EURASIP Journal on Advances in Signal Processing
a SpringerOpen Journal

RESEARCH Open Access

Cardiology knowledge free ECG feature extraction using generalized tensor rank one discriminant analysis

Kai Huang* and Liqing Zhang

Abstract

Applications based on electrocardiogram (ECG) signal feature extraction and classification are of major importance to the autodiagnosis of heart diseases. Most studies on ECG classification methods have targeted only 1- or 2-lead ECG signals. This limitation results from the unavailability of real clinical 12-lead ECG data, which would help train the classification models. In this study, we propose a new tensor-based scheme, which is motivated by the lack of effective feature extraction methods for direct tensor data input. In this scheme, an ECG signal is represented by third-order tensors in the spatial-spectral-temporal domain after using short-time Fourier transform on the raw ECG data. To overcome the limitations of tensor rank one discriminant analysis (TR1DA) inherited from linear discriminant analysis, we introduced a generalized tensor rank one discriminant analysis (GTR1DA). This approach involves considering the distribution of the data points near the classification boundary to calculate better projection tensors. The experimental results showed that the proposed method achieves greater classification accuracy than other vector- and tensor-based methods. Finally, GTR1DA features a better convergence property than the original TR1DA.

Keywords: ECG analysis; Feature extraction; Tensor learning approaches; Dimension reduction; TR1DA

1 Introduction

As the Internet of Things technology matures, remote and centralized electrocardiogram (ECG) diagnostic platforms have been developed. A highly centralized platform has a large amount of ECG data to diagnose daily; therefore, an aided diagnosis system providing reference diagnostics is beneficial. An ECG represents the working state of the heart by recording the magnitude and direction of the electrical activities related to the heart. It provides useful information for doctors to diagnose heart diseases. In recent years, the automatic diagnosis of heart diseases through the analysis of ECG signals has drawn much attention. ECG feature extraction consists of determining a set of discriminant features to represent ECG data to achieve strong classification performance. Many technologies and algorithms on machine learning and signal processing have been used to achieve this goal. Zhao et al. proposed a feature extraction method using wavelet transform [1]. Jen et al. described an approach that takes advantage of the neural networks for determining the features of a given ECG signal [2]. Pasolli et al. introduced an active learning method for ECG classification based on certain ECG morphology and temporal features [3]. Zhang et al. used the principal component analysis (PCA) algorithm to extract ECG features [4]. An ECG feature extraction scheme using independent component analysis (ICA) was presented by Wu et al. [5,6].

The main problem with the existing methods is their limitation to 1- or 2-lead ECGs because a public 12-lead ECG database is lacking. Traditional methods developed for 2-lead signals cannot be directly applied to 12-lead ECG signals. They typically use vectors to represent ECG signals; however, this is not a natural representation of ECG signals because a large amount of useful structural information is discarded. The structural information indicates the relative positions of the signals among different leads. The 12-lead ECG provides spatial information about the electrical activity of the heart in three approximately orthogonal directions. If all the information of the

*Correspondence: mariaki888888@sjtu.edu.cn
The MOE-Microsoft Key Laboratory for Intelligent Computing and Intelligent Systems, Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai 200240, China

12-lead signals is considered, then robust features can be obtained and then used for an automatic analysis. Thus, a high classification accuracy and an efficient representation of the ECG signals can be achieved. Moreover, the time-frequency domain features can also help improve the effect of the classification. Therefore, our method involves converting an original 12-lead ECG signal into a spatial-spectral-temporal domain such that it becomes a third-order tensor.

Scanning all tensor ECG signals into a vector greatly increases the dimensions of the feature space. Consequently, the number of training samples seems extremely small compared with the large dimensions of the feature space; this problem is called the small sample size (SSS) problem [7]. Another problem is related to the number of unknown parameters that can lead to an overfitting problem according to the principle of Ockham's razor. Therefore, the objective is to use the tensor feature extraction method to retain as much of the data structure information as possible. A more natural method to represent ECG compared with vectorization is to use third-order tensors. If the appropriate technology is adopted, then additional information can be extracted from the ECG data, and, therefore, the SSS problem can be addressed. Li et al. successfully used a general tensor discriminant analysis for single-trial electroencephalogram (EEG) classification [8]. This approach has also been adopted in gait recognition [9]. Other tensor methods include multilinear principal component analysis (MPCA) [10,11], uncorrelated multilinear PCA (UMPCA) [12,13], and tensor rank one discriminative analysis (TR1DA) [14]. These approaches can be classified into two categories [15]: the methods corresponding to tensor-to-tensor projection and the methods corresponding to tensor-to-vector projection. GTDA [9] and MPCA [10,11] belong to the tensor-to-tensor category, whereas TR1DA [14] and UMPCA [12,13] belong to the tensor-to-vector projection category. Because TR1DA is an extension of linear discriminant analysis (LDA), it also exhibits the same limitations. More precisely, it does not enable the characteristics of the original data distribution and adjacent points of the different classes to be fully considered. In this study, we propose to improve the original TR1DA method and overcome its limitations. We therefore consider the distribution of the closing points of the different classes to calculate better projection tensors and improve the accuracy of the classification.

In this paper, we introduce a new tensor-based scheme to extract features from 12-lead ECG signals. We represent them using high-order tensors, i.e., 12-lead signals in the spatial-spectral-temporal domain. The tensors were constructed using short-time Fourier transform (STFT) on the raw ECG signals. Generalized TR1DA (GTR1DA) was used to reserve the projection tensors from which the features of the original tensors were obtained. Finally, a support vector machine (SVM) with Gaussian kernel was used for the classification in the feature space. We tested the proposed method on our patient database and achieved a high classification accuracy.

The paper is organized as follows. The data preprocessing approach and the tensor data achievement process are introduced in Sections 2.1 and 2.2, respectively. We then discuss the tensor calculation used in this paper in Section 3. Section 4 indicates the major limitations of LDA, and Section 5 introduces an extension of LDA. In Section 6, the main idea of TR1DA is presented. We then show how to generalize the original TR1DA into GTR1DA in Section 7. Section 8 justifies the proposed method. In Section 9, we introduce the initial value calculation used in our approach. In Section 10, we clarify and briefly explain how SVM was applied for multiclass classification. We then briefly introduce our ECG database in Section 11.1 and describe the convergence improvement and the accuracy of the classification in Section 11.2. Section 12 presents the conclusion.

2 Tensor-based scheme

Our work covered the data preprocessing, the tensor data computation using STFT, the tensor feature extraction and dimension reduction based on our new GTR1DA approach, and the classification of the multiclass. The block diagram is provided in Figure 1.

2.1 Data preprocessing

Raw ECG signals typically display strong background noise. To suppress the noise without ruining the data, we used common methods, such as wavelet transformation, to remove high-frequency noise and median filters

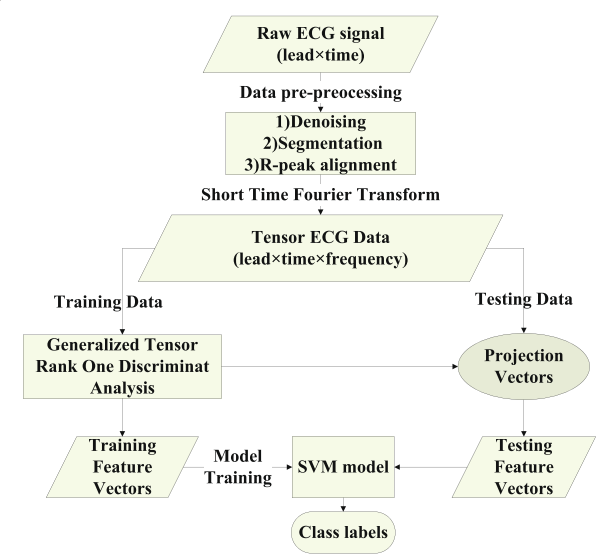

Figure 1 The tensor-based scheme for ECG feature extraction.

to eliminate the baseline drift. An ECG signal from a diagnosis is approximately 20 s long and comprises approximately 25 beats. Therefore another critical preprocessing step is to segment the signal into 'ECG pieces', each of which contains one heartbeat.

2.2 Short-time Fourier transform

The original signals represent the features in the spatial-temporal domain. Because ECG signals are nonstationary, we used STFT [16] instead of the regular Fourier transform to collect information on when a particular frequency component occurs. STFT can provide useful information regarding the time resolution of the spectrum. STFT involves extracting several frames from the signals and analyzing them using a time-sliding window such that the relation between the variation of the frequency and the time can be identified. We used STFT to transform the original signals into a spatial-spectral-temporal domain as high-dimensional third-order tensors [17]. Given a signal that varies over time, STFT was used to determine the sinusoidal frequency and phase the content of the local sections.

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

$$\begin{aligned}\mathbf{STFT}\{s[l,n]\}(m,w) &\equiv S(l,m,n)\\ &= \Sigma_{m=-\infty}^{\infty}\omega(n-m)s(l,m)e^{-j2\pi fm}\end{aligned} \tag{1}$$

where $w[n]$ is the window function that selectively determines the portion of $s[l,n]$ to use for the analysis. In this study, we used the Hann window. Once STFT had been applied to the ECG signals, all of the signals were in a third-order tensor form for the remainder of the analysis. In accordance with previous studies on extending an EEG signal to a third-order tensor [18-20], we extended the original ECG signal such that we could effectively extract valuable features in the spatial-spectral-temporal domain. To appropriately manage this type of data, using a tensor-based learning approach was necessary.

3 Tensor operations

We first introduce definitions of tensor operations to fix the notations. In our paper, Mathcal font and uppercase letters denote tensors (e.g., $\mathcal{X}, \mathcal{Y}, \mathcal{Z}$). Matrices are expressed as bold uppercase letters (e.g., $\mathbf{X}$, $\mathbf{B}$). Lowercase bold letters are used for vectors (e.g., $\mathbf{u}$, $\mathbf{a}$). Lowercase and uppercase letters are scalers (e.g., a, b, c, D, E).

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

$$(\mathbf{x} \otimes \mathbf{y})_{ij} = \mathbf{x}_i \times \mathbf{y}_j, \tag{2}$$

which is a rank 1 tensor of mode 2. Here, $0 < i \leq M$ and $0 < j \leq N$. The tensor product of three vectors $\mathbf{x} \in R^M$ $\mathbf{y} \in R^N$, and $\mathbf{z} \in R^S$ is a mode-3 tensor:

$$(\mathbf{x} \otimes \mathbf{y} \otimes \mathbf{z})_{ijk} = \mathbf{x}_i \times \mathbf{y}_j \times \mathbf{z}_k, \tag{3}$$

which is also of rank 1. Here, $0 < i \leq M$, $0 < j \leq N$, and $0 < k \leq S$.

