Huang K, Zhang LQ. Semisupervised sparse multilinear discriminant analysis. JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY 29(6): 1058–1071 Nov. 2014. DOI 10.1007/s11390-014-1490-1

Semisupervised Sparse Multilinear Discriminant Analysis

Kai Huang (黄　锴) and Li-Qing Zhang (张丽清), *Member*, *IEEE*

MOE-Microsoft Key Laboratory for Intelligent Computing and Intelligent Systems, Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai 200240, China

E-mail: mariaki888888@sjtu.edu.cn; zhang-lq@cs.sjtu.edu.cn

Received December 4, 2013; revised May 4, 2014.

Abstract Various problems are encountered when adopting ordinary vector space algorithms for high-order tensor data input. Namely, one must overcome the Small Sample Size (SSS) and overfitting problems. In addition, the structural information of the original tensor signal is lost during the vectorization process. Therefore, comparable methods using a direct tensor input are more appropriate. In the case of electrocardiograms (ECGs), another problem must be overcome; the manual diagnosis of ECG data is expensive and time consuming, rendering it difficult to acquire data with diagnosis labels. However, when effective features for classification in the original data are very sparse, we propose a semisupervised sparse multilinear discriminant analysis (SSSMDA) method. This method uses the distribution of both the labeled and the unlabeled data together with labels discovered through a label propagation algorithm. In practice, we use 12-lead ECGs collected from a remote diagnosis system and apply a short-time-fourier transformation (STFT) to obtain third-order tensors. The experimental results highlight the sparsity of the ECG data and the ability of our method to extract sparse and effective features that can be used for classification.

Keywords ECG analysis, semisupervised learning, sparse coding, dimension reduction, tensor learning approach

1 Introduction

The importance of and the demand for classification and feature analysis mean that original electrocardiogram (ECG) signals must undergo a dimension-reduction process to prevent the "curse of dimensionality." The most common dimension-reduction approaches include principal component analysis (PCA)[1-3], independent component analysis (ICA)[4], and linear discriminant analysis (LDA)[5]. These methods can be classified into two types: supervised (LDA) and unsupervised (PCA and ICA). Supervised strategies require training data with class labels to generate a projection matrix. For the ECG classification problem, a number of machine learning methods have been used. Zhao and Zhang proposed a feature extraction method using a wavelet transform[6], and Hwang and Jen introduced a neural network approach for determining the features of an ECG signal[7]. Pasolli and Melgani presented an active learning method for ECG classification based on the morphology and temporal features of ECGs[8]. A PCA algorithm was used to extract features from ECG data[9], and an ECG feature extraction scheme using ICA was reported by Wu *et al.*[10-11].

With the development of research fields such as image analysis and multichannel biomedical signaling, methods using vector input data have been extended from the original vector space into matrix form. However, the original method must still consider the overfitting problem caused by too many coefficients[12]. In addition, converting a matrix to a vector causes the loss of structural information in the original data. LDA suffers from the Small Sample Size (SSS) problem that has led to the development of two-dimensional PCA (2DPCA)[13] and two-dimensional LDA (2DLDA)[14]. A common disadvantage of two-dimensional (2D) methods is that a single projection method is learned from only one side of the image matrix. Generalized low rank approximations of matrices (GLRAM)[15] and bidirectional LDA[16] have yet to be developed.

To more effectively extract valuable features from biomedical signals, the original signals are transformed into higher-order tensors using a wavelet transform, Gabor transform, or short-time Fourier transform (STFT). With the development of research topics in video analysis such as gait analysis and video emotion analysis, some widely used approaches have been extended to higher-order tensor versions, e.g., general

Regular Paper

The work was supported by the National Natural Science Foundation of China under Grant Nos. 91120305, 61272251, and the National Basic Research 973 Program of China under Grant No. 2015CB856004.

tensor discriminant analysis (GTDA)[17]. Other tensor methods include multilinear principal component analysis (MPCA)[18], uncorrelated multilinear PCA (UMPCA)[19-20], tensor rank 1 discriminative analysis (TR1DA)[21] and k-mode cluster-based discriminant analysis (DATER)[22]. These approaches can be divided into two classes[23]: those corresponding to a tensor-to-tensor (T2T) projection and those corresponding to a tensor-to-vector (T2V) projection. GTDA[17], MPCA[18], and DATER[22] belong to the T2T projection type, whereas TR1DA[21] and UMPCA[19-20] use T2V projections. DATER[22] is different from GTDA[17]. The within-class and the between-class scatter matrices of DATER are the sums of the cluster-based within-class and the cluster-based between-class scatter matrices, respectively.

Generally, 12-lead labeled ECG signals are not easy to obtain. Early studies mostly dealt with 1- or 2-lead ECG signals. The diagnosis of heart disease using an ECG is based on a specific waveform (i.e., specific frequency component) of a specific channel at a specific time. Hence, the original 12-lead ECG signal is transformed using STFT into a tensor representation of valuable features. The tensor-based approach has been adopted for ECG classification tasks in [24-26].

Because of the specificity of ECGs, the manual diagnosis of ECG data is expensive and time consuming, requiring a semisupervised method. In addition, not all the features are useful, implying the need for a selection process. In the tensor representation of an ECG signal, useful features tend to be sparse. Those that can be used to diagnose heart disease are always in a specific waveform in a specific lead at a specific position. Sparse coding was proposed for the recognition of such patterns, and this was later extended to sparse PCA[27] and sparse LDA[28]. Note that a tensor-based algorithm has already been proposed[29]. In this paper, we modify the discriminant problem to a Maximum Scatter Difference Discriminant Analysis problem. In other words, we proposed a sparse PCA method to find the discriminant directions. In fact, we combine the main idea of sparse PCA[27] and semisupervised LDA[30-31] into a semisupervised sparse multilinear discriminant analysis (SSSMDA) method. The main idea of semisupervised LDA in our paper is different from the previous one, as we add two regularization items by considering the labels of unlabeled data, assigned via label propagation[32]. Note that our sparse tensor discriminant analysis differs from that in [29], as we employ a T2V rather than a T2T projection. In addition, in this paper, we use the sparse PCA approach instead of sparse discriminant analysis.

The paper is organized as follows. Section 2 introduces some common tensor operations. In Section 3, we explain our choice of a sparse tensor representation for ECG data, giving a proper description of our semisupervised sparse discriminant analysis (SSSDA) method in Section 4. SSSDA is extended to a multilinear version in tensor space in Subsection 5.1. Subsections 5.2 and 5.3 address the convergence issue and the computational complexity, respectively. The effectiveness of our semisupervised model is tested on a toy dataset in Subsection 6.1, Subsection 6.2 introduces the 12-lead ECG database, and Subsection 6.3 presents our experimental results on the ECG dataset. Section 7 concludes this work.

2 Tensor Operations

We first introduce some definitions of tensor operations. In our paper, mathcal and uppercase letters denote tensors, e.g., $\mathcal{X}, \mathcal{Y}, \mathcal{Z}$. Matrices are expressed as uppercase bold italic letters, e.g., $\boldsymbol{X}, \boldsymbol{B}$. Lowercase bold italic letters are used for vectors, e.g., $\boldsymbol{u}, \boldsymbol{a}$, whereas regular lowercase and uppercase italic letters denote scalars, e.g., a, b, c, D, E.

Definition 1 (Tensor Product). *The tensor product of two vectors $\boldsymbol{x} \in \mathbb{R}^M$ and $\boldsymbol{y} \in \mathbb{R}^N$ is a matrix*:

$$(\boldsymbol{x} \otimes \boldsymbol{y})_{ij} = x_i \times y_j,$$

which is a rank-1 tensor of mode 2. Here, $0 < i \leqslant M$ and $0 < j \leqslant N$. x_i and y_j denote the i-th and j-th element of vector $\boldsymbol{x}$ and $\boldsymbol{y}$. The tensor product of three vectors $\boldsymbol{x} \in \mathbb{R}^M$, $\boldsymbol{y} \in \mathbb{R}^N$, and $\boldsymbol{z} \in \mathbb{R}^S$ is a mode-3 tensor:

$$(\boldsymbol{x} \otimes \boldsymbol{y} \otimes \boldsymbol{z})_{ijk} = x_i \times y_j \times z_k,$$

which is also of rank 1. Here, $0 < i \leqslant M$, $0 < j \leqslant N$, and $0 < k \leqslant S$. z_k denotes the k-th element of vector $\boldsymbol{z}$.

Definition 2 (Tensor Mode Product). *A mode-M tensor $\mathcal{X}$ of size $X \in \mathbb{R}^{N_1 \times N_2 \times \cdots \times N_M}$ multiplied by a vector of mode r is a tensor of size $N_1 \times N_2 \times \cdots \times N_{r-1} \times 1 \times N_{r+1} \times \cdots \times N_M$*:

$$(\mathcal{X} \times_r \boldsymbol{u})_{i_1 \times i_2 \times \cdots \times i_{r-1} \times 1 \times i_{r+1} \times \cdots \times i_M} = \sum_{i_r} (\mathcal{X}_{i_1 \times i_2 \times \cdots \times i_{r-1} \times i_r \times i_{r+1} \times \cdots \times i_M} \boldsymbol{u}_{i_r}),$$

which is in fact a tensor of mode $M-1$.

Definition 3 (Multiple Tensor Product). *The tensor product of multiple vectors forms a rank-1 tensor. To simplify its notation, we use the following form to represent the tensor product of several vectors*:

$$\boldsymbol{u}^1 \otimes \boldsymbol{u}^2 \otimes \cdots \otimes \boldsymbol{u}^n = \prod_{l=1}^{M} \otimes (\boldsymbol{u}^l)^{\mathrm{T}},$$

where $\boldsymbol{u}^i$ $(1 \leqslant i \leqslant n)$ denotes any vector.

3 Sparse Tensor Representation of ECG Data

ECG data are measured by a standard 12-lead diagnosis system, which includes channels I, II, III, aVR, aVL, aVF, V1, V2, V3, V4, V5, and V6. I, II, and III are limb leads, and aVR, aVL, and aVF are augmented limb leads. Note that the raw ECG signals usually feature strong background noise. Therefore, we apply several methods, such as a wavelet transformation, to remove the high-frequency noise, and use a median filter to eliminate the baseline drift[33]. The original ECG signal of one diagnosis is approximately 20 seconds long, and is composed of about 25 beats. Another pre-processing step consists of segmenting the signals into a set of ECG pieces, each of which contains one heartbeat. Fig.1 displays such an example of a 12-lead ECG signal. It is important to understand that if the 12 beats are combined into a single one, then the whole structure of the information is destroyed. Hence, the matrix form of a 12-beat ECG cannot be changed.

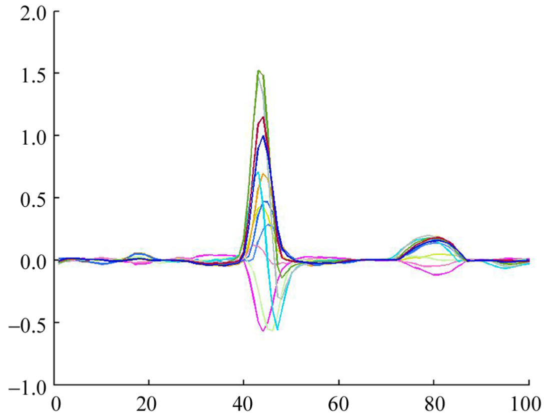


Fig.1. Example of 12-lead ECG signals in our ECG database.

The original signals represent features in the spatiotemporal domain. As ECG signals are non-stationary, we employ an STFT[34], rather than a regular Fourier transform, to recover information on the time at which a frequency component occurs. STFT provides useful information on the time resolution of the spectrum. In essence, a useful feature for classification is the specific waveform (specific frequency component) at a specific time point of a specific channel. We use STFT to transform the original signals into the spatial-spectral-temporal domain, and represent them as high-dimensional third-order tensors.

