 doi:10.20944/preprints201911.0261.v1

MDPI

Article

Feature selection based on robust LLE vote and its application to bearing fault diagnosis

Haishuang Yin [1,†], Zebiao Hu [2,†] and Yuanhong Liu*[†]

[1] Affiliation 1; yinhaishuang@nepu.edu.cn
[2] Affiliation 2; Huzebiaoya@163.com
* Correspondence: liuyuanhong@nepu.edu.cn
† Current address : School of Information and Electrical Engineering, Northeast Petroleum University, Daqing 163318, PR China.

Version November 20, 2019 submitted to Journal Not Specified

Abstract: The purpose of feature selection is to find important features from the original high-dimensional space. As a typical feature selection algorithm, Locally linear embedding(LLE)-based feature selection algorithm, which applies the idea of LLE to the graph-preserving feature selection framework, has been received wide attention. However, LLE-based feature selection framework is sensitive to noise and K-nearest neighbors. To address these problems, an improved LLE-based feature selection algorithm, robust LLE (RLLE) vote, is proposed. In this algorithm, l_1 and l_2 regularization are introduced into the high-dimensional reconstruction model of LLE. Furthermore, RLLE vote also proposes a criterion to measure the difference between the reconstruction features and the original features, and then the importance features can be selected by this criteria. Extensive experiments are carried out on a benchmark fault data set and the bearing data set collected from our own laboratory, and the experimental results demonstrate that RLLE vote achieves the most significant performance compared existing state-of-art methods.

Keywords: Feature selection; Locally linear embedding; Regularization technology; Bearing fault diagnosis

1. Introduction

In many pattern recognition and machine learning applications, the dimensionality of the features(or variables) is becoming much higher[1,2]. Such examples can be found in face recognition[3], handwriting character recognition[4], data mining[5], bearing fault diagnosis[6–8] and so on[9,10]. Usually, the high-dimensionality property of data brings at least two challenges for the learning algorithm, 1) it increases the computational burden of the algorithm; 2) the curse of dimensionality may degrade the performance of the learning algorithm[11]. To overcome these challenges, one always utilizes the dimension reduction techniques prior to processing data to the learning algorithm.

Typically, dimension reduction can be divided into two types: (1) feature selection[12] and (2) feature extraction[13]. Feature selection methods reduce the dimensionality of data by selecting a subset from original input, while feature extraction algorithms reduce the data's dimensionality by relying on a certain property of the original input. Compared with feature selection, usually, low-dimensional features obtained by feature extraction are lack of meaningful interpretations, which restricts its application in practical engineering. On the contrary, the feature obtained by feature selection methods has distinct interpretations, which is widely used in many practical engineering[14], such as pattern recognition[15], image retrieval[16] and son on[17,18]. Consequently, we are particularly interested in feature selection in this paper.

Feature selection methods can be classified into three groups : *filter*, *wrapper*, and *embedded*[12]. The filter-based feature selection algorithms evaluate the importance of features by employing a predefined criterion, which is independent on the learning algorithms. The wrapper-based methods measure the significance of features by employing a predetermined learning algorithm. The embedded-based methods integrate the feature evaluation criteria into the learning algorithm to evaluate each feature. Note that both wrapper-based and embedded-based methods outperform the filter-based ones, since they take account of the learning algorithm. However, these approaches bring huge computational burden, which impedes their application dealing with high-volume data. According to the above analysis, the filter-based methods are more attractive and practical, especially when the amount of the data are large. In this paper, we focus on filter-based methods for feature selection.

Depending on whether the label information is available, filter-based feature selection methods can be categorized into supervised ones and unsupervised ones. The key of the supervised method is to evaluate the importance of each feature by employing a priori information, like Fisher score[19] which evaluates the importance of each feature depending on its discriminative. While the unsupervised methods sort the features based on its ability of preserving some properties of original input, like data variance[20] which ranks the features by their variance. Overall, the performance of supervised filter-based feature selection methods are superior to unsupervised ones, since the label information is available. However, obtaining label information is expensive and the amount of labeled is usually very limited in many cases. In other words, most supervised feature selection methods may bring 'small labeled-sample problem'[21].

More recently, inspired by the phenomenon that the data with the same class often cluster together, while the data with different class may separate in the original space, various extensions to the basic local structure of data have gained great popularity in feature selection. Some works demonstrate that the local structure of data is beneficial to seeking important features in unsupervised methods. Laplacian score[22] is one of such method, which scores each feature by its capability of preserving the learnt local structure. More recently, Liu et al.[23] proposed a filter-based graph-preserving feature selection framework. Generally speaking, data variance[20], Laplacian score[22], Fisher score[19], and constraint score[24] are all unified into this framework[25]. In such methods, feature selection problem is formulated to evaluate each feature by evaluating its ability of preserving the graph-structure which is learned by a predefined algorithm.

Despite the fact that some filter-based unsupervised feature selection methods have already gained great popularity in many real-word applications, it can still be further improved. Yao et al.[25] integrated locally linear embedding[26] into the graph-preserving feature selection framework. To be specific, it utilizes the locality information of data to construct a graph, and then measures the significance of each feature by evaluating its ability of preserving the graph-structure. Moreover, Lots of researchers have revealed the effectiveness of LLE[27,28]. However, we find that it still brings at least two drawbacks that directly incorporate LLE into the graph-preserving framework, which will impede the performance of LLE in feature selection. To solve these drawbacks, we propose a new unsupervised filter-based feature selection with new criteria to measure the graph-preserving ability of the feature, and we called it LLE vote. Experimental results on two rolling bearing data sets reveal the effectiveness of the proposed method.

It is worth noting that the main contributions of this paper are summarized as follows:

(1) With analysis of directly embedding LLE into the graph-preserving feature selection framework, we find it have at least three drawbacks: 1) it is susceptible to noise in computing graph-preserving; 2) it is sensitive to K-nearest neighbors in constructing graph-preserving framework.

(2) To overcome the problems of directly embedding LLE into the graph-preserving framework, we propose a new criteria to measure the importance of each feature. In the new criteria, l_1 and l_2 regularization are introduced into the high-dimensional reconstruction model of LLE. Then, the weights are utilized to evaluate the importance of the feature.

