.V. All rights reserved.

1. Introduction

Over the past decade pattern recognition techniques have been extensively used to solve several real-life problems that involve very high dimensional data. Dimensionality reduction is almost always necessary to remove the redundant features while retaining the salient characteristics of the data as far as possible (Kwak and Choi, 2002).

Feature selection algorithms can be divided into two categories based on the feature evaluation methodology, namely, filter and wrapper methods (Dash and Liu, 1997). In the filter approaches, a candidate feature subset is evaluated at each iteration based on certain statistical measures. Some known filter type approaches are based on *t*-test (Hua et al., 2008), chi-square test (Jin et al., 2006), Wilcoxon Mann–Whitney test (Liao et al., 2007), mutual information (Battiti, 1994; Kwak and Choi, 2002; Peng et al., 2005; Estévez et al., 2009; Vinh et al., 2010), Pearson correlation coefficients (Biesiada and Duch, 2008), etc. On the other hand, wrapper methods utilize the performance of a classifier as the evaluation criteria for measuring the goodness of a candidate feature subset (Kohavi and John, 1997).

Based on the availability of class labels, feature selection algorithms can also be classified in two ways, namely, supervised and unsupervised feature selection. Supervised feature selection is generally employed when the class information are in hand, otherwise unsupervised approach is used. Most known filter type approaches, belong to the category of supervised learning. On the other hand, a limited number of researches have been conducted in the field of unsupervised feature selection. Unsupervised feature selection using feature similarity measure (FSFS) (Mitra et al., 2002), Laplacian Score for Feature Selection (LSFS) (He et al., 2005), SPectral Feature Selection (SPFS) (Zhao and Liu, 2007), Multi Cluster Feature Selection (MCFS) (Cai et al., 2010), Unsupervised Discriminative Feature Selection (UDFS) (Yang et al., 2011), etc. are some existing algorithms in this domain.

Feature selection is inherently a combinatorial optimization problem (Kohavi and John, 1997). Conventional feature selection methods usually follow a greedy approach and choose top-ranking features on an individual level. This ignore the mutual dependency among the selected features. As a result of this, the optimal feature subset is sometimes difficult to find. The above mentioned five unsupervised feature selection algorithms except MCFS and UDFS follow the same methodology for obtaining the reduced feature set.

We attempt to incorporate the combinatorial effect, by adopting a graph theoretic approach utilising the notion of densest subgraph. The subgraph finding task is a known problem for a diverse number of applications like community mining, web mining, computational biology (Bahmani et al., 2012). Densest subgraph finding is a NP-hard problem. Recently, approximation algorithms for finding the densest subgraph have been devised in literature (Bahmani et al., 2012). Finding a subset of representative features by mining dense subgraph has also been addressed in Liu et al.

☆ This paper has been recommended for acceptance by S. Sarkar.

* Corresponding author. Tel.: +91 33 2575 3114; fax: +91 33 2578 3357.
E-mail addresses: sanghami@isical.ac.in (S. Bandyopadhyay), tapas.bhadra@isical.ac.in (T. Bhadra), pabitra@cse.iitkgp.ernet.in (P. Mitra), umaulik@cse.jdvu.ac.in (U. Maulik).

0167-8655/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.patrec.2013.12.008

(2011) and Mandal and Mukhopadhyay (2013). Liu et al. (2011) proposed a supervised method for obtaining the most informative features while Mandal and Mukhopadhyay (2013) used an unsupervised approach for obtaining the minimally redundant features. Here we have developed a new unsupervised feature selection technique based on the principle of densest subgraph finding followed by feature clustering.

We first obtain a graph representation by considering the entire feature set as the vertex set and having the inter-feature similarity as the corresponding edge weight. Here, the inter-feature similarity is computed using a normalized form of mutual information.

The densest subgraph finding approach has one major advantage that the vertices of this densest subgraph, i.e., the features of the reduced feature set, will be highly dissimilar. However, it is likely that these features may not be the optimal feature set. The reason behind this is that these features may not be the best representatives of the features that have been excluded, even though they are highly dissimilar to each other. To overcome this situation, a clustering approach is further applied on this densest subgraph for obtaining a better subgraph so that no important feature can be excluded from this set. The variance is used in the clustering phase to select the prototype feature while the same normalized mutual information is utilized for assigning each non-selected feature into its closest cluster representative. The subgraph thus obtained essentially contains a subset of the original features that can maximally represent the entire feature space. Thus our approach proceeds in a two-phase manner in which the first phase deals with finding out the densest subgraph while clustering the subgraph is performed in the second.

The remaining part of the paper is organized as follows: Section 2 discusses some preliminary concepts following which some of the existing unsupervised feature selection algorithms are discussed in Section 3. The proposed two-phase unsupervised feature selection algorithm is described in Section 4. Subsequently, the experiential design and the comparative results are provided in Section 5. Finally, some concluding comments are made in Section 6.

2. Preliminary concepts

This section describes some fundamental information and graph theory measures.

2.1. Density of a subgraph

Let $G = (V, E)$ be an unweighted undirected graph. The density of a subgraph $S \subseteq V$, denoted as $d(S)$, is defined as $d(S) = \frac{|E(S)|}{|S|}$, where $E(S)$ is the induced edge set of the subgraph S and $|S|$ is the cardinality of S.

The maximum density of the graph, denoted as $d^*(G)$, is defined as $d^*(G) = \max_{S \subseteq V}\{d(S)\}$. Similarly, the density of a subgraph $S \subseteq V$ within a weighted graph $G = (V, E)$ can also be defined as $d(S) = \frac{\sum_{e \in E(S)} w_e}{|S|}$, where $E(S)$ is the induced edge set of the subgraph S and w_e is the weight of the edge $e \in E(S)$.

2.2. Mutual information measures

2.2.1. Entropy

Entropy of a random variable is the amount of uncertainty associated with it (Cover and Thomas, 2012). The entropy of a discrete variable X, denoted by $H(X)$, is defined as

$$H(X) = -\sum_{x \in X} p(x) \log_b p(x), \tag{1}$$

where $p(x)$ indicates the probability mass function of X. The value of b is generally assumed to be 2.0 and this value is used in the present paper.

