Bridging Feature Selection and Extraction: Compound Feature Generation

Sreevani, *Student Member, IEEE* and C. A. Murthy

Abstract—Dimensionality reduction is an essential pre-processing technique in many of the data analysis tasks. Popular approaches for dimensionality reduction are Feature Selection (FS) and Feature Extraction (FE). Till now, these approaches are often studied separately or independently so that the final result contains either original or transformed features. In our work, we propose to bridge these two approaches with the aim of finding reduced feature set to contain both kinds (original as well as transformed) of features. A new framework, called Minimum Projection error Minimum Redundancy (MPeMR), is introduced to obtain this result while maintaining orthogonality property among selected original and linear combinations of features. A unified iterative algorithm, for both supervised and unsupervised cases, is also developed under this framework. For each case, the performance of the proposed algorithm is successfully compared with the state-of-the-art methods on real-life data sets.

Index Terms—Dimesionality reduction, feature selection, feature extraction, classification, clustering, compound features

1 Introduction

In recent years, high dimensional data sets have become very common in machine learning and data mining applications. Processing of such data sets requires huge computational time and resources [1]. Moreover, with the presence of irrelevant, redundant and noisy features, the performance of the learning algorithm degrades [2]. Hence, it is crucial to reduce the dimensionality of the data to improve both the efficiency and effectiveness of most of the data mining algorithms. Also it is important for better visualization, data compression, noise removal, improved understanding ability, and generalization of the learning algorithms [3]. Traditional and state-of-the-art dimensionality reduction methods fall into two categories: feature selection and feature extraction [2]. These approaches have been successfully applied in many real applications, such as Image processing, text categorization, bioinformatics [4], [5], [6], etc. Feature selection aims at finding a subset of most useful features from the original set of features, whereas feature extraction methods provide combinations (linear or nonlinear) of the original features.

In the past few decades, these two approaches have been studied extensively [7], [8]. However, all the studies have been done separately or independently. Although the ultimate aim of both the approaches is to improve the efficiency of a learning algorithm, none of the feature selection methods provide even a single combination of features, which may be more informative than the original features. On the other hand, a feature extraction approach provides transformed features, where each transformed feature is a combination of all original features, and no original feature appears among the transformed features. If these two approaches can be integrated in a systematic way, to provide reduced set with both types of features, they could complement each other. In the next section, two synthetic data sets in $3D$ are provided to show the effectiveness of having both types of features. For these data, when the dimension is reduced from three to two, one original and one combination of features (here the combination does not involve all of the original features) produce better representation, than having either two original features or two transformed features. So, there must exist methods where the final result will be a few original features and a few linear combinations.

Many algorithms for feature selection/extraction have been suggested in the literature. The main idea of feature selection is to choose a subset of original features by eliminating features with little or no predictive information. With respect to whether the label information is available, different methods for feature selection can be divided into supervised, unsupervised, or semi-supervised algorithms. In supervised feature selection algorithms [9], [10], [11], [12], [13], important features are determined by estimating their correlation with the class labels or their performance in prediction. Unsupervised feature selection algorithms select features by exploiting data variance or distribution [14], [15], [16], [17]. In a semi-supervised feature selection algorithm, small amount of labeled data is used as additional information to improve the performance of the unsupervised feature selection algorithm [18], [19]. Based on different selection strategies used, methods for feature selection can be categorized into three groups, filter, wrapper and embedded methods. Filter algorithms evaluate features using certain statistical criteria and independent of any classifier [20], [21], [22]. On the contrary, wrapper methods [23], [24] select a set of features based on a selection criteria with respect to a given classifier, such as: Bayes, Knn, SVM. Wrapper methods in general are more computationally expensive and hence, for real-life applications with large data sets, the filter model is more popular. However, the wrapper model has been

• *The authors are with the Machine Intelligence Unit, Indian Statistical Institute, Kolkata, West Bengal 700108, India. E-mail: {sreevani_r, murthy}@isical.ac.in.*

Manuscript received 10 Nov. 2015; revised 23 July 2016; accepted 7 Oct. 2016. Date of publication 20 Oct. 2016; date of current version 3 Mar. 2017. Recommended for acceptance by J. Ye.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TKDE.2016.2619712

empirically proven to be superior, in terms of classification accuracy, to a filter model. Finally, the embedded method achieves model fitting and feature selection simultaneously [25], [26]. In addition, feature selection algorithms can also be categorized as subset selection algorithms [22], [27], which returns a subset of selected features or feature weighting algorithms [9], which returns weight corresponding to each feature.

Feature extraction, linearly or non-linearly, transforms the original high dimensional data to a low dimensional data. The objective of feature extraction is to find an appropriate transformation that maps the original D-dimensional space to a new d-dimensional feature space, where $d \ll D$. According to the availability of the class label information, feature extraction methods are categorized into supervised or unsupervised methods. They are also broadly divided into linear and non-linear methods. Linear feature extraction seeks a meaningful low dimensional subspace in a high dimensional input space by linear transformation. Among all the linear feature extraction methods, the most well known are Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA) [28]. PCA seeks a transformation to produce uncorrelated and orthogonal principal components and LDA produces a transform while preserving as much class discriminatory information as possible [28], [29]. Other unsupervised feature extraction methods are: Factor Analysis (FA) [30], projection pursuit [31], Independent Component Analysis (ICA) [32], etc. Some of the well known supervised feature extraction methods are: Maximum Margin Criterion (MMC) [33], Angular Liner Discriminant Embedding (ALDE) [34], etc. Transformed features of these methods usually contain all the original variables in their linear combinations which may be difficult to interpret. To overcome this drawback, sparse principal component analysis (SPCA) [35], sparse linear discriminant analysis (SLDA) [36] are introduced to produce modified principal components which just contain a few original variables. However, unlike PCA, sparse PCA cannot guarantee that different principal components are uncorrelated [35].

Some methods which study feature selection and extraction together exists in the literature. In [37], a general transformation-based dimensionality reduction algorithm has been converted in to a feature selection formulation. In [38], a joint framework to do feature selection and subspace learning simultaneously based on using $L_{2,1}$- norm on the projection matrix, which leads to selecting relevant features and learning transformation simultaneously.

All the methods listed above provide reduced set either with original or transformed features, but not both of them. In this paper, we propose to bridge the gap between feature selection and extraction approaches, which exists as one provides original and the other provides transformed features. We study these two methods together with the aim of obtaining a reduced feature set to contain both kinds (original and combinations) of features. An approach for dimensionality reduction where linear combinations of features are considered, and orthogonality is maintained on selected linear combinations of features and original features is suggested. We also present an approximation algorithm under this framework.

The organization of the article is as follows: In the next section, new methodology for dimensionality reduction to provide reduced set with both types of features is proposed. Section 3 reports experimental results to support the proposed method. Finally, section 4 contains discussion and conclusions.

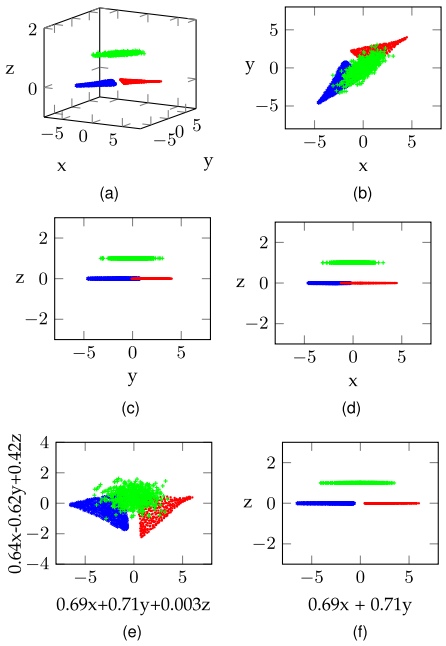

Fig. 1. (a) Example data. (b)-(d) Joint projections (xy, yz, xz) of the data. (e) Projection according to top two eigen vectors of PCA. (f) Projection according to $\{0.69x + 0.71y,\ z\}$.

2 Proposed Methodology

In this section, initially, motivation for the stated problem is explained by using two examples. Later, a framework for generation of compound features (defined later in this section) is suggested. Then an efficient algorithm for implementation of the framework is proposed. The preliminary part of the proposed work was carried out as a dissertation in Masters of one of the authors [39].

2.1 Motivation (Toy Examples)

Here, two synthetic data sets[1] are used to show the effectiveness of having both types of features in the reduced set. These two data sets are shown in Figs. 1a and 2a. In both figures, corresponding (b)-(d) show the projections of the data on the plane of two joint features i.e., on xy, yz, xz respectively.

The first data set has three clusters and three features, and the number of features is to be reduced to 2. Popular feature selection methods (e.g., Laplacian Score and MCFS) select features y and z. The projection of this data on to yz-plane is shown in Fig. 1c. It is clear that the clusters in the projected data are overlapping. Even with the other joint projections (xy and xz), the same phenomenon is observed. Fig. 1e shows the two dimensional transformed data using PCA with the extracted features $\{0.69x + 0.71y + 0.003z,\ 0.64x - 0.62y + 0.42z\}$. Even with the transformed features, the clusters are still overlapping. Now, we consider one combined feature, where the combination does not involve

1. The actual data sets can be found at http://www.isical.ac.in/~sreevani_r/CFG.html

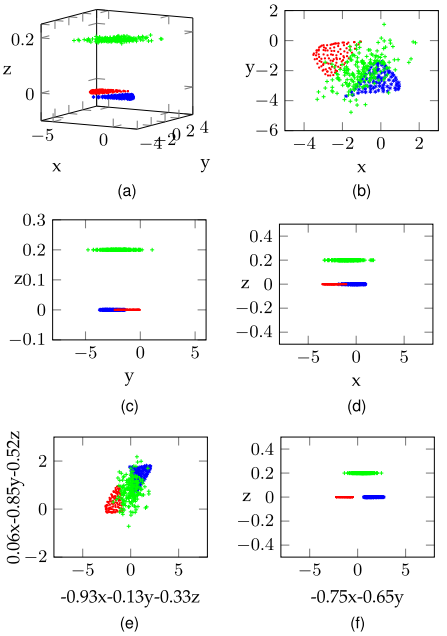

Fig. 2. (a) Example data. (b)-(d) Joint projections (xy, yz, xz) of the data. (e) Projection according to top two eigen vectors of LDA. (f) Projection according to $\{-0.75x - 0.65y,\ z\}$.

all of the original features, and one original feature, which are $0.69x + 0.71y$ (first principal component of x and y) and original z variable. Projection on to these features is provided in Fig. 1f. It may be observed that the points in the three clusters are separated properly by these two features. Another example with similar conclusion for supervised case is stated below.

The second data set also has three classes and three features and is shown in Fig. 2a. Here also the dimensionality to be reduced is 2. Projection spaces for all joint features (xy, yz, xz) are shown in Figs. 2b, 2c, and 2d. Popular feature extraction method LDA's Projection space with first two eigen vectors $\{-0.93x - 0.13y - 0.33z, 0.06x - 0.85y - 0.52z\}$ is shown in Fig. 2e. Clearly, the projected classes are overlapped. Now we consider first eigen vector of x and y variables using LDA, i.e., $-0.75x - 0.65y$ along with original z variable and the resulting projection space is provided in Fig. 2f. These two features, one is a linear combination of two features and the second is one of the original features, provide better separation among the classes.

From the above two examples, one can observe that reduced set containing both types of features sometimes might provide better results than having only a subset of originals or a subset of transformed features. The existing methods do not provide us an opportunity to look at the original features and linear combinations of features simultaneously to evaluate their effectiveness for dimensionality reduction. This is due to the fact that the mathematical formulation of such a problem and the solution are not easy to obtain. These are explained below.

Let $O = \{f_1, f_2, \ldots, f_D\}$ be the given set of features. And suppose that one would like to look at the linear combinations of f_1 and f_2. The number of linear combinations i.e., the cardinality of the set $\{a_1 f_1 + a_2 f_2 : a_1, a_2 \in \mathbb{R}\}$ is uncountable. Even if a restriction is put on the coefficients, such as, the sum of squares of the coefficients is one i.e., $\{a_1 f_1 + a_2 f_2 : a_1, a_2 \in \mathbb{R}, a_1^2 + a_2^2 = 1\}$, it would still be uncountable. This is the main difficulty in the formulation of the problem. A way of formulation of the optimization problem using linear combinations, as well as the original features is stated here.

We know that, PCA or LDA follow the principle of variance/discrimination maximization to obtain linear combinations of all original features. Let $Y = \{w_i \mid w_i = a_{i1} f_1 + a_{i2} f_2 + \cdots + a_{iD} f_D, i = 1, \ldots, D\}$ be the set of D principal components of O and also let $A = Y \cup O$. One can optimize over A to obtain a subset of features and it is given by,

$$A_0 = \max_{A' \subset A} J(A'),$$

where J is the criterion function. The difficulty in this setup is that we may ultimately choose a feature, say f_1, and a linear combination which has a non-zero coefficient of f_1 i.e., A_0 might contain f_1 and $w = a_1 f_1 + a_2 f_2 + \cdots + a_D f_D$ with $a_1 \neq 0$. And as the inner product between the coefficient vector $(1, 0, \ldots, 0)$ of f_1 and the coefficient vector $(a_1, a_2, \ldots, a_D)$ of w is non-zero, the features f_1 and w are not orthogonal. Thus the above formulation leads us to produce non-orthogonal features, which is not desirable. One may still write an algorithm with constraints under this setup, which we are not attempting it here. The suggested procedure next will have the constraint that if a feature from the original set of features is selected, then that feature will not appear in any linear combination.

We know that feature extraction is done on the whole set of original features, and the transformed features are the combinations of all of the original features. In our work, as we want to have the final reduced set to contain some original features and some combinations without losing orthogonality among them, we would like to define the following.

