IEICE TRANS. INF. & SYST., VOL.E97–D, NO.10 OCTOBER 2014

PAPER

Fuzzy Multiple Subspace Fitting for Anomaly Detection

Raissa RELATOR[†a)], ***Nonmember***, **Tsuyoshi KATO**[†], ***Member***, **Takuma TOMARU**[†], ***Nonmember***, ***and*** **Naoya OHTA**[†], ***Member***

SUMMARY Anomaly detection has several practical applications in different areas, including intrusion detection, image processing, and behavior analysis among others. Several approaches have been developed for this task such as detection by classification, nearest neighbor approach, and clustering. This paper proposes alternative clustering algorithms for the task of anomaly detection. By employing a weighted kernel extension of the least squares fitting of linear manifolds, we develop fuzzy clustering algorithms for kernel manifolds. Experimental results show that the proposed algorithms achieve promising performances compared to hard clustering techniques.

key words: *fuzzy algorithm, subspace fitting, kernel vector subspace, kernel affine subspace, anomaly detection*

1. Introduction

Anomaly detection is the task of finding patterns within the data that do not conform to the norm. Depending on the specialization domain, the uncovered patterns are usually termed as anomalies, outliers, or novelty points. These often translate to some significant information about the data, which allows us to interpret the data or determine a suitable approach to handle it. Over the years, anomaly detection has gained interest from the scientific community due to its significance in practical applications such as intrusion detection, fraud detection, image processing, sensor networks, traffic networks, and behavior analysis [1]–[4].

Several techniques have been developed addressing anomaly detection problems, depending on their specific application [2], [4]. Some popular straightforward methods include employing classifiers, nearest neighbor-based techniques, and clustering [2], [4]–[6]. While classification is a supervised technique and requires known labels, and nearest neighbor-based classification performs better for semi-supervised than unsupervised, clustering may be more advantageous when the data is assumed to be free of anomaly, which is the case for most anomaly detection settings. In this study, clustering is not an objective but a means for fitting multiple simple units to a given dataset to represent a complicated normal class model, as depicted in Fig. 1. The focus is on fuzzy clustering for anomaly detection, in particular, algorithms are developed by applying weighted least squares fitting to update cluster models, and we discuss the case where only normal data are given.

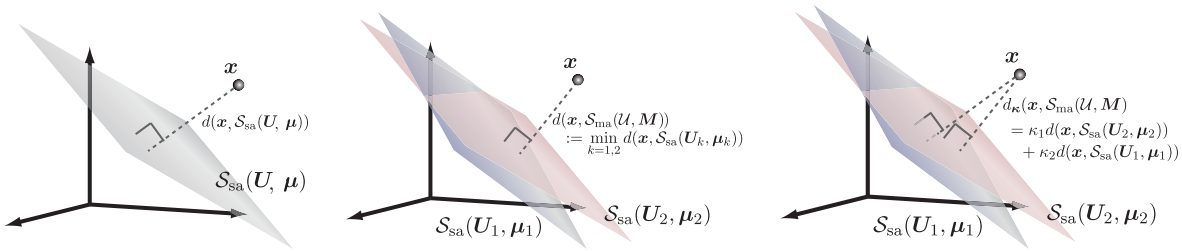
Least squares fitting of a linear manifold such as vector subspaces and affine subspaces have been utilized in a variety of applications such as pattern recognition, anomaly detection [2], [4], de-noising [7], data compression [8], and visualization [9]. However, linear manifolds cannot represent curved surfaces, and thus kernel methods have been often employed to address this issue, and other concerns such as handling high-dimensional data. Existing clustering methods related to manifold learning and kernel functions are summarized in Table 1, together with the contributions of this paper. The techniques for fitting vector subspaces and affine subspaces have been used for several decades [10] and were generalized to kernel methods in the late 1990s [11]–[13]. Ho et al. [14] developed a clustering method with vector subspaces, and Lu and Vidal [15] presented its affine subspace version. Ho et al.'s clustering method was kernelized by Li and Fukui [16] and extended to fuzzy clustering by Li et al. [17]. To develop the fuzzy clustering algorithm with vector subspaces, Li et al. devised an elementary technique called the weighted singular value decomposition (WSVD). The WSVD assumes that each entry in the design matrix is to be weighted, unlike in our setting where each input vector is assumed to be weighted. Also, the WSVD is an iterative algorithm and the theoretical guarantee for optimality is not given in their paper, whereas the algorithm presented in this paper is guaranteed to achieve the optimal solution. Although weighted PCA [18] has also been proposed, the deviation for the kernel version of the weighted least squares fitting of the manifold with more than one dimension has not been clearly described.