Definition 2. *Tensor mode product.* A mode M tensor $\mathcal{X}$ of size $X \in R^{N_1 \times N_2 \times \ldots \times N_M}$ multiplied by a vector in mode r is a tensor with a size of $N_1 \times N_2 \times \ldots \times N_{r-1} \times 1 \times N_{r+1} \times \ldots \times N_M$:

$$\begin{aligned}&(\mathcal{X} \times_r \mathbf{u})_{i_1 \times i_2 \times \ldots \times i_{r-1} \times 1 \times i_{r+1} \times \ldots \times i_M}\\ &\quad = \sum_{i_r} (\mathcal{X}_{i_1 \times i_2 \times \ldots \times i_{r-1} \times i_r \times i_{r+1} \times \ldots \times i_M} \mathbf{u}_{i_r}),\end{aligned} \tag{4}$$

which is 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:

$$\mathbf{u}^1 \otimes \mathbf{u}^2 \otimes \ldots \otimes \mathbf{u}^n = \prod_{l=1}^{M} \otimes (\mathbf{u}^l)^T. \tag{5}$$

4 Limitations of classical linear discriminant analysis

In this section, we review the main limitations of classical LDA, including the SSS problem, the low-rank problem, the heteroscedastic problem, the problem raised by the unreasonable between-class scatter matrix, and the unconsidered distribution structure between classes.

4.1 Small-size problem

In the classical LDA approach, the calculation of the ratio of the between-class covariance distance to the within-class covariance distance is a generalized Rayleigh quotient problem. We illustrate it using the most widely used optimization problem, which is the first optimization problem in Equation 6.

$$R(\mathbf{x}) = \frac{\mathbf{x}^T\mathbf{S}_b\mathbf{x}}{\mathbf{x}^T\mathbf{S}_w\mathbf{x}}, R(\mathbf{x}) = \frac{\mathbf{x}^T\mathbf{S}_b\mathbf{x}}{\mathbf{x}^T\mathbf{S}_t\mathbf{x}}, R(\mathbf{x}) = \frac{\mathbf{x}^T\mathbf{S}_w\mathbf{x}}{\mathbf{x}^T\mathbf{S}_t\mathbf{x}}. \tag{6}$$

The Lagrange multiplier method can be used to solve this problem. It is clear from Equation 6 that $\mathbf{x}$ has an infinite number of solutions. When $\mathbf{x}$ is multiplied by a constant, $R(\mathbf{x})$ maintains the same value (only offsetting the numerator and denominator). Therefore, the length of $\mathbf{x}$ is always restricted, such that the denominator is 1. The

restriction is used as a condition for the Lagrange method. The solution to this problem maximizes the equation:

$$\begin{aligned} c(\mathbf{x}) &= \mathbf{x}^T\mathbf{S}_b\mathbf{x} - \lambda(\mathbf{x}^T\mathbf{S}_w\mathbf{x} - 1) \\ \Rightarrow \frac{dc}{d\mathbf{x}} &= 2\mathbf{S}_b\mathbf{x} - 2\lambda\mathbf{S}_w\mathbf{x} = 0 \\ \Rightarrow \mathbf{S}_b\mathbf{x} &= \lambda\mathbf{S}_w\mathbf{x}. \end{aligned} \tag{7}$$

The above equation is a typical generalized eigenvalue problem. If the within-class scatter matrix $\mathbf{S}_w$ is invertible, this problem can be converted into an ordinary eigenvalue problem, allowing us to calculate the result:

$$\mathbf{S}_w{}^{-1}\mathbf{S}_b\mathbf{x} = \lambda\mathbf{x} \tag{8}$$

The within-class scatter matrix is the sum of each class covariance matrix, and $\text{rank}(\mathbf{C}) \leq \text{rank}(\mathbf{A}) + \text{rank}(\mathbf{B})$; therefore, the rank of the within-class scatter matrix is at most s (the number of classes) less than the sample number.

In most cases, such as image processing, video analysis, FMRI, or CT, the dimensions of the original signal are considerably large. However, if the training set is excessively small and the number of cases is less than the dimension count, the application of LDA is limited, and modifications are required to solve the problem.

4.2 Class number 1: low-rank problem

The classical LDA approach involves calculating the between-class scatter matrix by computing the covariance matrix of each class data center. $\mathbf{H}_b$ can also be expressed in the following form:

$$\begin{aligned} \mathbf{H}_b &= \frac{1}{\sqrt{n}}\left[\mathbf{C}^{(1)}, \mathbf{C}^{(2)}, \ldots, \mathbf{C}^{(i)}, \ldots, \mathbf{C}^{(s)}\right] \\ \mathbf{C}^{(i)} &= \left[(\mathbf{c}_1^{(i)} - \mathbf{c}), (\mathbf{c}_2^{(i)} - \mathbf{c}), \ldots, (\mathbf{c}_{n_i}^{(i)} - \mathbf{c})\right]. \end{aligned} \tag{9}$$

Here, $\mathbf{H}_b \times \mathbf{H}_b^T = \mathbf{S}_b$. In the first equation, s is the class number. In the second equation, c is the mean of all of the data. The term n_i represents the point count for the class i. The term $c_{n_i}^{(i)}$ is the n_ith mean of class i. The vectors of each class are the same, and, therefore, their rank is 1. Because each column of the matrix is reduced by the average of all columns, the weighted average is 0. In other words, they are linearly correlated:

$$\sum_{i=1}^{s} n_i\left(\mathbf{c}^{(i)} - \mathbf{c}\right) = 0 \tag{10}$$

Here, n_i is the number of points in class i. Generally, the centers of each class are linearly independent; therefore, the rank of H_b is $(s-1)$. The term s again represents the number of classes. Using the following lemma, it is easy to compute the rank of the between-class scatter matrix as $(s-1)$.

Lemma 1. *For any $m \times n$ matrix* $\mathbf{A}$ *$rank$* $(\mathbf{A}) = rank(\mathbf{A}') = rank(\mathbf{A}\mathbf{A}') = rank(\mathbf{A}'\mathbf{A})$.

Using Lemma 1, it is easy to conclude that S_b and H_b share the same rank; specifically, one less than the number of classes s. Although $s-1$ is an upper bound, the actual value is close. The rank of $\mathbf{S}_b$ is less than the upper bound $s-1$ only if the centers of the various classes are on the same line, which must pass through the origin. In practice, this case is extremely rare and, therefore, in most cases, the rank of $\mathbf{S}_b$ is close to $s-1$.

This reveals a paradox of classical LDA: if the number of classes is high, then the accuracy of the results is low, but if the number is comparatively low, then the dimension extracted using LDA is also considerably low. In this case, if two classes are used for calculating the projection matrix, and regardless of the height of the original data dimension, the calculated reduced dimension can only be 1. Therefore, in this condition, the classification result is greatly affected.

4.3 Heteroscedastic problem

Classical LDA assumes that the data distribution in each class is Gaussian:

$$\begin{aligned} \mathbf{S}_w &= \frac{1}{n}\sum_{i=1}^{k}\mathbf{S}_{w_i} \\ \mathbf{S}_{w_i} &= \sum_{\mathbf{x}\in A_i}(\mathbf{x} - \mathbf{c}^{(i)})(\mathbf{x} - \mathbf{c}^{(i)})^T. \end{aligned} \tag{11}$$

Here, A_i represents the point of class i, and $c^{(i)}$ represents its mean. In this approach, it is assumed that the data distribution in each class approximates a Gaussian distribution and that the covariance matrices of all of the classes are similar. However, in practice, this assumption has not been verified because they might have some type of structure in the feature space or a weak clustering structure. Data for various classes may not be linearly or even nonlinearly separable. Different class data can also overlap.

4.4 Unreasonable between-class scatter matrix

In the traditional LDA approach, the design of the between-class scatter matrix is unreasonable as $\mathbf{S}_b$. The design of the matrix involves calculating the between-class covariance distance of class data centers. This approach seems simple and effective but exhibits several problems. It entails the assumption that the centers of each class, as a whole, approximate a Gaussian distribution. This is often not the case for real data. Conversely, if the number of classes is low, then the data centers can be considerably sparse. In this situation, even if they have a Gaussian distribution, the calculated axis direction can be incorrect because of the sparsity of the data, hardening the

calculation. The low-rank problem of dimension $(s-1)$ is caused by this design.

5 Extension of classical LDA

We first introduce a method to extend the original LDA approach and then generalize it to the tensor space to form the GTR1DA method. The original LDA method exhibits the small-size problem, the $(s-1)$ low-rank problem, and the heteroscedastic problem. Most importantly, the unreasonable between-class scatter matrix does not consider the distribution of the data points nearing the classification boundary. To overcome these limitations, we introduce the following improvement:

$$R(\mathbf{x}) = \frac{\mathbf{x}^T \mathbf{S}_b \mathbf{x} + w_2 \mathbf{x}^T \mathbf{S}_{oo} \mathbf{x}}{\mathbf{x}^T \mathbf{S}_w \mathbf{x}} \tag{12}$$

$$\mathbf{S}_{oo} = \sum_{i,j} w_{ij} \sum_{\mathbf{u}\in A_i, \mathbf{v}\in A_j} \mathbf{S}_{\mathbf{uv}} \qquad i \neq j \tag{13}$$

$$\mathbf{S}_{\mathbf{uv}} = (\mathbf{u} - \mathbf{c}^{(\mathbf{uv})})(\mathbf{u} - \mathbf{c}^{(\mathbf{uv})})^T + (\mathbf{v} - \mathbf{c}^{(\mathbf{uv})})(\mathbf{v} - \mathbf{c}^{(\mathbf{uv})})^T \tag{14}$$

where A_i and A_j are points of classes i and j , respectively. The term $\mathbf{c}^{\mathbf{uv}}$ represents the mean of $\mathbf{u}$ and $\mathbf{v}$. The matrix $\mathbf{S}_{\mathbf{uv}}$ used here with only two vectors is similar to the case of S_b, in which there are only two classes of data. Its rank is 1, and its eigenvector corresponding to the nonzero eigenvalue is the connection of the two data centers.

This approach involves considering each pair of points from the two classes. The final projection direction is the sum of the angle between the direction and the connection of each pair of points.

