A Novel Method of Constrained Feature Selection by the Measurement of Pairwise Constraints Uncertainty

Kamal Berahmand[1], Mehrdad Rostami[2], Saman Forouzandeh[3]
Department of Information Technology and Communications, Azarbaijan Shahid Madani University, Tabriz, Iran[1]
Department of Computer Engineering, University of Kurdistan, Sanandaj, Iran[2]
Department of Computer Engineering University of Applied Science and Technology, Center of Tehran Municipality ICT org.Tehran, Iran[3]
berahmand@azaruniv.ac.ir[1], m.rostami@eng.uok.ac.ir[2], Saman.forouzandeh@gmail.com[3]

Abstract:
In recent years, with the development of science and technology, there were considerable advances in datasets in various sciences, and many features are also shown for these datasets nowadays. With a high-dimensional dataset, many features are generally redundant and/or irrelevant for a provided learning task, which has adverse effects with regard to computational cost and/or performance. The goal of feature selection over partially labeled data (semi-supervised feature selection) is to choose a subset of available features with the lowest redundancy with each other and the highest relevancy to the target class, which is the same objective as the feature selection over entirely labeled data. By appropriate reduction of the dimensions, in addition to time-cost savings, performance increases as well. In this paper, side information such as pairwise constraint is used to rank and reduce the dimensions. In the proposed method, the authors deal with checking the quality (strength or uncertainty) of the pairwise constraint. Usually, the quality of the pair of constraints on the dimension reduction is not calculated. In the first step, the strength matrix is created through a similarity matrix and uncertainty region. And then, by using the strength and similarity matrices, a new constraint feature selection ranking is proposed. The performance of the presented method was compared to the performance of the state-of-the-art, and well-known semi-supervised feature selection approaches on eight datasets. The findings indicate that the proposed approach improves previous related approaches with respect to the accuracy of constrained clustering. In particular, the numerical results showed that the presented approach improved the classification accuracy by about 3% and reduced the number of selected features by 1%. Consequently, it can be said that the proposed method has reduced the computational complexity of the machine learning algorithm despite increasing the classification accuracy.

Keyword: Pairwise constraint, semi-supervised, machine learning, feature selection, uncertainty

1. Introduction:
Along with the growth of data such as image data, meteorological data, particularly documents, dimensions of these data also increase. According to the studied extensively, the accuracy of current clustering methods generally decreases with high dimensional data that event referred to as the curse of dimensionality. For preventing the curse of dimensionality, some dimension (feature) reduction techniques are used [1-3]. Traditional techniques to reduce the dimensions are divided into two main categories: feature extraction and feature selection[4]. In the first approach, instead of the original features, secondary features with low dimensions are extracted. That means that a high dimensional space is transferred to low dimensional space. However, the second approach includes four sub-categories that include filter method, wrapper method, hybrid methods, and embedded methods [5, 6]. The subset of features in the pre-processing step is selected in filter methods independent of any learner method [7]. In contrast, Wrapper methods apply a learner method to investigate the subsets of features based on their predictive power. Dealing with extensive data and side information, each of these methods has advantages and disadvantages regarding the time being used, consistency with data, efficiency, and accuracy. The feature selection approaches are divided into two main groups: supervised and unsupervised [3]. In the supervised method, the label of dataset exists, based on which the evaluation and selection of suitable features are made. That is, while in unsupervised type, the classes of the label are not available, and evaluating and selecting are done based on the ability to meet some of the

properties of the data set, including the locality preserving ability and/or variance. Since in most datasets, label or side information is available in small quantities, and obtaining these labels is costly, semi-supervised or constrained methods are used. The semi-supervised feature selection method uses data with labels and unlabeled; in contrast, the other choice of semi-supervised method is the pairwise constraint. In this method, not all data sets have labels, but there is side information like a pairwise constraint [8, 9].

A pairwise constraint is a pair of data belonging to the different clusters (cannot-link) or the same cluster (must-link) [10]. In fact, in the real world, in case of lack of label, the best possible information to select the feature is pairwise constraints. Overall, obtaining label is too costly, and in many cases, these constraints inherently exist. In the case of the existence of labels, one can turn this type of data set into pairwise constraint (by transitive closure and vice versa), which is one of the advantages of working on the pairwise constraint [11]. Because of the importance of pairwise constraint and inherent and low-cost nature of this pairwise constraint, many studies have been conducted such as the development of constrained clustering algorithms to consider the pairwise constraint in the process of clustering, active learning algorithms to obtain the best and most valuable pair to increase the accuracy, the transformation of the objective functions in clustering, and the like. One of the studies that have rarely been done in the field of feature selection on the basis of the pairwise constraint. The purpose of this method is to reduce the dimension size by considering the pairwise constraint so that the constraint clustering algorithm has the best results, accuracy, and efficiency. Most of the methods available in this field are improvements to previous similar methods (usually unsupervised feature selection).

In the present paper, a novel pairwise constraints-based method is proposed for feature selection and reduce dimensions. Our method is complementary to previous methods. In this study, in addition to the constraints, the quality of the constraints is also used. The quality of the pair of constraints is the power of the relationship between two pairs of data or vice versa (uncertainty). In the proposed method, in the first, the similarity between the pair constraints is calculated. Then an uncertainty region is created based on it. The uncertainty region and its coefficient are used to indicate the power and quality of the pair of constraints. These coefficients are then ensemble with a previous basic method, then in an iterative process are selected most informative pairs. There was a considerable improvement by comparing the proposed method with the previous methods. It might be argued that the proposed method has reduced the computational complexity of the machine learning algorithm despite increasing the classification accuracy.

The structure of the present study is as follows: Section 2 reviews the related work, Section 3 introduces the proposed method, Sections 4 presents the experiment, and Section 5 provides the conclusion and future work.

2. Related work:

The dimensionality reduction techniques are mostly divided into two categories: feature extraction and feature selection [12-14]. In the feature extraction methods, the data is transformed from the original space into a new space with fewer dimensions. On the contrary, the size of the dataset is directly reduced by the feature selection methods by picking a subset of relevant and non-redundant features and retaining adequate information for the learning task. The objective of the feature selection methods is seeking the related features with the most predictive information from the original feature set. The feature selection was determined to be an essential technique in many practical applications, including text processing [15-17], face recognition [18-20], image retrieval [21, 22], medical diagnosis [23], case-based reasoning [24] and bioinformatics [25]. One of the basic research subjects in pattern recognition is feature selection, with a long history started in the 1970s. Also, many attempts have been made to review the feature selection approaches [26-28]. In this section, various feature selection methods are summarily reviewed by the authors. These methods might be divided into four categories: filter, wrapper, embedded, and hybrid approaches.

