Journal of Big Data

RESEARCH

Open Access

A novel method of constrained feature selection by the measurement of pairwise constraints uncertainty

Mehrdad Rostami[1], Kamal Berahmand[2*] and Saman Forouzandeh[3]

*Correspondence:
kamal.berahmand@hdr.qut.edu.au
[2] Department of Science and Engineering, Queensland University of Technology, Brisbane, Australia
Full list of author information is available at the end of the article

Abstract

In the past decades, the rapid growth of computer and database technologies has led to the rapid growth of large-scale datasets. On the other hand, data mining applications with high dimensional datasets that require high speed and accuracy are rapidly increasing. Semi-supervised learning is a class of machine learning in which unlabeled data and labeled data are used simultaneously to improve feature selection. 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. This method actually used the classification to reduce ambiguity in the range of values. First, the similarity values of each pair are collected, and then these values are divided into intervals, and the average of each interval is determined. In the next step, for each interval, the number of pairs in this range is counted. Finally, 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 results indicate that the proposed approach improves previous related approaches with respect to the accuracy of the constrained score. 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.

Keywords: Machine learning, Feature selection, Semi-supervised, Pairwise constraint, Uncertainty

Introduction

Along with the growth of data such as image data, meteorological data, particularly documents, dimensions of these data also increase [1]. According to the studied extensively, the accuracy of current machine learning methods generally decreases with high dimensional data that event referred to as the curse of dimensionality. An essential issue with machine learning techniques is the high-dimensionality problem of a dataset where the

Rostami *et al. J Big Data*   (2020) 7:83

Page 2 of 21

feature subset size much greater than pattern size. For example, in the medical applications that include very high-dimensional datasets, the classification parameters are also increased. Therefore, the performance of the classifier declines significantly [2–4].

For preventing the curse of dimensionality, some dimension (feature) reduction techniques are used [5–7]. Traditional techniques to reduce the dimensions are divided into two main categories: feature extraction and feature selection [8]. 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 [9, 10]. The subset of features in the pre-processing step is selected in filter methods independent of any learner method [11]. 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 three main groups: supervised, unsupervised and semi supervised [7]. 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 [12, 13].

A pairwise constraint is a pair of data belonging to the different clusters (cannot-link) or the same cluster (must-link) [14]. 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 [15]. 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 algorithms to consider the pairwise constraint in the process of the machine learning task, active learning algorithms to obtain the best and most valuable pair to increase the accuracy, the transformation of the objective functions in the machine learning task, 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 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. On the other, the number of final selected features imposes another challenge on feature selection methods. In other words, the number of relevant and non-redundant features is unknown; thus, the optimal number of selected features is not known either. In this proposed method, unlike many previous works, the optimal number of selected features is determined automatically based on the overall structure of the original features and their inner similarities.

The rest of this paper is organized as follows. "Related work" section summarizes related works to feature selection. "Proposed methodology" section introduces some preliminaries of this work, and our proposed method (PCFS) in details. The results of simulation and experimental analysis are illustrated in "Experimental analysis" section. The conclusion is given in "Conclusion" section.

Related work

The dimensionality reduction techniques are mostly divided into two categories: feature extraction and feature selection [16–18]. 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 [19]. The objective of the feature selection methods is seeking the related features with the most predictive information from the original feature set [20]. The feature selection was determined to be an essential technique in many practical applications, including text processing [21–23], face recognition [24–26], image retrieval [27, 28], medical diagnosis [29], case-based reasoning [30] and bioinformatics [31]. 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 [2–4].

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 [2, 29, 32]. 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 [33]. Since the labeled information is used, the supervised feature selection methods often show better performance compared to unsupervised and semi-supervised techniques [34]. 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. In semi-supervised learning, part of the data is labeled and part of it is unlabeled. 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. Sheikhpour et al. [35] provides a survey of feature selection methods. In this study, semi-supervised feature selection approaches are surveyed and taxonomies of these methods are introduced based on two different aspects. In [36] a novel Graph-based Semi-Supervised Sparse Feature Selection method is developed based on the mixed convex and non-convex minimization. The reported results of this method showed that the method selects the non-redundant and optimal subset of features and improves the performance of the machine learning task. In [37] a semi-supervised feature selection method is presented that integrates the neighborhood discriminant index and the Laplacian score method to efficiently work with both unlabeled and labeled data. The aim of this method is to find a set of relevant features that has a good ability to hold local geometrical structure and to identify samples belonging to different classes. Moreover, in [38] a semi-supervised feature selection method is developed for bipolar disorder. In this method, a novel semi-supervised technique is utilized to reduce the dimension of high-dimensional data. Also, Liu et al. [39] proposed Rough set based semi-supervised feature selection method. In this method, the unlabeled data can be predicted via various semi-supervised learning methods and the Local Neighborhood Decision Error Rate is developed to create multiple fitness functions to evaluate the relevance of the generated feature sets.

Feature selection methods might be divided into four categories: filter, wrapper, embedded, and hybrid approaches [40, 41]. 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 [42, 43]. 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 [44, 45]. 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 [46].

Term Variance (TV) [47], Laplacian Score for feature selection (LS) [48], Relevance-Redundancy Feature Selection (RRFS) [49], Unsupervised Feature Selection based on

Ant Colony Optimization (UFSACO) [50] 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 graph theory [51–53], spectral embedding [54], spectral clustering [55], and semi-supervised learning [56], 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. [57]. 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 [58], 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 [59], a dense subgraph finding method is selected for the unsupervised feature selection problem. In this paper, a novel normalized mutual information is used to calculate the similarity among two features.

