



Manifold learning for novelty detection and its application in gesture recognition

Yang Luo¹ · Yibiao Yuan² · Wei Zheng³ · Xiaohui Mo³

Received: 21 July 2021 / Accepted: 30 January 2022 / Published online: 14 March 2022
© The Author(s) 2022

Abstract

As a state-of-the-art novelty detection method, Kernel Null Foley–Sammon Transform (KNFST) could identify multiple known classes and detect novelties from an unknown class via a single model. However, KNFST only captures the global information of the training set. The local geometrical structure is neglected. In this paper, a manifold is incorporated into KNFST to solve this issue. First, we use manifold graphs to depict the local structure for within-class scatter and total scatter. Second, the training samples from the same class are mapped into a single point in null space via null projected directions (NPDs). The proposed method can overcome the weakness of KNFST caused by ignoring local geometrical structure in the class. The experimental results on several toy and benchmark datasets show that manifold learning novelty detection (MLND) is superior to KNFST.

Keywords Manifold learning · Novelty detection · Null projected directions · Gesture recognition

Introduction

In many real-world applications, a test sample may be from an unknown class that is not available in the training set. These samples can be regarded as novelties or anomalies for known classes as they are far away from the distributions of known classes [1,2]. The problem is termed as novelty detection or anomaly detection. When the known classes are more than one, it is called multi-class supervised novelty detection [3,4]. Novelty detection is widely used in the community of pattern recognition. For instance, traffic police wants to find

the illegal traffic flow [5], ophthalmologist wants to detect retinal damage [6], cyber security expert needs to monitor cyber-intrusion from massive visiting [7,8], engineer needs to analyze big data in Internet of Things (IoT) [9], abrupt changes in air temperature [10], unknown pixels in hyperspectral image [11], and medical ultrasound image analysis [12,13] to name just a few. During the past several decades, the works about supervised novelty detection mainly focus on the training set that only contains one known class [14,15]. It is called one-class classification [16,17]. The one-class classifier can only tell us whether a test sample is normal or not [18,19]. When the training set contains more than one known class, it requires to treat all known classes as a superclass [20] or learn several one-class classifiers [21]. For the former way, it still requires to learn a multi-class classifier to tell us which class a test sample is from if it is not a novelty.

Kernel null Foley–Sammon Transform (KNFST) [3] can deal with one-class classification as well as multi-class supervised novelty detection. For multi-class supervised novelty detection, KNFST only learns a single model which can tell us a test sample whether is a novelty or which class it comes from if it is a normal sample. KNFST maps the samples from the same class to a single point in reproducing kernel Hilbert space (RKHS) via null projection directions. Let c represent the number of classes. Then, the label of a test sample is determined by the minimum distance to the mapping points. If the distances to mapping points are all

✉ Yang Luo
luoyang@jit.edu.cn

Yibiao Yuan
yuanyb@njmu.edu.cn

Wei Zheng
zhengwei@jit.edu.cn

Xiaohui Mo
mxh@jit.edu.cn

¹ The School of Software Engineering, Jinling Institute of Technology, Nanjing 211169, People's Republic of China

² The Laboratory Center for Basic Medical Sciences, Nanjing Medical University, Nanjing 211169, People's Republic of China

³ The School of Computer Engineering, Jinling Institute of Technology, Nanjing 211169, People's Republic of China

very large, it is a novelty. However, KNFST can only capture the global information and neglects the local geometrical structure. It may fail when the local geometrical structure is complex, which also exists in classification [22] or ordinal regression [23]. In order to address this issue, we propose a manifold learning-novelty detection (MLND) method in which a manifold graph is introduced to regularize within- and between-class scatter matrices, respectively. The manifold graph is used to depict local geometrical structure. The experimental results demonstrate that MLND is superior to KNFST on several toy and benchmark datasets. The main contributions are summarized as following three points.

- First, we introduce a manifold into within-class scatter and total scatter to depict local structure in class for Foley–Sammon transform (or Fisher discriminant analysis).
- Second, a new criteria for the projected directions is proposed, which requires the regularized within-class scatter to be zero and the regularized total scatter to be greater than zero in the projected space.
- Third, the manifold regularized Foley–Sammon transform is used as a detect novelty method and evaluated on several toy and benchmark datasets.

The rest of this paper is organized as follows. A simple review of supervised novelty detection and kernel null Foley–Sammon transformation (KNFST) is provided in “Related work”. A manifold regularized null Foley–Sammon transformation (NFST) is proposed in “Manifold regularized NFST”. In “Experiments and simulations”, we evaluate manifold learning-novelty detection (MLND) on two toy datasets, eight benchmark datasets, and Gestures dataset. The last section is “Discussion and conclusion”.

Related work

A review of supervised novelty detection

The supervised novelty detection is used to predict the test sample whether is from an unknown class by learning a model from a training set which consists of massive labelled samples. When the labelled samples follow i.i.d. assumption, they can be considered to be from the same class and the supervised novelty detection is a one-class classification problem [2]. For instance, Scholkopf et al. [16] proposed to find a decision hyperplane which could make the minimum margin between the samples in reproducing kernel Hilbert space (RKHS) and the origin be maximized; David and Robert [17] proposed to find a hypersphere which could enclose most of the training samples with minimum volume; Ruff et al. [24] proposed a one-class classifier based on deep learning;

Iosifidis et al. [25] used extreme learning to do one-class classification, to name just a few.

When the labelled samples are from a mixture of distributions, they are from several known classes. Then, it is a multi-class supervised novelty detection problem [3]. Compared with multi-class classification, multi-class supervised novelty detection can identify whether a test sample is from an unknown class or which class the test sample is from if it comes from a known class. One way to solve multi-class supervised novelty detection is to treat all known classes as a superclass and learn a one-class classifier to detect the test sample whether is from an unknown class. If the test sample is not from an unknown class, then we train a multi-class classifier to predict which class it comes from [20]. Obviously, this way needs two models: a one-class classifier and a multi-class classifier. Additionally, the one-class classifier would be effected by the complex structure of the superclass. Another way to solve multi-class supervised novelty detection is to train several one-class classifiers. Each one-class classifier is associated with a known class [21]. It has many issues to train several models, such as more training time, more parameters to tune et al. To solve the issues in previous ways, some researchers proposed to learn a single model that can identify a test sample whether is from an unknown class and which class the test sample comes from if it is from a known class, simultaneously. For instance, Bodesheim et al. [3] proposed a multi-class supervised novelty detection method in which the samples from the same class are mapped to a single point in reproducing kernel Hilbert space (RKHS) via null projected directions (NPDs). Zhang et al. [26] proposed a semi-supervised version of KNFST and used it in person re-identification problem. Liu et al. [27] proposed a kernel null space discriminant analysis for incremental supervised novelty detection. Huang et al. [28] used incremental KNFST in person re-identification problem. T Ali and Chaudhuri [29] combined maximum margin metric learning with null space to do supervised novelty detection. The common ground of these methods is that they all adopt null space skill. However, null space only considers the global information and neglects the local geometrical structure. To solve this issue, we propose a manifold learning-based supervised novelty detection method in which the local geometrical structure is depicted by a manifold.

Recap of Kernel null Foley–Sammon transform (KNFST)

Foley–Sammon transform, also called Fisher transform or linear discriminant analysis (LDA), maximizes the between-class scatter and minimizes the within-class scatter, simultaneously. Let \mathbf{X}_j , $j = 1, \dots, c$ represent the set consisting of the samples which belong to class j ; the \mathbf{S}_w , \mathbf{S}_b , \mathbf{S}_t represent the within-class scatter, between-class scatter, and total scat-

ter matrices, respectively; $\varphi \in \mathbb{R}^D$ be one direction in the discriminant subspace. Then, the Fisher discriminant criterion is written as follows.

$$J(\varphi) = \frac{\varphi^T \mathbf{S}_b \varphi}{\varphi^T \mathbf{S}_w \varphi}. \tag{1}$$

Maximizing Eq. (1) can be done via solving a generalized eigenvalue problem as follows:

$$\mathbf{S}_b \varphi = \lambda \mathbf{S}_w \varphi. \tag{2}$$

The eigenvectors $\varphi^1, \dots, \varphi^k$ associated with the largest eigenvalues $\lambda_1, \dots, \lambda_k$ are selected as the discriminant directions.