In the filter-based methods, every single feature is ranked with no consideration of learning algorithms on the basis of its discriminating power among various classes. The statistical analysis of the feature set is required in the filter approach to select the final feature set. On the contrary, a learning algorithm is applied in the wrapper-based feature selection methods to assess the quality of feature subsets in the search space iteratively. The wrapper approach needs a high computational cost for high-dimensional datasets since every single subset is investigated by a specified learning model. In the embedded model, it is considered that the model building process includes the feature

selection procedure as a part of it, in which both redundant and irrelevant features can be handled; as a result, training learning algorithms with a considerable number of features will take a great deal of time. On the other hand, the purpose of the hybrid-based approaches is employing the proper performance of the wrapper model and the computational efficiency of the filter model. However, the accuracy issue may be challenging in the hybrid model since the filter and wrapper models are taken into account as two separate steps.

Following the availability of the class labels of training data, the feature selection methods can be roughly divided into three categories: supervised feature selection, unsupervised feature selection, and semi-supervised feature selection [23, 26, 29]. In the supervised approaches, training samples are characterized by the vector of feature values with class labels, which are applied to direct the search process to associated information; however, in the unsupervised feature selection, the feature vectors value are described without class labels. Since the labeled information is used, the supervised feature selection methods often show better performance compared to unsupervised and semi-supervised techniques. In a large number of real-world applications, collecting the labeled patterns will be hard, and there are abundant unlabeled data and small labeled patterns. In order to handle this 'incomplete supervision,' semi-supervised (pairwise constraint) feature selection methods were developed, which use both unlabeled and labeled data for the machine learning task. In the semi-supervised feature selection methods, the local structure of both labeled and unlabeled data or the label information of labeled data and data distribution is used for the purpose of selecting final related and non-redundant features. Consequently, the interesting topic of feature selection for semi-supervised feature selection is a more complex problem, and researching this area is recently attracting more interest in many communities.

Term Variance (TV) [30], Laplacian Score for feature selection (LS) [31], Relevance-Redundancy Feature Selection (RRFS) [32], Unsupervised Feature Selection based on Ant Colony Optimization (UFSACO) [33] are some existing filter-based unsupervised feature selection methods. Furthermore, a clustering algorithm is used in the unsupervised wrapper feature selection methods to investigate the quality of picked features. On the one hand, the higher computational complexity in learning is considered as the major disadvantage of these approaches, which is because of the application of specified learning algorithms. Also, the inefficiency of them on the datasets with many features has been shown. On the contrary, the statistical analysis of the feature set is required by the unsupervised filter method only for solving the feature selection task without employing any learning models. A feature selection method may be investigated in accordance with effectiveness and efficiency. Although the time needed to discover a subset of features is important for the efficiency, the effectiveness is associated with the quality of the subset of features. These issues are in disagreement with each other: in general, one is reduced by improving the other. Alternatively stated, the computational time is advantageous in the filter-based feature selection methods, and they are typically faster, although the quality of selected features is considered in the unsupervised wrapper methods.

Recently, the graph-based methods, including spectral embedding [34], spectral clustering [35], and semi-supervised learning [36], have contributed significantly to feature selection because of their capability of encoding similarity relationships among the features. Recently, many graph-based unsupervised and semi-supervised feature selection methods are presented to extract the relationships among the features. For example, a spectral semi-supervised feature selection criterion called the s-Laplacian score was presented by Cheng et al. [37]. According to this criterion, a Graph-based Semi-supervised Feature Selection method called GSFS was proposed. In this method, in order to select relevant features as well as to remove redundant features, the conditional mutual information and spectral graph theory are employed. Moreover, in [38], the authors designed a graph-theoretic method for non-redundant unsupervised feature selection. In this method, the feature selection tasks as the densest subgraph finding from a weighted graph. In [39], a dense subgraph finding method is selected for the unsupervised feature selection problem. In this paper, a novel normalized mutual information is used for the purpose of calculating the similarity among two features.

3. The proposed method

The detail of the proposed method will be explained in this section. First, the general concepts related to the proposed method will be expressed, and then the details of the proposed semi-supervised feature selection method are introduced.

3.1: Background and Notation

Weights of the terms obtained:

When document sets are tokenized, the document-term matrix becomes very sparse. It is also time-consuming to read texts for labeling. Since we use document sets, for instance, and they are embedded in traditional methods. The document set is expressed as where x shows the document, and then, using traditional pre-processing and tfidf method, the terms of the documents, becomes final weights with different values [1-2]. In this set, w represents the weights of the terms obtained by Eq. (1).

$$w_{ib} = tf_{ib} \times idf_{ib} = tf_{ib} \times \log_2 {}^{n}/_{df_{ib}} \quad (1)$$

In this equation, tf is the frequencies of term b in document i, df is the frequency of the documents where this term exists, and n implies to the number of documents. Finally, the term-document matrix is extracted for use in the proposed algorithm. It is noteworthy that for all distances expressed in this article, the Minkowski distance method is used. This method is the most famous and adopted methods in clustering.

Neighborhoods and pairwise constraint:

In general, if $\{x_i, x_j, x_k\}$ is the three data of the data set, then each pair's relationship is expressed as $\{\mathrm{ML}, \mathrm{CL}\}$, and the clustering label is expressed with *lab*, then relations and Eq. (2) must be established. By closure of pairwise constraints, neighborhoods can be formed.

$$(x_i, x_j, Ml) \wedge (x_i, x_k, Ml) \Rightarrow (x_j, x_k, Ml) \quad (2)$$

$$(x_i, x_j, Cl) \wedge (x_i, x_k, Cl) \Rightarrow (x_j, x_k, Cl)$$

$$(x_i, x_j, Ml) \Leftrightarrow lab_i = lab_j \text{ , in same cluster}$$

$$(x_i, x_j, Cl) \Leftrightarrow lab_i \neq lab_j \text{ , not in same cluster}$$

Neighborhoods are a set of a neighborhood whose number is usually smaller or equal to the number of clusters defined in the algorithm. Each neighborhood includes several sample data that must be in the same cluster together. The basic premise in that neighborhood is that different data in different clusters should be placed in different neighborhoods, and no two Neighborhoods should be found where data exists as the same cluster.

Measuring the uncertainty of constraints:

In the real world, constraints arise from domain knowledge or expert knowledge. Pairwise constraints have weak relationships, and strangeness (uncertainty) of the relations is variable. Hence, it is needed to create an uncertainty region. By finding the region, it is easy to have an impact on our ranking and see better results in reduced dimensions. In order to do this, the authors use the thresholding histogram method. This method actually used the classifying method with two classes, and its purpose is to reduce ambiguity in the range of values. First, the similarity values of each pair S_{en} matrix are collected, and then these values are divided into intervals, and the average of each interval is determined as (D_i). In the next step, for each interval, the number of pairs in this range is counted as the g (D_i). So, from these values, a weighted moving average with five windows, f (D_i), is calculated by Eq (3). The authors start from the beginning of the intervals and find the first valley points in the modified histogram$f(D_v)$. Finally, the uncertainty region is calculated.