Proposed methodology

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.

Background and notation

Let us review some definitions and concepts, which are the foundations of the proposed algorithm, before getting to the algorithm.

Neighborhoods and pairwise constraint

Laplacian ranking is the basis for the unsupervised method, including the selection of features with pairwise constraints, and in this method, the strongest feature in terms of the ability for preserving local is selected. The main key in assumptions in Laplacian feature selection is on the basis that the data belonging to the same class are closing

together and more similar. Laplacian ranking of the r_{th} feature of L_r that should be a minimum is expressed by Eq. (1):

$$
\left[
\begin{array}{l}
L_r = \dfrac{\sum_{i,j}\left(f_{ri}-f_{rj}\right)^2 S_{ij}}{\sum_i\left(f_{ri}-\mu_j\right)^2 D_{ij}} \\[2ex]
D_{ij} = \sum_j S_{ij} \\[2ex]
S_{ij} = \begin{cases} e^{-\frac{x_i-x_j^2}{t}} \; if x_i \, \text{and} \; x_j \, \text{are neighborhood} \\ \qquad 0, \qquad\quad \text{otherwise} \end{cases}
\end{array}
\right]
\tag{1}
$$

which S_{ij} can be expressed based on the relationship between the neighborhood and each data, and t is a fixed value that is initialized and neighborhood means that xi via the K of the nearest neighborhood reaches x_j and neighborhoods can have various concepts such as the similarity of data to each other. Rankings expressed are unsupervised and use no other information except for the data set. This article uses concepts such as Laplacian ranking and neighborhood, and on the assump.ion that pairwise constraint exists as ML (Must-link) and CL (Cannot-link), it attempts to select and rank appropriate features. So, all ML and CL set with datasets are prepared. Then, using Eq. (2), it is attempted to rank features. It should be noted that with the use of concepts of the neighborhood.

$$
\left[
\begin{array}{l}
C_r^1 = \dfrac{\sum_{(x_i,x_j)\in CL}\left(f_{ri}-f_{rj}\right)^2}{\sum_{(x_i,x_j)\in ML}\left(f_{ri}-f_{rj}\right)^2} \\[3ex]
C_r^2 = \left(\sum_{(x_i,x_j)\in CL}\left(f_{ri}-f_{rj}\right)^2\right) - \lambda\left(\sum_{(x_i,x_j)\in ML}\left(f_{ri}-f_{rj}\right)^2\right)
\end{array}
\right]
\tag{2}
$$

In where, C_r^1 and C_r^2 represent two types of rankings based on the pairwise constraint. In fact, features are selected that have the best ability to protect constraints. If there are two samples are in the ML set so the relevant feature means that the feature values are close together. If the two samples are in the CL set, relevant feature means that features values are far apart. In the follow.ng, for each feature, two types of ranking are calculated and from the maximum value, two rankings, feature selection is done.

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 {ML, CL}, and the clustering label is expressed with *lab*, then relations and Eq. (3) must be established. By closure of pairwise constraints, neighborhoods can be formed.

(3)

$$
\left[
\begin{array}{l}
\left(x_i, x_j, Ml\right) \wedge \left(x_i, x_k, Ml\right) \Rightarrow \left(x_j, x_k, Ml\right) \\
\left(x_i, x_j, Cl\right) \wedge \left(x_i, x_k, Cl\right) \Rightarrow \left(x_j, x_k, Cl\right) \\
\left(x_i, x_j, Ml\right) \Leftrightarrow lab_i = lab_j, \; in \; same \; cluster \\
\left(x_i, x_j, Cl\right) \Leftrightarrow lab_i \neq lab_j, \; not \; in \; same \; cluster
\end{array}
\right]
\tag{3}
$$

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 (4). 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:

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

(4)

Step 2: find the first valley points subject to:

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

(5)

Step 3: find the boundary of the uncertainty region:

$$m_d = D_v \text{ and } m_c = \max(D_i) - m_d$$

(6)

Step 4: find the pairs in similarity matrix that h.ve uncertainty relationship:

$$SimilarityMatrix S_{enij} : \begin{cases} m_d \leq if S_{enij} \leq m_c : uncertainity\ region \\ else : \qquad\qquad\qquad strong\ region \end{cases} \forall i, j$$

(7)

Weights of the terms obtained

Given that each feature has a certain weight and importance, and not all features may be required for the machine learning task, so in the first step it is necessary to determine the weight of each feature. For this purpose, Laplacian Score (LS) is used. LS is an unsupervised univariate filtering method which is based on the observation that if a data point is close to each other; it may belong to the same class. The basic idea of LS is to evaluate the feature relevance according to its power of locality preserving. The LS for the feature A is determined using Eq. (8):

Rostami *et al. J Big Data*   (2020) 7:83

Page 8 of 21

$$LS(S, A) = \frac{\sum_{i,j} (A(i) - A(j)) S_{ij}}{\sum_i (A(i) - \bar{A}) D_{ii}} \tag{8}$$

where, $A(i)$ represents the value of the feature A in the i-th a pattern, $\bar{A}$ denotes the average of the feature A, D is a diagonal matrix that $D_{ii} = \sum_j S_{ij}$, and S_{ij} represents the neighborhood relation between patterns, calculated as Eq. (9):

$$S_{ij} = \begin{cases} e^{\frac{x_i - x_j}{t}}, & if x_i \text{ and } x_j \text{ are } neighbors \\ 0, & otherwise \end{cases} \tag{9}$$

where, t is a suitable constant, x_i represents i-th pattern, and x_i and x_j are neighbors if x_i is among k nearest neighbors of x_j or x_j is among k nearest neighbors of x_i.