From a geometrical perspective, the projection direction is determined by the angle it makes with the connection of each point pair of the various classes:

$$\mathbf{x}^T S_{\mathbf{uv}} \mathbf{x} = \sum_{i\neq j, \mathbf{u}\in A_i, \mathbf{v}\in A_j} |\mathbf{x}| \cos\theta_{\mathbf{xuv}} \lambda_{\mathbf{uv}} \cos\theta_{\mathbf{xuv}} |\mathbf{x}| \tag{15}$$

$$= \sum_{i\neq j, \mathbf{u}\in A_i, \mathbf{v}\in A_j} |\mathbf{x}|^2 \cos^2\theta_{\mathbf{xuv}} \lambda_{\mathbf{uv}}. \tag{16}$$

$cos(\theta)_{\mathbf{xuv}}$ is the angle between $\mathbf{x}$ and the connection of $\mathbf{u}$ and $\mathbf{v}$. Evidently, features in the feature space have a distribution that can be highly scattered. This method, which involves calculating pairs of points from pairs of classes, considers the spatial distribution of each class. It is superior to the LDA approach, which only considers the centers of each class. Conversely, a better classification plane is determined based on the closer point pairs. Closing point pair means the point pairs which is closer to the classification boundary of each two classes. Therefore, a natural extension of this method consists of assigning a weight to each point pair, depending on the distance between them:

$$\mathbf{S}_{oo} = \sum_{i,j} w_{ij} \sum_{x\in A_i, y\in A_j} w(d_{\mathbf{uv}}) \mathbf{S}_{\mathbf{uv}} \qquad i \neq j \tag{17}$$

the term $d_{\mathbf{uv}}$ is the distance between $\mathbf{u}$ and $\mathbf{v}$.

The easiest method for determining a weight is to use the reciprocal of a distance $\left(\frac{1}{d_{\mathbf{uv}}}\right)$,

$$\mathrm{w}(d_{\mathbf{uv}}) = {d_{\mathbf{uv}}}^{-n} \tag{18}$$

which can be expressed in the following form:

$$\mathrm{w}(d_{\mathbf{uv}}) = \begin{cases} =1 & \text{if } d_{\mathbf{uv}} \in (N_{\%} \sim M_{\%})(\max(d_{\mathbf{u}_i\mathbf{v}_j}) - \min(d_{\mathbf{u}_i\mathbf{v}_j})) \\ =0 & \text{if } d_{\mathbf{uv}} \notin (N_{\%} \sim M_{\%}) \max(d_{\mathbf{u}_i\mathbf{v}_j}) - \min(d_{\mathbf{u}_i\mathbf{v}_j})) \end{cases}. \tag{19}$$

The two approaches can be combined as follows:

$$\mathrm{w}(d_{\mathbf{uv}}) = \begin{cases} = {d_{\mathbf{uv}}}^{-n} & \text{if } d_{\mathbf{uv}} \in (N_{\%} \sim M_{\%})(\max(d_{\mathbf{u}_i\mathbf{v}_j}) - \min(d_{\mathbf{u}_i\mathbf{v}_j})) \\ = 0 & \text{if } d_{\mathrm{xy}} \notin (N_{\%} \sim M_{\%})(\max(d_{\mathbf{u}_i\mathbf{v}_j}) - \min(d_{\mathbf{u}_i\mathbf{v}_j})) \end{cases}. \tag{20}$$

For example, we can assign 1 or ${d_{\mathbf{uv}}}^{-1}$ to the point pairs as the weight for which the distance is in the range of $(2_{\%} \sim 10_{\%})(\max(d_{\mathbf{u}_i\mathbf{v}_j}) - \min(d_{\mathbf{u}_i\mathbf{v}_j}))$. Because data points of different classes might overlap, we added a weight to the interval such as (2% ∼ 10%), depending on the distance between the data pair of different types. Therefore, our approach involves adding nonzero weight to data pairs close to each other but excludes the nearest and farthest pairs. Thus, the distribution of closing point pairs is considered to calculate the projection vector, and the time cost is also effectively reduced. In the experiment, this approach produced a considerably strong performance. Hence, a more reasonable classification plane and projection direction were achieved.

Another benefit of our method is that it enables the $(s-1)$ low-rank problem to be overcome. The rank of $\mathbf{S}_{oo}$ is expressed as follows:

$$\mathrm{rank}(\mathbf{S}_{oo}) \leq \sum_{i\neq j} \mathrm{num}(\mathrm{data}_i) \times \mathrm{num}(\mathrm{data}_j). \tag{21}$$

Here, the rank is an upper bound and is affected only when data points are in one line passing through the origin. For real data, this condition is rare, and therefore, the rank of the matrix is close to the upper bound if the data point count is low. Otherwise, the matrix is full rank.

The above-mentioned method can be used to solve the $s-1$ low-rank problem, the heteroscedastic problem, the problem of the unreasonable between-class scatter matrix, and the problem of the data distribution between classes. However, the small-size problem remains to be solved. Classical LDA is a multiobjective optimization problem that can be used to simultaneously optimize the following equations:

$$\arg\max_{\mathbf{x}}\{\mathbf{x}^T\mathbf{S}_b\mathbf{x}\} \qquad \arg\min_{\mathbf{x}}\{\mathbf{x}^T\mathbf{S}_w\mathbf{x}\}. \tag{22}$$

A multiobjective optimization problem can be solved by optimizing several equations simultaneously or by combining multiple optimization targets. Classical LDA combines the maximization and minimization problems by dividing the first by the second, which causes the small-size problem. Therefore, the $\mathbf{S}_w$ must be full rank. However, changing the division to subtraction yields

$$\arg\max_{\mathbf{x}}\{\mathbf{x}^T\mathbf{S}_b\mathbf{x}+w_1\mathbf{x}^T\mathbf{S}_{oo}\mathbf{x}-w_2\mathbf{x}^T\mathbf{S}_w\mathbf{x}\} \tag{23}$$

Thus, the problem can be easily solved using a method similar to classical PCA and maximum scatter difference discriminant analysis [21,22]. We performed an eigenvalue decomposition on the following matrix:

$$\mathbf{S}_b+w_1\mathbf{S}_{oo}-w_2\mathbf{S}_w. \tag{24}$$

We ordered the eigenvectors according to their corresponding eigenvalues. The ordered eigenvectors were the expected projection vectors, which represent the solution for the objective function. Other studies have followed a similar strategy [21,22].

6 Tensor rank one discriminant analysis

TR1DA is an extension of the original LDA approach from the vector space to the tensor space [14]. This method is used for feature extraction and dimension reduction. Let $\mathcal{X}_{ij}^k$ denote the jth ($1\le j\le n_i$) training sample (tensor) in the ith ($1\le i\le c$) class and k be the kth extracted feature in the training procedure. There is a total of $n=\Sigma_{i=1}^c n_i$ training samples. We denote the mean tensor of the ith class in the kth iteration by $\mathcal{M}_i^k=\frac{1}{n_i}\Sigma_{j=1}^{n_i}\mathcal{X}_{ij}^k$. The total mean tensor in the kth iteration is $\mathcal{M}^k=\Sigma_{i=1}^c\frac{n_i}{n}\mathcal{M}_l^k$. The jth projection vector in the kth iteration is defined by $\mathbf{u}_k^j$. In TR1DA the kth feature extraction iteration is defined by

$$\mathcal{X}_{ij}^k=\mathcal{X}_{ij}^{k-1}-\lambda^{k-1}\mathbf{u}_{k-1}^1\otimes\mathbf{u}_{k-1}^2\otimes\ldots\otimes\mathbf{u}_{k-1}^M \tag{25}$$

$$\begin{aligned}\mathbf{u}_k^l|_{l=1}^M=arg_{\mathbf{u}_k^l|_{l=1}^M}\max\Bigg(&\frac{1}{n}\Sigma_{i=1}^c\left(\left(\mathcal{M}_i^k-\mathcal{M}^k\right)\prod_{l=1}^M\times_l\left(\mathbf{u}_k^l\right)^T\right)\\&\times\left(\left(\mathcal{M}_i^k-\mathcal{M}^k\right)\prod_{l=1}^M\times_l\left(\mathbf{u}_k^l\right)^T\right)^T\\&-\zeta_k^l\Sigma_{i=1}^c\Sigma_{j=1}^{n_i}\left(\left(\mathcal{X}_{ji}^k-\mathcal{M}_i^k\right)\prod_{l=1}^M\times_l\left(\mathbf{u}_k^l\right)^T\right)\\&\times\left(\left(\mathcal{X}_{ji}^k-\mathcal{M}_i^k\right)\prod_{l=1}^M\times_l\left(\mathbf{u}_k^l\right)^T\right)^T\Bigg).\end{aligned} \tag{26}$$

In the TR1DA case, there is no close form solution for the optimal projection vectors $\mathbf{u}_k^d|_{d=1}^M$. Thus, alternate projection methods are applied in order to overcome this issue. In fact, after getting the projection vectors, every tensor can be transformed into a vector in the feature space spanned by $\mathbf{u}_k^l|_{1\le k\le R}^{1\le l\le M}$; the corresponding value for the kth dimension is given by

$$\lambda^k=\mathcal{X}^k\prod_{l=1}^M\times_l\mathbf{u}_{k-1}^l. \tag{27}$$

where $\mathcal{X}^1=\mathcal{X}$ and $\mathcal{X}^k=\mathcal{X}^{k-1}-\lambda^{k-1}\prod_{l=1}^M\otimes\mathbf{u}_{k-1}^l$. Therefore, each tensor is mapped into an R-dimensional vector $\left[\lambda^1,\lambda^2,\ldots,\lambda^R\right]$. Once extracting feature from the tensor data is completed, the classification in the feature space becomes a much simpler task.

The benefits of TR1DA include [14] (1) a natural way of representing data without losing the structure information, (2) the small sample size problem is easier to solve as through the conventional discriminant learning the number of training samples becomes much smaller than the dimension of the feature space, (3) a better convergence during the training procedure, and (4) The (C-1) low-rank problem is partially solved using a strategy similar to the one used for the complementary space LDA approach which searches for the next projection vector in the subspace orthogonal to the space spanned by the projection vectors [23,24].

The main limitations of the tensor rank one discriminant analysis are (1) the heteroscedastic problem, (2) the problem of the unreasonable between-class scatter matrix, and (3) the problem of data distribution between classes not being considered.