In null Foley–Sammon transform (NFST), the direction should make the within-class scatter be zero and between-class scatter be positive. Therefore, Eq. (1) becomes $J(\varphi) = \infty$, which has the best separability. The solution in NFST should satisfy

$$\varphi^T \mathbf{S}_w \varphi = 0 \quad \text{and} \quad \varphi^T \lambda \mathbf{S}_b \varphi > 0. \tag{3}$$

The directions satisfying Eq. (3) are called null projection directions (NPDs). The Eq. (3) is equivalent to

$$\varphi^T \mathbf{S}_w \varphi = 0 \quad \text{and} \quad \varphi^T \lambda \mathbf{S}_t \varphi > 0. \tag{4}$$

Here, \mathbf{S}_t is the total scatter, $\mathbf{S}_t = \mathbf{S}_w + \mathbf{S}_b$. The samples in the same class are mapped to a single point due to $\varphi^T \mathbf{S}_w \varphi = 0$. An illustration of NFST is shown in Fig. 1.

Figure 1 is a three-class problem. The c_1, c_2 , and c_3 are the mapped points of class 1, class 2, and class 3 in null space. For a test sample, the associated label is decided by the minimum distance to the points, c_1, c_2 , and c_3 . If the test sample is far away from all points, it comes from an unknown class.

In both FST and NFST, within-class scatter and between-class scatter only capture global information and neglect local geometrical structure. In this paper, we adopt a manifold to regularize the within-class scatter and between-class scatter to describe the local structure in a class.

Manifold regularized NFST

Manifold learning for novelty detection

In manifold learning, it assumes that if two data points are close in the original distribution, they are also close in the projected subspace. In this paper, we use the neighborhood preserving embedding (NPE) to describe the local geometrical structure in within-class scatter and total scatter. Then,

we propose a regularized within-class scatter and a regularized total scatter instead of within-class scatter and local total scatter, respectively.

Definition 1 (Regularized within-class scatter) Given a dataset $\mathbf{X} \in \mathbb{R}^{N \times D}$. The associated label is \mathbf{Y} ($y_i \in \{1, \dots, c\}$). \mathbf{X}_j consists of all samples belonging to class j . The local within-class scatter is defined as

$$\begin{aligned} \mathbf{S}_{w\text{reg}} = & \sum_{i=1}^N \left(\alpha (\mathbf{x}_i - \boldsymbol{\mu}_j) - (1 - \alpha) \left(\mathbf{x}_i - \sum_{p=1}^N W_{i,p} \mathbf{x}_p \right) \right) \\ & \times \left(\alpha (\mathbf{x}_i - \boldsymbol{\mu}_j) - (1 - \alpha) \left(\mathbf{x}_i - \sum_{p=1}^N W_{i,p} \mathbf{x}_p \right) \right)^T. \end{aligned} \tag{5}$$

Here, \mathbf{W} is an adjacency graph. If \mathbf{x}_p is one of the k -nearest neighbors of \mathbf{x}_i and has the same label as that of \mathbf{x}_i , there is an edge between \mathbf{x}_i and \mathbf{x}_p ($W_{i,p} \neq 0$); otherwise, $W_{i,p} = 0$.

The Eq. (5) can be rewritten as follows

$$\begin{aligned} \mathbf{S}_{w\text{reg}} = & \sum_{i=1}^N \left(\mathbf{x}_i - \alpha \frac{1}{N_j} \sum_{\mathbf{x}_p \in \mathbf{X}_j} \mathbf{x}_p - (1 - \alpha) \sum_{p=1}^N W_{i,p} \mathbf{x}_p \right) \\ & \times \left(\mathbf{x}_i - \alpha \frac{1}{N_j} \sum_{\mathbf{x}_p \in \mathbf{X}_j} \mathbf{x}_p - (1 - \alpha) \sum_{p=1}^N W_{i,p} \mathbf{x}_p \right)^T. \end{aligned} \tag{6}$$

Let \mathbf{I} be a $N \times N$ identity matrix and \mathbf{L} be a block diagonal matrix whose block size is N_j and elements are $\frac{1}{N_j}$. The

$$\begin{aligned} \mathbf{S}_{w\text{reg}} = & \mathbf{X} (\alpha (\mathbf{I} - \mathbf{L}) + (1 - \alpha) (\mathbf{I} - \mathbf{W})) \\ & (\alpha (\mathbf{I} - \mathbf{L}) + (1 - \alpha) (\mathbf{I} - \mathbf{W}))^T \mathbf{X} = \mathbf{X} \\ & (\mathbf{I} - (\alpha \mathbf{L} + (1 - \alpha) \mathbf{W})) (\mathbf{I} - (\alpha \mathbf{L} + (1 - \alpha) \mathbf{W}))^T \mathbf{X}^T \end{aligned}$$

holds.

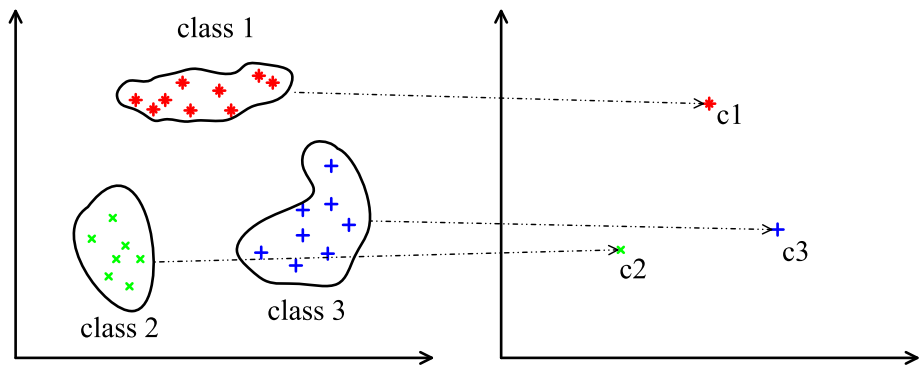
Let $\mathbf{X}_w = \mathbf{X} (\mathbf{I} - (\alpha \mathbf{L} + (1 - \alpha) \mathbf{W}))$. Then, the regularized within-class scatter is rewritten as $\mathbf{S}_{w\text{reg}} = \mathbf{X}_w \mathbf{X}_w^T$.

The weights are computed by minimizing the following objective function.

$$\min : \left\| \mathbf{x}_i - \sum_{j=1}^N W_{i,j} \mathbf{x}_j \right\|^2 \quad \text{s.t.} \quad \sum_{j=1}^N W_{i,j} = 1, \quad j = 1, \dots, N. \tag{7}$$

The term, $\sum_{\mathbf{x}_j \in KNN(\mathbf{x}_i) \& c(\mathbf{x}_j) \neq c(\mathbf{x}_i)} W_{i,j} \mathbf{x}_j$, can be regarded as the weighted mean of k -nearest neighbors of \mathbf{x}_i .

Fig. 1 An illustration of NFST. The stars, x-marks, and pluses belong to class 1, class 2, and class 3, respectively. The left sub-figure is the original space and the right-subfigure is null space. The samples from the same class in original space are mapped to a single point in null space



The details to solve formula (7) can refer to the reference [30].

