AUT Journal of Mathematics and Computing

AUT J. Math. Comput., 5(2) (2024) 91-109
https://doi.org/10.22060/AJMC.2023.21449.1090

Original Article

Unsupervised feature selection by integration of regularized self-representation and sparse coding

Masoud Karimzadeh[a], Parham Moradi*[b], Abdulbaghi Ghaderzadeh[a]

[a] *Department of Computer Engineering, Sanandaj Branch, Islamic Azad University, Sanandaj, Iran*
[b] *Department of Computer Engineering, University of Kurdistan, Sanandaj, Iran*

ABSTRACT: Due to the development of social networks and the Internet of things, we recently have faced with large datasets. High-dimensional data is mixed with redundant and irrelevant features, so the performance of machine learning methods is reduced. Feature selection is a common way to tackle this issue with the aim of choosing a small subset of relevant and non-redundant features. Most of the existing feature selection works are for supervised applications, which assume that the information on class labels is available. While in many real-world applications, it is not possible to provide complete knowledge of class labels. To overcome this shortcoming, an unsupervised feature selection method is proposed in this paper. The proposed method uses the matrix factorization-based regularized self-representation model to weight features based on their importance. Here, we initialize the weights of features based on the correlation among features. Several experiments are performed to evaluate the effectiveness of the proposed method. Then the results are compared with several baselines and state-of-the-art methods, which show the superiority of the proposed method in most cases.

Review History:
Received:05 June 2022
Revised:28 January 2023
Accepted:12 February 2023
Available Online:01 April 2024

Keywords:
Unsupervised feature selection
Subspace learning
Self-representation

MSC (2020):
68T30; 68T01

1. Introduction

High-dimensional data results from the improvement of digital technologies, social networks, and the Internet of things [39]. In general, the high-dimensional involves irrelevant, redundant, missing, and noisy features, which harm the performance of machine learning methods and increase the time complexity [38, 39]. A primary solution is to choose a small number of features that can approximate the properties of the original data [39]. For example, a prediction task on social media includes many features as inputs which increase the time and memory complexities of the machine learning methods. Therefore, using the data processing to choose a small set of informative features helps machine learning methods analytically be practicable in low-dimensional space [4].

The main aim of the feature selection algorithm is to improve predictive accuracy, increasing comprehensibility and speeds of machine learning algorithms by identifying relevant features and eliminating irrelevant, redundant, or noisy ones. Those features without carrying predictive information regarding the class attribute are known as irrelevant features. Moreover, those redundant features provide no more information than the currently selected

*Corresponding author.
E-mail addresses: masoud.karimzadeh@iau.ac.ir, p.moradi@uok.ac.ir, b.ghaderzadeh@iausdj.ac.ir

ones. Feature selection methods have many real-world applications such as text classification [1, 21], clinical dataset classification [35], Alzheimer's classification [20].

In [30] an unsupervised feature selection initiated from the subspace clustering to keep the similarities by representation learning of low dimensional subspaces among the samples. A unified objective function aligned with an $L_{2,1}-$norm to address a regularized regression model.

Many feature selection methods are proposed as a pre-processing method to select relevant features and eliminate redundant ones. These methods are divided into filter, wrapper, and embedded approaches [32]. Wrapper methods utilize a learning model to evaluate the feature subsets. Although, these methods provide accurate results, however, learning algorithms are time-consuming and cannot be applied to real-world applications with thousands of features [12]. In contrast, filter-based methods use a statistical measure to evaluate the importance of features. They generally compute the relevancy of features to the target classes and measure the redundancy of selected features. Thus they are more efficient than the wrapper ones in terms of time complexity [9, 19]. Embedded-based methods integrate both wrapper and filter methods into the learning process. It is generally an NP-complete task and finding an accurate solution is impossible for real-world applications. To this end, several swarm intelligence-based methods such as Ant Colony Optimisation (UFSACO) [37], Particle Swarm Optimisation, Artificial Bee Colony [11], Firefly Algorithm [21, 27], Grasshopper Optimisation Algorithm [14], and Salp Swarm algorithm [13] was utilized to find a near-optimal solution within a reasonable time.

By considering using the class label in the feature selection process, these methods are categorized into supervised, semi-supervised, and unsupervised machine learning [32]. Most existing techniques are only proposed for supervised tasks and fail while the class labels are inaccessible. To this end, several unsupervised feature selection methods are introduced [18]. Unsupervised feature selection is indispensable in many data mining applications because of instances with unknown labels in data [26]. The method proposed in [44] combines the correlation coefficient and mutual information to compute the correlation between features and consider the correlation value as the redundant weight among features.

Some unsupervised feature selection methods use evaluation measures to rank them and choose those high-ranked ones [45]. Recently, Nonnegative Matrix Factorization (NMF) was used to weigh features effectively. In [39], the idea of NMF for subspace clustering was used to formalize the feature selection task and developed an iterative method to obtain the NMF components. Most of these works learned a cluster indicator and used it to perform the feature selection. However, the learned indicator may be far from the real clusters and mislead the feature selection process. To solve this issue the idea of self-representation is proposed. Each feature is represented by a linear combination of the other features using this idea. For example, in [45] a self-representation method combined with the objective function of NMF is used for unsupervised feature selection. In [46] the authors combined the idea of self-representation with principal component analysis for unsupervised feature selection.

Authors of [25] incorporate the manifold regularization into the feature selection model to find the low-dimensional embedding. Due to the improvement in the performance and simplicity of implementation, the local linear embedding has received much attention from researchers. However, self-representation and manifold learning are usually applied to the original feature subspace, which results in suboptimal solution. The authors of [33], integrated both graph matrix learning and low-dimensional space learning into a single framework. In [38] self-representation and manifold regularization are embedded for unsupervised feature selection. The authors of [10] proposed an unsupervised feature selection by using graph matrix learning and low-dimensional space learning. In [43], the authors recently employed self-paced learning regularization for unsupervised feature selection. Self-paced learning uses the theory of curriculum learning, which first learns the simple knowledge and then gradually increases the learning difficulty. In [6] deep learning strategies along with some non-linear functions are used together to provide a representation or a decision, and a cascade of embedding is used to rank features and eliminate the redundant ones.

Existing unsupervised feature selection methods suffer from some significant issues. First, most of them are based on self-representation learning. Each feature is considered as a linear combination of the other features, results to getting more information about the original feature space. Most of the existing unsupervised feature selection methods employ the global geometrical data structure. Using the local structure of data is much more critical than the global one in the unsupervised feature selection. To solve these issues, in this paper, an unsupervised feature selection method based on nonnegative matrix factorization and subspace clustering is proposed. The proposed method uses the idea of subspace clustering to preserve the local structure of data and employ it in the feature selection process. Then the idea of nonnegative matrix factorization is used to weigh the features and then select those of the top high-weight ones. The gradient descent method is then used to solve the objective function. Finally, an iterative update algorithm is proposed to identify the main components. Here we have used a novel method based on subspace clustering for initializing the weights of features. The proposed method has several novelties compared to the state of the art methods, which are listed as follows:

- The proposed method is a filter feature selection which means that it does not employ any learning model to

evaluate the feature subsets.

- This method first weights the features byconsidering the importance and then chooses a set of top valued ones as a final feature set. Unlike self-representation methods which consider each feature as a combination of all other features that may result in losing information.
- Whole feature space is used to weigh the features.
- The proposed method uses subspace clustering to incorporate the local manifold of features in its process.
- Our method is based on the nonnegative matrix factorization methods. In this paper, we proposed an iterative process to find its components. Here, the results of subspace clustering are used as initial values for the components to avoid trapping into the local optima.

Several experiments were performed to assess the performance of the proposed method. The obtained results show the effectiveness of the proposed method compared to a set of traditional and state-of-the-art feature selection methods. The remainder of this paper is organized as follows. Section 2 provides a survey on existing feature selection methods. Section 3 provides the details of the proposed method, and the results of experiments are provided in Section 4. Finally Section 5 concludes the paper and provides some future works.