2.2.2. Mutual information

Mutual information between two random variables measures how much information can be extracted through the knowledge of the other (Cover and Thomas, 2012). The value of mutual information becomes zero when the associated variables are completely independent whereas its higher value signifies their high mutual dependency. The mutual information between two discrete variables X and Y, denoted as $I(X;Y)$, is defined as follows

$$I(X;Y) = \sum_{x \in X}\sum_{y \in Y} p(x,y) \log_b \left(\frac{p(x,y)}{p(x)p(y)} \right), \tag{2}$$

where $p(x), p(y)$ and $p(x,y)$ denote the probability mass function of X, the probability mass function of Y and the joint probability mass function between X and Y, respectively.

2.2.3. Normalized mutual information

Mutual information has a disadvantage due to its non-comparability among variable pairs that have different mutual information values in various ranges. To overcome this, mutual information is often normalized into a closed interval, say $[0, 1]$.

Several researchers have used various methods to construct normalized mutual information. A few of them are mentioned below

$$\tilde{I}(X,Y) = \frac{2I(X;Y)}{H(X) + H(Y)}, \tag{3}$$

$$\hat{I}(X,Y) = \frac{I(X;Y)}{\min(H(X), H(Y))}, \tag{4}$$

$$I'(X,Y) = \frac{I(X;Y)}{\sqrt{H(X)H(Y)}}. \tag{5}$$

Witten and Frank (2005) proposed the first one, known as symmetric uncertainty in the form of the weighted average of the two uncertainty coefficients. Strehl and Ghosh (2002) favoured the third form over the second one for ensembling several clusters due to the closeness to a normalized inner product in Hilbert space.

3. Review of unsupervised feature selection

Many of the earlier feature selection algorithms are based on supervised learning. Among the unsupervised feature selection approaches, data variance is the simplest measure for evaluating the discriminating power of a feature.

In the context of unsupervised feature selection algorithm, FSFS, proposed by Mitra et al. (2002), is a popular one. In this work, Mitra et al. (2002) proposed a new similarity measure, known as Maximal Information Compression Index (MICI) that was used to iteratively remove some number of features, say k, decrementing k until no removal was possible. The MICI between two variables x and y, denoted by $\lambda_2(x,y)$, was defined as follows

$$\lambda_2(x,y) = (var(x) + var(y)) - \sqrt{(var(x) + var(y))^2 - 4\,var(x)\,var(y)(1 - \rho(x,y)^2)}, \tag{6}$$

where $var(x)$, $var(y)$ and $\rho(x,y)$ denote the variance of x, the variance of y, and the correlation coefficient between x and y, respectively.

A benefit of the approach is that it does not require any search which in turn makes the selection problem fast. However, this

approach has a major drawback regarding choosing the proper value for k.

Laplacian Score for Feature Selection (LSFS) is another feature selection algorithm that is designed for serving both supervised and unsupervised learning (He et al., 2005). Like any other filter type approach, LSFS selects some top-ranking features that have maximum locality preserving power computed in terms of Laplacian score. The motivation behind LSFS is that two closest data points are likely to be in the same class. The underlying idea for this is that the local structure of the data is given more priority over the global structure for several classification problems like K-NN rule, etc.

SPectral Feature Selection (SPFS), designed using spectral graph theory, is one of the first research works where a general framework of feature selection is proposed for both supervised as well as unsupervised learning. This work deals with capturing the structural information of a graph from the corresponding spectrum. In this study, the spectrum of the graph is used to measure the feature relevance. Furthermore, two existing feature selection algorithms, namely, ReliefF (supervised) (Kononenko, 1994) and LSFS (unsupervised), have been derived as a special case of this framework.

All of these algorithms consider the feature importance individually and finally choose some user defined number of features as the reduced feature set. The main problem of these approaches is that the mutual relationship among the selected features is not modeled adequately. To overcome this problem, MCFS is designed by incorporating spectral clustering analyzes of the data (manifold learning) with $L1$-regularized models (Cai et al., 2010). The main motivation of using spectral analysis technique here is to efficiently compute the correlations among different features of a candidate feature subset in an unsupervised manner. So identifying the multi-cluster data structure is a major advantage of this approach. Here the optimization problem related to feature selection is effectively solved by employing a sparse eigen-problem as well as a $L1$-regularized least squares problem.

Unsupervised Discriminative Feature Selection (UDFS) is a very recently proposed unsupervised feature selection algorithm based on the joint effect of discriminative analysis and $L_{2,1}$-norm minimization. Similar to MCFS, UDFS also analyzes features collectively in a batch mode. Additionally, UDFS exploits the discriminative power of a feature set along with considering the local structure of data distribution.

4. Proposed feature selection technique

We attempt to model the feature selection and dependency modeling problem using a graph theoretic representation. For this purpose, we have mapped the given feature set into its equivalent graph $G = (V, E, W_F)$, where V is the set of features, E is the set of edges between the feature pairs and $W_F : e_i \rightarrow \mathbb{R}$ indicates the mutual redundancy between two features connected by edge e_i. In the present work we have used the variant of normalized mutual information (NMI), defined by Strehl and Ghosh (2002), to measure the inter-feature redundancy. The main intuition for representing the feature selection problem into an equivalent graph notation is to apply the existing densest subgraph finding approach for finding a good feature set in optimal time. Moreover, we will be able to obtain an optimal densest feature subset, the features of which will be minimally redundant with each other.

Recently densest subgraph finding problem has attracted a great deal of attention for obtaining a smaller subset of vertices that has the highest ratio of the number of edges to the number of vertices. The above problem is a natural mapping of the maximally independent feature subset finding task. Selected feature subset, however, needs to address the criterion of 'representing' the non-selected features in addition to being maximally independent. To address this issue we have proposed a two-phase approach so that one can get a subset of original features that are not only highly dissimilar to each other but also hold sufficient similarity with respect to the non-selected features. We first describe the dense subgraph finding algorithm that we have used.

4.1. Approximate dense subgraph finding algorithm

Dense subgraph finding either for directed or undirected graph is a hard problem that lies at the centre of very large-scale graph algorithms (Bahmani et al., 2012). Several researchers have made valuable contributions towards a good approximation to this problem. Recently Bahmani et al. (2012) have provided three approximation algorithms in which the first two deal with the densest subgraph finding approach for directed as well as undirected graphs without any size constraint while the last one, Densest At Least k Subgraph (DALS), is applicable for the graph with a size constraint k (Bahmani et al., 2012). They have proved that each of the first two algorithms leads a $(2 + 2\epsilon)$-approximation whereas the last one DALS is a $(3 + 3\epsilon)$-approximation. These algorithms have been tested on large scale graph with more than half billion vertices and six billion edges signifying the scalability of their algorithms.