Definition 1 (Semi-feature). *A feature s is called a* semi-feature *if it is a combination of only a proper subset of original features i.e., $s = a_{i1} f_{i1} + a_{i2} f_{i2} + \cdots + a_{ik} f_{ik}$, $f_{i1}, \ldots, f_{ik} \in O$, $a_{ij} \in \mathbb{R}$ and $k < D$.*

Definition 2 (Compound Feature Set (CFS)). *A set C which contains both original (f) and semi-features (s) while maintaining orthogonality among them i.e.,*

$$C = \{u \mid u \in \{f_{i1}, \ldots, f_{ip}, s_{j1}, \ldots, s_{jq}\}, \\ \text{any two } u\text{'s are orthogonal } \& \ p + q \leq D\}.$$

Definition 3 (Compound Feature Generation (CFG)). *Process of generating compound features without losing orthogonality among the selected original and combinations of features.*

When a semi-feature $s = a_{i1} f_{i1} + a_{i2} f_{i2} + \cdots + a_{ik} f_{ik}$ is present in the reduced feature set, the original features $f_{i1}, \ldots, f_{ik}$ should not be present in the final set of features so as to maintain orthogonality property. Throughout this paper, if a new semi-feature is considered, it also means that the underlying original features are discarded. In rest of the manuscript, a 'feature' is denoted by 'u' to indicate that it can be either original or semi-feature.

General procedure in any spectral feature extraction method involves formulating a criterion function and finding corresponding eigen values and eigen vectors. Top eigen vectors (with larger eigen values) contain most of the information. For example, in case of PCA, the top eigen vectors contain maximum variance and in case of LDA, the top eigen vectors contain maximum discrimination information. Smallest eigen value means the little information along its corresponding eigen vector. This can be considered as the amount of error introduced while projecting the data to a reduced dimension in a best possible way. If the error introduced is not significant, one can discard the eigen vector with the smallest eigen value and retain the top eigen vectors. This error, we call it as 'Projection error' as it is introduced while projecting the original data in a best possible way.

Definition 4 (Normalized Projection Error (NPe)). *For a given subset of features $\{u_1, u_2, \ldots, u_k\}$ and a spectral feature extraction method M, let $\lambda_1, \ldots, \lambda_k$ be the resulting eigen values with $\lambda_1 > \cdots > \lambda_k$. Since, the smallest eigen value λ_k provides the information along the kth component, it can be considered as the amount of error introduced while projecting the data to reduced dimension, from k to $k-1$. Since λ_k is arbitrary, it can be normalized by sum of all eigenvalues $\lambda_1 + \cdots + \lambda_k$. This ratio is called 'Normalized Projection error' of the feature set $u_1, \ldots, u_k$ introduced when dimension is reduced from k to $k-1$ i.e.,*

$$NPe^{k,k-1}_{1,\ldots,k} = \frac{\lambda_k}{\sum_{i=1}^{k} \lambda_i}.$$

1. *If the error $NPe^{k,k-1}_{1,\ldots,k}$ of the feature set $u_1, \ldots, u_k$ is less than a predefined threshold, then one can reduce the dimension from k to $k-1$ and consider the top $k-1$ extracted features while discarding original k features.*
2. *The above definition can be extended based on the dimension to be reduced i.e., if one wishes to reduce the dimension from k to m, where $m < k$, then one can consider $NPe^{k,m}_{1,\ldots,k}$ and it is given by,*

$$NPe^{k,m}_{1,\ldots,k} = \frac{\sum_{i=m+1}^{k} \lambda_i}{\sum_{i=1}^{k} \lambda_i}.$$

It is widely accepted that the performance of the learning algorithm could be degraded when the considered features include redundant information and we say that two features are redundant to each other if their values are completely correlated [40]. Some measures, correlation coefficient (ρ), Symmetrical Uncertainty (SU), Decision Independent Correlation (DIC), Decision Dependent Correlation (DDC), for calculating redundancy ($Red(u_i, u_j)$) between any two features u_i, u_j, are provided in the Appendix, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TKDE.2016.2619712. The first three measures ρ, SU, DIC can also be used to find the correlation between a feature (u) and the class variable (C). Given these measures, *Ranking Criterion* of a feature and the representative feature (*R-feature*) of a feature subset can be defined as follows.

Definition 5 (Ranking Criterion (RC)). *For a feature u, in case of unsupervised learning, Ranking Criterion ($RC(u)$) is defined in terms of data variance or data distribution. And, in supervised case, its correlation with the class or their efficacy in prediction, for example $\rho(u, C)$ or $SU(u, C)$, can be considered as RC and it is denoted by $RC^{(c)}(u)$.*

Definition 6 (Ranking-coefficient). *Based on unsupervised or supervised learning, Ranking-coefficient for a feature u_i in the feature set $\{u_1, u_2, \ldots, u_k\}$ is denoted by Ranking-coefficient(u_i) or Ranking-coefficient$^{(c)}(u_i)$ respectively. If $\{u'_1, u'_2, \ldots, u'_k\}$ is a permutation of $\{u_1, u_2, \ldots, u_k\}$ s.t. $RC(u'_1) \geq RC(u'_2) \geq \cdots \geq RC(u'_k)$ or $RC^{(c)}(u'_1) \geq RC^{(c)}(u'_2) \geq \cdots \geq RC^{(c)}(u'_k)$, then Ranking-coefficient(u'_k)(or Ranking-coefficient$^{(c)}(u'_k)$) $= k$.*

Definition 7 (R-feature). *A feature u_r in the feature subset $A = \{u_1, u_2, \ldots, u_k\}$ is a Representative feature (R-feature) of A if Ranking-coefficient of u_r is 1. And the remaining features $\{u_j \mid j \neq r, j = 1, 2, \ldots, k\}$ are called 'weak' features.*

We propose a new framework to obtain a set of reduced features by (i) extracting semi-features from those features for which projection errors are minimum, and (ii) redundancy is minimized among selected original and combinations of features. We call this framework as *M*inimum *P*rojection *e*rror and *M*inimum *R*edundancy (MPeMR) approach.

2.2 A New Framework for Generation of Compound Features

Our goal is to select an effective and more representative compound feature set. We can achieve this goal through a new framework, named as minimum projection error and minimum redundancy framework. MPeMR tries to provide informative compound features with minimum projection error and redundancy while maintaining orthogonality among them. This framework consists of two stages: first stage finds compound features with the minimum projection-error i.e., this stage involves first finding feature subsets based on NPe and then semi-feature extraction is performed on the feature subsets for which NPe is minimum. Finally, this stage produces orthogonal compound features with minimum projection error. In the second stage, redundant features are grouped from the above learned compound features, and then $R\text{-}features$ from these groups are considered while removing the weak features. So, Final reduced set consists of orthogonal compound features with both minimum projection error and minimum redundancy. Flow diagram of this framework is given in Fig. 3.

The benefits of this approach can be realized in two ways. (i) NPe, redundancy and $R\text{-}feature$ of a set of features are not restricted to any specific measurement in MPeMR. Therefore, many existing feature extraction, redundancy and $R\text{-}feature$ selection methods can be incorporated into the proposed framework. (ii) With the same number of features, we expect that the MPeMR feature set to be more informative as it contains both types of features in the reduced and therefore leading to better generalization capability.

As searching for exhaustive set of features is difficult and as explained before the linear combinations are

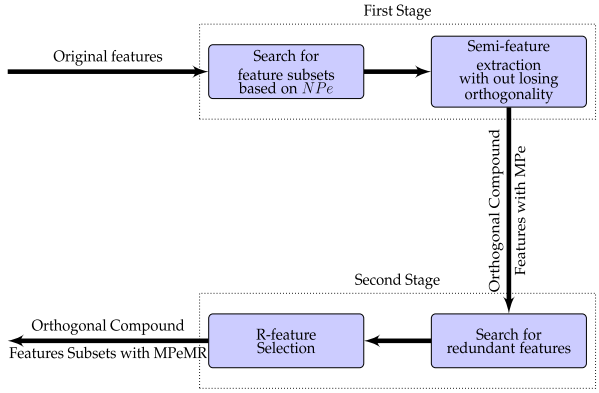

Fig. 3. A new framework for compound feature generation.

computationally intractable, we next present an approximation method under this framework.

2.3 Efficient CFG Method Based on the Proposed Framework

Proposed framework mainly depends on NPe, redundancy and $R\text{-}feature$ selection measures to generate compound features, and these measures can be used in both supervised and unsupervised learning. So, we provide a unified approximation method for CFG for both supervised and unsupervised cases under the proposed framework. This is an iterative method, uses backward elimination search strategy and based on two features at a time.

Proposed method initially considers all original features $f_1, f_2, \ldots, f_D$ as candidate features, and it has two stages in every iteration. First stage involves extraction of semi-features. Second stage involves redundancy removal. In the first stage, for every distinct pair of available features, the measure NPe is calculated. If the error $NPe^{2,1}_{i,j}$ for a pair (u_i, u_j) is less than a predefined threshold T_1, then semi-feature extraction is performed on $\{u_i, u_j\}$. First eigen component (s_k) of $\{u_i, u_j\}$ is added and both u_i, u_j are removed so as to maintain orthogonality among the features in the reduced set. In the second stage, from each distinct pair of available features (u_i, u_j), if the value of the redundancy measure $Red(u_i, u_j)$ between the pair is greater than a predefined threshold T_2, then $R\text{-}feature$ is learned by calculating Ranking-Criterions of both the features. This $R\text{-}feature$, say u_i is retained and the weak feature, say u_j is discarded to constitute the reduced feature set. After the first iteration, reduced set contains some originals and some combinations, and each combination involves two features. In general, after the kth iteration, reduced set contains some originals (f) and some combinations (s) and each combination involves at most $k+1$ features. The above two stage process is followed in every iteration and repeated till no more changes occur in the number of features. The algorithmic details of this method are given in Algorithm 1.

The proposed method can be illustrated with the following example data. Suppose that the data has eight features $f_1, f_2, \ldots, f_8$. At first stage, suppose for feature pairs (f_1, f_3) and (f_2, f_6), projection errors (NPe) (PCA can be used in unsupervised case) are less than T_1, so the first principal components $s_1 = a_1 f_1 + a_3 f_3$, $s_2 = a_2 f_2 + a_6 f_6$ of $\{f_1, f_3\}$, $\{f_2, f_6\}$ respectively have been

Algorithm 1. Compound Feature Set Generation (Minimum Projection Error - Minimum Redundancy)

Input: D - number of original features, $O = \{f_1, f_2, \ldots, f_D\}$ - original feature set,
T_1, T_2 - predefined threshold values.
Output: C - compound feature set

$S \leftarrow \Phi$ (set of semi-features, initially empty), $flag \leftarrow TRUE$, $k \leftarrow 1$;
while($flag == TRUE$) **do**
  $Dim \leftarrow |O \cup S|$;
  for ***each*** $u_i \in O \cup S$ ($u_i = f_j$ or s_k for some j, k) **do**
    for ***each*** $u_j (\neq u_i) \in O \cup S$ **do**
      if $Npe^{2,1}_{i,j} < T_1$ **then**
        $s_k \leftarrow$ first eigen component of u_i & u_j;
        $S \leftarrow S \cup \{s_k\}$;
        $k \leftarrow k + 1$;
        if $u_i, u_j \in O$ **then**
          $O \leftarrow O - \{u_i, u_j\}$;
        else if $u_i \in O,\ u_j \in S$ **then**
          $O \leftarrow O - \{u_i\}$;
          $S \leftarrow S - \{u_j\}$;
        else if $u_i \in S,\ u_j \in O$ **then**
          $O \leftarrow O - \{u_j\}$;
          $S \leftarrow S - \{u_i\}$;
        else
          $S \leftarrow S - \{u_i, u_j\}$;
  for ***each*** $u_i \in O \cup S$ **do**
    for ***each*** $u_j (\neq u_i) \in O \cup S$ **do**
      if $\text{Red}(u_i, u_j) > T_2$ **then**
        $u' \leftarrow$ R-feature of $\{u_i, u_j\}$;
        $u'' \leftarrow$ weak feature of $\{u_i, u_j\}$;
        if $u'' \in O$ **then**
          $O \leftarrow O - \{u''\}$;
        else
          $S \leftarrow S - \{u''\}$;
  if ($Dim == |O \cup S|$) **then**
    $flag \leftarrow FALSE$;
  else
    $flag \leftarrow TRUE$;
Return $C \leftarrow O \cup S$;

added. And f_1, f_3, f_2, f_6 are removed to maintain orthogonality. In the second stage, if 'correlation coefficient' is used as the redundancy measure, and suppose that s_1 and f_4 are redundant. And using 'variance' as the Ranking-Criterion, suppose f_4 is the 'weak' feature and so it should be removed from the list. So, after 1^{st} iteration, reduced set contains s_1, s_2, f_5, f_7, f_8. In the second iteration, suppose NPe between s_1 and f_5 is minimum, so first principal component s_3 of them should be added while removing both of them. And if s_2, f_7 are redundant with s_2 being the $R\text{-}feature$, then final reduced set contains three features, among them two are semi-features (s_3, s_2) and one is a original feature (f_8).