The proposed PCFS algorithm

In this section, a novel Pairwise Constraint Feature Selection method (PCFS) is proposed. This method uses pck-mean which is one of the soft constraints clustering algorithms 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-reduce() function, is affected

---

Algorithm 1 PCFS Clustering

Input: data set matrix $\{M_{ib}\}_{i=1...n}^{b=1...m}$,Data set $\mathcal{D} = \{x_1, x_2, x_3, ..., x_n\}$, Number of clusters K, must-link set ML, cannot-link set CL ,
Number of must-link ml, Number of cannot_link cl, Wm and Wc penalty of violthe ation, Neighborhoods $\{N_h\}_{h=1}^l$, data-reduced-term
matrix $\{RM_{ib}\}_{i=1...n}^{b=1...r}$, the number of features after reducing r.

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

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

---

by the current clustering and vice versa.

Briefly, the data set are embedded as a data-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 (lines 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 pck-means. 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 fea-

---

Algorithm 2   Dim-reduce()

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

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

  1. **Initializations:**
        $\{RM_{ib}\}_{i=1\ldots n}^{b=1\ldots r}$ = $\emptyset$,
  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.10**
  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** $\{RM_{ib}\}_{i=1\ldots n}^{b=1\ldots r}$

---

ture selection, pairwise constraints arise as input.

There are two main functions in this algorithm that respectively, S_{en}-func() in algorithm 3 and S*tr*-unc() 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. (10). Finally, Repeat will continue until the selected features are changed.

$$C_b = \frac{\sum_{(x_i, x_j) \in ML} \left(f_{bi} - f_{bj}\right)^2 \times S_{trij} + \frac{(1 - S_{enij})}{\sum_{(x_k, x_z) \in ML}(1 - S_{enkz})} \times \left(1 - S_{trij}\right)}{\sum_{(x_i, x_j) \in CL} \left(f_{bi} - f_{bj}\right)^2 \times S_{trij} + \frac{(1 - S_{enij})}{\sum_{(x_k, x_z) \in cL}(1 - S_{enkz})} \times \left(1 - S_{trij}\right)} \tag{10}$$

In which, S_{trij} 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:** $S_{en}[]\,_{i=1\ldots n}^{i=1\ldots n}$ **= Ø**

   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. $Temp - Sen[]\,_i^j$**=1**

  13.   **Else**

  14. $Temp - Sen[]\,_i^j$**=0**

  15.   **Endif**

  16.   **End for**

  17. S_{en}**=(Temp_s_en** S_{en}**)/2**

End for

Finally, the Matrix calculation of strength, S_{tr} and the uncertainty region as algorithm 4 is summarized. After finding the uncertainty region (line 3), 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. $S_{tr}\,[]\,_{j=1\ldots n}^{i=1\ldots n}$ **= Ø,**

   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 for**

Experimental analysis

To investigate 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 LS [48], GCNC [60], FGUFS [61], FS [62], FAST [63], FJMI [64], LS [48], PCA [65] and the description of this method is described below.

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

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

calculate the importance score of each feature, and then the selection of features is performed on the basis of the final importance scores.

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.

PCA (principal component analysis): PCA is a linear transformation-based multivariate analytical dimensionality reduction algorithm. PCA is often utilized to extract significant information from the high dimensional dataset.

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:

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} \tag{11}$$

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

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 [66] 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 a 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.

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 10 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.

Moreover, Weka (Waikato Environment for knowledge analysis) is the experimental workbench [67], 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. WEKA can be considered an advanced tool for machine learning and data mining. This free software can be used under the GNU General Public License. The software includes a set of "visualization" tools, data analysis methods and forecasting models that are put together in a graphical interface so that the user has the best way to execute commands. For this purpose, first the selected feature subset is determined by each feature selection method and then each

selected subset is sent to Weka tool for evaluation. Moreover, the used parameters of the mentioned classifiers 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 each of the compared methods were used to adjust the parameters.

Experimental result and discussion

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 different feature selection methods 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 italicface. Table 2 reveals that in most case, the proposed method performs better compared to other feature selection methods.

Moreover, Figs. 1, 2, 3 show the average classification accuracy over all datasets on the SVM, Naive Bayes, and Decision Tree classifiers, respectively. As can be seen in these figures, on SVM and Naive Bayes classifiers, the proposed method had the highest average classification accuracy, and on the Decision Tree classifier, FJUFS method won the highest rank. The results of Fig. 1 show that the proposed method obtained 82.87% average classification accuracy and achieved the first rank with a margin of 1.95 percent compared to the FJMI method, which obtained the second-best average classification accuracy. Moreover, from the Fig. 2 results, it can be seen that the differences between the obtained classification accuracy of the proposed method and the second-best ones (FJMI) and third-best ones (FGUFS) on Naive Bayes classifier were reported 1.17 (i.e., 80.38–79.21) and 3.07 (i.e., 80.38–77.31) percent. Furthermore, on the Decision Tree classifier, FGUFS method feature selection method gained the first rank with an average classification accuracy of 79.66%, and the proposed PCFS method was ranked second with an average classification accuracy of 79.02%.

