Frontiers of Information Technology & Electronic Engineering
www.jzus.zju.edu.cn; engineering.cae.cn; www.springerlink.com
ISSN 2095-9184 (print); ISSN 2095-9230 (online)
E-mail: jzus@zju.edu.cn

Frequency-hopping transmitter fingerprint feature recognition with kernel projection and joint representation*

Ping SUI†‡, Ying GUO, Kun-feng ZHANG, Hong-guang LI

Institute of Information and Navigation, Air Force Engineering University, Xi'an 710077, China

†E-mail: ziwuningxin@163.com

Received Jan. 9, 2018; Revision accepted Apr. 13, 2018; Crosschecked Aug. 14, 2019

Abstract: Frequency-hopping (FH) is one of the commonly used spread spectrum techniques that finds wide applications in communications and radar systems because of its inherent capability of low interception, good confidentiality, and strong anti-interference. However, non-cooperation FH transmitter classification is a significant and challenging issue for FH transmitter fingerprint feature recognition, since it not only is sensitive to noise but also has non-linear, non-Gaussian, and non-stability characteristics, which make it difficult to guarantee the classification in the original signal space. Some existing classifiers, such as the sparse representation classifier (SRC), generally use an individual representation rather than all the samples to classify the test data, which over-emphasizes sparsity but ignores the collaborative relationship among the given set of samples. To address these problems, we propose a novel classifier, called the kernel joint representation classifier (KJRC), for FH transmitter fingerprint feature recognition, by integrating kernel projection, collaborative feature representation, and classifier learning into a joint framework. Extensive experiments on real-world FH signals demonstrate the effectiveness of the proposed method in comparison with several state-of-the-art recognition methods.

Key words: Frequency-hopping; Fingerprint feature; Kernel function; Joint representation; Transmitter recognition
https://doi.org/10.1631/FITEE.1800025 **CLC number:** TN911.72

1 Introduction

The largest difference between frequency-hopping (FH) and fixed-frequency communication is the pseudo random jump over time; as a result, the research in FH signal reconnaissance focuses mostly on signal detection and parameter estimation. Angelosante et al. (2010) proposed a sparse linear regression based multiple FH parameter estimation method. Zhao et al. (2015) proposed a robust FH spectrum estimation method based on sparse Bayesian. Liu et al. (2018) proposed an FH spectrum estimation method based on structure-aware Bayesian compressive sensing, which can be used for conditions which lack observations. However, these algorithms are all about parameter estimation, and there is relatively little research on the feature extraction and classification of FH signals.

With the increasing amount of radiation such as radar and communications, the electromagnetic environment is increasingly complicated, and the signals received by electronic receivers are increasingly complicated. Specifically, in the case of various types of radiation with the same type of systems and parameters, how to effectively identify these radiation sources has been a major problem in the field of signal processing. Due to the high anti-jamming performance of FH communication and the low probability of interception, individual identification of FH transmitters as a specific application of radiation source identification has been of considerable concern for scholars.

‡ Corresponding author

* Project supported by the National Natural Science Foundation of China (No. 61601500)

ORCID: Ping SUI, http://orcid.org/0000-0001-9132-9431

For individual recognition of FH transmitters, there are individual nuances between the internal originals of any two identical transmitters even if they are from the same production line. Transient signals transmitted by FH transmitters contain different impulse responses during radio switching on, mode switching, frequency switching, power supply excitation changes, etc. Therefore, these transient response signals contain abundant information about inherent transmitter features. In the meantime, during the normal communication process of an FH transmitter, the steady-state signal includes the individual nuances of certain FH transmitters, and these inherent features are unique to individual transmitters. Because of the individual nuances of FH transmitters, there exist inherent features, and these can be used to identify individual transmitters. Such an inherent feature based on individual nuances can be called the fingerprint of the transmitters.

Recently there have been many classification algorithms for the individual recognition of radiation sources, such as the decision tree algorithm (Yoshikawa et al., 1995; Tadjudin and Landgrebe, 1996; Friedl and Brodley, 1997; Lawrence and Wright, 2001), which has the advantages of high flexibility, good intuition, strong robustness, and high computational efficiency, and is applied mostly in image classification. However, there are still limitations. For example, Quinlan proposed an ID3 algorithm in 1979. This method uses local non-retrospective heuristics of information gain, from which it is difficult to obtain the global optimal decision tree. Subsequently, Quinlan (1993) proposed the C4.5 algorithm to improve ID3, but because of the need for multiple sequential scanning and sorting of the sample dataset, the algorithm is inefficient. The *k*-nearest neighbor algorithm (*k*-NN) (Cover and Hart, 1967; Wu XD et al., 2008) is a common clustering method because of its advantages of being simple, effective, and easy to implement. However, its clustering performance is affected by the nearest neighbor parameter *k*, the similarity measure of neighbor points, and the size and distribution of data sets, and therefore there are significant differences in the efficiency of different algorithms. The support vector machine (SVM) algorithm (Cherkassky, 1997; Cherkassky and Mulier, 1998), which is a pattern classification technique proposed by Vapnik in 1995, is effective in solving the small sample classification problem. At the same time, this method can realize the classification of non-linear problems, using the kernel trick to map the original non-linear problem into its linear high-dimensional feature space, where samples belonging to the same class can be better grouped. However, the major disadvantage of this method is that when the number of training samples is too large, the efficiency of the algorithm is very low, and different kernel functions have different classification results.

The above classification algorithms are not ideal when they are used for the FH transmitter recognition problem. At the same time, the fingerprints of FH transmitters are subtle, when it comes to the condition of serious noise and a complicated electromagnetic environment, especially when the transmitters are non-cooperative. The practical classification of such methods is not ideal. In recent years, with the advantages of being insensitive to noise and having good classification performance, some methods based on sparse representation have been developed in various fields of classification. Wright et al. (2009) proposed a sparse representation based classifier (SRC) for a given set of testing data. This method codes every sample over the training data as a sparse representation, and then classifies it into the class with the least representation error. However, this method turns the sparse representation problem into L_1-minimization optimization, which is very computationally demanding. This method uses an individual representation rather than a collective one to classify such a set of data, and in doing so obviously ignores correlation among the given data. Consequently, a collaborative representation classifier (CRC) based on L_2-minimization was proposed in Zhang et al. (2011) and Yang M et al. (2012). The experimental results verified that CRC is significantly more computationally efficient and can result in a similar performance to SRC. Also, Wang and Chen (2017) proposed a joint representation classification (JRC) for collective face recognition. This codes all the testing samples over the base samples simultaneously to facilitate recognition. Experimental results showed that this method not only greatly reduces computational cost but also achieves better performance. However, all these methods are conducted in the original signal space rather than the non-linear high-dimensional feature space. The performance of the FH transmitter fingerprints is generally of an irregular non-stationary, non-linear, and non-Gaussian nature, and thus the effectiveness of these methods for the recognition of the FH transmitter fingerprint feature is difficult to guarantee in the original signal space. To address these weaknesses and make sufficient use of the collaborative relationship among the given set of training data, a novel FH transmitter fingerprint feature recognition method is proposed, by integrating kernel projection, feature representation, and classifier learning into a joint framework. Here the proposed method extracts the square integrated bi-spectral (SIB) feature of the original FH signals to characterize the fingerprint features of the individual FH transmitters first, and then a Gaussian kernel is used for feature representation. Given that the given samples are generally related to each other, this method takes the correlation of multiple samples and a single representation into account. Then a joint representation framework is designed for the recognition problem. At the same time, a unified expression of recognition is developed for the final optimization problem. Extensive experiments on real-world FH signals show the effectiveness of the proposed method. Fig. 1 gives the process of FH transmitter recognition based on a kernel joint representation classifier.

The main contributions of this paper are summarized as follows:

1. The FH transmitter fingerprint feature recognition method is proposed by integrating kernel projection, feature representation, and classifier learning into a joint framework. Using a kernel function, this method maps the original signal space into its non-linear higher-dimensional feature space, in which features belonging to the same class can be better grouped.

2. Given that the given samples are generally related to each other, the collaboration among the given samples is considered in our formulation to obtain robust experimental results.

3. By joint representation this method can implement the kernel function, feature representation, and classifier learning simultaneously, which is more economical and efficient.