(3) The RLLE vote algorithm is employed to select the features of two kinds of bearing data sets to validate the algorithm.

The reminder of this paper is organized as follows. Section 2 briefly reviews several filter-based feature selection methods. Then, we introduce the graph-preserving feature selection framework by LLE in detail and propose the RLLE vote in Section 3. Section 4 shows the experiments results. Finally, the conclusions are drawn in Section 5.

Table 1. Notations

Notation	Description
C	number of classes
d	sample's dimensionality
n	number of samples
x_i	the i-th sample, where $x_i \in R^d$
X	data matrix, where $X = \{x_1, ..., x_n\}$
n_P	number of samples in the P-th class
1	a vector with all elements equal to 1
I	identity matrix
f_r	the r-th feature of all the data
f_r^P	the r-th feature of the P-th class
f_{ri}^P	the r-th feature of the i-th sample in the P-th class
μ_r	center of the r-th feature
μ_r^P	center of the r-th feature in the P-class
e	$e = (e_1, ..., e_C)$
e_P	$e_P(i)$ = 1, if the i-th data belongs to the P-class, or $e_P(i)$ = 0
s(k)	a spectrum for k = 1,...,K (K is the number of spectrum lines)
f_k	the frequency value of k-th lines

2. Related works

In this section, we will briefly introduce several relevant filter-based feature selection methods. Some related notations are listed in Table 1 for explanation. Vectors are represented by lowercase letters (e.g., x), and matrices are indicated by capital boldface (e.g., **X**).

Data variance[20], the simplest unsupervised feature selection method, is utilized to evaluate the importance of each feature by its variance. Let Var_r denote the variance of r-th feature, and it can be computed as follows:

$$Var_r = \frac{1}{n}\sum_{i=1}^{n}(f_{ri} - \mu_r)^2. \tag{1}$$

where $\mu_r = \frac{1}{n}\sum_{i=1}^{n} f_{ri}$. The large Var_r means that the feature is representative.

Fisher score[19], a supervised feature selection method, measures the importance of the feature by evaluating its ability of maximizing the distance of inter-class and minimizing the distances of intra-class simultaneously. We denote the Fisher score of the r-th feature as FS_r, which is computed as follows:

$$FS_r = \frac{\sum_{P=1}^{C}(\mu_r^P - \mu_r)^2}{\sum_{P=1}^{C}\sum_{i=1}^{n_P}(f_{ri}^p - \mu_r^P)^2}. \tag{2}$$

where $\mu_r^P = \frac{1}{n_P}\sum_{i=1}^{n_P} f_{ri}^P$.

Laplacian score[22], an unsupervised feature selection method, evaluates the feature by its ability of preserving the local structure. Note that Laplacian score supposes that the local structure of the data plays a important role in feature selection. Let LS_r represent the Laplacian score of r-th feature, and it can be computed as follows:

$$LS_r = \frac{\sum_{i=1}^{n}\sum_{j=1}^{n}(f_{ri} - f_{rj})^2 S_{ij}}{\sum_{i=1}^{n}(f_{ri} - \mu_r)^2 d_{ii}}. \tag{3}$$

where $\mathbf{D}$ is a diagonal matrix with elements $d_{ii} = \sum_{j=1}^{n} S_{ij}$, and S_{ij} denotes the weight coefficient between x_i and x_j. It is defines as follows:

$$S_{ij} = \begin{cases} e^{-\frac{||x_i - x_j||^2}{\sigma}} & \text{if } x_i \text{ and } x_j \text{ are neighbors,} \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

where σ is a constant set. The term "if x_i and x_j are neighbors" denotes the local structure of sample. In practice, one always employs σ-ball and k-nearest neighbors to find the neighborhood of each sample. We denote the weight matrix $\mathbf{S} = (s_1, s_2, ..., s_n)$, then $\mathbf{D} = Diag(\mathbf{S1})$, where $Diag(\cdot)$ represents a diagonal matrix.

Constraint score[24], a semi-supervised feature selection method, can deal with partial label information. It utilizes the pairwise constraints. Specifically, when the pairwise belong to the same class, must-link constraints should be used in the model, otherwise, cannot-link constraints should be used. Two constraints scores are proposed, CS_r^1 and CS_r^2, to evaluate the importance of the r-th feature. They are defined as follows[24]:

$$CS_r^1 = \frac{\sum_{(x_i,x_j)\in M}(f_{ri} - f_{rj})^2}{\sum_{(x_i,x_j)\in C}(f_{ri} - f_{rj})^2}. \tag{5}$$

$$CS_r^2 = \sum_{(x_i,x_j)\in M}(f_{ri} - f_{rj})^2 - \lambda \sum_{(x_i,x_j)\in C}(f_{ri} - f_{rj})^2. \tag{6}$$

where $\mathbf{M}$ = {$(x_i,x_j)|$ x_i and x_j belong to the same class} and $\mathbf{C}$ = {$(x_i, x_j)|x_i$ and x_j belong to different classes} respectively represent the must-link constraints and the cannot-link constraints, and λ is a parameter to balance the two terms in Eq.(6).