4.2. Two stage feature selection algorithm

Recently Mandal and Mukhopadhyay (2013) proposed a graph theoretic approach for solving the unsupervised feature selection problem. The outcome of this approach produces a dense feature subset, the features of which are maximally non-redundant to each other. However they did not verify whether the reduced features are the optimal representatives of the features that have been excluded. To address this issue, we have integrated the densest subgraph finding approach with feature clustering in this paper.

The detailed two-phase feature selection algorithm, named as Dense Subgraph Finding with Feature Clustering (DSFFC), is provided in Algorithm 1. Steps 1–18 constitute the first phase, i.e., finding the densest subgraph of size at least k, while steps 19–25 form the second phase, i.e., feature clustering for obtaining the reduced features. The input of the first phase is a graph representation $G = (V, E, W_F)$ of the dataset and three user parameters k, l, r denoting the minimum size of the reduced feature set, the number of features that needs to be inserted at each iteration and the number of features that needs to be discarded at each iteration, respectively. The main objective of this phase is to find out a subset of vertices $R \subseteq V$ having size at least k whose average density is the minimum. For this purpose, we first assign the set V to two other sets, say, S and R. Then, the vertex i from the set S is put into the $A'(S)$ provided the induced degree of the vertex i in the induced edge set $E(S)(= E \cap S^2)$, denoted by $deg_S(i)$, is greater than or equal to twice the density of the set S, denoted by $d(S)$. Here density is computed (as described in Section 2.1) by using only the edge weights of the set S where the edge weights are measured in terms of Í (X,Y) as defined in Eq. (5) in Section 2.2.3. Then we follow different strategies depending upon the cardinality value of the set $A'(S)$. If the cardinality value equals to 0 we stop the first phase and go for the second phase. If the cardinality value becomes 1 we explicitly set the value of r to be 1; otherwise we set r equals to be half of the cardinality value. Then we rearrange the elements of the set $A'(S)$ based on decreasing values of the degrees of all the vertices and remove the top-ranking r vertices. Afterwards we check the two conditions: (i) whether the cardinality of the set S is greater than or equal to k and (ii) whether density of the set S

is less than that of the set R. If both the conditions become true, then we replace the set R by S. As an important feature that is removed at an earlier stage may join at a later phase of the algorithm, we check this condition at the end of each iteration and update the set S accordingly. Basically, we have incorporated the advantage of so called l-r principle of feature selection in the core stage of the algorithm. In this way steps 1–18 constitute the first phase of the algorithm which provides a subset of at least k features. These constitute the k' ($k' \geqslant k$) prototype features for the second phase, i.e., feature clustering. In the second phase, k' clusters are first created by using these k' prototype features. All the other non-selected features are put into their nearest cluster (by using maximal similarity measure in terms of NMI). Next, each cluster prototype is replaced by a feature whose variance is the maximum among those belonging to the same cluster. The above two-steps are repeated until no further change occurs in the cluster structure or the prototype elements.

Algorithm 1. DSFFC

Input: Graph $G = (V, E, W_F)$; Parameters $k > 0, l \geqslant 0$ and $r > 0$.
Output: R be the resultant reduced feature set.
Algorithm:
Step 1: Set $S \leftarrow V, R \leftarrow V$;
Step 2: while $S \neq 0$ do
Step 3: $A'(S) \leftarrow \{i \in S | deg_S(i) \geqslant 2 * d(S)\}$;
Step 4: if $|A'(S)| = 0$ then
Step 5: goto **Step 19**;
Step 6: else if $|A'(S)| = 1$ then
Step 7: $r = 1$;
Step 8: else if $|A'(S)| < r$ then
Step 9: $r = 0.5 * |A'(S)|$;
Step 10: end if
Step 11: Arrange $A'(S)$ in descending order based on $deg_S(i), i \in A'(S)$;
Step 12: Assign top-ranking r features from $A'(S)$ into $A(S)$;
Step 13: $S \leftarrow S \setminus A(S)$;
Step 14: if $|S| \geqslant k$ and $d(S) < d(R)$ then
Step 15: $R \leftarrow S$;
Step 16: end if
Step 17: Set $S \leftarrow S \cup S_l$ if $d(S \cup S_l) < d(S), S_l \cap S \neq 0$;
Step 18: end while
Step 19: $k' = |R|$;
Step 20: Set $p_j = R_j$, where R_j is the j-th element (feature) of $R, \forall j = 1, \ldots, k'$;
Step 21: Let p_j be initial center corresponding to j-th cluster $C_j, \forall j = 1, \ldots, k'$;
Step 22: Associate each non-selected feature $f_i, i = 1, \ldots, |V|$ and $f_i \notin \{p_1, \ldots, p_{k'}\}$, to cluster $C_j, j \in \{1, \ldots, k'\}$ iff $NMI(f_i, p_j) = \max_{m=1}^{k'}(NMI(f_i, p_m))$;
Step 23: Select new prototype feature $p'_j, \forall j = 1, \ldots, k'$ such that $var(p'_j) = \max(var(f_i)), \forall f_i \in C_j$;
Step 24: If $p_j = p'_j, \forall j = 1, \ldots, k'$ then goto **Step 25** else goto **Step 22**;
Step 25: Output k' number of prototype features as the set R.

The overall schematic of the proposed two-phase approach DSFFC is illustrated in Fig. 1. First, the feature space is mapped into an equivalent graph representation (as described in Section 4) shown in Fig. 1(a). The edge weights denote the similarity values between the corresponding pair of features. In the figure, a longer (shorter) edge denotes less (more) similarity. After applying the densest subgraph approximation algorithm to this graph, a subgraph with eight features is obtained as an output of the first phase as shown in Fig. 1(b). Finally the second phase, i.e., feature clustering produces eight feature clusters as shown in Fig. 1(c). One feature for each cluster is selected as the prototype feature corresponding to that cluster.

5. Experimental results

Extensive experiments have been conducted to evaluate the proposed algorithm with respect to three existing unsupervised feature selection algorithms, namely, FSFS (Mitra et al., 2002), LSFS (He et al., 2005) and MCFS (Cai et al., 2010). For the present work, we have set the values of both the user parameters, i.e., l and r, to be 1. For all the feature selection algorithms the number of reduced features (k) has been kept to be half of the number of original features. The detailed descriptions about used datasets, used classifiers, evaluation criteria and experimental results, are mentioned below.