Initial Set of features		$\{f_1, f_2, f_3, f_4, f_5, f_6, f_7, f_8\}$
Iteration 1	1^{st} stage	$\{s_1, s_2, f_4, f_5, f_7, f_8\}$
	2^{nd} stage	$\{s_1, s_2, f_5, f_7, f_8\}$
Iteration 2	1^{st} stage	$\{s_3, s_2, f_7, f_8\}$
	2^{nd} stage	$\{s_3, s_2, f_8\}$

2.4 Remarks

- *Computational Complexity*: For every iteration, since in each stage every pair of features is considered, with respect to dimension(D) where D is the number of original features, the proposed method has complexity $\mathcal{O}(D^2)$. If the data set contains N number of samples, evaluation of NPe (PCA, LDA), redundancy measure (ρ, SU, DIC, DDC) and $R\text{-}feature$ selection measure, for a feature pair is of complexity $\mathcal{O}(N)$. Thus, every iteration of the proposed method has overall complexity $\mathcal{O}(D^2N)$.
- Note that, in each stage, pairs of features are considered i.e., NPe is calculated based on two features and removal of redundancy is also based on two features. Instead of the combination (2,2), one could have considered (2,3), or (3,2) or any other such combination of features. The algorithm needs to be changed accordingly then.
- Also in the first stage, one feature is extracted from each pair of features based on the value of $Npe^{2,1}_{i,j}$. Instead of one out of two, one could extract m semi-features from each group of k $(k > 2)$ features depending on the value $Npe^{k,m}_{i1,\ldots,i_k}$, where $m < k$.

3 Experimental Study

An approximation method for generating compound features under the MPeMR framework has been provided. In this section, we evaluate the performance of this CFG method in both supervised and unsupervised cases which are named as SMPeMR (**S**upervised **MPeMR**) and UnMPeMR (**Un**supervised **MPeMR**), respectively. For this purpose, 18 publicly available data sets[2,3,4] have been used. The number of features for these data sets vary from 7 to 20,790 and the number of instances ranges from 100 to 13,910. Descriptions of these data sets are given in Table 1. The three-class version of the abalone data set is considered here. All the images in Extended Yale data set were cropped to 154×135 pixels to show the efficiency of the proposed method. The position of the faces in the images were normalized in a way that the eyes, and the lips have almost the same position in all the images. All the experiments are performed on windows 8, with 3.4 GHz processor and 32 GB RAM machine.

As stated before, one can derive new CFG methods by using different measures for NPe, redundancy and $R\text{-}feature$ selection of features. In our experiments, for unsupervised case, PCA has been considered as the feature extraction method and used to calculate the measure NPe. Anyone of the redundancy measures ρ, SU, DIC can be used to find redundant features. Variance (Var) has been used to find representative feature set. In case of supervised learning, LDA has been considered as the feature extraction method and used to calculate the measure NPe. Anyone of the redundancy measures ρ, SU, DDC can be used to find redundant features. ρ and SU have been used to find representative feature set. For each learning, three different methods (by varying the measures for Npe, redundancy and $R\text{-}feature$) have been considered. We denote these methods as UnMPeMR$_1$, UnMPeMR$_2$, UnMPeMR$_3$ for the combinations (PCA, ρ, Var), (PCA, SU, Var), (PCA, DIC, Var) to be incorporated in to the proposed iterative method, respectively. And SMPeMR$_1$, SMPeMR$_2$, SMPeMR$_3$ denote for the combinations (LDA, ρ, ρ), (LDA, SU, SU), (LDA, DDC, SU) to be applied in to the proposed method, respectively. As these methods produce both original and combinations of features in the reduced set, we compare the results with both state-of-the-art feature selection and extraction methods. For unsupervised case, four feature selection and four feature extraction methods have been considered for comparison purpose. They are Maximum Variance (MaxVar, which selects the features corresponding to the maximum variances), Laplacian Score (LS) [15], SPEC [41], Mutli-Cluster Feature Selection (MCFS) [16], and PCA [28], FA [30], ICA [32], Sparse PCA (SPCA) [35]. The proposed method for supervised learning, is compared with six different types of feature selection methods. They are 1) Fisher Score [2], 2) ReliefF [9], 3) Double Input Symmetrical Relevance (DISR) [42], 4) SPEC [41], 5) Trace Ratio (TR) [10], and 6) Minimum Redundancy Spectral Feature selection (MRSF) [11]. And, six feature extraction methods: 1) LDA [28], 2) MMC [33], 3) ALDE [34], 4) Marginal Fisher Analysis (MFA) [43], 5) Quadratic Mutual Information (QMI) [44] and 6) Sparse LDA (SLDA) [36] have been used for comparison. For both LS and MCFS, the parameter k (nearest neighbors) is set to 5 and m (instances) is set to 30 throughout the experiments.

The proposed methods SMPeMR and UnMPeMR were compared based on their performance in classification and clustering tasks respectively. Two types of classification algorithms, namely 1) Knn (K = 1, 3, 5), and 2) Linear SVM with the 'c' parameter set to 1 in all the experiments, are used to classify data sets after dimensionality reduction. Classification accuracy rate has been considered as the evaluation measure. In case of clustering, two algorithms were used to compare different methods: one is the well-known K-means algorithm, and other one is the state-of-the-art Affinity Propagation (AP) algorithm [45]. For K-means, the algorithm is restarted 25 times using different random initializations and the average result is considered. As AP is

TABLE 1
Summary of the 18 Benchmark Data Sets

Data Set	Instances	Attributes	Classes
Abalone	4,177	8	3
COIL20	1,440	1,024	20
Ecoli	332	7	6
Extended Yale	2,414	20,790	38
Gas Sensor	13,910	128	6
Isolet	7,797	618	26
Libras Movement	360	90	15
Multiple	2,000	649	10
ORL	400	1,024	40
orlraws10P	100	10,304	10
PageBlocks	5,473	10	5
pixraw10P	100	10,000	10
Seeds	210	7	3
Segmentation	2,310	16	7
USPS	9,298	256	10
warpPIE10P	210	2,420	10
Waveform	5,000	21	3
Yeast	1,484	8	10

2. http://archive.ics.uci.edu/ml/index.html
3. http://featureselection.asu.edu/datasets.php
4. http://vision.ucsd.edu/~iskwak/ExtYaleDatabase/ExtYaleB.html

deterministic there is no need for random restarts. The number of clusters to be formed by these clustering methods is set to the number of classes in the data set for all experiments. After the clustering is performed using the subset of selected features, the quality of the clustering solutions is measured using the following indices.

As the true labels are known, the quality of the clustering solutions can be measured using external criteria which measure the discrepancy between the structure defined by a clustering and what is defined by the true labels. Many criteria have been proposed in the literature to compare two partitions of a data set and we choose four measures, two from counting pairs measures: Jaccard coefficient (Jacc), Fowlkes-Mallows index (FM) and two from information-theoretic measures: Normalized Mutual Information (NMI), Normalized Variation of Information (NVI) [46], [47]. Consider $K = \{K_1, \ldots, K_m\}$ is a clustering structure of the given data set and $P = \{P_1, \ldots, P_s\}$ is defined partition of the data. Suppose 'a' denotes the number of point pairs belonging to same partition in K as well as in P, 'b' the number of point pairs belonging to the same cluster in K but to different in P, 'c' the number of point pairs belonging to different clusters in K but to same clusters in P and 'd' the number of point pairs belonging to different clusters in K and different clusters in P. The Jaccard and FM between partitions K and P is defined as

$$\text{Jacc} = \frac{a}{a+b+c}, \ \text{FM} = \frac{a}{\sqrt{(a+b)(a+c)}}.$$

The values of the above two indices lie in the interval [0,1], and higher values of the indices indicate greater similarity between K and P. The two entropy based measures are computed as follows:

$$\text{NMI} = \frac{H(P) + H(K) - H(P,K)}{\sqrt{H(P)H(K)}},$$
$$\text{NVI} = \frac{H(P|K) + H(K|P)}{H(P)},$$

where $H(.)$, $H(.|.)$, $H(.,.)$ denote marginal, conditional and joint entropies respectively. The NMI value lies in the range $[0, 1]$ and values close to 1 indicate a good clustering. Values of NVI decrease as P and K become more similar and NVI would be 0 when they are identical.

For each data set, except for face image data sets (COIL20, ORL, orlraws10P,pixraw10P, warpPIE10P, ExtendedYale), each dimensionality reduction algorithm and each classification method, the ten-fold cross validation is conducted. For face image data sets, holdout cross validation, with 30 percent samples used for training and 70 percent samples used for testing, is conducted. The mean values over 25 such independent runs are reported. For each evaluation measure, "↑" indicates "the larger the better" and "↓" indicates "the smaller the better". Furthermore, the best performance among the comparing algorithms is shown in bold face.

3.1 Results and Analysis

In this section, we present and compare the experimental results of MPeMR in both supervised and unsupervised case. In case of unsupervised learning, we compare the quality of clustering solutions using Kmeans and AP after dimensionlity reduction. And for supervised case, we present the experimental results in terms of the classification accuracy.

For the purpose of exploring the statistical significance of the results, we performed a non-parametric Friedman test [48] followed by Nemenyi test [49] to statistically compare algorithms on multiple data sets. Thus, the Friedman and the Nemenyi test results are reported as well.

Note that, In each learning (supervised or unsupervised), there are three methods and each method requires two thresholds. So, in the corresponding clustering/classification performance tables, these values are given as $\text{T} = \{T_{11}, T_{12}; T_{21}, T_{22}; T_{31}, T_{32}\}$, where T_{i1}, T_{i2} are the threshold values for the ith method. For each data set, the reduced dimensionality (d) is also provided. As the proposed method provides compound features in the reduced set, the number of original and the number of combinations are also provided. These are denoted by $\text{d}_0 = \{d_{11} + d_{12}; d_{21} + d_{22}; d_{31} + d_{32}\}$, where d_{i1}, d_{i2} represent the number of originals and the number of combinations respectively provided by the ith method. For each data, in case of feature extraction method LDA, reduced dimensionality is taken to be $(l-1)$ if $d > (l-1)$, where l denotes the number of classes of the data. And when number of instances is less than the number of features, Null space based LDA (NLDA) [50] has been considered.

3.1.1 Unsupervised Case

Table 2 records the quality of Kmeans clustering in terms of Jacc, FM, NMI and NVI, for each of the comparing methods. Compared to MaxVar, UnMPeMR$_1$ has better performance on all data sets except for one (Abalone). And other two versions of UnMPeMR are performing better than MaxVar for all data sets across all evaluation metrics. Against LS, UnMPeMR$_1$ has better performance on almost all data sets except for two data sets (Abalone and Segmentation). Both LS and MaxVar have superior performances for Abalone data set over UnMPeMR$_1$. UnMPeMR$_2$ has superior performance with LS on seventeen data sets w.r.t NMI and NVI. And w.r.t Jacc and FM, it has better performance for around fifteen data sets. And UnMPeMR$_3$ is superior to LS in 80 percent of the cases. Compared to SPEC, all proposed three versions have very good performance for almost all data sets. In comparison with MCFS method, for about 80 percent of the cases, at least one of the three versions is having superior performance. For more than 60 percent of the cases, all three versions are performing better and for three data sets (Segmentation, USPS, and orlraws10P), MCFS is better than the proposed methods. On around 80 percent of the data sets, all three proposed methods performing better than ICA. Among the remaining data, for two data sets (Isolet, Ecoli) UnMPeMR$_1$ is better and for Abalone data UnMPeMR$_2$ is superior to ICA. In comparison with FA, UnMPeMR$_2$ is always better and both UnMPeMR$_1$, UnMPeMR$_3$ are superior on almost 80 percent of the data sets. For about 40 percent of the data, all versions are superior to PCA. For almost 80 percent of the data sets, at least one of the three versions of UnMPeMR is better and for other cases (Isolet, Seeds, Coil20), PCA is superior to the proposed. Against SPCA, UnMPeMR$_1$ has better performance on almost all data sets except for three data sets (Abalone, Segmentation and Waveform). For around

TABLE 2
Kmeans Clustering Performance of Each Comparing Algorithm on 18 Data Sets