More recently, inspired by the phenomenon that the sparsity linear representation can improve the robustness of the model against the noise. Liu et al.[23] proposed an unsupervised filter-based feature selection method called sparsity score. It first utilizes l_1 regularization to construct a sparsity graph S, and it formulates as follows:

$$\min_{s_i} ||s_i||_1, \quad s.t.\ x_i = \mathbf{X}s_i, \ \sum_{j=1}^{n} s_{ij} = 1. \tag{7}$$

where $s_i = (s_{i,1}, ..., s_{i,i-1}, 0, s_{i,i+1}, ..., s_{i,n})^T$ and $\mathbf{S} = (s_1, ..., s_n)^T$. Then the measurement SS_r of the r-th feature can be calculated as:

$$SS_r{}^1 = \frac{\sum_{i=1}^{n}(f_{ri} - \sum_{j=1}^{n} s_{ij} f_{rj})^2}{\frac{1}{n}\sum_{i=1}^{n}(f_{ri} - \mu_r)^2}. \tag{8}$$

In [23], Liu et al. also proposed a filter-based graph-preserving feature selection method as follows:

$$score_r^1 = \frac{f_r^T \mathbf{A} f_r}{f_r^T \mathbf{B} f_r}. \tag{9}$$

$$score_r^2 = f_r^T \mathbf{A} f_r - \lambda f_r^T \mathbf{B} f_r. \tag{10}$$

where λ is a parameter to balance the two terms in Eq(10). Then, the aforementioned feature selection methods can be embedded into this framework, and the corresponding $\mathbf{A}$ and $\mathbf{B}$ are listed in Table 2. In this table, $\mathbf{D}^M = Diag(\mathbf{S}^M\mathbf{1})$, $\mathbf{D}^C = Diag(\mathbf{S}^C\mathbf{1})$, and the elements in matrix $\mathbf{S}^M$ and $\mathbf{S}^C$ are calculated as:

$$s_{ij}^M = \begin{cases} 1 & if\ (x_i, x_j) \in \mathbf{M}\ or\ (x_j, x_i) \in \mathbf{M}, \\ 0 & otherwise. \end{cases} \tag{11}$$

$$s_{ij}^{C} = \begin{cases} 1 & if\ (x_i, x_j) \in \mathbf{C}\ or\ (x_j, x_i) \in \mathbf{C}, \\ 0 & otherwise. \end{cases} \tag{12}$$

Table 2. The definitions of **A** and **B** for several filter-based feature selection methods.

Algorithm	**A** and **B** definition	Graph-preserving form
Data variance[20]	$\mathbf{A}$ = I; $\mathbf{B} = \frac{1}{n}11^T$	Eq.(10) with $\lambda = 1$
Fisher score[19]	$\mathbf{A} = \sum_{P=1}^{C} \frac{1}{n_P} e_P e_P^T - \frac{1}{n} e e^T$; $\mathbf{B} = \mathbf{I} - \frac{1}{n_P} e_P e_P^T$	Eq.(9)
Laplacian score[22]	$\mathbf{A}$ = **D** - **S**; $\mathbf{B}$ = **D**	Eq.(9)
Constraint score[24]	$\mathbf{A} = \mathbf{D}^M - \mathbf{S}^M$; $\mathbf{B} = \mathbf{D}^C - \mathbf{S}^C$;	Eq.(9) and Eq.(10)
Sparsity score[23]	$\mathbf{A} = \mathbf{I} - \mathbf{S} - \mathbf{S}^T + \mathbf{S}\mathbf{S}^T$; $\mathbf{B} = \mathbf{I} - \frac{1}{n_P} e_P e_P^T$	Eq.(10) with λ = 0

3. The proposed method

3.1. Problem formulation

As a typical manifold learning algorithm, LLE first learns the local structure of data in the high-dimensional space, and then obtains the low-dimensional embedding results by preserving these structures. In the previous work[26], many researcher have been embedded LLE into the graph framework for feature extraction. Therefore, it is reasonable to extend LLE into filter-based feature selection task. In order to investigate the potential of LLE in feature selection, Yao et al.[25] proposed a graph-framework based on LLE to feature selection. However, to our best knowledge, we have not found any work using LLE to vote the feature so far. In this study, we first introduce how to embed LLE into the graph-preserving framework. To do so, we first employ LLE algorithm to model the local structure, which can be summarized as follows:

1) Find the neighborhood $N_i = \{x_j, j \in Q_i\}$ for each sample x_i;

2) Compute the reconstruction weights by minimizing the reconstructing error of x_i using samples in Q_i.

In step 1), the Euclidean distance is commonly utilized to find K-nearest neighbors for x_i. And then step 2) aims to find the optimal reconstruction weights based on the obtained K-nearest neighbors. The reconstruction weights are calculated by solving the following formula:

$$\min_{\{w_{ij}, j \in Q_i\}} ||x_i - \sum_{j \in Q_i} w_{ij} x_j||^2, \quad s.t. \sum_{j \in Q_i} w_{ij} = 1. \tag{13}$$

The construction weights matrix $\mathbf{W} = [w_{ij}]_{n \times n}$ is obtained by repeating step 1) and step 2) for all samples. In matrix **W**, $w_{ij} = 0$, if $x_j \notin Q_i$. Note that the least squares method is always utilized to solve Eq.(13).

Then, the importance of each feature is evaluated by its ability to preserving these weights. The measurement $Score_r$ of the r-th feature can be computed as follows[25]:

$$\begin{aligned} Score_r &= \sum_{i=1}^{n} (f_{ri} - \sum_{j=1}^{n} w_{ij} f_{rj})^2 \\ &= f_r^T (\mathbf{I} - \mathbf{W} - \mathbf{W}^T + \mathbf{W}^T \mathbf{W}) f_r. \end{aligned} \tag{14}$$

Then the ranking list of the features can be obtained according to their $Score_r$, and select the top d features with lowest scores. The detailed procedures of this method is shown in Algorithm 1. Let $\mathbf{A} = \mathbf{I} - \mathbf{W} - \mathbf{W}^T + \mathbf{W}^T W$, $\lambda = 0$, the proposed method can be unified into the aforementioned framework in Eq.(10).

The aforementioned method that directly embeds LLE into the graph-preserving framework feature selection are shown in Algorithm 1. Hence, the features reconstructed by LLE plays an

important role for this method to select the representative features. Recalling the measurement of Algorithm 1 in Eq.(14), we find three drawbacks in it which are summarized as follows:

• Because the ordinary least square algorithm is utilized to calculate the reconstruction weights, the model of Algorithm 1 is sensitive to noise[29].

• The model cannot select the K-nearest neighbors adaptively using Euclidean distance to measure the pairwise similarity. As shown in Figure 1, we can see that the 3-nearest neighbors of sample x_i are samples x_1, x_2 and x_3 in original space. However, x_3 is the false neighbor of sample x_i. The measurement in Eq.(14) could not capture this case. Actually, the graph-preserving ability of the feature should take this case into consideration.