In this paper, we develop new fuzzy algorithms that learn multiple subspaces, and provide empirical evidence indicating promising performances of the proposed methods in the area of anomaly detection compared to other known algorithms. The rest of the paper is organized as follows. An overview of the problem is presented in Sect. 2. We then proceed to the development of the algorithms for fuzzy clustering of subspaces in Sects. 3 and 4. In Sect. 5, we give the details of the experimental results in applying the aforementioned algorithms to anomaly detection on real-world data including face images, sounds, and amino acid sequences, and provide a performance comparison with hard clustering. Finally, we conclude the paper in Sect. 6. Although all the algorithms presented in this paper are developed in the reproducing kernel Hilbert space, all discussions are de-

Manuscript received January 22, 2014.
Manuscript revised April 17, 2014.

[†]The authors are with the Graduate School of Science and Technology, Gunma University, Kiryu-shi, 376–8515 Japan.
a) E-mail: relator-raissa@kato-lab.cs.gunma-u.ac.jp

DOI: 10.1587/transinf.2014EDP7027



(a) Single subspace model. (b) Multiple subspace model with classical distance. (c) Multiple subspace model with κ distance.

Fig. 1 Normal class models. In this study, the normal class is modeled by a set of points $\mathcal{S}(\boldsymbol{\Theta})$. Classically, the anomalism of an input data point is examined with the distance to its projection onto the point set $\mathcal{S}(\boldsymbol{\Theta})$. Figure (a) describes a model that uses a single affine subspace $\mathcal{S}(\boldsymbol{\Theta}) = \mathcal{S}_{\rm sa}(\boldsymbol{U}, \boldsymbol{\mu})$. In (b), the normal class model is given by the union of two affine subspaces $\mathcal{S}(\boldsymbol{\Theta}) = \mathcal{S}_{\rm sa}(\boldsymbol{U}_1, \boldsymbol{\mu}_1) \cup \mathcal{S}_{\rm sa}(\boldsymbol{U}_2, \boldsymbol{\mu}_2)$. The classical distance to the set is the square Euclidean distance to the nearest subspace. In this study, we introduce the $\boldsymbol{\kappa}$-distance that is the linear combination of the distances with weights κ_1 and κ_2, as in (c).

Table 1 Clustering methods with linear manifolds.

	Vector Subspace	Kernel Vector Subspace	Affine Subspace	Kernel Affine Subspace
Single Cluster	Oja [10]	Tsuda [13]	Oja [10]	Maeda and Murase [12]
Hard Clustering	Ho et al. [14]	Li and Fukui [16]	Lu and Vidal [15]	New
Fuzzy Clustering	(Li et al. [17])	New	New	New

scribed in Euclidean space for the sake of readability. We underline that all the proposed algorithms can be kernelized straightforwardly with the kernel trick [19].

Notation. We denote vectors by bold-faced lower-case letters and matrices by bold-faced upper-case letters. Entries of vectors and matrices are not bold-faced. The transpose of a matrix $\boldsymbol{A}$ is denoted by $\boldsymbol{A}^\mathsf{T}$, and the inverse of $\boldsymbol{A}$ by $\boldsymbol{A}^{-1}$. The $n \times n$ identity matrix is denoted by $\boldsymbol{I}_n$. The n-dimensional vector whose entries are all one is denoted by $\boldsymbol{1}_n$. We use $\mathbb{R}$ and $\mathbb{N}$ to denote the set of real and natural numbers, $\mathbb{R}^n$ and $\mathbb{N}^n$ to denote the set of n-dimensional real and natural vectors, and $\mathbb{R}^{m\times n}$ to denote the set of $m \times n$ real matrices. The set of real nonnegative numbers is represented by $\mathbb{R}_+$. For any $n \in \mathbb{N}$, we use $\mathbb{N}_n$ to denote the set of natural numbers less than or equal to n. We use $\mathbb{S}^n$ to denote the set of $n \times n$ symmetric matrices, and $\mathbb{O}^{m\times n}$ the set of $m \times n$ orthonormal matrices, i.e. $\mathbb{O}^{m\times n} \equiv \{\boldsymbol{A} \in \mathbb{R}^{m\times n} \,|\, \boldsymbol{A}^\mathsf{T}\boldsymbol{A} = \boldsymbol{I}_n\}$. This definition implies that $\mathbb{O}^{m\times n} = \emptyset$ if $m < n$. A *permutation* of the set $\mathbb{N}_n$ is a bijective map from $\mathbb{N}_n$ to $\mathbb{N}_n$. We use $\mathcal{P}_n$ to denote the family of permutations of $\mathbb{N}_n$. The n-dimensional probabilistic simplex is denoted by $\boldsymbol{\Delta}_n \equiv \{\boldsymbol{x} \in \mathbb{R}^n_+ \,|\, \boldsymbol{x}^\mathsf{T}\boldsymbol{1}_n = 1\}$.