For a 12-lead (lead × time) ECG signal sample, $s[l,n]$ represents the discrete-time signal at time n for lead l. The STFT at time $n\triangle t$ and frequency f is defined by

$$\begin{aligned}STFT\{s[l,n]\}(m,w) &\equiv S(l,m,n)\\ &= \Sigma_{m=0}^{M}\omega(n-m)s(l,n)e^{-j2\pi fm},\end{aligned}$$

where $w[n]$ is the window function that selectively determines the portion of $s[l,n]$ for analysis. In this work, we choose the Hann window. After applying the STFT to the ECG signals, they are represented as third-order tensors for the rest of the analysis. Fig.2 shows an example of tensor ECG data for six classes. Following previous work on expending an ECG signal to a third-order tensor[35-37], we extend the original ECG signal to enable more effective extraction of valuable features in the spatial-spectral-temporal domain. To properly handle this type of data, a tensor-based learning approach is necessary. Fig.2 displays an example of six classes of tensor data.

For diagnostic purposes, the most useful portions of the ECG signal are the specific shapes of P, QRS, and T waves. Thus, useful features for classification are very sparse in the original tensor representation. Hence, the projection tensor should be sparse to allow the extraction of valuable features for classification purposes. Here, we plot the sparse tensor representation of the original six-class, 12-lead ECG signal corresponding to the sparse projection tensor in Fig.3. It appears that the sparse representation is more discriminative than the original tensor representation, and as such is clearly more suitable for classification.

In addition, we observe that tensor-based methods, especially the sparse version, can reduce the parameter count. To some extent, this prevents the overfitting problem from occurring. As for LDA-like methods, the well-known SSS problem can also be avoided. For instance, when given a tensor $\mathcal{X}$ of size N_1, N_2, N_3, we only need to apply the projection tensor $u_1 \otimes u_2 \otimes u_3$ and estimate $N_1 + N_2 + N_3$ parameters, instead of $N_1 \times N_2 \times N_3$. The parameter count can be even lower in the sparse case.

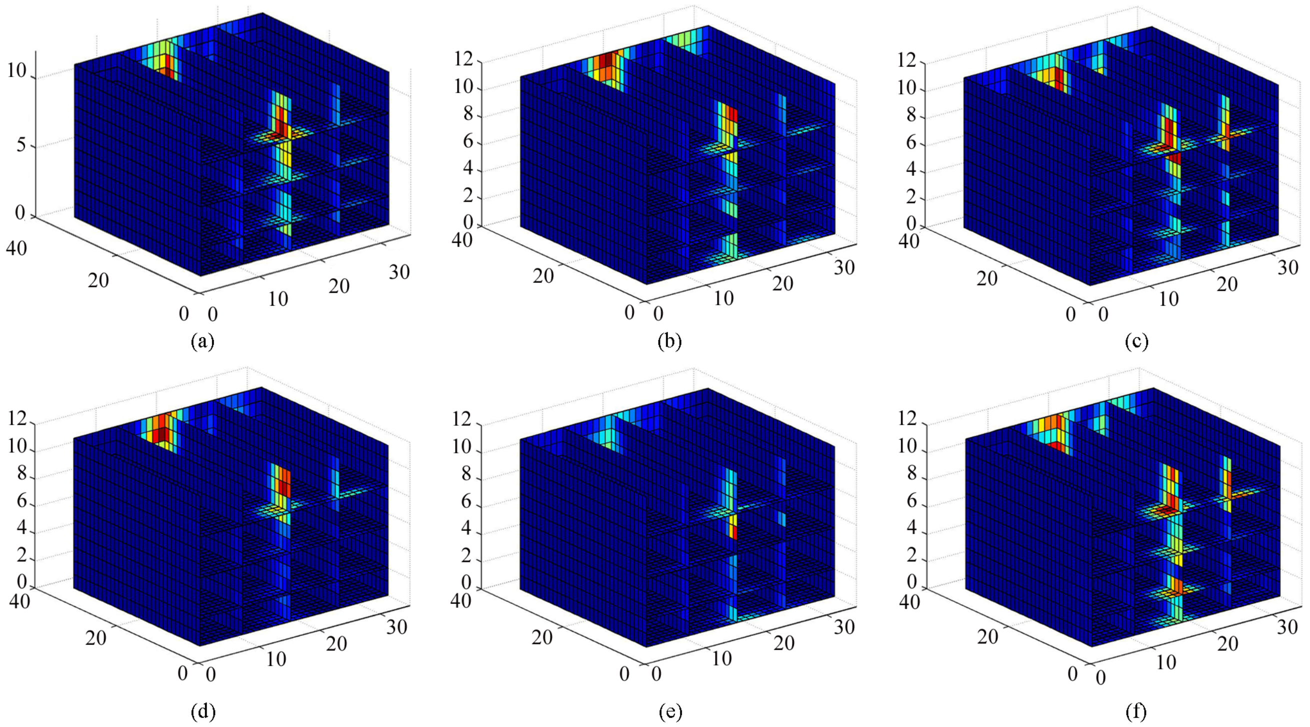
4 Semi-Supervised Sparse Discriminant Analysis

Classical LDA is to used to solve the following there equivalent optimization problems. The target is to map high-dimensional data into a subspace with lower dimension.

$$R(\boldsymbol{x}) = \frac{\boldsymbol{x}^{\mathrm{T}}\boldsymbol{S}_{\mathrm{b}}\boldsymbol{x}}{\boldsymbol{x}^{\mathrm{T}}\boldsymbol{S}_{\mathrm{w}}\boldsymbol{x}}, \frac{\boldsymbol{x}^{\mathrm{T}}\boldsymbol{S}_{\mathrm{b}}\boldsymbol{x}}{\boldsymbol{x}^{\mathrm{T}}\boldsymbol{S}_{\mathrm{t}}\boldsymbol{x}}, \frac{\boldsymbol{x}^{\mathrm{T}}\boldsymbol{S}_{\mathrm{w}}\boldsymbol{x}}{\boldsymbol{x}^{\mathrm{T}}\boldsymbol{S}_{\mathrm{t}}\boldsymbol{x}}.$$

$\boldsymbol{S}_{\mathrm{w}}$, $\boldsymbol{S}_{\mathrm{b}}$, and $\boldsymbol{S}_{\mathrm{t}}$ are the within-class scatter matrix, the between-class scatter matrix, and the total scatter matrix. These matrices can be calculated as

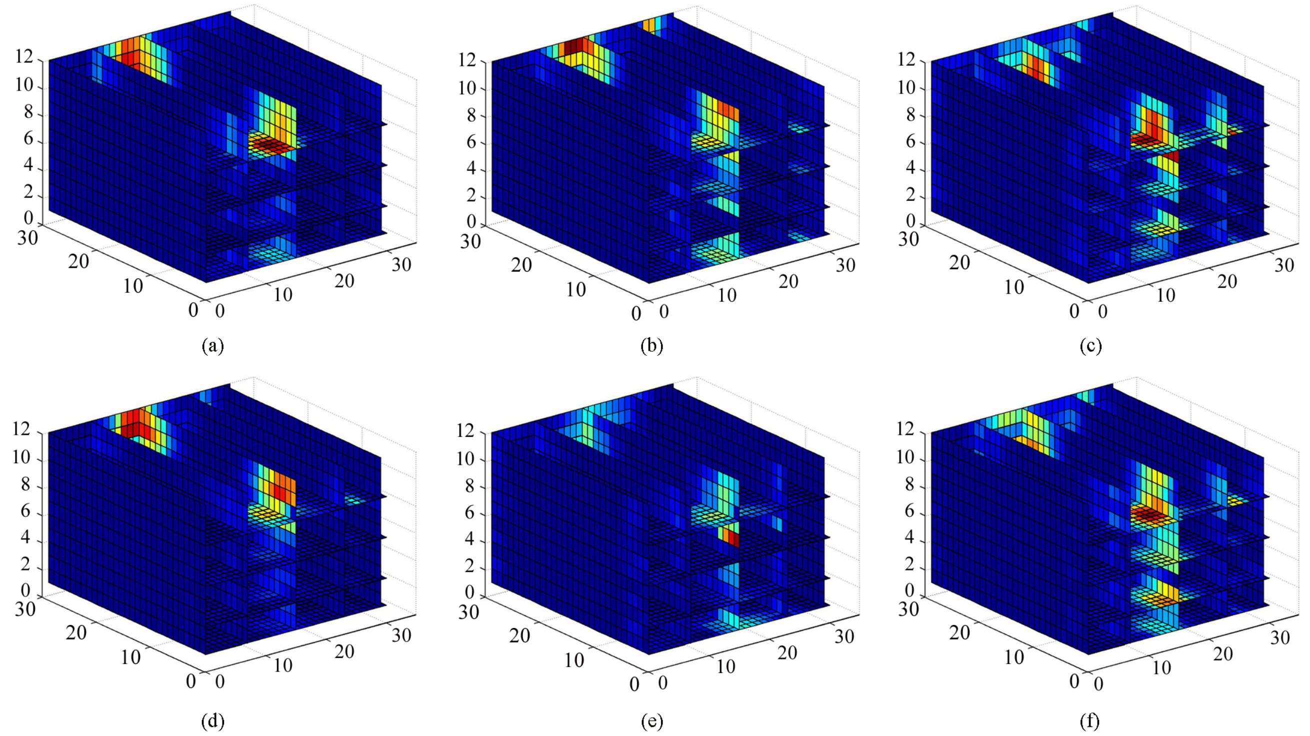
$$\boldsymbol{S}_{\mathrm{w}} = \frac{1}{n}\sum_{i=1}^{k}\sum_{\boldsymbol{x}\in A_i}(\boldsymbol{x}-\boldsymbol{c}^{(i)})(\boldsymbol{x}-\boldsymbol{c}^{(i)})^{\mathrm{T}},$$



(a) (b) (c)

(d) (e) (f)

Fig.2. Example of tensor ECG data in the spatial-spectral-temporal domain. (a) Class 1. (b) Class 2. (c) Class 3. (d) Class 4. (e) Class 5. (f) Class 6.