DataSet	Evaluation Metric	Method										
		MaxVar	LS	SPEC	MCFS	PCA	FA	ICA	SPCA	U_1	U_2	U_3
Abalone	Jacc ↑	0.291	0.302	0.285	0.303	0.297	0.293	0.298	0.297	0.290	**0.311**	0.303
T={0.1,0.9;0.1,0.5;0.1,0.4}	FM↑	0.451	0.465	0.444	0.466	0.458	0.454	0.459	0.458	0.449	**0.480**	0.464
d=3,d_0={3+1,2+1,2+1}	NMI↑	0.191	0.170	0.162	0.169	0.197	0.191	0.199	0.196	0.191	**0.212**	0.201
	NVI↓	1.582	1.626	1.677	1.618	1.598	1.610	1.594	1.600	1.603	**1.538**	1.576
COIL20	Jacc↑	0.247	0.300	0.214	0.363	0.395	0.286	0.334	0.374	**0.397**	0.353	0.343
T={0.1,0.65;0.1,0.3;0.1,0.3}	FM↑	0.405	0.512	0.358	0.535	**0.571**	0.446	0.503	0.548	**0.571**	0.535	0.523
d=65,d_0={56+3,73+3,74+3}	NMI↑	0.597	0.699	0.553	0.719	**0.759**	0.645	0.687	0.743	0.742	0.730	0.730
	NVI↓	0.770	0.580	0.865	0.550	**0.491**	0.701	0.615	0.503	0.501	0.548	0.563
Ecoli	Jacc↑	0.451	0.465	0.424	0.502	0.505	0.417	0.516	0.473	**0.551**	0.468	0.447
T={0.3,0.5;0.35,0.3;0.3,0.3}	FM↑	0.625	0.638	0.604	0.666	0.672	0.592	0.680	0.642	**0.708**	0.642	0.624
d=5,d_0={3+2,3+2,3+2}	NMI↑	0.586	0.606	0.572	0.600	0.637	0.585	0.642	0.627	**0.642**	0.624	0.610
	NVI↓	0.880	0.834	0.925	0.837	0.761	0.887	0.761	0.779	**0.760**	0.796	0.829
Extended Yale	Jacc↑	0.032	0.020	0.022	0.042	0.022	0.020	0.019	0.025	**0.046**	0.042	0.042
T={0.15,0.8;0.15,0.3;0.15,0.3}	FM↑	0.067	0.039	0.044	0.080	0.044	0.041	0.038	0.049	**0.087**	0.080	0.081
d=70,d_0={16+51,13+56,10+52}	NMI↑	0.254	0.141	0.164	0.249	0.137	0.115	0.125	0.161	**0.278**	0.261	0.262
	NVI↓	1.733	1.690	1.653	1.486	1.708	1.728	1.733	1.661	**1.428**	1.461	1.461
Gas Sensor	Jacc↑	0.140	0.177	0.180	0.175	0.154	0.134	0.146	0.163	**0.187**	0.175	0.185
T={0.05,0.7;0.09,0.3;0.09,0.4}	FM↑	0.251	0.303	0.310	0.302	0.271	0.238	0.258	0.290	**0.337**	0.308	0.334
d=11,d_0={8+3,1+6,2+9}	NMI↑	0.171	0.272	0.193	0.253	0.191	0.117	0.176	0.227	0.281	0.245	**0.293**
	NVI↓	1.610	1.400	1.234	1.403	1.509	1.618	1.559	1.406	1.243	1.365	**1.233**
Isolet	Jacc↑	0.117	0.243	0.119	0.266	**0.322**	0.105	0.277	0.315	0.318	0.253	0.250
T={0.3,0.6;0.3,0.15;0.3,0.15}	FM↑	0.210	0.393	0.213	0.422	**0.487**	0.191	0.434	0.481	0.484	0.416	0.411
d=70,d_0={25+47,24+50,22+47}	NMI↑	0.394	0.628	0.353	0.652	**0.700**	0.418	0.649	0.672	0.691	0.637	0.630
	NVI↓	1.195	0.732	1.284	0.686	**0.585**	1.155	0.696	0.646	0.628	0.755	0.769
Liras Movement	Jacc↑	0.096	0.146	0.106	0.188	0.211	0.136	0.179	0.209	0.219	**0.220**	0.218
T={0.07,0.8;0.07,0.4;0.07,0.4}	FM↑	0.175	0.255	0.192	0.319	0.340	0.240	0.305	0.348	0.361	**0.363**	0.359
d=15,d_0={3+8,6+8,5+8}	NMI↑	0.350	0.464	0.386	0.553	0.582	0.448	0.535	0.577	0.588	**0.593**	0.592
	NVI↓	1.287	1.054	1.214	0.878	0.823	1.087	0.911	0.831	0.808	**0.795**	0.803
Multiple	Jacc↑	0.482	0.462	0.443	0.534	0.581	0.559	0.526	0.578	0.609	0.569	**0.615**
T={0.3,0.6;0.3,0.2;0.2,0.2},d=220	FM↑	0.651	0.633	0.614	0.694	0.736	0.718	0.687	0.733	0.759	0.730	**0.764**
d_0={146+61,153+70,204+14}	NMI↑	0.720	0.700	0.683	0.761	0.790	0.765	0.752	0.787	0.805	0.793	**0.815**
	NVI↓	0.549	0.592	0.625	0.470	0.416	0.463	0.487	0.421	0.395	0.406	**0.369**
ORL	Jacc↑	0.219	0.208	0.132	0.248	0.284	0.139	0.217	0.294	0.303	**0.304**	0.303
T={0.3,0.7;0.3,0.2;0.3,0.2}	FM↑	0.364	0.347	0.234	0.430	0.448	0.246	0.360	0.458	0.467	**0.468**	0.467
d=65,d_0={10+50,10+56,10+54}	NMI↑	0.727	0.707	0.625	0.743	0.778	0.639	0.726	0.776	0.782	**0.785**	0.782
	NVI↓	0.535	0.574	0.739	0.506	0.433	0.711	0.537	0.441	0.430	**0.422**	0.430
orlraws10P	Jacc↑	0.343	0.480	0.454	**0.497**	0.454	0.197	0.450	0.460	**0.497**	0.453	0.473
T={0.1,0.8;0.1,0.4;0.2,0.3},d=500	FM↑	0.514	0.656	0.629	**0.670**	0.628	0.330	0.625	0.632	**0.670**	0.627	0.647
d_0={341+113,425+118,420+82}	NMI↑	0.699	0.828	0.784	**0.835**	0.793	0.511	0.792	0.794	**0.835**	0.796	0.806
	NVI↓	0.588	0.331	0.421	**0.321**	0.405	0.957	0.403	0.403	**0.321**	0.397	0.357
Page Blocks	Jacc↑	0.343	0.372	0.395	0.354	0.396	0.357	0.356	0.393	0.411	**0.438**	0.438
T={0.05,0.7;0.18,0.85;0.16,0.85}	FM↑	0.555	0.593	0.611	0.576	0.608	0.563	0.577	0.605	0.624	0.645	**0.646**
d=5,d_0={4+1,4+1,4+1}	NMI↑	0.076	0.180	0.183	0.154	0.164	0.068	0.164	0.165	0.218	**0.224**	0.223
	NVI↓	3.304	3.245	3.065	3.344	2.997	3.304	3.293	3.006	2.831	**2.697**	2.701
pixraw10P	Jacc↑	0.180	0.559	0.219	0.463	0.592	0.594	0.485	0.604	0.574	**0.640**	0.607
T={0.3,0.95;0.25,0.8;0.25,0.7}	FM↑	0.312	0.720	0.362	0.638	0.748	0.749	0.656	0.755	0.731	**0.780**	0.755
d=10,d_0={1+9,1+13,1+9}	NMI↑	0.484	0.841	0.553	0.803	0.879	0.888	0.818	0.881	0.864	**0.895**	0.872
	NVI↓	0.978	0.312	0.871	0.382	0.235	0.216	0.355	0.232	0.265	**0.205**	0.252
Seeds	Jacc↑	0.658	0.614	0.717	0.717	**0.740**	0.628	0.660	0.680	0.728	0.730	0.728
T={0.1,0.95;0.1,0.5;0.1,0.5}	FM↑	0.793	0.761	0.835	0.835	**0.853**	0.772	0.795	0.809	0.842	0.852	0.842
d=4,d_0={3+1,3+1,3+1}	NMI↑	0.657	0.625	0.716	0.716	**0.732**	0.658	0.665	0.700	0.732	0.730	0.725
	NVI↓	0.685	0.749	0.568	0.569	**0.528**	0.681	0.668	0.598	0.549	0.549	0.539
Segmentation	Jacc↑	0.387	0.420	0.427	0.427	0.382	0.370	0.398	**0.428**	0.411	0.409	0.407
T={0.05,0.65;0.12,0.85;0.12,0.85}	FM↑	0.564	0.595	0.601	0.600	0.558	0.543	0.571	**0.600**	0.587	0.589	0.589
d=8,d_0={7+1,6+2,6+2}	NMI↑	0.597	0.630	0.627	0.640	0.618	0.600	0.622	**0.651**	0.630	0.647	0.651
	NVI↓	0.767	0.699	0.716	0.698	0.725	0.767	0.732	**0.652**	0.703	0.665	0.653
USPS	Jacc↑	0.271	0.246	0.176	**0.421**	0.342	0.359	0.334	0.346	0.368	0.377	0.375
T={0.2,0.9;0.2,0.4;0.2,0.4}	FM↑	0.428	0.397	0.307	**0.598**	0.511	0.521	0.518	0.506	0.540	0.547	0.545
d=110,d_0={69+67,59+55,59+52}	NMI↑	0.453	0.451	0.361	**0.632**	0.552	0.565	0.552	0.542	0.568	0.601	0.572
	NVI↓	1.076	1.065	1.195	**0.737**	0.884	0.874	0.895	0.898	0.858	0.817	0.855
warpPIE10P	Jacc↑	0.104	0.126	0.120	0.138	0.102	0.067	0.081	0.104	0.143	**0.147**	0.145
T={0.15,0.6;0.1,0.3;0.1,0.3}	FM↑	0.190	0.226	0.217	0.245	0.186	0.126	0.151	0.189	0.254	**0.258**	0.256
d=70,d_0={21+13,62+14,54+14}	NMI↑	0.288	0.315	0.358	0.391	0.308	0.174	0.230	0.314	0.404	**0.433**	0.415
	NVI↓	1.365	1.330	1.241	1.175	1.357	1.626	1.497	1.330	1.134	**1.110**	1.126
Waveform	Jacc↑	0.331	0.333	0.326	0.333	0.336	0.305	0.322	0.337	**0.338**	0.332	0.332
T={0.3,0.7;0.3,0.11;0.3,0.11}	FM↑	0.497	0.500	0.492	0.499	0.504	0.468	0.487	**0.504**	**0.504**	0.498	0.498
d=11,d_0={11+1,10+1,10+1}	NMI↑	0.333	0.345	0.333	0.359	0.360	0.272	0.306	**0.366**	**0.366**	0.358	0.358
	NVI↓	1.330	1.306	1.333	1.282	1.267	1.457	1.386	**1.265**	1.267	1.284	1.284
Yeast	Jacc↑	0.181	0.184	0.176	0.182	0.187	0.147	0.178	0.185	**0.189**	0.188	0.184
T={0.38,0.6;0.38,0.5;0.38,0.5}	FM↑	0.314	0.317	0.310	0.316	0.322	0.264	0.311	**0.324**	**0.324**	0.323	0.320
d=7,d_0={6+1,6+1,6+1}	NMI↑	0.286	0.293	0.290	0.295	0.294	0.235	0.287	0.292	**0.295**	0.294	0.292
	NVI↓	1.562	1.549	1.579	1.551	1.544	1.709	1.588	1.553	**1.543**	1.544	1.558
Average	Jacc ↑	0.271	0.314	0.274	0.342	0.350	0.284	0.321	0.348	0.366	0.356	0.355
	FM ↑	0.409	0.464	0.410	0.494	0.497	0.417	0.468	0.496	0.517	0.508	0.507
	NMI↑	0.437	0.494	0.439	0.529	0.526	0.439	0.496	0.526	0.551	0.548	0.546
	NVI ↓	1.194	1.093	1.178	1.027	1.015	1.197	1.095	1.013	0.959	0.964	0.964

U_1 - $UnMPeMR_1$, U_2 - $UnMPeMR_2$, U_3 - $UnMPeMR_3$.

60 percent of the cases, both UnMPeMR$_2$, UnMPeMR$_3$ have better clustering performance than SPCA. On Segmentation data set, SPCA is superior compared to all of the three methods. To summarize, w.r.t average performance over multiple data sets, each of the UnMPeMR methods is superior to all feature selection methods. UnMPeMR$_1$ (best among three versions of UnMPeMR) achieves 2.2, 2.3, 2.2 percent average improvements when measured by Jacc, FM and NMI, respectively in comparison with MCFS (best among considered feature selection methods). And with PCA (rank 1 method among feature extraction methods), UnMPeMR$_1$ achieves 1.6, 2.0, 2.5 percent average improvements when measured by Jacc, FM and NMI, respectively.

Table 3 records the clustering quality of AP in terms of the above mentioned evaluation metrics. All versions of UnMPeMR are performing better for all data sets across all evaluation metrics when compared to MaxVar. In case of LS, for about 75 percent of the data, all of UnMPeMR are superior. And for all data sets considered, at least one of the versions of the proposed method has better performance than LS. Compared to SPEC, proposed methods have very good AP clustering performance on all data sets. In comparison with MCFS, for about half of the data sets, each of the proposed achieve better performance. On 4 data sets (Isolet, Multiple, USPS and orlraws10P) MCFS has better clustering performance than all of the UnMPeMR methods. In comparison with feature extraction method ICA, for 13 data sets, all the versions of UnMPeMR method are superior. On Multiple and orlraws10P data sets, ICA has superior performance. On 14 out of 18 cases, proposed versions have better performance than FA across all evaluation metrics and for the remaining data, one of UnMPeMRs is superior. On only two data sets (Isolet, Waveform), PCA is superior to all of the UnMPeMR methods. For about 12 data sets, UnMPeMR$_1$ has better performance PCA w.r.t all measures. For about half of the number of data sets, all three variations of UnMPeMR are superior to PCA. Against SPCA, on 9 out of 18 data sets, two of proposed methods are having better performances. On only three data sets (Isolet, Multiple, Waveform), SPCA has superior clustering performance than all of proposed methods. Against MCFS (best among feature selection methods), UnMPeMR$_3$ (best of three versions, in this case) achieves 3.0, 2.3 percent average improvements when measured by Jacc, FM, respectively. When compared with PCA, UnMPeMR$_3$ achieves 1.5, 1.4 percent average improvements when measured by Jacc and FM.

Friedman's test has been suggested as preferable when comparing k algorithms over N data sets and normal distribution cannot be assumed [48], [51]. It ranks each algorithm on each data set separately. In case of ties, average ranks are assigned. Then the average ranks of all algorithms are calculated and used for comparison. If r_i^j denotes the rank of the jth algorithm on the i-the data set, let $R_j = \frac{1}{N}\sum_{i=1}^{N} r_i^j$ denotes the average rank for the jth algorithm. Under the null hypothesis (all algorithms have "equal" performance), the following Freidman statistic F_F will be distributed according to the F- distribution with $k-1$ numerator degrees of freedom and $(k-1)(N-1)$ denominator degrees of freedom:

$$F_F = \frac{(N-1)\chi_F^2}{N(k-1)-\chi_F^2},$$

$$\text{where } \chi_F^2 = \frac{12N}{k(k+1)}\left[\sum_{j=1}^{k} R_j^2 - \frac{k(k+1)^2}{4}\right].$$

If the null hypothesis is rejected by applying the Freidman test statistic, other tests such as the Nemenyi test can be employed to detect between which algorithms significant differences exist. According to this test, the performance of two algorithms is significantly different if their average ranks over all data sets differ by at least one critical difference (CD), where $CD_\alpha = q_\alpha\sqrt{\frac{k(k+1)}{6N}}$; q_α is based on the Studentized range statistic [49].