Algorithm 1 Embedding LLE into the graph-preserving feature selection

Input: The data matrix **X**.
Output: The rank feature list.
Procedure:
1): Find K-nearest neighbors of x_i, then compute its weights w_{ij} by Eq.(13). Repeated these two procedures for all samples, and construct matrix **W**;
2): Evaluate the importance of the d feature by Eq.(14);
3): Rank the d feature in ascending order according to its score;
4): **return** The ranking list of the feature.

Due to these drawbacks, the measurement of Algorithm 1 may fail in some cases, which means that its performance will degrade. To address these problems, we propose a new criteria in next subsection.

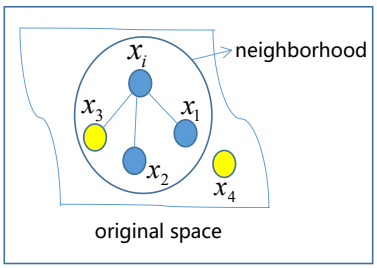


Figure 1. Select the local structure of the graph by embedding LLE into the graph-preserving framework

3.2. RLLE vote

As previous mentioned, one can find that there are two weaknesses that directly embeds LLE into the graph-preserving framework. To solve these weaknesses, we propose a new criteria to evaluate the importance of the feature. In the new criteria, we integrate the regularization technology into the computation of reconstruction weights. Due to the fact that the computation of objective function with l_1 regularization is expensive, and many important variables may be lose by this way. Thus, in this new criteria, we integrate the l_1 and l_2 regularization into the computation of the local structure. Specifically, we calculate the reconstruction weights of each element in f_r as follows:

$$\min_{\{w_{ij}^r, j\in Q_i^r\}} ||f_{ri} - \sum_{j\in Q_i^r} w_{ij}^r f_{rj}||^2 + \lambda_1 ||w_{ij}^r||_1 + \lambda_2 ||w_{ij}^r||_2^2 \tag{15}$$

where

$$||w_{ij}^r||_1 = \sum_{j=1}^{k} |w_{ij}^r|, \quad ||w_{ij}^r||_2^2 = \sum_{j=1}^{k} w_{ij}^{r^2}.$$

the neighborhood index set Q_i^r : = {j : if f_{rj} is one of the K-nearest neighbors of f_{ri} }, and λ_1 and λ_2 are nonnegative tuning parameters. It is difficult to directly calculate Eq.(15), because it is not

differentiable when W_{ij}^r is equal to zero[30]. LARS-EN, a relatively conservative iterative algorithm, is commonly utilized to compute the optimal solution.

After obtaining the reconstruction weight matrix $\mathbf{W}^r = [w_{ij}^r]$ for the r-th feature using Eq.(15), we vote each feature by its ability to preserving these weights. We denote $RLLEV_r$ as the vote of the r-th feature, which is calculated as follows:

$$\begin{aligned} Vote_r &= \sum_{i=1}^{n}(f_{ri} - \sum_{j=1}^{n} w_{ij}^r f_{rj})^2 \\ &= f_r^T(\mathbf{I} - \mathbf{W}^r - \mathbf{W}^{r^T} + \mathbf{W}^{r^T}\mathbf{W}^r) f_r. \end{aligned} \tag{16}$$

We use the above measurement to evaluate the graph-preserving ability of each feature, and choose the top d features with lowest votes. The detailed procedure of RLLE vote is presented in Algorithm 2.

Algorithm 2 RLLE vote

Input: The data matrix **X**.
Output: The ranked feature list.
Procedure:
1): For each f_r, recompute its K-nearest neighborhood set Q_i^r;
Construct the reconstruction weighting matrix **W** and $\mathbf{W}^r$
via Eq.(13) and Eq.(15).
2): Compute the importance of the d feature by Eq.(16);
3): Rank the d feature in ascending order according to its RLLE vote;
4): **return** The ranking list of the feature.

Recalling the aforementioned weaknesses of Algorithm 1, we can find that LLE vote can overcome them efficiently. To be specifically, it can adaptively select K-nearest neighbors by setting an iterative termination condition in the LARS-EN algorithm. Furthermore, Zhang et al.[29] have proved that integrating l_1 and l_2 regularization into the model of local reconstruction can improve the robustness of the model.

4. Experiments results

In this section, we utilize the following experiments to evaluate the efficiency of our proposed methods on benchmark fault data set and the bearing data set collected from our own laboratory, by comparing with several relevant dimensionality reduction methods.

5. Experiments results

Bearing data set 1: The bearing data set is collected from the Case Western Reserve University Bearing Data Center (CWRU). This data set has become a benchmark for validating fault diagnosis algorithms. As shown in Figure 2, the test platform is mainly consisted of motor (left), torque transducer/encoder (centre) and dynamometer (right).

This bearing data set includes four types of data set(normal condition, ball fault, inner race fault and outer race fault), in which each kind of data contains 100 samples. Moreover, we select 1024 features as a sample, that is, the dimensionality of each sample is equal to 1024.