2. Related Work

The feature selection method aims to choose a set of relevant and non-redundant features among thousands of those features to improve the performance of the machine learning methods. These methods are generally classified into supervised, semi-supervised, and unsupervised methods. The class label is provided through the feature selection process in supervised methods. A majority of feature selection methods belong to this category. While in many real-world applications, the class labels are unknown. To solve this shortcoming, unsupervised feature selection methods aim at choosing the features without requiring the class labels. This is a complex task, and till now, only a few unsupervised methods have been proposed. Some evaluation metrics such as Fisher score (FS), Rank ratio, Laplace score (LS), and Variance to evaluate were used in unsupervised methods to specify the importance of features. In some cases, some instances include labels, and any label does not provide others. Semi-supervised methods are proposed to deal with the tasks with labeled and unlabeled instances. Semi-supervised methods seek to take the model structure of the data from labeled samples and then utilize it for unlabeled ones [43]. In [46] a sparse learning framework combined with subspace learning is used for unsupervised feature selection. The locality preserving property is used to keep the locality preserving projection to keep the local structure of data. Unlike previous works, using the principal component analysis preserves the maximum variance of the data. However, fisher score ignores the local information just as ignores the correlation between features. To tackle this issue, the authors of [9] proposed a criterion called iteratively local fisher score which pays more attention to the local structure of data.

Several unsupervised feature selection algorithms were proposed to deal with the high-dimensional issue in the absence of the class label. For instance, in [19] the top high ranked with maximum variance and discriminative enough for classification are chosen as the final feature set. Laplacian score [12] is another unsupervised method that identifies features that preserve the local manifold structure of data. Some other criteria such as feature similarity [26] and trace ratio [29] were proposed to identify the importance of features. In [4], a method called Multi-Cluster Feature Selection (MCFS) was proposed, which holds the local manifold structure by using the spectral analysis and then identifies those features that keep the clustering properties in advance. A flexible manifold embedding as an accepted dimension reduction framework was proposed in [28]. Several feature selection methods used this type of embedding. A robust unsupervised method was proposed in [33]. This method uses a flexible manifold embedding, NMF, and $L_{2,1}-$norm to run both feature selection and clustering methods simultaneously. A general framework using sparse representation and joint embedding was proposed in [15]. Recently, some methods based on the self-representation-based methods were proposed [16, 24, 34, 38, 43, 45, 46]. The main idea approximates features via a linear combination of its relevant features. The coefficient matrix with sparsity constraint is then employed as weights of features.

Early studies show that keeping the local manifold structure of data is an essential task in unsupervised feature selection [23]. As a result, many unsupervised feature selection methods make extensive use of the graph Laplacian regularization term to capture local geometric structure [16, 22, 23, 34, 40, 42]. Recently several unsupervised methods for feature selection that use self-representation and graph regularization employ the Frobenius norm in their objective functions. The models become sensitive to outliers if this norm is used. A fixed similarity graph is also part of most existing graph-regularized methods, which are manually set beforehand to preserve the local geometric structure. Due to the unreliable similarity graph and improper assignment of neighbors, suboptimal

results are generated. To solve the issues mentioned, an unsupervised feature selection algorithm was proposed In [38] by utilizing dual self-representation with manifold regularization.

The authors of [22] integrated the local geometric structure consistency and redundancy minimization into a unified framework for unsupervised feature selection. The pairwise constraints are also used in [11] to specify whether a pair of data samples belong to the same class or different classes. In many tasks, pairwise constraints arise and are more practical and cheaper than class labels. In [28], a robust feature selection method is proposed that utilizes $L_{2,1}-$norm which is robust against outliers. The regression-based objective function used in this method efficiently identifies prominent features. Many methods have also employed flexible manifold embedding as a general framework for dimensionality reduction. In [42] a spectral feature selection algorithm is proposed for managing feature redundancy. The formulation for this method is based on a sparse multi-output regression with a $L_{2,1}-$norm constraint. This is done by measuring their ability to keep sample similarity and relevant feature identification [42].

The main aim of unsupervised feature selection algorithms is to choose those features that preserve the data's manifold structure in advance. The authors of [40] proposed a method that merges both discriminative analysis and $L_{2,1}-$norm minimization. A linear classifier was used to approximate the class label in this method. This method simultaneously exploits discriminative information and feature correlations. In [38] proposed an idea based on the dual self-representation and manifold regularization for weighting features without requiring the class label. This method learns the feature representation to indicate the importance of features. This method learns the similarity graph to preserve the local structure of data. Due to the use of $L_{2,1}-$norm it is robust to outliers. A hybrid method by integration of feature selection and feature weighting was proposed in [36].

The method proposed in [7], keeps the local structure of data by using $L_{2,1}-norm$ and clustering. This method uses the cluster centers as pseudo labels of the data. This method proposed two matrices, one for the latent cluster centers, and the other for sparse representation. The first matrix ensures that pseudo labels are closer enough to the real cluster centers and also helps the sparse representation of different classes far enough away from each other. Therefore, sparse representation preserves local structures and manifold regularization preserves the geometrical structure of data.

The idea of the proposed method in [8] was that a sparse representation of data ideally corresponds to a combination of a few points from its subspace. An unsupervised feature selection combined with subspace clustering was proposed in [39]. This method uses matrix factorizing along with a kernel method for unsupervised feature selection. The authors of [45], proposed an algorithm named RMFFS, which stands for Regularized Matrix Factorization Feature Selection which works as unsupervised feature selection. Due to the use of matrix factorization for feature selection, taking the correlation among features into account is the main advantage of their algorithm. In comparison with RSR and MFFS, this method has the following advantages first feature selected by RMFFS can approximately represent all features of the original dataset, and also they have low redundancy that is a result of imposing a combination of L_1-norm and L_2-norm as regularization. In [34] an unsupervised feature selection is based on self-representation. The algorithm preserves the local similarity, and the manifold of data is the main properties of this method. A coefficient matrix is constructed using an $L_{2,1/2}-$norm and the subset of feature is selected by this matrix. An unsupervised graph-preserving feature selection was studied [10]. Local and global correlation among features is considered in this study. In this study, LLE is embedded to achieve a promising result for classification application. Similarity information has an essential role in many feature selection methods. In [17], the similarity is used to construct a matrix graph, and it makes the algorithm to be more reliable against the noises and outliers. They used $L_{2,1}-$norm in AGUFS to select a more acceptable subset of features. An unsupervised feature selection based on graph regularization and local linear embedding was proposed in [25]. They focused on preserving the local manifold of data in selected subspace. This method uses a matrix in which the neighborhood's relationship among data is preserved. GLLE implies the L_1-norm for eliminating the negative effect of noise on outlier data samples.

In [31] an unsupervised feature selection approach by applying dictionary learning idea in a low-rank representation is introduced, named DLUF. Low-rank dictionary learning not only provides a new data representation but also maintains feature correlation. Then, spectral analysis is employed to preserve sample similarities. Finally, a unified objective function for unsupervised feature selection is proposed in a sparse way by an $L_{2,1}-$norm regularization.

In [41] a sparse learning approach is designed to propose an unsupervised method, called SLSP, which takes both global and local structures of the samples into account, and considers the discriminative information through a clustering approach. Same as before mentioned studies a unified objective function by an L_1-norm regularization is applied for feature selection.

3. Proposed method

This section aims to provide the details of the proposed unsupervised feature selection method. The main purpose of the proposed method is to choose a set of feature subset without requiring the label information. Real-world datasets often have a lot of redundant features and outlier samples. A robust and efficient feature selection algorithm is a method that can indicate the redundant features and reduce the effect of the outliers. Inspired by [45], the objective function of the unsupervised feature selection using self-representation is defined as:

$$OF_1 = \min_w F(X - XW) + \lambda R(W), \tag{1}$$

where $X = \left[x_1, x_2, \ldots, x_n\right] \in \mathbb{R}^{(n\times m)}$ shows data, m and n are the numbers of features and instances, respectively. Moreover, $W \in \mathbb{R}^{(m\times m)}$ is the weight matirx of features, F is the loss function, R(W) shows a regularization term to control the diversity of weights, and the positive coefficient is defined by a constant value called λ. The model in Eq. 1 is a type of regularized self-representation that can be used for unsupervised feature selection approaches. This model replaces the input matrix (X) with the target matrix (Y). This is the main idea of the self-representation principle. In other words, each feature is represented a linear combination of other features, which shows as follows:

$$f_i = \sum_{f=1}^{m} f_i w_{ij} + e_i, \tag{2}$$

wheref_i is the feature vector in X and w_{ij} shows the correlation between f_i and f_j. The input data can be computed as a linear function using the self-representation property:

$$X = XW + E. \tag{3}$$

Note that W represents the weight of features and forces E to small values. Also, Frobenius norm is used to minimize E:

$$\min \|X - XW\|_F^2. \tag{4}$$

Note that the Frobenius norm is sensitive to outliers. To solve this issue the $L_{2,1}$–norm has been proposed to improve the robustness of E in the presence of outliers. Therefore, the objective function of equation (1) can be rewritten as:

$$OF_2 = \min_w \|X - XW\|_{2,1} + \lambda R(W). \tag{5}$$

Let $W = \left[w_1; \ldots; w_i; \ldots; w_m\right]$, where w_i is i_{th} row of W. $\|w_i\|_2$ reflects the importance of the i_{th} feature in representation, therefore, it can be used as the feature weight. And let $R(W) = \|w_j\|_{2,1} = \sum_{j=1}^{m} \|w_j\|_2$ to force the sparsity of weights. Consequently, the weights of features are obtained through optimizing the following equation:

$$W = argmin_w \|X - XW\|_{2,1} + \lambda \|w\|_{2,1}. \tag{6}$$

Using IRLS (Iterative Reweighted Least-Squares) [45] two diagonal weighting matrices are defined,G_L^t and G_R^t. Where $g_{L,i}^t = \frac{1}{2\|x_i - x_i W^t\|}$ and $g_{R,j}^t = \frac{1}{\|w_j^t\|_2}$. W is updated when the following weighted least squares problem is solved:

$$W^{t+1} = argmin_W Q(W|W^t) = argmin_W \{\mathrm{tr}((X - XW)^t G_L^t (X - XW)) + \lambda (W^T G_R^t W). \tag{7}$$