Also, Tables 3, 4, 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, 4, 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 classifiers.
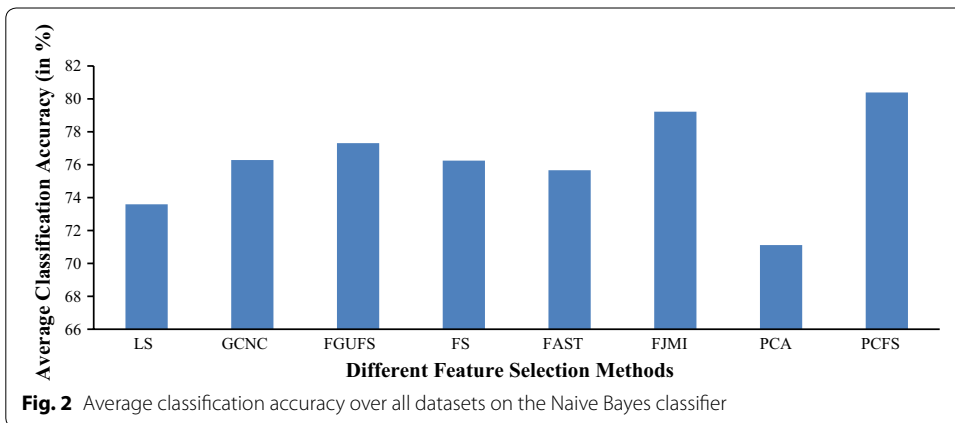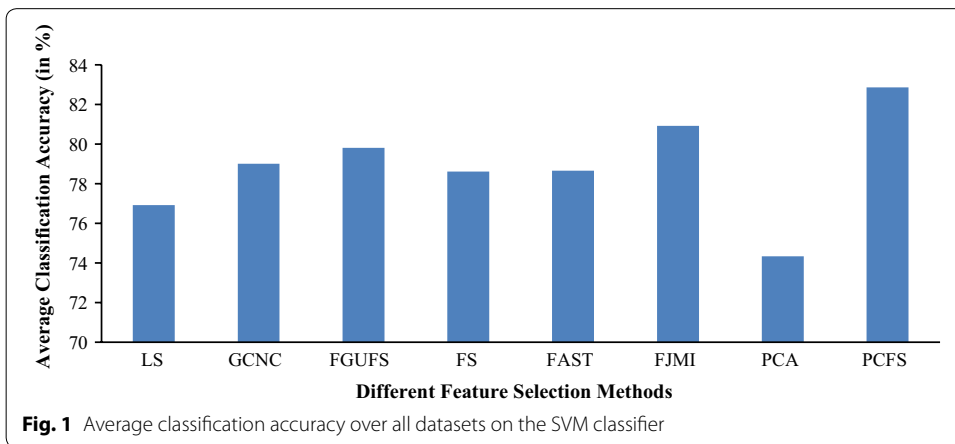
Table 6 records the average number of selected features of the seven feature selection methods in the ten independent runs for each dataset. It can be observed that, in general, a significant reduction of dimensionality is achieved by all the different methods by picking only a small portion of the original features. Overall, the proposed method the minimum number of selected features of 40.3 features. While this value for LS, GCNC, FGUFS, FS, FAST, FJMI, and PCA equal to 40.7, 41.2, 46.5, 47.0, 46.2, 46.6, and 44.4 respectively.
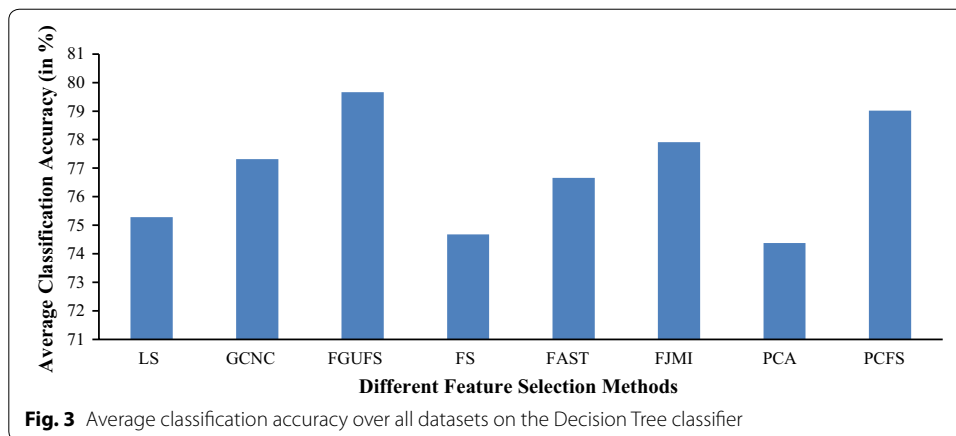
Table 2 Performance comparison of different feature selection methods on eight datasets