2. Problem Setting

Anomaly detection is a task of learning from a set of normal examples. Given ℓ normal examples $\boldsymbol{x}_i \in \mathbb{R}^d\, (i \in \mathbb{N}_\ell)$, an anomalous pattern is detected if it is distant from the normal class model learned using the examples. This section focuses on the normal class model which is given by a set of points in a d-dimensional space, say $\mathcal{S}(\boldsymbol{\Theta}) \subset \mathbb{R}^d$, where $\boldsymbol{\Theta}$ is the set of model parameters. If $\boldsymbol{x} \in \mathbb{R}^d$ is an unknown input vector, its square Euclidean distance to the model $\mathcal{S}(\boldsymbol{\Theta})$ is given by

$$d_{\rm euc}(\boldsymbol{x}, \mathcal{S}(\boldsymbol{\Theta})) \equiv \min_{\boldsymbol{y}\in\mathcal{S}(\boldsymbol{\Theta})} \|\boldsymbol{x} - \boldsymbol{y}\|^2.$$

This provides the confidence level for the anomalism of $\boldsymbol{x}$, where larger distances yield higher confidence levels. Instead of the square Euclidean distance, one may also use the Mahalanobis distance [20] or the distance with a learnable Gram matrix [21]. However, these are not rotationally invariant, thus prohibiting us from kernelizing them.

3. Distance to Normal Class Models

We now introduce four normal class models wherein values of the parameter $\boldsymbol{\Theta}$ are determined such that the mean square deviation of the training examples given by

$$J(\boldsymbol{\Theta}) \equiv \frac{1}{\ell}\sum_{i=1}^{\ell} d_{\rm euc}(\boldsymbol{x}_i, \mathcal{S}(\boldsymbol{\Theta})) \tag{1}$$

is minimized in the conventional method.

Single Vector Subspace Model. When we employ the vector subspace with orthonormal bases $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_m$, the normal class model is given by

$$\mathcal{S}(\boldsymbol{\Theta}) = \mathcal{S}_{\rm ss}(\boldsymbol{U}) \equiv \{\boldsymbol{y} \in \mathbb{R}^d \,|\, \exists \boldsymbol{\alpha} \in \mathbb{R}^m \text{ s.t. } \boldsymbol{y} = \boldsymbol{U}\boldsymbol{\alpha}\},$$

and $\boldsymbol{\Theta} = \boldsymbol{U} \equiv [\boldsymbol{u}_1, \ldots, \boldsymbol{u}_m] \in \mathbb{O}^{d\times m}$.

Single Affine Subspace Model. The normal class model using a single affine subspace is described as

$$\mathcal{S}(\boldsymbol{\Theta}) = \mathcal{S}_{\rm sa}(\boldsymbol{U}, \boldsymbol{\mu}) \equiv \{\boldsymbol{y} \in \mathbb{R}^d \,|\, \exists \boldsymbol{\alpha} \in \mathbb{R}^m \text{ s.t. } \boldsymbol{y} - \boldsymbol{\mu} = \boldsymbol{U}\boldsymbol{\alpha}\},$$

where $\boldsymbol{\Theta} = \{\boldsymbol{U}, \boldsymbol{\mu}\}$, $\boldsymbol{U} \in \mathbb{O}^{d\times m}$ and $\boldsymbol{\mu} \in \mathbb{R}^d$.

Vector Subspace Set Model. This model is defined by a set of L vector subspaces:

$$\mathcal{S}(\boldsymbol{\Theta}) = \mathcal{S}_{\mathrm{ms}}(\mathcal{U}) \equiv \bigcup_{k=1}^{L} \mathcal{S}_{\mathrm{ss}}(\boldsymbol{U}_k),$$

where $\mathcal{U} \equiv \{\boldsymbol{U}_k\}_{k\in\mathbb{N}_L}$ and $\boldsymbol{\Theta} = \mathcal{U}$. The clustering method of [14] finds a local optimum of $\mathcal{U}$. Each subspace is expected to represent a cluster of normal class data.

Affine Subspace Set Model. The normal class model can also be defined by a set of L affine subspaces,

$$\mathcal{S}(\boldsymbol{\Theta}) = \mathcal{S}_{\mathrm{ma}}(\mathcal{U}, \boldsymbol{M}) \equiv \bigcup_{k=1}^{L} \mathcal{S}_{\mathrm{sa}}(\boldsymbol{U}_k, \boldsymbol{\mu}_k),$$

where $\mathcal{U} = \{\boldsymbol{U}_k\}_{k\in\mathbb{N}_L}$, $\boldsymbol{M} \equiv [\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_L]$, and $\boldsymbol{\Theta} = \{\mathcal{U}, \boldsymbol{M}\}$. A local optimum of $\boldsymbol{\Theta}$ is found using a trick similar to the K-means method, as in the algorithm of Lu and Vidal [15].

4. Learning Normal Class Models

In this section, we present how to determine the model parameter $\boldsymbol{\Theta}$. First, a conventional method, Hard Clustering, is described in 4.1, and then two Fuzzy Clustering Methods are presented in 4.2 and 4.3.