Definition 2 (Regularized total scatter) Given a dataset $\mathbf{X} \in \mathbb{R}^{N \times D}$. The associated label is \mathbf{Y} ($y_i \in \{1, \dots, c\}$). \mathbf{X}_j consists of all samples belonging to class j . The regularized total scatter is defined as

$$\mathbf{S}_{\text{treg}} = \sum_{i=1}^N (\beta (\mathbf{x}_i - \boldsymbol{\mu}) + (1 - \beta) (\mathbf{x}_i - \boldsymbol{\mu}'))^T (\beta (\mathbf{x}_i - \boldsymbol{\mu}) + (1 - \beta) (\mathbf{x}_i - \boldsymbol{\mu}')) \quad (8)$$

Here, $\boldsymbol{\mu}'$ and $\boldsymbol{\mu}'_j$ are defined as Eq. (9) and (10), respectively.

$$\boldsymbol{\mu}' = \frac{1}{N_c} \sum_{j=1}^{N_c} \boldsymbol{\mu}'_j. \quad (9)$$

$$\boldsymbol{\mu}'_j = \frac{1}{N_j} \sum_{\mathbf{x}_i \in C_j} \sum_{\mathbf{x}_h \in KNN(\mathbf{x}_i)} W_{i,h} \mathbf{x}_h. \quad (10)$$

The Eq. (8) can be rewritten as follows:

$$\mathbf{S}_{\text{treg}} = \sum_{i=1}^N (\mathbf{x}_i - (\beta \boldsymbol{\mu} + (1 - \beta) \boldsymbol{\mu}'))^T (\mathbf{x}_i - (\beta \boldsymbol{\mu} + (1 - \beta) \boldsymbol{\mu}')) \quad (11)$$

The NPDs for regularized within-class scatter and regularized total scatter satisfy the following conditions.

$$\boldsymbol{\varphi}^T \mathbf{S}_{\text{wreg}} \boldsymbol{\varphi} = 0 \text{ and } \boldsymbol{\varphi}^T \mathbf{S}_{\text{treg}} \boldsymbol{\varphi} > 0. \quad (12)$$

Since \mathbf{S}_{wreg} is a semipositive definite matrix and can be represented as $\mathbf{S}_{\text{wreg}} = \mathbf{X}_w \mathbf{X}_w^T$, we can obtain $\boldsymbol{\varphi}^T \mathbf{X}_w \mathbf{X}_w^T \boldsymbol{\varphi} = 0 \Rightarrow (\mathbf{X}_w^T \boldsymbol{\varphi})^T (\mathbf{X}_w^T \boldsymbol{\varphi}) = 0 \Rightarrow \mathbf{X}_w^T \boldsymbol{\varphi} = 0$. By multiplying both sides of the equation from the left with matrix \mathbf{X}_w , then we can obtain $\mathbf{X}_w \mathbf{X}_w^T \boldsymbol{\varphi} = 0 \Rightarrow \mathbf{S}_{\text{wreg}} \boldsymbol{\varphi} = 0$ holds.

On the other hand, the solution of $\mathbf{S}_{\text{wreg}} \boldsymbol{\varphi} = 0$ exactly satisfies $\boldsymbol{\varphi}^T \mathbf{S}_{\text{wreg}} \boldsymbol{\varphi} = 0$. The $\mathbf{S}_{\text{wreg}} \boldsymbol{\varphi} = 0 \Leftrightarrow \boldsymbol{\varphi}^T \mathbf{S}_{\text{wreg}} \boldsymbol{\varphi} = 0$ holds.

From the above two points, we can obtain $\boldsymbol{\varphi}^T \mathbf{S}_{\text{wreg}} \boldsymbol{\varphi} = 0 \Leftrightarrow \mathbf{S}_{\text{wreg}} \boldsymbol{\varphi} = 0$.

Let $\mathbf{Z}_w = \{\mathbf{z} | \mathbf{S}_{\text{wreg}} \mathbf{z} = 0\}$ be the null space of \mathbf{S}_{wreg} , $\mathbf{Z}_t = \{\mathbf{z} | \mathbf{S}_{\text{treg}} \mathbf{z} = 0\}$ be the null space of \mathbf{S}_{treg} , and \mathbf{Z}_t^\perp be the orthogonal complement space of \mathbf{Z}_t . The NPDs satisfy

$$\boldsymbol{\varphi} \in \mathbf{Z}_t^\perp \cap \mathbf{Z}_w. \quad (13)$$

In order to ensure $\boldsymbol{\varphi} \in \mathbf{Z}_t^\perp$, each $\boldsymbol{\varphi}$ can be represented as

$$\boldsymbol{\varphi} = \gamma_1 \boldsymbol{\theta}_1 + \gamma_2 \boldsymbol{\theta}_2 + \dots + \gamma_m \boldsymbol{\theta}_m = \mathbf{Q} \boldsymbol{\gamma}. \quad (14)$$

Here, $\mathbf{Q} = (\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \dots, \boldsymbol{\theta}_m)$ and $\boldsymbol{\gamma} = (\gamma_1, \gamma_2, \dots, \gamma_m)$. Since the \mathbf{S}_{treg} is semipositive definite, the solution of $\boldsymbol{\varphi}^T \mathbf{S}_{\text{treg}} \boldsymbol{\varphi} > 0$ is just \mathbf{Z}_t^\perp ($\mathbf{Z}_t = \{\mathbf{z} | \mathbf{S}_{\text{treg}} \mathbf{z} = 0\}$). It means that the subspace $\boldsymbol{\varphi}^T \mathbf{S}_{\text{treg}} \boldsymbol{\varphi} > 0$ is just the \mathbf{Z}_t^\perp . Let $\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \dots, \boldsymbol{\theta}_m$ be the basis of subspace which is spanned by $\mathbf{x}_1 - (\beta \boldsymbol{\mu} + (1 - \beta) \boldsymbol{\mu}'), \dots, \mathbf{x}_m - (\beta \boldsymbol{\mu} + (1 - \beta) \boldsymbol{\mu}')$ and can be obtained by principal component analysis (PCA).

Substituting Eq. (14) into $\boldsymbol{\varphi}^T \mathbf{S}_{\text{wreg}} \boldsymbol{\varphi} = 0$, then $(\mathbf{Q})^T \mathbf{S}_{\text{wreg}} (\mathbf{Q}) = 0$. It is equivalent to the following eigenproblem.

$$(\mathbf{Q}^T \mathbf{S}_{\text{wreg}} \mathbf{Q}) \boldsymbol{\gamma} = 0. \quad (15)$$

Due to $\mathbf{X}_w = \mathbf{X} (\mathbf{I} - (\alpha \mathbf{L} + (1 - \alpha) \mathbf{W}))$ and $\mathbf{S}_{\text{wreg}} = \mathbf{X}_w \mathbf{X}_w^T$, the Eq. (15) can be rewritten as follows.

$$\mathbf{H} \mathbf{H}^T \boldsymbol{\gamma} = 0. \quad (16)$$

Here, $\mathbf{H} = \mathbf{Q}^T \mathbf{X}_w = \mathbf{Q}^T \mathbf{X} (\mathbf{I} - (\alpha \mathbf{L} + (1 - \alpha) \mathbf{W}))$. Eq. (16) is an eigenproblem. After solve Eq. (16), the null projection directions $\boldsymbol{\gamma}_1, \boldsymbol{\gamma}_2, \dots, \boldsymbol{\gamma}_l$ can be obtained from Eq. (14).

The null projection directions $\boldsymbol{\gamma}_1, \boldsymbol{\gamma}_2, \dots, \boldsymbol{\gamma}_l$ are associated different eigenvalues $\lambda_1, \lambda_2, \dots, \lambda_l$ ($\lambda_i \neq \lambda_j$). From Eq. (16), we can obtain $\boldsymbol{\gamma}_i^T \mathbf{H} \mathbf{H}^T = \lambda_i \boldsymbol{\gamma}_i^T, \boldsymbol{\gamma}_j^T \mathbf{H} \mathbf{H}^T = \lambda_j \boldsymbol{\gamma}_j^T \Rightarrow \boldsymbol{\gamma}_i^T \mathbf{H} \mathbf{H}^T \boldsymbol{\gamma}_j = \lambda_i \boldsymbol{\gamma}_i^T \boldsymbol{\gamma}_j, \boldsymbol{\gamma}_j^T \mathbf{H} \mathbf{H}^T \boldsymbol{\gamma}_i = \lambda_j \boldsymbol{\gamma}_j^T \boldsymbol{\gamma}_i \Rightarrow 0 = (\lambda_i - \lambda_j) \boldsymbol{\gamma}_i^T \boldsymbol{\gamma}_j$. Due to $\lambda_i \neq \lambda_j, \boldsymbol{\gamma}_i^T \boldsymbol{\gamma}_j = 0$ holds. Therefore, $\boldsymbol{\gamma}_i$ and $\boldsymbol{\gamma}_j$ are orthogonalized.