5.1. Used datasets

In our experimental evaluation, eight publicly available datasets have been used to show the effectiveness of the proposed algorithm. These are *Colon*, *Multiple Features*, *Isolet*, *Spambase*, *Ionosphere*, *WDBC*, *Sonar* and *SPECTF*. All of them are collected from UCI machine learning repository (Bache and Lichman, 2013). Some basic characteristics of these eight datasets are summarized in Table 1. As the features of these datasets contain values of different ranges, the datasets are normalized using max–min normalization. The main objective of taking max–min normalization over other kind of normalization such as z-score is that the former can partially preserve the information related to standard deviation while the latter one can not retain the topological structure of the datasets in many cases. For the sake of simplicity, the feature values are scaled in the [0, 1] interval. These datasets have been chosen by considering diverse characteristics of the datasets such as number of samples, number of features, number of different classes, etc. For example, *Colon* is a very high dimensional dataset with a small sample size while *Spambase* is the example of a very large sample size dataset. *Multiple Features* and *Isolet* are two multi-class datasets that have 10 and 25 different kind of classes, respectively.

5.2. Used classifiers

Four classifiers, namely, Support Vector Machines (SVM), Naive Bayes, K-nearest neighbor (KNN) and AdaBoost are used to compare the classification performance of the feature selection algorithms.

For the SVM classifier, we have used the famous RBF kernel whose performance is dependent on two user defined parameters, namely, C and γ. In our experiment, their suitable values are obtained by using a grid search done on the training data. For the KNN classifier, the value of K is set as the square root of the sample size. The second classifier Naive Bayes has one advantage of not owing any such user defined parameter. For the same reason, the Naive Bayes classifier is also employed as the underlying base classifier for the last one, i.e., Adaboost classifier. However, Adaboost also has some parameters for which the default values are considered in the present work. Corresponding to each classifier, we have run the 10-fold cross validation ten times on the training data and subsequently calculated the average results.

In the present work, LIBSVM software (Chang and Lin, 2011) is used for building the SVM classifier while the remaining three classifiers are built using WEKA tool (Hall et al., 2009).

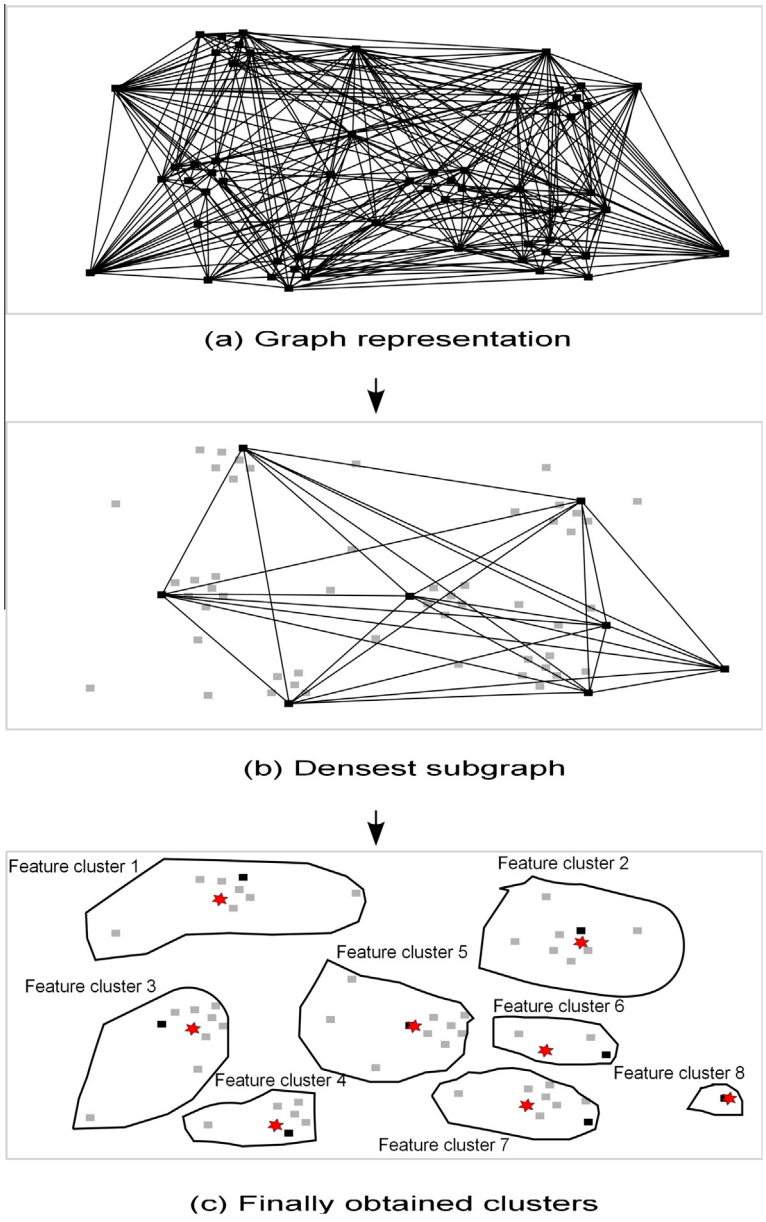

Fig. 1. Illustration of the proposed graph-based clustering algorithm for unsupervised feature selection. (a) Original set of features in a graph representation, (b) the features in the densest subgraph, and (c) the prototype features in the finally obtained clusters are marked as red stars. (For interpretation of the references to colour in this figure caption, the reader is referred to the web version of this article.)

Table 1
Characteristics of the used datasets.

Dataset	No. of Samples	No. of Features	No. of Classes
Colon	62	2000	2
Multiple Features	2000	649	10
Isolet	6238	617	26
Spambase	4601	57	2
Ionosphere	351	34	2
WDBC	569	30	2
Sonar	208	60	2
SPECTF	80	44	2

5.3. Evaluation criteria

For all the two-class datasets, the performance of each of the above mentioned four classifiers are measured using two evaluation criteria, i.e., accuracy (*Acc*) and Matthews correlation coefficient (*MCC*). On the other hand, only *Acc* values are considered for the multi-class datasets. These two criteria are obtained as follows $Acc = \frac{TP+TN}{TP+TN+FP+FN}$ and $MCC = \frac{(TP\times TN - FP\times FN)}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}}$, where TP, TN, FP and FN stand for the number of true positives, true negatives, false positives and false negatives, respectively.