4.1 Hard Clustering

When using a vector or an affine subspace set (i.e. $\mathcal{S}(\boldsymbol{\Theta}) = \mathcal{S}_{\mathrm{ms}}(\boldsymbol{U})$ or $\mathcal{S}(\boldsymbol{\Theta}) = \mathcal{S}_{\mathrm{ma}}(\boldsymbol{U}, \boldsymbol{\mu})$), the distance from the model can be expressed as

$$d_{\mathrm{euc}}(\boldsymbol{x}, \mathcal{S}(\boldsymbol{\Theta})) = \min_{k\in\mathbb{N}_L} d_{\mathrm{euc}}(\boldsymbol{x}, \mathcal{S}_k(\boldsymbol{\Theta}_k)), \tag{2}$$

where $\mathcal{S}_k(\boldsymbol{\Theta}_k)$ is the kth cluster model with parameter $\boldsymbol{\Theta}_k$. Meanwhile, the cluster model is given by $\mathcal{S}_k(\boldsymbol{\Theta}_k) = \mathcal{S}_{\mathrm{ss}}(\boldsymbol{U}_k)$ with $\boldsymbol{\Theta}_k = \{\boldsymbol{U}_k\}$, when a single vector subspace is used, and $\mathcal{S}_k(\boldsymbol{\Theta}_k) = \mathcal{S}_{\mathrm{sa}}(\boldsymbol{U}_k, \boldsymbol{\mu}_k)$ with $\boldsymbol{\Theta}_k = \{\boldsymbol{U}_k, \boldsymbol{\mu}_k\}$ when using a single affine subspace. The mean square deviation (1) is rewritten as

$$J(\boldsymbol{\Theta}) = \frac{1}{\ell} \sum_{i=1}^{\ell} \min_{k_i\in\mathbb{N}_L} d_{\mathrm{euc}}(\boldsymbol{x}_i, \mathcal{S}_{k_i}(\boldsymbol{\Theta}_{k_i})).$$

This implies that minimizing $J(\boldsymbol{\Theta})$ with respect to $\boldsymbol{\Theta}$ is equivalent to finding the optimal partitions of training examples, index sets $\{\mathcal{I}_k\}_{k=1}^{L}$, such that $\bigcup_{k=1}^{L} \mathcal{I}_k = \mathbb{N}_\ell$, and

$$\forall k, \forall k' \in \mathbb{N}_L \text{ s.t. } k \neq k' : \quad \mathcal{I}_k \cap \mathcal{I}_{k'} = \emptyset.$$

If $\{\mathcal{I}_k^*\}_{k=1}^{L}$ is the optimal partition, then

$$\min_{\boldsymbol{\Theta}} J(\boldsymbol{\Theta}) = \frac{1}{\ell} \sum_{k=1}^{L} \min_{\boldsymbol{\Theta}_k} \sum_{i\in\mathcal{I}_k^*} d_{\mathrm{euc}}(\boldsymbol{x}_i, \mathcal{S}_k(\boldsymbol{\Theta}_k)),$$

which elucidates that, in learning the model parameters $\boldsymbol{\Theta}$, each example contributes to only one out of L cluster models. That is, training examples are not shared among cluster models.

A self-organizing map (SOM) [22] is a clustering model in which clusters share training examples. Usually, SOM is designed so that each cluster is put on the grid in advance, and each example is shared with the clusters near to the winner cluster in the grid. The cluster models of SOM are, thereby, learned as a 'single connected component.' However, clusters in real world data are not always 'connected'. Motivated by this observation, similarly to fuzzy c-means method [23], we now propose two fuzzy algorithms that learn multiple subspaces. In the algorithms, cluster models can share examples, but do not necessarily have to be 'connected.'

4.2 Fuzzy Clustering with κ-Distance

Using the set of predefined weights $\boldsymbol{\kappa} \in \mathbb{R}_+^L$ of the clusters, such that

$$\kappa_1 \geq \kappa_2 \geq \cdots \geq \kappa_L \geq 0, \tag{3}$$

we replace the distance in (2) with a linear combination of L distances with weights in $\boldsymbol{\kappa}$,

$$d_{\boldsymbol{\kappa}}(\boldsymbol{x}, \mathcal{S}(\boldsymbol{\Theta})) = \min_{\boldsymbol{\pi}\in\mathcal{P}_L} \sum_{k=1}^{L} \kappa_{\pi(k)} d_{\mathrm{euc}}(\boldsymbol{x}, \mathcal{S}_{\pi(k)}(\boldsymbol{\Theta}_{\pi(k)})), \tag{4}$$

during learning, where $\mathcal{P}_L$ is the set of permutations on the index set $\mathbb{N}_L$ and $\pi(k)$ returns the index value where $\boldsymbol{\pi} \in \mathcal{P}_L$ maps k. Here, we redefine the objective function $J(\boldsymbol{\Theta})$ (defined previously in (1)) by replacing d_{euc} with $d_{\boldsymbol{\kappa}}$. We refer to this distance as the *κ-distance*. Since the (4) is equal to (2) when

$$\kappa_k = \begin{cases} 1 & \text{for } k = 1, \\ 0 & \text{for } k \geq 2, \end{cases}$$

the κ-distance is a generalization of the square Euclidean distance in (2). From the assumption in (3), if $\boldsymbol{\pi}^*$ is the optimal permutation, then

$$d_{\mathrm{euc}}(\boldsymbol{x}, \mathcal{S}_{\pi^*(1)}(\boldsymbol{\Theta}_{\pi^*(1)})) \leq \cdots \leq d_{\mathrm{euc}}(\boldsymbol{x}, \mathcal{S}_{\pi^*(L)}(\boldsymbol{\Theta}_{\pi^*(L)})).$$