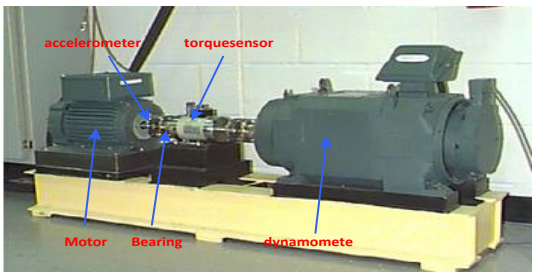


Figure 2. The bearing test platform 1

Table 3. Time-domain and Frequency-domain statistical features

Time-domain features		
$T_1 = \frac{1}{D}\sum_{i=1}^{D} x(i)$	$T_2 = \sqrt{\frac{1}{D-1}\sum_{i=1}^{D}(x(i)-T_1)^2}$	$T_3 = \sqrt{\frac{1}{D}\sum_{i=1}^{D}(x(i))^2}$
$T_4 = \left(\frac{1}{D}\sum_{i=1}^{D}\sqrt{\lvert x(i)\rvert}\right)^2$	$T_5 = \frac{\sum_{i=1}^{D}(x(i)-T_1)^2}{(D-1)T_2^3}$	$T_6 = \frac{\sum_{i=1}^{D}(x(i)-T_1)^4}{(D-1)T_2^4}$
$T_7 = \frac{1}{T_3}\max\lvert x(i)\rvert$	$T_8 = \frac{max(x_i)-min(x_i)}{T_4}$	$T_9 = \frac{T_3}{\frac{1}{D}\sum_{i=1}^{D}\lvert x(i)\rvert}$
$T_{10} = \frac{max(x_i)-min(x_i)}{\frac{1}{D}\sum_{i=1}^{D}\lvert x(i)\rvert}$	$T_{11} = max\ x(i)$	$T_{12} = min\ x(i)$
$T_{13} = T_{11} - T_{12}$	$T_{14} = \frac{1}{D}\sum_{i=1}^{D}\lvert x(i)\rvert$	$T_{15} = \frac{T_{11}}{T_{14}}$
$T_{16} = \sum_{i=1}^{D} x(i)^2$	$T_{17} = \frac{\sum_{i=1}^{D} x(i)^3}{D}$	$T_{18} = \frac{T_{11}}{T_3}$
$T_{19} = \frac{T_{11}}{T_4}$	$T_{20} = \frac{T_{11}}{T_2^2}$	
Frequency-domain features		
$F_1 = \frac{1}{K}\sum_{k=1}^{K} s(k)$	$F_2 = \sqrt{\frac{1}{K-1}\sum_{k=1}^{K}(s(k)-F_1)^2}$	$F_3 = \frac{\sum_{k=1}^{K}(s(k)-F_1)^3}{K\sqrt{F_2}^3}$
$F_4 = \frac{1}{KF_2^2}\sum_{k=1}^{K}(s(k)-F_1)^4$	$F_5 = \frac{1}{\sum_{k=1}^{K} s(k)}\sum_{k=1}^{K}(s(k)f_k)$	$F_6 = \sqrt{\frac{\sum_{k=1}^{K}(f_k-F_5)^2 f_k}{K}}$
$F_7 = \sqrt{\frac{\sum_{k=1}^{K}(f_k)^4 s(k)}{\sum_{k=1}^{K}(f_k)^2 s(k)}}$	$F_8 = \frac{\sum_{k=1}^{K}(f_k)^2 s(k)}{\sqrt{\sum_{i=1}^{K}(s(k)\sum_{i=1}^{K} f_k^4 s(k)}}$	$F_9 = \frac{F_6}{F_5}$
$F_{10} = \frac{\sum_{k=1}^{K}(f_k-F_5)^4 s(k)}{KF_6^2}$	$F_{11} = \frac{\sum_{k=1}^{K}\sqrt{(f_k-F_5)^4 s(k)}}{KF_6^2}$	$F_{12} = \frac{\sum_{k=1}^{K}(f_k-F_5)^3 s(k)}{KF_6^3}$
$F_{13} = \frac{\sum_{i=1}^{K}(f_k-F_5)^4 s(k)}{KF_6^4}$	$F_{14} = \frac{\sum_{i=1}^{K}(f_k-F_5) s(k)}{K\sqrt{T_6}}$	

Bearing data set 2: This bearing data set is obtained from a real test platform in our own laboratory. As shown in Figure 3, the test platform consists of a motor (left), a gearbox (centre) and a bearing (right). There are four different data sets(bearing case 1–bearing case 4) collected from the test platform under different operating conditions. In this data set, the rotational speed of the motor is 1400 r/min. The vibration signals are obtained from the bearings with a sample frequency of 1kHz and 10kHz. Note that each data set also contains four types of data(normal condition, ball fault, inner race fault and outer race fault), in which each kind of data contains 100 samples. Furthermore, according to the sampling rate and the frequency of signal, the dimensionality of each sample is also 1024. The detailed description of this data sets are summarized in Table 4.

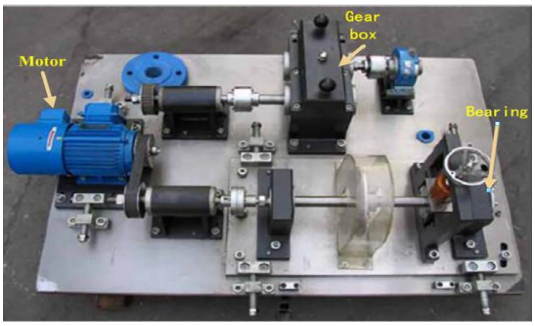


Figure 3. The bearing test platform 2

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 22 November 2019 doi:10.20944/preprints201911.0261.v1

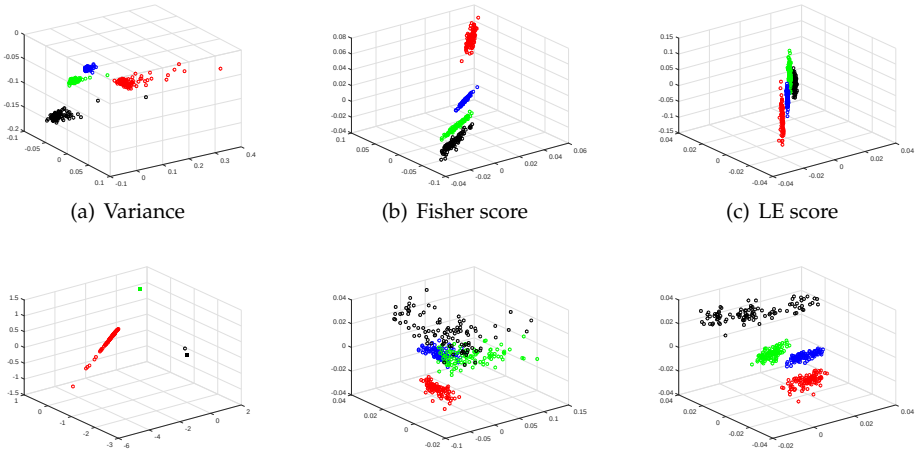
Moreover, for reducing the influence of nonlinear characteristic and noise, in the following experiments, we first compute the statistic features of each sample in the time-domain and frequency-domain spaces. The detailed depictions of the statistical features of the original data sets are shown in Table 3. Besides, all time-domain statistical features are also calculated in frequency-domain space, but they are not listed in Table 3 because of the page limitation. In order to improve the performance of our proposed methods, excellent statistical features are selected.