In addition to the above mentioned two supervised measures, we have also computed one unsupervised measure, namely, representation entropy (RE) that helps us to identify the discriminative power of the respective feature subsets (Mitra et al., 2002). The RE of a d-size feature set, denoted by H_R, is defined as follows

$$H_R = -\sum_{j=1}^{d} \widetilde{\lambda}_j \log \widetilde{\lambda}_j,$$

where $\widetilde{\lambda}_j, j = 1, \ldots, d$, are obtained as follows

$$\widetilde{\lambda}_j = \frac{\lambda_j}{\sum_{j=1}^{d} \lambda_j}.$$

Here, $\lambda_j, j = 1, \ldots, d$, are the eigenvalues of the $d \times d$ covariance matrix of the respective feature space of size d.

The value of H_R attains a maximum value when all the eigen vectors become equally important, i.e., the level of uncertainty is maximum. On the other hand, its value equals to zero when all the eigen values except one are zero. Higher value of RE indicates better selection of features. RE is one of the most desired property to compare several feature selection algorithms.

5.4. Comparative study

We have performed a number of experiments to evaluate the usefulness of the proposed approach. First, to show the efficacy of the proposed two-phase approach over its first phase, the performance of the classifier corresponding to features selected after the first phase only and after both the phases are evaluated. These are shown in Fig. 2. It is seen from the figure that the proposed two-phase approach generally results in some improvement over only the first phase. The improvement appears to be significantly more for three datasets, namely, *Isolet*, *Spambase* and *SPECTF*. In case of the remaining five datasets, the performance of the proposed method is slightly better than the first phase except the Colon data. For the *Colon* data, the performance of first phase is found to be better than the proposed two-phase method only using SVM classifier whereas the two-phase method supersedes the single phase for the remaining three classifiers. One of the major reason behind this performance degradation is probably due to the mismatch between the number of samples with the number of features. Therefore our method guarantees that it will definitely perform well or fairly better as compared to the first phase.

Next, extensive experiments have been carried out to show the effectiveness of the proposed approach over some existing unsupervised feature selection algorithms, namely, FSFS, LSFS and MCFS. The experimental results are provided in Table 2. The table reveals that, on the *Colon* data, the proposed algorithm performs better than the other feature selection algorithms for the three classifiers, namely, SVM, Naive Bayes and AdaBoost. For KNN, MCFS performs the best while the proposed approach performs the second best, providing almost comparable result with the best one. In terms of *MCC*, the proposed method comprehensively outperforms the others for SVM, Naive Bayes and AdaBoost, while MCFS provides the best value for KNN followed by our DSFFC. For this dataset, the performance of LSFS is very poor as compared to the other three. For the second data, *Multiple Features*, all the methods provide high ($> 93\%$) and comparable accuracies, with the proposed approach performing the best for SVM as well as KNN while MCFS performs the best for the other classifiers. For the third data *Isolet*, the proposed method performs the best on two classifiers, namely, Naive Bayes and AdaBoost while MCFS beats the others on the remaining two classifiers. FSFS provides

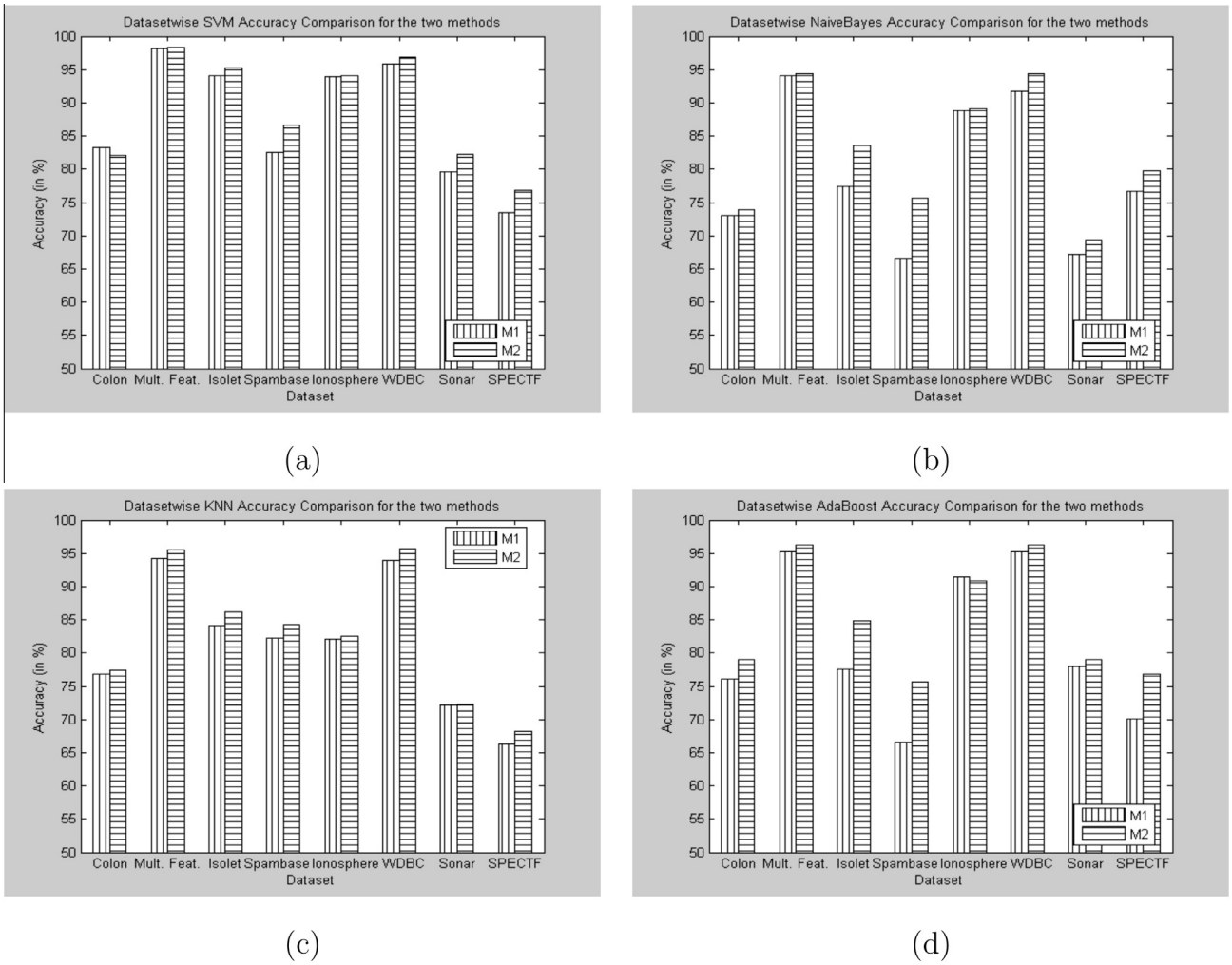

Fig. 2. Comparing the classification accuracy of the proposed feature selection method when only the first phase is used ($M1$) and both the phases are used ($M2$), corresponding to four classifiers, namely, (a) SVM, (b) Naive Bayes, (c) KNN and (d) AdaBoost.