7 Generalized tensor rank one discriminant analysis

Similarly, we define GTR1DA by replacing x and c by $\mathcal{X}$ and $\mathcal{M}$, respectively:

$$\mathcal{X}_{ij}^k=\mathcal{X}_{ij}^{k-1}-\lambda^{k-1}\mathbf{u}_{k-1}^1\otimes\mathbf{u}_{k-1}^2\otimes\ldots.\otimes\mathbf{u}_{k-1}^M \tag{28}$$

$$T=\left(\begin{array}{l}\left(\frac{1}{n}\Sigma_{i=1}^{c}\left(\left(\mathcal{M}_i^k-\mathcal{M}_k\right)\prod_{l=1}^{M}\times_l\left(\mathbf{u}_k^l\right)^T\right)\times\left(\left(\mathcal{M}_i^k-\mathcal{M}_k\right)\prod_{l=1}^{M}\times_l\left(\mathbf{u}_k^l\right)^T\right)^T\right.\\ \left.-\zeta_k^l\Sigma_{i=1}^{c}\Sigma_{j=1}^{n_i}\left(\left(\mathcal{X}_{ji}^k-\mathcal{M}_i^k\right)\prod_{l=1}^{M}\times_l\left(\mathbf{u}_k^l\right)^T\right)\times\left(\left(\mathcal{X}_{ji}^k-\mathcal{M}_i^k\right)\prod_{l=1}^{M}\times_l\left(\mathbf{u}_k^l\right)^T\right)^T\right)\end{array}\right) \tag{29}$$

Here, $\mathcal{M}^k$ is the mean tensor of all the means of each class $\mathcal{M}_i^k$. k is the kth projection tensor calculation iteration:

$$G=\left(\begin{array}{l}\sum_{i=1}^{C_c^2}\sum_{j_1=1}^{n_{i1}}\sum_{j_2=1}^{n_{i2}}\left(\left(\mathcal{X}_{j_1}-\mathcal{M}_{j_1j_2}\right)\prod_{l=1}^{M}\times_l\left(\mathbf{u}_k^l\right)^T\right)\times\left(\left(\mathcal{X}_{j_1}-\mathcal{M}_{j_1j_2}\right)\prod_{l=1}^{M}\times_l\left(\mathbf{u}_k^l\right)^T\right)^T\\ \quad+\left(\left(\mathcal{X}_{j_2}-\mathcal{M}_{j_1j_2}\right)\prod_{l=1}^{M}\times_l\left(\mathbf{u}_k^l\right)^T\right)\times\left(\left(\mathcal{X}_{j_2}-\mathcal{M}_{j_1j_2}\right)\prod_{l=1}^{M}\times_l\left(\mathbf{u}_k^l\right)^T\right)^T\end{array}\right). \tag{30}$$

Here, $\mathcal{M}_{j_1j_2}$ represents the mean of $\mathcal{X}_{j_1}$ and $\mathcal{X}_{j_2}$. Adding G to the TR1DA equation, the target function of GTR1DA becomes

$$\mathbf{u}_k^l|_{l=1}^M=arg_{\mathbf{u}_k^l|_{l=1}^M}\max(T+G) \tag{31}$$

$\mathbf{u}_k^l$ is the l mode projection vector of the projection tensor k. Since the above target function has no close form solution, an alternative projection method is adopted, and G becomes:

$$\left(\begin{array}{l}\sum_{i=1}^{C_c^2}\sum_{j_1=1}^{n_{i1}}\sum_{j_2=1}^{n_{i2}}\left(\left(\mathcal{X}_{j_1}-\mathcal{M}_{j_1j_2}\right)\bar{\times}_l\left(\mathbf{u}_k^l\right)^T\times_l\left(\mathbf{u}_k^l\right)^T\right)\\ \quad\times\left(\left(\mathcal{X}_{j_1}-\mathcal{M}_{j_1j_2}\right)\bar{\times}_l\left(\mathbf{u}_k^l\right)^T\times_l\left(\mathbf{u}_k^l\right)^T\right)^T\\ \quad+\left(\left(\mathcal{X}_{j_2}-\mathcal{M}_{j_1j_2}\right)\bar{\times}_l\left(\mathbf{u}_k^l\right)^T\times_l\left(\mathbf{u}_k^l\right)^T\right)\\ \quad\times\left(\left(\mathcal{X}_{j_2}-\mathcal{M}_{j_1j_2}\right)\bar{\times}_l\left(\mathbf{u}_k^l\right)^T\times_l\left(\mathbf{u}_k^l\right)^T\right)^T\end{array}\right) \tag{32}$$

$$\mathbf{u}_k^l\left(\begin{array}{l}\sum_{i=1}^{C_c^2}\sum_{j_1=1}^{n_{i1}}\sum_{j_2=1}^{n_{i2}}\left(\left(\mathcal{X}_{j_1}-\mathcal{M}_{j_1j_2}\right)\bar{\times}_l\left(\mathbf{u}_k^l\right)^T\right)\times\left(\left(\mathcal{X}_{j_1}-\mathcal{M}_{j_1j_2}\right)\bar{\times}_l\left(\mathbf{u}_k^l\right)^T\right)^T\\ \quad+\left(\left(\mathcal{X}_{j_2}-\mathcal{M}_{j_1j_2}\right)\bar{\times}_l\left(\mathbf{u}_k^l\right)^T\right)\times\left(\left(\mathcal{X}_{j_2}-\mathcal{M}_{j_1j_2}\right)\bar{\times}_l\left(\mathbf{u}_k^l\right)^T\right)^T\end{array}\right)\times\left(\mathbf{u}_k^l\right)^T. \tag{33}$$

Replacing the equation in the bracket yields

$$\mathbf{u}_k^l(G')(\mathbf{u}_k^l)^T. \tag{34}$$

Similarly, the target function of TR1DA can also be rewritten as

$$T'=\left(\begin{array}{l}\left(\frac{1}{n}\Sigma_{i=1}^{c}\left(\left(\mathcal{M}_i^k-\mathcal{M}_k\right)\bar{\times}_l\left(\mathbf{u}_k^l\right)^T\right)\times\left(\left(\mathcal{M}_i^k-\mathcal{M}_k\right)\bar{\times}_l\left(\mathbf{u}_k^l\right)^T\right)^T\right.\\ \left.-\zeta_k^l\Sigma_{i=1}^{c}\Sigma_{j=1}^{n_i}\left(\left(\mathcal{X}_{ji}^k-\mathcal{M}_i^k\right)\bar{\times}_l\left(\mathbf{u}_k^l\right)^T\right)\times\left(\left(\mathcal{X}_{ji}^k-\mathcal{M}_i^k\right)\bar{\times}_l\left(\mathbf{u}_k^l\right)^T\right)^T\right)\end{array}\right). \tag{35}$$

The optimization problem of GTR1DA is transformed into an m-subproblem, where m is the mode count of the original data:

$$\arg\max_{\mathbf{u}_k^l}=\left(\mathbf{u}_k^l\left(T'+G'\right)(\mathbf{u}_k^l)^T\right). \tag{36}$$

It is common that features in the feature space display a scattered distribution. The above method, which calculates pairs of points from pairs of classes, considers the spatial distribution of each class. These additional information renders it better than the LDA approach where only the center of each class is considered. On the other hand, a better classification plane can be determined if more closing point pairs are used. Thus, a natural extension to the above method is to assign a weight to each point depending on its position:

$$\begin{aligned}G&=\left(\sum_{i=1}^{C_c^2}\sum_{j_1=1}^{n_{i1}}\sum_{j_2=1}^{n_{i2}}(G_1+G_2)\right)\\ G_1&=\left(\left(\mathcal{X}_{j_1}-\mathcal{M}_{j_1j_2}\right)\prod_{l=1}^{M}\times_l\left(\mathbf{u}_k^l\right)^T\right)\\ &\quad\times\left(\left(\mathcal{X}_{j_1}-\mathcal{M}_{j_1j_2}\right)\prod_{l=1}^{M}\times_l\left(\mathbf{u}_k^l\right)^T\right)^T\\ G_2&=\left(\left(\mathcal{X}_{j_2}-\mathcal{M}_{j_1j_2}\right)\prod_{l=1}^{M}\times_l\left(\mathbf{u}_k^l\right)^T\right)\\ &\quad\times\left(\left(\mathcal{X}_{j_2}-\mathcal{M}_{j_1j_2}\right)\prod_{l=1}^{M}\times_l\left(\mathbf{u}_k^l\right)^T\right)^T.\end{aligned} \tag{37}$$

Adding a weight to the above equation leads to

$$G=\left(\sum_{i=1}^{C_c^2}\sum_{j_1=1}^{n_{i1}}\sum_{j_2=1}^{n_{i2}}w(d_{j_1j_2})(G_1+G_2)\right). \tag{38}$$

The simplest way of determining a weight is to use the inverse of the distance $\frac{1}{d_{\mathcal{X}_{j_1}\mathcal{X}_{j_2}}}$,

$$\mathrm{w}(d_{\mathcal{X}_{j_1}\mathcal{X}_{j_2}}) = {d_{\mathcal{X}_{j_1}\mathcal{X}_{j_2}}}^{-n} \tag{39}$$

or

$$\mathrm{w}(d_{\mathcal{X}_{j_1}\mathcal{X}_{j_2}}) = \begin{cases} =1 & \text{if } d_{\mathcal{X}_{j_1}\mathcal{X}_{j_2}} \in (N_{\%} \sim M_{\%})(\max(d_{\mathcal{X}_{j_1}\mathcal{X}_{j_2}}) - \min(d_{\mathcal{X}_{j_1}\mathcal{X}_{j_2}})) \\ =0 & \text{if } d_{\mathcal{X}_{j_1}\mathcal{X}_{j_2}} \notin (N_{\%} \sim M_{\%})(\max(d_{\mathcal{X}_{j_1}\mathcal{X}_{j_2}}) - \min(d_{\mathcal{X}_{j_1}\mathcal{X}_{j_2}})) \end{cases} \tag{40}$$

or by combining the two previous approaches:

$$\mathrm{w}(d_{\mathcal{X}_{j_1}\mathcal{X}_{j_2}}) = \begin{cases} ={d_{\mathcal{X}_{j_1}\mathcal{X}_{j_2}}}^{-n} & \text{if } d_{j_1 j_2} \in (N_{\%} \sim M_{\%})(\max(d_{\mathcal{X}_{j_1}\mathcal{X}_{j_2}}) - \min(d_{\mathcal{X}_{j_1}\mathcal{X}_{j_2}})) \\ =0 & \text{if } d_{\mathcal{X}_{j_1}\mathcal{X}_{j_2}} \notin (N_{\%} \sim M_{\%})(\max(d_{\mathcal{X}_{j_1}\mathcal{X}_{j_2}}) - \min(d_{\mathcal{X}_{j_1}\mathcal{X}_{j_2}})) \end{cases}. \tag{41}$$

Here, the method is similar to the one used to extend the classical LDA (Equations 18 and 19).

The main idea behind this geometrical approach is to fully consider the distribution of the data points of the different classes when they are close from one another while excluding their influence when they are far from the classification plane.

Algorithm 1 describes the GTR1DA procedure.