In Eq. (14), the matrix \mathbf{Q} is obtained from PCA. The column vectors of \mathbf{Q} are orthogonalized. We can obtain

$\varphi_i^T \varphi_j = (\mathbf{Q}\gamma_i)^T (\mathbf{Q}\gamma_j) = \gamma_i^T \mathbf{Q}^T \mathbf{Q}\gamma_j = \gamma_i^T \gamma_j = 0$. Therefore, the directions obtained from Eq. (14) are orthogonalized as well.

Let \mathbf{P} be a matrix whose columns are null projection directions $\varphi_1, \dots, \varphi_l$ ($l < N$). A test sample \mathbf{x} is mapped into null space via Eq. (17) and scored via Eq. (18).

$$\mathbf{x}^* = \mathbf{P}\mathbf{x}. \tag{17}$$

$$\text{Score}(\mathbf{x}) = \min_{1 \leq j \leq c} \text{dist}(\mathbf{x}^*, \mathbf{t}_j). \tag{18}$$

Here, \mathbf{t}_j is the mapped point of class j in null space. The $\text{Score}(\mathbf{x})$ is the novelty score of \mathbf{x} , which reflects the probability of a test sample from an unknown class. When it is very large, \mathbf{x} is a novelty with high probability.

The procedure to find NPDs is summarized as the following Algorithm 1

Algorithm 1 (Manifold Learning for Novelty Detection, MLND)

- Input:**
- The training set X and the label Y
 - The number of nearest neighbours in manifold
 - Parameters α, β
- Output:**
- NPDs $\varphi_j, j = 1, \dots, l$
- 1: Compute the regularized within-class scatter and total scatter;
 - 2: Compute the orthonormal basis $\theta_1, \theta_2, \dots, \theta_m$ of the subspace \mathbf{Z}_l^\perp ;
 - 3: Solve the eigenvalue problem Eq. (16) to get $\gamma_1, \gamma_2, \dots, \gamma_l$;
 - 4: Output the NPDs via Eq. (14).

Compared with null space Foley–Sammon transform (NFST), the extra cost of MLND includes solving Eq. (7) and calculating regularized within-class scatter and regularized total scatter. The rest cost of MLND is the same as that of NFST.

Kernel form manifold learning for novelty detection

In MLND, it assumes that $\mathbf{S}_{w\text{reg}}$ is singular. When $\mathbf{S}_{w\text{reg}}$ is full rank, the samples are mapped into a reproducing Hilbert space (RKHS) via kernel trick to avoid the null space \mathbf{Z}_w being empty. The mapping of a sample \mathbf{x} in RKHS is represented as $\Phi(\mathbf{x})$ where $\Phi(\mathbf{x})$ is an implicit function. The inner product of the mappings of two samples can be calculated via kernel function which is defined as $k(\mathbf{x}_i, \mathbf{x}_j) = \langle \Phi(\mathbf{x}_i), \Phi(\mathbf{x}_j) \rangle$, such as Radial Basis Function (RBF) kernel $k(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|}{2\sigma^2}\right)$. Obviously, $\mathbf{S}_{w\text{reg}}$ is a high dimensional space other than a $d \times d$ matrix any more. For instance, $\mathbf{S}_{w\text{reg}}$ is a $\text{inf} \times \text{inf}$ matrix when RBF kernel is adopted.

Let $\tilde{\Phi}(\mathbf{x}_i) = \Phi(\mathbf{x}_i) - \left(\frac{\beta}{N} \sum_{j=1}^N \Phi(\mathbf{x}_j) + \frac{1-\beta}{N} \sum_{j=1}^N \sum_{\mathbf{x}_h \in K_{NN}(\mathbf{x}_i)} W_{i,j} \Phi(\mathbf{x}_h)\right)$, $\tilde{\mathbf{K}} = (\mathbf{I} - (\beta \mathbf{1}_N + (1-\beta) \mathbf{1}_N \mathbf{W})) \mathbf{K} (\mathbf{I} - (\beta \mathbf{1}_N + (1-\beta) \mathbf{1}_N \mathbf{W}))^T$, $\tilde{\mathbf{X}} = [\Phi(\mathbf{x}_1), \dots, \Phi(\mathbf{x}_N)]$, \mathbf{K} be the kernel matrix where $K(i, j) = \langle \Phi(\mathbf{x}_i), \Phi(\mathbf{x}_j) \rangle$. Then, \mathbf{S}_{treg} in RKHS is rewritten as follows

$$\mathbf{S}_{\text{treg}} = \sum_{i=1}^N \tilde{\Phi}(\mathbf{x}_i) \tilde{\Phi}(\mathbf{x}_i)^T. \tag{19}$$

The eigenvector θ_j in high dimensional feature space lies in the space of $\tilde{\Phi}(\mathbf{x}_1), \dots, \tilde{\Phi}(\mathbf{x}_N)$ and there exist coefficients $\delta_{1,j}, \dots, \delta_{N,j}$ satisfying the following equation.

$$\theta_j = \sum_{i=1}^N \delta_{i,j} \tilde{\Phi}(\mathbf{x}_i). \tag{20}$$

The eigenvalues and eigenvectors of \mathbf{S}_{treg} satisfy

$$\lambda_j \theta_j = \mathbf{S}_{\text{treg}} \theta_j. \tag{21}$$

Then,

$$\lambda_j \tilde{\Phi}(\mathbf{x}_i)^T \theta_j = \tilde{\Phi}(\mathbf{x}_i)^T \mathbf{S}_{\text{treg}} \theta_j, \quad i = 1, \dots, l. \tag{22}$$

Substituting Eqs. (20) and (21) into Eq. (22), we can obtain

$$\lambda_j \tilde{\mathbf{K}} \delta_j = \tilde{\mathbf{K}} \tilde{\mathbf{K}} \delta_j. \tag{23}$$

Here, δ_j is the vector form of $\delta_{1,j}, \dots, \delta_{N,j}$ and can be obtained by solving the following eigenvalue problem

$$\lambda_j \delta_j = \tilde{\mathbf{K}} \delta_j. \tag{24}$$

Since $\langle \theta_j, \theta_j \rangle = \delta_j^T \tilde{\mathbf{K}} \delta_j = \lambda_j \langle \delta_j, \delta_j \rangle$, the orthonormal basis of \mathbf{S}_{treg} in high dimensional space is represented as follows:

$$\tilde{\theta}_j = \sum_{i=1}^N \tilde{\delta}_{i,j} \tilde{\Phi}(\mathbf{x}_i). \tag{25}$$

Here $\tilde{\delta}_{i,j} = \lambda_j^{-\frac{1}{2}} \delta_{i,j}$. Equation (25) can be solved implicitly. By introducing Eq. (25) and inner products in reproducing kernel Hilbert space (RKHS), the matrix \mathbf{H} is rewritten as follows.

$$\mathbf{H} = ((\mathbf{I} - (\beta \mathbf{1}_N + (1-\beta) \mathbf{1}_N \mathbf{W})) \tilde{\mathbf{V}})^T \mathbf{K} (\alpha (\mathbf{I} - \mathbf{L}) + (1-\alpha) (\mathbf{I} - \mathbf{W})). \tag{26}$$

Here, $\tilde{\mathbf{V}} = \{\theta_1, \dots, \theta_l\}$. Then, substituting Eq. (26) into Eq. (16), we can obtain γ_j in RKHS. The final null space

directions in RKHS are obtained by the

$$\boldsymbol{\varphi} = ((\mathbf{I} - \mathbf{1}_N)\tilde{\mathbf{V}}) \boldsymbol{\gamma}_j. \quad (27)$$

Let $\mathbf{P} = [\boldsymbol{\varphi}_1, \dots, \boldsymbol{\varphi}_l]$. In kernel MLND, the test sample \mathbf{x}^* is mapped into null space through $\mathbf{K}_*^T \mathbf{P}$. The \mathbf{K}_*^T is the vector form of $[k(\mathbf{x}_1, \mathbf{x}_*); \dots; k(\mathbf{x}_n, \mathbf{x}_*)]$. The novelty score of \mathbf{x}^* is the minimum distance between the mapped point to each class. The procedure of kernel MLND is summarized as Algorithm 2.