(a) (b) (c)

(d) (e) (f)

Fig.3. Example of tensor ECG data in the sparse spatial-spectral-temporal domain. (a) Class 1. (b) Class 2. (c) Class 3. (d) Class 4. (e) Class 5. (f) Class 6.

$$\begin{aligned}\boldsymbol{S}_{\mathrm{b}} &= \frac{1}{n}\sum_{i=1}^{k}\sum_{\boldsymbol{x}\in A_i}(\boldsymbol{c}^{(i)}-\boldsymbol{c})(\boldsymbol{c}^{(i)}-\boldsymbol{c})^{\mathrm{T}}\\ &= \frac{1}{n}\sum_{i=1}^{k} n_i(\boldsymbol{c}^{(i)}-\boldsymbol{c})(\boldsymbol{c}^{(i)}-\boldsymbol{c})^{\mathrm{T}},\\ \boldsymbol{S}_{\mathrm{t}} &= \frac{1}{n}\sum_{j=1}^{k}(\boldsymbol{x}_j-\boldsymbol{c})^{\mathrm{T}}(\boldsymbol{x}_j-\boldsymbol{c}).\end{aligned}$$

Here A_i means class i and n_i represents its point count. $\boldsymbol{c}^{(i)}$ is the sample mean of class i and $\boldsymbol{c}$ is the mean of all samples. It is easy to understand that $\boldsymbol{S}_{\mathrm{w}}$ is the sum of each class covariance matrix. $\boldsymbol{S}_{\mathrm{b}}$ is the weighted sum of the class mean covariance matrix where the weight is the number in each data class, and $\boldsymbol{S}_{\mathrm{t}}$ is the covariance matrix of all the points. Obtaining the equation $\boldsymbol{S}_{\mathrm{t}} = \boldsymbol{S}_{\mathrm{w}} + \boldsymbol{S}_{\mathrm{b}}$ is straightforward. $\boldsymbol{S}_{\mathrm{t}}$ is called the total scatter matrix.

$$\begin{aligned}\boldsymbol{S}_{\mathrm{b}} &= \sum_{k=1}^{c} l_k(\boldsymbol{u}^{(k)})(\boldsymbol{u}^{(k)})^{\mathrm{T}}\\ &= \sum_{k=1}^{c} l_k\Big(\frac{1}{l_k}\sum_{i=1}^{l_k}\boldsymbol{x}_i^{(k)}\Big)\Big(\frac{1}{l_k}\sum_{i=1}^{l_k}\boldsymbol{x}_i^{(k)}\Big)^{\mathrm{T}}\\ &= \sum_{k=1}^{c}\boldsymbol{X}^{(k)}\boldsymbol{U}^{(k)}(\boldsymbol{X}^{(k)})^{\mathrm{T}},\end{aligned}$$

where $\boldsymbol{U}^{(k)}$ is an $l_k \times l_k$ matrix with all the elements equal to l_k, $\boldsymbol{u}^{(k)}$ is the demeaned sample matrix of class k, and $\boldsymbol{X}^{(k)}$ is the original sample matrix of class k.

$$\boldsymbol{U}_{l\times l} = \begin{pmatrix}\boldsymbol{U}^{(1)} & 0 & \cdots & 0\\ 0 & \boldsymbol{U}^{(2)} & \cdots & 0\\ \vdots & \vdots & \ddots & \vdots\\ 0 & 0 & \cdots & \boldsymbol{U}^{(c)}\end{pmatrix}.$$

The final objective function is defined as:

$$\begin{aligned}\boldsymbol{a}_{\mathrm{opt}} &= \arg\max_{\boldsymbol{a}}\frac{\boldsymbol{a}^{\mathrm{T}}\boldsymbol{S}_{\mathrm{b}}\boldsymbol{a}}{\boldsymbol{a}^{\mathrm{T}}\boldsymbol{S}_{\mathrm{t}}\boldsymbol{a}}\\ &= \arg\max_{\boldsymbol{a}}\frac{\boldsymbol{a}^{\mathrm{T}}\boldsymbol{X}\boldsymbol{U}_{l\times l}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{a}}{\boldsymbol{a}^{\mathrm{T}}\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{a}}.\end{aligned}$$

Because the labeled data is very limited and the manual tagging task is expensive and time costing, we here consider the cluster character of unlabeled data to calculate the regularization item $\boldsymbol{J}_{\mathrm{W}}(\boldsymbol{a})$. In this way, we can achieve a better projection vector which makes the data cluster closely to each class. In addition, to further take advantage of the distribution nearing the boundary of different classes, the logistic label propagation as a semi-supervised method is used to give a category label to each unlabeled sample data and the between-class regularization item $\boldsymbol{J}_{\mathrm{B}}(\boldsymbol{a})$ is calculated to achieve better boundary separation. Thus we add these two items $\boldsymbol{J}_{\mathrm{B}}(\boldsymbol{a})$ and $\boldsymbol{J}_{\mathrm{W}}(\boldsymbol{a})$ to the original LDA target function. This is different from any existing work as usually only a single regularized item[30] is used:

$$\max_{\boldsymbol{a}}\frac{\boldsymbol{a}^{\mathrm{T}}\boldsymbol{S}_{\mathrm{b}}\boldsymbol{a}+\alpha\boldsymbol{J}_{\mathrm{B}}(\boldsymbol{a})}{\boldsymbol{a}^{\mathrm{T}}\boldsymbol{S}_{\mathrm{t}}\boldsymbol{a}+\beta\boldsymbol{J}_{\mathrm{W}}(\boldsymbol{a})}.$$

The calculations of these two items rely on an adjacent matrix[38-39]. In our case we take advantage of the labels generated by label propagation in order to work out the adjacent matrix[32]. Its definition is as follows:

$$S_{ij} = \begin{cases}1, & \text{if } \boldsymbol{x}_i \in N_p(\boldsymbol{x}_j) \text{ or } \boldsymbol{x}_j \in N_p(\boldsymbol{x}_i),\\ 0, & \text{otherwise.}\end{cases} \tag{1}$$

Here N_p means the set of p nearest points of the argument which is a point. Based on the semi-supervised learning label propagation algorithm[32,38-39], each unlabeled data is given a category label. Then two additional adjacent matrices are calculated. They are the between class adjacent matrices $\boldsymbol{B}$ and the within class adjacent matrix $\boldsymbol{W}$. The between class adjacent matrix $\boldsymbol{B}$ is defined by:

$$B_{ij} = \begin{cases}1, & \text{if } S_{ij}=1 \text{ and } \boldsymbol{x}_i,\boldsymbol{x}_j \notin \text{the same class},\\ 0, & \text{otherwise.}\end{cases}$$

The within class adjacent matrix $\boldsymbol{W}$ is as follows:

$$W_{ij} = \begin{cases}1, & \text{if } S_{ij}=1, \text{ and } \boldsymbol{x}_i,\boldsymbol{x}_j \in \text{ the same class},\\ 0, & \text{otherwise.}\end{cases}$$

Obviously, we have the following relation:

$$S_{ij} = B_{ij} + W_{ij}.$$

Hence if either of these matrices is computed, then the other one can be easily calculated by subtracting the matrix to the full adjacent matrix $\boldsymbol{S}$. With these two adjacent matrices, the two additional items in the objective function can be easily expressed as follows:

$$\begin{aligned}\boldsymbol{J}_{\mathrm{B}}(\boldsymbol{a}) &= \sum_{ij}(\boldsymbol{a}^{\mathrm{T}}\boldsymbol{x}_i-\boldsymbol{a}^{\mathrm{T}}\boldsymbol{x}_j)^2B_{ij},\\ \boldsymbol{J}_{\mathrm{W}}(\boldsymbol{a}) &= \sum_{ij}(\boldsymbol{a}^{\mathrm{T}}\boldsymbol{x}_i-\boldsymbol{a}^{\mathrm{T}}\boldsymbol{x}_j)^2W_{ij}.\end{aligned}$$

These two items can be expressed in a matrix form. The between class item is calculated as follows:

$$\begin{aligned}\boldsymbol{J}_{\mathrm{B}}(\boldsymbol{a}) &= \sum_{ij}(\boldsymbol{a}^{\mathrm{T}}\boldsymbol{x}_i-\boldsymbol{a}^{\mathrm{T}}\boldsymbol{x}_j)^2B_{ij}\\ &= 2\sum_{i}\boldsymbol{a}^{\mathrm{T}}\boldsymbol{x}_i\boldsymbol{D}_{B_{ii}}\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{a} - 2\sum_{ij}\boldsymbol{a}^{\mathrm{T}}\boldsymbol{x}_iB_{ij}\boldsymbol{x}_j^{\mathrm{T}}\boldsymbol{a}\\ &= 2\boldsymbol{a}^{\mathrm{T}}\boldsymbol{X}\boldsymbol{L}_{\mathrm{B}}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{a},\end{aligned}$$

where $\boldsymbol{D}$ is a diagonal matrix similar to $\boldsymbol{I}$, but its diagonal value is not 1. Its entries are column (or row, since $\boldsymbol{B}$ is symmetric) sum of $\boldsymbol{B}$, and $D_{ii} = \sum_j B_{ij}$. $\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{B}$ is the Laplacian matrix[40]. B in $\boldsymbol{L}_{\rm B}$ is the same as that in $\boldsymbol{J}_{\rm B}$. Then we have the within class item:

$$\begin{aligned}\boldsymbol{J}_{\rm W}(\boldsymbol{a}) &= \sum_{ij}(\boldsymbol{a}^{\rm T}\boldsymbol{x}_i - \boldsymbol{a}^{\rm T}\boldsymbol{x}_j)^2 B_{ij}\\ &= 2\sum_i \boldsymbol{a}^{\rm T}\boldsymbol{x}_i\boldsymbol{D}_{W_{ii}}\boldsymbol{x}_i^{\rm T}\boldsymbol{a} - 2\sum_{ij}\boldsymbol{a}^{\rm T}\boldsymbol{x}_i W_{ij}\boldsymbol{x}_j^{\rm T}\boldsymbol{a}\\ &= 2\boldsymbol{a}^{\rm T}\boldsymbol{X}\boldsymbol{L}_{\rm W}\boldsymbol{X}^{\rm T}\boldsymbol{a}.\end{aligned}$$

The final objective function with additional between class item and within class item is:

$$\begin{aligned}&\max_{\boldsymbol{a}}\frac{\boldsymbol{a}^{\rm T}\boldsymbol{S}_{\rm b}\boldsymbol{a} + \alpha\boldsymbol{J}_{\rm B}(\boldsymbol{a})}{\boldsymbol{a}^{\rm T}\boldsymbol{S}_{\rm t}\boldsymbol{a} + \beta\boldsymbol{J}_{\rm W}(\boldsymbol{a})}\\ &= \max_{\boldsymbol{a}}\frac{\boldsymbol{a}^{\rm T}(\boldsymbol{S}_{\rm b} + \alpha\boldsymbol{J}_{\rm B})\boldsymbol{a}}{\boldsymbol{a}^{\rm T}(\boldsymbol{S}_{\rm t} + \beta\boldsymbol{J}_{\rm W})\boldsymbol{a}},\end{aligned}$$

where α and β are the weights that can be adjusted to make a trade-off between the original LDA and our approach.

Since we choose to apply a semi-supervised method, these two additional items $\boldsymbol{B}$ and $\boldsymbol{W}$ use all the data including the labeled training data and the unlabeled testing data. However $\boldsymbol{S}_{\rm b}$ and $\boldsymbol{S}_{\rm t}$ only use the labeled training sample. To uniformly express the objective function over all the data $\boldsymbol{X}$, the original matrices $\boldsymbol{W}$ and $\boldsymbol{I}$ in the objective function must be extended.

$$\begin{aligned}\boldsymbol{U} &= \begin{pmatrix}\boldsymbol{U}_{l\times l} & 0\\ 0 & 0\end{pmatrix},\\ \widetilde{\boldsymbol{I}} &= \begin{pmatrix}\boldsymbol{I} & 0\\ 0 & 0\end{pmatrix},\end{aligned}$$

where $\boldsymbol{I}$ is the identical matrix.

From the above derivation, the objective function can be expressed following a uniform equation:

$$\max_{\boldsymbol{a}}\frac{\boldsymbol{a}^{\rm T}(\boldsymbol{X}\boldsymbol{U}\boldsymbol{X}^{\rm T} + \alpha\boldsymbol{X}\boldsymbol{L}_{\rm B}\boldsymbol{X}^{\rm T})\boldsymbol{a}}{\boldsymbol{a}^{\rm T}(\boldsymbol{X}\widetilde{\boldsymbol{I}}\boldsymbol{X}^{\rm T} + \beta\boldsymbol{X}\boldsymbol{L}_{\rm W}\boldsymbol{X}^{\rm T})\boldsymbol{a}}.$$

It can be transformed into a Maximum Scatter Difference Discriminant Analysis problem:

$$\max_{\boldsymbol{a}}\boldsymbol{a}^{\rm T}(\boldsymbol{X}(\boldsymbol{U} - \widetilde{\boldsymbol{I}} + \alpha\boldsymbol{L}_{\rm B} - \beta\boldsymbol{L}_{\rm W})\boldsymbol{X}^{\rm T})\boldsymbol{a}. \qquad (2)$$

Here we have changed the target function for a Maximum Scatter Difference Discriminant Analysis problem[41-42]. Thus it is unnecessary to use the sparse discriminant analysis approach. Therefore we use the sparse PCA approach to calculate the target discriminant projection vectors[27].