Dataset	Method	Evaluation criteria		
		SVM	NB	DT
SPECTF	LS	75.87(1.78)	76.79(1.84)	76.88(1.19)
	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(1.78)
	FAST	78.91(1.24)	76.78(0.25)	73.18(1.60)
	FJMI	79.08(2.18)	*78.29(1.45)*	74.22(2.43)
	PCA	75.56(1.98)	75.72(1.83)	77.81(1.71)
	PCFS	*79.41(0.92)*	78.10(0.31)	*78.94(1.04)*
SpamBase	LS	85.69(0.17)	75.63(0.12)	75.71(1.15)
	GCNC	88.27(1.19)	88.11(0.75)	88.96(1.09)
	FGUFS	88.63(1.16)	87.71(2.37)	86.71(1.35)
	FS	86.23(0.08)	86.42(3.89)	86.49(2.81)
	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)
	PCA	87.17(1.13)	84.12(0.75)	83.66(1.11)
	PCFS	*90.06(1.28)*	*90.76(2.34)*	88.46(1.63)
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)
	PCA	77.20(1.38)	68.42(2.31)	79.35(1.84)
	PCFS	81.81(2.09)	*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	57.74(3.56)	52.72(4.33)	57.33(1.24)
	FJMI	59.09(1.84)	58.74(3.22)	*59.56(3.16)*
	PCA	56.72(2.67)	55.31(3.54)	57.40(3.92)
	PCFS	*61.61(2.34)*	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(2.32)	63.68(2.48)
	FGUFS	67.78(2.15)	63.55(1.76)	65.79(2.13)
	mRMR	76.82(1.81)	73.15(2.14)	71.38(2.14)
	FAST	71.43(1.24)	73.35(2.48)	70.91(1.94)
	FJMI	77.12(3.02)	72.28(1.12)	*71.44(2.54)*
	PCA	66.72(1.47)	62.58(2.66)	67.18(2.32)
	PCFS	*78.76(1.67)*	*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)	75.41(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)
	PCA	64.71(1.83)	61.81(2.89)	63.48(2.21)
	PCFS	*93.55(2.75)*	87.06(1.39)	*87.55(1.80)*

Table 2 (continued)

Dataset	Method	Evaluation criteria		
		SVM	NB	DT
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	92.21(1.24)	89.93(1.35)	92.79(2.38)
	FS	94.78(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)
	PCA	92.29(1.64)	89.82(1.98)	91.78(2.42)
	PCFS	94.87(0.25)	*94.90(2.68)*	*93.14(2.45)*
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)	72.61(2.21)
	FAST	74.11(1.71)	73.26(3.52)	71.83(1.65)
	FJMI	82.37(4.04)	76.58(2.09)	81.48(1.72)
	PCA	76.12(1.54)	76.43(1.51)	76.42(2.12)
	PCFS	*83.19(2.32)*	*79.24(3.01)*	*81.92 (1.33)*

Fig. 1 Average classification accuracy over all datasets on the SVM classifier

Fig. 2 Average classification accuracy over all datasets on the Naive Bayes classifier

Fig. 3 Average classification accuracy over all datasets on the Decision Tree classifier

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

Dataset	LS	GCNC	FGUFS	FS	FAST	FJMI	PCA	PCFS
SPECTF	0	1	1	0	0	2	0	6
SpamBase	0	2	0	0	1	1	0	6
Sonar	0	1	0	0	0	1	1	7
Arrhythmia	0	1	1	0	0	0	0	8
Madelon	0	1	0	1	1	1	0	6
Isolet	0	1	1	0	2	1	0	5
Multiple Features	0	1	1	1	1	1	0	5
Colon	0	1	1	0	1	1	1	5
Average	**0**	**1.12**	**0.62**	**0.25**	**0.75**	**1**	**0.25**	**6**

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

Dataset	LS	GCNC	FGUFS	FS	FAST	FJMI	PCA	PCFS
SPECTF	1	1	1	0	0	2	0	5
SpamBase	0	1	0	1	1	1	0	6
Sonar	0	1	0	0	1	0	1	7
Arrhythmia	0	1	1	0	0	1	0	7
Madelon	0	1	1	0	1	1	0	6
Isolet	0	1	1	1	1	1	0	5
Multiple features	0	2	1	1	1	1	0	4
Colon	0	1	0	0	0	2	1	6
Average	0.12	1.12	0.62	0.37	0.62	1.12	0.25	5.75

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. 4 and 5. The results of this table indicated that the proposed method, in most cases, is superior to other methods and has highest classification accuracy.

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

Dataset	LS	GCNC	FGUFS	FS	FAST	FJMI	PCA	PCFS
SPECTF	1	1	1	0	0	2	1	4
SpamBase	0	1	0	1	1	0	0	7
Sonar	0	1	0	0	1	1	1	6
Arrhythmia	0	1	1	0	1	1	0	6
Madelon	0	1	1	0	1	0	1	6
Isolet	0	1	0	0	1	1	1	6
Multiple Features	0	2	0	1	1	1	0	5
Colon	0	1	0	1	0	1	1	6
Average	0.12	1.12	0.37	0.37	0.75	0.87	0.62	5.75