Algorithm 2 (Kernel MLND)

Input:

- The training set X and the label Y
- The number of nearest neighbours in manifold
- Parameters α, β

Output:

- NPDs $\varphi_j, j = 1, \dots, l$

- 1: Compute K ;
 - 2: Solve Eq. (24) to get $\delta_j, j = 1, \dots, l$;
 - 3: Obtain the matrix \mathbf{H} via Eq. (26);
 - 4: Solve the eigenvalue problem Eq. (16) to get $\boldsymbol{\gamma}_1, \dots, \boldsymbol{\gamma}_N$;
 - 5: Output the NPDs via Eq. (27).
-

When the parameters $\alpha = 1$ and $\beta = 1$, kernel form MLND degenerates as KNFST. When $\alpha = 0$ and $\beta = 0$, the \mathbf{S}_{wreg} is just a LLE manifold. Compared with kernel null space Foley–Sammon transform (KNFST), the extra cost of MLND includes solving Eq. (7) and the time complexity of rest part is the same as that of KNFST.

Experiments and simulations

In this section, manifold learning novelty detection (MLND) will be evaluated on several datasets. Here, we use the kernel MLND. The code of MLND is implemented by Matlab 2018b. To verify the validity of MLND, we compare MLND with some state-of-the-art null space methods, including KNFST [3], Local KNFST [31], and NK3ML [29]. The codes of KNFST, Local KNFST, and NK3ML are provided by the authors.

The generalized histogram intersection kernel (HIK) is used as kernel function in KNFST, Local KNFST, and NK3ML. For fair comparison, the HIK is used as kernel function, which is defined as $k(\mathbf{x}_i, \mathbf{x}_j) = \exp(2\kappa_{\text{HIK}}(\mathbf{x}_i, \mathbf{x}_j) - \kappa_{\text{HIK}}(\mathbf{x}_i, \mathbf{x}_i) - \kappa_{\text{HIK}}(\mathbf{x}_j, \mathbf{x}_j))$. The $\kappa_{\text{HIK}}(\mathbf{x}_i, \mathbf{x}_j)$ is defined as $\kappa_{\text{HIK}}(\mathbf{x}_i, \mathbf{x}_j) = \sum_{d=1}^D \min(x_{i,d}, x_{j,d})$.

First, we adopt an EMG dataset to demonstrate the effectiveness of MLND on real dataset; then, two toy datasets are used to further evaluate MLND; lastly several benchmark

datasets which are collected by UCI¹ or Libsvm website [32] are adopted to further evaluate MLND. The experimental results are reported in terms of AUC value, ROC curve and accuracy. The AUC value and ROC curve are used to evaluate novelty detection methods. The higher AUC value is, the better the novelty detector is. Accuracy is defined as the ratio of correctly predicted normal samples to all normal samples. It is used to measure the classification performance of multi-class supervised novelty detection for normal samples.

Experiments on EMG dataset

In this section, we use Gestures, which is an electromyogram signal (EMG) dataset, to verify MLND. The signals are collected via MYO Thalmic bracelet which is worn on user's forearm. The bracelet is equipped with 8 sensors to collect myographic signals simultaneously. The raw signals are from 36 subjects. Each subject performs 2 series. Each series contains 6 or 7 basic gestures: hand at rest, hand clenched in a fist, wrist flexion, wrist extension, radial deviations, ulnar deviations, and extended palm. In this experiment, we only consider former six gestures since the extended palm is not performed in some subjects. An illustration of the signals of former six gestures is shown in Fig. 2. The label of horizontal axis is the channel which the signal are collected from. A channel is associated with a sensor in MYO Thalmic bracelet.

Different from previous gesture recognition work [33], this paper converts the gesture recognition as a multi-class supervised novelty detection problem to identify unknown gesture. Except seven basic gestures, some of the signals are not marked as basic gestures. In this section, we use hand at rest, hand clenched in a fist, wrist flexion, wrist extension, radial deviations, and ulnar deviations as normal classes. The extended palm and unmarked signals are used as anomalies. Therefore, the task of Gesture recognition becomes to recognize whether the EMG signal are from basic hand gestures or which basic hand gesture the EMG signal comes from if it belongs to one of the basic hand gestures. Obviously, it is a multi-class supervised novelty detection problem. The gesture recognition can be widely used in robot control [34,35] and traffic control [36,37].

We set a 200 ms window for sampling. The window overlaps with a 100 ms step. Then, we generate 30,240 normal samples (5,040 samples per class) and 10,000 abnormal samples as novelties. The normal samples are divided into two parts equally. One part is used as a training set, the other part and abnormal samples are used as a test set. The features from eight channels are reorganized as an 800*1 vector.

In MLND, we directly set parameters as ($\alpha = 0.5, \beta = 0.5$). The number of nearest neighbors in Definition 1 and

¹ <https://archive.ics.uci.edu/ml/index.php>.

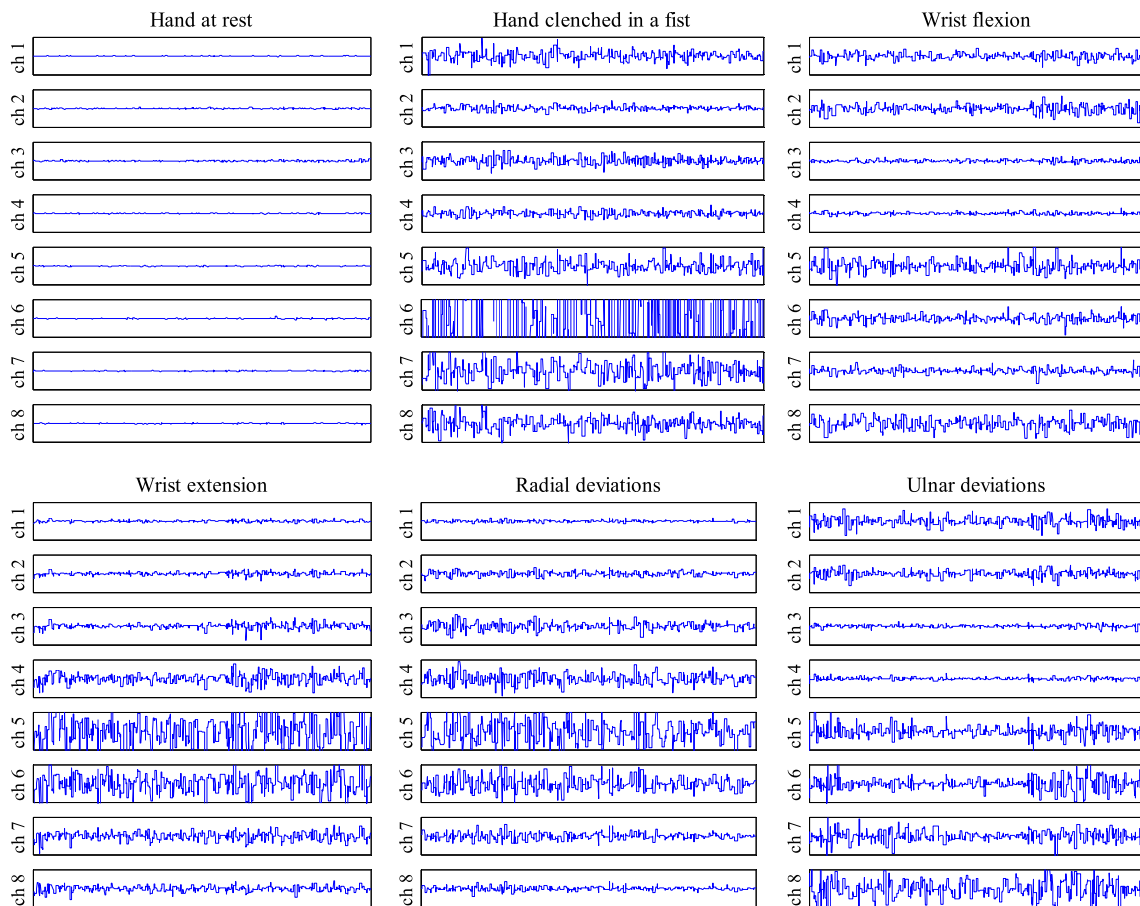


Fig. 2 The signals of hand gestures. In the first row, the signals are from hand at rest, hand clenched in a fist, and wrist flexion, respectively. In the second row, the signals are from wrist extension, radial deviations,

and ulnar deviations, respectively. The ch1 to ch8 are associated with eight sensors on MYO Thalmic bracelet

Definition 2 is directly set as 20 to avoid extra cost to tune parameters. To avoid randomness, we repeat the experiment 30 times. The results are reported as mean \pm std. form in terms of AUC value and accuracy. The results are reported in Table 1.