Table 4 summarizes the Freidman statistic F_F and the corresponding critical values on each evaluation metric of Kmeans clustering method. As shown in the table, at significance level 0.1 (eleven algorithms, eighteen data sets), the null hypothesis is clearly rejected in terms of each evaluation metric of the clustering method. Fig. 4 provide the CD diagrams for each evaluation criterion with the average rank of each comparing algorithm is marked along the axis (lower ranks to the right). Groups of algorithms that are not significantly different according to Nemenyi test are connected with a thick line. The critical difference (CD=3.29 at 0.1 significance level) is also shown above the axis in each subfigure. From Fig. 4, we observe that the values of each evaluation metric of Kmeans with the proposed are statistically better than those with MaxVar, SPEC, LS, FA and ICA. In terms of each evaluation metric, proposed achieves statistically comparable performance with SPCA, PCA and MCFS.

3.1.2 Supervised Case

Table 5 shows the classification accuracy rates of the Knn (K=1, 3 and 5) and SVM classifiers on the 18 data sets for each of the supervised DR algorithms.

In comparison with Fisher Score, ReliefF and DISR, for about 60 percent of the data sets, all versions of the proposed method provided better accuracy w.r.t all Knn classifiers. For remaining cases, atleast one of the proposed methods have better accuracy to all of them w.r.t Knn classifier. Compared to SPEC, SMPeMR has superior Knn classification accuracy on all data sets. Feature selection method TR has better accuracy than the proposed SMPeMR$_1$ on three data sets (Abalone, PageBlocks, Segmentation) only. SMPeMR$_2$ has better accuracy on all data sets, and SMPeMR$_3$ is better on all data sets except for one (PageBlocks) than TR. Compared to MRSF, SMPeMR$_1$ is better for about 70 percent of the data sets and SMPeMR$_2$ has superior accuracy on all data sets except for one (USPS). SMPeMR$_1$ superceded LDA on almost half of the data sets. And, SMPeMR$_2$ has better accuracy than LDA for all data sets except for Isolet, ORL, orlraws and Waveform. For about 10 data sets, the other version of SMPeMR is superior to LDA. On only two data sets, Isolet and orlraws10P, LDA is superior to all of the proposed methods in terms of Knn classification accuracy. For almost 60 percent of the data sets, one of the proposed methods is better than MMC. For Isolet and Coil20 data sets, MMC has better accuracy. Feature extraction method ALDE provides better accuracy than all SMPeMR methods on only 3 data sets (Coli20, Isolet and orlraws10P) w.r.t Knn (K=1) and on 4 data sets w.r.t Knn (K=3) accuracy. All three versions of SMPeMR are better than ALDE for about 60 percent of the data sets. For almost all data sets, proposed methods are superior to MFA. With QMI, each of the proposed is superior except for one case (Abalone). SMPeMR$_1$ is superior to SLDA on almost half of the data sets w.r.t. Knn accuracy. Both SMPeMR$_2$ and

TABLE 3
AP Clustering Performance of Each Comparing Algorithm on 18 Data Sets

DataSet	Evaluation Metric	Method										
		MaxVar	LS	SPEC	MCFS	PCA	FA	ICA	SPCA	U_1	U_2	U_3
Abalone	Jacc↑	0.265	0.287	0.280	0.287	0.292	0.291	0.295	0.293	0.285	**0.308**	0.296
T={0.1,0.9;0.1,0.5;0.1,0.4}	FM↑	0.426	0.446	0.438	0.446	0.453	0.451	0.456	0.454	0.446	**0.468**	0.458
d=3,d_0={3+1,2+1,2+1}	NMI↑	0.168	0.187	0.149	0.187	0.197	0.195	0.202	0.196	0.198	**0.210**	0.208
	NVI↓	1.656	1.608	1.699	1.608	1.575	1.580	1.563	1.577	1.570	**1.540**	1.541
COIL20	Jacc↑	0.212	0.369	0.183	0.383	0.379	0.322	0.334	0.376	**0.399**	0.353	0.343
T={0.1,0.65;0.1,0.3;0.1,0.3}	FM↑	0.382	0.540	0.317	0.555	0.555	0.487	0.502	0.550	**0.560**	0.525	0.513
d=65,d_0={56+3,73+3,74+3}	NMI↑	0.580	0.710	0.512	0.716	0.728	0.669	0.672	0.722	**0.744**	0.720	0.714
	NVI↓	0.765	0.560	0.933	0.562	0.528	0.657	0.647	0.544	**0.515**	0.548	0.563
Ecoli	Jacc↑	0.642	0.455	0.429	0.454	0.673	0.421	0.681	0.642	0.672	**0.682**	0.679
T={0.3,0.5;0.35,0.3;0.3,0.3}	FM↑	0.782	0.632	0.612	0.627	0.804	0.601	0.810	0.782	0.803	**0.810**	0.809
d=5,d_0={3+2,3+2,3+2}	NMI↑	0.625	0.609	0.583	0.570	0.672	0.590	0.670	0.657	0.668	**0.675**	0.670
	NVI↓	0.765	0.832	0.914	0.895	0.662	0.878	0.669	0.696	0.670	**0.660**	0.667
Extended Yale	Jacc↑	0.033	0.020	0.022	0.047	0.030	0.030	0.021	0.034	**0.049**	0.042	0.043
T={0.15,0.8;0.15,0.3;0.15,0.3}	FM↑	0.071	0.039	0.043	0.090	0.059	0.056	0.042	0.066	**0.098**	0.080	0.078
d=70,d_0={16+51,13+56,10+52}	NMI↑	0.256	0.135	0.158	0.277	0.187	0.167	0.141	0.212	**0.308**	0.277	0.276
	NVI↓	1.401	1.685	1.658	1.420	1.607	1.637	1.698	1.558	**1.360**	1.441	1.451
Gas Sensor	Jacc↑	0.155	0.178	0.180	0.175	0.172	0.136	0.169	0.180	**0.185**	0.176	0.171
T={0.05,0.7;0.09,0.3;0.09,0.4}	FM↑	0.285	0.309	0.350	0.310	0.306	0.242	0.297	0.340	**0.356**	0.319	0.306
d=11,d_0={8+3,1+6,2+9}	NMI↑	0.137	0.261	0.246	0.259	0.238	0.124	0.233	0.240	**0.270**	0.259	0.237
	NVI↓	1.477	1.367	1.266	1.339	1.357	1.642	1.398	1.270	**1.241**	1.293	1.353
Isolet	Jacc↑	0.163	0.215	0.127	**0.287**	0.262	0.133	0.226	0.265	0.240	0.256	0.254
T={0.3,0.6;0.3,0.15;0.3,0.15}	FM↑	0.283	0.357	0.227	**0.449**	0.418	0.232	0.370	0.420	0.378	0.401	0.370
d=70,d_0={25+47,24+50,22+47}	NMI↑	0.540	0.637	0.473	**0.697**	0.672	0.500	0.635	0.680	0.639	0.645	0.649
	NVI↓	0.893	0.710	1.037	**0.591**	0.642	0.988	0.722	0.622	0.710	0.713	0.711
Libras Movement	Jacc↑	0.097	0.147	0.108	0.187	0.202	0.146	0.182	0.201	**0.222**	0.207	0.214
T={0.07,0.8;0.07,0.4;0.07,0.4}	FM↑	0.177	0.259	0.196	0.316	0.337	0.255	0.311	0.335	**0.366**	0.346	0.356
d=15,d_0={3+8,6+8,5+8}	NMI↑	0.356	0.463	0.385	0.539	0.562	0.471	0.548	0.566	**0.588**	0.571	0.581
	NVI↓	1.276	1.047	1.218	0.906	0.863	1.047	0.885	0.858	**0.806**	0.837	0.819
Multiple	Jacc↑	0.506	0.477	0.320	0.595	0.590	0.598	0.633	**0.635**	0.605	0.609	0.606
T={0.3,0.6;0.3,0.2;0.2,0.2},d=220	FM↑	0.673	0.647	0.485	0.750	0.742	0.749	**0.775**	0.775	0.750	0.756	0.755
d_0={146+61,153+70,204+14}	NMI↑	0.709	0.702	0.537	0.785	0.759	0.791	**0.787**	0.785	0.770	0.771	0.773
	NVI↓	0.557	0.520	0.514	**0.401**	0.482	0.412	0.424	0.419	0.456	0.457	0.450
ORL	Jacc↑	0.199	0.176	0.101	0.246	0.279	0.149	0.207	0.296	0.306	**0.310**	0.300
T={0.3,0.7;0.3,0.2;0.3,0.2}	FM↑	0.338	0.308	0.186	0.398	0.443	0.261	0.346	0.459	0.459	**0.475**	0.464
d=65,d_0={10+50,10+56,10+54}	NMI↑	0.710	0.690	0.585	0.733	**0.773**	0.654	0.709	0.762	0.760	**0.773**	0.769
	NVI↓	0.566	0.601	0.809	0.527	**0.445**	0.680	0.571	0.469	0.451	0.450	0.456
orlraws10P	Jacc↑	0.292	0.528	0.380	**0.601**	0.504	0.470	0.540	0.495	0.445	0.523	0.553
T={0.1,0.8;0.1,0.4;0.2,0.3},d=500	FM↑	0.458	0.694	0.566	**0.723**	0.671	0.652	0.702	0.663	0.616	0.687	0.707
d_0={341+113,425+118,420+82}	NMI↑	0.646	0.824	0.735	**0.846**	0.791	0.703	0.816	0.786	0.763	0.802	0.822
	NVI↓	0.682	0.306	0.519	**0.245**	0.415	0.475	0.366	0.424	0.468	0.393	0.353
Page Blocks	Jacc↑	0.471	0.376	0.403	0.365	0.467	0.480	0.416	0.500	0.503	**0.518**	0.500
T={0.05,0.7;0.18,0.85;0.16,0.85}	FM↑	0.643	0.594	0.616	0.585	0.653	0.661	0.622	0.691	0.690	**0.698**	0.684
d=5,d_0={4+1,4+1,4+1}	NMI↑	0.144	0.187	0.199	0.168	0.144	0.185	0.187	0.198	**0.201**	0.191	0.186
	NVI↓	2.634	3.117	2.923	3.224	2.689	2.710	2.983	2.391	**2.448**	2.516	2.566
pixraw10P	Jacc↑	0.173	0.514	0.220	0.558	0.724	0.667	0.610	0.654	0.634	0.617	**0.757**
T={0.3,0.95;0.25,0.8;0.25,0.7}	FM↑	0.328	0.685	0.363	0.717	0.840	0.804	0.759	0.793	0.781	0.767	**0.862**
d=10,d_0={1+9,1+13,1+9}	NMI↑	0.467	0.822	0.556	0.831	0.912	0.918	0.868	0.886	0.901	0.888	**0.920**
	NVI↓	0.944	0.345	0.871	0.336	0.174	0.162	0.260	0.224	0.193	0.219	**0.160**
Seeds	Jacc↑	0.644	0.611	0.704	0.738	0.708	0.610	0.678	0.666	0.730	**0.750**	0.751
T={0.1,0.95;0.1,0.5;0.1,0.5}	FM↑	0.784	0.759	0.826	0.849	0.829	0.757	0.808	0.800	0.847	**0.857**	0.859
d=4,d_0={3+1,3+1,3+1}	NMI↑	0.654	0.620	0.703	0.733	0.708	0.636	0.685	0.702	0.740	**0.745**	0.743
	NVI↓	0.691	0.759	0.593	0.534	0.582	0.724	0.629	0.593	0.507	**0.505**	0.509
Segmentation	Jacc↑	0.348	0.357	0.293	0.426	0.411	0.374	**0.435**	0.430	0.426	0.410	0.389
T={0.05,0.65;0.12,0.85;0.12,0.85}	FM↑	0.521	0.529	0.459	0.595	0.588	0.548	0.603	**0.604**	**0.604**	0.593	0.574
d=8,d_0={7+1,6+2,6+2}	NMI↑	0.581	0.630	0.493	0.660	0.650	0.613	0.650	0.665	**0.670**	0.662	0.648
	NVI↓	0.792	0.720	0.937	0.669	0.661	0.741	0.694	0.650	**0.628**	0.631	0.661
USPS	Jacc↑	0.261	0.242	0.231	**0.373**	0.319	0.342	0.315	0.341	0.335	0.371	0.362
T={0.2,0.9;0.2,0.4;0.2,0.4}	FM↑	0.414	0.392	0.373	**0.545**	0.485	0.511	0.479	0.508	0.512	**0.545**	0.535
d=110,d_0={69+67,59+55,59+52}	NMI↑	0.440	0.431	0.413	**0.573**	0.516	0.552	0.506	0.526	0.531	0.571	0.560
	NVI↓	1.106	1.104	1.112	**0.850**	0.958	0.884	0.982	0.946	0.932	0.860	0.889
warpPIE10P	Jacc↑	0.127	0.147	0.128	0.163	0.109	0.061	0.104	0.092	**0.206**	0.162	0.184
T={0.15,0.6;0.1,0.3;0.1,0.3}	FM↑	0.240	0.264	0.231	0.285	0.198	0.115	0.189	0.170	**0.342**	0.282	0.312
d=70,d_0={21+13,62+14,54+14}	NMI↑	0.369	0.381	0.373	0.459	0.312	0.148	0.308	0.269	**0.508**	0.438	0.463
	NVI↓	1.145	1.179	1.199	1.043	1.331	1.671	1.366	1.439	**0.968**	1.089	1.054
Waveform	Jacc↑	0.312	0.324	0.218	0.323	**0.347**	0.322	0.329	0.340	0.337	0.335	0.331
T={0.3,0.7;0.3,0.11;0.3,0.11}	FM↑	0.474	0.488	0.355	0.487	**0.514**	0.485	0.493	0.510	0.500	0.500	0.495
d=11,d_0={11+1,10+1,10+1}	NMI↑	0.245	0.282	0.224	0.283	**0.376**	0.293	0.280	0.350	0.333	0.307	0.295
	NVI↓	1.508	1.439	1.947	1.432	**1.274**	1.420	1.443	1.290	1.341	1.388	1.416
Yeast	Jacc↑	0.190	0.188	0.173	0.188	0.190	0.150	0.171	**0.205**	0.195	**0.205**	0.195
T={0.38,0.6;0.38,0.5;0.38,0.5}	FM↑	0.325	0.324	0.305	0.324	0.326	0.316	0.304	0.339	0.331	**0.341**	0.335
d=7,d_0={6+1,6+1,6+1}	NMI↑	0.280	0.298	0.275	0.298	0.299	0.249	0.274	**0.295**	0.292	**0.295**	0.293
	NVI↓	1.541	1.531	1.619	1.531	1.539	1.529	1.630	**1.525**	1.530	1.530	1.530
Average	Jacc ↑	0.283	0.312	0.250	0.355	0.370	0.317	0.353	0.369	0.376	0.380	0.385
	FM↑	0.422	0.459	0.386	0.503	0.512	0.455	0.493	0.514	0.524	0.525	0.526
	NMI↑	0.439	0.493	0.422	0.534	0.528	0.470	0.509	0.528	0.549	0.544	0.545
	NVI↓	1.133	1.079	1.209	1.006	0.988	1.102	1.052	0.972	0.933	0.948	0.953

U_1 - $UnMPeMR_1$, U_2 - $UnMPeMR_2$, U_3 - $UnMPeMR_3$.