Here $\boldsymbol{V}[, 1 : k]$ is the first k principal components. Given a fixed $\boldsymbol{A} = (\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_k)$, we solve the elastic net problem for $j = 1, 2, \ldots, k$.

$$\begin{aligned}\boldsymbol{\beta}_j = \arg\min_{\boldsymbol{\beta}}(\boldsymbol{\alpha}_j - \boldsymbol{\beta})^{\rm T}\boldsymbol{X}^{\rm T}\boldsymbol{X}(\boldsymbol{\alpha}_j - \boldsymbol{\beta}) +\\ \lambda\|\boldsymbol{\beta}\|^2 + \lambda_{1,j}\|\boldsymbol{\beta}\|_1.\end{aligned}$$

For a fixed $\boldsymbol{B} = (\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_1)$, we compute the SVD (singular value decomposition) of $\boldsymbol{X}\boldsymbol{X}^{\rm T}\boldsymbol{B} = \boldsymbol{U}\boldsymbol{D}\boldsymbol{V}^{\rm T}$ and then update $\boldsymbol{A} = \boldsymbol{U}\boldsymbol{V}^{\rm T}$. By repeating these two steps until it converges, the projection vector can be computed.

To use the above algorithm, we have to apply the Cholesky decomposition to the *innermatrix* $\boldsymbol{A} = \boldsymbol{U} - \widetilde{\boldsymbol{I}} + \boldsymbol{L}_{\rm B} - \boldsymbol{L}_{\rm W}$. But first, the innermatrix must be regularized to be positive definite (so we must make all the eigenvalue not smaller than $\mathbf{0}$ which is listed as the third equation below):

$$\begin{aligned}&[\boldsymbol{V}, \boldsymbol{D}] = \mathit{eig}(\boldsymbol{A});\\ &\boldsymbol{d} = \mathit{diag}(\boldsymbol{D});\\ &(\boldsymbol{d} \leqslant \mathbf{0}) = \min(\boldsymbol{d} > \mathbf{0});\\ &\boldsymbol{A_r} = \boldsymbol{V}\cdot\mathit{diag}(\boldsymbol{d})\cdot\boldsymbol{V}^{\rm T}.\end{aligned}$$

Because the eigenvalue $\boldsymbol{d}$ is not always larger than $\mathbf{0}$, we set the one which is not larger than $\mathbf{0}$ to be the minimum of eigenvalue which is larger than $\mathbf{0}$. Then the new innermatrix $\boldsymbol{A_r}$ can be used using the Cholesky decomposition $\boldsymbol{A_r} = \boldsymbol{A}_c\cdot\boldsymbol{A}_c^{\rm T}$. Next we multiply $\boldsymbol{X}$ by $\boldsymbol{A}_c$ and get $\boldsymbol{X}' = \boldsymbol{A}_c\cdot\boldsymbol{X}$. Finally we can solve the elastic net problem as follows:

$$\begin{aligned}\boldsymbol{\beta}_j = \arg\min_{\boldsymbol{\beta}}(\boldsymbol{\alpha}_j - \boldsymbol{\beta})^{\rm T}\boldsymbol{X}^{\rm T}\boldsymbol{X}(\boldsymbol{\alpha}_j - \boldsymbol{\beta}) +\\ \lambda\|\boldsymbol{\beta}\|^2 + \lambda_{1,j}\|\boldsymbol{\beta}\|_1,\end{aligned} \qquad (3)$$

where vector $\boldsymbol{\alpha}_j$ is the j-th principal component. Here $\boldsymbol{\beta}_k$ is the expected result as $\boldsymbol{\mu}_{k-1}$. It means $\boldsymbol{\beta}_k$ will converge to $\boldsymbol{\mu}_{k-1}$. Here we must use $\boldsymbol{\mu}_{k-1}$ to represent the result. The original LDA rises $(c-1)$ low rank problems, which means that it can only get $c-1$ projection vector corresponding to nonzero eigenvalues. We use a similar method as that in complementary space LDA to overcome the limitation[21,43-44]. We repeat the process each time we get a projection vector.

$$\begin{aligned}&\lambda^k = \boldsymbol{x}\boldsymbol{\mu}_{k-1},\\ &\boldsymbol{x}^1 = \boldsymbol{x},\\ &\boldsymbol{x}^k = \boldsymbol{x}^{k-1} - \lambda^k\cdot\boldsymbol{\mu}_{k-1}.\end{aligned}$$

We actually calculate one projection vector each time and then the original data is adjusted for the calculation of the next projection vector.

$$\begin{aligned}\boldsymbol{\beta} = \arg\min_{\boldsymbol{\beta}} &(\boldsymbol{\alpha}-\boldsymbol{\beta})^{\mathrm{T}}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}(\boldsymbol{\alpha}-\boldsymbol{\beta})+\\ &\lambda\|\boldsymbol{\beta}\|^2+\lambda_{1,j}\|\boldsymbol{\beta}\|_1,\end{aligned} \tag{4}$$

where $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are matrixes different from the above definition. In above equations, we calculate each vector in $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ one by one. Here we just simplify (3) to be a matrix form (4).

Following this strategy our method will not be confronted to the $c-1$ rank low problem, and it can calculate as many projection vectors as the dimension of the original data vector.

5 Semi-Supervised Sparse Multilinear Discriminant Analysis

5.1 Algorithm

For 12-lead ECG or other high-dimensional tensor data, using the vector space algorithm by expanding the original tensor data into vectors is not a good choice as it will introduce the SSS and the overfitting problems. In addition, the structure information will be lost during the vectorization process. Therefore, designing a multilinear or tensor version algorithm which takes tensor data as direct input is useful and necessary. We proposed a multilinear version of the semi-supervised sparse algorithm in the previous section. The original $\boldsymbol{S}_{\mathrm{t}}$ and $\boldsymbol{S}_{\mathrm{b}}$ in the original LDA can be easily transformed into a multilinear version by replacing $\boldsymbol{x}$ with $\mathcal{X}$ based on an analogy with (2).

$$\boldsymbol{P} = \begin{pmatrix} \left(\dfrac{1}{n}\displaystyle\sum_{i=1}^{c}\left((\mathcal{M}_i^k-\mathcal{M}^k)\prod_{l=1}^{M}\times_l(\boldsymbol{u}_k^l)^{\mathrm{T}}\right)\times\right.\\ \left((\mathcal{M}_i^k-\mathcal{M}^k)\displaystyle\prod_{l=1}^{M}\times_l(\boldsymbol{u}_k^l)^{\mathrm{T}}\right)^{\mathrm{T}}-\\ \zeta_k^l\displaystyle\sum_{j=1}^{n}\left((\mathcal{X}_{ji}^k-\mathcal{M}_i^k)\prod_{l=1}^{M}\times_l(\boldsymbol{u}_k^l)^{\mathrm{T}}\right)\times\\ \left.\left((\mathcal{X}_{ji}^k-\mathcal{M}_i^k)\displaystyle\prod_{l=1}^{M}\times_l(\boldsymbol{u}_k^l)^{\mathrm{T}}\right)^{\mathrm{T}}\right) \end{pmatrix}, \tag{5}$$

where $\mathcal{X}$ and $\mathcal{M}$ are both tensors.

The most important two additional items $\boldsymbol{J}_{\mathrm{B}}$ and $\boldsymbol{J}_{\mathrm{W}}$ can be transformed by following a similar strategy. The transformed form of additional between class item $\boldsymbol{J}_{\mathrm{B}}$ is as follows:

$$\boldsymbol{J}_{\mathrm{B}} = \left(\sum_{ij}\begin{matrix} B_{ij}\left((\mathcal{X}_i-\mathcal{X}_j)\displaystyle\prod_{l=1}^{M}\times_l(\boldsymbol{u}_k^l)^{\mathrm{T}}\right)\times\\ \left((\mathcal{X}_i-\mathcal{X}_j)\displaystyle\prod_{l=1}^{M}\times_l(\boldsymbol{u}_k^l)^{\mathrm{T}}\right)^{\mathrm{T}}\end{matrix}\right).$$

The additional within class item $\boldsymbol{J}_{\mathrm{W}}$ should be transformed in a similar way.

$$\boldsymbol{J}_{\mathrm{W}} = \left(\sum_{ij}\begin{matrix} W_{ij}\left((\mathcal{X}_i-\mathcal{X}_j)\displaystyle\prod_{l=1}^{M}\times_l(\boldsymbol{u}_k^l)^{\mathrm{T}}\right)\times\\ \left((\mathcal{X}_i-\mathcal{X}_j)\displaystyle\prod_{l=1}^{M}\times_l(\boldsymbol{u}_k^l)^{\mathrm{T}}\right)^{\mathrm{T}}\end{matrix}\right).$$

By adding the $\boldsymbol{J}_{\mathrm{B}}$ and $\boldsymbol{J}_{\mathrm{W}}$ items to (5), the objective function of SSSMDA is defined by:

$$\boldsymbol{u}_k^l|_{l=1}^{M} = \arg_{\boldsymbol{u}_k^l|_{l=1}^{M}}\max(\boldsymbol{P}+\boldsymbol{J}_{\mathrm{B}}-\boldsymbol{J}_{\mathrm{W}}).$$

As the objective function above has no close form solution, an alternate projection method is adopted. Here $\boldsymbol{J}_{\mathrm{B}}$ and $\boldsymbol{J}_{\mathrm{W}}$ can be transformed:

$$\begin{aligned}
\boldsymbol{J}_{\mathrm{B}} &= \left(\sum_{ij}\begin{matrix} B_{ij}((\mathcal{X}_i-\mathcal{X}_j)\bar{\times}_l(\boldsymbol{u}_k^l)^{\mathrm{T}}\times_l(\boldsymbol{u}_k^l)^{\mathrm{T}})\times\\ ((\mathcal{X}_i-\mathcal{X}_j)\bar{\times}_l(\boldsymbol{u}_k^l)^{\mathrm{T}}\times_l(\boldsymbol{u}_k^l)^{\mathrm{T}})^{\mathrm{T}}\end{matrix}\right),\\
\boldsymbol{J}_{\mathrm{W}} &= \left(\sum_{ij}\begin{matrix} W_{ij}((\mathcal{X}_i-\mathcal{X}_j)\bar{\times}_l(\boldsymbol{u}_k^l)^{\mathrm{T}}\times_l(\boldsymbol{u}_k^l)^{\mathrm{T}})\times\\ ((\mathcal{X}_i-\mathcal{X}_j)\bar{\times}_l(\boldsymbol{u}_k^l)^{\mathrm{T}}\times_l(\boldsymbol{u}_k^l)^{\mathrm{T}})^{\mathrm{T}}\end{matrix}\right)\\
&= \boldsymbol{u}_k^l\left(\sum_{ij}\begin{matrix} B_{ij}((\mathcal{X}_i-\mathcal{X}_j)\bar{\times}_l(\boldsymbol{u}_k^l)^{\mathrm{T}})\times\\ ((\mathcal{X}_i-\mathcal{X}_j)\bar{\times}_l(\boldsymbol{u}_k^l)^{\mathrm{T}})^{\mathrm{T}}\end{matrix}\right)(\boldsymbol{u}_k^l)^{\mathrm{T}}\\
&= \boldsymbol{u}_k^l\left(\sum_{ij}\begin{matrix} W_{ij}((\mathcal{X}_i-\mathcal{X}_j)\bar{\times}_l(\boldsymbol{u}_k^l)^{\mathrm{T}})\times\\ ((\mathcal{X}_i-\mathcal{X}_j)\bar{\times}_l(\boldsymbol{u}_k^l)^{\mathrm{T}})^{\mathrm{T}}\end{matrix}\right)(\boldsymbol{u}_k^l)^{\mathrm{T}},\\
\boldsymbol{J}'_{\mathrm{B}} &= B_{ij}((\mathcal{X}_i-\mathcal{X}_j)\overline{\mathcal{X}}_l(\boldsymbol{u}_k^l)^{\mathrm{T}}\times((\mathcal{X}_i-\mathcal{X}_j)\overline{\mathcal{X}}_l(\boldsymbol{u}_k^l)^{\mathrm{T}})^{\mathrm{T}},\\
\boldsymbol{J}'_{\mathrm{W}} &= W_{ij}((\mathcal{X}_i-\mathcal{X}_j)\overline{\mathcal{X}}_l(\boldsymbol{u}_k^l)^{\mathrm{T}})\times((\mathcal{X}_i-\mathcal{X}_j)\overline{\mathcal{X}}_l(\boldsymbol{u}_k^l)^{\mathrm{T}})^{\mathrm{T}}.
\end{aligned}$$