4. The proposed method boosts the recognition results of the frequency-hopping transmitter fingerprint feature on five real-world transmitters in comparison with state-of-the-art classification methods.

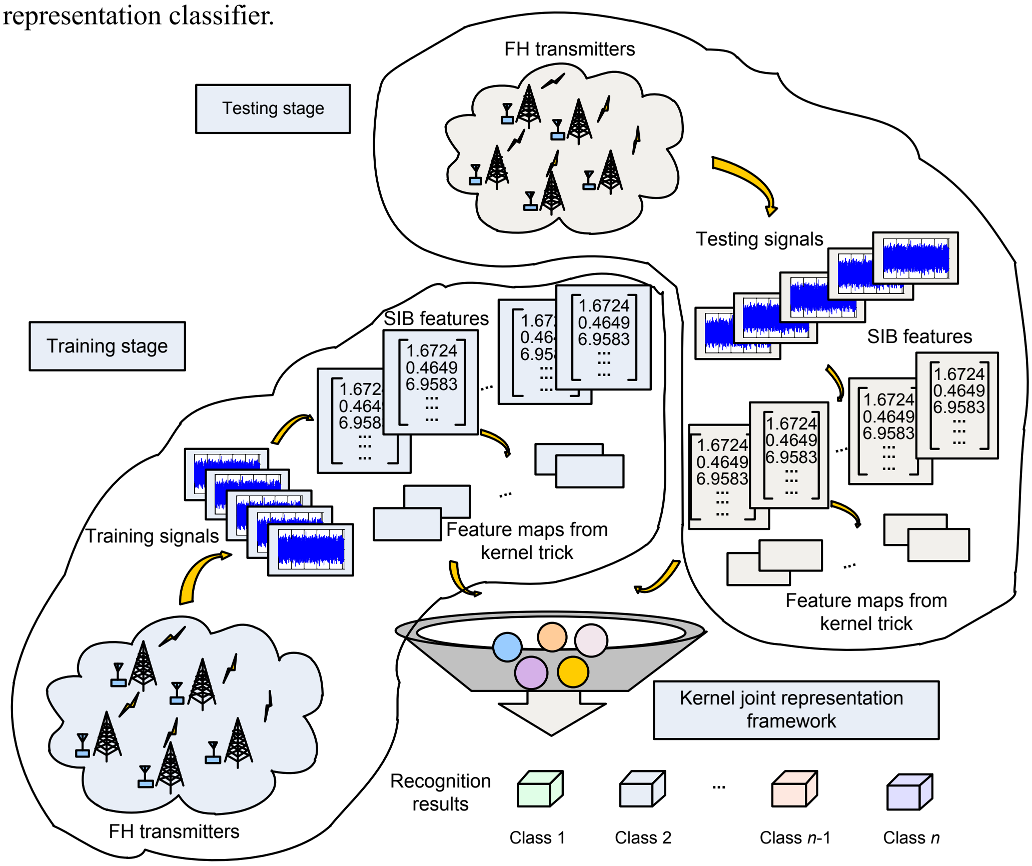

Fig. 1 The transmitter fingerprint feature classification process of our method

2 Preliminaries

2.1 Kernel trick

The kernel trick is a well-known technique in pattern recognition that can project a linear algorithm to its non-linear counterpart by a mapping function. After such non-linear transformation, the recognition can be conducted in the intrinsic non-linear higher-dimensional feature space where a decision line can be used to classify samples efficiently rather than the original signal space.

In kernel learning the following idea is used via a non-linear mapping (Muller et al., 2001):

$$\Phi:\mathbb{R}^d\rightarrow\mathbb{F},\ \boldsymbol{x}\rightarrow\Phi(\boldsymbol{x}). \tag{1}$$

The data $\boldsymbol{x}$ in the original signal space $\mathbb{R}^d$ is mapped into a potentially higher-dimensional space $\mathbb{F}$, where the intrinsic non-linear features can be better represented. For a given learning problem one now considers the same algorithm in $\mathbb{F}$ instead of $\mathbb{R}^d$, i.e., $\Phi(\boldsymbol{x})\in\mathbb{F}$. Given this mapped representation a simpler classification or regression might be found in $\mathbb{F}$ than in $\mathbb{R}^d$.

The key of the kernel trick is to provide a method to perform statistical learning by directly using the inner product function (namely kernel function $k(\boldsymbol{x},\ \boldsymbol{y})=<\Phi(\boldsymbol{x}),\ \Phi(\boldsymbol{y})>$) defined in the data feature space. Since there is no need to specify a specific non-linear mapping, the corresponding mapping $\Phi(\boldsymbol{x})$ could have complex expressions or high dimensions (Boser, 1992).

The basis of the kernel method is the Mercer condition, which gives the necessary and sufficient conditions for any function to be a kernel function.

The Mercer condition is described as follows (Mercer, 1909): For any given continuous symmetric function $k(\boldsymbol{x}, \boldsymbol{y})$, which is an inner product of a feature space, the necessary and sufficient condition is that for any non-constant zero function $\Phi(\boldsymbol{x})$ where $\int\Phi(\boldsymbol{x})\mathrm{d}\boldsymbol{x}<\infty$,

$$\int k(\boldsymbol{x},\boldsymbol{y})\Phi(\boldsymbol{x})\Phi(\boldsymbol{y})\mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{y}>0. \tag{2}$$

The kernel learning method transforms the low-dimensional linear inseparable pattern classification problem into a high-dimensional linear separable problem, and at the same time provides an effective method of constructing the kernel function. In this way, the traditional linear classification method can be used to achieve non-linear data classification in the high-dimensional feature space of data.

Some commonly used kernel functions include the linear kernel $k(\boldsymbol{x},\ \boldsymbol{y})=(\boldsymbol{x}\cdot\boldsymbol{y})$, polynomial kernel $k(\boldsymbol{x},\boldsymbol{y})=(1+\boldsymbol{x}\cdot\boldsymbol{y})^d$, Gaussian radial basis function (RBF) kernel $k(\boldsymbol{x},\ \boldsymbol{y})=\exp(-\gamma\|\boldsymbol{x}-\boldsymbol{y}\|^2)$, and sigmoid kernel $k(\boldsymbol{x},\ \boldsymbol{y})=\tanh[-v(\boldsymbol{x}\cdot\boldsymbol{y})+c]$. The performance of these kernels varies on different datasets. However, in many works on kernel learning (Cherkassky, 1997), it has been indicated that the Gaussian kernel can be used for general purpose classification and regression tasks, because it will output moderate results for most testing datasets. Therefore, in this study we use the Gaussian RBF kernel.

2.2 SIB feature of the FH signal

Compared with the traditional first- and second-order spectra, a high-order spectrum can extract more significant features of non-stationary, non-Gaussian, and non-linear signals. In this study, we extract the square integral bi-spectral features to characterize the fingerprints of the FH transmitters in the feature space.

If x is a continuous random variable with a probability density of $f(x)$, then the characteristic function of x is

$$\Phi(\omega)=E\left[\mathrm{e}^{\mathrm{j}\omega x}\right]=\int_{-\infty}^{\infty}f(x)\mathrm{e}^{\mathrm{j}\omega x}\mathrm{d}x. \tag{3}$$

Taking the logarithm of Eq. (3), the cumulant generation function $\Psi(\omega)$ of the random variable x is as follows:

$$\Psi(\omega)=\ln\left[\Phi(\omega)\right]. \tag{4}$$

Then the k^{th}-order cumulant c_k of the random variable x is defined as the value of the k^{th}-order derivative of its cumulant generation function $\Psi(\omega)$ at the origin:

$$c_k=\left.\frac{\mathrm{d}^k\Psi(\omega)}{\mathrm{d}\omega^k}\right|_{\omega=0}. \tag{5}$$

Similarly, the characteristic function of the k-dimensional random variable $\boldsymbol{x}=[x_1, x_2, \ldots, x_k]^{\mathrm{T}}$ is

$$\Phi(\boldsymbol{\omega})=E\left[\mathrm{e}^{\mathrm{j}\boldsymbol{\omega}^{\mathrm{T}}\boldsymbol{x}}\right]=E\left[\mathrm{e}^{\mathrm{j}(\omega_1x_1+\cdots+\omega_kx_k)}\right], \tag{6}$$

where $\boldsymbol{\omega}=[\omega_1, \omega_2, \ldots, \omega_k]^{\mathrm{T}}$. Correspondingly, the cumulant generation function of $\boldsymbol{x}$ is

$$\Psi(\boldsymbol{\omega})=\ln\left[\Phi(\boldsymbol{\omega})\right]=\ln\left[\Phi(\omega_1,\cdots,\omega_k)\right]=\ln E\left[\mathrm{e}^{\mathrm{j}\boldsymbol{\omega}^{\mathrm{T}}\boldsymbol{x}}\right]. \tag{7}$$