Table 4. Description of bearing data set 2

Data Name	sampling	Class	Number	load
Bearing case 1	1k	4	400	0Hp
Bearing case 2	10k	4	400	0Hp
Bearing case 3	1k	4	400	1Hp
Bearing case 4	10k	4	400	1Hp

5.1. Visualization evaluation

We evaluate the clustering performance of our proposed method by comparing the visualization results with several other relevant dimensionality reduction methods, i.e., Variance, Laplacian score, Fisher score, LLE and Algorithm 1. We perform all the dimensionality reduction algorithms on the bearing data set 1, and the embedding results are shown in Figure 4. From this Figure, we can easily find that: 1) The performance of Algorithm 1 is the worst, since it is sensitive to noise and the number of neighborhoods K; 2) The results of LE score and LLE have excellent intra-class compactness, but their inter-class separation are poor, that is, part of the samples with different labels overlap; 3) The performance of Variance and Fisher score are superior to LE score, LLE and Algorithm 1, due to the fact that the variance model can select features with large variance by its measurement criteria, and the supervised Fisher score can employ the label information for training samples; 4) Compared with other methods, RLLE vote outperforms than other related methods. The main reason is that RLLE vote can reveal the neighborhood relations between data samples clearly due to the imposed regularization technology, and by this way, it can adaptively select K-nearest neighbors and simultaneously improve the robustness of the model.



(a) Variance (b) Fisher score (c) LE score

(d) LLE (e) Algorithm 1 (f) RLLE vote

Figure 4. The three-dimensional(3D)embedding results obtained by different dimension reduction algorithms on the bearing data set 1. The red points denote normal data. The green points indicate inner fault data. The blue points represent ball fault data. The black points indicate outer fault data.

5.2. Quantitative clustering evaluation

In order to quantitative analysis the proposed methods, Fisher criterion is introduced. The Fisher criterion is a statical method that is used to compare variances of the two variational series, and it is defined as follows[31]:

$$F = S_b / S_w \tag{17}$$

$$S_w = \sum_{i=1}^{c} \sum_{j=1}^{N_c} (x_j - \mu_i)(x_j - \mu_i)^T \tag{18}$$

$$S_b = \sum_{i=1}^{c} (\mu_i - \bar{\mu})(\mu_i - \bar{\mu})^T \tag{19}$$

where S_b measures the distance of inter-class, and S_w denotes the distance of intra-class. The larger value of F is, the better performance of the corresponding algorithm will be.

In the second experiment, bearing data set 1, bearing case 1 and bearing case 2 are examined for quantitative evaluation. We show the comparison results in Table 5. From the Table, we can easily find that the quantitative results generally keep consistent with the visualization results. Specifically, the value of F obtained by Algorithm 1 is smallest, that is, the Algorithm 1 is the worst. Moreover, Fisher score can deliver higher value of F than Variance, LE score and LLE in most cases, due to the fact that it can make use of the label information of samples so that the performance of inter-class separation and intra-class compactness can be enhanced. In addition, RLLE vote deliver the largest F value in most cases so that it is superior to other methods. According to the aforementioned analysis, it shows the validity of the proposed measurement in RLLE vote.

Table 5. Quantitative clustering evaluation results

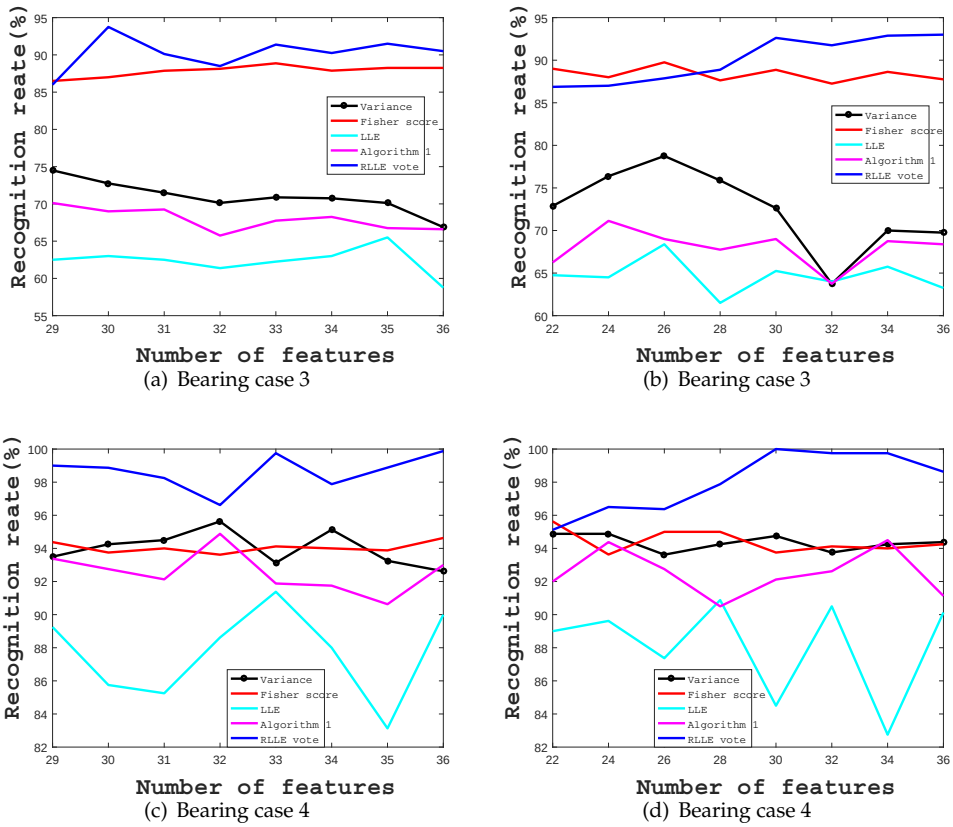
Methods	CWRU	Bearing case 1	Bearing case 2
	F	F	F
Variance	3.5234	0.1450	0.4826
Fisher score	47.311	11.382	3.2519
LE score	6.2297	19.071	0.9313
LLE	1.4567	1.2046	0.7835
Algorithm 1	0.5980	0.0994	0.1828
RLLE vote	111.35	19.240	2.9025