Here $\overline{\mathcal{X}}_l$ means multiplying the tensor in all the modes except model l. By substituting the equation in the bracket:

$$\boldsymbol{J}_{\mathrm{B}}-\boldsymbol{J}_{\mathrm{W}} = \boldsymbol{u}_k^l(\boldsymbol{J}'_{\mathrm{B}}-\boldsymbol{J}'_{\mathrm{W}})(\boldsymbol{u}_k^l)^{\mathrm{T}}.$$

The item $\boldsymbol{P}$ in the objective function corresponding to $\boldsymbol{S}_{\mathrm{t}}$ and $\boldsymbol{S}_{\mathrm{b}}$ in the original LDA can also be similarly transformed:

$$\boldsymbol{P}' = \begin{pmatrix} \Big(\dfrac{1}{n}\displaystyle\sum_{i=1}^{c}((\mathcal{M}_i^k-\mathcal{M}^k)\bar{\times}_l(\boldsymbol{u}_k^l)^{\mathrm{T}})\times\\ ((\mathcal{M}_i^k-\mathcal{M}^k)\bar{\times}_l(\boldsymbol{u}_k^l)^{\mathrm{T}})^{\mathrm{T}}-\\ \zeta_k^l\displaystyle\sum_{j=1}^{n_i}((\mathcal{X}_{ji}^k-\mathcal{M}_i^k)\bar{\times}_l(\boldsymbol{u}_k^l)^{\mathrm{T}})\times\\ ((\mathcal{X}_{ji}^k-\mathcal{M}_i^k)\bar{\times}_l(\boldsymbol{u}_k^l)^{\mathrm{T}})^{\mathrm{T}}\Big) \end{pmatrix}.$$

The optimization problem of SSSMDA is transformed into m subproblems, where m is the mode count of the original data.

$$\arg\max_{\boldsymbol{u}_k^l} = \left(\boldsymbol{u}_k^l(\boldsymbol{P}'+\boldsymbol{J}'_{\mathrm{B}}-\boldsymbol{J}'_{\mathrm{W}})(\boldsymbol{u}_k^l)^{\mathrm{T}}\right).$$

In the alternate projection process, the original data is multiplied by the projection tensor except one mode as $\mathcal{X}_i \bar{\times}_l (\boldsymbol{u}_k^l)^{\mathrm{T}}$ forms a matrix in which each column represents one data as $\boldsymbol{X}$. The whole matrix is $\mathcal{X}_{\mathrm{all}}$. In this way, we can reformulate the algorithm as follows:

$$\max_{\boldsymbol{a}} \boldsymbol{a}^{\mathrm{T}}(\mathcal{X}(\boldsymbol{U} - \tilde{\boldsymbol{I}} + \boldsymbol{L}_{\mathrm{B}} - \boldsymbol{L}_{\mathrm{W}})\mathcal{X}^{\mathrm{T}})\boldsymbol{a}.$$

It is similar to the semi-supervised sparse discriminant analysis. In the process of calculating each mode in the alternate projection, $\boldsymbol{U}$, $\boldsymbol{I}$, $\boldsymbol{B}$, $\boldsymbol{W}$ remain unchanged.

$$((\mathcal{X}_{\mathrm{all}})\bar{\times}_l(\boldsymbol{u}_k^l)^{\mathrm{T}}) \times (\boldsymbol{U} - \tilde{\boldsymbol{I}} + \boldsymbol{L}_{\mathrm{B}} - \boldsymbol{L}_{\mathrm{W}})\times \\ ((\mathcal{X}_{\mathrm{all}})\bar{\times}_l(\boldsymbol{u}_k^l)^{\mathrm{T}})^{\mathrm{T}}.$$

Here we get a target function which is similar to the semisupervised sparse discriminant analysis. Thus for our case we only adopt the calculation method.

$$\begin{aligned} &[\boldsymbol{V}, \boldsymbol{D}] = eig(\boldsymbol{A}); \\ &\boldsymbol{d} = diag(\boldsymbol{D}); \\ &(\boldsymbol{d} \leqslant \boldsymbol{0}) = \min(\boldsymbol{d} > \boldsymbol{0}); \\ &\boldsymbol{A_r} = \boldsymbol{V} \cdot diag(\boldsymbol{d}) \cdot \boldsymbol{V}^{\mathrm{T}}. \end{aligned}$$

We now apply the Cholesky decomposition to $\boldsymbol{A_r} = \boldsymbol{A}_c \cdot \boldsymbol{A}_c^{\mathrm{T}}$. Substituting it into the original equation, we get $\mathcal{X}' = \boldsymbol{A}_c \cdot ((\mathcal{X}_j^k)\bar{\times}_l(\boldsymbol{u}_k^l)^{\mathrm{T}})$. Then the objective function can still be solved through the generalized elastic net problem.

The target function of sparse discriminant analysis is defined sequentially:

$$\boldsymbol{\beta} = \arg\min_{\boldsymbol{\beta}} (\boldsymbol{\alpha} - \boldsymbol{\beta})^{\mathrm{T}} \boldsymbol{X'}^{\mathrm{T}} \boldsymbol{X'} (\boldsymbol{\alpha} - \boldsymbol{\beta}) + \\ \lambda \|\boldsymbol{\beta}\|^2 + \lambda_{1,j}\|\boldsymbol{\beta}\|_1.$$

Here we still use the alternative optimization in order to solve the problem. Then a method similar to the one used in the complementary space computation is adopted to process the original tensor data[21,43-44].

$$\mathcal{X}_{ij}^k = \mathcal{X}_{ij}^{k-1} - \lambda^{k-1}\boldsymbol{u}_{k-1}^1 \otimes \boldsymbol{u}_{k-1}^2 \otimes \cdots \otimes \boldsymbol{u}_{k-1}^M.$$

The algorithm we use is described in Algorithm 1.

As previously mentioned, we know that the original LDA encounters the small size problem, the $(c-1)$ low rank problem, the heteroscedastic problem and the unreasonable between-class scatter matrix. The proposed method is developed such as to tackle these issues. In addition, our approach fully considers the distribution structure of both labeled and unlabeled data such that it calculates a better projection tensor for classification purpose.

Algorithm 1. Semi-Supervised Sparse Multilinear Discriminant Analysis

Input: training tensors $\mathcal{X}_{ij}$, $1 \leqslant i \leqslant c$, $1 \leqslant j \leqslant n_i$, the number R of rank-1 tensors allowed in SSSMDA, and the tuning parameters

Output: the projection vectors $\boldsymbol{u}_k^l$, $1 \leqslant l \leqslant M$

```
 1: Set $\mathcal{X}_{i,j}^l = \mathcal{X}_{i,j}$, $1 \leqslant i \leqslant c$, $1 \leqslant j \leqslant n_i$, $\boldsymbol{u}_k^d$ = Optimal $\boldsymbol{u}_k^d$
 2: $\boldsymbol{U} - \tilde{\boldsymbol{I}} + \boldsymbol{L}_{\mathrm{B}} - \boldsymbol{L}_{\mathrm{W}} = \boldsymbol{A}_c \cdot \boldsymbol{A}_c^{\mathrm{T}}$
 3: for $k = 1$ to $R$ do
 4:   Calculate $\mathcal{X}_{i,j}^k = \mathcal{X}_{i,j}^{k-1} - \lambda_{i,j}^{k-1} \prod_{l=1}^M \otimes \boldsymbol{u}_{k-1}^l$ with
      $\lambda_{i,j}^{k-1} = \mathcal{X}_{i,j}^{k-1} \prod_{l=1}^M \times_l \boldsymbol{u}_{k-1}^l$
 5:   for $t = 1$ to $L$ do
 6:     for $l = 1$ to $M$ do
 7:       $\mathcal{X}' = \boldsymbol{A}_c \cdot ((\mathcal{X}_j^k)\bar{\times}_l(\boldsymbol{u}_k^l)^{\mathrm{T}})$
 8:       $((\mathcal{X}_{\mathrm{all}})\bar{\times}_l(\boldsymbol{u}_k^l)^{\mathrm{T}}) \times (\boldsymbol{U} - \tilde{\boldsymbol{I}} + \boldsymbol{L}_{\mathrm{B}} - \boldsymbol{L}_{\mathrm{W}})\times$
          $((\mathcal{X}_{\mathrm{all}})\bar{\times}_l(\boldsymbol{u}_k^l)^{\mathrm{T}})^{\mathrm{T}}$
          % this result can be easily got by $\boldsymbol{X'}^{\mathrm{T}} \cdot \boldsymbol{X'}$
 9:       % elastic net problem
          $\boldsymbol{\beta} = \arg\min_{\boldsymbol{\beta}} (\boldsymbol{\alpha} - \boldsymbol{\beta})^{\mathrm{T}} \boldsymbol{X'}^{\mathrm{T}} \boldsymbol{X'} (\boldsymbol{\alpha} - \boldsymbol{\beta}) + \lambda\|\boldsymbol{\beta}\|^2 +$
          $\lambda_{l,j}\|\boldsymbol{\beta}\|_1$
10:     end for % for loop in step 4
11:     Convergence check: If $\|\boldsymbol{u}_k^l - \boldsymbol{u}_{k-1}^l\|_F \leqslant \varepsilon$ for all
        directions $l$ in the $k$-th iteration, stop the loop in
        step 3.
12:   end for % for loop in step 3.
13: end for % for loop in step 2.
```

To make a summary, we take unlabeled data into consideration to calculate the regularization item. We first use a logistic label propagation to assign a category label to each unlabeled sample. And then the within-class regularization item and the between-class regularization item are calculated to get better projection vector. The reason why it works is that the cluster character in addition to the distribution of different classes are considered. Therefore, better projection vector can be achieved and very few training samples with label are needed for the calculation method. In this way, with fewer training samples, the cost of calculation is reduced and the model can be with more generation power. Also our method takes the tensor data as direct input with less parameters to be determined. Consequently, according to the Occam's Razor principle, the over-fitting problem can be relieved. Thus the combination of semi-supervised learning and tensor learning model makes our method more effective.