Table 2
Performance comparison of different unsupervised feature selection algorithms on eight datasets. The mean value of ten independent runs is mentioned in each table entry while their standard deviation is shown in parentheses. The best mean values of percentage accuracy and MCC are marked in boldface.

Dataset	Algorithm	Evaluation criteria								
		SVM		Naive Bayes		KNN		AdaBoost		RE
		Acc	MCC	Acc	MCC	Acc	MCC	Acc	MCC	
Colon	FSFS	81.45(0.85)	0.585(0.02)	73.39(3.16)	0.439(0.07)	74.84(1.56)	0.439(0.044)	76.29(3.57)	0.465(0.092)	**4.06**
	LSFS	71.62(2.04)	0.336(0.061)	51.29(1.83)	0.16(0.042)	73.55(1.56)	0.406(0.052)	60.97(4.35)	0.232(0.088)	2.42
	MCFS	79.52(1.09)	0.54(0.026)	67.96(3.41)	0.347(0.076)	**78.06**(1.13)	**0.528**(0.025)	77.10(2.72)	0.495(0.056)	3.81
	DSFFC	**82.10**(1.19)	**0.600**(0.028)	**73.87**(1.67)	**0.461**(0.039)	77.42(1.32)	0.512(0.037)	**79.03**(3.48)	**0.537**(0.084)	3.94
Multiple Features	FSFS	97.91(0.11)	–	95.51(0.16)	–	94.49(0.21)	–	96.54(0.29)	–	5.43
	LSFS	97.74(0.11)	–	94.32(0.2)	–	93.02(0.2)	–	96.15(0.20)	–	4.38
	MCFS	98.13(0.13)	–	**95.59**(0.13)	–	95.58(0.13)	–	**97.06**(0.19)	–	4.59
	DSFFC	**98.35**(0.13)	–	94.43(0.12)	–	**95.61**(0.12)	–	96.22(0.17)	–	**5.52**
Isolet	FSFS	88.17(0.23)	–	65.82(0.21)	–	71.42(0.25)	–	65.78(0.19)	–	5.34
	LSFS	92.95(0.11)	–	75.49(0.27)	–	82.6(0.19)	–	75.53(0.31)	–	4.42
	MCFS	**95.75**(0.12)	–	82.09(0.33)	–	**87.99**(0.13)	–	81.99(0.21)	–	5.02
	DSFFC	95.26(0.08)	–	**83.61**(0.22)	–	86.19(0.14)	–	**84.82**(0.38)	–	**5.74**
Spambase	FSFS	78.95(0.11)	0.554(0.002)	66.68(0.10)	0.456(0.002)	80.81(0.18)	0.613(0.004)	66.85(0.15)	0.459(0.003)	4
	LSFS	83.84(0.16)	0.659(0.004)	69.26(0.11)	0.497(0.002)	82.68(0.16)	0.633(0.003)	69.28(0.22)	0.497(0.004)	4.15
	MCFS	80(0.09)	0.586(0.002)	65.27(0.09)	0.451(0.002)	82.27(0.14)	0.624(0.003)	65.24(0.12)	0.449(0.002)	4.11
	DSFFC	**86.69**(0.07)	**0.719**(0.002)	**75.63**(0.12)	**0.585**(0.002)	**84.31**(0.11)	**0.668**(0.002)	**75.71**(0.15)	**0.586**(0.002)	**4.31**
Ionosphere	FSFS	91.77(0.49)	0.823(0.011)	73.73(0.61)	0.435(0.013)	75.41(0.64)	0.462(0.016)	85.93(1.36)	0.689(0.030)	**3.60**
	LSFS	91.37(0.43)	0.814(0.01)	76.84(0.71)	0.521(0.01)	**84.67**(0.6)	**0.669**(0.013)	88.83(1.18)	0.755(0.026)	2.82
	MCFS	**94.22**(0.7)	**0.874**(0.015)	87.89(0.73)	0.746(0.013)	82.11(0.6)	0.615(0.013)	90.46(0.91)	0.792(0.020)	3.3
	DSFFC	94.07(0.29)	0.873(0.006)	**89.06**(0.57)	**0.766**(0.011)	82.54(0.72)	0.627(0.017)	**90.85**(0.81)	**0.822**(0.015)	3.47
WDBC	FSFS	94.41(0.18)	0.88(0.004)	91.11(0.22)	0.809(0.005)	93.22(0.43)	0.854(0.01)	94.22(0.63)	0.876(0.014)	**2.56**
	LSFS	**96.87**(0.2)	**0.933**(0.004)	93.71(0.16)	0.866(0.003)	95.87(0.21)	0.912(0.004)	95.85(0.50)	0.911(0.011)	1.59
	MCFS	96.68(0.24)	0.929(0.005)	93.39(0.24)	0.859(0.005)	**96.22**(0.24)	**0.92**(0.005)	95.11(0.44)	0.895(0.009)	2.06
	DSFFC	96.82(0.15)	0.932(0.003)	**94.34**(0.16)	**0.879**(0.003)	95.73(0.17)	0.909(0.004)	**96.22**(0.31)	**0.919**(0.007)	2.54
Sonar	FSFS	80.24(1.35)	0.606(0.026)	70.82(2.41)	0.415(0.048)	68.51(1.62)	0374(0.036)	77.16(1.97)	0.541(0.039)	3.37
	LSFS	81.01(1.27)	0.620(0.026)	**71.88**(1.98)	**0.438**(0.039)	67.98(1.20)	0.360(0.027)	75.67(1.64)	0.511(0.033)	3.2
	MCFS	**82.45**(1.04)	**0.650**(0.021)	67.36(1.37)	0.379(0.026)	70.14(1.12)	0.408(0.024)	77.21(2.07)	0.543(0.042)	3.4
	DSFFC	82.21(1.38)	0.642(0.028)	69.42(0.94)	0.409(0.020)	**71.83**(1.09)	**0.440**(0.022)	**79.09**(1.94)	**0.580**(0.039)	**3.88**
SPECTF	FSFS	73.38(2.13)	0.493(0.039)	73.63(1.61)	0.480(0.032)	66(1.94)	0.424(0.037)	65.50(2.78)	0.312(0.055)	**3.69**
	LSFS	74(1.42)	0.513(0.030)	72.75(1.42)	0.474(0.029)	**69.63**(2.50)	**0.472**(0.060)	69(3.48)	0.381(0.069)	3.31
	MCFS	71.88(2.14)	0.479(0.039)	72.13(1.45)	0.468(0.028)	66.38(2.32)	0.383(0.047)	72.75(3.16)	0.458(0.066)	3.29
	DSFFC	**76.88**(1.79)	**0.540**(0.033)	**79.75**(1.84)	**0.600**(0.038)	68.13(1.59)	0.468(0.027)	**76.88**(1.79)	**0.540**(0.033)	3.67