In addition, by transforming presented $L_{2,1} - norm$ in equation (6) to trace equivalent of it can be shown as:

$$W = argmin_w \mathrm{tr}(X - XW)^T G_L^t (X - XW) + \lambda \mathrm{tr}(W^t G_R^t W), \tag{8}$$

where G_L^t and G_R^t are diagonal matrices, both are necessary for transforming to trace form. Next, it will be explained how the feature weight matrix will be updated. As it mentioned the objective function of the problem is shown below:

$$J = \min \|X - XW\|_{2,1} + \lambda \|w\|_{2,1}. \tag{9}$$

As the weight matrix of features, W is the target variable of this problem, therefore, by deriving the target function with respect to W, its update equation can be obtained:

$$\frac{\partial J}{\partial W} = 0. \tag{10}$$

Before solving the above problem, it is necessary to raise some points about the derivation of $L_{2,1}-$norm:

Table 1: Summary of Literature Review.

Method	Supervised/Unsupervised	Year	Filter/ Wrapper	Feature /sample similarity	Matrix factorization	Innovation
IG	S	2006	-	No	No	Elements of information theory
Fisher Score	S	2000	-	No	No	Pattern classification
ILFS	S	2021	-	No	No	pays more attention to the local structure of data
Pearson correlation coefficient	S	1998	-	No	No	Thirteen ways to look at the correlation coefficient
MFFS/ KMFFS	U	2015	Filter	No	Yes	Subspace learning
DSRMR	U	2018	Filter	Yes	Yes	Dual Self-representation
MCFS	U	2010	Filter	No	No	Multi-cluster structure, manifold learning, Spectral clustering
FSSEM	U	2004	Wrapper	No	No	Expectation-Maximization (EM)
MRFS	S/U	2010	Filter	Yes	No	Sparse multi-output regression
SCFS	U	2020	Filter	Yes	No	Subspace learning, Similarity matrix
UDFS	U	2011	Filter	No	No	Discriminative analysis and $L_{2,1}-$norm
Constraint Score	S	2008	-	No	No	Pairwise constraints
RSR	U	2015	Filter	Yes	No	Self-representation
Laplacian Score(LS)	U	2005	Filter	Yes	No	-
Trace Ratio		2008	Filter	No	No	Subset-level score(calculates the score of entire subset)
RUFS	U	2013	Filter	No	Yes	Pseudo cluster
JELSR	U	2014	Filter	Yes	No	Embedding learning , spare regression
DISR	U	2017	Embedded	Yes	No	Diversity-induced Self-representation
Non-convex RSR	U	2017	Filter	yes	No	Self-representation,$L_2, p-$norm
GSR-SFS	U	2017	Filter	Yes	No	Self-representation, Subspace learning, self-representation
LRSL	U	2017	filter	Yes	Yes	Low-rank-approximation, structure learning
SR-FS	U	2017	Filter	yes	No	Self-representation
NSCR	U	2015	Wrapper	Yes	No	Nonnegative spectral clustering
EUFS	U	2015	Wrapper	Yes	Yes	-
GLSPFS	S/U	2014	Filter	Yes	No	Global and local structure
GLoSS	U	2016	Filter	No	No	Subspace learning
RUFSM	U	2017	Filter	No	Yes	Feature selection + clustering
SSC	U	2013	Filter	No	no	Sparse representation, self-expressiveness
UFSRL	U	2019	Filter	No	no	Self-representation, sparse reconstruction
RUFS2	U	2020	Filter	No	no	Graph matrix learning, low-dimensional space learning
AGUFS	U	2021	Filter	yes	No	Uncorrelated constraints, local structure learning
JLSPRM	U	2020	Filter	No	No	utilizing nonnegative spectral analysis to learn the cluster labels
DLUF	U	2022	Filter	Yes	No	Low-rank representation, sparse learning, subspace leanring, dictionary learning
SLSP	U	2020	Filter	Yes	No	Takes both global and local structures of the samples into account

Note 1: $L_{2,1} - norm$ of a matrix can be written as follow:

$$\|A\|_{2,1} = \mathrm{tr}(ADA^T)\ ,\ D = GG^T\ ,\ g_{ii} = \frac{1}{\|A_{:,i}\|_2}. \tag{11}$$

Based on this point, the above objective function Eq.(9) is rewritten as below:

$$\begin{aligned} j &= \min \|X - XW\|_{2,1} + \lambda \|W\|_{2,1} = F_1 + F_2 \\ &= \mathrm{tr}((X - XW)G_L(X - XW)^T) + \lambda \mathrm{tr}(WG_RW^T). \end{aligned} \tag{12}$$

Now we need to get the derivative of this function with respect to W. Before, it is necessary to provide some points about the derivative of the trace function.

Note 2:

$$\frac{\partial \mathrm{tr}(XAX^T)}{\partial X} = XA^T + XA. \tag{13}$$

Therefore, the derivative of the second part of the objective function will be as:

$$\frac{\partial F_2}{\partial W} = \frac{\partial \lambda \mathrm{tr}(WG_RW^T)}{\partial W} = WG_R^T + WG_R. \tag{14}$$

Now we rewrite the first part of the objective function (F_1):

$$\begin{aligned} F_1 &= \mathrm{tr}\big((X - XW)G_L(X - XW)^T\big) \\ &= \mathrm{tr}\big(G_L((X - XW)(X - XW)^T\big) \\ &= \mathrm{tr}\big(G_L(XX^T - X(XW)^T - (XW)X^T + (XW)(XW)^T)\big) \\ &= \mathrm{tr}\big(G_L(XX^T - XW^TX^T - XWX^T + XWW^TX^T)\big) \\ &= \mathrm{tr}(G_LXX^T - G_LXW^TX^T - G_LXWX^T + G_LXWW^TX^T) \\ &= \mathrm{tr}(G_LXX^T) - \mathrm{tr}(G_LXW^TX^T) - \mathrm{tr}(G_LXWX^T) + \mathrm{tr}(G_LXWW^TX^T). \end{aligned} \tag{15}$$

We get the derivative of the first part with respect to the variable W and we obtain:

$$\frac{\partial F_1}{\partial W} = \frac{\partial \mathrm{tr}(G_LXX^T)}{\partial W} - \frac{\partial \mathrm{tr}(G_LXW^TX^T)}{\partial W} - \frac{\partial \mathrm{tr}(G_LXWX^T)}{\partial W} + \frac{\partial \mathrm{tr}(G_LXWW^TX^T)}{\partial W}. \tag{16}$$

By calculating the derivative of each of these terms separately we obtain the derivative of the first term is zero:

$$\frac{\partial \mathrm{tr}(G_LXX^T)}{\partial W} = 0. \tag{17}$$

We should state the following point for obtaining the derivative of the second term of Eq. (16):

Note 3:

$$\frac{\partial \mathrm{tr}(AX^TB)}{\partial X} = BA. \tag{18}$$

Now, based on this note and considering $A = G_LX$ and $B = X^T$ with we obtain the derivative of the second term of Eq. (16) as follows:

$$\frac{\partial \mathrm{tr}(AW^TB)}{\partial W} = BA \rightarrow \frac{\partial \mathrm{tr}(G_LXW^TX^T)}{\partial W} = X^TG_LX. \tag{19}$$

Now we get the derivative of the third term. It is necessary to mention the following point:

Note 4:

$$\frac{\partial \mathrm{tr}(AXB)}{\partial X} = A^TB^T. \tag{20}$$

Therefor we obtain the derivative of the third term of Eq. (16) with replacements $A = G_LX$ and $B = X^T$ as:

$$\frac{\partial \mathrm{tr}(AWB)}{\partial W} = A^TB^T \rightarrow \frac{\partial \mathrm{tr}(G_LXWX^T)}{\partial W} = (G^LX)^T(X^T)^T = X^TG_L^TX. \tag{21}$$

Due to the diagonality of G_1, we have $G_1 = G_1^T$, then the above equation can be written as:

$$\frac{\partial \mathrm{tr}(G_L X W X^T)}{\partial W} = X^T G_L X. \tag{22}$$

Note 5:

$$\frac{\partial \mathrm{tr}(AXX^TB)}{\partial X} = A^TB^TX + BAX. \tag{23}$$

So we get the derivative of the fourth expression of Eq. (16) using last note and assigning A with $G_L X$ and $B = X^T$ as:

$$\begin{aligned} \frac{\partial \mathrm{tr}(AWW^TB)}{\partial W} &= A^TB^TW + BAW \\ \rightarrow \frac{\partial \mathrm{tr}(G_L X W W^T X^T)}{\partial W} &= X^T G_L^T X W + X^T G_L X W. \end{aligned} \tag{24}$$

Now, based on Eq. (15) to Eq. (24) , the derivative of F_1 with respect to W will be as follows:

$$\begin{aligned} \frac{\partial F_1}{\partial W} &= \frac{\mathrm{tr}(G_L X X^T)}{\partial W} - \frac{\partial \mathrm{tr}(G_L X W^T X^T)}{\partial W} - \frac{\partial \mathrm{tr}(G_L X W X^T)}{\partial W} + \frac{\partial \mathrm{tr}(G_L X W W^T X^T)}{\partial W} \\ &= (0) - (X^T G_L X) - (X^T G_L X) + (X^T G_L^T X W + X^T G_L X W) \end{aligned} \tag{25}$$

Finally, based on the above equations, the derivative of the objective function will be as follows:

$$\begin{aligned} J &= \min \|X - XW\|_{2,1} + \lambda \|W\|_{2,1} = F_1 + F_2 \\ &= \mathrm{tr}\big((X - XW) G_L (X - XW)^T\big) + \lambda \mathrm{tr}(W G_R W^T), \end{aligned} \tag{26}$$

and since

$$\begin{aligned} \frac{\partial J}{\partial W} &= \frac{\partial F_1}{\partial W} + \frac{\partial F_2}{\partial W} \\ &= (-X^T G_L X - X^T G_L X + X^T G_L^T X W + X^T G_L X W) + \lambda (W G_R^T + W G_R), \end{aligned} \tag{27}$$

so

$$\frac{\partial J}{\partial W} = -2X^T G_L X + X^T G_L^T X W + X^T G_L X W + \lambda W G_R^T + \lambda W G_R. \tag{28}$$

To optimize W, we get the root of the above equation, then:

$$\frac{\partial J}{\partial W} = -2X^T G_L X + X^T G_L^T X W + X^T G_L X W + \lambda W G_R^T + \lambda W G_R = 0, \tag{29}$$

and we imply that

$$X^T G_L^T X W + X^T G_L X W + \lambda W G_R^T + \lambda W G_R = 2X^T G_L X. \tag{30}$$

We multiply both sides of the equation by G_R^{-1} so that W can be decomposed:

$$(G_R^{-1} X^T G_L^T X W + G_R^{-1} X^T G_L X W + \lambda G_R^{-1} W G_R^T + \lambda G_R^{-1} W G_R) = 2 G_R^{-1} X^T G_L X. \tag{31}$$

Considering that G_R is a diagonal matrix and its dimensions are the same as W, therefore the following equation can be changed as:

$$\lambda G_R^{-1} W G_R = \lambda G_R^{-1} G_R W = \lambda I W. \tag{32}$$

Therefore, the Eq (31) is rewritten as follows:

$$G_R^{-1} X^T G_L^T X W + G_R^{-1} X^T G_L X W + 2\lambda I W = 2 G_R^{-1} X^T G_L X. \tag{33}$$

By decomposing W and considering that $G_L^T = G_L$, the equation is rewritten as follows:

$$2(G_R^{-1}X^TG_LX + \lambda I)W = 2G_R^{-1}X^TG_LX. \tag{34}$$

Therefore, the update equation of W will be as follows:

$$W = (G_R^{-1}X^TG_LX + \lambda I)^{-1}G_R^{-1}X^TG_LX, \tag{35}$$

where $I \in \mathbb{R}^{n\times n}$.
In this updating rule, the weights are initialized with equal and small weigh values ($w_{ij} = \epsilon$) at the first iterations. To improve the convergence rate we proposed an intelligent initialization mechanism for the weight vector. Additional details are described in subsequent sections.

3.1. Initialization of weight vectors

In this paper, to enhance the convergence property of self-representation based on feature selection methods, a new method called (SSRSR) is proposed to initialize the weight matrix W. To this end, first suppose that each $f_i \in \mathbb{R}^{(1\times n)}$ is the i_{th} feature. In our method, the feature space is first mapped to a low-dimension space called similarity space. To this aim, the Pearson correlation coefficient (PCC) is used to obtain the correlation between features as:

$$PCC_{ij} = \frac{cov(F_i, F_j)}{\sigma_{F_i}\sigma_{F_j}}, \tag{36}$$

where $cov(F_i, F_j)$ function calculates the covariance value between the corresponding features and $\sigma(F_i)$ shows the standard deviation of F_i.Since each feature f_i can be demonstrated as a linear combination of other vectors in the similarity space, the subspace coding has been used in this step as [8]:

$$sim(i, j) = \frac{cov(f_i, f_j)}{\sigma_{f_i}\sigma_{f_j}}. \tag{37}$$

Based on the subspace mapping theory [8], high-dimensional data points are located in the low-dimensional subspaces [5]. The similarity value of each feature is a linear form of the other features as:

$$wf_i = \sum_{j=1,\dots,n,\ j\neq i} \gamma_{i,j} \cdot wf_i \tag{38}$$

where the similarity weights between features denote by γ_{ij}. These concepts are used to form the optimization problem with aims to lead to least square errors as follows:

$$\gamma_i^* = argmin_{\gamma_i}\|wf_i + \overline{wf}\|_2^2 + \lambda\|\gamma_i\|_1, \tag{39}$$

where $(\overline{wf}) = wf\backslash wf_i$. Also, L_1 regularization is used to satisfy the sparseness of solutions. The Eq. (39) is convex; to solve this type of optimization problem. There are several convex optimization tools [3]. In this paper, ADMM method [2] is used to obtain the sparse representation of each data point in the feature space. Then the similarity weight matrix W is determined as:

$$w_{ij} = \frac{\gamma_{ij} + \gamma_{ji}}{2}. \tag{40}$$

Instead of equal values, we can use the weight values obtained from Eq. (40) to improve the convergence rate of the unsupervised feature selection method. The pseudo-code of the proposed method is given as follows:

3.2. Time complexity

In the proposed method we need to update W in each iteration, whose computational complexity is $O(m^3 + m^3n)$, where n and m are the number of instances features and samples, respectively. The time complexity of the proposed method is $O(T(m^3 + m^3n))$, where T is the total number of iterations.

Algorithm 1 Pseudo-code of the proposed method

Input:
X: m × n matrix
T: Number of iterations

Output:
W: m × m matrix

begin algorithm
using Eq. 14

$$W_{ij}^{0} = \frac{\gamma_{ij} + \gamma ji}{2} \qquad ,\forall i, j = 1, ..., m$$

for $t \leftarrow 1$ to T **do**

$$W^{t+1} = ((G_R^t)^{-1} X^T G_L^t X + \lambda I)^{-}1 (G_R^t)^{-1} X^T G_L^t X$$

end for
return W
end algorithm

Figure 1: Pseudocode of the proposed method

4. Experiment and results

4.1. Datasets

The proposed method runs over the IRIS dataset with 150 samples and four features. Due to the few feature numbers of this dataset and to see the properties of the proposed method we add seven features to the IRIS dataset. To this end, we add some relevant, irrelevant, and repeated features. We first run the proposed method on a fake IRIS-like dataset with 11 features that features 1_{st} to 4_{th} were copied to features 5_{th} to 8_{th}, feature 9_{th} was completely related with the class label, and features 10_{th} and 11_{th} were randomly filled and completely irrelevant. The results show that this algorithm does not tend to select the 9_{th} feature and select the other important feature, while the first selected feature with the old version of this algorithm is the 9_{th} feature. This means that the proposed method rejects the class-like features. The old version of the algorithm focuses on the 10_{th} and 11_{th} features with random values, but the new version solves this problem. The proposed method was implemented on three data sets: Wine, Sonar and Ionosphere.