From Table 1, it can be found the average AUC value of MLND reaches 0.9251 which is higher than KNFST, Local KNFST, and NK3ML; the average accuracy of MLND reaches 93.87% which is also higher than KNFST, Local KNFST, and NM3ML. The ROC curve of one trail is drawn in Fig. 3.

From Fig. 3, the ROC curve of MLND is still superior to that of KNFST, Local KNFST, and NK3ML. MLND per-

forms better than KNFST, Local KNFST, and NK3ML on Gesture.

Furthermore, we also consider the influence of different parameter k in Definitions 1 and 2 on the performance of MLND. The parameter k is in the range from 10 to 100 steps by 10. Here, the parameters α and β are both set as 0.5 directly. The curve between AUC value and the parameter k is shown in the left sub-figure of Fig. 4. The curve between accuracy and the parameter k is shown in the right sub-figure of Fig. 4.

From the results in Fig. 4, it can be found that both AUC value and accuracy decrease with the increasing of the number of k nearest neighbors in MLND when $k > 30$. The reason is that the manifold is used to depict a small

Table 1 The experimental results of gestures recognition

	KNFST	Local KNFST	NK3ML	MLND
AUC value	0.8919 \pm 0.0034	0.8986 \pm 0.0041	0.9152 \pm 0.0047	0.9251 \pm 0.0051
Acc. (%)	90.21 \pm 0.19	90.73 \pm 0.27	91.79 \pm 0.23	93.87 \pm 0.37

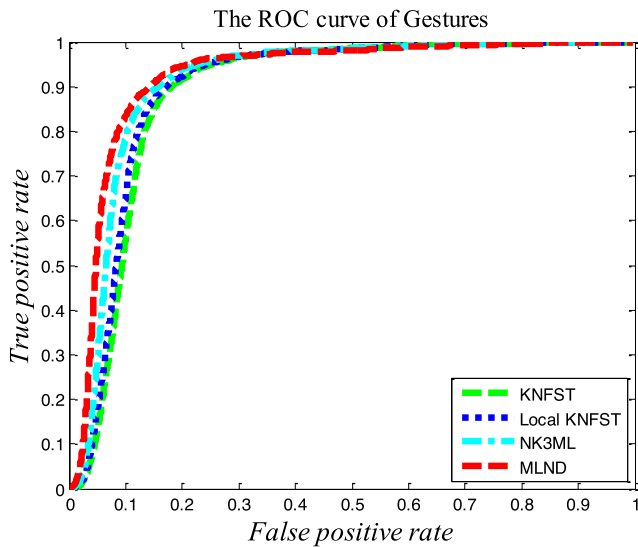


Fig. 3 The AUC curve of Gestures. The greed dashed, dotted, dashdot, and red dashed represent the ROC curves of KNFST, Local KNFST, NK3ML, and MLND respectively

region. When the neighborhood is too large, the manifold is invalid. When $k = 20$, the AUC value reaches the peak ($AUC = 0.9152$). When $k = 30$, the accuracy reaches the peak ($accuracy = 94.65\%$). In our experience, the parameter k in MLND cannot be set too large. In the following experiments, the parameter k is set as 20 directly.

Experiments on toy datasets

In this subsection, we will evaluate MLND on two toy datasets. The first one contains 3 normal classes and the second one contains 2 normal classes. In toy 1, the samples in $\mathbf{X}_j, j = 1, 2, 3$ follows the below distributions:

$$\mathbf{X}_j = \mathbf{N}_j. \tag{28}$$

Here, $\mathbf{N}_1 \sim \left([0 \ 0], \begin{bmatrix} 0.5^2 & 0 \\ 0 & 1.25^2 \end{bmatrix} \right)$,
 $\mathbf{N}_2 \sim \left([2 \ 0], \begin{bmatrix} 0.5^2 & 0 \\ 0 & 1.25^2 \end{bmatrix} \right), \mathbf{N}_3 \sim \left([4 \ 0], \begin{bmatrix} 0.5^2 & 0 \\ 0 & 1.25^2 \end{bmatrix} \right)$.
 An illustration of toy 1 is shown in Fig. 5a.

In toy 2, the samples in $\mathbf{X}_j, j = 1, 2$ follows the below distributions:

$$\mathbf{X}_j = \mathbf{N}_j + \left[1 - \frac{x_{i,2}^3}{25} + \epsilon \right]. \tag{29}$$

Here, $\mathbf{N}_1 \sim \left([0 \ 0], \begin{bmatrix} 2^2 & 0 \\ 0 & 2^2 \end{bmatrix} \right), \mathbf{N}_2 \sim \left([3 \ 3], \begin{bmatrix} 2^2 & 0 \\ 0 & 2^2 \end{bmatrix} \right)$,
 $\epsilon \sim N(0, 0.25^2)$. An illustration of toy 2 is shown in Fig. 5b.

In toy 1, we generate 600 samples in the training set (200 samples per class) and 2000 samples in the test set (500 samples per class and 500 novelties). In toy 2, we generate 400 samples in the training set (200 samples per class) and 1500 samples in the test set (500 samples per class and 500 novelties). To refrain the randomness, we repeat the experiments 30 times. The AUC value and accuracy are reported in the form of mean \pm std. in Table 2.

In Table 2, the sixth column represents the results of MLND via fine-tuned parameters α, β . The parameters α, β are tuned via grid search in the range from 0.1 to 1 stepped 0.1. The seventh column of Table 2 represents the results of MLND with fixed parameters $\alpha = 0.5, \beta = 0.5$.

For Toy 1, the average AUC value of MLND is 0.9589 when tuning parameters α, β via grid search and is 0.9492 when $\alpha = 0.5, \beta = 0.5$. For Toy 2, the average AUC value of MLND is 0.9314 when tuning parameters α, β via grid search and is 0.9249 when $\alpha = 1, \beta = 1$. The average AUC value of

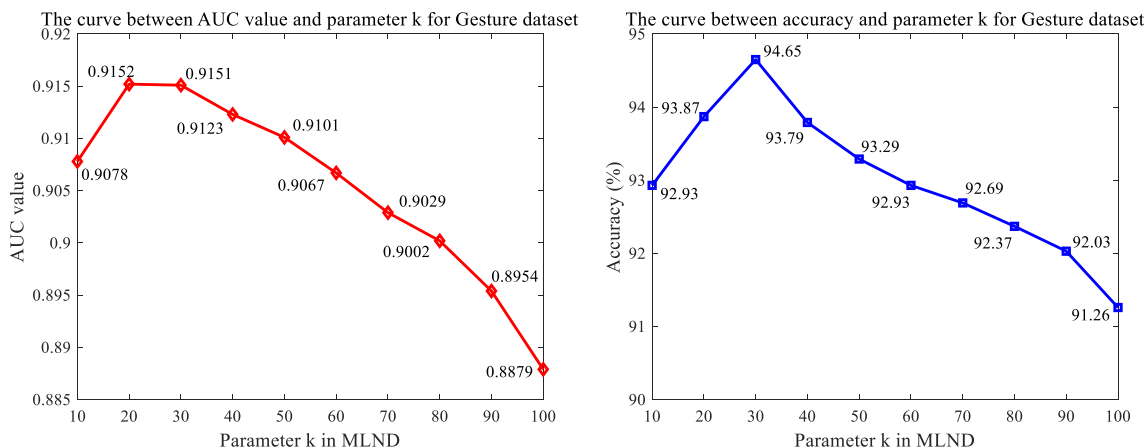


Fig. 4 The performance with different parameter k in MLND. Left: the curve between the AUC value and the parameter k ; right: the curve between the accuracy and the parameter k