Step 1: $f(D_i) = \frac{g(D_i)}{\sum_{e=1}^{z-1} g(D_e)} \times \frac{g(D_{i-2})+g(D_{i-1})+g(D_i)+g(D_{i+1})+g(D_{i+2})}{5}, \forall i = 2,3,\ldots.,z-3$ (3)

Step 2: find the first valley points subject to:

$$f(D_{v-1}) > f(D_v) \; and \; f(D_v) < f(D_{v+1}) \quad (4)$$

Step 3: find the boundary of the uncertainty region:

$$m_d = D_v \;\; and \;\; m_c = \max(D_i) - m_d \quad (5)$$

Step 4: find the pairs in similarity matrix that have uncertainty relationship:

$$in\ similarity\ matrix\ \ S_{en_{ij}}: \begin{cases} m_d \le if\ S_{en_{ij}} \le m_c & : uncertainty\ \ region \\ else & : \ \ strong\ region \end{cases} \quad \forall i, j \quad (6)$$

3.2. The pairwise constraint feature selection:

In this section, a novel Pairwise Constraint Feature Selection method (PCFS) is proposed. This method uses pc-k-mean with small and effective changes. The proposed method has been able to use both standard objective function and a penalty for the violation of constraints, with changing the objective function. These two sections together constitute the objective function and are locally minimized. The proposed method, named Dim_raduce() function, is affected by the current clustering and vice versa. The PCFS method is

Algorithm 1 *PCFS Clustering*

Input: *document-term matrix* $\{\boldsymbol{M_{ib}}\}_{\boldsymbol{i=1\ldots.n}}^{\boldsymbol{b=1\ldots.m}}$ *,Data set* $\boldsymbol{\mathcal{D} = \{x_1, x_2, x_3, \ldots., x_n\}}$ *, Number of clusters* $\boldsymbol{K}$*, must_link set* $\boldsymbol{ML}$*, cannot_link set* $\boldsymbol{CL}$ *, Number of must_link* $\boldsymbol{ml}$*, Number of cannot_link* $\boldsymbol{cl}$*,* $\boldsymbol{Wm}$ *and* $\boldsymbol{Wc}$ *penalty of violthe ation, Neighborhoods* $\{\boldsymbol{N_h}\}_{\boldsymbol{h=1}}^{\boldsymbol{l}}$ *, document-reduced-term matrix* $\{\boldsymbol{RM_{ib}}\}_{\boldsymbol{i=1\ldots.n}}^{\boldsymbol{b=1\ldots.r}}$ *, the number of features after reducing* $\boldsymbol{r}$*.*

Output: *assignment clusters* $\{\boldsymbol{Xp}\}_{\boldsymbol{p=1}}^{\boldsymbol{K}}$

1. *Repeat loop until clusters not changed or (with a predefined number of the loop)*
2. *Dim_reduce()*
3. *Find* $\{\boldsymbol{N_h}\}_{\boldsymbol{h=1}}^{\boldsymbol{l}}$ *from the closure of ML and CL*
4. *Find the centroid of Neighborhoods* $\{\boldsymbol{N_h}\}_{\boldsymbol{h=1}}^{\boldsymbol{l}}$
5. *If l<k then find k-l random data point, not in Neighborhoods and assign them.*
6. *Initial* $\left\{\boldsymbol{\mu}_{\boldsymbol{p}}^{(0)}\right\}_{\boldsymbol{p=1}}^{\boldsymbol{K}}$ *with centroid of* $\{\boldsymbol{N_h}\}_{\boldsymbol{h=1}}^{\boldsymbol{l}}$ *and randomly chosen data*
7. *Repeat until convergence*
8. *A : assign_cluster :*
   Assign each data point $\boldsymbol{x_i}$ *with reduced features to the cluster* p^* *which :* $\boldsymbol{p^* = argmin(\frac{1}{2}\left\|x_i - \mu_p^{(t)}\right\|^2 + Wm\sum_{(x_i,x_j)\in Ml}\mathbb{I}[p \neq lab_j] + Wc\sum_{(x_i,x_j)\in Cl}\mathbb{I}[p = lab_j])}$
9. *B : estimate_means:*
   $\left\{\boldsymbol{\mu}_{\boldsymbol{p}}^{(\boldsymbol{t+1})}\right\}_{\boldsymbol{p=1}}^{\boldsymbol{K}} \leftarrow \boldsymbol{\alpha}\left\{\frac{1}{\left|X_p^{(t)}\right|}\sum_{x\in X_p^t} X\right\}_{\boldsymbol{p=1}}^{\boldsymbol{K}} + (\boldsymbol{1-\alpha})\{\boldsymbol{centeroid\ of\ Neighborhoods}\ \}_{\boldsymbol{h=1}}^{\boldsymbol{l}}$
10. *C :* $\boldsymbol{t \leftarrow t+1}$
11. *End Repeat*
12. *End loop*

Briefly, the data set are embedded as a document-term matrix, and then other variables values are initialized. The whole of the procedure is repeated in a loop until the clusters not changed (or with the predefined number of the loop). In each iteration, given the current clustering and set of constraints ML and CL, Dim_reduce() performs to produce a reduced feature. (line 2). After this, neighborhoods are formed from the closure of pairwise constraints,

and then the center of pairwise constraints of each neighborhood is calculated. If a neighborhood does not have any data, randomly a data, it should not be a member of other neighborhoods, is as the center of that cluster. Finally, centers of clusters are initialized by the center of neighborhoods. (Line 3-6). For `assigning clusters and estimating` (updating) center of clusters, section A and B is performed (8-9). These two sections are repeated until convergence, as pckmeans. After convergence, the procedure is repeated until meet stop conditions. Dim_reduce() function is the core of PCFS that is summarized in Algorithm 2. In this method, in addition to the usual input in feature selection, pairwise constraints arise as input.

Algorithm 2 *Dim_reduce()*

Input: $\mathcal{D} = \{x_1, x_2, x_3, \dots, x_n\}$, *document-term matrix* $\{M_{ib}\}_{i=1\dots n}^{b=1\dots m}$, *document-reduced-term matrix* $\{RM_{ib}\}_{i=1\dots n}^{b=1\dots r}$, *number of features after reducing* **r**, *the temporary vector of reduced features* $\{F_r\}$

Output: $\{RM_{ib}\}_{i=1\dots n}^{b=1\dots r}$

1. *Initializations:* $\{RM_{ib}\}_{i=1\dots n}^{b=1\dots r} = \emptyset$,
2. *Repeat until items of* $\{F_r\}$ *is changed*
3. *S_en_func ()*
1. *S_tr_func()*
4. *Calculate score* C_b *for each feature use Eq.8*
5. *Sort features Ascending on based* C_b
6. *Select first* **r** *features and set to* $\{F_r\}$
7. *End repeat*
8. *Set final* $\{F_r\}$ *to* $\{RM_{ib}\}_{i=1\dots n}^{b=1\dots r}$

There are two main functions in this algorithm that respectively, S_en_func() in algorithm 3 and S_tr_func() in algorithm 4 are expressed. The first function extracts the matrix of similarities between data pairs, and then in the second function, the uncertainty region and strength of the relationship is calculated for each pair. After calculating the two functions within an iterative process, the authors rank the features by Eq. (7). Finally, Repeat will continue until the selected features are changed.