4.2. Evaluation measures

In this paper, for evaluating the performance of the proposed method, three standard measures, including F-measure, precision, and recall, are used. To this end, each method's selected features are givenUniFeat package in JAVA, and the outputs are saved in a file and then plotted. The graphical results are shown in a subsequent section.

4.3. Results and discussion

Figures 2 to 7 show the result of F-measure, precision, and recall on the obove-mentioned dataset. SSRSR is compared with RSR and some supervised and unsupervised algorithms in terms of the above measures. The result of implementing the proposed method on these datasest is illustrated in Figures 2 to 7. The vertical axis shows the feature selection performance value in all these figures, and the horizontal axis represents the number of selected features. In all experiments including plots and tables, the average value on ten independent runs for each algorithm is reported. Moreover, 70% of each dataset is randomly selected to train the algorithm, and the remaining 30% is used to test the method.
In this subsection, we have drawn tables 2-7 which show the comparison results of the supervised (tables 2-4) and unsupervised feature selection methods (tables 4-7) in terms of F-measure, Precision, and Recall evaluation measures, respectively. The best results are boldfaced in all tables. Also, the last row of the tables shows the average of the obtained results of the methods.

Table 2 shows the F-measure values of the proposed method compared to the four well-known supervised feature selection methods. It can be seen from the results that the SSRSR attained the highest F-measure value compared

to the other methods. This can be shown especially for a bigger number of features. Tables 3 and 4 report the Precision and Recall values obtained by the feature selection methods. It is clear based on these tables; our method has achieved acceptable results compared to the other methods.

Table 2: The F-measure results of the supervised feature selection methods with different number of features for Wine dataset.

k	SSRSR	FS	MRMr	LS	IG
2	0.7	0.81	0.8	0.72	**0.925**
4	**0.92**	0.875	0.86	0.89	0.908
6	0.925	0.924	0.895	0.914	**0.93**
8	**0949**	0.939	0.93	0.918	0.93
10	**0949**	0.93	0.918	0.918	0.93
12	**094**	0.912	0.9	0.918	0.935
Avg.	0.903	0.898	0.883	0.746	**0936**

Table 3: The Precision results of the supervised feature selection methods with different number of featuresfor Wine dataset.

k	SSRSR	FS	MRMr	LS	IG
2	0.777	0.839	0.819	0.75	**0.865**
4	0.93	0.9	0.85	0.88	**0.945**
6	**0.94**	0.92	0.89	0.92	0.92
8	0.949	0.92	0.9	**0.949**	0.92
10	**0942**	0.92	0.92	0.94	0.92
12	**0941**	0.9	0.929	0.931	0.92
Avg.	**0.912**	0.899	0.834	0.895	0.911

Table 4: The Recall results of the supervised feature selection methods with different number of features for Wine dataset.

k	SSRSR	FS	MRMr	LS	IG
2	0.738	0.82	0.852	0.738	**0.944**
4	0.922	0.894	0.86	0.875	**0.948**
6	0.93	0.91	0.915	0.911	**0.95**
8	**0.935**	0.91	0.902	0.941	0.932
10	**0.935**	0.91	0.924	0.934	0.923
12	**0.935**	0.9	0.915	0.921	0.92
Avg.	0.899	0.89	0.894	0.886	**0.936**

Tables 5-7 show the results of the unsupervised feature selection methods in terms of F-measure, Precision, and Recall criteria, respectively. It can be seen that the overall performance of the proposed method is much better than other methods. Also, the results show that the proposed method obtained the best results. It should be noted that the proposed method works better than other methods in more features and on average.

Table 5: The F-measure results of the unsupervised feature selection methods with different number of features for Wine dataset.

k	SSRSR	RSR	RRFS	UFSACO
2	0.742	0.645	0.659	**0.808**
4	**0.911**	0.67	0.874	0.802
6	0.91	0.861	0.939	**0.952**
8	0.948	0.94	0.901	**0.952**
10	**0.949**	0.937	0.901	0.934
12	**0.945**	0.937	0.917	0.917
Avg.	**0.908**	0.831	0.865	0.894

Table 8 shows the F-measure values of the proposed method compared to the four well-known supervised feature selection methods. It can be seen from the results that the SSRSR attained the highest F-measure value compared to the other methods. This can be shown especially for more number of features. Tables 9 and 10 report the Precision and Recall values obtained by the feature selection methods. It is clear based on these tables; our method has achieved acceptable results compared to the other methods.

Table 6: The Precision results of the unsupervised feature selection methods with different number of features for Wine dataset.

k	SSRSR	RSR	RRFS	UFSACO
2	0.773	0.611	0.725	**0.817**
4	**0.922**	0.708	0.86	0.889
6	0.938	0.881	**0.974**	0.885
8	0.939	**0.942**	0.931	0.93
10	**0.939**	0.935	0.936	0.923
12	**0.939**	**0.939**	0.936	0.923
Avg.	**0.908**	0.836	0.893	0.894

Table 7: The Recall results of the unsupervised feature selection methods with different number of features for Wine dataset.

k	SSRSR	RSR	RRFS	UFSACO
2	0.738	0.611	0.7	**0.839**
4	**0.921**	0.692	0.871	0.901
6	0.937	0.884	**0.95**	**0.95**
8	0.941	**0.952**	0.923	0.922
10	**0.942**	0.937	0.923	0.93
12	**0.942**	**0.942**	0.923	0.923
Avg.	0.903	0.836	0.881	**0.91**

Table 8: The F-measure results of the supervised feature selection methods with different number of features for Sonar dataset.

k	SSRSR	FS	MRMr	LS	IG
8	0.535	0.672	0.706	**0.731**	0.672
16	0.635	0.64.	0.659	**0.733**	0.645
24	0.622	0.696	0.64	**0.735**	0.696
32	0.711	**0.764**	0.638	0.692	**0.764**
40	0.697	**0.758**	0.646	0.692	0.754
48	0.749	**0.751**	0.689	0.692	0.749
56	0.692	0.692	0.692	0.692	0.692
Avg.	0.663	**0.710**	0.667	0.709	**0.710**

Table 9: The Precision results of the supervised feature selection methods with different number of features for Sonar dataset.

k	SSRSR	FS	MRMr	LS	IG
8	0.653	0.672	0.716	**0.732**	0.672
16	0.635	0.643.	**0.659**	**0.659**	0.648
24	0.634	**0.697**	0.643	0.643	**0.697**
32	0.739	**0.764**	0.639	0.639	**0.764**
40	0.747	0.746	0.646	0.646	**0.755**
48	**0.751**	**0.751**	0.694	0.694	**0.751**
56	0.692	0.692	0.692	0.692	0.692
Avg.	0.693	0.709	0.669	0.672	**0.711**

Table 10: The Recall results of the supervised feature selection methods with different number of features for Sonar dataset.

k	SSRSR	FS	MRMr	LS	IG
8	0.558	0.673	0.707	**0.731**	0.673
16	0.635	0.639.	**0.659**	**0.736**	0.644
24	0.635	**0.697**	0.639	**0.736**	0.697
32	0.721	**0.764**	0.639	0.692	**0.764**
40	0.697	0.745	0.646	0.692	**0.755**
48	0.750	**0.751**	0.692	0.692	0.750
56	0.692	0.692	06929	0.692	0.692
Avg.	0.669	0.708	0667.	**0.710**	**0.710**

Table 11: The F-measure results of the unsupervised feature selection methods with different number of features for Sonar dataset.

k	SSRSR	RSR	RRFS	UFSACO
8	0.535	**0.717**	0.642	0.654
16	0.635	0.672	0.657	**0.673**
24	0.622	0.709	**0.731**	0.685
32	0.711	0.721	0.673	**0.726**
40	0.697	0.702	0.707	**0.721**
48	**0.749**	0.687	0.706	0.701
56	0.692	0.692	0.692	0.692
Avg.	0.663	**0.700**	0686.	0.693

Table 12: The Precision results of the unsupervised feature selection methods with different number of features for Sonar dataset.

k	SSRSR	RSR	RRFS	UFSACO
8	0.653	**0.716**	0.642	0.654
16	0.635	**0.673**	0.658	**0.673**
24	0.634	0.712	**0.731**	0.688
32	**0.739**	0.721	0.673	0.726
40	**0.697**	0.702	0.707	0.721
48	**0.751**	0.688	0.707	0.702
56	0.692	0.692	0.692	0.692
Avg.	0.685	**0.702**	0.687	0.693

Table 13: The Recall results of the unsupervised feature selection methods with different number of features for Sonar dataset.

k	SSRSR	RSR	RRFS	UFSACO
8	0.653	**0.719**	0.642	0.653
16	0.635	0.672	0.659	**0.673**
24	0.634	0.712	**0.731**	0.687
32	**0.739**	0.728	0.673	0.726
40	**0.725**	0.706	0.707	0.721
48	**0.751**	0.687	0.707	0.701
56	0.692	0.692	0.692	0.692
Avg.	0.689	0.836	0.687	**0.693**

Table 14 shows the F-measure values of the proposed method compared to the four well-known supervised feature selection methods. It can be seen from the results that the SSRSR attained the highest F-measure value compared to the other methods. This can be shown especially for more number of features. Tables 15 and 16 report the Precision and Recall values obtained by the feature selection methods. It is clear based on these tables; our method has achieved acceptable results compared to the other methods.