TABLE 4
Summary of Friedman Statistics F_F and the Critical Value in Terms of Each Evaluation Metric

Evaluation Metric	F_F	critical value ($\alpha = 0.1$)
Jacc	15.90	1.64
FM	14.84	
NMI	19.43	
NVI	16.82	

$SMPeMR_3$ have better accuracy than SLDA for almost 60 percent of the cases. According to Knn (K=1) classifier, $SMPeMR_3$ obtains 6.9, 11.0, 4.5, 19.0, 6.1, 13.0 percent average improvements in comparison with feature selection methods Fisher, ReliefF, DISR, SPEC, TR and MRSF respectively, and with feature extraction methods LDA, MMC, ALDE, MFA, QMI and SLDA, it obtains 5.8, 2.2, 1.2, 10.0, 10.7, 4.0 percent average improvements respectively.

In terms of SVM accuracy, all SMPeMR methods are better than Fisher Score, for about 80 percent of the data sets, and for others, one of the SMPeMR methods is better. Compared to ReliefF and TR, for almost all data sets, proposed methods are superior. DISR has better accuracy than all of the $SMPeMR_2$ methods for only three data sets (Ecoli, Isolet and warPIE10P). And for almost all other data sets, proposed methods have better SVM accuracy. And in comparison with SPEC, all of the proposed methods have better SVM accuracy. $SMPeMR_1$ is better than MRSF on all data sets except for Coil20 and Gas Sensor data sets. And other two versions of supervised MPeMR have superior SVM classification accuracy than MRSF for around 70 percent of the cases. All SMPeMR methods are superior to LDA for about 60 percent of the cases. LDA is better than all SMPeMR methods on only 3 data sets (Abalone, Ecoli and Isolet). MMC has better SVM accuracy than all of the proposed methods for only three data sets (Ecoli, Isolet and USPS), and in remaining cases, at least one of the proposed is better. With ALDE, for about 60 percent of the data, all SMPeMR methods are superior. All the supervised CFG methods are achieved better accuarcy than QMI on almost all data sets. Against MFA, proposed techniques achieve superior accuracy except for Ecoli and warpPIE10P. For more than 70 percent of the data sets, atleast one of SMPeMR methods have superseded SLDA in terms of SVM accuracy. In comparison with feature selection methods Fisher, ReliefF, DISR, SPEC, TR and MRSF, $SMPeMR_2$ improves average accuracy by 11.9, 11.9, 6.2, 20.6, 9.0, 12.1 percent respectively, and with LDA, MMC, ALDE, MFA, QMI and SLDA, it obtains 2.8, 3.5, 1.9, 10.0, 9.6, 3.5 percent improvements respectively.

Table 6 summarizes the Freidman statistic F_F and the corresponding critical values on each classification accuracy. As shown in the table, at significance level 0.1 (fifteen algorithms, eighteen data sets), the null hypothesis is clearly rejected in terms of classification accuracy of each considered classifier. Fig. 5 gives the CD diagrams for each accuracy with average rank of each comparing algorithm is marked along the axis (lower ranks to the right). Groups of algorithms that are not significantly different according to Nemenyi test are connected with a thick line. The critical difference (CD=4.70 at 0.1 significance level) is also shown above the axis in each sub-figure. From Fig. 5, we observe that the proposed is statistically superior than

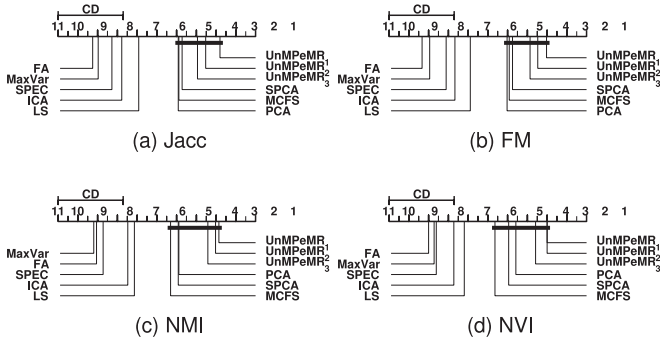

Fig. 4. Kmeans Clustering Performance Comparison of 11 DR algorithms against each other with the Nemenyi test.

the feature extraction methods MFA and QMI w.r.t all classifiers. There is no consistent evidence to indicate statistical differences between the proposed and feature extraction methods LDA, MMC, ALDE and SLDA. Also, with feature selection method DISR, proposed has statistically comparable accuracy w.r.t SVM classifier. The accuracy of Knn (K=1 or 3) with the proposed is statistically better than that of TR.

3.1.3 Sensitivity Analysis

In the proposed CFG method, T_1 and T_2 control the size of the reduced set. Since T_1 and T_2 determine the error thresholds, data representation is controlled by their choice. We investigate the effect of T_1 and T_2 by analyzing how changes of T_1 and T_2 affect the performance of the proposed method in both supervised and unsupervised learning. As an evaluation measure, we have considered NMI in case of clustering (Kmeans) and accuracy rate (Knn1) in case of classification. And two data sets, namely, PageBlocks and Gas Sensor have been considered. We vary the value of T_1 as $\{0.01, 0.03, 0.05,\ 0.07, 0.09\}$ and T_2 as $\{0.1, 0.3, 0.5, 0.7,\ 0.9\}$. Figs. 6 and 7 show the effect of T_1, T_2 for all UnMPeMR, SMPeMR methods on both data sets. For the above two data sets, for each of the supervised and unsupervised CFG methods, different combinations of T_1, T_2 values provide different classification/clustering performances. For each data, there is some combination of T_1, T_2 which provide better performance. It has been observed that, for each fixed value of T_1, as T_2 (which controls the level of redundancy) increases, the clustering/classification performance (NMI/Accuracy) starts increasing first, then after it gets decreased. Performance decreases if either the lower level of redundancy (i.e., if T_2 = 0.1 and the features with $Red(u_i, u_j) > 0.1$) or no redundancy (i.e., if $T_2 = 0.9$ and the features with $Red(u_i, u_j) > 0.9$) is removed i.e., the performance would decrease if very large number of features are taken to be redundant or very few number of features are taken to be redundant. It is also noticed in some cases that for smaller values of T_1, i.e., reduced set contains no semi-features and for larger values of T_1 i.e., if all reduced features are semi-features (i.e., no redundancy is removed and redundant features are involved in the combination) then also performance decreases.

It has been observed experimentally that one can get as many features in the reduced data set as we want by setting appropriate values to thresholds T_1, T_2. But it is not known how to fix the values of T_1 and T_2 before hand for a given data set when d is a user given value.

TABLE 5
Classification Performance (Accuracy) of Each Comparing Algorithm on 18 Data Sets