Algorithm 1 Generalized tensor rank one discriminant analysis

Require:

Input: Training tensors $\mathcal{X}_{ij}, 1 \le c, 1 \le j \le n_i$, the number R of rank one tensors allowed in GTR1DA

Output: The projection vectors $\mathbf{u}_k^l, 1 \le l \le M$,

1: Set $\mathcal{X}_{i,j}^1 = \mathcal{X}_{i,j}, 1 \le i \le c, 1 \le j \le n_i, \mathbf{u}_k^d =$ Optimal $\mathbf{u}_k^d$
2: **for** $k = 1$ to R **do**
3: **for** $t = 1$ to L **do**
4: **for** $l = 1$ to M **do**
5: update $\mathcal{X}_{i,j}^k$ according to Equations 27 and 28
6: calculate the class mean tensor $\mathcal{M}_j^k = (1/n_j)\sum_{i=1}^{n_j} \mathcal{X}_{i,j}^k$
7: calculate the total mean tensor $\mathcal{M}^k = (1/c)\sum_{j=1}^{c} \mathcal{X}_j^k$
8: calculate mean tensor of tensor pair $\mathcal{M}_{j_1 j_2} = (1/2)(\mathcal{X}_{j_1} + \mathcal{X}_{j_2})$
9: Update $\mathbf{u}_k^l$ according to Equation 36
10: **end for** % For loop in Step 4.
11: Convergence check: If $\left\|\mathbf{u}_k^l - \mathbf{u}_{k-1}^l\right\|_{Fro} \le \varepsilon$ for all directions l in the kth iteration, stop the loop in Step 3.
12: **end for** % For loop in Step 3.
13: **end for** % For loop in Step 2.

8 Justification of generalized tensor rank one discriminant analysis

As mentioned earlier (Section 4), the original LDA method has some major drawbacks since the small-size problem, the (C-1) low-rank problem, the heteroscedastic problem, and the unreasonable between-class scatter matrix need to be solved. Being an extension of LDA, TR1DA presents the same defects. Our new approach brings a new light on the matter as it does not suffer from these downsides. The most important characteristic of our approach is that it considers the distribution characteristics of the original data and of the adjacent points over the different classes.

Another important aspect of our algorithm is the convergence. In our approach, we adopt an alternative projection method. In the end, it can be proved that our algorithm is monotonic and that the target function for each iteration is monotonically decreasing. We define the target function using the mode and the iteration numbers:

$$\arg\max_{\mathbf{u}_k^l} = \left(\mathbf{u}_k^l \left(T' + G'\right) \left(\mathbf{u}_k^l\right)^T\right) \tag{42}$$

$$F(\mathbf{u}_k^l, k) = \left(\mathbf{u}_k^l \left(T' + G'\right) \left(\mathbf{u}_k^l\right)^T\right) \tag{43}$$

where k is the kth projection tensor and l is the mode number. Our algorithm generates a sequence of target function values for each mode l and projection tensor count k. The sequence is as follows:

$$\begin{aligned} F(\mathbf{u}_k^1, 1) \le F(\mathbf{u}_k^2, 1) \le \ldots \le F(\mathbf{u}_k^M, 1) \le F(\mathbf{u}_k^1, 2) \le F(\mathbf{u}_k^2, 2) \\ \le \ldots \le F(\mathbf{u}_k^1, k) \le F(\mathbf{u}_k^2, k) \\ \le \ldots \le F(\mathbf{u}_k^1, K) \le F(\mathbf{u}_k^2, K) \\ \le \ldots \le F(\mathbf{u}_k^M, K). \end{aligned} \tag{44}$$

The alternating projection algorithm is actually a composition of M subalgorithms. To check the convergence at each step and know whether or not to stop the algorithm, we calculate the value of the following equation and compare it to a given threshold:

$$\left\| \prod_{l=1}^{M} \otimes (\mathbf{u}_k^l)^T - \prod_{l=1}^{M} \otimes (\mathbf{u}_{k-1}^l)^T \right\|_{Fro}. \tag{45}$$

We use this method to determine whether the algorithm converges and to terminate the entire algorithm.

9 Optimal calculation of tensor learning approaches

In general, tensor learning algorithms use alternative optimization algorithms to calculate the result. The required number of iterations tends to be largely determined by the choice of the initial value. Therefore, we proposed an initial value selection algorithm to improve the computational efficiency of the tensor learning algorithm.

Since the original optimization problem is non-convex, it is possible to fall into the non-optimal local solution. Choosing a good initial value greatly affects the convergence property and hence the final result too. The general tensor method often assigns a randomly generated value or 1 to the initial value. This approach is often far from satisfactory. Since the goal is to compute an optimal solution in the rank 1 tensor space, as close as possible to the optimal solution of the original problem, we choose the initial value to be the closest rank 1 tensor to the optimal solution of the original vector space problem:

$$\min f(\mathbf{a}^{(1)},\ldots,\mathbf{a}^{(N)}) \equiv \frac{1}{2}\left\|\mathcal{Z}-\left[\left[\mathbf{a}^{(1)},\ldots,\mathbf{a}^{(N)}\right]\right]\right\|^2. \tag{46}$$

Here, the method for this problem is alternating least squares. The basic idea behind this algorithm is similar to the one used in supervised tensor learning. When calculating the value of a certain mode, we fix the other modes. A problem of this form is convex and therefore easy to solve. The equation is as follows [25-27]:

$$\min_{\mathbf{a}^{(n)}} f\left(\mathbf{a}^{(1)},\ldots,\mathbf{a}^{(N)}\right)=\frac{1}{2}\left\|\mathcal{Z}-\mathbf{a}^{(1)}\circ\cdots\circ\mathbf{a}^{(n)}\circ\cdots\circ\mathbf{a}^{(N)}\right\|^2. \tag{47}$$

Expanding the equation yields

$$\begin{aligned}=\min_{\mathbf{a}^{(n)}}\Bigg\|\mathcal{Z}_{(n)}-\mathbf{a}^{(n)}\Big(\mathbf{a}^{(N)}\otimes\cdots\otimes\mathbf{a}^{(n-1)}\otimes\\ \times\ \mathbf{a}^{(n+1)}\otimes\cdots\otimes\mathbf{a}^{(1)}\Big)^T\Bigg\|^2.\end{aligned} \tag{48}$$

$\otimes$ stands for the Kronecker product and $\mathcal{Z}_{(n)}$ means transforming a tensor $\mathcal{Z}$ into a matrix corresponding to mode n. The solution to this problem is

$$=\mathcal{Z}_{(n)}\left(\left(\mathbf{a}^{(N)}\otimes\cdots\otimes\mathbf{a}^{(n-1)}\otimes\mathbf{a}^{(n+1)}\otimes\cdots\otimes\mathbf{a}^{(1)}\right)^{\mathrm{T}}\right)^{\dagger}. \tag{49}$$

10 Classification

We take the popular and high-performance SVM [28] as our classifier. SVM tries to find the decision boundary that gives the smallest generalization error by maximizing the margin. Consider a classification task for two classes: $x_1, x_2, \ldots, x_N$ are N samples in the training set, and $t_1, t_2, \ldots, t_N$ are the corresponding label, where $t_n \in \{-1, +1\}$. As the data is not always separable linearly, the soft margin method is proposed as a modified maximum margin idea that allows for mislabeled examples, i.e., it allows some of the training set data points to be misclassified. The goal now is to maximize the margin while softly penalizing points that lie on the wrong side of the margin boundary. This forms the following primal optimization problem:

$$\begin{aligned}&\min_{W,b,\xi} C\Sigma_{n=1}^{N}\xi_n+\frac{1}{2}\|w\|^2\\ &\text{subject to } y_i(w^T\phi(x_i)+b)\geq 1-\xi_n, \xi_n\geq 0, i=1,2\ldots,n\end{aligned} \tag{50}$$

where the parameter $C > 0$ controls the trade-off between the slack variable penalty and the margin.The corresponding Lagrangian is given by

$$\begin{aligned}L(w,b,a)=\frac{1}{2}\|w\|^2+C\Sigma_{n=1}^{N}\xi_n\\ -\Sigma_{n=1}^{N}a_n\left\{t_n y(x_n)-1+\xi_n\right\}-\Sigma_{n=1}^{N}\mu_n\xi_n\end{aligned} \tag{51}$$

where $\{a_n \geq 0\}$ and $\{\mu_n \geq 0\}$ are Lagrange multipliers, and its dual Lagrangian is in the form

$$\tilde{L}(a)=\Sigma_{i=1}^{N}a_n-\frac{1}{2}\Sigma_{n=1}^{N}\Sigma_{n=1}^{N}a_n a_m t_n t_m k(x_n,x_m) \tag{52}$$

with the constraints $0 \leq a_n \leq C$ and $\Sigma_{n=1}^{N} a_n t_n = 0$, and $k(x, x') = \phi(x)^T\phi(x')$ is the kernel function. This is known as the kernel trick. It represents the inner product in the feature space. Kernels allow us to use feature spaces of high, even infinite, dimensionality. Polynomial kernel and Gaussian kernel are commonly used kernel functions. In our work, we select Gaussian kernel as the kernel function for its better performance.

The support vector machine is fundamentally a two-class classifier. In practice, however, there are various categories of heart diseases, so we have to tackle problems involving $K > 2$ classes of ECG signals. Various methods have been proposed to build a multiclass classifier from a set of two-class SVMs. Some commonly used approaches include 'one-versus-the-rest', 'one-versus-one', DAGSVM, and binary tree. Also, we take the one-versus-one strategy in our approach. Hsu and Lin [29] give a detailed comparison demonstrating that one-versus-one is a competitive strategy. In this approach, totally $K(K-1)/2$ different 2-class SVMs on all possible pairs of classes are trained. The test points are then classified to the class that has the highest number of 'votes'.

11 Experiments and results

11.1 12-Lead ECG database

To evaluate the performance of our method, we test it on a large dataset provided by a local hospital, courtesy of the SiWei medical company (Shanghai, China) and SiWei Remote ECG diagnostic center. This data is the real diagnostic data generated by a regular medical diagnostic system. The entire database accumulated by the SiWei Remote ECG diagnostic center covers 3 years and consists

Table 1 Dataset amount of each class

Beat type	N	L	R	V	S	E
Training beats	5,397	2,445	3,164	2,256	4,694	2,044
Testing beats	14,003	4,611	6,916	4,464	9,846	5,876

of 98,287 pieces of ECG data. Each piece is approximately a 20-s long diagnosis; the sampling rate is 500 Hz. The data is a standard 12-lead signal which includes channels I, II, III, aVR, aVL, aVF, V1, V2, V3, V4, V5, and V6. I, II, and III are limb leads. aVR, aVL, and aVF are augmented limb leads, while V1, V2, V3, V4, V5, and V6 are chest leads. The database consists of 1,251 kinds of single or mixed diseases. There are 249 single-disease categories. The dataset used in our experiment is a subset of the whole database. It consists of 3,000 pieces of high-quality 12-lead ECG records. Each piece includes 10 to 25 beats, for a total of 65,716 beats. These records originate from a wide variety of people: men, women, young, old, healthy, and unhealthy. The doctors' diagnoses are taken as label for the beats; it 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). Once the raw ECG signal has been preprocessed, we get the following single heartbeat segments: 19,400 N type; 7,056 L type; 10,080 R type; 6,720 V type; 14,540 S type; and 7,920 E type.[a] We then split the dataset into two categories: the training set and the test set. We use the former to generate the projection tensors and train the SVM model. The models are then used for the classification of the testing dataset. We randomly split the original dataset into two subsets: the training set consists of 20,000 beats, and the 45,716 remaining beats are used for testing. The ratio are 1:3 and 2:3 of the total data for the training and testing sets, respectively. The size of the training set and of the testing set for each specific type are listed in Table 1. Figure 2 presents an example of a 12-lead ECG signal found in our database. In the experiment, the set is randomly split; the listed result is an average over several experiments.