the very worst results as compared to the others for this data. For the fourth data *Spambase*, the performance of the proposed algorithm is the best among the four candidates irrespective of the underlying used classifiers in terms of both accuracy and *MCC*. In fact, the gain in performance obtained by DSFFC is remarkable for this dataset. For the fifth data *Inonosphere*, each of LSFS and MCFS performs the best only for one classifier, i.e., KNN and SVM, respectively. On the other hand, the proposed DSFFC provides the best performances for the remaining two classifiers. For the sixth data *WDBC*, the proposed approach supersedes the others for two classifiers, namely, Naive Bayes and AdaBoost, whereas each of LSFS and MCFS provides the best value only for one classifier. For the seventh data *Sonar*, DSFFC beats the others for two classifiers, namely, KNN and AdaBoost whereas each of LSFS and MCFS offers one best result. For the last data *SPECTF*, the proposed approach outperforms the others for all the classifiers except KNN. In terms of the RE, FSFS and DSFFC emerge as the top performers. Overall, the proposed method appears to be the most effective among the four competitors, being the top performer in a majority of the cases, while never being the worst in any case.

Table 3 shows the number of times the best results are obtained by each of the above mentioned four algorithms. For *Colon*, it is observed that our method provides the best results six times, whereas MCFS provides the best results two times. For each of the second data *Multiple Features* as well as the third data *Isolet*, the proposed one provides the best results three times while MCFS performs better than all others in two cases. For the fourth data *Spambase*, the proposed method outperforms the others in all of the nine cases. For the fifth as well as sixth data, namely, *Ionosphere* and *WDBC*, the proposed one provides the best results four times individually whereas this value becomes two for the other two algorithms, namely, LSFS and MCFS. For the remaining two data, i.e., *Sonar* and *SPECTF*, the proposed approach provides the best results for five and six times, respectively. As an overall, the table reveals that the proposed one performs the best by giving the best results forty times among total sixty-four cases whereas MCFS ranks the second acquiring the best results only for twelve times. On the other hand, these values for FSFS and LSFS are only four and eight, respectively. This seems that our algorithm ranks the top position in terms of providing the number of best results in maximum times.

Table 3
Summary of number of times the best results are obtained by different unsupervised feature selection algorithms.

Dataset	FSFS	LSFS	MCFS	DSFFC
Colon	1	0	2	6
Multiple Features	0	0	2	3
Isolet	0	0	2	3
Spambase	0	0	0	9
Ionosphere	1	2	2	4
WDBC	1	2	2	4
Sonar	0	2	2	5
SPECTF	1	2	0	6
Overall	4	8	12	40

The comparative performances of the proposed approach against each of the three other existing algorithms have been

Table 4
Summary of comparative performances of different unsupervised feature selection algorithms. The entry in the row X under the column W–D–L (Y) means win–draw–loss of DSFFC compared to Y on the X dataset. The entry in the row X under the column SW–SL (Y) means significant win–significant loss of DSFFC compared to Y on the X dataset.

	W–D–L (FSFS)	SW–SL (FSFS)	W–D–L (LSFS)	SW–SL (LSFS)	W–D–L (MCFS)	SW–SL (MCFS)
Colon	8–0–1	2–0	9–0–0	8–0	7–0–2	4–0
Multiple Features	3–0–2	2–2	5–0–0	2–0	3–0–2	1–2
Isolet	5–0–0	4–0	5–0–0	4–0	3–0–2	2–2
Spambase	9–0–0	8–0	9–0–0	8–0	9–0–0	8–0
Ionosphere	8–0–1	8–0	7–0–2	6–2	7–0–2	2–0
WDBC	8–0–1	8–0	5–0–4	2–0	7–0–2	4–2
Sonar	7–0–2	2–0	7–0–2	4–1	7–0–2	2–0
SPECTF	8–0–1	4–0	7–0–2	4–0	9–0–0	3–0
Overall	56–0–8	38–2	54–0–10	38–3	52–0–12	26–6

summarized in Table 4. The table has mainly analyzed two criteria, namely, Win-Draw-Loss (W–D–L) and Significant Win-Significant Loss (SW–SL) in which the value of SW–SL is computed using one-way paired sample t-test. In the present analysis, the p-value = 0.01 is considered to be the threshold for showing the corresponding result to be significant. For the first dataset *Colon*, the W–D–L of the proposed algorithm over FSFS, LSFS and MCFS are 8–0–1, 9–0–0 and 7–0–2, respectively. Also, the values of SW–SL of DSFFC over the other three approaches are 2–0, 8–0 and 4–0, respectively. These results indicate that the proposed one attains very good results as compared to the other three. For the second dataset *Multiple Features*, the W–D–L of the proposed technique over the other three are 3–0–2, 5–0–0 and 3–0–2, respectively and accordingly the values corresponding to SW–SL are 2–2, 2–0 and 1–2, respectively. These results indicate that DSFFC performs better than LSFS and has almost similar performance against FSFS. This is the only dataset in which any other feature selection algorithm (MCFS in this case) wins significantly the most number of times than it loses significantly in comparison to DSFFC. For the third dataset *Isolet*, DSFFC achieves better performance as compared to FSFS as well as LSFS whereas it performs equally well with respect to MCFS. For the fourth data *Spambase*, the W–D–L and SW–SL of the proposed technique over each of the remaining three algorithms are 9–0–0 and 8–0, respectively. These results signify that DSFFC achieves outstanding performance for this data. For the fifth dataset *Ionosphere*, almost the same observation is found as compared to the remaining three methods. However, each of LSFS and MCFS provides two win values against the proposed approach in which the win of only LSFS is found to be significant. For the sixth data *WDBC*, the proposed DSFFC performs remarkably well as compared to FSFS just like the fourth and fifth datasets. Although the proposed technique beats LSFS in terms of the number of significant win, this is the only one case where LSFS performs almost equally well as compared to DSFFC in terms of the number of win. For the seventh as well as eight data, i.e., *Sonar* and *SPECTF*, we observe that the proposed algorithm performs very well as compared to each of the three competitors. These summary information once again establish the superiority of the proposed approach over the other existing unsupervised feature selection algorithms.