5.3. Fault recognition

In the third experiment, we evaluate our RLLE vote for recognizing bearing data sets via comparing with other relevant methods. The involved bearing data sets include bearing case 3 and bearing case 4. For quantitative recognition evaluations, we perform k-nearest-neighbor(kNN) over the low-dimensional Y by each method due to its simplicity. For each experimental setting, we select 80 samples from each type of this data set as training set and the others as test set. Therefore, the results are averaged over 10 random splits of training/test samples to alleviate the bias.

In this experiment, we evaluate the recognition performance under different numbers of features and show the results in Figure 5. From this Figure, we can have the following findings. First, the recognition evaluation results of Algorithm 1 and LLE are usually the worst, since they cannot provide a clear separation of samples from different classes, and cannot enhance compactness for intra-class samples simultaneously; Second, Fisher score obtains comparable and even better outcomes than other remaining approaches on the whole, since it can employ the label information of samples, and the performance of inter-class separation and intra-class compactness can be enhanced; Third, the performance of RLLE vote can be improved via increasing the numbers of features in virtually all cases. To be specifically, the performance of RLLE vote firstly increases faster as the number of features is

relatively small, while the recognition result goes up slower when the numbers of features is large; Final, our RLLE vote is superior to other relevant methods in most cases, especially on the bearing case 4. Therefore, the experiment results validate the efficiency of our proposed method.



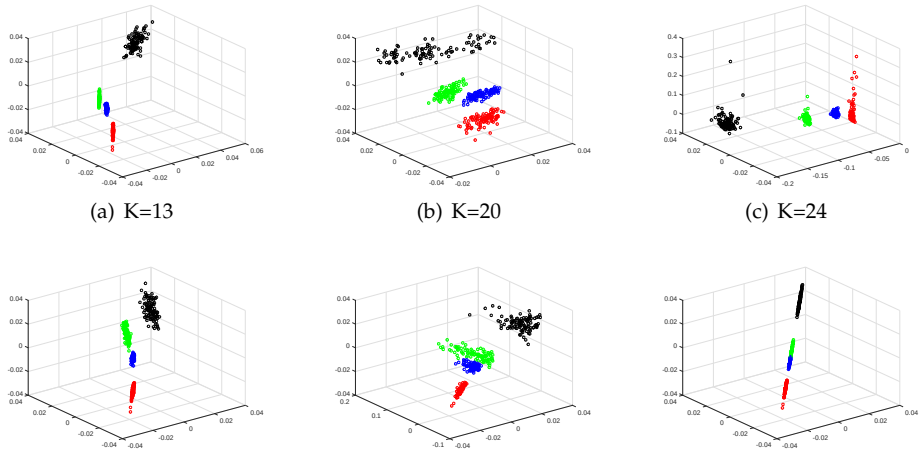
(a) Bearing case 3 (b) Bearing case 3

(c) Bearing case 4 (d) Bearing case 4

Figure 5. Recognition rate (%) of different dimension reduction algorithms using different numbers of features.

5.4. Parameter analysis of RLLE vote

Finally, we investigate the influence of the model parameter selection on the data visualization results. In this experiment, bearing data set 1 is employed for evaluation. Due to the fact that it is impossible to visualize all the possible results under different parameter settings. In this study, we mainly explore the effects of parameter K on the visualization results, and we change K from 13 to 40. The parameter analysis results on the visualizations are shown in Figure 6. From this figure, one can find that when the value of K is small than 32, our proposed method can offer a distinct separation among the different kinds of samples and simultaneously obtain the enhanced compactness for intra-class samples. On the contrary, when the value of K is larger than 32, RLLE vote has excellent intra-class compactness, but their inter-class separation are degrade, that is, part of the samples with different labels overlap. Generally speaking, RLLE vote can perform well in a wide range of parameter settings, which demonstrates that the performance of our proposed method is robust to the parameter K, i.e., the selection of K will be relatively easier for the real applications. It is worth noting that we can obtain a guidance of the selection of model parameters based on the above experimental results and analysis.



(a) K=13 (b) K=20 (c) K=24

(d) K=28 (e) K=32 (f) K=36

Figure 6. The three-dimensional(3D)embedding results over different parameter K on the bearing data set 1. The red points denote normal data. The green points indicate inner fault data. The blue points represent ball fault data. The black points indicate outer fault data.

6. Conclusion

In this paper, the idea of LLE embeds LLE into the graph-preserving feature selection framework, and then a new filter-based feature selection method called LLE vote is proposed. RLLE vote introduces l_1 and l_2 regularization in high-dimensional reconstruction of LLE, which can address the existing problems that directly embedding LLE into the graph-preserving feature selection framework. Specifically, the importance of each feature is evaluated by measuring the difference between feature reconstructed by RLLE vote and the original data. Extensive experimental results on two rolling bearing data sets not only validate the effectiveness of our the proposed method, but also demonstrate that our proposed method is superior to the existing state-of-art methods.

In feature, the optimal determination of the sparse reconstruction still remains an open problem in reality, which needs further investigation in future. Furthermore, the local structure is actually consisted of both the location of the neighbors and the reconstruction weights, it is difficult to determine the effects of them yet. In addition, investigating the joint of feature selection learning and tensor learning will be considered, since tensor learning algorithms ease both the curse dimensionality and the computation issues.