11.2 Results on dataset

We first compare the convergence of TR1DA to our extended version GTR1DA. The result is shown in Figure 3a. Note that the pattern shape of the two methods look alike; the same observation also applies to the shape of the curves. For the feature of the 5th, 8th, 12th, and 19th dimension, both GTR1DA and TR1DA need more iteration steps. However, by comparing the two methods, it is obvious that GTR1DA needs less steps to convergence than the original TR1DA method. Overall, the number of iterations required by our new method is about 1:3 of the original TR1DA. In our experiments, the iteration count threshold was fixed to 30. Therefore, when the iteration count reaches 30, in the TR1DA case, it may indicate that the algorithm did not converge. In fact, during the

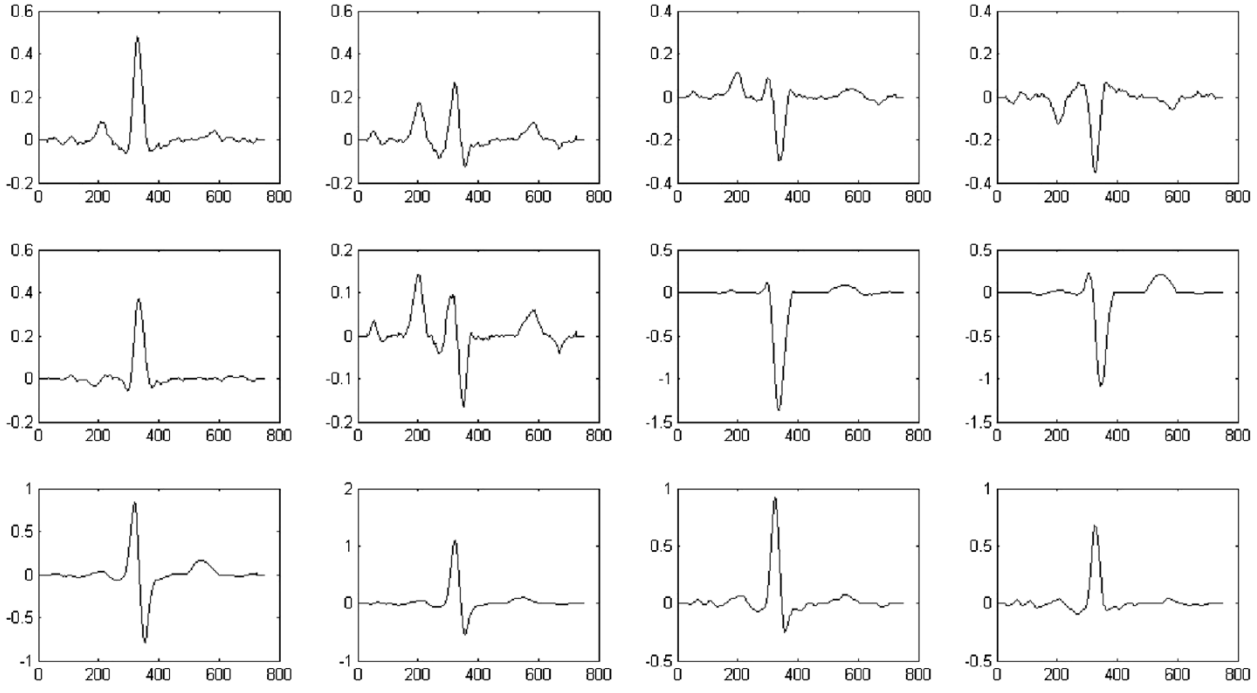

Figure 2 An example of 12-lead ECG signals in our ECG database.

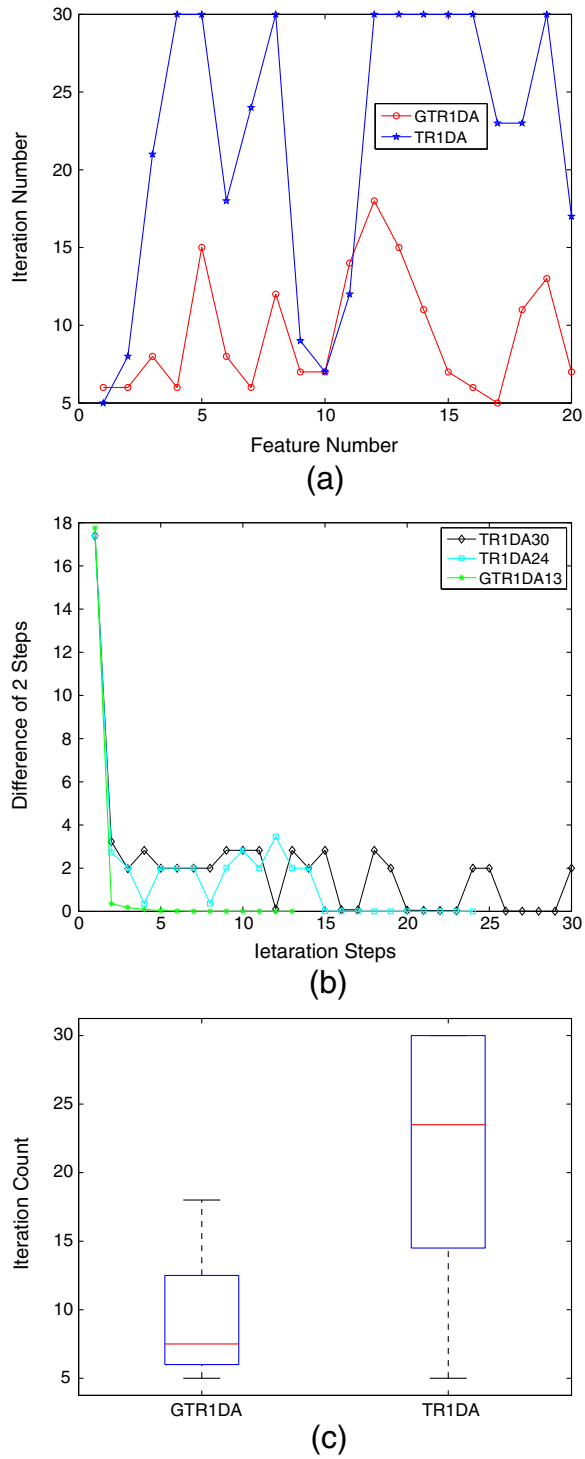

Figure 3 Comparison of the convergence condition of GTR1DA and TR1DA. (a) Iteration number of each dimension, **(b)** converge process of GTR1DA and TR1DA (number behind the method name is the iteration count), and **(c)** iteration count statistics.

experiment, if TR1DA does not converge within 30 steps, it iterates with a small gap and, in most cases, cannot converge even with more steps.

On Figure 3b, the Frobenius norm of the difference of a two-step projection tensor is shown. GTR1DA appears to converge very fast and very smoothly, without any fluctuations or sudden increase in the process. It remains substantially and monotonically decreasing; on the contrary, TR1DA starts fluctuating after a few steps. Despite this behavior, it sometimes achieves good convergence, although it always requires more iteration steps than GTR1DA. It can also happen that TR1DA fails to converge after 30 iterations while only fluctuating in a very small range. The number following the method name in the diagram represents the total number of iterations. Figure 3c shows a statistics of the iteration count of GTR1DA and TR1DA. GTR1DA displays an evident advantage over TR1DA.

In addition, since both TR1DA and GTR1DA are tensor learning approaches, the selection of a proper initial value has a great impact on the convergence of the algorithm. Figure 4 shows the improvement brought by our optimal initial value selection method. It compares the number of iterations when run with a random initial value and with our optimal initial value. The difference in the final result is quite large as in the random case. The number of iterations fluctuates between four and seven, against three and four using our optimal initial value.

Figure 4 shows how important a proper choice for the initial value is. Indeed, it not only improves the convergence speed but also decreases the number of iterations, leading to a much higher computational efficiency. The iteration step count is more stable: fluctuating less and displaying only small variations. Figure 5 shows the time cost of calculating the initial value. Actually, a good initial value sometimes leads to a better solution. Here, we compare the target function value of the calculated initial value and the random initial value in Figure 6. Sometimes, the calculated initial value leads to a better solution.

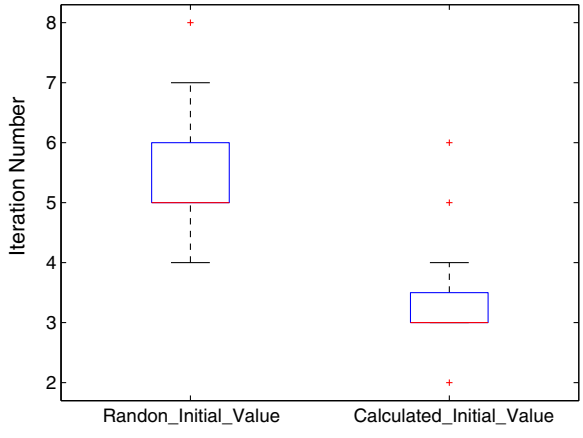

Figure 4 Iteration number comparison of random initial value and calculated optimal initial value.

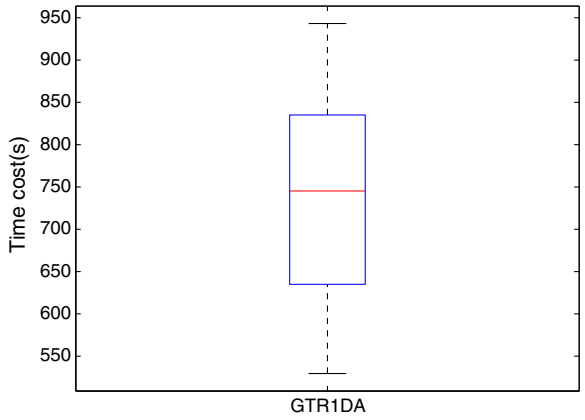

Figure 5 Time cost of calculating the initial value.