Then, the k^{th}-order cumulant of $\boldsymbol{x}$ is

$$\begin{aligned} c_{kx} &= (-\mathrm{j})^k \left.\frac{\mathrm{d}^k\Psi(\omega_1,\omega_2,\cdots,\omega_k)}{\mathrm{d}\omega^k}\right|_{\omega_1=\omega_2=\cdots=\omega_k=0} \\ &= (-\mathrm{j})^k\Psi^k(\omega). \end{aligned} \tag{8}$$

Given that the random variable x obeys the Gaussian distribution with μ mean and σ^2 variance, the characteristic function of x is

$$\Phi(\omega)=\int_{-\infty}^{\infty} f(x)\mathrm{e}^{\mathrm{j}\omega x}\mathrm{d}x=\exp\left(\mu\omega+\frac{1}{2}\sigma^2\omega^2\right). \tag{9}$$

Therefore, its cumulant generation function is

$$\Psi(\omega)=\ln\left[\Phi(\omega)\right]=\mu\omega+\frac{1}{2}\sigma^2\omega^2. \tag{10}$$

Developing Eq. (10) by Taylor series, we have

$$\Psi(\omega)=c_1\omega+c_2\frac{\omega^2}{2!}+\cdots+c_k\frac{\omega^k}{k!}+\cdots, \tag{11}$$

where $c_1=\mu$, $c_2=\sigma^2$, $c_k=0$, $k>3$. From Eq. (11) we can see that the high-order cumulant of the Gaussian process is always equal to zero; that is, the high-order cumulant is "insensitive" to the Gaussian signal. Thus, if we have the k^{th}-order cumulant, the bi-spectra of noise suppression data $\boldsymbol{x}$ can be defined as

$$B_x\left(\omega_1,\omega_2\right)=\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} c_{3x}\left(\tau_1,\tau_2\right)\mathrm{e}^{-\mathrm{j}(\omega_1\tau_1+\omega_2\tau_2)}\mathrm{d}\tau_1\mathrm{d}\tau_2, \tag{12}$$

where $c_{3x}(\tau_1, \tau_2)$ is the third-order cumulant of $\boldsymbol{x}$. After obtaining the bi-spectrum, we use the square integral bi-spectra analysis method to process the bi-spectral estimation result. As shown in Fig. 2, the integral path is a square centered at the origin, and each point represents a bi-spectral estimate. Based on the bi-spectrum, a number of other integral bi-spectra have been proposed, i.e., radial integral bi-spectra (RIB) (Chandran and Elgar, 1993), axial integral bi-spectra (AIB) (Tugnait, 1994), and circular integral bi-spectra (CIB) (Liao and Bao, 1998).

This integral path does not miss out or reuse any bi-spectrum value, which ensures the integrity of the target information. Furthermore, this calculation transforms the results from two dimensions into one, reducing the computational complexity (Wang et al., 2013; Tang and Lei, 2017). Finally, datasets of fingerprint characteristics are established according to the square integral bi-spectral features $B_x(\omega_1, \omega_2)$.

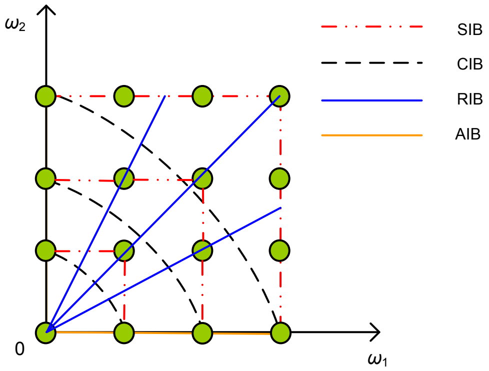

Fig. 2 Integral paths of integral bi-spectra

3 Proposed method

In this section, the proposed kernel joint representation method is elaborated. From this a kernel joint representation classifier is derived. Specifically, first the problem formulation is presented in a joint framework, in which the kernel trick is directly used to generalize the linear algorithm to its non-linear counterpart. After that, the collaborative relationship among the given samples is considered in this framework to obtain a robust recognition. Finally, a unified expression of the induced optimization problem is formulated.

3.1 Problem formulation

Suppose that there are C classes of subjects, and let $\boldsymbol{X}=[\boldsymbol{X}_1,\boldsymbol{X}_2,\cdots,\boldsymbol{X}_C]\in\mathbb{R}^{d\times n}$ denote a set of n training data with multiple samples per class, where $\boldsymbol{X}_i\in\mathbb{R}^{d\times n_i}$, $\sum_i^C n_i=n$ is the dataset of the i^{th} class, and d is the feature dimension of data points. Then the classification function of a new test data $\boldsymbol{y}\in\mathbb{R}^d$ can be formulated as follows:

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \|\boldsymbol{\theta}\|_1 \tag{13}$$

$$\text{s.t.} \quad \boldsymbol{y} = \boldsymbol{X}\boldsymbol{\theta}, \tag{14}$$

$$\text{identity}(\boldsymbol{y}) = \arg\min_i \| \boldsymbol{y} - \boldsymbol{X}_i \hat{\boldsymbol{\theta}}_i \|_2, \tag{15}$$

where $\hat{\boldsymbol{\theta}}$ is the coding vector of $\boldsymbol{y}$ over training data $\boldsymbol{X}$ via L_1-minimization, $\hat{\boldsymbol{\theta}}_i$ is the coding coefficient vector associated with class i, and the classification result can be obtained by Eq. (15). Note that by directly using sparse coding the problem of classification and recognition can be represented compactly. The experimental results reported in Wright et al. (2009) showed that this method achieves an amazing performance. However, the above formulation of the classification function has the following two issues: (1) As mentioned before, the performance of the FH transmitter fingerprints is generally of an irregular non-stationary, non-linear, and non-Gaussian nature, and thus it is difficult to guarantee the effectiveness of SRC for the recognition of the FH transmitter fingerprint feature in the original signal space; (2) As argued in Zhang et al. (2011), since the importance of sparsity is much emphasized in SRC, the effect of collaboration among the given samples is ignored.

3.2 Proposed method

To handle the first issue, a non-linear mapping function $\Phi(\cdot): \mathbb{R}^d \to \mathbb{R}^D (d << D)$ is introduced to map samples to their higher-dimensional feature space, i.e., $\boldsymbol{X} \to \Phi(\boldsymbol{X}) = [\Phi(\boldsymbol{X}_1), \Phi(\boldsymbol{X}_2), \ldots, \Phi(\boldsymbol{X}_C)]$, $\boldsymbol{y} \to \Phi(\boldsymbol{y})$. Then the objective function can be written as

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \|\Phi(\boldsymbol{y}) - \Phi(\boldsymbol{X})\boldsymbol{\theta}\|_2^2 + \lambda \|\boldsymbol{\theta}\|_2^2, \tag{16}$$

where $\Phi(\cdot)$ can be accessed by kernel function $k(\boldsymbol{x}, \boldsymbol{y}) = <\Phi(\boldsymbol{x}), \Phi(\boldsymbol{y})> = \Phi^{\mathrm{T}}(\boldsymbol{x})\Phi(\boldsymbol{y})$, and $k(\boldsymbol{x}, \boldsymbol{y}) = \exp(-\gamma\|\boldsymbol{x}-\boldsymbol{y}\|^2)$ is the Gaussian kernel. After such a non-linear transformation, the test data in the high-dimensional feature space can be better grouped, and thus can be easily recognized. The kernel function has been successfully used in several classification algorithms, such as SVM (Cherkassky and Mulier, 1998), KPCA (Wang et al, 2013), and KLDA (Yang, 2002).