References

1. Dernoncourt, D.; Hanczar, B.; Zucker, J.D. Analysis of feature selection stability on high dimension and small sample data. *Computational statistics & data analysis* **2014**, *71*, 681–693.
2. He, R.; Zheng, W.S.; Tan, T.; Sun, Z. Half-quadratic-based iterative minimization for robust sparse representation. *IEEE transactions on pattern analysis and machine intelligence* **2013**, *36*, 261–275.
3. Yang, M.H.; Kriegman, D.J.; Ahuja, N. Detecting faces in images: A survey. *IEEE Transactions on pattern analysis and machine intelligence* **2002**, *24*, 34–58.
4. Liu, C.L.; Yin, F.; Wang, D.H.; Wang, Q.F. Online and offline handwritten Chinese character recognition: benchmarking on new databases. *Pattern Recognition* **2013**, *46*, 155–162.
5. Purarjomandlangrudi, A.; Ghapanchi, A.H.; Esmalifalak, M. A data mining approach for fault diagnosis: An application of anomaly detection algorithm. *Measurement* **2014**, *55*, 343–352.
6. Meng, Z.; Zhan, X.; Li, J.; Pan, Z. An enhancement denoising autoencoder for rolling bearing fault diagnosis. *Measurement* **2018**, *130*, 448–454.

7. Gao, Z.; Cecati, C.; Ding, S.X. A survey of fault diagnosis and fault-tolerant techniques—Part I: Fault diagnosis with model-based and signal-based approaches. *IEEE Transactions on Industrial Electronics* **2015**, *62*, 3757–3767.
8. Glowacz, A.; Glowacz, W.; Glowacz, Z.; Kozik, J. Early fault diagnosis of bearing and stator faults of the single-phase induction motor using acoustic signals. *Measurement* **2018**, *113*, 1–9.
9. Kumar, S.; Pandey, A.; Satwik, K.S.R.; Kumar, S.; Singh, S.K.; Singh, A.K.; Mohan, A. Deep learning framework for recognition of cattle using muzzle point image pattern. *Measurement* **2018**, *116*, 1–17.
10. Zhang, L.; Tian, F.; Pei, G. A novel sensor selection using pattern recognition in electronic nose. *Measurement* **2014**, *54*, 31–39.
11. Jain, A.K.; Duin, R.P.W.; Mao, J. Statistical pattern recognition: A review. *IEEE Transactions on pattern analysis and machine intelligence* **2000**, *22*, 4–37.
12. Guyon, I.; Elisseeff, A. An introduction to variable and feature selection. *Journal of machine learning research* **2003**, *3*, 1157–1182.
13. Guyon, I.; Elisseeff, A. An introduction to feature extraction. In *Feature extraction*; Springer, 2006; pp. 1–25.
14. Motoda, H.; Liu, H. Feature selection, extraction and construction. *Communication of IICM (Institute of Information and Computing Machinery, Taiwan) Vol* **2002**, *5*, 2.
15. Mu, H.Q.; Yuen, K.V. Modal frequency-environmental condition relation development using long-term structural health monitoring measurement: Uncertainty quantification, sparse feature selection and multivariate prediction. *Measurement* **2018**, *130*, 384–397.
16. Bar-Hillel, A.; Hertz, T.; Shental, N.; Weinshall, D. Learning a mahalanobis metric from equivalence constraints. *Journal of Machine Learning Research* **2005**, *6*, 937–965.
17. Gao, Z.; Cecati, C.; Ding, S.X. A Survey of Fault Diagnosis and Fault-Tolerant Techniques—Part II: Fault Diagnosis With Knowledge-Based and Hybrid/Active Approaches. *IEEE Transactions on Industrial Electronics* **2015**, *62*, 3768–3774.
18. Chen, L.; Li, J.; Zhang, Y.H.; Feng, K.; Wang, S.; Zhang, Y.; Huang, T.; Kong, X.; Cai, Y.D. Identification of gene expression signatures across different types of neural stem cells with the Monte-Carlo feature selection method. *Journal of cellular biochemistry* **2018**, *119*, 3394–3403.
19. Gu, Q.; Li, Z.; Han, J. Generalized fisher score for feature selection. *arXiv preprint arXiv:1202.3725* **2012**.
20. Bishop, C.M.; others. *Neural networks for pattern recognition*; Oxford university press, 1995.
21. Jain, A.; Zongker, D. Feature selection: Evaluation, application, and small sample performance. *IEEE transactions on pattern analysis and machine intelligence* **1997**, *19*, 153–158.
22. He, X.; Cai, D.; Niyogi, P. Laplacian score for feature selection. Advances in neural information processing systems, 2006, pp. 507–514.
23. Liu, M.; Zhang, D. Sparsity score: A novel graph-preserving feature selection method. *International Journal of Pattern Recognition and Artificial Intelligence* **2014**, *28*, 1450009.
24. Zhang, D.; Chen, S.; Zhou, Z.H. Constraint Score: A new filter method for feature selection with pairwise constraints. *Pattern Recognition* **2008**, *41*, 1440–1451.
25. Yao, C.; Liu, Y.F.; Jiang, B.; Han, J.; Han, J. LLE score: A new filter-based unsupervised feature selection method based on nonlinear manifold embedding and its application to image recognition. *IEEE Transactions on Image Processing* **2017**, *26*, 5257–5269.
26. Roweis, S.T.; Saul, L.K. Nonlinear dimensionality reduction by locally linear embedding. *science* **2000**, *290*, 2323–2326.
27. Su, Z.; Tang, B.; Ma, J.; Deng, L. Fault diagnosis method based on incremental enhanced supervised locally linear embedding and adaptive nearest neighbor classifier. *Measurement* **2014**, *48*, 136–148.
28. Zhang, S.q. Enhanced supervised locally linear embedding. *Pattern Recognition Letters* **2009**, *30*, 1208–1218.
29. Zhang, Y.; Ye, D.; Liu, Y. Robust locally linear embedding algorithm for machinery fault diagnosis. *Neurocomputing* **2018**, *273*, 323–332.
30. Zou, H.; Hastie, T. Regularization and variable selection via the elastic net. *Journal of the royal statistical society: series B (statistical methodology)* **2005**, *67*, 301–320.
31. Fisher, R.A. The use of multiple measurements in taxonomic problems. *Annals of eugenics* **1936**, *7*, 179–188.