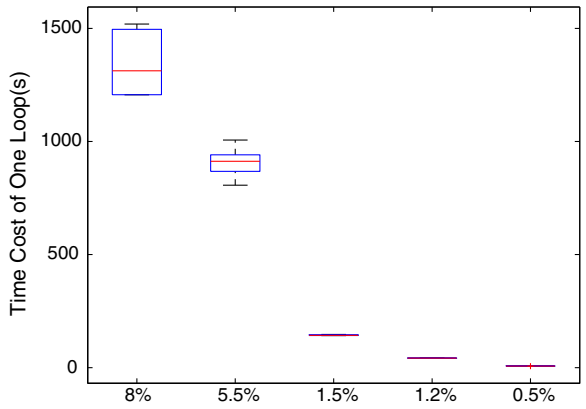

Figure 7 Single-loop time cost of different closing point proportions.

In GTR1DA, we consider the closing points of different classes to calculate the optimal projection tensor. In fact, choosing an optimal proportion requires a trade-off between efficiency and effectiveness. A lower proportion leads to a higher efficiency as in Figure 7, but at the same time, the effectiveness follows a curve which depends on the point proportion. Therefore, using either a too-high or a too-low proportion can lower the classification accuracy.

Table 2 compares the classification accuracy for different point proportions. It lists each class accuracy and average accuracy for the following m% values in Equation 19: 0.5%, 1.2%, 1.5%, 5.5%, and 8%. To prevent the overlapping point pairs, the n% is set with 0.2%. Through this comparison, we observe that the 1.2% proportion results in the highest classification accuracy.

On the other hand, the proportion can greatly influence the time cost. Figure 7 compares the time cost of one loop for the proportions 0.5%, 1.2%, 1.5%, 5.5%, and 8%. It is clear on the figure that a higher proportion implies a higher time cost and greater fluctuations and variations. One loop using 8% takes from 1,200 to 1,500 s, while with 5.5%, it is between 800 and 1,000 s. It even drops down to 120, 50, and 20 s for 1.5% 1.2%, and 0.5%, respectively. As the fluctuations in the latter two cases remain small, we need to reduce the proportion of the points as much as possible in order to improve the computational efficiency while maintaining a high classification accuracy.

Figure 8 shows the convergence process with respect to the time cost. Figure 9 represents the statistics of a one-feature calculation time. In this case, the closing point proportion is 1.2%. Also, note that the time cost remains acceptable compared to that in TR1DA.

As for the ECG signal processing, the most important is the classification accuracy. Therefore, we compared the following different methods over the dimensions 1 to 20 in Figure 10: GTR1DA, TR1DA, UMPCA, PCA, ICA, LDA, and regularized LDA. As for rLDA, PCA, ICA, and LDA, the tensor data is first expanded into a vector, and then the projection vectors are calculated based on the vectorized data. Finally, the top 20 projection vectors are selected to calculate the features exploited in the classification.

Figure 10 highlights the superiority of the tensor-based approach over the vector space-based methods. The main reason is that tensor-based approaches take the tensor

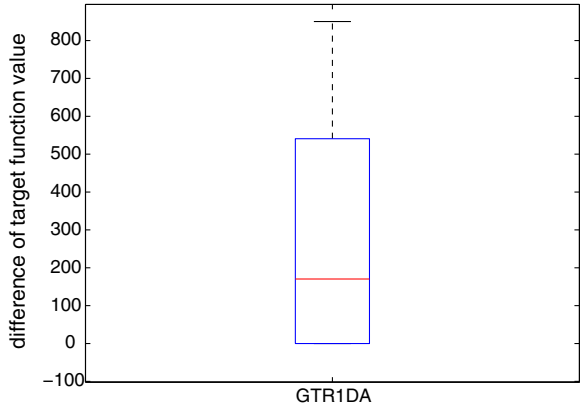

Figure 6 The target function value difference. Calculated initial value target function value minus random initial value target function value.

Table 2 Classification accuracy of different closing points proportion

Percentage (%)	N	L	R	V	S	E	Average
8	97.34	96.80	97.65	96.30	97.47	97.67	97.21
5.5	98.45	97.67	99.20	97.56	97.26	98.47	98.10
1.5	99.11	98.53	100.00	99.60	99.90	99.60	99.46
1.2	99.86	99.73	100.00	100.00	100.00	100.00	99.93
0.5s	98.07	97.24	99.75	98.50	97.65	96.53	97.96

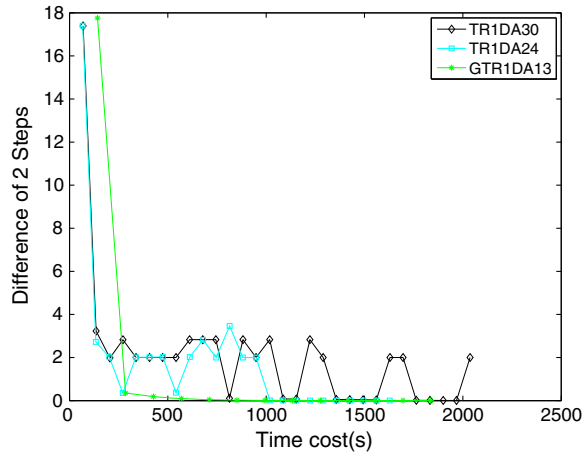

Figure 8 Converge process of GTR1DA and TR1DA while calculating one projection tensor with time cost.

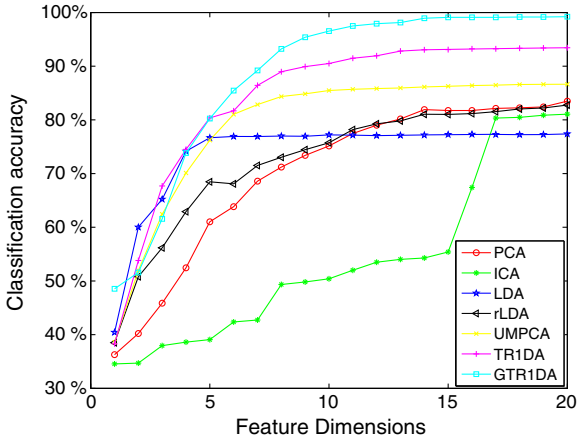

Figure 10 Classification accuracy of different approaches with different feature dimensions.

data directly as input, preventing SSS problem [7]. Moreover, it enables the utilization of the structural information in the tensor data. The original TR1DA displays a higher classification accuracy under low dimension conditions; however, for higher dimensions, GTR1DA outperforms TR1DA. Both perform better than UMPCA though. As for the vector space methods, from the best to the worst, they are classified as follows for the lower dimension case: LDA, rLDA, PCA, and ICA. They are classified as follows for higher dimensions: PCA, rLDA, ICA, and LDA. ICA displays a sudden accuracy increase after dimension 15.

Table 3 presents a clear comparison of the different methods depending on the classification accuracy of each class. It is obvious that TR1DA and GTR1DA have higher accuracy and outperform the others. In addition, our method GTR1DA performs better than the original TR1DA. Four classes of ECG have a 100% accuracy, so it is obvious that our extension impacts the experiments.

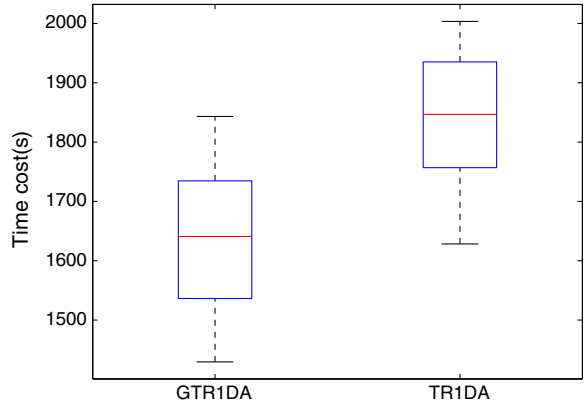

Figure 9 Total time cost of calculating one projection tensor comparison between GTR1DA and TR1DA.

To further illustrate the effectiveness of our approach and show a clear illustration of better class separation, we extract three dimension features of three classes using some feature extraction methods and plot their 3D distribution (Figure 11).

In Figure 11, it is clear that GTR1DA, TR1DA, and LDA display cluster characteristics. This is to be contrasted by other methods such as PCA, ICA, and UMPCA. Both LDA and TR1DA seem to display a better cluster characteristic; however, there is an obvious overlap at the boundary of each two classes. Our method calculates the projection tensors based on the data distribution nearing the boundary of each two classes. That explains why we achieve an almost 100% accuracy. Hence, in the end, GTR1DA features a better classification accuracy than the other two methods.

To analyze the performance of our approach, we compare the classification accuracy of the wavelet transform approach [1], of the neural networks approach for determining the features of a given ECG signal [2], of the active learning method [3], and of GTR1DA. The results are shown in Table 4. Our method clearly displays an advantage over all the other methods.

Table 3 Comparison of different approaches' classification accuracy

Model	N (%)	L (%)	R (%)	V (%)	S (%)	E (%)
PCA	87.79	76.54	83.62	93.90	97.30	86.97
ICA	92.96	83.09	87.97	98.10	97.10	84.43
LDA	83.64	64.77	67.38	85.50	95.10	73.60
UMPCA	94.43	85.43	86.46	92.20	90.80	91.45
TR1DA	92.96	93.53	95.26	93.26	100.00	92.16
GTR1DA	99.86	99.73	100.00	100.00	100.00	100.00