Table 6 Average number of selected features in ten independent run

Data Set	PCFS	LS	GCNC	FGUFS	FS	FAST	FJMI	PCA
SPECTF	18.8	19.3	17.2	18.7	18.6	19.2	19.3	21.4
SpamBase	26.2	29.4	27.5	31.5	31.7	30.5	30.1	22.1
Sonar	16.8	18.2	17.4	23.4	21.5	21.7	22.6	18.2
Arrhythmia	19.2	18.7	18.3	20.3	21.5	21.6	21.7	22.8
Madelon	55.8	54.7	57.7	61.2	61.6	60.8	61.5	58.2
Isolet	73.4	71.8	72.8	81.3	83.4	80.2	81.5	79.1
Multiple Features	72.5	73.2	75.5	84.7	86.8	85.1	85.8	85.7
Colon	40.4	41.0	43.6	51.5	51.2	50.7	50.9	48.1
Average	40.3	40.7	41.2	46.5	47.0	46.2	46.6	44.4
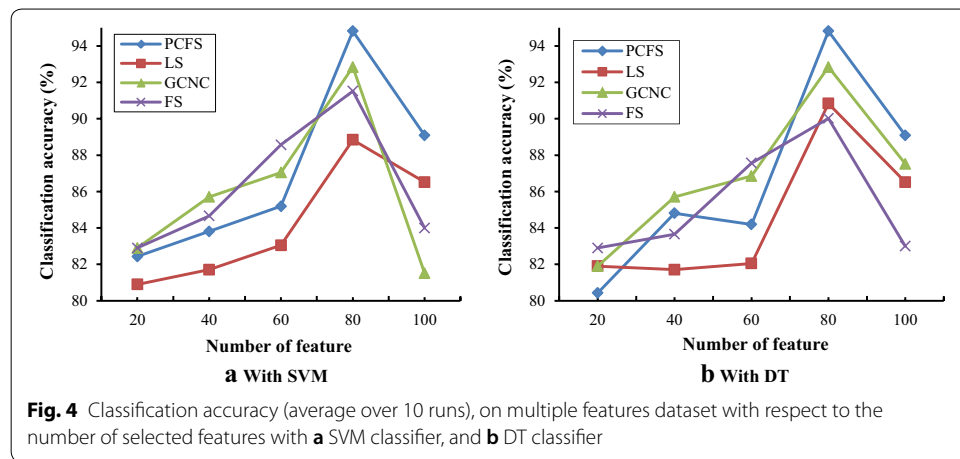
Fig. 4 Classification accuracy (average over 10 runs), on multiple features dataset with respect to the number of selected features with **a** SVM classifier, and **b** DT classifier

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
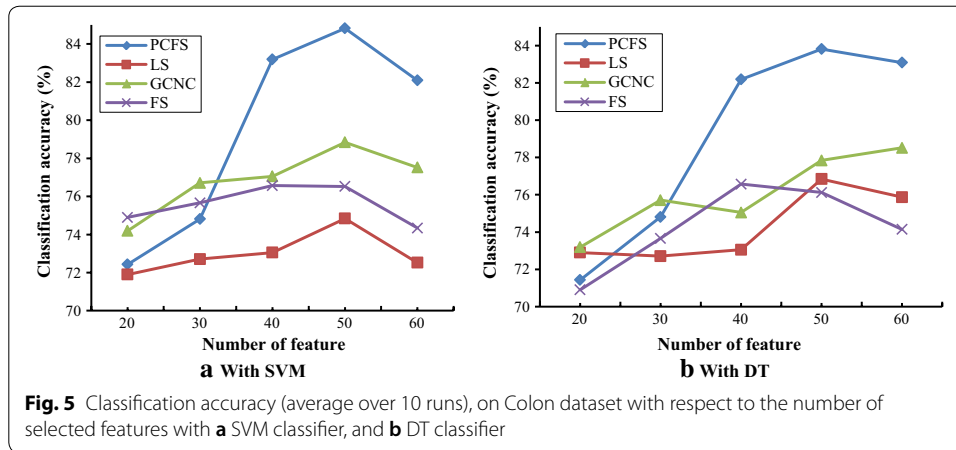
Fig. 5 Classification accuracy (average over 10 runs), on Colon dataset with respect to the number of selected features with **a** SVM classifier, and **b** DT classifier

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	PCA	PCA
SPECTF	161	162	168	230	158	1906	87	2452	162	162
SpamBase	1456	1572	3780	5267	3926	15180	2908	17383	3921	3921
Sonar	260	289	275	741	165	3511	181	4781	2892	2892
Arrhythmia	4387	4088	7734	6282	5617	4842	2863	5906	5814	5814
Madelon	8932	17852	31849	44289	31795	19045	9642	22575	36725	36725
Isolet	8468	19710	33126	47891	32771	23734	11928	27826	31732	31732
Multiple Features	9765	18941	32674	36891	29810	21778	10403	27681	30348	30348
Colon	7987	10154	113598	139897	109884	11962	8295	16783	112864	112864

table that, in most cases, the PCFS proposed method has lower running times than the other methods.

Complexity analysis

this subsection, the computational complexity of the proposed method is calculated. The first phase of the method which utilizes the PCFS clustering to determine of clusters. The time complexity of this phase is $O(In^2s)$ where I. Number of iterations for algorithm convergence indicates, n denotes the total number of initial features and s is the number of samples. In the next phase, Dim-reduce function is used to produce a reduced feature The complexity of Dim-reduce function is $O(n^2)$. Consequently, the final computational complexity of the PCFS methods is $O(In^2s + n^2)$. When the number of samples (i.e., s), and number of iterations (i.e., I), much smaller than the total number of features, the final time complexity of the proposed method can be reduced to $O(n^2)$.

Conclusion

Over the last 10 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, FJMI and PCA. 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 the machine learning algorithm, despite the increase in classification accuracy.

Authors' contributions
All authors read and approved the final manuscript.

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

Author details
[1] Department of Computer Engineering, University of Kurdistan, Sanandaj, Iran. [2] Department of Science and Engineering, Queensland University of Technology, Brisbane, Australia. [3] Department of Computer Engineering, University of Applied Science and Technology, Center of Tehran Municipality, ICT org, Tehran, Iran.