$$C_b = \frac{\sum_{(x_i,x_j)\in ML}(f_{bi}-f_{bj})^2 \times S_{tr_{ij}} + \frac{(1-S_{en_{ij}})}{\sum_{(x_k,x_z)\in ML}(1-S_{en_{kz}})} \times (1-S_{tr_{ij}})}{\sum_{(x_i,x_j)\in CL}(f_{bi}-f_{bj})^2 \times S_{tr_{ij}} + \frac{(1-S_{en_{ij}})}{\sum_{(x_k,x_z)\in CL}(1-S_{en_{kz}})} \times (1-S_{tr_{ij}})} \qquad (7)$$

Hence, $S_{tr_{ij}}$ indicates the quality (power) of the relationship between each data pairs, and each element in the matrix are calculated through the uncertainty region. For the ranking of features, this formula assumes that if the power of pairs (in the set of pairwise constraints) is low, the authors mostly use similarity matrix; otherwise, (in case of reliability and high strength of the relationship of pairwise), Minkowski distance is used. In fact, using this method, strength and quality are added to the formula, and thereby better results can be obtained. The summarization of calculating the similarity matrix is possible in algorithm 3. First, the authors assigned clusters as labels of data set (lines 3-6). Then the classification model is performed on the dataset with produced labels from clustering (line 8). In the iterative process, a similarity matrix based on anticipated labels (from the classification model) is created. During different iterations, this similarity matrix is updated and normalized.

Algorithm 3 *S_en_func*

Input: $\mathcal{D} = \{x_1, x_2, x_3, \ldots, x_n\}$, *number of iteration* ***en****, similarity matrix* $S_{en}[]_{i=1\ldots n}^{i=1\ldots n}$

Output: *S_en*

1. *Initializations:* $S_{en}[]_{i=1\ldots n}^{i=1\ldots n} = \emptyset$
2. *For itr=0 ……en*
3. *If (first loop of Alghoritm1)*
4. $Y_{i=1\ldots n}$ *= run K-means clustering on D and assign clusters as the label*
5. *else*
6. $Y_{i=1\ldots n}$ *=assign current clusters as label*
7. *End if*
8. *M= Take a sample from D*
9. *model = Run classification model on* ***MY***
10. *For each data pair*
11. *If the anticipated label of two-point is the same*
12. $Temp\_S\_en[]_i^j$*=1*
13. *Else*
14. $Temp\_S\_en[]_i^j$*=0*
15. *Endif*
16. *End for*
17. S_{en}*=(Temp_s_en +* S_{en} *)/2*
18. *Endfor*

Finally, the Matrix calculation of strength S_{tr} and the uncertainty region as algorithm 4 is summarized. After finding the uncertainty region (line3), it is time to calculate S_{tr} matrix. For data pairs that are in the uncertainty region, the relative strength of them is equal to β, and outside of this range, it is 1-β. This β parameter was chosen after several preliminary runs, and this the value of β is empirically considered as 0.3.

Algorithm 4 *S_tr_func*

Input: $\mathcal{D} = \{x_1. x_2. x_3. \ldots. x_n\}$, *first valley point of the modified histogram, similarity matrix* S_{en} *, Matrix of strength* S_{tr} *,*

Output: S_{tr}*,md,mc*

1. *Initializations:*
2. $S_{tr}[]_{j=1\ldots n}^{i=1\ldots n} = \emptyset$,
3. *Find boundary (*m_d, m_c*)of uncertainty region by Eq (5)*
4. *For each pair in* $S_{en_{ij}}$
5. *If the pair value is between* m_d *and* m_c
6. $S_{tr}[]_i^j = \beta$
7. *Else*
8. $S_{tr}[]_i^j = 1 - \beta$
9. *End if*

4. Experimental results:

For the purpose of investigating the performance of the proposed method (i.e., PCFS), several extensive experiments are performed. The obtained results are compared with six state-of-the-art and well-known methods such as FS [40], FAST [41], FJMI [42] (in supervised mode) and, LS [31], GCNC [43], and FGUFS [44] (in unsupervised mode). The description of this method is described below.

FS (Fisher Score): This method is a univariate filter method that scores features such that based on that feature, the distance between the samples from the same class is short, and the distance between the samples from different classes is long. Therefore, this criterion gives higher ratings to features that have such a separation property.

FAST (Fast clustering-based feature selection method): In this method, the graph-theoretic clustering methods are used to divide the features into clusters. Then the most representative feature that is significantly associated with target classes is picked from each cluster to develop a subset of features.
FJMI (Five-way Joint Mutual Information): In this paper, a feature selection method is proposed, in which a two-through five-way interaction between features and the class label is considered.
LS (Laplacian Score): This is a graph-based feature selection method that works in unsupervised mode. This method models the data space into a graph, and probably belong to the same class based on the idea of whether two data points are near to each other.
GCNC (Graph Clustering with the Node Centrality): GCNC is a feature selection method, in which the concept of graph clustering is integrated with the node centrality. This approach can handle both redundant and irrelevant features.
FGUFS (Factor Graph Model for Unsupervised Feature Selection): The similarities between features are explicitly measured in this method. These similarities are passed to each other as messages in the graph model. The message-passing algorithm is applied to calculate the importance score of each feature, and then the selection of features is performed on the basis of the final importance scores.
The results are reported in terms of two measures, including the classification accuracy (ACC) and the number of selected features. ACC is defined as follow:

$$Acc = \frac{TP + TN}{TP + TN + FP + FN} \tag{8}$$

Where TP, TN, FP, and FN stand for the number of true positives, true negatives, false positives, and false negatives, respectively.

4.1 Datasets:

In the present study, a large number of datasets with different properties are applied in the experiments to demonstrate the robustness and effectiveness of the proposed approach. SPECTF, SpamBase, Sonar, Arrhythmia, Madelon, Isolet, Multiple Features, and Colon has taken from the UCI repository are included in these datasets [45] and have been extensively used in the literature. Table 1 presents the basic characteristics of these datasets. The datasets have been chosen in such a way that they consider several characteristics, including the number of different classes, the number of features, and the number of samples. For instance, Colon is a significantly high dimensional dataset with small sample size; however, SpamBase is the example of a low dimensional with a large sample size dataset. Again, Isolet is a multi-class dataset that has 26 different kinds of classes. In these experiments, the generations of pairwise constraints are simulated as the following: The pairs of samples from the training data and created cannot-link or must-link constraints are randomly selected on the basis of whether the underlying classes of the two samples are similar or dissimilar.
Some of these datasets contain features that take a wide range of values. Note that features with small values will be dominated by those features with large values. The normalization of datasets is performed to tackle this issue. The primary reason for selecting this normalization method is that the information related to standard deviation can be partially preserved by the other methods; however, the topological structure of the datasets is retained by the max-min normalization in many cases. For each dataset, the results are achieved over ten independent runs to obtain relatively more stable and accurate approximations. In every single run, each dataset is firs normalized and is randomly split into a test set (1/3 of the dataset) and a training set (2/3 of the dataset). The test set is applied for evaluating the selected features, while the training set is applied to pick the final feature subset. A number of these datasets include features with missing values; thus, every single missing value was replaced with the mean of the available data on the respective feature to handle these kinds of data in the experiments.