Table 14: The F-measure results of the supervised feature selection methods with different number of features for Ionosphere dataset.

k	SSRSR	FS	MRMr	LS	IG
5	0.846	**0.906**	0.876	0.900	0.867
10	0.858	**0.887**	0.866	0.880	0.858
15	0.858	0.884	0.876	**0.896**	0.862
20	0.864	0.849	0.849	**0.899**	0.862
25	0.865	0.849	0.849	**0.866**	0.864
30	**0.869**	0.849	0.849	0.866	0.842
Avg.	0.860	0.875	0.860	**0.884**	0.859

4.4. Comparison to supervised algorithms

In another point of view, the methods are compared based on a different number of features, and the results are reported in Figs. 2 and 3. The vertical axis shows the feature selection performance value in all these figures, and the horizontal axis represents the number of selected features. Figs. 2 and 3 report F-measure, Precision, and Recall criteria of the supervised methods, respectively. All these results are summarized over ten independent

Table 15: The Precision results of the supervised feature selection methods with different number of features for Ionosphere dataset.

k	SSRSR	FS	MRMr	LS	IG
5	0.849	**0.917**	0.877	0.910	0.868
10	0.860	**0.889**	0.869	0.887	0.860
15	0.866	0.886	0.877	**0.898**	0.862
20	0.866	0.880	0.851	**0.901**	0.862
25	0.861	0.851	0.851	**0.869**	0.865
30	**0.871**	0.851	0.849	0.869	0.846
Avg.	0.862	0.879	0.862	**0.889**	0.860

Table 16: The Recall results of the supervised feature selection methods with different number of features for Ionosphere dataset.

k	SSRSR	FS	MRMr	LS	IG
5	0.849	**0.909**	0.877	0.903	0.869
10	0.860	**0.889**	0.860	0.883	0.860
15	0.860	0.866	0.877	**0.897**	0.863
20	0.897	0.880	0.852	**0.900**	0.863
25	0.849	0.852	0.852	0.860	**0.866**
30	0.852	0.852	0.852	**0.869**	0.846
Avg.	0.861	0.874	0.861	**0.885**	0.861

Table 17: The F-measure results of the unsupervised feature selection methods with different number of features for Ionosphere dataset.

k	SSRSR	RSR	RRFS	UFSACO
5	0.846	**0.887**	0.854	0.82
10	**0.858**	**0.858**	0.85	0.832
15	0.858	0.849	0.85	**0.881**
20	0.864	0.849	0.85	**0.876**
25	**0.856**	0.849	0.85	0.855
30	0.849	0.687	**0.85**	0.846
Avg.	0.855	**0.856**	0.850	0.851

Table 18: The Precision results of the unsupervised feature selection methods with different number of features for Ionosphere dataset.

k	SSRSR	RSR	RRFS	UFSACO
5	0.846	**0.889**	0.854	0.828
10	**0.860**	0.858	0.85	0.833
15	0.858	0.851	0.85	**0.883**
20	0.864	0.852	0.85	**0.877**
25	0.846	0.851	0.85	**0.857**
30	0.849	**0.851**	0.85	0.848
Avg.	0.853	**0.858**	0.850	0.854

Table 19: The Recall results of the unsupervised feature selection methods with different number of features for Ionosphere dataset.

k	SSRSR	RSR	RRFS	UFSACO
5	0.846	**0.889**	0.855	0.826
10	**0.858**	**0.86**	0.852	0.835
15	0.858	0.852	0.852	**0.883**
20	0.864	0.852	0.852	**0.877**
25	0.852	0.852	0.852	**0.858**
30	0.849	**0.852**	**0.852**	0.849
Avg.	0.854	**0.859**	0.852	0.854

runs. According to these figures, it is clear that the proposed method achieved the highest values in terms of the evaluation measures. It has been able to improve performance much faster than other methods. Thus, the predictive power of selected features will be significantly increased when a higher number of features are selected. These claims are illustrated in Figures. We can see the good performance of SSRSR against other methods. Figure

2 illustrates the above metrics of SSRSR and well-known supervised feature selection algorithms Information Gain (IG), Fisher Score (FS), MRMr, and Laplacian Score (LS). SSRSR is an unsupervised method, in some points of these charts, SSRSR is more effective than supervised methods.

4.5. Comparison to unsupervised algorithms

F-measure, precision, and recall statistics of SSRSR compared to the old version of this algorithm (RSR) and several well-known unsupervised feature selection methods that are shown in Figure 3 respectively. Finally the proposed method is compared with RSR and two other well-known unsupervised algorithms (RRFS & UFSACO) [37]. The performance of SSRSR is better than others, relatively.

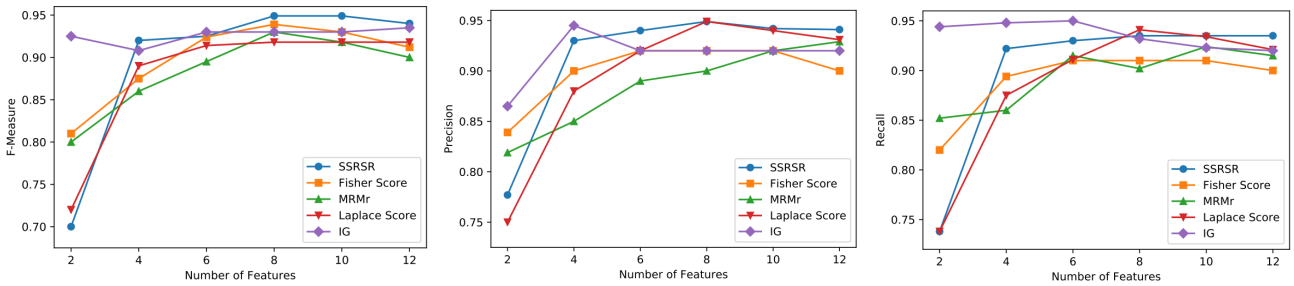

Figure 2: The comparison of wine dataset values of the proposed method and several supervised feature selection methods with the various number of features.

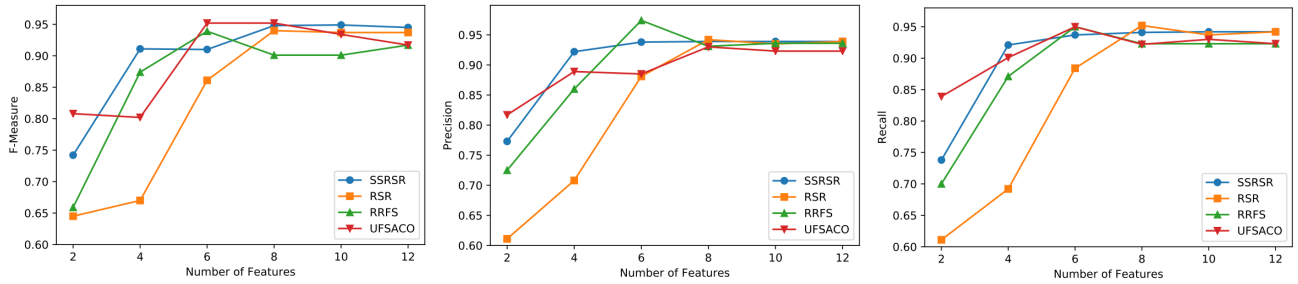

Figure 3: The comparison of wine dataset values of the proposed method and some unsupervised feature selection methods with the various numbers of a feature.