References
1. Rostami M, et al. Integration of multi-objective PSO based feature selection and node centrality for medical datasets. Genomics. 2020;112(6):4370–84.
2. Saeys Y, Inza I, Larranaga P. A review of feature selection techniques in bioinformatics. Bioinformatics. 2007;23(19):2507–17.
3. Chandrashekar G, Sahin F. A survey on feature selection methods. Comput Electr Eng. 2014;40(1):16–28.
4. Liu H, Yu L. Toward integrating feature selection algorithms for classification and clustering. IEEE Trans Knowl Data Eng. 2005;17(4):491–502.
5. Mafarja M, Mirjalili S. Whale optimization approaches for wrapper feature selection. Appl Soft Comput. 2018;62:441–53.
6. Huang D, Cai X, Wang C-D. Unsupervised feature selection with multi-subspace randomization and collaboration. Knowl Based Syst. 2019;182:104856.
7. Tang C, et al. Unsupervised feature selection via latent representation learning and manifold regularization. Neural Netw. 2019;117:163–78.
8. Moradi P, Rostami M. Integration of graph clustering with ant colony optimization for feature selection. Knowl Based Syst. 2015;84:144–61.

9. Zhang Y, et al. Binary differential evolution with self-learning for multi-objective feature selection. Inf Sci. 2020;507:67–85.
10. Pacheco F, et al. Attribute clustering using rough set theory for feature selection in fault severity classification of rotating machinery. Expert Syst Appl. 2017;71:69–86.
11. Dadaneh BZ, Markid HY, Zakerolhosseini A. Unsupervised probabilistic feature selection using ant colony optimization. Expert Syst Appl. 2016;53:27–42.
12. Tang B, Zhang L. Local preserving logistic I-relief for semi-supervised feature selection. Neurocomputing. 2020;399:48–64. https://doi.org/10.1016/j.neucom.2020.02.098.
13. Shi C, et al. Multi-view adaptive semi-supervised feature selection with the self-paced learning. Signal Processing. 2020;168:107332.
14. Masud MA, et al. Generate pairwise constraints from unlabeled data for semi-supervised clustering. Data Knowl Eng. 2019;123:101715.
15. Lu H, et al. Community detection algorithm based on nonnegative matrix factorization and pairwise constraints. Phys A Stat Mech Appl. 2019;545:123491.
16. Farahat AK, Ghodsi A, Kamel MS. Efficient greedy feature selection for unsupervised learning. Knowl Inf Syst. 2013;35(2):285–310.
17. Liu Y, Zheng YF. FS_SFS: a novel feature selection method for support vector machines. Pattern Recogn. 2006;39(7):1333–45.
18. Zhang Y, et al. Binary PSO with mutation operator for feature selection using decision tree applied to spam detection. Knowl Based Syst. 2014;26:22–31.
19. Xue B, et al. A survey on evolutionary computation approaches to feature selection. IEEE Trans Evol Comput. 2015;20(4):606–26.
20. Mishra M, Mishra P, Somani AK. Understanding the data science behind business analytics. In: Big Data Analytics; 2017. p. 93–116.
21. Aghdam MH, Ghasem-Aghaee N, Basiri ME. Text feature selection using ant colony optimization. Expert Syst Appl. 2009;36(3):6843–53.
22. Uğuz H. A two-stage feature selection method for text categorization by using information gain, principal component analysis and genetic algorithm. Knowl Based Syst. 2011;24(7):1024–32.
23. Shamsinejadbabki P, Saraee M. A new unsupervised feature selection method for text clustering based on genetic algorithms. J Intell Inf Sys. 2011;38(3):669–84.
24. Chakraborti T, Chatterjee A. A novel binary adaptive weight GSA based feature selection for face recognition using local gradient patterns, modified census transform, and local binary patterns. Eng Appl Artif Intell. 2014;33:80–90.
25. Vignolo LD, Milone DH, Scharcanski J. Feature selection for face recognition based on multi-objective evolutionary wrappers. Expert Syst Appl. 2013;40(13):5077–84.
26. Kanan HR, Faez K. An improved feature selection method based on ant colony optimization (ACO) evaluated on face recognition system. Appl Math Comput. 2008;205(2):716–25.
27. Silva SF, et al. Improving the ranking quality of medical image retrieval using a genetic feature selection method. Decis Support Syst. 2011;51(4):810–20.
28. Rashedi E, Nezamabadi-pour H, Saryazdi S. A simultaneous feature adaptation and feature selection method for content-based image retrieval systems. Knowl Based Syst. 2013;39:85–94.
29. Inbarani HH, Azar AT, Jothi G. Supervised hybrid feature selection based on PSO and rough sets for medical diagnosis. Comput Methods Programs Biomed. 2014;113(1):175–85.
30. Zhu G-N, et al. An integrated feature selection and cluster analysis techniques for case-based reasoning. Eng Appl Artif Intell. 2015;39:14–22.
31. Jaganathan P, Kuppuchamy R. A threshold fuzzy entropy based feature selection for medical database classification. Comput Biol Med. 2013;43(12):2222–9.
32. Huang H, et al. Ant colony optimization-based feature selection method for surface electromyography signals classification. Comput Biol Med. 2012;42(1):30–8.
33. Janecek, A., et al. *On the relationship between feature selection and classification accuracy*. in *New challenges for feature selection in data mining and knowledge discovery*. 2008.
34. Rostami M, Moradi P. A clustering based genetic algorithm for feature selection. In: 2014 6th Conference on information and knowledge technology (IKT). IEEE, Shahrood, Iran, 27–29 May 2014. https://doi.org/10.1109/IKT.2014.7030343.
35. Sheikhpour R, et al. A Survey on semi-supervised feature selection methods. Pattern Recogn. 2017;64:141–58.
36. Sheikhpour R, et al. A robust graph-based semi-supervised sparse feature selection method. Inf Sci. 2020;531:13–30.
37. Pang Q-Q, Zhang L. Semi-supervised neighborhood discrimination index for feature selection. Knowl Based Syst. 2020;204:106224.
38. Squarcina L, et al. Automated cortical thickness and skewness feature selection in bipolar disorder using a semi-supervised learning method. J Affect Disord. 2019;256:416–23.
39. Liu K, et al. Rough set based semi-supervised feature selection via ensemble selector. Knowl Based Syst. 2019;165:282–96.
40. Hall MA, Smith LA, Practical feature subset selection for machine learning; 1998. p. 181–91.
41. Kira K, Rendell LA, A practical approach to feature selection. In: Machine Learning Proceedings 1992. Elsevier. 1992, p. 249–256.
42. Dash M, Liu H. Feature selection for classification. Intell Data Anal. 1997;1(3):131–56.
43. Tang J, Alelyani S, Liu H. Feature selection for classification: a review. Data classification: Algorithms and applications, 2014, p. 37
44. Semwal VB, et al. An optimized feature selection technique based on incremental feature analysis for bio-metric gait data classification. Multimed Tools Appl. 2017;76(22):24457–75.
45. Masoudi-Sobhanzadeh Y, Motieghader H, Masoudi-Nejad A. FeatureSelect: a software for feature selection based on machine learning approaches. BMC Bioinform. 2019;20(1):170.