5.2 Convergence Issue

Another important aspect of our algorithm is the convergence. In this paper, the convergence issue is analyzed in a similar way as papers of Tao *et al.*[17,45]

Indeed as we use the alternative projection method, it can be proved that our algorithm is monotonic, that is, the target function with the $\boldsymbol{\mu}_k^l$ achieved by each iteration is monotonically decreasing. We give a definition of the target function with respect to the mode and iteration numbers:

$$\begin{aligned}\arg\max_{\boldsymbol{u}_k^l} &= \big(\boldsymbol{u}_k^l(\boldsymbol{P}' + \boldsymbol{J}'_{\boldsymbol{B}} - \boldsymbol{J}'_{\boldsymbol{W}})(\boldsymbol{u}_k^l)^{\mathrm{T}}\big),\\ F(\boldsymbol{u}_k^l, k) &= \big(\boldsymbol{u}_k^l(\boldsymbol{P}' + \boldsymbol{J}'_{\boldsymbol{B}} - \boldsymbol{J}'_{\boldsymbol{W}})(\boldsymbol{u}_k^l)^{\mathrm{T}}\big),\end{aligned}$$

where k is the iteration number and l is the mode number. Our algorithm generates a sequence of objective function value with each mode l and iteration k. The sequence is as follows:

$$\begin{aligned}F(\boldsymbol{u}_k^1, 1) &\leqslant F(\boldsymbol{u}_k^2, 1) \leqslant \cdots \leqslant F(\boldsymbol{u}_k^M, 1)\\ &\leqslant F(\boldsymbol{u}_k^1, 2) \leqslant F(\boldsymbol{u}_k^2, 2) \leqslant \cdots \leqslant F(\boldsymbol{u}_k^1, k)\\ &\leqslant F(\boldsymbol{u}_k^2, k) \leqslant \cdots \leqslant F(\boldsymbol{u}_k^1, K) \leqslant F(\boldsymbol{u}_k^2, K)\\ &\leqslant \cdots \leqslant F(\boldsymbol{u}_k^M, K).\end{aligned}$$

The alternate projection algorithm is actually a composition of M sub-algorithms. To check the convergence at each step and whether the algorithm should be stopped, we solve the following equation and compare the result with a given threshold.

$$\Big\|\prod_{l=1}^{M}\otimes(\boldsymbol{u}_k^l)^{\mathrm{T}} - \prod_{l=1}^{M}\otimes(\boldsymbol{u}_{k-1}^l)^{\mathrm{T}}\Big\|_F.$$

This method allows us to determine whether the algorithm converges or not and then to terminate the entire algorithm.

5.3 Computational Complexity

To evaluate the actual performance of our algorithm, we examine the computational complexity and memory requirements, which provide relative measures of its practicality and usefulness. We study the computational issues in a fashion similar to that introduced in [18].

Because this is an iterative solution, the computational complexity analysis considers a single iteration. For simplicity, it is assumed that $I_1 = I_2 = \cdots = I_M = (\prod_{m=1}^{M} I_m)^{\frac{1}{M}} = I$. Here, M is the mode count of the tensor data. From a computational complexity point of view, the most demanding steps are the formation of the matrices $((\mathcal{X}_{\mathrm{all}})\bar{\times}_l(\boldsymbol{u}_k^l)^{\mathrm{T}}) \cdot (\boldsymbol{U} - \tilde{\boldsymbol{I}} + \boldsymbol{L}_{\boldsymbol{B}} - \boldsymbol{L}_{\boldsymbol{W}}) \cdot ((\mathcal{X}_{\mathrm{all}})\bar{\times}_l(\boldsymbol{u}_k^l)^{\mathrm{T}})^{\mathrm{T}}$. First, we must calculate $(\boldsymbol{U} - \tilde{\boldsymbol{I}} + \boldsymbol{L}_{\boldsymbol{B}} - \boldsymbol{L}_{\boldsymbol{W}})$. The time complexity of these matrices is $O(n), O(n), O(n^2 \times I^M)$, and $O(n^2 \times I^M)$, respectively, where n is the number of sample data. This matrix experiences a Cholesky decomposition $\boldsymbol{A}_{\boldsymbol{r}} = \boldsymbol{A}_c \cdot \boldsymbol{A}_c^{\mathrm{T}}$, which has $O(n^3)$ time cost. Then, the matrix $\boldsymbol{X}' = \boldsymbol{A}_c \cdot ((\mathcal{X}_j^k)\bar{\times}_l(\boldsymbol{u}_k^l)^{\mathrm{T}})$ must be calculated. The calculation of $((\mathcal{X}_{\mathrm{all}})\bar{\times}_l(\boldsymbol{u}_k^l)^{\mathrm{T}})$ has a time cost of $(M-1) \times I(M+1)$. To obtain $\boldsymbol{X}'$, the time cost is $n^2 I$. The calculation of the $I \times I$ matrix $\boldsymbol{X}'^{\mathrm{T}}\boldsymbol{X}'$ requires nI^2 operations. $\boldsymbol{X}'^{\mathrm{T}}\boldsymbol{X}'\beta$ costs I^2, and the singular value decomposition of $\boldsymbol{X}'^{\mathrm{T}}\boldsymbol{X}'\beta$ is of order $O(I)$. Each elastic net solution requires at most $O(I^3)$ operations. To calculate $\mathcal{X}_{i,j}^k = \mathcal{X}_{i,j}^{k-1} - \lambda_{i,j}^{k-1} \prod_{l=1}^{M} \otimes \boldsymbol{u}_{k-1}^l$ with $\lambda_{i,j}^{k-1} = \mathcal{X}_{i,j}^{k-1} \prod_{l=1}^{M} \times_l \boldsymbol{u}_{k-1}^l$, the time cost is $2 \times I^n + 2 \times I^{n-1} + \cdots + 2 \times I$, which is $O(I^n)$. The total complexity is $O(n^2 \times I^M + n^3 + R(n \times I^M + L \times M \times ((M-1) \times I^{(M+1)} + n^2 I + nI^2 + tI^3)))$, where t is the number of iterations before the convergence of one elastic net problem, L is the number of iterations needed by our algorithm, and R is the projection tensor count of our algorithm.

As for the memory requirements of the SSSMDA algorithm, because many calculations are performed incrementally, the time cost is not especially high. At first, the matrix $(\boldsymbol{U} - \tilde{\boldsymbol{I}} + \boldsymbol{L}_{\mathrm{B}} - \boldsymbol{L}_{\mathrm{W}})$ should be calculated. The calculation process of $\boldsymbol{L}_{\mathrm{B}} - \boldsymbol{L}_{\mathrm{W}}$ requires n^2 space to store the distance matrix. Another $2n^2$ space is needed to store the adjacent matrix. To store $\boldsymbol{U}$ and $\tilde{\boldsymbol{I}}$ requires $2n^2$ space, and calculating the Cholesky decomposition takes a further n^2 space. To compute $\mathcal{X}_{i,j}^k = \mathcal{X}_{i,j}^{k-1} - \lambda_{i,j}^{k-1} \prod_{l=1}^{M} \otimes \boldsymbol{u}_{k-1}^l$ with $\lambda_{i,j}^{k-1} = \mathcal{X}_{i,j}^{k-1} \prod_{l=1}^{M} \times_l \boldsymbol{u}_{k-1}^l$, we require memory space of $O(I^M)$. Other steps in the calculation need $O(I^M)$ memory space, because $\mathcal{X}$ can be read into memory sequentially without loss of information. Thus, the total memory requirement is $O(n^2 + I^M)$.

6 Experiments and Results

6.1 Experiments on a Toy Dataset

Here we use a synthetic toy dataset to discuss the effectiveness of our method that takes advantage of the distribution of unlabeled data. We assign the labels to unlabeled data using label propagation[32]. In this way, the distribution near the boundary of labeled and unlabeled data is considered. We generate three classes of non-Gaussian three-dimensional (3D) data whose selected sample is shown in Fig.4. Class 1 is a Gaussian sample cluster, and classes 2 and 3 consist of two Gaussian sample clusters. To explore the nature of our tensor feature extraction method and compare the performance of our method to other vector space and tensor-based approaches, we group 10 local 3D samples to form a 10 × 3 tensor sample.

To validate the effectiveness of the semisupervised model, the generated samples are divided into a trai-

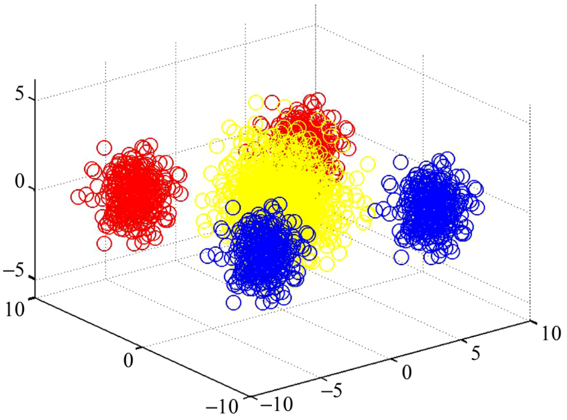


Fig.4. Three classes of 3D non-Gaussian distributions.

ning sample set with labels and a testing sample set without labels. We generate 110 000 random sample vectors for each class of which 100 000 are allocated to the testing sample set and 10 000 are reserved for the training sample set. Because 10 samples form one tensor sample, we generate a total of 1 000 training tensor samples and 10 000 testing tensor samples.

The classification accuracy of our method compared with other methods is given in Table 1. It can be seen that our semisupervised model is effective on the toy dataset.

Table 1. Classification Accuracy of Different Approaches

Approach	Classification Accuracy (%)
PCA	76.2
ICA	75.1
LDA	76.8
UMP	82.9
TR1	85.6
SSM	92.3

6.2 12-Lead ECG Database

To evaluate the performance of our method, we test the proposed method on a large dataset[①] collected from a local hospital. Our database is provided by the Si Wei Medical Company and the Ren Ji Hospital Remote ECG Diagnostic Center. It consists of the clinical diagnostic data of a medical diagnostic system, and has been accumulated by the Ren Ji Hospital over a period of about three years. The entire database consists of 98 287 pieces of ECG data, and one piece of data consists of a 12-lead ECG signal of 20 seconds at a sampling rate of 500 Hz. The ECG data are measured by a standard 12-lead diagnosis system and include the channels I, II, III, aVR, aVL, aVF, V1, V2, V3, V4, V5, and V6. Channels I, II, III are limb leads, while aVR, aVL, aVF are augmented limb leads, and V1, V2, V3, V4, V5, and V6 are chest leads. The database consists of 1 251 types of single or mixed diseases. There are 249 single disease categories. The dataset used in our experiment is a subset of the entire database. It consists of 3 000 pieces of high-quality 12-lead ECG records. Each piece includes about 10 to 25 beats for a total of 65 716 beats. These records are collected from people of different genders, ages and physical conditions. The doctor's diagnosis is taken as the label for the beats; this is one of the following six types: normal beat (N), left bundle branch block beat (L), right bundle branch block beat (R), left ventricular hypertrophy (V), sinus bradycardia (S), and electrical axis left side (E). After the preprocessing step for the raw ECG signal, we get the following single heartbeat segments: 19 400 of N type, 7 056 of L type, 10 080 of R type, 6 720 of V type, 14 540 of S type, and 7 920 of E type. Next, we split the dataset into two parts: training and test. We use the training part to calculate the projection vectors and then train the SVM model. The models are then used for classifying the test dataset. We randomly split the original dataset into two parts: 10 952 beats for training and 54 764 beats for testing. The training set consists of one sixth of the total data while the testing set consists of the remaining five sixths. Details regarding the size of the training and the testing sets for each specific type are listed in Table 2.

Table 2. Number of Beats for Each Class in the Dataset

Beat Type	Number of Training Beats	Number of Testing Beats
N	3 233	16 167
L	1 176	5 880
R	1 680	8 400
V	1 120	5 600
S	2 423	12 117
E	1 320	6 600

6.3 Dataset Results

Our overall process includes the following steps: data preprocessing, tensor data computation, STFT, tensor feature extraction, dimension reduction based on our proposed SSSMDA approach, and multiclass classification. Fig.5 presents the block diagram of this process.

In our method, we use label propagation to assign a label to each unlabeled datum. In this specific case, we use logistic label propagation to calculate the label[32]. The calculated result is shown in Fig.6. The label propagation accuracy is close to 65%, which is sufficient to be used in our method.