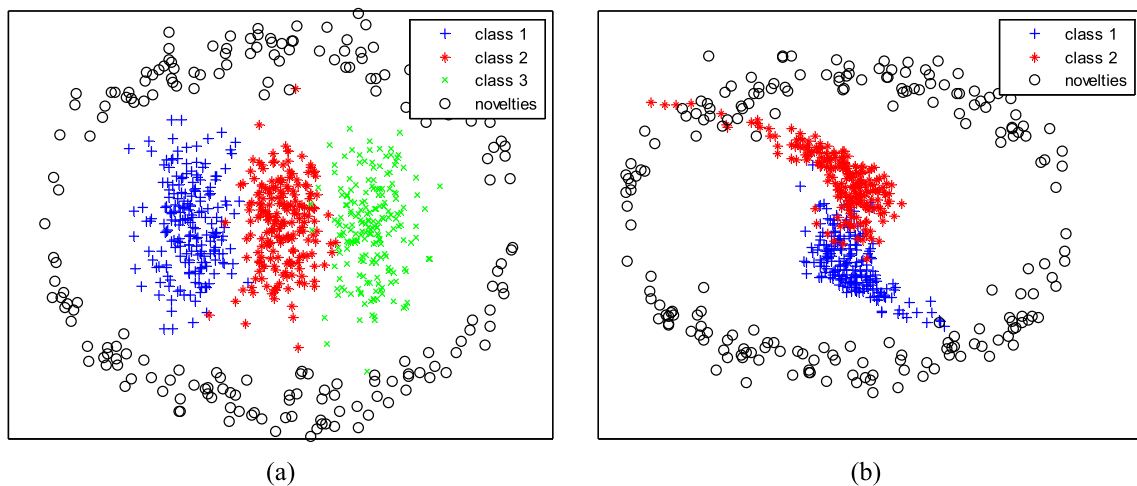


Fig. 5 The illustration of toy datasets. In **a**, the samples from class 1, 2, and 3 are denoted as pluses, stars, and x-marks, respectively; the novelties are denoted as circles. In **b**, there are two normal classes in which the samples are denoted as pluses and stars, respectively; the circles are novelties

Table 2 The experimental results of toy datasets

	KNFST	Local KNFST	NK3ML	MLND	
				Fine-tuned α, β	$\alpha = 0.5, \beta = 0.5$
Toy 1					
AUC value	0.9159±0.0026	0.9250±0.0031	0.9398±0.0023	0.9589±0.0042	0.9492±0.0047
Acc. (%)	92.17±0.54	92.53±0.67	94.01±0.57	96.71±0.63	95.17±0.58
Toy 2					
AUC value	0.8977±0.0034	0.9074±0.0029	0.9165±0.0033	0.9314±0.0045	0.9249±0.0051
Acc. (%)	90.34±0.61	90.43±0.52	92.53±0.64	93.47±0.72	93.28±0.67

MLND is higher than KNFST, Local KNFST, and NK3ML even the parameters are directly set as $\alpha = 0.5, \beta = 0.5$.

For Toy 1, the average accuracy of MLND is 96.71% when tuning parameters α, β via grid search and is 95.17% when $\alpha = 0.5, \beta = 0.5$. For Toy 2, the average accuracy of MLND is 93.47% when tuning parameters α, β via grid search and is 93.28% when $\alpha = 0.5, \beta = 0.5$. The average accuracy of MLND is higher than KNFST, Local KNFST, and NK3ML even the parameters are directly set as $\alpha = 0.5, \beta = 0.5$ as well.

This is because MLND considers both global information and local structure in class. The ROC curves of Toy 1 and Toy 2 are shown in Fig. 6.

From Fig. 6, we can obtain the same conclusion that is from AUC value and accuracy for Toy 1 and Toy 2.

Experiments on benchmark datasets

In this subsection, we will compare MLND with KNFST, Local KNFST, and NK3ML on several benchmark dataset which are collected from UCI repository and Libsvm website [32]. The details of these datasets are listed in Table 3.

These datasets are reorganized to suit for evaluating multi-class supervised novelty detection. For DNA, protein, satimage, and shuttle, we remove a class from the training set and add the samples from this class into a test set for testing. For pendigits poker, SVHN, and usps, we remove five classes from the training set and add the samples from these classes into the test set for testing. The parameters in MLND is directly set as $\alpha = 0.5, \beta = 0.5$ and $k = 20$. The AUC value and accuracy are reported in Tables 4 and 5, respectively.

In Tables 4 and 5, the last row is the win-loss-tie (W-L-T) of AUC value and accuracy, respectively. The MLND is used as a based method. From Table 4, it can be found that the AUC value of MLND is higher than that of KNFST on eight datasets, Local KNFST on eight datasets, and NK3ML on seven datasets. From Table 5, it can be found that the accuracy of MLND is higher than that of KNFST on eight datasets, Local KNFST on eight datasets, and NK3ML on six datasets. The MLND is superior to KNFST, Local KNFST, and NK3ML on most of these benchmark datasets.

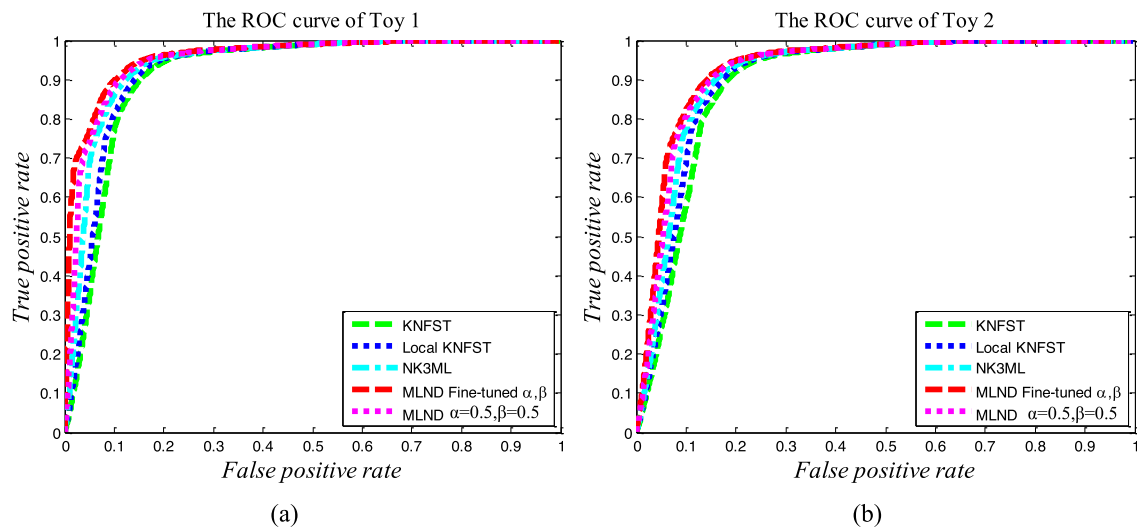


Fig. 6 The ROC curves of toy datasets. **a** Toy 1; **b** Toy 2. The green dashed, dotted, dashdot, red dashed, and magenta dotted lines represent the ROC curves of KNFST, Local KNFST, NK3ML, MLND with fine-tuned parameters, and MLND with fixed parameters ($\alpha = 0.5, \beta = 0.5$), respectively

Table 3 The details of benchmark datasets

Datasets	#Classes	#Training samples (classes)	#Test samples (classes)	#Features
DNA	3	2000	1186	180
Protein	3	17,766	6621	357
Satimage	6	4435	2000	36
Shuttle	7	43,500	14,500	9
Pendigits	10	7494	3498	16
Poker	10	25,010	1,000,000	10
SVHN	10	73,257	26,032	3072
usps	10	7291	2007	256

Table 4 The AUC value results of benchmark datasets

Datasets	KNFST	Local KNFST	NK3ML	MLND
DNA	0.8623	0.8773	0.8832	0.8907
Protein	0.8359	0.8438	0.8542	0.8601
Satimage	0.9368	0.9342	0.9439	0.9397
Shuttle	0.7987	0.8051	0.8115	0.8179
Pendigits	0.8907	0.8832	0.8944	0.9073
Poker	0.8347	0.8215	0.8279	0.8369
SVHN	0.7853	0.7974	0.8093	0.8164
USPS	0.9304	0.9357	0.9423	0.9546
Avg.	0.8593	0.8622	0.8708	0.8779
W-L-T	0-8-0	0-8-0	1-7-0	–