Using the optimal permutation $\boldsymbol{\pi}_i^* \ \forall i \in \mathbb{N}_\ell$, the minimum of the mean square deviation is expressed as

$$\min_{\boldsymbol{\Theta}} J(\boldsymbol{\Theta}) = \frac{1}{\ell} \sum_{k=1}^{L} \min_{\boldsymbol{\Theta}_k} \sum_{i=1}^{\ell} v_{k,i} d_{\mathrm{euc}}(\boldsymbol{x}_i, \mathcal{S}_k(\boldsymbol{\Theta}_k)), \tag{5}$$

where $v_{k,i}$ is defined as

$$\forall i \in \mathbb{N}_\ell, \ \forall k \in \mathbb{N}_L : \quad v_{\pi^*(k),i} \equiv \kappa_k. \tag{6}$$

The kth cluster model is learned from every training example with weight $v_{k,i} > 0$, hence, training examples can be shared with multiple cluster models, resulting to 'naturally connected' cluster models.

The block coordinate ascent method [24] for minimizing $J(\boldsymbol{\Theta})$ with respect to $\boldsymbol{\Theta}$ is given in the following algorithm.

Algorithm 1. *Alternate the following steps until convergence.*

- *Step 1: For each example* $\boldsymbol{x}_i$, $i \in \mathbb{N}_\ell$, *update the permutation such that* $\forall k \in \mathbb{N}_{L-1}$:

$$\begin{aligned} &d_{\mathrm{euc}}(\boldsymbol{x}_i, \mathcal{S}_{\pi^{(t+1)}(k)}(\boldsymbol{\Theta}_{\pi^{(t+1)}(k)})) \\ \leq\ &d_{\mathrm{euc}}(\boldsymbol{x}_i, \mathcal{S}_{\pi^{(t+1)}(k+1)}(\boldsymbol{\Theta}_{\pi^{(t+1)}(k+1)})) \end{aligned}$$

  by sorting the L distances and compute $v_{k,i}^{(t+1)}$ *using* (6).
- *Step 2: For each cluster model, update the parameters as:* $\forall k \in \mathbb{N}_L$,

$$\boldsymbol{\Theta}_k^{(t+1)} := \arg\min_{\boldsymbol{\Theta}_k} \sum_{i=1}^{\ell} v_{k,i}^{(t+1)} d_{\mathrm{euc}}(\boldsymbol{x}_i, \mathcal{S}_k(\boldsymbol{\Theta}_k)).$$

The definition of the algorithm guarantees that the sequence $\{J(\boldsymbol{\Theta}^{(t)})\}_{t\in\mathbb{N}}$ is monotonically decreasing. Furthermore, the values of model parameters become unchanged within a finite number of iterations since the cardinality of the permutation set $\mathcal{P}_L$ is finite.

In updating the model parameters in Step 2, Theorem 1 and Corollary 2 (in Appendix) are applied to each cluster model to update $\boldsymbol{\Theta}_k$ for the vector subspace set model. For the affine subspace set model, Theorem 4 and Corollary 5 (in Appendix) are employed.

4.3 Fuzzy Clustering with Bezdek Distance

Another approach to sharing the training examples with multiple clusters is to extend the square Euclidean distance (2) to the *Bezdek distance* [23] defined by

$$d_{\mathrm{bez}}(\boldsymbol{x}, \mathcal{S}(\boldsymbol{\Theta})) = \min_{\boldsymbol{w}\in\boldsymbol{\Delta}_L} \sum_{k=1}^{L} w_k^{b_{\mathrm{bez}}} d_{\mathrm{euc}}(\boldsymbol{x}, \mathcal{S}_k(\boldsymbol{\Theta}_k)), \tag{7}$$

where b_{bez} (≥ 1) is constant. A similar distance is used in fuzzy c-means method [23] and is equivalent to (2) when $b_{\mathrm{bez}} = 1$. If $b_{\mathrm{bez}} > 1$, the optimal weight $\boldsymbol{w}^*$ can be derived using the method of Lagrange multipliers, and the kth entry is given by

$$w_k^* \propto \frac{1}{(d_{\mathrm{euc}}(\boldsymbol{x}, \mathcal{S}_k(\boldsymbol{\Theta}_k)))^{1/(b_{\mathrm{bez}}-1)}}. \tag{8}$$

The clustering algorithm using the Bezdek distance is given as follows.