One fact in signal data is that test data of different classes share similarities. Some data from class j may be very helpful in respect of the testing data with label i. That is, when some testing data lack samples, it can be solved by taking the training data from all the other classes as the possible samples of each class. That is, it codes the testing data $\boldsymbol{y} \in \mathbb{R}^d$ collaboratively over the dictionary of all training data $\boldsymbol{X} = [\boldsymbol{X}_1, \boldsymbol{X}_2, \cdots, \boldsymbol{X}_C] \in \mathbb{R}^{d \times n}$. This similarity can be called a collaborative relationship. To overcome the second weakness and make sufficient use of the collaborative relationship among the given test/unlabeled data, the coding coefficient vector is formulated into a recent manifold framework (Nie et al., 2010; Yang Y et al., 2012; Song et al., 2017). Specifically, the proposed method represents all the test/unlabeled data $\boldsymbol{Y} = [\boldsymbol{y}_1, \boldsymbol{y}_2, \cdots, \boldsymbol{y}_m] \in \mathbb{R}^{d \times m}$ simultaneously over the training data and then introduces a coding prediction matrix $\boldsymbol{\Theta} = [\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \cdots, \boldsymbol{\theta}_m]^{\mathrm{T}} \in \mathbb{R}^{m \times n}$ to satisfy the data collaboration. That is, $\boldsymbol{\Theta}$ should be consistent with the collaborative relationship over all data. The final generalized cost function can be formulated as

$$\min \frac{1}{2} \sum_{i,j} S_{i,j} \|\boldsymbol{\theta}_i - \boldsymbol{\theta}_j\|_2^2, \tag{17}$$

where $\boldsymbol{S} = \{S_{i,j}\} \in \mathbb{R}^{m \times m}$ is the weight matrix and its element $W_{i,j} = \exp(-\|\boldsymbol{y}_i - \boldsymbol{y}_j\|^2)$ reflects the collaborative relationship between two data $\boldsymbol{y}_i$ and $\boldsymbol{y}_j$. When the data $\boldsymbol{y}_i$ and $\boldsymbol{y}_j$ are similar, $W_{i,j}$ is large and the distance between $\boldsymbol{\theta}_i$ and $\boldsymbol{\theta}_j$ should be very small to minimize (17). Denote $\boldsymbol{L}$ as a Laplacian matrix computed by $\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{S}$, and $\boldsymbol{D}$ as a diagonal matrix whose diagonal elements are $D_{ii} = \sum_{j=1}^{m} S_{i,j}$, the solution to problem (17) can be given by optimizing

$$\min_{\boldsymbol{\Theta}} \operatorname{tr}(\boldsymbol{\Theta}\boldsymbol{L}\boldsymbol{\Theta}^{\mathrm{T}}). \tag{18}$$

By leveraging the kernel trick, feature representation, and classifier learning, the proposed method integrates Eq. (16) and expression (17) into a joint representation framework:

$$\begin{aligned} \min_{\boldsymbol{\Theta}} \sum_{i=1}^{m} (\|\Phi(\boldsymbol{y}_i) - \Phi(\boldsymbol{X})\boldsymbol{\theta}_i\|_2^2 + \lambda \sum_{i=1}^{m} \|\boldsymbol{\theta}_i\|_2^2 \\ + \beta \frac{1}{2} \sum_{i,j} S_{i,j} \|\boldsymbol{\theta}_i - \boldsymbol{\theta}_j\|_2^2), \end{aligned} \tag{19}$$

where $\beta > 0$ is a regularization parameter and $\lambda > 0$ is a weight parameter. After a simple transformation, the definition of $\|\boldsymbol{\Theta}\|_2^2$ can be rewritten as

$$\sum_{i=1}^{m}\|\boldsymbol{\theta}_i\|_2^2=\mathrm{tr}(\boldsymbol{\Theta}\boldsymbol{\Theta}^{\mathrm{T}}), \tag{20}$$

where tr(·) denotes the trace operator, and the objective function of problem (19) can be reformulated to

$$\begin{aligned}J(\boldsymbol{\Theta})=\min_{\boldsymbol{\Theta}}\mathrm{tr}((\varPhi(\boldsymbol{Y})-\varPhi(\boldsymbol{X})\boldsymbol{\Theta})^{\mathrm{T}}(\varPhi(\boldsymbol{Y})\\ -\varPhi(\boldsymbol{X})\boldsymbol{\Theta}))+\lambda\mathrm{tr}(\boldsymbol{\Theta}^{\mathrm{T}}\boldsymbol{\Theta})+\beta\mathrm{tr}(\boldsymbol{\Theta}\boldsymbol{L}\boldsymbol{\Theta}^{\mathrm{T}}).\end{aligned} \tag{21}$$

By taking the derivative of Eq. (21) with respect to $\boldsymbol{\Theta}$ and setting it to zero, we have

$$\begin{aligned}&\frac{\partial J(\boldsymbol{\Theta})}{\partial\boldsymbol{\Theta}}=-2\varPhi^{\mathrm{T}}(\boldsymbol{X})\varPhi(\boldsymbol{Y})+2\varPhi^{\mathrm{T}}(\boldsymbol{X})\varPhi(\boldsymbol{X})\\&\qquad+2\lambda\boldsymbol{\Theta}+2\beta\boldsymbol{\Theta}\boldsymbol{L}=\boldsymbol{0}\\&\Rightarrow(\varPhi^{\mathrm{T}}(\boldsymbol{X})\varPhi(\boldsymbol{X})+\lambda\boldsymbol{I}_m)\boldsymbol{\Theta}+\beta\boldsymbol{\Theta}\boldsymbol{L}=\varPhi^{\mathrm{T}}(\boldsymbol{X})\varPhi(\boldsymbol{Y}).\end{aligned} \tag{22}$$

In this study, we denote $\boldsymbol{Q}=\varPhi^{\mathrm{T}}(\boldsymbol{X})\varPhi(\boldsymbol{X})+\lambda\boldsymbol{I}_m$, $\boldsymbol{P}=\beta\boldsymbol{L}$, and $\boldsymbol{W}=\varPhi^{\mathrm{T}}(\boldsymbol{X})\varPhi(\boldsymbol{Y})$, where $\boldsymbol{I}_m$ is the $m\times m$ identity matrix. Clearly, $\boldsymbol{Q}$ is independent of $\boldsymbol{Y}$ such that it can be pre-calculated. This reduces the computational complexity. Then we have the matrix equation

$$\boldsymbol{Q}\boldsymbol{\Theta}+\boldsymbol{\Theta}\boldsymbol{P}=\boldsymbol{W} \tag{23}$$

for $\boldsymbol{\Theta}$, where $\boldsymbol{\Theta}$ and $\boldsymbol{W}$ are $m\times n$ real matrices, $\boldsymbol{Q}$ is an $m\times m$ real matrix, and $\boldsymbol{P}$ is an $n\times n$ real matrix. According to Jameson (1968), using the notation $\boldsymbol{A}\times\boldsymbol{B}$ to denote the Kronecker product ($A_{i,j}\boldsymbol{B}$) (Bellman, 1997), in which each element of $\boldsymbol{A}$ is multiplied by $\boldsymbol{B}$, the equation written out in full for the mn unknowns $\boldsymbol{\theta}_{1,1}$, $\boldsymbol{\theta}_{2,1}$, …, $\boldsymbol{\theta}_{1,2}$ … in terms of $w_{1,1}$, $w_{2,1}$, …, $w_{1,2}$ … becomes

$$[\boldsymbol{I}_m\boldsymbol{Q}+(\boldsymbol{P}^{\mathrm{T}}+\boldsymbol{I}_n)]\boldsymbol{\theta}=\boldsymbol{w}. \tag{24}$$

If $\boldsymbol{u}$ is a characteristic vector of $\boldsymbol{Q}$ with characteristic value μ, and $\boldsymbol{v}$ is a characteristic vector of $\boldsymbol{P}^{\mathrm{T}}$ with characteristic value v, then

$$\boldsymbol{Q}\boldsymbol{u}\boldsymbol{v}^{\mathrm{T}}+\boldsymbol{u}\boldsymbol{v}^{\mathrm{T}}\boldsymbol{P}=(\mu+v)\boldsymbol{u}\boldsymbol{v}^{\mathrm{T}}. \tag{25}$$

Thus, $\mu+v$ is a characteristic value of system (24). This can therefore be solved if and only if

$$\mu_i+v_j\neq 0, \tag{26}$$

for all i, j.