46. Solorio-Fernández S, Carrasco-Ochoa JA, Martínez-Trinidad JF. A new hybrid filter–wrapper feature selection method for clustering based on ranking. Neurocomputing. 2016;214:866–80.
47. Theodoridis S, Koutroumbas C. Pattern recognition. 4th ed. Amsterdam: Elsevier Inc; 2009.
48. He X, Cai D, Niyogil P. Laplacian score for feature selection. Adv Neural Inf Process Syst. 2005;18:507–14.
49. Ferreira AJ, Figueiredo MAT. An unsupervised approach to feature discretization and selection. Pattern Recogn. 2012;45(9):3048–60.
50. Tabakhi S, Moradi P, Akhlaghian F. An unsupervised feature selection algorithm based on ant colony optimization. Eng Appl Artif Intell. 2014;32:112–23.
51. Berahmand K, Bouyer A, Vasighi M. Community detection in complex networks by detecting and expanding core nodes through extended local similarity of nodes. IEEE Transact Comput Soc Syst. 2018;5(4):1021–33.
52. Berahmand K, Bouyer A. A link-based similarity for improving community detection based on label propagation algorithm. J Syst Sci Complexity. 2019;32(3):737–58.
53. Berahmand K, Bouyer A. LP-LPA: a link influence-based label propagation algorithm for discovering community structures in networks. Int J Mod Phys B. 2018;32(06):1850062.
54. Belkin M, Niyogi P. Laplacian eigenmaps and spectral techniques for embedding and clustering. Neural Inform Process Syst. 2002;1:585–92.
55. Shi J, Malik J. Normalized cuts and image segmentation. IEEE Trans Pattern Anal Mach Intell. 2000;22(8):888–905.
56. Chung F. Spectral graph theory. Region Conf Ser Math Am Math Soc. 1997;92(92):1–212.
57. Cheng H, et al. Graph-based semi-supervised feature selection with application to automatic spam image identification. Comput Sci Environ Eng EcoInform. 2011;159:259–64.
58. Mandal M, Mukhopadhyay A. 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. 373–380.
59. Bandyopadhyay S, et al. Integration of dense subgraph finding with feature clustering for unsupervised feature selection. Pattern Recogn Lett. 2014;40:104–12.
60. Moradi P, Rostami M. A graph theoretic approach for unsupervised feature selection. Eng Appl Artif Intell. 2015;44:33–45.
61. Wang H, et al. A factor graph model for unsupervised feature selection. Inf Sci. 2019;480:144–59.
62. Gu Q, Li Z, Han J. Generalized Fisher score for feature selection. In: Proceedings of the International Conference on Uncertainty in Artificial Intelligence, 2011.
63. Song Q, Ni J, Wang G. A Fast Clustering-Based Feature Subset Selection Algorithm for High-Dimensional Data. IEEE Trans Knowl Data Eng. 2013;25(1):1–14.
64. Tang X, Dai Y, Xiang Y. Feature selection based on feature interactions with application to text categorization. Expert Syst Appl. 2019;120:207–16.
65. Abdi H, Williams LJ. Principal component analysis. Wiley interdisciplinary reviews: computational statistics. 2010;2(4):433–59.
66. Asuncion A, Newman D. UCI repository of machine learning datasets. 2007; http://archive.ics.uci.edu/ml/datasets.html.
67. Hall M et al. The WEKA data mining software. http://www.cs.waikato.ac.nz/ml/weka.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.