Algorithm 2. *Alternate the following steps until convergence.*

- *Step 1: For each example* $\boldsymbol{x}_i$, $i \in \mathbb{N}_\ell$, *update weight* $\boldsymbol{w}_i^{(t+1)}$ *using* (8)*, and set*

$$\forall k \in \mathbb{N}_L,\ \forall i \in \mathbb{N}_\ell :\ v_{k,i}^{(t+1)} = ([\boldsymbol{w}_i^{(t+1)}]_k)^{b_{\mathrm{bez}}}.$$

- *Step 2: For each cluster model, update the parameters as:* $\forall k \in \mathbb{N}_L$,

$$\boldsymbol{\Theta}_k^{(t+1)} := \arg\min_{\boldsymbol{\Theta}_k} \sum_{i=1}^{\ell} v_{k,i}^{(t+1)} d_{\mathrm{euc}}(\boldsymbol{x}_i, \mathcal{S}_k(\boldsymbol{\Theta}_k)).$$

Fig. 2 Examples of face images. (a) Normal data. (b) Anomalous data. (c) Anomalous data that were detected by MA-κC, but were not detected by other existing methods when the specificity is 0.95. (d) Normal data that was not detected as an anomaly by MA-κC, but falsely detected by existing methods.

Algorithm 2 is formed such that the sequence $\{J(\boldsymbol{\Theta}^{(t)})\}_{t\in\mathbb{N}}$ is decreasing monotonically, where $J(\boldsymbol{\Theta})$ (firstly defined in (1)) is redefined by changing d_{euc} to d_{bez}. Moreover, from the definition of $J(\cdot)$, the sequence is non-negative. Hence, the algorithm must converge.

5. Experiments and Results

To investigate the anomaly detection performance of each model presented in the previous section, experiments on face images, sounds, and string patterns were conducted.

The face images data are from the Extended Yale Face Database B which contains 2,350 192×168 gray-scaled images of 39 people. The images of the first three people are used as normal data, and the remaining images are regarded as anomalous data (See Fig. 2 (a),(b)), yielding 192 normal images and 2,158 anomalous images. The kernel function defined by $K(\boldsymbol{I}, \boldsymbol{I}') = \mathrm{tr}(\boldsymbol{I}^{\mathsf{T}}\boldsymbol{I}')$, for any $\boldsymbol{I}, \boldsymbol{I}' \in \mathbb{R}^{192\times168}$, is used.

For the second data, sounds are recorded in a bathroom and the power spectra are extracted to obtain 256-dimensional input vectors. The linear kernel is used to obtain the kernel values. This dataset contains $1,406$ normal data and 94 anomalous data.

Lastly, for string patterns, we used $3,427$ amino-acid sequences in 12 folds classified in the protein structure database SCOP [25]. The sequences in the first four folds are assumed to be in the normal class, and the remaining data in the anomalous class, yielding 909 normal data and $2,518$ anomalous data. Kernel values are obtained via a string kernel by Lodhi et al. [26] that are capable of efficient inner product computation without explicit extraction of very high-dimensional input vectors.

Eighty percent of the normal data are randomly selected and used as training data, while the remaining normal data together with the anomalous data are used for performance evaluation. For the single vector and single affine subspace models, we determine the number of dimensions m of the manifold so that m is the maximum dimension for which the ratio of the cumulative variances is below 0.95. To obtain the value of m for the subspace set models, we performed single-linkage clustering to divide the training data

Table 2 Performances on anomaly detections.

(a) Face Images

	SS	MS-κC	MS-BC	MS-HC	SA	MA-κC	MA-BC	MA-HC
Highest	0.757 (0.034)	0.954 (0.010)	0.952 (0.004)	0.921 (0.018)	0.835 (0.028)	**0.955** (0.007)	0.950 (0.007)	0.911 (0.040)
Average	0.757 (0.034)	0.924 (0.018)	0.918 (0.020)	0.842 (0.026)	0.835 (0.028)	**0.936** (0.012)	0.930 (0.008)	0.860 (0.025)

(b) Sounds

	SS	MS-κC	MS-BC	MS-HC	SA	MA-κC	MA-BC	MA-HC
Highest	0.846 (0.019)	0.955 (0.007)	0.944 (0.004)	0.964 (0.004)	0.846 (0.019)	**0.993** (0.004)	**0.993** (0.001)	0.990 (0.006)
Average	0.846 (0.019)	0.911 (0.014)	0.943 (0.004)	0.908 (0.015)	0.846 (0.019)	0.988 (0.005)	**0.990** (0.001)	0.976 (0.004)

(c) Strings

	SS	MS-κC	MS-BC	MS-HC	SA	MA-κC	MA-BC	MA-HC
Highest	0.811 (0.014)	0.838 (0.017)	0.841 (0.017)	0.813 (0.015)	0.813 (0.014)	0.836 (0.017)	**0.845** (0.017)	0.814 (0.015)
Average	0.811 (0.014)	0.836 (0.016)	0.834 (0.015)	0.811 (0.015)	0.813 (0.014)	0.831 (0.017)	**0.838** (0.016)	0.813 (0.015)

into the predetermined number of clusters and searched for the maximum number of dimensions for which the ratio of the cumulative variances is below 0.95 in any cluster. We varied the number of clusters with $L = 10, 20, 30$. The weights $\boldsymbol{\kappa}$ in the κ-distance are set as $\kappa_1 = 0.9, \kappa_2 = 0.1$, and $\kappa_3 = 0$, while the parameter of the Bezdek distance is set to $b_{\text{bez}} = 2$.