When $\boldsymbol{Q}$ and $\boldsymbol{P}$ can both be reduced to the diagonal form by similarity transformations:

$$\boldsymbol{U}^{-1}\boldsymbol{Q}\boldsymbol{U}=\begin{bmatrix}\mu_1&&&\\&\mu_2&&\\&&\ddots&\\&&&\mu_m\end{bmatrix} \tag{27}$$

and

$$\boldsymbol{V}^{-1}\boldsymbol{P}\boldsymbol{V}=\begin{bmatrix}v_1&&&\\&v_2&&\\&&\ddots&\\&&&v_n\end{bmatrix}, \tag{28}$$

the solution to Eq. (23) is easily obtained as

$$\boldsymbol{\Theta}=\boldsymbol{U}\tilde{\boldsymbol{\Theta}}\boldsymbol{V}^{-1}, \tag{29}$$

where

$$\tilde{\theta}_{i,j}=\frac{\tilde{w}_{i,j}}{\mu_i+v_j},\ \tilde{\boldsymbol{W}}=\boldsymbol{U}^{-1}\boldsymbol{W}\boldsymbol{V}. \tag{30}$$

For each testing sample $\boldsymbol{y}_j\in\mathbb{R}^d$ $(j=1,2,\cdots,m)$, the final recognition result can be formulated as

$$\mathrm{class}(\boldsymbol{y}_j)=\arg\min_i\|(\varPhi(\boldsymbol{Y})-\varPhi(\boldsymbol{X})\hat{\boldsymbol{\Theta}}_i)_j\|_2^2, \tag{31}$$

where $\hat{\boldsymbol{\Theta}}_i$ denotes the coded matrix associated with class i, that is

$$\hat{\boldsymbol{\Theta}}_i=\begin{bmatrix}0\\\vdots\\\hat{\Theta}_i\\\vdots\\0\end{bmatrix}, \tag{32}$$

and $(\varPhi(\boldsymbol{Y})-\varPhi(\boldsymbol{X})\hat{\boldsymbol{\Theta}}_i)_j$ is the j^{th} column vector of $\varPhi(\boldsymbol{Y})-\varPhi(\boldsymbol{X})\hat{\boldsymbol{\Theta}}_i$. The proposed method picks out the result outputting the least error.

The proposed kernel joint representation method is summarized in Algorithm 1.

Algorithm 1 Kernel joint representation classifier

```
Input: training data $\boldsymbol{X}=[\boldsymbol{X}_1,\boldsymbol{X}_2,\cdots,\boldsymbol{X}_C]\in\mathbb{R}^{d\times n}$; testing data $\boldsymbol{Y}=[\boldsymbol{y}_1,\boldsymbol{y}_2,\cdots,\boldsymbol{y}_m]\in\mathbb{R}^{d\times m}$; parameters $\lambda$ and $\beta$.
Output: recognition result of $\boldsymbol{Y}=[\boldsymbol{y}_1,\boldsymbol{y}_2,\cdots,\boldsymbol{y}_m]\in\mathbb{R}^{d\times m}$ as Recognition ($\boldsymbol{y}_j$)= $\arg\min_i \|(\varPhi(\boldsymbol{Y})-\varPhi(\boldsymbol{X})\hat{\boldsymbol{\varTheta}}_i)_j\|_2^2$.
1: Calculate the collaborative relationship weight matrix $\boldsymbol{S}$
2: Calculate the diagonal matrix $\boldsymbol{D}$ and Laplacian matrix $\boldsymbol{L}$
3: Calculate $\boldsymbol{Q}=\varPhi^{\mathrm{T}}(\boldsymbol{X})\varPhi(\boldsymbol{X})+\lambda\boldsymbol{I}$, $\boldsymbol{P}=\beta\boldsymbol{L}$, and $\boldsymbol{W}=\varPhi^{\mathrm{T}}(\boldsymbol{X})\varPhi(\boldsymbol{Y})$
4: Calculate $\boldsymbol{U}$ and $\boldsymbol{V}$ by SVD of matrices $\boldsymbol{Q}$ and $\boldsymbol{P}$
5: Calculate $\tilde{\boldsymbol{\varTheta}}$ and $\tilde{\boldsymbol{W}}$ by Eq. (30)
6: Calculate the coding result of $\hat{\boldsymbol{\varTheta}}$ by Eq. (29)
```

3.3 Analysis of implementation

The original kernel joint representation classifier takes the collaborative relationship among the given samples into account, in which samples belonging to the same class can be better grouped. However, this method could face a computation problem when obtaining the collaborative relationship weight matrix $\boldsymbol{S}$ if there are a lot of samples in the dataset. The KJR classifier might not be practical for applications with many samples. To deal with this computational complexity problem, the proposed method applies the recently proposed anchoring graph (Liu et al., 2010; Song et al., 2017) to approximate the weight matrix $\boldsymbol{S}$ and then obtain $\boldsymbol{L}$.

Also, it is worth pointing out that our method could be easily extended to noised signal recognition. Rewrite Eq. (14) as

$$\text{s.t.}\quad \boldsymbol{Y}=\boldsymbol{X\varTheta}+\boldsymbol{E}, \tag{33}$$

where $\boldsymbol{E}\in\mathbb{R}^{d\times m}$ is a noise matrix. Substituting $\breve{\boldsymbol{X}}=[\boldsymbol{X},\boldsymbol{I}]\in\mathbb{R}^{d\times(m+d)}$ and $\breve{\boldsymbol{\varTheta}}=\begin{bmatrix}\boldsymbol{\varTheta}\\ \boldsymbol{E}\end{bmatrix}\in\mathbb{R}^{(m+d)\times m}$ for $\boldsymbol{X}$ and $\boldsymbol{\varTheta}$, respectively, the flexible kernel joint representation model can be formulated as

$$\begin{aligned}J(\breve{\boldsymbol{\varTheta}})=\min_{\breve{\boldsymbol{\varTheta}}}&\operatorname{tr}((\varPhi(\boldsymbol{Y})-\varPhi(\breve{\boldsymbol{X}})\breve{\boldsymbol{\varTheta}})^{\mathrm{T}}(\varPhi(\boldsymbol{Y})-\varPhi(\breve{\boldsymbol{X}})\breve{\boldsymbol{\varTheta}}))\\&+\lambda\operatorname{tr}(\breve{\boldsymbol{\varTheta}}^{T}\breve{\boldsymbol{\varTheta}})+\beta\operatorname{tr}(\breve{\boldsymbol{\varTheta}}\boldsymbol{L}\breve{\boldsymbol{\varTheta}}^{\mathrm{T}}).\end{aligned}\tag{34}$$

Once a solution $\breve{\boldsymbol{\varTheta}}^*=\begin{bmatrix}\boldsymbol{\varTheta}^*\\ \boldsymbol{E}^*\end{bmatrix}\in\mathbb{R}^{(m+d)\times m}$ to Eq. (34) is computed, setting $\boldsymbol{Y}^*=\boldsymbol{Y}^*-\boldsymbol{E}^*$ recovers clear data from a corrupted subject. To identify the testing data $\boldsymbol{y}_j$, we slightly modify the recognition result of $\boldsymbol{y}_j$ as

$$\text{Recogntion}(\boldsymbol{y}_j)=\arg\min_i \|(\varPhi(\boldsymbol{Y}-\boldsymbol{E}^*)-\varPhi(\boldsymbol{X})\hat{\boldsymbol{\varTheta}}_i)_j\|_2^2 .$$

The experimental results of the proposed method against the noise are shown below. This study will not concentrate on this subject.

4 Experimental results

To assess the effectiveness of the proposed KJRC, we have performed extensive experiments on 100 records of a real-world signal for each of five different FH transmitters and compared several state-of-the-art approaches: k-nearest neighbor classifier, support vector machine (SVM), sparse representation classifier (SRC), and collaborative representation classifier (CRC). A summary listing the particulars of these five transmitters can be found in Table 1. For comparison, we use the publicly available codes under the best settings. The number of neighbors k in k-NN is chosen from {2, 3, …, min(N_c)−1}, where N_c is number of data from the c^{th} FH transmitter. The kernel function of SVM is a Gaussian kernel with the kernel width chosen from $\{2^{-5}, 2^{-4}, \ldots, 2^{4}, 2^{5}\}$ by cross-validation. SRC uses constrained L_1-minimization to compute the sparse coefficients with the regularization λ=1000 and error tolerance ε=0.05. CRC relaxes the sparsity constraint by L_2-minimization, and the regularization λ is set to 0.001.