Table 1: Characteristics of the used datasets

Dataset	Features	Classes	Patterns
SPECTF	44	2	80
SpamBase	57	2	4601
Sonar	60	2	208
Arrhythmia	279	16	351
Madelon	500	2	4400
Isolet	617	26	6238
Multiple Features	649	10	2000
Colon	2000	2	62

4.2 Classifiers used in the experiments:

In order to demonstrate the generality of the proposed method, several well-known classical classifiers such as Support Vector Machine (SVM), Decision Tree (DT), and Naïve Bayes (NB) were employed to test the classification prediction capability of the selected features. SVM is a learning machine which is generally used for the classification problem. SVM was presented by Vapnik and became very popular over the past ten years. The maximization of a margin between data samples is the purpose of SVM. NB is a family of simple probabilistic classifiers on the basis of using Bayes theorem with strong (naive) independence assumptions between the features. In simple terms, it is assumed in a Naïve Bayes classifier that in terms of the target class, the features are conditionally independent of each other. Decision Tree (DT) is considered as one of the most successful methods for the classification problem. The tree is created by training samples, and a rule is represented by each path from the root to a leaf, which gives a classification of the pattern. The normalized information gain is examined in this classifier to make decisions. Weka (Waikato Environment for knowledge analysis) is the experimental workbench [46], which is a collection of machine learning algorithms for mostly data mining tasks. In this work, SMO, AdaBoostM1, and Naïve Bayes as the WEKA implementation of SVM, NB, and AB have been applied. Moreover, the parameters of the mentioned classifiers for each experiment have been set to the default values of the WEKA software. The proposed method involves several parameters that must be set before starting the method. The appropriate values for some of these parameters are chosen as trial and error after a number of primary runs so they do not mean the best value for these parameters. Moreover, in all of these experiments, the values used in the each of the compared methods were used to adjust the parameters.

4.3. Results:

In the experiments, the number of selected features and the classification accuracy is used as the performance measures, and first, the performance of the proposed method is investigated over different classifiers. The summary of average classification accuracy (in %) over ten independent runs of the unsupervised methods (i.e., LS, GCNC, and FGUFS) and supervised methods (i.e., FS, Fast and FJMI) using SVM, NB, and DT classifier is listed in Table 2. Each entry of these tables denotes the mean value and also standard deviation (indicated in parenthesis) of 10 independent runs. The best mean values of average percentage accuracy are marked in boldface. Table 4 reveals that the proposed method performs better compared to other feature selection methods in most cases. Although in Multiple Features, DSFFC obtained the best results, the proposed method performs the second-best, giving approximately comparable results with the best one. The table results may reveal that the best results are obtained by the proposed method for the datasets with a higher number of features.

Table 2: Performance comparison of different feature selection methods on eight

datasets.

Dataset	method	Evaluation criteria		
		SVM	**NB**	**DT**
SPECTF	LS	75.87(1.78)	76.79(1.84)	76.88(1.79)
	GCNC	77.32(0.35)	78.18(1.05)	77.07(1.92)
	FGUFS	78.68(0.83)	78.19(2.15)	76.23(2.08)
	FS	77.83(2.32)	73.83(0.92)	69.25(1.88)
	FAST	78.91(1.24)	76.78(0.25)	73.18(1.80)
	FJMI	79.08(2.18)	**78.29(1.45)**	74.28(2.33)
	PCFS	**79.48(0.94**)	78.15(0.35)	**78.84(1.04)**
SpamBase	LS	85.69(0.17)	75.63(0.12)	75.71(0.15)
	GCNC	88.27(1.19)	88.11(0.75)	88.96(1.09)
	FGUFS	88.63(1.16)	87.71(2.37)	87.71(1.34)
	FS	86.23(0.08)	86.42(3.89)	86.49(2.88)
	FAST	87.47(1.87)	80.50(4.14)	**88.88(1.93)**
	FJMI	86.89(3.21)	86.97(3.44)	87.35(2.36)
	PCFS	**90.06(1.25)**	**90.76(1.24)**	88.56(1.68)
Sonar	LS	79.21(1.38)	69.42(0.94)	79.09(1.94)
	GCNC	**82.33(2.52)**	74.36(2.48)	78.72(1.65)
	FGUFS	80.34(1.34)	75.73(1.84)	79.96(1.42)
	FS	73.81(1.89)	73.31(3.06)	73.76(0.54)
	FAST	75.95(0.98)	72.52(1.51)	73.24(2.73)
	FJMI	77.31(0.66)	75.96(1.16)	78.68(1.83)
	PCFS	81.89(2.08)	**77.28(1.88)**	**80.89(2.42)**
Arrhythmia	LS	56.78(2.34)	56.36(3.24)	57.49(3.24)
	GCNC	58.18(1.28)	59.04(2.30)	58.99(2.43)
	FGUFS	59.37(1.26)	**59.37(1.67)**	57.81(2.12)
	FS	52.89(0.25)	59.34(4.81)	54.15(2.68)
	FAST	57.76(3.56)	52.78(4.33)	57.33(1.24)
	FJMI	59.09(1.84)	58.74(3.22)	**59.56(3.16)**
	PCFS	**61.65(2.02)**	58.28(0.25)	59.34(2.76)
Madelon	LS	65.76(1.27)	61.88(2.80)	64.48(2.28)
	GCNC	66.56(1.38)	62.68(2.02)	63.68(2.48)
	FGUFS	67.78(2.17)	63.55(1.96)	65.79(2.13)
	mRMR	76.87(1.83)	73.15(2.14)	71.38(2.14)
	FAST	71.43(1.24)	73.35(3.78)	70.91(1.94)
	FJMI	77.12(3.02)	72.18(1.12)	**71.44(2.54)**
	PCFS	**78.76(1.57)**	**76.24(3.08)**	64.82(1.92)
Isolet	LS	83.78(1.08)	83.61(1.22)	82.72(2.42)
	GCNC	88.58(2.29)	82.78(2.42)	81.29(3.39)
	FGUFS	91.74(3.08)	86.66(1.82)	84.44(2.56)
	FS	87.95(2.21)	75.49(0.27)	75.58(1.31)
	FAST	84.28(2.48)	81.25(0.78)	80.49(3.16)
	FJMI	91.16(1.48)	**88.92(1.92)**	81.08(2.88)
	PCFS	**93.55(2.75)**	87.06(1.39)	**87.55(0.80)**
Multiple Features	LS	91.36(0.13)	91.43(0.12)	90.62(0.12)
	GCNC	91.81(2.26)	88.78(1.02)	92.55(0.54)
	FGUFS	92.28(1.24)	89.93(1.35)	92.79(2.38)
	FS	94.73(0.11)	92.32(1.42)	92.15(1.20)
	FAST	94.91(1.91)	92.48(1.95)	92.59(2.67)
	FJMI	**95.82(0.64)**	93.28(3.14)	93.04(2.05)
	PCFS	94.17(0.25)	**94.90(2.68)**	**93.14(1.55)**

Colon	LS	74.10(1.18)	73.82(1.67)	76.03(2.48)
	GCNC	75.17(1.76)	76.53(1.21)	76.47(3.38)
	FGUFS	78.23(2.23)	75.91(2.34)	77.96(1.22)
	FS	72.31(3.92)	69.15(2.26)	72.69(2.21)
	FAST	74.16(1.71)	73.26(3.52)	71.83(1.65)
	FJMI	82.37(4.04)	76.58(2.09)	81.43(1.72)
	PCFS	**83.19(3.21)**	**79.24(2.51)**	**81.87(1.13)**

Also, Tables 3-5 show the number of times the best results are achieved by different feature selection methods in ten independent run on SVM, NB and DT classifiers, respectively. It can be seen from Table 3-5 results that in most cases, the proposed methods obtained the highest rate compared to those of other methods in ten independent run with different classifier.