6. Conclusion

In this paper, a novel unsupervised feature selection algorithm has been developed by integrating the concept of densest subgraph finding with feature clustering. The proposed two-phase approach improves classifier performance by selecting an optimal feature subset that not only minimizes the mutual dependency among the chosen features but also maximizes the mutual dependency of the selected features against the non-selected features. In this work, a novel existing normalized mutual information is also utilized to compute the similarity between two features.

Acknowledgements

Tapas Bhadra gratefully acknowledges Department of Science and Technology, India for awarding him the INSPIRE Fellowship (via. office order No. DST/INSPIRE Fellowship/2011/208) to carry out his Ph.D. research work. Sanghamitra Bandyopadhyay gratefully acknowledges the financial support from the Swarnajayanti project Grant No. DST/SJF/ET-02/2006-07 of the Department of Science and Technology, Government of India.

References

Bache, K., Lichman, M., 2013. UCI machine learning repository. University of California, Irvine, School of Information and Computer Sciences. <http://archive.ics.uci.edu/ml>.
Bahmani, B., Kumar, R., Vassilvitskii, S., 2012. Densest subgraph in streaming and mapreduce. In: Proceedings of VLDB Endowment, vol. 5, pp. 454–465.
Battiti, R., 1994. Using mutual information for selecting features in supervised neural net learning. IEEE Trans. Neural Networks 5, 537–550.
Biesiada, J., Duch, W., 2008. Feature selection for high-dimensional data: a Pearson redundancy based filter. Adv. Soft Comput. 45, 242–249.
Cai, D., Zhang, C., He, X., 2010. Unsupervised feature selection for multi-cluster data. In: Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining. Washington, USA, pp. 333–342.
Chang, C.C., Lin, C.J., 2011. LIBSVM: a library for support vector machines. ACM Trans. Intell. Syst. Technol. (TIST) 2, 27-1–27-27.
Cover, T.M., Thomas, J.A., 2012. Elements of Information Theory. John Wiley & Sons, New York, USA.
Dash, M., Liu, H., 1997. Feature selection for classification. Intell. Data Anal. 1, 131–156.
Estévez, P.A., Tesmer, M., Perez, C.A., Zurada, J.M., 2009. Normalized mutual information feature selection. IEEE Trans. Neural Networks 20, 189–201.
Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I., 2009. The WEKA data mining software: an update. ACM SIGKDD Explor. Newslett. 11, 10–18.
He, X., Cai, D., Niyogi, P., 2005. Laplacian score for feature selection. In: Advances in Neural Information Processing Systems. MIT Press, pp. 507–514.
Hua, J., Tembe, W., Dougherty, E.R., 2008. Feature selection in the classification of high-dimension data. In: IEEE International Workshop on Genomic Signal Processing and Statistics, pp. 1–2.
Jin, X., Xu, A., Bie, R., Guo, P., 2006. Machine learning techniques and chi-square feature selection for cancer classification using SAGE gene expression profiles. Lecture Notes in Computer Science, vol. 3916, pp. 106–115.
Kohavi, R., John, G.H., 1997. Wrappers for feature subset selection. Art. Intell. 97, 273–324.
Kononenko, I., 1994. Estimating attributes: analysis and extensions of RELIEF. In: Machine Learning: ECML-94, pp. 171–182.
Kwak, N., Choi, C., 2002. Input feature selection by mutual information based on parzen window. IEEE Trans. Pattern Anal. Mach. Intell. 24, 1667–1671.
Liao, C., Li, S., Luo, Z., 2007. Gene selection using Wilcoxon rank sum test and support vector machine for cancer classification. Lecture Notes in Computer Science, vol. 4456, pp. 57–66.
Liu, S., Liu, H., Latecki, L.J., Yan, S., Xu, C., Lu, H., 2011. Size adaptive selection of most informative features. AAAI, 392–397.
Mandal, M., Mukhopadhyay, A., 2013. Unsupervised non-redundant feature selection: a graph-theoretic approach. In: Proceedings of the International Conference on Frontiers of Intelligent Computing: Theory and Applications (FICTA), pp. 373–380.
Mitra, P., Murthy, C.A., Pal, S.K., 2002. Unsupervised feature selection using feature similarity. IEEE Trans. Pattern Anal. Mach. Intell. 24, 301–312.
Peng, H., Long, F., Ding, C., 2005. Feature selection based on mutual information: criteria of max-dependency, max-relevance and min-redundancy. IEEE Trans. Pattern Anal. Mach. Intell. 27, 1226–1238.

Strehl, A., Ghosh, J., 2002. Cluster ensembles – a knowledge reuse framework for combining multiple partitions. J. Mach. Learn. Res. 3, 583–617.

Vinh, L.T., Thang, N.D., Lee, Y.K., 2010. An improved maximum relevance and minimum redundancy feature selection algorithm based on normalized mutual information. In: 10th Annual International Symposium on Applications and the Internet, pp. 395–398.

Witten, I.H., Frank, E., 2005. Data Mining: Practical Machine Learning Tools and Techniques. Morgan Kaufmann, San Francisco, USA.

Yang, Y., Shen, H., Ma, Z., Huang, Z., Zhou, X., 2011. $L_{2,1}$-norm regularized discriminative feature selection for unsupervised learning. In: Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence, Barcelona, Spain, pp. 1589–1594.

Zhao, Z., Liu, H., 2007. Spectral feature selection for supervised and unsupervised learning. In: Proceedings of the 24th International Conference on Machine Learning, Corvallis, OR, pp. 1151–1157.