All experiments are implemented in Matlab 2014a, and run on a PC with Intel Core i7, 2.93 GHz CPU, and 4 GB RAM.

Table 1 Parameter details of the five different FH transmitters

Group	Transmitter	Type	Frequency (MHz)	Hop speed (hop/s)
1	Harris	RF5800H-MP	1.6–60	20.0
2	Q-MAC	HF-90	2–30	5.0
3	Grintek	TR2400	1.6–30	10.0
4	Thales	SystEme3000	1.5–30	20.0
5	Rohde & Schwarz	MR3000H	1.5–108	8.5

4.1 Choice of the kernel function

The kernel function maps the original low-dimensional non-linear sample to the high-dimensional feature space through kernel function transformation, constructing an optimal classification plane in the high-dimensional space and transforming the classification into a linear separable problem. Any function that satisfies the Mercer condition can be used as a kernel function.

Determining the proper kernel function can realize linear classification in the high-dimensional space after transformation without increasing the computational complexity. Different kernel functions generate different optimal classification planes. Therefore, the choice of the kernel function is one of the important factors that determine the recognition performance. Commonly used kernel functions include:

1. Linear kernel $k(\boldsymbol{x}, \boldsymbol{y})=\boldsymbol{x}\cdot\boldsymbol{y}$

The linear kernel function is suitable for classifying linearly separable samples in a low-dimensional data space. However, most samples are linearly inseparable from the low-dimensional space.

2. Polynomial kernel $k(\boldsymbol{x}, \boldsymbol{y})=(1+\boldsymbol{x}\cdot\boldsymbol{y})^d$

The parameter d represents the dimension of the kernel function. When the feature space dimension is high, the polynomial order is also relatively high, which is very computationally demanding. Also, the polynomial kernel function has poor local performance.

3. Gaussian radial basis function (RBF) kernel $k(\boldsymbol{x}, \boldsymbol{y})=\exp(-\gamma\|\boldsymbol{x}-\boldsymbol{y}\|^2)$

The local performance of the RBF kernel function is good, and it has a good classification effect on sample data that are closer together.

4. Sigmoid kernel $k(\boldsymbol{x}, \boldsymbol{y})=\tanh[-v(\boldsymbol{x}\cdot\boldsymbol{y})+c]$

The implementation of the sigmoid kernel function needs to meet certain conditions. The sigmoid kernel function is equivalent to two neural networks and has good global convergence.

The performance of these kernels varies on different datasets. However, in many works on kernel learning, it has been indicated that the Gaussian RBF kernel can be used for general-purpose classification and regression tasks, because it is not subject to sample size and feature dimensions and can output moderate results for most testing datasets. Therefore, in this study we use the Gaussian RBF kernel.

To evaluate the effect of different kernel functions on recognition results, we have compared the proposed method with different kernel functions. The training data are the same as those in Section 4.1. The experimental results are shown in Table 2. At the same time, we add a comparison of the performance before and after kernel tricks in our method. The experimental results are shown in Table 3.

Table 2 The recognition rate of the proposed method with different kernel functions

Number of training samples	Recognition rate (%)			
	Linear	Polynomial	Gaussian RBF	Sigmoid
50	57.5	55.4	72.7	71.9
100	61.0	62.9	77.1	77.5
150	68.3	66.5	83.8	80.3
200	71.9	73.1	86.5	86.7
250	78.8	79.6	90.2	89.9
300	85.6	86.8	92.3	92.2
350	89.4	90.5	93.8	93.5
400	91.3	92.2	96.4	95.8
450	92.9	93.1	97.8	97.7

Table 3 The recognition rate of our method before and after the kernel tricks

Number of training samples	Recognition rate (%)	
	Before	After
50	48.9	72.7
100	55.7	77.1
150	63.8	83.8
200	76.4	86.5
250	85.1	90.2
300	88.7	92.3
350	91.6	93.8
400	93.7	96.4
450	95.9	97.8

From Table 2, linear kernel and polynomial kernel have poor results. This is mainly due to the high non-linearity of the sample data and the high dimensionality of the features. The sigmoid kernel and Gaussian RBF kernel have similar recognition results, but the sigmoid kernel has a strict requirement in parameter validation. Compared with the sigmoid kernel, the Gaussian RBF kernel function is more suitable for non-linearity of sample mapping in the high-dimensional space, the parameters to be validated are fewer, and the complexity is low. Therefore, we choose Gaussian RBF as the classification kernel function.

From Table 3, it is obvious that whether or not one conducts the kernel tricks has a great influence on the final recognition result, especially when the training data are relatively small. Therefore, a Gaussian kernel is used in our study to improve the recognition rate.

4.2 Recognition results of some real-world FH transmitter signals

Recognition efficiency is important in a real-world FH transmitter classification application. In this subsection, five kinds of FH transmitter signals collected by the external field are used as identification targets, and each transmitter has a total of 100 signals, of which 5%, 10%, 20%, 30%, 40%, 50%, 60%, 70%, 80%, and 90% signals are used as training data, and the remaining signals are used as test data to verify the influence of the training data number on the individual identification of FH transmitters. The results are illustrated in Fig. 3. Note that all signals have been processed by an SIB feature extraction algorithm as presented in Section 2.2. Namely, the final training data and testing data are SIB features of FH transmitters. From the results, we can learn that the performance is improved for all five classifiers as the amount of training data increases, and when the amount of training data becomes large enough, the recognition rates of different classifier vary slightly. Overall, the proposed method always shows superior recognition ability to other classifiers.

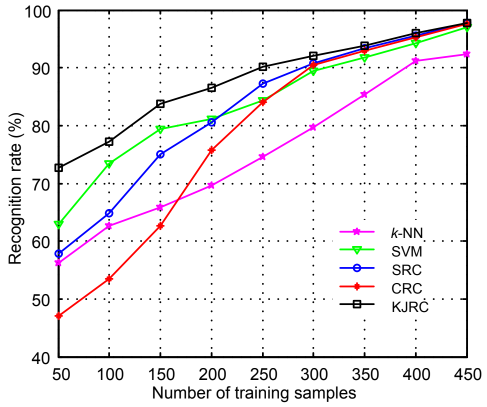

Fig. 3 The recognition rate with different methods

To analyze the impact of SIB feature dimension on recognition, we have extracted the SIB features of FH transmitter signals with different dimensions, i.e., 64, 128, 192, etc. Fig. 4 shows the identification performance of the proposed KJRC method for differing amounts of training data with different SIB feature dimensions. The experimental results verify that high dimensionality of the SIB feature is beneficial for the proposed method, and when the dimension of the SIB feature becomes large enough, the identification performance of the proposed method varies hardly at all. For the five methods, with different training data and SIB feature dimensions, the identification results are shown in Figs. 5a and 5b, which show the comparison results with training data amount of 30% and 60% of the whole data, respectively. In the two experiments, the proposed KJRC always shows the highest recognition rate. As illustrated in Algorithm 1, the proposed method requires mainly a kernel function to generalize a linear algorithm to its non-linear counterpart in which the accuracy of recognition can be ensured. The proposed method represents all the test samples simultaneously over the training data set. Also, the correlation of multiple samples and a single representation have been considered, and therefore the experimental results are more robust.

4.3 Computational time of different methods

To evaluate the proposed algorithm comprehensively, we also test the efficiency of the *k*-NN, SVM, SRC, CRC methods and the proposed method. The training dataset and test samples are the same as those in Section 4.1. Fig. 6 shows the computational time for different methods. We can see that the computational time of SVM is much longer than that of other methods. This is mainly because SVM involves the computation of the inverse of some large matrices. This is computationally expensive. SRC is relatively slow, and CRC and the proposed method take negligible computational time compared with SVM and SRC. The SRC method addresses the sparse representation problem into L_1-minimization optimization, which is very computationally demanding. However, CRC and our method are based on L_2-minimization, and thus are more computationally efficient. *k*-NN takes the least computational time of all the methods because the recognition results are generated without the learning process, but its recognition rate is also the lowest.