DataSet	Classifier	Method														
		Fisher	ReliefF	DISR	SPEC	TR	MRSF	LDA	MMC	ALDE	MFA	QMI	SLDA	S_1	S_2	S_3
Abalone	Knn1	0.5463	0.5370	0.5396	0.4757	0.5804	0.5485	0.5548	0.5362	0.5230	0.5077	**0.6007**	0.5582	0.5761	0.5768	0.5440
T={0.001,0.9;0.001,0.5;0.02,0.6}	Knn3	0.5600	0.5614	0.5710	0.5157	0.6138	0.5655	0.5870	0.5675	0.5534	0.5334	**0.6434**	0.5908	0.6071	0.6095	0.5773
d=2,d_0={1+1,1+1,1+1}	Knn5	0.5899	0.5837	0.5877	0.5157	0.63966	0.5923	0.6031	0.5864	0.5720	0.5494	**0.6452**	0.6061	0.6269	0.6335	0.6229
	SVM	0.5760	0.5605	0.5755	0.4941	0.5965	0.5684	0.6071	0.4960	0.5828	0.4945	0.583	**0.6076**	0.5843	0.5843	0.5964
Coil20	Knn1	0.9680	0.9730	0.9690	0.8630	0.9550	0.9600	0.9465	**0.9854**	0.9847	0.8797	0.8129	0.9601	0.9527	0.9679	0.9783
T={0.045,0.9;0.045,0.6;0.045,0.3}	Knn3	0.9450	0.9570	0.9350	0.8130	0.9350	0.9370	0.9468	**0.9751**	0.9721	0.8610	0.7737	0.9538	0.9157	0.9379	0.9571
d=130,d_0={47+74,49+107,16+108}	Knn5	0.9170	0.9260	0.9030	0.7470	0.9140	0.9130	0.9468	**0.9670**	0.9588	0.8429	0.7424	0.9508	0.8941	0.9217	0.9366
	SVM	0.9060	0.9000	0.8820	0.7220	0.8730	0.9280	0.8751	0.8948	0.8646	0.8339	0.8667	0.9037	0.9155	0.9218	**0.9368**
Ecoli	Knn1	0.7835	0.8252	0.7831	0.7324	0.7832	0.8381	0.7911	0.7831	0.7676	0.8006	0.7424	0.8327	0.8177	0.8360	**0.8542**
T={0.1,0.7;0.15,0.85;0.1,0.8}	Knn3	0.7913	0.8483	0.7920	0.7806	0.7959	0.8374	0.8145	0.8115	0.7776	0.8316	0.7706	0.8618	0.8137	0.8563	**0.8620**
d=5,d_0={4+1,4+1,4+1}	Knn5	0.8226	0.8701	0.8159	0.7627	0.8228	0.8555	0.8225	0.8259	0.7856	0.8419	0.7927	**0.8725**	0.8457	0.8598	0.8651
	SVM	0.7544	0.7160	0.7570	0.6803	0.7403	0.7526	**0.8067**	0.7841	0.7752	0.7732	0.7687	0.7818	0.7689	0.7359	0.7599
Extended Yale	Knn1	0.6379	0.5186	0.7611	0.6454	0.7730	0.6489	0.8892	0.8848	0.8520	0.6309	0.6940	0.9039	0.8672	0.9210	**0.9211**
T={0.19,0.9;0.18,0.5;0.18,0.5}	Knn3	0.6234	0.3665	0.6240	0.3199	0.6467	0.6206	0.8894	0.8740	0.8517	0.6323	0.6896	0.9002	0.8607	0.9159	**0.9177**
d=100,d_0={76+25,10+105,3+98}	Knn5	0.5969	0.3899	0.6000	0.3380	0.6900	0.6161	0.8890	0.8747	0.8269	0.6216	0.6610	0.8925	0.8627	0.8999	**0.9073**
	SVM	0.4154	0.3724	0.4945	0.3905	0.5545	0.3667	0.6552	0.6605	0.6627	0.6265	0.6720	0.7567	0.7023	**0.7702**	0.7600
Gas Sensor	Knn1	0.9826	0.9852	0.9862	0.9697	0.9920	0.9938	0.9932	0.9875	0.9852	0.9911	0.9817	0.9883	0.9903	0.9940	**0.9947**
T={0.0001,0.75;0.07,0.5;0.15,0.75}	Knn3	0.9874	0.9856	0.9818	0.9590	0.9869	0.9916	0.9914	0.9855	0.9816	0.9872	0.9761	0.9900	0.9885	0.9926	**0.9928**
d=13,d_0={12+1,5+8,1+13}	Knn5	0.9861	0.9833	0.9885	0.9588	0.9867	0.9913	0.9890	0.9845	0.9793	0.9849	0.9716	0.9880	0.9861	0.9902	**0.9922**
	SVM	0.9869	0.9803	0.9875	0.9522	0.9822	0.9893	0.9872	0.9830	0.9764	0.9812	0.9670	0.8711	0.9846	0.9889	**0.9902**
Isolet	Knn1	0.8769	0.8513	0.8897	0.6372	0.8692	0.8897	0.9197	0.9210	0.9337	0.8219	0.7334	**0.9403**	0.8316	0.9036	0.9046
T={0.01,0.73;0.05,0.5;0.03,0.45}	Knn3	0.8833	0.8397	0.9000	0.6423	0.8756	0.8906	0.9107	0.9225	0.9363	0.8090	0.7201	**0.9473**	0.8337	0.8977	0.8997
d=150,d_0={146+4,67+72,58+80}	Knn5	0.8705	0.8487	0.9013	0.6641	0.8795	0.8980	0.9178	0.9237	0.9364	0.8174	0.7207	**0.9478**	0.8377	0.9017	0.9047
	SVM	0.8768	0.8462	0.9000	0.6551	0.8821	0.8256	0.9227	0.9247	**0.9346**	0.8296	0.7273	0.7909	0.9126	0.8936	0.8946
Libras Movement	Knn1	0.6569	0.7591	0.7420	0.6074	0.6535	0.8684	0.6448	0.7972	0.8487	0.6848	0.7072	0.7361	0.8292	**0.8737**	0.8221
T={0.025,0.95;0.03,0.5;0.07,0.55}	Knn3	0.5716	0.6911	0.6960	0.5353	0.5690	0.8330	0.6236	0.7330	0.7908	0.5809	0.7102	0.7145	0.8079	**0.8507**	0.7878
d=21,d_0={19+1,19+1,19+2}	Knn5	0.5904	0.6713	0.6789	0.5217	0.5733	0.7987	0.6160	0.7051	0.7719	0.5652	0.6751	0.7076	0.8033	**0.8456**	0.8309
	SVM	0.2792	0.3332	0.3299	0.2095	0.2969	0.4497	0.3667	0.4200	0.4099	0.2960	0.3998	**0.4945**	0.4608	0.4838	0.4149
Multiple	Knn1	0.9820	0.9785	0.9825	0.9530	0.9805	0.9850	0.9730	0.9805	0.9795	0.9687	0.9424	0.9831	0.9935	**0.9968**	0.9934
T={0.05,0.88;0.06,0.9;0.06,0.65}	Knn3	0.9820	0.9820	0.9830	0.9610	0.9820	0.9845	0.9733	0.9807	0.9788	0.9706	0.9206	0.9867	0.9938	**0.9971**	0.9940
d=230	Knn5	0.9810	0.9810	0.9840	0.9610	0.9810	0.9840	0.9717	0.9792	0.9797	0.9699	0.9267	0.9865	0.9931	**0.9944**	0.9936
d_0={117+113,92+136,85+147}	SVM	0.8780	0.8555	0.8750	0.8235	0.8770	0.8700	0.8991	0.8503	0.8638	0.9649	0.7670	**0.9702**	0.9158	0.8770	0.8880
ORL	Knn1	0.6411	0.5964	0.7521	0.3346	0.6432	0.5657	0.7598	0.8271	0.8567	0.5622	0.6800	0.8469	**0.8993**	0.6707	0.8243
T={0.15,0.8;0.15,0.3;0.15,0.3}	Knn3	0.5861	0.5704	0.7136	0.3236	0.5957	0.5668	0.7662	0.7878	0.8198	0.5289	0.6596	0.8444	**0.9012**	0.6464	0.7814
d=30,d_0={1+27,5+31,3+25}	Knn5	0.5154	0.5179	0.6532	0.2936	0.5239	0.5704	0.7493	0.7497	0.7828	0.4813	0.6191	0.8454	**0.8982**	0.6396	0.7389
	SVM	0.3529	0.4196	0.5871	0.2039	0.3454	0.5471	0.6687	0.6791	0.6919	0.3964	0.5500	0.6684	0.6729	0.6443	**0.7186**
orlraws10P	Knn1	0.7914	0.6486	0.8814	0.4757	0.8029	0.4800	**0.9220**	0.8976	0.9131	0.8211	0.7032	0.4313	0.9126	0.8721	0.8730
T={0.01,0.6;0.01,0.4;0.01,0.35}	Knn3	0.7714	0.6086	0.8400	0.4514	0.7871	0.4700	**0.9242**	0.8903	0.9050	0.8024	0.7200	0.4313	0.9215	0.9028	0.8825
d=35,d_0={24+13,29+7,25+9}	Knn5	0.7429	0.5529	0.7943	0.4043	0.7614	0.4614	0.9202	0.8910	0.9060	0.7829	0.7300	0.4313	**0.9228**	0.9029	0.8822
	SVM	0.5557	0.4814	0.7914	0.3900	0.5886	0.4814	0.8731	0.8767	0.8758	0.7410	0.7080	0.4439	0.8831	**0.9021**	0.8922
PageBlocks	Knn1	0.9638	0.9629	0.9562	0.9563	0.9640	0.9644	0.9636	0.9625	0.9622	0.9638	0.9520	0.9620	0.9608	**0.9817**	0.9498
T={0.1,0.95;0.1,0.75;0.1,0.75}	Knn3	0.9673	0.9660	0.9605	0.9578	0.9675	0.9684	0.9659	0.9649	0.9657	0.9663	0.9526	0.9657	**0.9819**	0.9633	0.9512
d=6,d_0={3+3,2+4,2+4}	Knn5	0.9671	0.9653	0.9591	0.9598	0.9666	0.9666	0.9661	0.9649	0.9659	0.9651	0.9600	0.9660	**0.9854**	0.9593	0.9533
	SVM	0.7482	0.7008	0.7748	0.7869	0.7256	0.7108	0.7466	0.7466	**0.8787**	0.7412	0.7395	0.7555	0.8497	0.8455	0.8474
pixraw10P	Knn1	0.9057	0.7200	0.9429	0.7100	0.9129	0.7500	0.9651	0.9650	0.9544	0.9395	0.8937	0.9146	0.9375	0.9574	**0.9693**
T={0.005,0.98;0.005,0.8;0.005,0.7}	Knn3	0.9071	0.6586	0.9229	0.6571	0.9029	0.7057	**0.9641**	0.9481	0.9469	0.9228	0.8730	0.9132	0.9302	0.9427	0.9399
d=20,d_0={2+21,3+14,1+20}	Knn5	0.9143	0.6171	0.9114	0.5700	0.9343	0.6771	**0.9661**	0.9453	0.9386	0.9201	0.8620	0.9115	0.9292	0.9394	0.9429
	SVM	0.5000	0.5129	0.8729	0.3743	0.8903	0.6700	0.8979	0.8828	0.8761	0.8697	0.8038	**0.9150**	0.8905	0.9005	0.9113
Seeds	Knn1	0.8286	0.9095	0.8143	0.8667	0.8429	0.8714	0.9133	0.9098	0.9081	0.8695	0.8364	0.9476	0.9126	0.9408	**0.9574**
T={0.03,0.75;0.05,0.65;0.05,0.65}	Knn3	0.8476	0.8857	0.8524	0.8524	0.8667	0.9048	0.9305	0.9327	0.9238	0.8814	0.8167	0.9333	0.8917	**0.9608**	0.9298
d=3,d_0={2+1,3+1,2+1}	Knn5	0.8524	0.9095	0.8571	0.8905	0.8762	0.9048	0.9376	0.9293	0.9276	0.8914	0.8277	0.9381	0.9178	**0.9721**	0.9571
	SVM	0.6905	0.7905	0.6810	0.7857	0.6810	0.8286	0.9071	0.9068	0.8933	0.7605	0.7913	0.9143	0.8964	**0.9610**	0.9373
Segmentation	Knn1	0.9710	0.9758	0.9719	0.9290	0.9714	0.4511	0.9745	0.9584	0.9565	0.9358	0.9510	0.9735	0.9687	**0.9941**	0.9917
T={0.005,0.8;0.02,0.85;0.02,0.7}	Knn3	0.9654	0.9719	0.9641	0.9242	0.9641	0.4338	0.9544	0.9484	0.9455	0.9298	0.9406	0.9710	0.9527	**0.9827**	0.9764
d=8,d_0={7+1,5+3,5+2}	Knn5	0.9602	0.9641	0.9602	0.9156	0.9615	0.4788	0.9496	0.9484	0.9449	0.9203	0.9600	0.9685	0.9467	**0.9805**	0.9776
	SVM	0.5933	0.5892	0.6055	0.4641	0.6103	0.2887	0.5687	0.5801	0.5536	0.4171	0.3948	**0.6744**	0.5961	0.6156	0.6170
USPS	Knn1	0.8312	0.9054	0.8149	0.5851	0.8244	0.9280	0.6133	0.9002	0.9385	0.5475	0.6090	0.9339	**0.9532**	0.9188	0.9052
T={0.1,0.9;0.15,0.5;0.15,0.4}	Knn3	0.8452	0.9140	0.8228	0.6200	0.8421	0.9387	0.6281	0.9079	0.7591	0.5744	0.6380	0.9416	**0.9584**	0.9295	0.9212
d=30,d_0={5+25,2+13,5+29}	Knn5	0.8516	0.9117	0.8235	0.6222	0.8550	0.9387	0.6539	0.9176	0.9413	0.5994	0.6465	0.9434	**0.9573**	0.9342	0.9216
	SVM	0.6989	0.7688	0.7029	0.5463	0.7008	0.7796	0.6745	0.7735	0.7926	0.5040	0.6644	0.8431	**0.8446**	0.7517	0.7291
warPIE10P	Knn1	0.9457	0.8357	0.9400	0.6107	0.9336	0.7000	**0.9916**	0.9676	0.9316	0.9706	0.9006	0.8510	0.9687	0.9805	0.9820
T={0.05,0.9;0.05,0.5;0.05,0.5}	Knn3	0.8864	0.7671	0.8993	0.5493	0.8971	0.7050	0.9596	0.9594	0.9080	0.9293	0.9015	0.8509	0.9699	**0.9800**	0.9792
d=150,d_0={31+113,6+151,17+148}	Knn5	0.8343	0.6907	0.8621	0.4579	0.8421	0.7200	0.9788	0.9511	0.8862	0.8679	0.8928	0.8509	0.9694	**0.9836**	0.9804
	SVM	0.9001	0.9257	0.9293	0.8664	0.9164	0.7729	0.8980	0.8887	0.9145	**0.9616**	0.8809	0.9024	0.8997	0.9302	0.9389
Waveform	Knn1	0.7404	0.7092	0.7692	0.5984	0.7474	0.7394	0.8112	0.6541	0.8082	0.6360	0.6800	**0.8215**	0.7995	0.8025	0.7995
T={0.01,0.8;0.01,0.8;0.01,0.8}	Knn3	0.7708	0.7432	0.7986	0.6252	0.7706	0.7646	0.8338	0.6876	0.8308	0.6710	0.6680	0.8407	0.8261	0.8291	**0.8521**
d=9,d_0={3+6,3+6,3+6}	Knn5	0.7776	0.7542	0.8086	0.6484	0.7786	0.7762	0.8442	0.7031	0.8393	0.6903	0.6910	**0.8481**	0.8345	0.8364	0.8365
	SVM	0.7664	0.7490	0.7820	0.6732	0.7622	0.7678	0.8089	0.6951	0.8075	0.7041	0.7090	0.8074	**0.8298**	0.8273	0.8279
Yeast	Knn1	0.5378	0.5324	0.5371	0.4759	0.5586	0.3242	0.5338	0.5198	0.5249	0.5072	0.5002	0.5329	0.5582	0.5590	**0.5686**
T={0.05,0.75;0.04,0.6;0.03,0.55}	Knn3	0.5727	0.5506	0.5735	0.4826	0.5748	0.3543	0.5492	0.5473	0.5575	0.5305	0.5153	0.5506	0.5784	**0.5832**	0.5741
d=7,d_0={6+1,6+1,6+1}	Knn5	0.5828	0.5741	0.5862	0.5123	0.5817	0.3515	0.5726	0.5707	0.5737	0.5523	0.5632	0.5653	0.6021	0.6045	**0.6072**
	SVM	0.4670	0.4682	0.4792	0.4307	0.4764	0.3774	0.4852	0.4838	0.4507	0.4553	0.4340	0.4217	0.5029	**0.5053**	0.4825
Average	Knn1	0.811	0.790	0.835	0.690	0.819	0.750	0.842	0.858	0.868	0.780	0.773	0.840	0.874	0.875	0.880
	Knn3	0.804	0.770	0.824	0.665	0.807	0.748	0.845	0.857	0.856	0.775	0.772	0.844	0.874	0.877	0.876
	Knn5	0.797	0.762	0.815	0.652	0.807	0.750	0.850	0.857	0.862	0.770	0.772	0.846	0.879	0.878	0.881
	SVM	0.667	0.667	0.724	0.580	0.696	0.665	0.758	0.751	0.767	0.686	0.690	0.751	0.784	0.785	0.786

S_1 - *SMPeMR*$_1$, S_2 - *SMPeMR*$_2$, S_3 - *SMPeMR*$_3$.

TABLE 6
Summary of Friedman Statistics F_F and the Critical Value in Terms of Classification Accuracy

Accuracy	F_F	critical value ($\alpha = 0.1$)
Knn1	11.57	1.53
Knn3	12.67	
Knn5	13.46	
SVM	11.92	

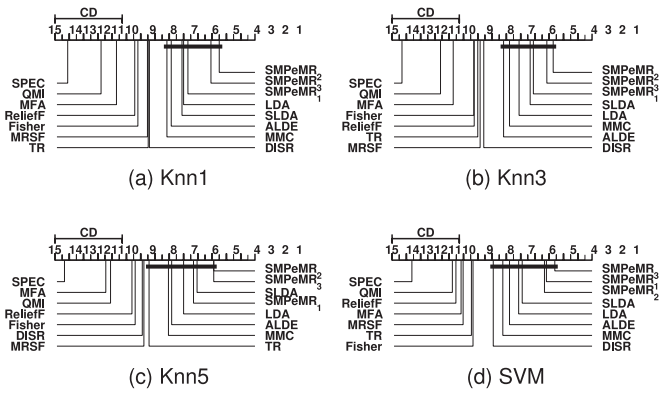

Fig. 5. Accuracy Comparison of Supervised DR algorithms against each other with the Nemenyi test.