Table 3: Number of times different methods achieve the best results with SVM classifier

Dataset	LS	GCNC	FGUFS	FS	FAST	FJMI	PCFS
Colon	0	1	1	0	0	2	6
SRBCT	0	1	0	1	1	1	6
Leukemia	0	1	1	0	2	1	5
Prostate Tumor	0	1	1	0	1	2	5
Lung Cancer	0	1	1	0	1	2	5
Average	0	1	0.8	0.2	1	1.6	5.4

Table 4: Number of times different methods achieve the best results with NB classifier

Dataset	LS	GCNC	FGUFS	FS	FAST	FJMI	PCFS
Colon	0	1	2	0	0	1	6
SRBCT	1	0	0	1	1	2	5
Leukemia	0	1	1	0	2	2	4
Prostate Tumor	0	0	1	1	1	2	5
Lung Cancer	0	1	1	1	2	0	5
Average	0.2	0.6	1	0.6	1.2	1.4	5

Table 5: Number of times different methods achieve the best results with DT classifier

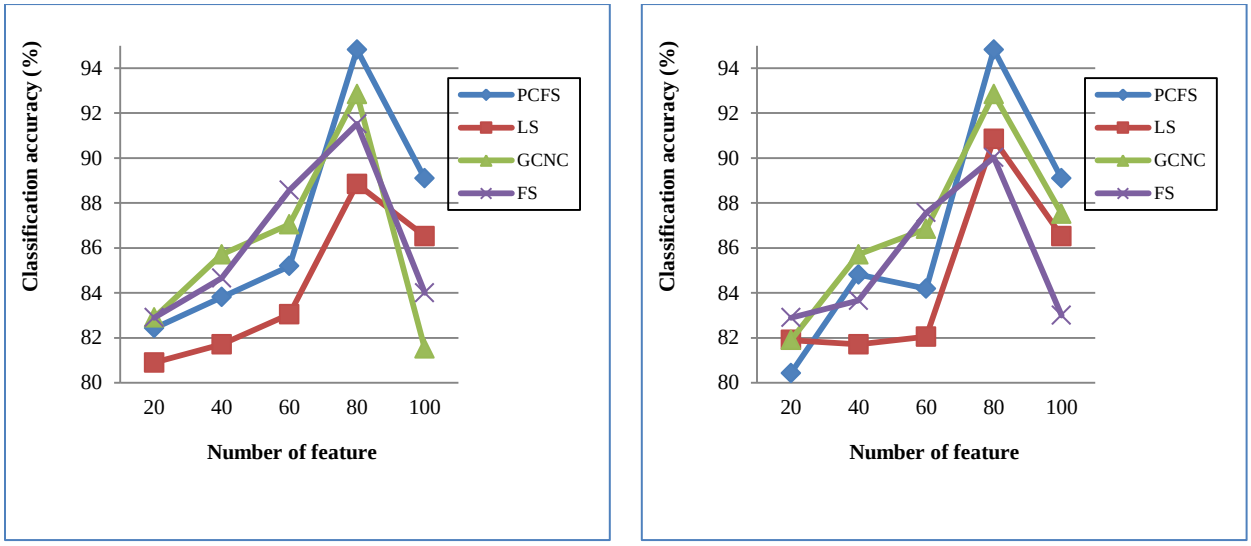
Dataset	**LS**	**GCNC**	**FGUFS**	**FS**	**FAST**	**FJMI**	**PCFS**
Colon	0	1	2	0	1	1	5
SRBCT	0	1	1	0	2	1	5
Leukemia	0	1	1	1	0	1	6
Prostate Tumor	0	1	1	1	1	2	4
Lung Cancer	0	1	1	1	1	1	5
Average	0	1	1.2	0.6	1	1.2	5

Table 6 records the average number of selected features of the seven feature selection methods in the ten independent runs for each dataset. It can be observed that, in general, a significant reduction of dimensionality is achieved by all the different methods by picking only a small portion of the original features. Overall, the proposed method the minimum number of selected features of 40.3 features. While this value for LS, GCNC, FGUFS, FS, FAST, and FJMI equal to 40.7, 41.2, 46.5, 47.0, 46.2, and 46.6, respectively.

Table 6: Average number of selected features in ten independent run

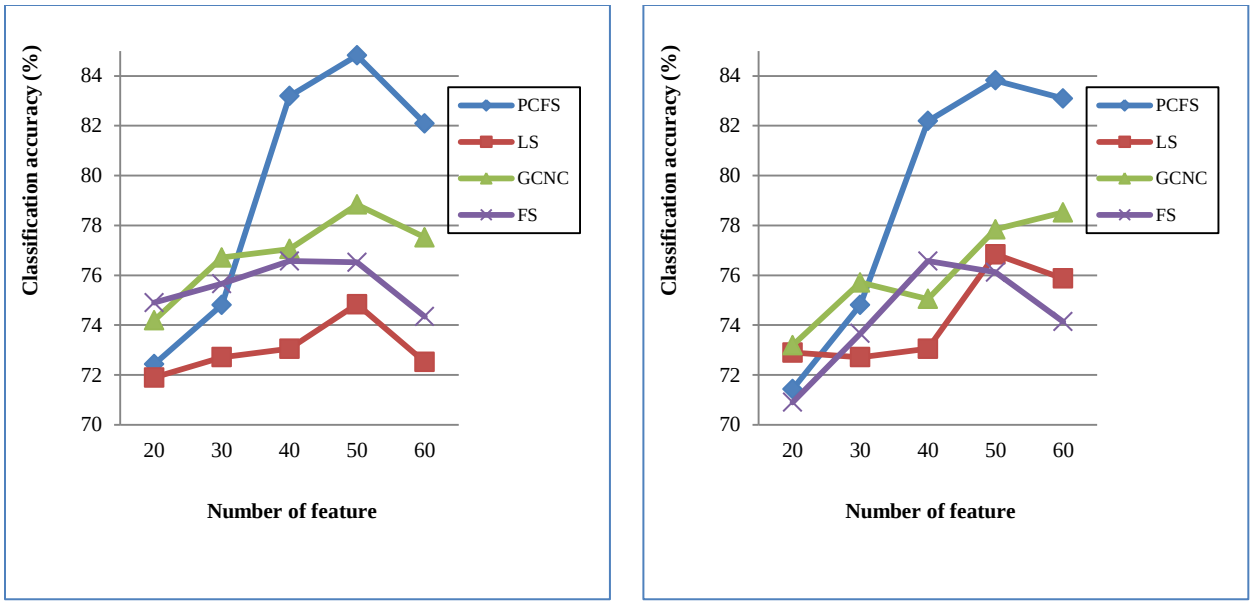
Dataset	PCFS	LS	GCNC	FGUFS	FS	FAST	FJMI
SPECTF	18.8	19.3	17.2	18.7	18.6	19.2	19.3
SpamBase	26.2	29.4	27.5	31.5	31.7	30.5	30.1
Sonar	16.8	18.2	17.4	23.4	21.5	21.7	22.6
Arrhythmia	19.2	18.7	18.3	20.3	21.5	21.6	21.7
Madelon	55.8	54.7	57.7	61.2	61.6	60.8	61.5
Isolet	73.4	71.8	72.8	81.3	83.4	80.2	81.5
Multiple Features	72.5	73.2	75.5	84.7	86.8	85.1	85.8
Colon	40.4	41.0	43.6	51.5	51.2	50.7	50.9
Average	40.3	40.7	41.2	46.5	47.0	46.2	46.6