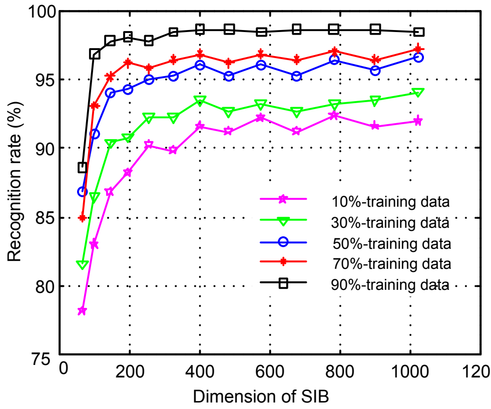

Fig. 4 The recognition rate of KJRC with different dimensions of the SIB feature and training data

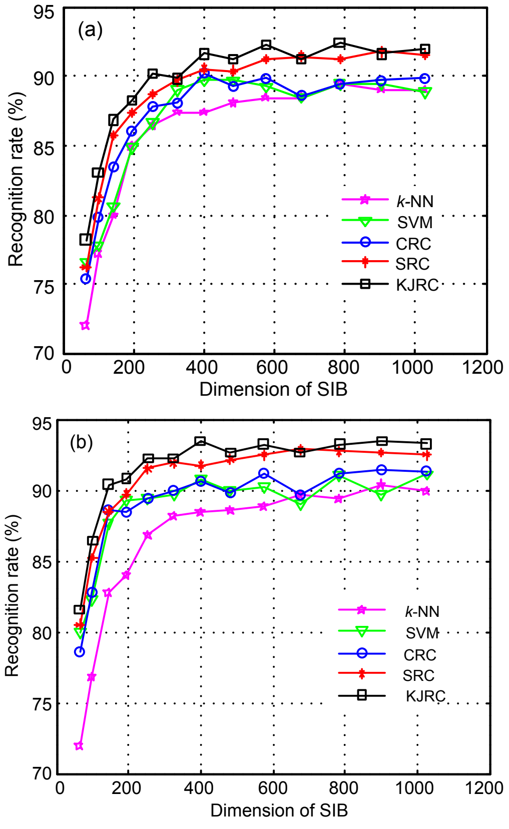

Fig. 5 The recognition rate of different methods with different dimensions of the SIB feature: (a) 30% training data; (b) 60% training data

4.4 Robustness of our method to free parameters

In the above experiments, the fixed parameters in the proposed method are used to perform the recognition. In this test, we investigate the influence of parameters λ and β. The variations of recognition rates are shown in Fig. 7. We can see that the best recognition can be obtained by choosing suitable combinations of λ and β, and there is a wide range to choose these best combinations, showing the robustness of our method to the two parameters. We use the parameter set of λ=2.0 and β=0.1 for all experiments by default.

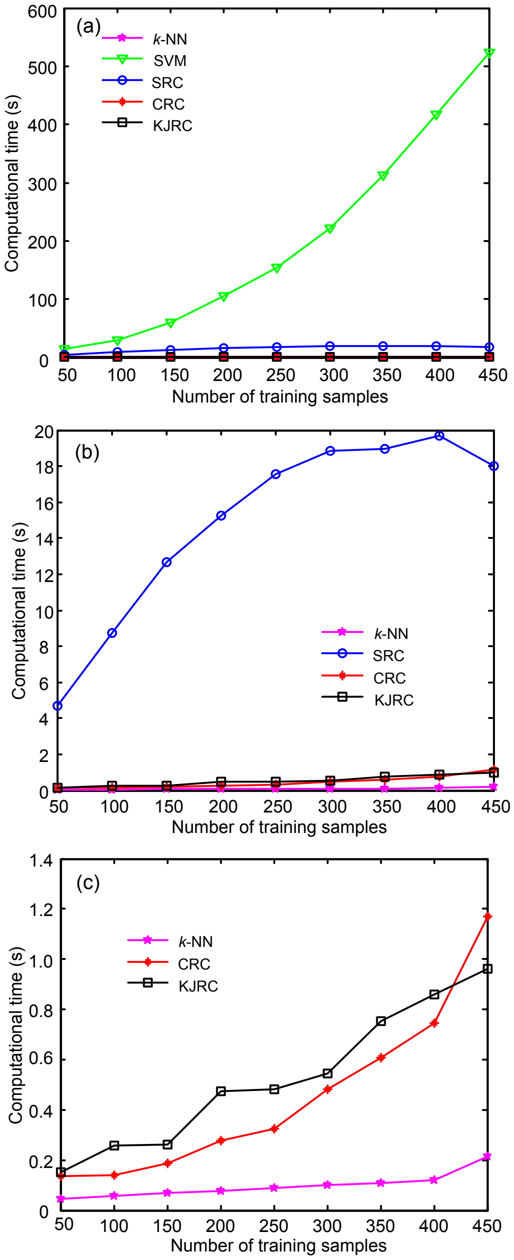

Fig. 6 The computational time of different methods: (a) comparison of five algorithms; (b) partial enlargement of *k*-NN, SRC, CRC, and KJRC; (c) partial enlargement of *k*-NN, CRC, and KJRC

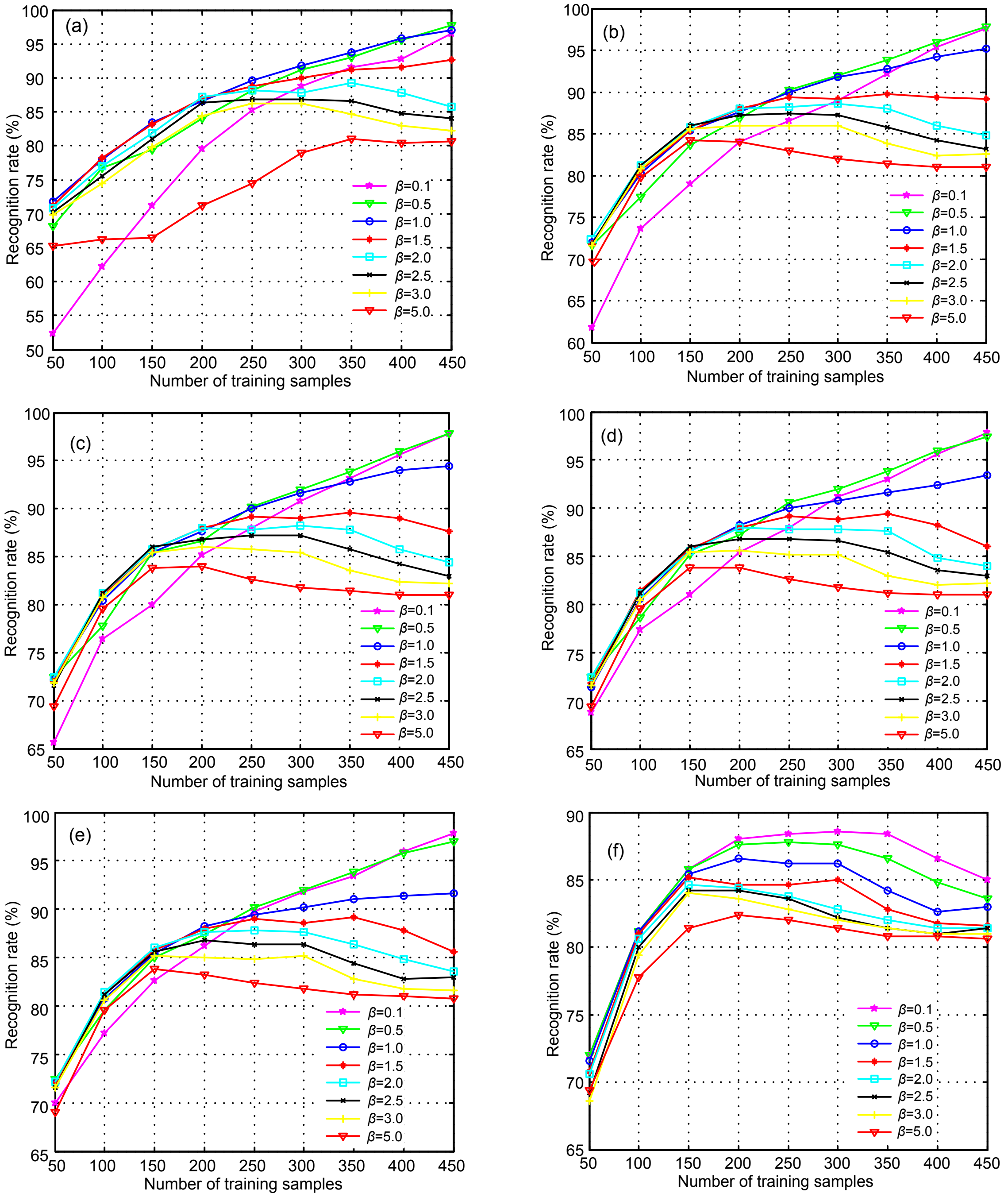

Fig. 7 The recognition rate of different parameters: (a) λ=0.001; (b) λ=0.005; (c) λ=0.01; (d) λ=0.1; (e) λ=1.0; (f) λ=10.0