[①] This dataset will be made public in the near future. See http://bcmi.sjtu.edu.cn/ehealth/ for further information.

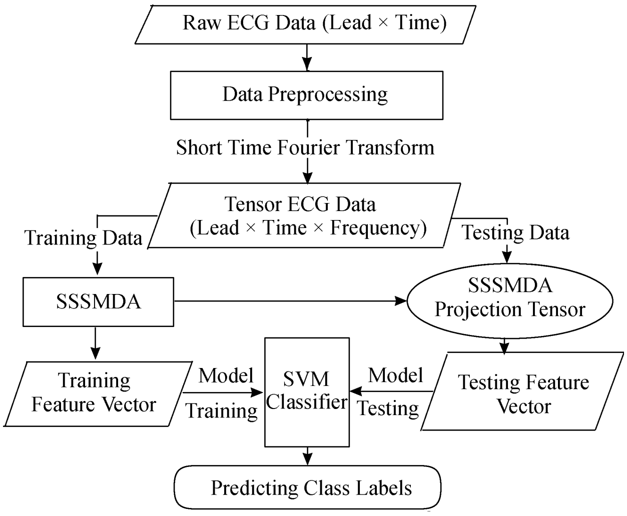


Fig.5. Tensor-based process for ECG feature extraction.

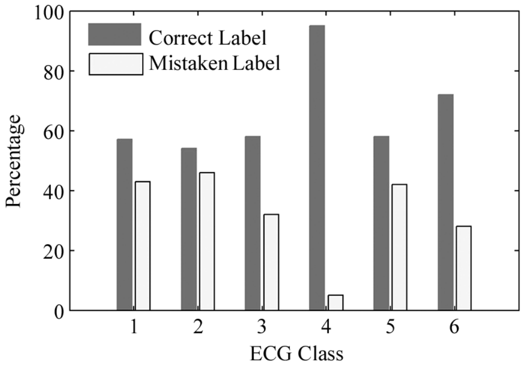


Fig.6. Logistic label propagation correct labels and error rates for each class.

In our method, parameter N_p represents the N_p closing point graph as shown in (1). Here we calculate the classification accuracy corresponding to various N. Fig.7 displays the results while the variance is plotted in Fig.8. Clearly, numbers near 30 are the best choices.

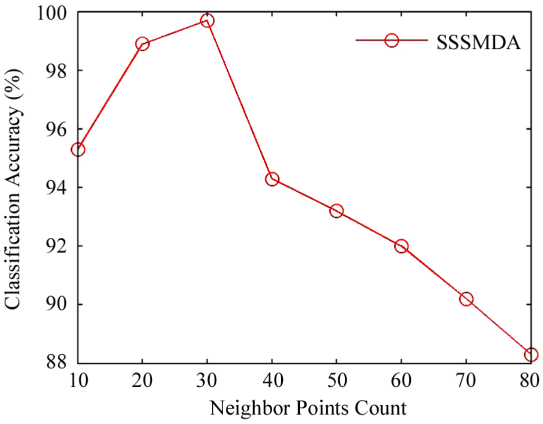


Fig.7. Classification accuracy corresponding to different choices of closing point number.

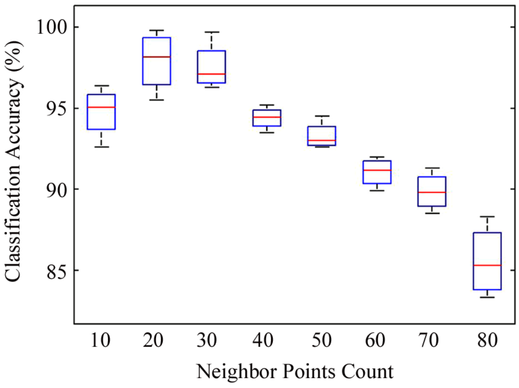


Fig.8. Variance of classification accuracy corresponding to different choices of closing point number.

A major characteristic of our method is the sparseness of the projection tensor. Thus we compare here the classification accuracy of different nonzero value counts in the projection tensor. Fig.9 highlights the fact that 20∼30 nonzero values in the projection tensor provide better accuracy. A plot of the variance is shown in Fig.10. Hence, the original ECG tensor data are sparse and, as such, facilitate the classification process.

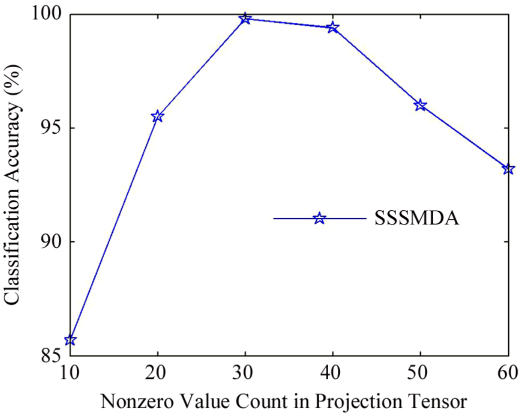


Fig.9. Classification accuracy corresponding to different numbers of nonzero values in the projection tensor.

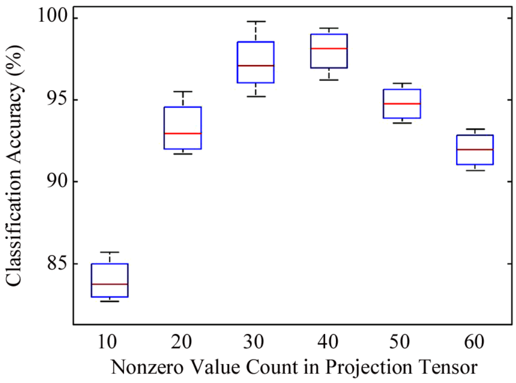


Fig.10. Variance of classification accuracy corresponding to different numbers of nonzero values in the projection tensor.

As for our tensor feature extraction method, the most important performance metric is classification accuracy. Here the classification accuracies of SSSMDA, TR1DA, UMPCA, PCA, ICA, and LDA are compared for dimensions 1~20 (Fig.11).

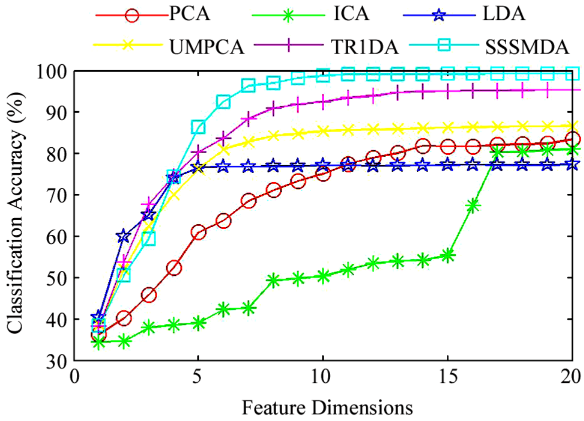


Fig.11. Classification accuracy of different approaches with different feature dimensions.

In Fig.11, it can be seen that the tensor-based approach is much better than any vector space based method. The main reason for this relies in the fact that the tensor-based approaches use the tensor data directly as input, preventing the well-known SSS problem[46] as well as preserving the structural information in the tensor data. SSSMDA achieves the best performance compared with other tensor-based methods. As for vector space based methods, the order from the best to the worst is: LDA, PCA, and ICA for low dimensions. Note that the order differs for higher dimensions: LDA, ICA, and PCA. The accuracy of ICA increases sharply for dimensions over 15.

The classification accuracy of each class is listed in Table 3. Clearly, tensor-based methods reach a higher accuracy and outperform other approaches. In addition, our SSSMDA method performs much better than all other tensor-based methods. Four classes of ECG have a 100% classification accuracy rate, meaning that SSSMDA gives good results in practice.

Table 3. Comparison of Classification Accuracy (%) for Different Approaches

Beat Type	PCA	ICA	LDA	UMPCA	TR1DA	SSSMDA
N	79.24	85.78	82.49	89.97	91.58	95.32
L	73.12	80.82	84.41	90.98	90.35	94.78
R	79.98	78.10	70.56	88.97	89.24	96.47
V	89.37	88.22	80.78	85.78	88.99	92.98
S	95.90	92.14	91.34	91.89	92.85	97.59
E	82.16	81.32	75.48	86.49	93.01	96.12

To further illustrate the effectiveness of our approach, we extract 3D features using feature extraction methods and plot the distribution of these features in Fig.12.

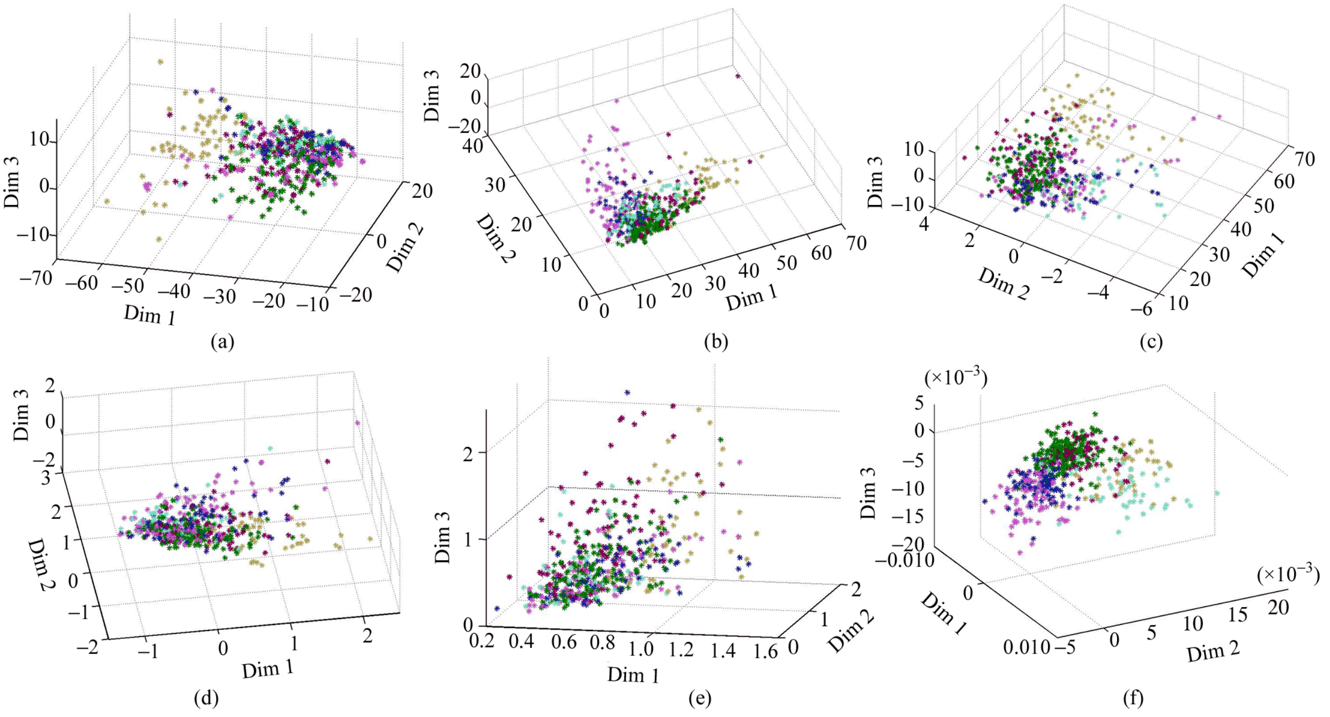


Fig.12. 3D dimensional distribution of extracted features. (a) SSSMDA. (b) TR1DA. (c) UMPCA. (d) PCA. (e) ICA. (f) LDA. Dim: dimension.