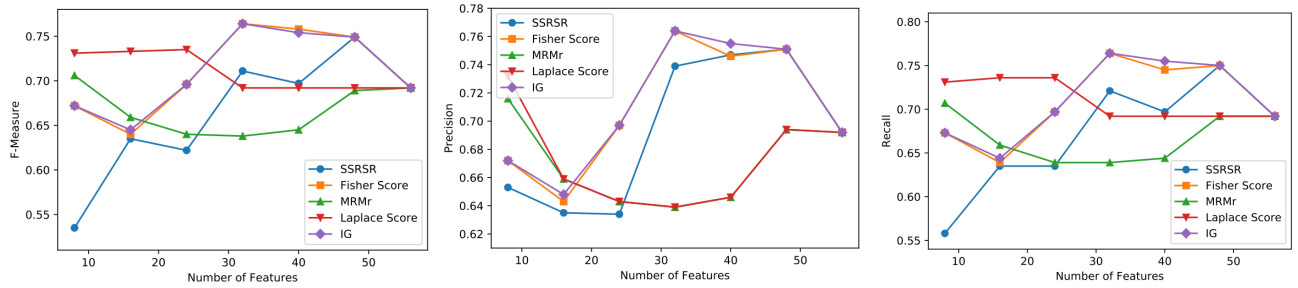

Figure 4: The comparison of Sonar dataset values of the proposed method and some supervised feature selection methods with the various numbers of a feature.

Based on the tests whose results are displayed in the graphs, the advantage of the proposed method over other methods used in the comparisons can be stated as follows:

- For all three criteria in all graphs, the efficiency of the proposed algorithm is improving by increasing the number of selected features.
- In some cases, the efficiency of the algorithm is more than other methods.

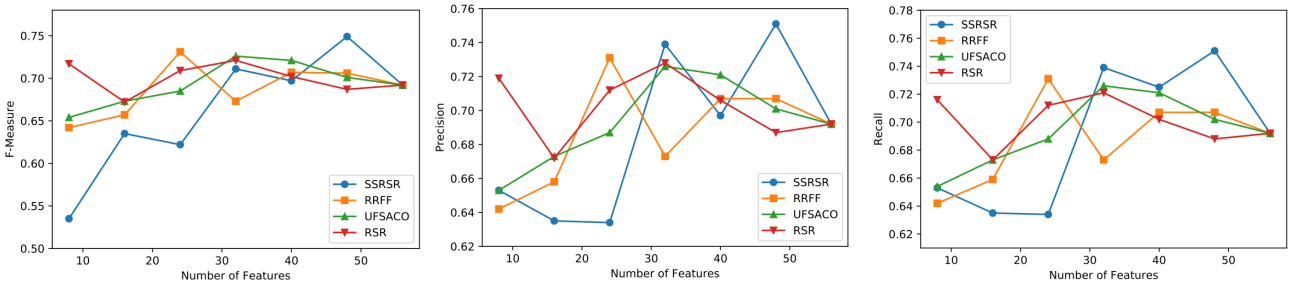

Figure 5: The comparison of Sonar dataset values of the proposed method and some unsupervised feature selection methods with the various numbers of a feature.

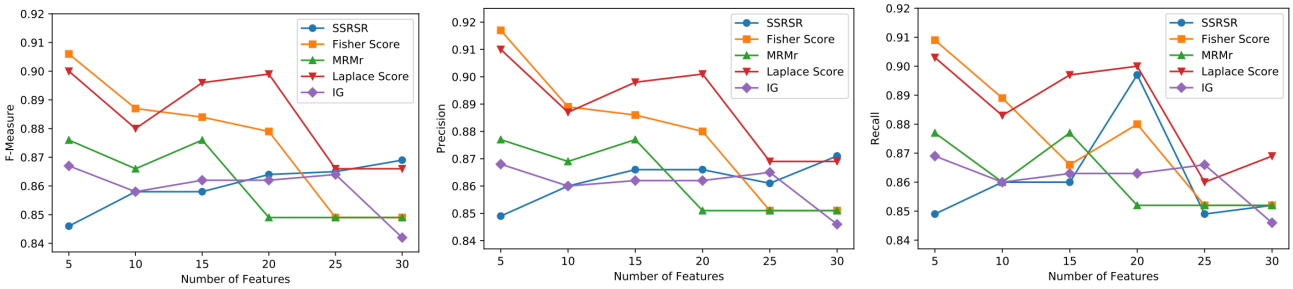

Figure 6: The comparison of Ionosphere dataset values of the proposed method and some supervised feature selection methods with the various numbers of a feature.

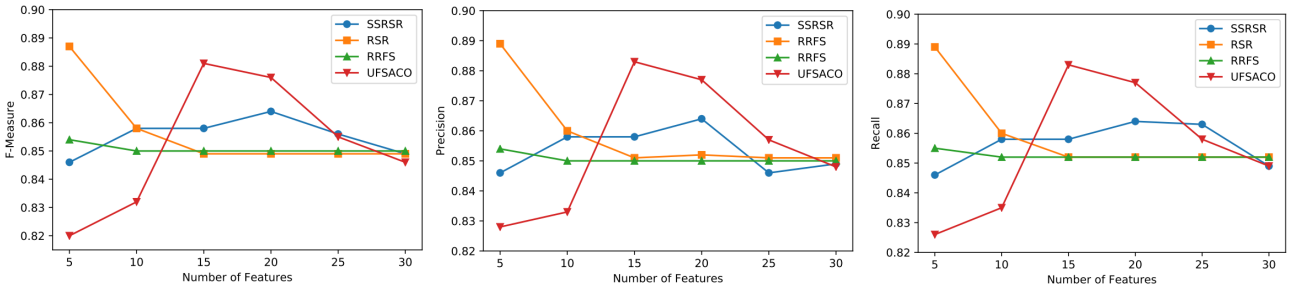

Figure 7: The comparison of Ionosphere dataset values of the proposed method and some unsupervised feature selection methods with the various numbers of a feature.

5. Conclusions

In this work, an effective unsupervised feature selection method using the self-representation model for and Non-negative Matrix Factorization (NMF) framework is proposed. The proposed method uses the inherent information among features such as similarity properties to select the most effective feature. Results show that the convergence speed of SSRSR is more acceptable rather than unsupervised (e.g. RSR, UFSACO& RRFS) and even the supervised methods (such as IG, Fisher Score, MRMr & Laplacian Score). In other words, using the intrinsic information hidden among features results in a better subset of original features. There are several ideas to improve our work. The first one is using some kernel methods embedded into the process of NMF objective function to capture the nonlinear relation between features. Another one is hybridizing the current work with self-paced methods. This is for differentiating between simple and hard samples through the optimization process.

6. Acknowledgement

We would like to thank Dr. Shahrokh Esmaeili, Associate Professor, Department of Mathematics, University of Kurdistan, for his helpful advice in refining the mathematic equations.

References

[1] S. Bahassine, A. Madani, M. Al-Sarem, and M. Kissi, *Feature selection using an improved Chi-square for arabic text classification*, J. King Saud Univ. - Comput. Inf. Sci., 32 (2020), pp. 225–231.

[2] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, *Distributed optimization and statistical learning via the alternating direction method of multipliers*, Foundations and Trends in Machine Learning, 3 (2011), pp. 1–122.

[3] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge university press.

[4] D. Cai, C. Zhang, and X. He, *Unsupervised feature selection for multi-cluster data*, in Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, New York, NY, USA, 2010, Association for Computing Machinery, pp. 333—342.

[5] Y. Chen, G. Li, and Y. Gu, *Active orthogonal matching pursuit for sparse subspace clustering*, IEEE Signal Processing Letters, 25 (2018), pp. 164–168.

[6] F. Dornaika, *Multi-layer manifold learning with feature selection*, Applied Intelligence, 50 (2020), pp. 1859–1871.

[7] S. Du, Y. Ma, S. Li, and Y. Ma, *Robust unsupervised feature selection via matrix factorization*, Neurocomputing, 241 (2017), pp. 115–127.

[8] E. Elhamifar and R. Vidal, *Sparse subspace clustering: Algorithm, theory, and applications*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 35 (2013), pp. 2765–2781.

[9] M. Gan and L. Zhang, *Iteratively local fisher score for feature selection*, Applied Intelligence, 51 (2021), pp. 6167–6181.