We tested eight methods: SS, MS-κC, MS-BC, MS-HC, SA, MA-κC, MA-BC, and MA-HC. SS and SA, respectively, are the single vector and the single affine subspace models, while MS and MA correspond to the vector and the affine subspace set models. We have κC, BC, and HC indicating the types of learning methods, where κC and BC are clustering methods with the κ-distance and the Bezdek distance, and HC is classical hard clustering such as the K-means method. The square Euclidean distance is used in the prediction stage, while the κ-distance or the Bezdek distance is employed during learning. Accordingly, MS-κC, MA-κC, MS-BC, and MA-BC are the new methods proposed in this paper. Furthermore, to the best of our knowledge, no work uses a kernel version of MA-HC (See Table 1).

The detection performances are summarized in Table 2. The values are the area under the ROC curve (AUC) where 'Highest' refers to the maximum among the three AUCs obtained when $L = 10, 20, 30$, and 'Average' is their mean. Four random partitions are made dividing the data into training and test sets. The average and the standard deviation of the AUCs are used to determine the performance quality given Table 2. We employed the one-sample t-test to detect statistical significance of the differences among the detection performances, and set the significance level to 0.01. In Table 2, bold-faced figures represent the best AUC, and underlined figures indicate performances with no significant difference from the best AUC.

In the first experiment using face images, MA-κC achieves the best performance when the Highest values are considered in all methods. The Highest AUC of MA-HC is 0.044 lower than that of MA-κC, however, they are not significantly different since standard deviation for MA-HC is large (0.040). The Highest AUC of MA-κC is also the best among the eight methods. In general, the values of the hyper-parameters, such as the number of clusters, can be determined by using cross-validation in supervised learning, although the use of the cross validation method is not easy in the scenario of anomaly detection. For MA-HC, Average is 0.051 lower than Highest, which is larger compared to the difference between Average and Highest in MA-κC (0.019). For MS-κC, MS-BC, and MA-BC, respectively, Average and Highest values differ by 0.030, 0.034, and 0.020, while the difference is 0.059 for MS-HC. This suggests that performances using fuzzy clustering do not change drastically with different number of clusters compared to the hard clustering. Similar observations are derived using the dataset of sounds, while MA-BC exceeded the performance of other methods using the string patterns.

For MS-κC, MS-BC, MA-κC, and MA-BC, the results presented in Table 2 make use of the square Euclidean distance in prediction, and the fuzzy distances — the κ-distance and the Bezdek distance — in learning, as described previously. Experiments using the generalized distances, $d_{\boldsymbol{\kappa}}$ and d_{bez}, both in learning and in prediction are also done. The prediction performances when the square Euclidean distance d_{euc} is used with the manifold learned using the generalized distances $d_{\boldsymbol{\kappa}}$ and d_{bez} are slightly better than the performances of the methods using the κ-distance $d_{\boldsymbol{\kappa}}$ and Bezdek distance d_{bez}.

Kernel methods are often employed to obtain nonlinear learning machines. Any kernel in our experiments enjoys no nonlinear effect, although the pre-computed kernel values can be transformed easily to nonlinear kernels: $(K(\boldsymbol{x}, \boldsymbol{x}') + c)^p$ for polynomial kernel, and $\exp(-\gamma(K(\boldsymbol{x}, \boldsymbol{x}) + K(\boldsymbol{x}', \boldsymbol{x}') - 2K(\boldsymbol{x}, \boldsymbol{x}')))$ for RBF kernel, where c, p, and γ are constants. We used these tricks to test the nonlinear kernels, but no improvements were exhibited.

We also applied the one-class SVM to anomaly detection for performance comparison, employing both linear and RBF kernels. Similar settings as above were employed while the value of the parameter ν of the one-class SVM was varied as $0.1, 0.2, \ldots, 0.7$. We report here the best AUC values among all trials for each dataset. For all datasets, Highest AUC is obtained using RBF kernel: 0.868 for the face images, 0.890 for the sound data, and 0.836 for the strings data, and their respective Averages are given by 0.751, 0.867, and 0.789. While the Highest AUC value in the last experiment using string patterns has no significant difference from the best AUC obtained using the proposed algorithm according to statistical tests performed, comparing the said values to the those from the proposed algorithms

as presented in Table 2 reveals that the proposed methods outperform one-class SVM.