5 Conclusions

In this paper, a kernel joint representation classifier is proposed for frequency-hopping transmitter fingerprint feature recognition. It integrates kernel projection, feature representation, and classifier learning into a joint framework. First, a kernel function is used to generalize the linear algorithm to its non-linear counterpart, in which features belonging to the same class can be better grouped. Then collaboration among the given samples is considered, under the assumption that the given samples are generally related to each other. Finally, we integrate kernel projection, feature representation, and classifier learning into a joint framework. At the same time a unified expression is developed to solve the optimization problem. Experimental results on five real-world FH transmitters validate the significant performance of the proposed kernel joint representation classifier compared to the state-of-the-art classifiers in terms of accuracy and efficiency.

Compliance with ethics guidelines

Ping SUI, Ying GUO, Kun-feng ZHANG, and Hong-guang LI declare that they have no conflict of interest.

References

Angelosante D, Giannakis GB, Sidiropoulos ND, 2010. Estimating multiple frequency-hopping signal parameters via sparse linear regression. *IEEE Trans Signal Process*, 58(10):5044-5056. https://doi.org/10.1109/tsp.2010.2052614

Bellman R, 1997. Introduction to Matrix Analysis (2nd Ed.). McGraw-Hill, New York. https://doi.org/10.1137/1.9781611971170

Boser BE, Guyon IM, Vapnik VN, 1992. A training algorithm for optimal margin classifiers. Proc 5th Annual Workshop on Computational Learning Theory, p.144-152. https://doi.org/10.1145/130385.130401

Chandran V, Elgar SL, 1993. Pattern recognition using invariants defined from higher order spectra one-dimensional inputs. *IEEE Trans Signal Process*, 41(1):205. https://doi.org/10.1109/tsp.1993.193139

Cherkassky V, 1997. The nature of statistical learning theory. *IEEE Trans Neur Netw*, 8(6):1564. https://doi.org/10.1109/TNN.1997.641482

Cherkassky V, Mulier F, 1998. Learning from Data: Concepts, Theory, and Methods. Wiley, New York, NY, USA.

Cover T, Hart P, 1967. Nearest neighbor pattern classification. *IEEE Trans Inform Theory*, 13(1):21-27. https://doi.org/10.1109/TIT.1967.1053964

Friedl MA, Brodley CE, 1997. Decision tree classification of land cover from remotely sensed data. *Remote Sens Environ*, 61(3):399-409. https://doi.org/10.1016/s0034-4257(97)00049-7

Jameson A, 1968. Solution of the equation $AX + XB = C$ by inversion of an $M \times M$ or $N \times N$ matrix. *SIAM J Appl Math*, 16(5):1020-1023. https://doi.org/10.1137/0116083

Lawrence RL, Wright A, 2001. Rule-based classification systems using classification and regression tree (CART) analysis. *Photogramm Eng Remote Sens*, 67(10):1137-1142.

Liao XJ, Bao Z, 1998. Circularly integrated bispectra: novel shift invariant features for high-resolution radar target recognition. *Electron Lett*, 34(19):1879-1880. https://doi.org/10.1049/el:19981307

Liu SH, Zhang YD, Shan T, et al., 2018. Structure-aware Bayesian compressive sensing for frequency-hopping spectrum estimation with missing observations. *IEEE Trans Signal Process*, 66(8):2153-2166. https://doi.org/10.1109/TSP.2018.2806351

Liu W, He JF, Chang SF, 2010. Large graph construction for scalable semi-supervised learning. Pros 27th Int Conf on Machine Learning, p.679-686.

Mercer J, 1909. Functions of positive and negative type, and their connection with the theory of integral equations. *Phil Trans R Soc A*, 209(441-458):415-446. https://doi.org/10.1098/rsta.1909.0016

Muller KR, Mika S, Ratsch G, et al., 2001. An introduction to kernel-based learning algorithms. *IEEE Trans Neur Netw*, 12(2):181-201. https://doi.org/10.1109/72.914517

Nie FP, Xu D, Tsang IWH, et al., 2010. Flexible manifold embedding: a framework for semi-supervised and unsupervised dimension reduction. *IEEE Trans Image Process*, 19(7):1921-1932. https://doi.org/10.1109/tip.2010.2044958

Quinlan JR, 1993. C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Song TC, Cai JF, Zhang TQ, et al., 2017. Semi-supervised manifold-embedded hashing with joint feature representation and classifier learning. *Patt Recogn*, 68:99-110. https://doi.org/10.1016/j.patcog.2017.03.004

Tadjudin S, Landgrebe DA, 1996. A decision tree classifier design for high-dimensional data with limited training samples. Int Geoscience and Remote Sensing Symp, p.790-792. https://doi.org/10.1109/igarss.1996.516476

Tang Z, Lei YK, 2017. Radio transmitter identification based on collaborative representation. *Wirel Pers Commun*, 96(1):1377-1391. https://doi.org/10.1007/s11277-017-4242-z

Tugnait JK, 1994. Detection of non-Gaussian signals using integrated polyspectrum. *IEEE Trans Signal Process*, 42(11):3137-3149. https://doi.org/10.1109/78.330373

Wang B, Li WF, Poh N, et al., 2013. Kernel collaborative representation-based classifier for face recognition. IEEE Int Conf on Acoustics, Speech and Signal Processing, p.2877-2881. https://doi.org/10.1109/icassp.2013.6638183

Wang LP, Chen SC, 2017. Joint representation classification for collective face recognition. *Patt Recogn*, 63:182-192. https://doi.org/10.1016/j.patcog.2016.10.004

Wright J, Yang AY, Ganesh A, et al., 2009. Robust face recognition via sparse representation. *IEEE Trans Patt Anal Mach Intell*, 31(2):210-227. https://doi.org/10.1109/TPAMI.2008.79

Wu XD, Kumar V, Quinlan JR, et al., 2008. Top 10 algorithms in data mining. *Knowl Inform Syst*, 14(1):1-37. https://doi.org/10.1007/s10115-007-0114-2

Yang M, Zhang L, Zhang D, et al., 2012. Relaxed collaborative representation for pattern classification. IEEE Conf on Computer Vision and Pattern Recognition, p.2224-2231.

https://doi.org/10.1109/cvpr.2012.6247931

Yang MH, 2002. Kernel Eigenfaces vs. Kernel Fisherfaces: face recognition using kernel methods. Pros 5th IEEE Int Conf on Automatic Face Gesture Recognition, p.215-220. https://doi.org/10.1109/afgr.2002.4527207

Yang Y, Wu F, Nie FP, et al., 2012. Web and personal image annotation by mining label correlation with relaxed visual graph embedding. *IEEE Trans Image Process*, 21(3): 1339-1351. https://doi.org/10.1109/tip.2011.2169269

Yoshikawa M, Shindo H, Nishii R, et al., 1995. A fully automated design of binary decision tree for land cover classification. Int Geoscience and Remote Sensing Symp on Quantitative Remote Sensing for Science and Applications, p.1921-1923. https://doi.org/10.1109/igarss.1995.524067

Zhang L, Yang M, Feng XC, 2011. Sparse representation or collaborative representation: which helps face recognition? Int Conf on Computer Vision, p.471-478. https://doi.org/10.1109/iccv.2011.6126277

Zhao LF, Wang L, Bi GA, et al., 2015. Robust frequency-hopping spectrum estimation based on sparse Bayesian method. *IEEE Trans Wirel Commun*, 14(2):781-793. https://doi.org/10.1109/twc.2014.2360191