Also, the comparison of the accuracy of the proposed method with the other feature selection methods according to the various numbers of selected features is performed by conducting several experiments. The classification accuracy (average over ten independent runs) curves of SVM and DT classifiers on Multiple Features and Colon datasets are respectively plotted in Figs. 1 and 2. The results of this table indicated that the proposed method, in most cases, is superior to other methods and has higher classification accuracy.



(a) With SVM **(b) With DT**

Fig. 1. Classification accuracy (average over 10 runs), on Multiple Features dataset with respect to the number of selected features with (a) SVM classifier, and (b) DT classifier



(a) With SVM **(b) With DT**

Fig. 2. Classification accuracy (average over 10 runs), on Colon dataset with respect to the number of selected features with (a) SVM classifier, and (b) DT classifier

Furthermore, a large number of experiments were performed to compare the execution time of the proposed method and other supervised and unsupervised feature selection methods. In these experiments, related execution times (in ms) for different methods are reported in Table 7. It can be concluded from the results reported in this table that, in most cases, the PCFS proposed method has lower running times than the other methods.

Table 7. Average execution time (in ms) of different feature selection methods over ten independent runs

Dataset	PCFS	LS	GCNC	FJUFS	FS	mRMR	FAST	FJMI
SPECTF	161	162	168	230	158	1906	**87**	2452
SpamBase	**1456**	1572	3780	5267	3926	15180	2908	17383
Sonar	260	289	275	741	**165**	3511	181	4781
Arrhythmia	4387	4088	7734	6282	5617	4842	**2863**	5906
Madelon	**8932**	17852	31849	44289	31795	19045	9642	22575
Isolet	**8468**	19710	33126	47891	32771	23734	11928	27826
Multiple Features	**9765**	18941	32674	36891	29810	21778	10403	27681
Colon	**7987**	10154	113598	139897	109884	11962	8295	16783

5. Conclusion:

Over the last ten years, the fast growth of computer and database technologies has led to the rapid growth of large-scale datasets. On the other hand, applications with high dimensional datasets that require high speed and accuracy are rapidly increasing. An important issue with data mining applications, including pattern recognition, classification, and clustering, is the curse of dimensionality, where the number of features is much higher compared to the number of patterns. From a general perspective, feature selection approaches are categorized into three groups, supervised, unsupervised, and semi-supervised. Supervised feature selection methods have a set of training patterns available, each of which is described by taking the values of the features with the labels, while in the unsupervised modes, feature selection methods encounter samples without labels. Semi-supervised feature selection is also a type of feature selection that employs both unlabeled and labeled data simultaneously to improve feature selection accuracy. In the present paper, a novel pairwise constraints-based method is proposed for feature selection. In the proposed method, in the first, the similarity between the pair constraints is calculated. Then an uncertainty region is created based on it. Then in an iterative process, most informative pairs are selected. The proposed method

was compared to different supervised, and unsupervised feature selection approaches, including LS, GCNC, FJUFS, FS, FAST, and FJMI. The reported findings indicate that, in most cases, the proposed approach is more accurate and selects fewer features. For example, numerical results showed that the proposed technique improved the classification accuracy by about 3% and reduced the number of picked features by 1%. Consequently, it can be said that the proposed method reduces the computational complexity of machine learning algorithm, despite the increase in classification accuracy.

Discussion:
Not applicable.

Abbreviations Used:
SVM= Support Vector Machine
TV= Term Variance
DT= Decision Tree
NB= Naïve Bayes
FS=Fisher Score
LS=Laplacian Score
LS= Laplacian Score for feature selection
RRFS= Relevance-Redundancy Feature Selection
GCNC= Graph Clustering with the Node Centrality
UFSACO= Unsupervised Feature Selection based on Ant Colony Optimization
FAST=Fast clustering-based feature selection method
FJMI=Five-way Joint Mutual Information
FGUFS=Factor Graph Model for Unsupervised Feature Selection

Availability of data and materials:
Data and material are available at any time. There are no restrictions on the development of data materials.

Competing interests:
The authors declare that they have no competing interests.

Funding:
This research did not receive any specific grant from funding agencies in the public, commercial, or not for profit sectors

Authors' contributions:
KB made substantial contributions in formulating the concept, design, and carried out experimentation, analysis and interpretation of data; MR participated in statistical analysis of result and investigation process. SF supported in coding, testing and drafting the manuscript. All authors read and approved the final manuscript.

Acknowledgements:
Not applicable.

Reference

[1]. Mafarja, M. and S. Mirjalili, Whale optimization approaches for wrapper feature selection. Applied Soft Computing, 2018. 62: p. 441-453.
[2]. Huang, D., X. Cai, and C.-D. Wang, Unsupervised feature selection with multi-subspace randomization and collaboration. Knowledge-Based Systems, 2019: p. 104856.
[3]. Tang, C., et al., Unsupervised feature selection via latent representation learning and manifold regularization. Neural Networks, 2019. 117: p. 163-178.