[10] X. Han, P. Liu, L. Wang, and D. Li, *Unsupervised feature selection via graph matrix learning and the low-dimensional space learning for classification*, Eng. Appl. Artif. Intell., 87 (2020), p. 103283.

[11] K. Hanbay, *A new standard error based artificial bee colony algorithm and its applications in feature selection*, J. King Saud Univ. - Comput. Inf. Sci., 34 (2022), pp. 4554–4567.

[12] X. He, D. Cai, and P. Niyogi, *Laplacian score for feature selection*, in Advances in neural information processing systems, vol. 18, 2005, pp. 507–514.

[13] A. E. Hegazy, M. Makhlouf, and G. S. El-Tawel, *Improved salp swarm algorithm for feature selection*, J. King Saud Univ. - Comput. Inf. Sci., 32 (2020), pp. 335–344.

[14] H. Hichem, M. Elkamel, M. Rafik, M. T. Mesaaoud, and C. Ouahiba, *A new binary grasshopper optimization algorithm for feature selection problem*, J. King Saud Univ. - Comput. Inf. Sci., 34 (2022), pp. 316–328.

[15] C. Hou, F. Nie, X. Li, D. Yi, and Y. Wu, *Joint embedding learning and sparse regression: A framework for unsupervised feature selection*, IEEE Transactions on Cybernetics, 44 (2014), pp. 793–804.

[16] R. Hu, X. Zhu, D. Cheng, W. He, Y. Yan, J. Song, and S. Zhang, *Graph self-representation method for unsupervised feature selection*, Neurocomputing, 220 (2017), pp. 130–137. Recent Research in Medical Technology Based on Multimedia and Pattern Recognition.

[17] Y. Huang, Z. Shen, F. Cai, T. Li, and F. Lv, *Adaptive graph-based generalized regression model for unsupervised feature selection*, Knowledge-Based Systems, 227 (2021), p. 107156.

[18] A. E. Isabelle Guyon, *An introduction to variable and feature selection*, J. Mach. Learn. Res., 3 (2003), pp. 1157–1182.

[19] C. E. B. Jennifer G. Dy, *An introduction to variable and feature selection*, J. Mach. Learn. Res., 5 (2004), pp. 845–889.

[20] M. N. K.P. and T. P., *Feature selection using efficient fusion of fisher score and greedy searching for alzheimer's classification*, J. King Saud Univ. - Comput. Inf. Sci., 34 (2022), pp. 4993–5006.

[21] S. Larabi Marie-Sainte and N. Alalyani, *Firefly algorithm based feature selection for arabic text classification*, J. King Saud Univ. - Comput. Inf. Sci., 32 (2020), pp. 320–328.

[22] H. Li, Y. Wang, Y. Li, P. Hu, and R. Zhao, *Joint local structure preservation and redundancy minimization for unsupervised feature selection*, Appl. Intell., 50 (2020), pp. 4394–4411.

[23] Z. Li and J. Tang, *Unsupervised feature selection via nonnegative spectral analysis and redundancy control*, IEEE Transactions on Image Processing, 24 (2015), pp. 5343–5355.

[24] J. Miao, Y. Ping, Z. Chen, X.-B. Jin, P. Li, and L. Niu, *Unsupervised feature selection by non-convex regularized self-representation*, Expert Syst. Appl., 173 (2021), p. 114643.

[25] J. Miao, T. Yang, L. Sun, X. Fei, L. Niu, and Y. Shi, *Graph regularized locally linear embedding for unsupervised feature selection*, Pattern Recognition, 122 (2022), p. 108299.

[26] P. Mitra, C. Murthy, and S. Pal, *Unsupervised feature selection using feature similarity*, IEEE Trans. Pattern Anal. Mach. Intell., 24 (2002), pp. 301–312.

[27] P. Moradi and M. Gholampour, *A hybrid particle swarm optimization for feature subset selection by integrating a novel local search strategy*, Applied Soft Computing, 43 (2016), pp. 117–130.

[28] F. Nie, H. Huang, X. Cai, and C. Ding, *Efficient and robust feature selection via joint $\ell_{2,1}$-norms minimization*, in Advances in neural information processing systems, J. Lafferty, C. Williams, J. Shawe-Taylor, R. Zemel, and A. Culotta, eds., vol. 23, Curran Associates, Inc., 2010, pp. 1813–1821.

[29] F. Nie, S. Xiang, Y. Jia, C. Zhang, and S. Yan, *Trace ratio criterion for feature selection.*, in Proceedings of the National Conference on Artificial Intelligence, vol. 2, 01 2008, pp. 671–676.

[30] M. G. Parsa, H. Zare, and M. Ghatee, *Unsupervised feature selection based on adaptive similarity learning and subspace clustering*, Eng. Appl. Artif. Intell., 95 (2020), p. 103855.

[31] ——, *Low-rank dictionary learning for unsupervised feature selection*, Expert Syst. Appl., 202 (2022), p. 117149.

[32] R. B. Pereira, A. Plastino, B. Zadrozny, and L. H. C. Merschmann, *Categorizing feature selection methods for multi-label classification*, Artificial Intelligence Review, 49 (2018), pp. 57–78.

[33] M. Qian and C. Zhai, *Robust unsupervised feature selection*, in Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence, F. Rossi, ed., AAAI Press, 2013, pp. 1621–1627.

[34] R. Shang, J. Chang, L. Jiao, and Y. Xue, *Unsupervised feature selection based on self-representation sparse regression and local similarity preserving*, Int. J. Mach. Learn. & Cyber., 10 (2019), pp. 757–770.

[35] P. D. Sheth, S. T. Patil, and M. L. Dhore, *Evolutionary computing for clinical dataset classification using a novel feature selection algorithm*, J. King Saud Univ. - Comput. Inf. Sci., 34 (2022), pp. 5075–5082.

[36] D. Singh and B. Singh, *Hybridization of feature selection and feature weighting for high dimensional data*, Appl. Intell., 49 (2019), pp. 1580–1596.

[37] S. Tabakhi, P. Moradi, and F. Akhlaghian, *An unsupervised feature selection algorithm based on ant colony optimization*, Eng. Appl. Artif. Intell., 32 (2014), pp. 112–123.

[38] C. Tang, X. Liu, M. Li, P. Wang, J. Chen, L. Wang, and W. Li, *Robust unsupervised feature selection via dual self-representation and manifold regularization*, Knowledge-Based Systems, 145 (2018), pp. 109–120.

[39] S. Wang, W. Pedrycz, Q. Zhu, and W. Zhu, *Subspace learning for unsupervised feature selection via matrix factorization*, Pattern Recognition, 48 (2015), pp. 10–19.

[40] Y. Yang, H. Shen, Z. Ma, Z. Huang, and X. Zhou, *$\ell_{2,1}$-norm regularized discriminative feature selection for unsupervised learning*, in Proceedings of the Twenty-Second international joint conference on Artificial Intelligence, T. Walsh, ed., vol. 2, AAAI Press, 2011, pp. 1589–1594.

[41] H. Zare, M. G. Parsa, M. Ghatee, and S. H. Alizadeh, *Similarity preserving unsupervised feature selection based on sparse learning*, in 2020 10th International Symposium onTelecommunications (IST), 2020, pp. 50–55.

[42] Z. Zhao, L. Wang, and H. Liu, *Efficient spectral feature selection with minimum redundancy*, in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 24, AAAI Press, July 2010, pp. 673–678.

[43] W. Zheng, X. Zhu, G. Wen, Y. Zhu, H. Yu, and J. Gan, *Unsupervised feature selection by self-paced learning regularization*, Pattern Recognition Letters, 132 (2020), pp. 4–11. Multiple-Task Learning for Big Data (MTL4BD).

[44] H. Zhou, X. Wang, and R. Zhu, *Feature selection based on mutual information with correlation coefficient*, Applied Intelligence, 52 (2021), pp. 5457–5474.

[45] P. Zhu, W. Zuo, L. Zhang, Q. Hu, and S. C. Shiu, *Unsupervised feature selection by regularized self-representation*, Pattern Recognition, 48 (2015), pp. 438–446.

[46] Y. Zhu, X. Zhang, R. Wang, W. Zheng, and Y. Zhu, *Self-representation and PCA embedding for unsupervised feature selection*, World Wide Web, 21 (2017), pp. 1675–1688.

Please cite this article using:

Masoud Karimzadeh, Parham Moradi, Abdulbaghi Ghaderzadeh, Unsupervised feature selection by integration of regularized self-representation and sparse coding, AUT J. Math. Comput., 5(2) (2024) 91-109
https://doi.org/10.22060/AJMC.2023.21449.1090