From Fig.12, it is clear that SSSMDA, TR1DA, and LDA show clustering characteristics. Both LDA and TR1DA seem to feature better clustering characteristics; however, there is an obvious overlap at the boundary of each data class. In the end, the proposed SSSMDA method achieves a better classification accuracy than both LDA and TR1DA.

7 Conclusions

To enable feature extraction, we overcame the dimension-reduction and classification problems related to large 12-lead hospital-standard ECG datasets by transforming the data into tensor form in the spatial-spectral-temporal domain using STFT. Conventional methods face three main challenges, namely: 1) ECG effective features are sparse in the tensor representation, 2) manual diagnosis incurs high expense, and 3) the lack of labeled ECG data. Therefore, we proposed a multilinear semisupervised sparse discriminative analysis approach (SSSMDA) that takes the tensor data as its direct input. The method effectively calculates the sparse projection tensor and extracts valuable features for classification. Compared with original LDA, our approach additionally considers the manifold structure and distribution of the unlabeled data. This new strategy allows us to determine the best projection tensor and extract valuable features for classification purposes. The experimental results show that tensor ECG data contain sparse valuable features. In addition, the tensor-based scheme outperforms traditional vector-based methods. In particular, SSSMDA outperforms both TR1DA and UMPCA in terms of classification accuracy, demonstrating the effectiveness and robustness of SSSMDA (and tensor-based schemes in general) for classifying 12-lead ECG signals.

References

[1] Sirovich L, Kirby M. Low-dimensional procedure for the characterization of human faces. *Journal of the Optical Society of America A*, 1987, 4(3): 519-524.

[2] Kirby M, Sirovich L. Application of the Karhunen-Loéve procedure for the characterization of human faces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1990, 12(1): 103-108.

[3] Turk M, Pentland A. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 1991, 3(1): 71-86.

[4] Hyvärinen A. Survey on independent component analysis. *Neural Computing Surveys*, 1999, 2:94-128.

[5] Liu Q S, Lu H Q, Ma S D. Improving kernel Fisher discriminant analysis for face recognition. *IEEE Transactions on Circuits and Systems for Video Technology*, 2004, 14(1): 42-49.

[6] Zhao Q B, Zhang L Q. ECG feature extraction and classification using wavelet transform and support vector machines. In *Proc. International Conference on Neural Networks and Brain*, October 2005, pp.1089-1092.

[7] Jen K K, Hwang Y R. ECG feature extraction and classification using cepstrum and neural networks. *Journal of Medical and Biological Engineering*, 2008, 28(1): 31-37.

[8] Pasolli E, Melgani F. Active learning methods for electrocardiographic signal classification. *IEEE Transactions on Information Technology in Biomedicine*, 2010, 14(6): 1405-1416.

[9] Zhang H, Zhang L Q. ECG analysis based on PCA and support vector machines. In *Proc. International Conference on Neural Networks and Brain*, October 2005, pp.743-747.

[10] Wu Y, Zhang L Q. ECG classification using ICA features and support vector machines. In *Lecture Notes in Computer Science 7062*, Lu B L, Zhang L Q, Kwok J T (eds.), 2011, Springer, pp.146-154.

[11] Huang K, Zhang L Q, Wu Y. ECG classification based on non-cardiology feature. In *Proc. the 9th International Symposium on Neural Networks*, July 2012, pp.179-186.

[12] Zhao W Y, Chellappa R, Phillips P J, Rosenfeld A. Face recognition: A literature survey. *ACM Computing Surveys*, 2003, 35(4): 399-458.

[13] Yang J, Zhang D, Frangi A F, Yang J Y. Two-dimensional PCA: A new approach to appearance-based face representation and recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2004, 26(1): 131-137.

[14] Yang J, Zhang D, Yong X, Yang J Y. Two-dimensional discriminant transform for face recognition. *Pattern Recognition*, 2005, 38(7): 1125-1129.

[15] Ye J P. Generalized low rank approximations of matrices. *Machine Learning*, 2005, 6(1/2/3): 167-191.

[16] Ye J P, Janardan R, Li Q. Two-dimensional linear discriminant analysis. In *Proc. the 18th Annual Conf. Neural Information Processing Systems*, December 2004, pp.1569-1576.

[17] Tao D C, Li X L, Wu X D, Maybank S J. General tensor discriminant analysis and Gabor features for gait recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2007, 29(10): 1700-1715.

[18] Lu H P, Plataniotis K N, Venetsanopoulos A N. Multilinear principal component analysis of tensor objects for recognition. In *Proc. the 18th International Conference on Pattern Recognition*, August 2006, pp.776-779.

[19] Lu H P, Plataniotis K N, Venetsanopoulos A N. Uncorrelated multilinear principal component analysis through successive variance maximization. In *Proc. the 25th International Conference on Machine Learning*, July 2008, pp.616-623.

[20] Lu H P, Plataniotis K N, Venetsanopoulos A N. Uncorrelated multilinear principal component analysis for unsupervised multilinear subspace learning. *IEEE Transactions on Neural Networks*, 2009, 20(11): 1820-1836.

[21] Tao D C, Li X L, Wu X D, Maybank S. Tensor rank one discriminant analysis — A convergent method for discriminative multilinear subspace selection. *Neurocomputing*, 2008, 71(10/11/12): 1866-1882.

[22] Yan S C, Xu D, Yang Q, Zhang L, Tang X, Zhang H J. Discriminant analysis with tensor representation. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, June 2005, pp.526-532.

[23] Lu H P, Plataniotis K N, Venetsanopoulos A N. A survey of multilinear subspace learning for tensor data. *Pattern Recognition*, 2011, 44(7): 1540-1551.

[24] Li D, Huang K, Zhang H L, Zhang L Q. UMPCA based feature extraction for ECG. In *Proc. the 10th International Symposium on Neural Networks*, July 2013, pp.383-390.

[25] Cheng B F, Huang K, Zhang L Q. Tensor-based feature extraction method for ECG. In *Proc. International Conference on Medical Physics and Biomedical Engineering*, September 2012, pp.383-390.

[26] Huang K, Zhang L Q. Optimal calculation of tensor learning approaches. In *Proc. the 10th International Symposium on Neural Networks*, July 2013, pp.326-333.

[27] Zhang Y, d'Aspremont A, Ghaoui L E. Sparse PCA: Convex relaxations, algorithms and applications. In *Handbook on Semidefinite, Cone and Polynomial Optimization: Theory, Algorithms, Software and Applications*, Anjos M, Lasserre J B (eds.), 2011, pp.915-940.

[28] Clemmensen L. Sparse discriminant analysis (sparseLDA) software in Matlab, version 2.0. May 2008. http://www2.imm.dtu.dk/projects/spasm/, May 2014

[29] Lai Z H, Xu Y, Yang J, Tang J H, Zhang D. Sparse tensor discriminant analysis. *IEEE Transactions on Image Processing*, 2013, 22(10): 3904-3915.

[30] Cai D, He X F, Han J W. Semi-supervised discriminant analysis. In *Proc. the 11th IEEE International Conference on Computer Vision*, October 2007, pp.1-7.

[31] Qu W, Song K S, Zhang Y F, Feng S, Wang D L, Yu G. A novel approach based on multi-view content analysis and semi-supervised enrichment for movie recommendation. *Journal of Computer Science and Technology*, 2013, 28(5): 776-787.

[32] Kobayashi T, Watanabe K J, Otsu N. Logistic label propagation. *Pattern Recognition Letters*, 2012, 33(5): 580-588.

[33] Huang K, Zhang L Q. ECG representation with simulated 2D and 3D VCG using prior knowledge based weighted PCA. In *Proc. International Conference on Medical Physics and Biomedical Engineering*, September 2012, pp.383-390.

[34] Allen J B, Rabiner L R. A unified approach to short-time Fourier analysis and synthesis. *Proceedings of IEEE*, 1977, 65(11): 1558-1564.

[35] Miwakeichi F, Martínez-Montes E, Valdés-Sosa P, Nishiyama N, Mizuhara H, Yamaguchi Y. Decomposing EEG data into space-time-frequency components using parallel factor analysis. *NeuroImage*, 2004, 22(3): 1035-1045.

[36] Vos M D, Vergult A, Lathauwer L D, Clercq W D, Huffel S V, Dupont P, Palmini A, Paesschen W V. Canonical decomposition of ictal scalp EEG reliably detects the seizure onset zone. *NeuroImage*, 2007, 37(3): 844-854.

[37] Acar E, Aykut-Bingol C, Bingol H, Bro R, Yener B. Multiway analysis of epilepsy tensors. In *Proc. the 15th International Conference on Intelligent Systems for Molecular Biology* (*ISMB*) & *the 6th European Conference on Computational Biology (ECCB)*, July 2007, pp.10-18.

[38] Zhu X j. Semi-supervised learning literature survey. Technical Report, Department of Computer Sciences University of Wisconsin, Madison, 2005. http://pages.cs.wisc.edu/~jerryzhu/pub/ssl_survey.pdf, Sept. 2014

[39] Belkin M, Niyogi P, Sindhwani V. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *Journal of Machine Learning Research*, 2006, 7: 2399-2434.

[40] Chung F R K. Spectral Graph Theory (CBMS Regional Conference Series in Mathematics, No.92). American Mathematical Society, 1996.

[41] Song F X, Zhang D, Mei D, Guo Z W. A multiple maximum scatter difference discriminant criterion for facial feature extraction. *IEEE Transactions on Systems Man and Cybernetics, Part B*: *Cybernetics*, 2007, 37(6): 1599-1606.

[42] Gao J, Xiang L. Laplacian maximum scatter difference discriminant criterion. In *Communications in Computer and Information Science 224*, Zeng D (ed.), Springer 2011, pp.691-697.

[43] Leiva-Murillo J M, Artes-Rodríguez A. Maximization of mutual information for supervised linear feature extraction. *IEEE Transactions on Neural Networks*, 2007, 18(5): 1433-1441.

[44] Dhir C, Lee S. Discriminant independent component analysis. In *Lecture Notes in Computer Science 5788*, Corchado E, Yin H (eds.), 2009, Springer, pp.219-225.

[45] Tao D C, Li X L, Hu W M, Maybank S J, Wu X D. Supervised tensor learning. In *Proc. the 5th IEEE International Conference on Data Mining*, Nov. 2005, pp.450-457.

[46] He Y H. Solving undersampled problem of LDA using Gram-Schmidt orthogonalization procedure in difference space. In *Proc. International Conference on Advanced Computer Control*, January 2009, pp.153-157.

Kai Huang received his B.S. degree in computer science from University of Shanghai for Science and Technology in 2005, and M.S. degree from Shanghai Jiao Tong University in 2008. He joined the Center for Brain-Like Computing and Machine Intelligence as a Ph.D. candidate in 2008. His research interests are brain computer interface with EEG signal, statistical learning, artificial intelligence, machine learning, data mining, and geographic information system.

Li-Qing Zhang received his B.S. degree in mathematics from Hangzhou University in 1983, M.S. and Ph.D. degrees in computer science from Zhongshan University, Guangzhou, in 1985 and 1988, respectively. Now he is a professor and vice chair of the Department of Computer Science at Shanghai Jiao Tong University. His long-term goal is to understand how intelligent information is processed in the brain and develop new type (brain-like) computational models and algorithms for visual and auditory information processing. Currently, his research interests cover brain-like computing model and its computing mechanism, visual information representation and global feature analysis, brain signal processing and brain-computer interface, perception and cognition computing model, statistical learning and inference. He has published more than 200 papers in international journals and conferences. He serves as the associate editor of International Journal of Computational Intelligence and Neuroscience, the director of the committee of Biocybernetics and Biomedical Engineering, Chinese Automation Association. He is also a reviewer of a number of international journals, such as IEEE Trans. Neural Networks，IEEE Trans. Signal Processing, and IEEE Signal Processing Letters.