[4]. Moradi, P. and M. Rostami, Integration of graph clustering with ant colony optimization for feature selection. Knowledge-Based Systems, 2015. 84: p. 144-161.
[5]. Zhang, Y., et al., Binary differential evolution with self-learning for multi-objective feature selection. Information Sciences, 2020. 507: p. 67-85.
[6]. Pacheco, F., et al., Attribute clustering using rough set theory for feature selection in fault severity classification of rotating machinery. Expert Systems with Applications, 2017. 71: p. 69-86.
[7]. Dadaneh, B.Z., H.Y. Markid, and A. Zakerolhosseini, Unsupervised probabilistic feature selection using ant colony optimization. Expert Systems with Applications, 2016. 53: p. 27-42.
[8]. Tang, B. and L. Zhang, Local Preserving Logistic I-Relief for Semi-supervised Feature Selection. Neurocomputing, 2020.
[9]. Shi, C., et al., Multi-view adaptive semi-supervised feature selection with the self-paced learning. Signal Processing, 2020. 168: p. 107332.
[10]. Masud, M.A., et al., Generate pairwise constraints from unlabeled data for semi-supervised clustering. Data & Knowledge Engineering, 2019. 123: p. 101715.
[11]. Lu, H., et al., Community detection algorithm based on nonnegative matrix factorization and pairwise constraints. Physica A: Statistical Mechanics and its Applications, 2019: p. 123491.
[12]. Ahmed K. Farahat, Ali Ghodsi, and M.S. Kamel, Efficient greedy feature selection for unsupervised learning. Knowledge and Information Systems 2013. 35(2): p. 285-310.
[13]. Liu, Y. and Y.F. Zheng, FS_SFS: A novel feature selection method for support vector machines. Pattern Recognition, 2006. 39(7): p. 1333-1345.
[14]. Yudong ZHANG, et al., Binary PSO with mutation operator for feature selection using decision tree applied to spam detection. Knowledge-Based Systems, 2014. 26: p. Pages 22–31.
[15]. Aghdam, M.H., N. Ghasem-Aghaee, and M.E. Basiri, Text feature selection using ant colony optimization. Expert Systems with Applications, 2009. 36(3): p. 6843-6853.
[16]. Uğuz, H., A two-stage feature selection method for text categorization by using information gain, principal component analysis and genetic algorithm. Knowledge-Based Systems, 2011. 24(7): p. 1024-1032.
[17]. Shamsinejadbabki, P. and M. Saraee, A new unsupervised feature selection method for text clustering based on genetic algorithms. Journal of Intelligent Information Systems, 2011. 38(3): p. 669-684.
[18]. Chakraborti, T. and A. Chatterjee, A novel binary adaptive weight GSA based feature selection for face recognition using local gradient patterns, modified census transform, and local binary patterns. Engineering Applications of Artificial Intelligence, 2014. 33: p. 80-90.
[19]. Vignolo, L.D., D.H. Milone, and J. Scharcanski, Feature selection for face recognition based on multi-objective evolutionary wrappers. Expert Systems with Applications, 2013. 40(13): p. 5077-5084.
[20]. Kanan, H.R. and K. Faez, An improved feature selection method based on ant colony optimization (ACO) evaluated on face recognition system. Applied Mathematics and Computation, 2008. 205(2): p. 716-725.
[21]. da Silva, S.F., et al., Improving the ranking quality of medical image retrieval using a genetic feature selection method. Decision Support Systems, 2011. 51(4): p. 810-820.
[22]. Rashedi, E., H. Nezamabadi-pour, and S. Saryazdi, A simultaneous feature adaptation and feature selection method for content-based image retrieval systems. Knowledge-Based Systems, 2013. 39: p. 85-94.
[23]. Inbarani, H.H., A.T. Azar, and G. Jothi, Supervised hybrid feature selection based on PSO and rough sets for medical diagnosis. Comput Methods Programs Biomed, 2014. 113(1): p. 175-85.
[24]. Zhu, G.-N., et al., An integrated feature selection and cluster analysis techniques for case-based reasoning. Engineering Applications of Artificial Intelligence, 2015. 39(0): p. 14-22.
[25]. Jaganathan, P. and R. Kuppuchamy, A threshold fuzzy entropy based feature selection for medical database classification. Comput Biol Med, 2013. 43(12): p. 2222-9.
[26]. Saeys, Y., I. Inza, and P. Larranaga, A review of feature selection techniques in bioinformatics. Bioinformatics, 2007. 23(19): p. 2507-17.
[27]. Chandrashekar, G. and F. Sahin, A survey on feature selection methods. Computers & Electrical Engineering, 2014. 40(1): p. 16-28.

[28]. Liu, H. and L. Yu, Toward Integrating Feature Selection Algorithms for Classification and Clustering. IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, 2005. 17(4): p. 491 - 502
[29]. Huang, H., et al., Ant colony optimization-based feature selection method for surface electromyography signals classification. Comput Biol Med, 2012. 42(1): p. 30-8.
[30]. S. Theodoridis and C. Koutroumbas, Pattern Recognition, 4th Edn. Elsevier Inc, 2009.
[31]. Xiaofei He, Deng Cai, and P. Niyogi1, Laplacian Score for Feature Selection. Adv. Neural Inf. Process. Syst, 2005. 18: p. 507-514.
[32]. Artur J. Ferreira and M.A.T. Figueiredo, An unsupervised approach to feature discretization and selection. Pattern Recognition, 2012. 45(9): p. 3048–3060.
[33]. Tabakhi, S., P. Moradi, and F. Akhlaghian, An unsupervised feature selection algorithm based on ant colony optimization. Engineering Applications of Artificial Intelligence, 2014. 32: p. 112-123.
[34]. Belkin, M. and P. Niyogi, Laplacian eigenmaps and spectral techniques for embedding and clustering. Neural Inform. Process. Systems 1, 2002: p. 585-592.
[35]. Shi, J. and J. Malik, Normalized cuts and image segmentation. IEEE Trans. Pattern Anal. Machine Intell, 2000. 22(8): p. 888–905.
[36]. Chung, F., Spectral Graph Theory. In: Regional Conference Series in Mathematics American Mathematical Society, 1997. 92(92): p. 1-212.
[37]. Hongrong Cheng, et al., Graph-Based Semi-supervised Feature Selection with Application to Automatic Spam Image Identification. Computer Science for Environmental Engineering and EcoInformatics, 2011. 159: p. pp 259-264
[38]. Monalisa Mandal and A. Mukhopadhyay, Unsupervised Non-redundant Feature Selection: A Graph-Theoretic Approach. In: Proceedings of the International Conference on Frontiers of Intelligent Computing: Theory and Applications (FICTA), 2013: p. pp 373-380.
[39]. Bandyopadhyay, S., et al., Integration of dense subgraph finding with feature clustering for unsupervised feature selection. Pattern Recognition Letters, 2014. 40: p. 104-112.
[40]. Quanquan Gu, Zhenhui Li, and J. Han, Generalized Fisher Score for Feature Selection. In: Proceedings of the International Conference on Uncertainty in Artificial Intelligence, 2011.
[41]. Song, Q., J. Ni, and G. Wang, A Fast Clustering-Based Feature Subset Selection Algorithm for High-Dimensional Data. IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, 2013. 25(1): p. 1 - 14.
[42]. Tang, X., Y. Dai, and Y. Xiang, Feature selection based on feature interactions with application to text categorization. Expert Systems with Applications, 2019. 120: p. 207-216.
[43]. Moradi, P. and M. Rostami, A graph theoretic approach for unsupervised feature selection. Engineering Applications of Artificial Intelligence, 2015. 44: p. 33-45.
[44]. Wang, H., et al., A factor graph model for unsupervised feature selection. Information Sciences, 2019. 480: p. 144-159.
[45]. Asuncion, A. and D. Newman, UCI repository of machine learning datasets. Availablefrom: <http://archive.ics.uci.edu/ml/datasets.html>, 2007.
[46]. Hall, M., et al., The WEKA data mining software. Available from: <http://www.cs.waikato.ac.nz/ml/weka>.