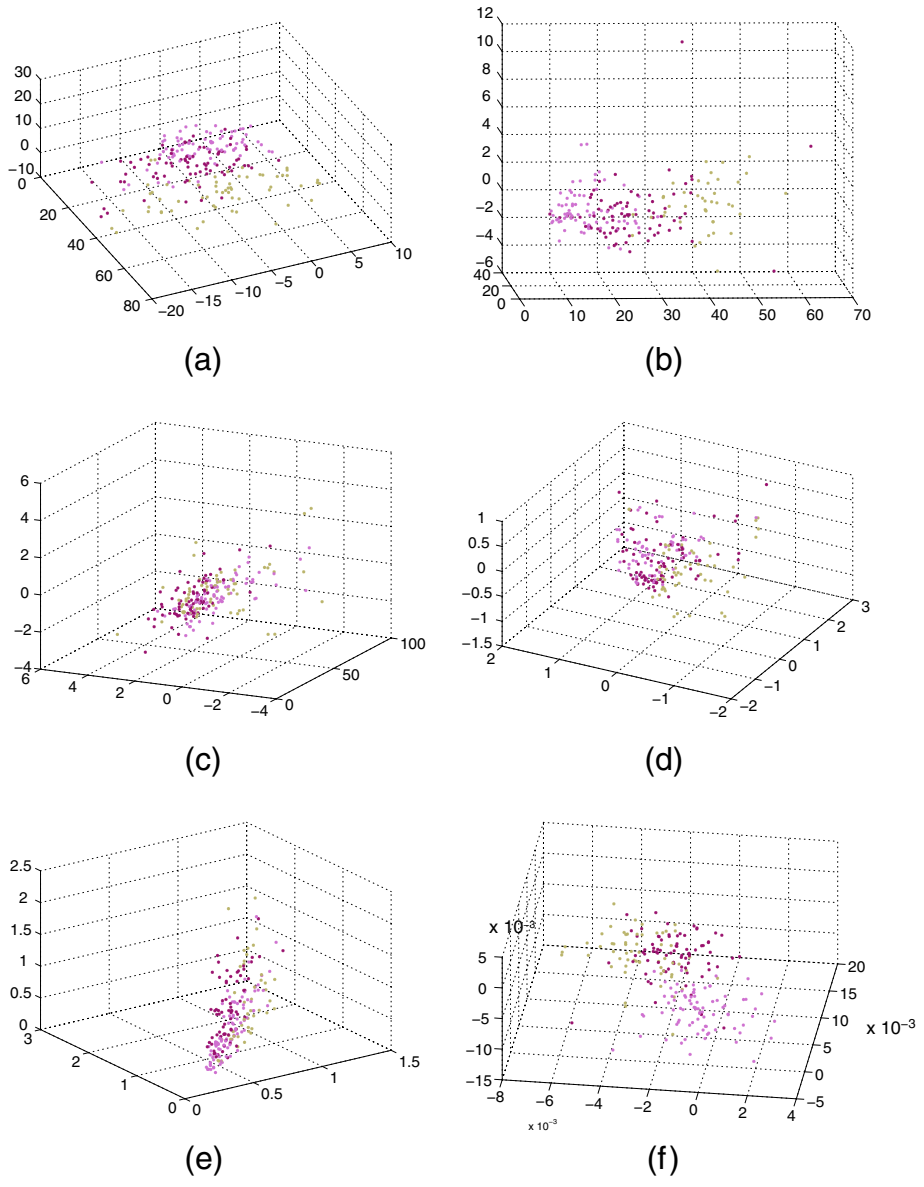

Figure 11 3D Distribution of extracted three dimension features. (a) GTR1DA, **(b)** TR1DA, **(c)** UMPCA, **(d)** PCA, **(e)** ICA, and **(f)** LDA.

Table 4 Comparison of our approach with other state-of-the-art ECG classification methods

Model	N (%)	L (%)	R (%)	V (%)	S (%)	E (%)
Wavelet	97.86	93.67	91.89	94.75	92.74	95.79
Neural networks	94.68	95.37	92.57	96.58	98.54	99.73
Active learning	93.36	94.49	89.47	96.37	91.63	92.47
GTR1DA	99.86	99.73	100.00	100.00	100.00	100.00

12 Conclusions

To solve the feature extraction, dimension reduction, and classification problems of large 12-lead hospital standard ECG datasets, a new tensor-based scheme was proposed. To determine the most discriminative feature, the 12-lead ECG was first transformed into a tensor in the spatial-spectral-temporal domain using STFT. The primary objective of extending the original TR1DA method was to extract the effective feature using the tensor data as direct inputs to overcome the SSS problem and take advantage of the structure information contained therein. In addition, our extension provides a new method for solving the heteroscedastic problem and the unreasonable

between-class scatter matrix problem by considering the distribution of closing points near the boundary of the various classes. To improve the calculation speed and stability, an optimal calculation method for tensor learning approaches was also proposed to compute an optimal initial value. The results of the experiment showed that our new tensor-based scheme outperforms traditional vector-based methods. In addition, classification is much more accurate when using GTR1DA than when using the original TR1DA and UMPCA methods. The experiment also shows a critical performance improvement caused by a judicious choice of the initial value. This indicates that GTR1DA, and tensor-based schemes in general, are well fitted, effective, and robust data exploration tools for classifying 12-lead ECG signals.

Endnote

[a] We will make the ECG data public. Please see the website http://bcmi.sjtu.edu.cn/ehealth/.

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

Acknowledgements
The work was supported by the National Natural Science Foundation of China (grant nos. 91120305 and 61272251).

Received: 29 June 2013 Accepted: 26 December 2013
Published: 14 January 2014

References

1. Q Zhao, L Zhang, ECG feature extraction and classification using wavelet transform and support vector machines, in *2005 International Conference on Neural Networks and Brain* (IEEE, Piscataway, 2005), pp. 1089–1092
2. Y-R Hwang, K-K Jen, ECG feature extraction and classification using cepstrum and neural networks. J. Med. Biol. Eng. **28**(1), 31 (2008)
3. E Pasolli, F Melgani, Active learning methods for electrocardiographic signal classification. IEEE Trans. Inf. Technol. Biomed. **14**(6), 1405–1416 (2010)
4. H Zhang, L Zhang, ECG analysis based on PCA and support vector machines. International Conference on Neural Networks and Brain **2**, 743–747 (2005)
5. Y Wu, L Zhang, ECG classification using ICA features and support vector machines, in *ICONIP (1)*, ed. by Lu B L, Zhang L, and Kwok J T. Lecture notes in Computer Science, vol. 7062 (Springer, Heidelberg, 2011), pp. 146–154
6. K Huang, L Zhang, Y Wu, ECG classification based on non-cardiology feature, in *Proceedings of the 9th international conference on Advances in Neural Networks - volume part II, ISNN'12* (Springer, Heidelberg, 2012), pp. 179–186
7. Y He, Solving undersampled problem of IDA using Gram-Schmidt orthogonalization procedure in difference space, in *Proceedings of the 2009 International Conference on Advanced Computer Control,* Washington DC (IEEE Computer Society, Piscataway, 2009), pp. 153–157
8. J Li, L Zhang, D Tao, H Sun, Q Zhao, A prior neurophysiologic knowledge free tensor-based scheme for single trial eeg classification. IEEE Trans. Neural Syst. Rehabil. Eng. **17**(2), 107–115 (2009)
9. D Tao, X Li, X Wu, SJ Maybank, General tensor discriminant analysis and Gabor features for gait recognition. IEEE Trans. Pattern Anal. Mach. Intell. **29**(10), 1700–1715 (2007)
10. H Lu, KN Plataniotis, AN Venetsanopoulos, Multilinear principal component analysis of tensor objects for recognition, in *Proc. Int. Conf. on Pattern Recognition* **2**, (2006), pp. 776–779
11. H Lu, KN Plataniotis, AN Venetsanopoulos, MPCA: multilinear principal component analysis of tensor objects. IEEE Trans. Neural Netw. **19**(1), 18–39 (2008)
12. H Lu, KN Plataniotis, AN Venetsanopoulos, Uncorrelated multilinear principal component analysis through successive variance maximization, in *25th International Conference on Machine Learning (ICML)* (Helsinki, 2008)
13. H Lu, KN Plataniotis, AN Venetsanopoulos, Uncorrelated multilinear principal component analysis for unsupervised multilinear subspace learning. IEEE Transactions on Neural Networks **20**(11), 1820–1836 (2009)
14. D Tao, X Li, X Wu, S Maybank, Tensor rank one discriminant analysis-a convergent method for discriminative multilinear subspace selection. Neurocomput. **71**(10–12), 1866–1882 (2008)
15. H Lu, KN Plataniotis, AN Venetsanopoulos, A survey of multilinear subspace learning for tensor data. Pattern Recognit. **44**(7), 1540–1551 (2011)
16. JB Allen, LR Rabiner, A unified approach to short-time Fourier analysis and synthesis. Proc. IEEE **65**(11), 1558–1564 (1977)
17. Y Panagakis, C Kotropoulos, GR Arce, Non-negative multilinear principal component analysis of auditory temporal modulations for music genre classification. Trans. Audio, Speech Lang. Proc. **18**(3), 576–588 (2010)
18. F Miwakeichi, PA Valdes-Sosa, E Aubert-Vazquez, JB Bayard, J Watanabe, H Mizuhara, Y Yamaguchi, Decomposing EEG data into space-time-frequency components using parallel factor analysis and its relation with cerebral blood flow, in *Neural Information Processing* (Springer, Heidelberg, 2008), pp. 802–810
19. M De Vos, A Vergult, L De Lathauwer, W De Clercq, S Van Huffel, P Dupont, A Palmini, W Van Paesschen, Canonical decomposition of ictal scalp {EEG} reliably detects the seizure onset zone. NeuroImage. **37**(3), 844–854 (2007)
20. E Acar, C Aykut-Bingol, H Bingol, R Bro, B Yener, Multiway analysis of epilepsy tensors, in *Proceedings 15th International Conference on Intelligent Systems for Molecular Biology (ISMB) 6th European Conference on Computational Biology (ECCB), Vienna, Austria, July 21–25, 2007* (Oxford University Press, Oxford, 2007), pp. 10–18
21. F Song, D Zhang, D Mei, Z Guo, A multiple maximum scatter difference discriminant criterion for facial feature extraction. IEEE Trans. Syst. Man Cybern Part B **37**(6), 1599–1606 (2007)
22. J Gao L Xiang, Laplacian maximum scatter difference discriminant criterion. ICAIC **224**(1), 691–697 (2011)
23. JM Leiva-Murillo, A Artés-Rodríguez, Maximization of mutual information for supervised linear feature extraction. IEEE Trans. Neural Netw. **18**(5), 1433–1441 (2007)
24. C Dhir, S Lee, Discriminant independent component analysis, in *Intelligent Data Engineering and Automated Learning - IDEAL 2009*, ed. by E Corchado, H Yin. Lecture Notes in Computer Science, vol. 5788 (Springer, Heidelberg, 2009), pp. 219–225
25. G Takács, D Tikk, Alternating least squares for personalized ranking, in *Proceedings of the Sixth ACM Conference on Recommender Systems, RecSys '12* (ACM, New York, 2012), pp. 83–90
26. E Acar, TG Kolda, DM Dunlavy, An optimization approach for fitting canonical tensor decompositions. Technical report, Sandia National Laboratories (2009)
27. TG Kolda, BW Bader, Tensor decompositions and applications. SIAM Rev. **51**(3), 455–500 (2009)
28. CM Bishop, *Pattern Recognition and Machine Learning,* 1st edn. (Springer, Heidelberg, 2006)
29. CW Hsu, CC Chang, CJ Lin. *A Practical Guide to Support Vector Classification,* Technical report, National Taiwan University (2000)

doi:10.1186/1687-6180-2014-2
Cite this article as: Huang and Zhang: **Cardiology knowledge free ECG feature extraction using generalized tensor rank one discriminant analysis.** *EURASIP Journal on Advances in Signal Processing* 2014 **2014**:2.