Table 5 The accuracy results of benchmark dataset

Datasets	KNFST	Local KNFST	NK3ML	MLND
DNA	86.73	87.14	89.03	91.31
Protein	84.34	85.47	86.63	87.74
Satimage	94.31	93.92	95.44	94.97
Shuttle	80.23	79.87	83.11	82.91
Pendigits	90.35	90.93	90.78	91.69
Poker	83.58	83.31	84.07	85.94
SVHN	79.37	81.09	82.36	84.15
USPS	94.06	95.13	95.89	96.93
Avg.	86.62	87.1	88.41	89.45
W-L-T	0-8-0	0-8-0	2-6-0	–

Discussions and conclusion

In this paper, we propose a manifold learning-based novelty detection method. The manifold learning novelty detection (MLND) can be regarded as an improvement of kernel null

space Foley–Sammon transformation (KNFST). In MLND, first we introduce a manifold into within-class scatter and total scatter to depict the local geometrical structure in class; then we map the samples from the same class into a single

point via null projection directions. Compared with KNFST, MLND considers both global information and local geometrical structure in the class. Therefore, MLND can overcome the weakness of KNFST caused by ignoring local geometrical structure in the class. We evaluate MLND on an EMG Gesture dataset, two toy dataset and eight benchmark datasets. The experimental results demonstrate MLND is superior to KNFST and its two improved methods: Local KNFST and NK3ML.

Acknowledgements The authors would like to thank the editor and the anonymous reviewers for their critical and constructive comments and suggestions. This work is partially supported by the University Philosophy and Social Science Research Project (2021SJA0522), and the Major Project of the Natural Science Foundation of the Jiangsu Higher Education Institutions of China (19KJA510004, 21KJA120001).

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Chalapathy R, Chawla S (2019) Deep learning for anomaly detection: a survey. arXiv preprint [arXiv:1901.03407](https://arxiv.org/abs/1901.03407)
- Chandola V, Banerjee A, Kumar V (2009) Anomaly detection: a survey. *ACM Comput Surv (CSUR)* 41(3):1–58
- Bodesheim P, Freytag A, Rodner E, et al (2013) Kernel null space methods for novelty detection. *Proceedings of the IEEE conference on computer vision and pattern recognition*. pp 3374–3381
- Chan FTS, Wang ZX, Patnaik S et al (2020) Ensemble-learning based neural networks for novelty detection in multi-class systems. *Appl Soft Comput* 93:106396
- Xie X, Wang C, Chen S, et al (2017) Real-time illegal parking detection system based on deep learning. *Proceedings of the 2017 international conference on deep learning technologies*. pp 23–27
- Schlegl T, Seeböck P, Waldstein SM, et al (2017) Unsupervised anomaly detection with generative adversarial networks to guide marker discovery. *International conference on information processing in medical imaging*. Springer, Cham, pp 146–157
- Javaid A, Niyaz Q, Sun W, et al (2016) A deep learning approach for network intrusion detection system. *Proceedings of the 9th EAI international conference on bio-inspired information and communications technologies*. pp 21–26
- Kravchik M, Shabtai A (2018) Detecting cyber attacks in industrial control systems using convolutional neural networks. *Proceedings of the 2018 workshop on cyber-physical systems security and privacy*. pp 72–83
- Mehdi M, Ala A-F, Sameh S, Mohsen G (2017) Deep learning for iot big data and streaming analytics: a survey. arXiv preprint [arXiv:1712.04301](https://arxiv.org/abs/1712.04301)
- Zhang W, Zhang B, Zhu W et al (2021) Comprehensive assessment of MODIS-derived near-surface air temperature using wide elevation-spanned measurements in China. *Sci Total Environ* 800:149535
- Guo Z, Min A, Yang B et al (2021) A sparse oblique-manifold nonnegative matrix factorization for hyperspectral unmixing. *IEEE Trans Geosci Remote Sens* 60:1–13
- Koundal D, Sharma B, Guo Y (2020) Intuitionistic based segmentation of thyroid nodules in ultrasound images. *Comput Biol Med* 121:103776
- Koundal D, Gupta S, Singh S (2018) Computer aided thyroid nodule detection system using medical ultrasound images. *Biomed Signal Process Control* 40:117–130
- Perera P, Patel VM (2019) Learning deep features for one-class classification. *IEEE Trans Image Process* 28(11):5450–5463
- Zhu F, Yang J, Gao C et al (2016) A weighted one-class support vector machine. *Neurocomputing* 189:1–10
- Scholkopf B, Platt JC, Shawe-Taylor J, Smola Alex J, Williamson RC (2001) Estimating the support of a high-dimensional distribution. *Neural Comput* 13(7):1443–1471
- Tax DMJ, Duin RPW (2004) Support vector data description. *Mach Learn* 54(1):45–66
- Zhu F, Ye N, Yu W et al (2014) Boundary detection and sample reduction for one-class support vector machines. *Neurocomputing* 123:166–173
- Zhu F, Yang J, Xu S et al (2016) Relative density degree induced boundary detection for one-class SVM. *Soft Comput* 20(11):4473–4485
- Landgrebe T, Paclík P, Tax DMJ et al (2005) Optimising two-stage recognition systems. *International workshop on multiple classifier systems*. Springer, Berlin, pp 206–215
- Tax DMJ, Duin RPW (2008) Growing a multi-class classifier with a reject option. *Pattern Recognit Lett* 29(10):1565–1570
- Zhu F, Ning Y, Chen X et al (2021) On removing potential redundant constraints for SVOR learning. *Appl Soft Comput* 102:106941
- Zhu F, Gao J, Yang J et al (2021) Neighborhood linear discriminant analysis. *Pattern Recognit* 123:108422
- Ruff L, Vandermeulen R, Goernitz N, et al (2018) Deep one-class classification. *International conference on machine learning*. pp 4393–4402
- Iosifidis A, Mygdalis V, Tefas A et al (2017) One-class classification based on extreme learning and geometric class information. *Neural Process Lett* 45(2):577–592
- Zhang L, Xiang T, Gong S (2016) Learning a discriminative null space for person re-identification. *Proceedings of the IEEE conference on computer vision and pattern recognition*. pp 1239–1248
- Liu J, Lian Z, Wang Y, et al (2017) Incremental kernel null space discriminant analysis for novelty detection. *IEEE conference on computer vision and pattern recognition (CVPR)*. IEEE, pp 4123–4131
- Huang X, Xu J, Guo G (2018) Incremental kernel null Foley–Sammon transform for person re-identification. *2018 24th international conference on pattern recognition (ICPR)*. IEEE, pp 1683–1688
- T Ali M F, Chaudhuri S (2018) Maximum margin metric learning over discriminative null space for person re-identification. *Proceedings of the European conference on computer vision (ECCV)*. pp 122–138
- He X, Cai D, Yan S, et al (2005) Neighborhood preserving embedding. *Tenth IEEE international conference on computer vision (ICCV'05) volume 1(2)*. IEEE, pp 1208–1213
- Bodesheim P, Freytag A, Rodner E, et al (2015) Local novelty detection in multi-class recognition problems. *2015 IEEE winter conference on applications of computer vision*. IEEE, pp 813–820
- Chang CC, Lin CJ (2011) LIBSVM: a library for support vector machines. *ACM Trans Intell Syst Technol (TIST)* 2(3):1–27

33. Duan H, Sun Y, Cheng W et al (2021) Gesture recognition based on multi-modal feature weight. *Concurr Comput Pract Exp* 33(5):e5991
34. Zhang X, Liu J, Gao Q et al (2020) Adaptive robust decoupling control of multi-arm space robots using time-delay estimation technique. *Nonlinear Dyn* 100(3):2449–2467
35. Zhang X, Liu J, Feng J et al (2019) Effective capture of non-graspable objects for space robots using geometric cage pairs. *IEEE/ASME Trans Mechatron* 25(1):95–107
36. Yue W, Li C, Chen Y, et al (2021) What is the root cause of congestion in urban traffic networks: road infrastructure or signal control?. *IEEE Trans Intell Transport Syst*
37. Zhou C, Gu Y, Fan X et al (2018) Direction-of-arrival estimation for coprime array via virtual array interpolation. *IEEE Trans Signal Process* 66(22):5956–5971

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.