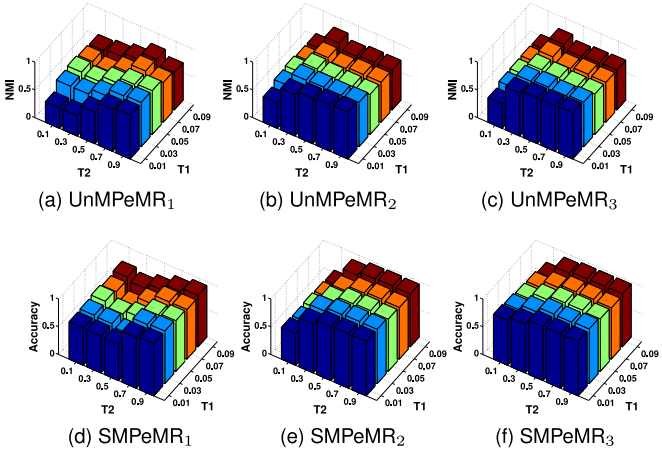

Fig. 6. Effect of T_1, T_2 on PageBlocks data set.

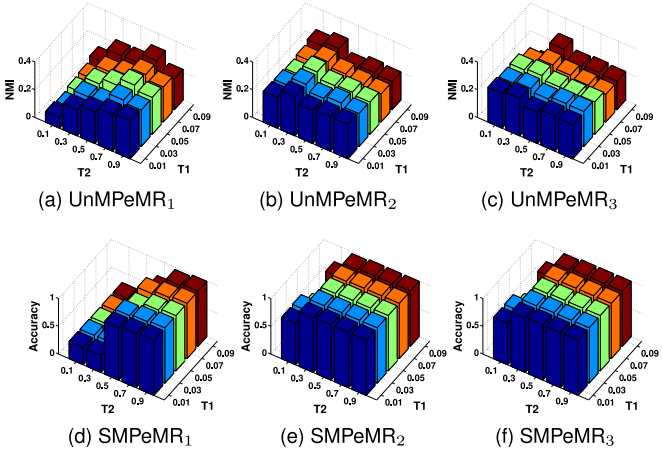

Fig. 7. Effect of T_1, T_2 on Gas Sensor data set.

4 Conclusion

A new strategy for Dimensionality Reduction with the aim of providing reduced set with both original and combinations of features is studied. For this purpose, a new framework MPeMR to generate orthogonal compound features by minimizing both projection error and redundancy among them is proposed. An iterative approximation method under the proposed framework for compound feature generation, with out losing orthogonality property, is also presented. Using different measures for projection error, redundancy and *R-feature* selection, three supervised and three unsupervised methods are considered. As the proposed approach provides both original and combinations of features in the reduced set, experimental results are compared with both state-of-the-art feature selection and extraction methods, in both supervised and unsupervised cases. From the average results over multiple data sets, it is observed that the proposed method is always better when compared to feature selection methods and with feature extraction methods, it has provided better or comparable results.

In each stage of the proposed approximation method, pairs of features are considered i.e., *NPe* is calculated based on *two* features and removal of redundancy is also based on *two* features. Instead of $(2, 2)$, one could have considered $(2, 3)$, or $(3, 2)$ or any other such combination of features. Further research is focused along this direction.

Acknowledgments

Sreevani is the corresponding author.

References

[1] H. Liu and H. Motoda, *Computational Methods of Feature Selection*. Boca Raton, FL, USA: CRC Press, 2007.
[2] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*. Hoboken, NJ, USA: Wiley, 1999.
[3] K. Bunte, M. Biehl, and B. Hammer, "A general framework for dimensionality-reducing data visualization mapping," *Neural Comput.*, vol. 24, no. 3, pp. 771–804, 2012.
[4] L. Zhang, L. Zhang, X. Mou, and D. Zhang, "FSIM: A feature similarity index for image quality assessment," *IEEE Trans. Image Process.*, vol. 20, no. 8, pp. 2378–2386, 2011.
[5] Y. Saeys, I. Inza, and P. Larrañaga, "A review of feature selection techniques in bioinformatics," *Bioinf.*, vol. 23, no. 19, pp. 2507–2517, 2007.
[6] T. Joachims, *Learning to Classify Text Using Support Vector Machines: Methods, Theory and Algorithms*. Norwell, MA, USA: Kluwer, 2002.
[7] H. Liu and H. Motoda, *Feature Selection for Knowledge Discovery and Data Mining*, vol. 454. Berlin, Germany: Springer, 2012.
[8] J. P. Cunningham and Z. Ghahramani, "Linear dimensionality reduction: Survey, insights, and generalizations," *J. Mach. Learn. Res.*, vol. 16, pp. 2859–2900, 2015.
[9] M. Robnik-Šikonja and I. Kononenko, "Theoretical and empirical analysis of relieff and rrelieff," *Mach. Learn.*, vol. 53, no. 1/2, pp. 23–69, 2003.
[10] F. Nie, S. Xiang, Y. Jia, C. Zhang, and S. Yan, "Trace ratio criterion for feature selection," in *Proc. 23rd Nat. Conf. Artif. Intell.*, 2008, vol. 2, pp. 671–676.
[11] Z. Zhao, L. Wang, and H. Liu, "Efficient spectral feature selection with minimum redundancy," in *Proc. 24th AAAI Conf. Artif. Intell.*, 2010, pp. 673–678.
[12] Z. Zhao, L. Wang, H. Liu, and J. Ye, "On similarity preserving feature selection," *IEEE Trans Knowl. Data Eng.*, vol. 25, no. 3, pp. 619–632, Mar. 2013.
[13] D. Wang, F. Nie, and H. Huang, "Global redundancy minimization for feature ranking," *IEEE Trans. Knowl. Data Eng.*, vol. 27, no. 10, pp. 2743–2755, 2015.

[14] W. Sheng, X. Liu, and M. Fairhurst, "A niching memetic algorithm for simultaneous clustering and feature selection," *IEEE Trans. Knowl. Data Eng.*, vol. 20, no. 7, pp. 868–879, Jul. 2008.
[15] X. He, D. Cai, and P. Niyogi, "Laplacian score for feature selection," in *Proc. Advances Neural Inf. Process. Syst.*, 2005, pp. 507–514.
[16] D. Cai, C. Zhang, and X. He, "Unsupervised feature selection for multi-cluster data," in *Proc. 16th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2010, pp. 333–342.
[17] M. Qian and C. Zhai, "Robust unsupervised feature selection," in *Proc. 23rd Int. Joint Conf. Artif. Intell.*, 2013, pp. 1621–1627.
[18] Z. Xu, I. King, M. R.-T. Lyu, and R. Jin, "Discriminative semi-supervised feature selection via manifold regularization," *IEEE Trans. Neural Netw.*, vol. 21, no. 7, pp. 1033–1047, Jul. 2010.
[19] K. Benabdeslem and M. Hindawi, "Efficient semi-supervised feature selection: Constraint, relevance, and redundancy," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 5, pp. 1131–1143, May 2014.
[20] P. Mitra, C. Murthy, and S. K. Pal, "Unsupervised feature selection using feature similarity," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 3, pp. 301–312, Mar. 2002.
[21] M. Dash, K. Choi, P. Scheuermann, and H. Liu, "Feature selection for clustering-a filter solution," in *Proc. 2nd Int. Conf. Data Mining*, 2002, pp. 115–122.
[22] Q. Song, J. Ni, and G. Wang, "A fast clustering-based feature subset selection algorithm for high-dimensional data," *IEEE Trans. Knowl. Data Eng.*, vol. 25, no. 1, pp. 1–14, Jan. 2013.
[23] R. Kohavi and G. H. John, "Wrappers for feature subset selection," *Artif. Intell.*, vol. 97, no. 1, pp. 273–324, 1997.
[24] H. Liu and L. Yu, "Toward integrating feature selection algorithms for classification and clustering," *IEEE Trans. Knowl. Data Eng.*, vol. 17, no. 4, pp. 491–502, Apr. 2005.
[25] B. Efron, et al., "Least angle regression," *Ann. Statist.*, vol. 32, no. 2, pp. 407–499, 2004.
[26] J. Liu, S. Ji, and J. Ye, "Multi-task feature learning via efficient l 2, 1-norm minimization," in *Proc. 25th Conf. Uncertainty Artif. Intell.*, 2009, pp. 339–348.
[27] L. Yu and H. Liu, "Feature selection for high-dimensional data: A fast correlation-based filter solution," in *Proc. 20th Int. Conf. Mach. Learn.*, 2003, vol. 3, pp. 856–863.
[28] K. Fukunaga, *Introduction to Statistical Pattern Recognition*. New York, NY, USA: Academic, 2013.
[29] P. A. Devijver and J. Kittler, *Pattern Recognition: A Statistical Approach*. London, U.K.: Prentice-Hall International, 1982.
[30] H. H. Harman, *Modern Factor Analysis*. Chicago, IL, USA: Univ. Chicago Press, 1976.
[31] P. J. Huber, "Projection pursuit," *Ann. Statist.*, vol. 13, pp. 435–475, 1985.
[32] A. Hyvärinen and E. Oja, "Independent component analysis: algorithms and applications," *Neural Netw.*, vol. 13, no. 4, pp. 411–430, 2000.
[33] H. Li, T. Jiang, and K. Zhang, "Efficient and robust feature extraction by maximum margin criterion," *IEEE Trans. Neural Netw.*, vol. 17, no. 1, pp. 157–165, Jan. 2006.
[34] S. Liu, L. Feng, and H. Qiao, "Scatter balance: An angle-based supervised dimensionality reduction," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 2, pp. 277–289, Feb. 2015.
[35] H. Zou, T. Hastie, and R. Tibshirani, "Sparse principal component analysis," *J. Comput. Graphical Statist.*, vol. 15, no. 2, pp. 265–286, 2006.
[36] L. Clemmensen, T. Hastie, D. Witten, and B. Ersbøll, "Sparse discriminant analysis," *Technometrics*, vol. 53, pp. 406–413, 2012.
[37] M. Masaeli, J. G. Dy, and G. M. Fung, "From transformation-based dimensionality reduction to feature selection," in *Proc. 27th Int. Conf. Mach. Learn.*, 2010, pp. 751–758.
[38] Q. Gu, Z. Li, and J. Han, "Joint feature selection and subspace learning," in *Proc. Int. Joint Conf. Artif. Intell.*, 2011, vol. 22, no. 1, Art. no. 1294.
[39] Sreevani, "Simultaneous feature selection and feature extraction for pattern classification," Master's thesis, Indian Statistical Inst., Kolkata, India, Jul. 2009. [Online]. Available: http://www.isical.ac.in/sreevani_r
[40] G. Qu, S. Hariri, and M. Yousif, "A new dependency and correlation analysis for features," *IEEE Trans. Knowl. Data Eng.*, vol. 17, no. 9, pp. 1199–1207, Sep. 2005.
[41] Z. Zhao and H. Liu, "Spectral feature selection for supervised and unsupervised learning," in *Proc. 24th Int. Conf. Mach. Learn.*, 2007, pp. 1151–1157.
[42] P. E. Meyer and G. Bontempi, "On the use of variable complementarity for feature selection in cancer classification," in *Applications of Evolutionary Computing*. Berlin, Germany: Springer, 2006, pp. 91–102.
[43] S. Yan, D. Xu, B. Zhang, and H.-J. Zhang, "Graph embedding: A general framework for dimensionality reduction," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 2005, vol. 2, pp. 830–837.
[44] D. Bouzas, N. Arvanitopoulos, and A. Tefas, "Graph embedded nonparametric mutual information for supervised dimensionality reduction," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 5, pp. 951–963, May 2015.
[45] B. J. Frey and D. Dueck, "Clustering by passing messages between data points," *Science*, vol. 315, no. 5814, pp. 972–976, 2007.
[46] M. Halkidi, Y. Batistakis, and M. Vazirgiannis, "On clustering validation techniques," *J. Intell. Inf. Syst.*, vol. 17, no. 2, pp. 107–145, 2001.
[47] N. X. Vinh, J. Epps, and J. Bailey, "Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance," *J. Mach. Learn. Res.*, vol. 11, pp. 2837–2854, 2010.
[48] M. Friedman, "A comparison of alternative tests of significance for the problem of m rankings," *Ann. Math. Statist.*, vol. 11, no. 1, pp. 86–92, 1940.
[49] P. B. Nemenyi, "Distribution-free multiple comparison," Ph.D. dissertation, Princeton Univ., Princeton, NJ, USA, 1963.
[50] J. Ye and T. Xiong, "Computational and theoretical analysis of null space and orthogonal linear discriminant analysis," *J. Mach. Learn. Res.*, vol. 7, pp. 1183–1204, Jul. 2006.
[51] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *J. Mach. Learn. Res.*, vol. 7, pp. 1–30, 2006.

Sreevani received the MSc degree in mathematics from the University of Hyderabad, India, during 2004-2006 and the MTech degree in computer science from the Indian Statistical Institute (ISI), India, in 2007-2009. Currently, she is working toward the PhD degree in the Machine Intelligence Unit, ISI. She has worked as a 'fellow scientist' at the National Institute of Science Technology and Development Studies, New Delhi, India, during 2009-2011. She was a research fellow with the Center for Soft Computing Research, ISI, Kolkata, during 2011-2013. Her research, mainly focuses on dimensionality reduction (feature selection and extraction), density estimation, and bandwidth selection for kernel density estimation. She is a student member of the IEEE.

C. A. Murthy received the BStat (Hons), MStat, and PhD degrees from the Indian Statistical Institute (ISI), India. He visited Michigan State University, East Lansing, in 1991-1992 for six months, and the Pennsylvania State University, University Park, for 18 months in 1996-1997. He is a professor in the Machine Intelligence Unit, ISI. His fields of research interest include pattern recognition, image processing, machine learning, neural networks, fractals, genetic algorithms, wavelets, and data mining. He received the best paper award in 1996 in Computer Science from the Institute of Engineers, India. He received the Vasvik award along with his two colleagues for Electronic Sciences and Technology for the year 1999. He is a fellow of the National Academy of Engineering, India, and National Academy of Sciences, India. He was the head of the Machine Intelligence Unit, ISI, from 2005 to 2010.