6. Conclusions

We have developed fuzzy clustering algorithms with vector and affine subspaces in a reproducing kernel Hilbert space. The performances of the algorithms in detecting anomalous patterns proved to be stable against using different number of clusters. Utilization of fuzzy multiple subspace fitting is not limited to anomaly detection, so our technique can also be applied to several other tasks. We consider de-noising as one promising application. Future work includes performance evaluation on other applications.

Acknowledgments

The work of TK is supported by MEXT KAKENHI Grant number 23500373. RR is supported by the Ministry of Education, Culture, Sports, Science and Technology of Japan.

References

[1] P. Chan, M. Mahoney, and M. Arshad, "Learning rules and clusters for anomaly detection in network traffic," Managing Cyber Threats: Issues, Approaches and Challenges, pp.81–99, 2005.
[2] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," ACM Comput. Surv., vol.41, pp.15:1–15:58, July 2009.
[3] J.F. Nieves, "Data clustering for anomaly detection in network intrusion detection," Research Alliance in Math and Science, pp.1–12, Aug. 2009.
[4] A. Patcha and J.M. Park, "An overview of anomaly detection techniques: Existing solutions and latest technological trends," Comput. Netw., vol.51, pp.3448–3470, Aug. 2007.
[5] M. Markou and S. Singh, "Novelty detection: A review – part 1," Signal Processing, vol.83, pp.2481–2497, 2003.
[6] M. Markou and S. Singh, "Novelty detection: A review – part 2," Signal Processing, vol.83, pp.2499–2521, 2003.
[7] S. Mika, B. Schölkopf, A. Smola, K.R. Müller, M. Scholz, and G. Rätsch, "Kernel PCA and de-noising in feature spaces," Proc. 1998 Conference on Advances in Neural Information Processing Systems II, pp.536–542, Cambridge, MA, USA, 1999.
[8] A. Abadpour and S. Kasaei, "Color PCA eigenimages and their application to compression and watermarking," Image Vision Comput., vol.26, pp.878–890, July 2008.
[9] C.M. Bishop and M.E. Tipping, "A hierarchical latent variable model for data visualization," IEEE Trans. Pattern Anal. Mach. Intell., vol.20, pp.281–293, March 1998.
[10] E. Oja, "New aspects on the subspace methods of pattern recognition," in Electron. Electr. Eng. Res. Stud. Pattern Recognition and Image Processing Ser. 5, pp.55–64, Letchworth, UK, 1984.
[11] B. Schölkopf, A. Smola, and K.R. Müller, "Nonlinear component analysis as a kernel eigenvalue problem," Neural Computation, vol.10, no.5, pp.1299–1319, 1998.
[12] E. Maeda and H. Murase, "Multi-category classification by kernel based nonlinear subspace method," Proc. 1999 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), pp.1025–1028, 1999.
[13] K. Tsuda, "Subspace classifier in the Hilbert space," Pattern Recognit. Lett., vol.20, pp.513–519, May 1999.
[14] J. Ho, M.H. Yang, J. Lim, K.C. Lee, and D. Kriegman, "Clustering appearances of objects under varying illumination conditions," CVPR, pp.11–18, 2003.
[15] L. Lu and R. Vidal, "Combined central and subspace clustering for computer vision applications," Proc. 23rd International Conference on Machine Learning, ICML '06, pp.593–600, New York, NY, USA, 2006.
[16] X. Li and K. Fukui, "Nonlinear k-subspaces based appearances clustering of objects under varying illumination conditions," ACCV Workshop Subspace 2007, pp.46–52, 2007.
[17] X. Li, Z. Ning, and L. Xiang, "Robust multi-body motion segmentation based on fuzzy k-subspace clustering," IEICE Trans. Inf. & Syst., vol.E88-D, no.11, pp.2609–2614, Nov. 2005.
[18] C. Alzate and J.A.K. Suykens, "Multiway spectral clustering with out-of-sample extensions through weighted kernel PCA," IEEE Trans. Pattern Anal. Mach. Intell., vol.32, pp.335–347, Feb. 2010.
[19] B. Schölkopf and A.J. Smola, Learning with kernels, MIT Press, Cambridge, MA, 2002.
[20] G.R. Lanckriet, L.E. Ghaoui, and M.I. Jordan, "Robust novelty detection with single-class MPM," In advances in neural information processing systems, pp.905–912, MIT Press, 2003.
[21] K.Q. Weinberger and L.K. Saul, "Fast solvers and efficient implementations for distance metric learning," Proc. 25th International Conference on Machine Learning, ICML '08, New York, NY, USA, pp.1160–1167, 2008.
[22] T. Kohonen, M.R. Schroeder, and T.S. Huang, eds., Self-Organizing Maps, 3rd ed., Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2001.
[23] R.O. Duda, P.E. Hart, and D.G. Stork, Pattern Classification, Second Edition, John Wiley & Sons, 2001.
[24] J.M. Ortega and W.C. Rheinboldt, Iterative Solution of Nonlinear Equations in Several Variables, Academic Press, New York, NY, 1970.
[25] A. Andreeva, D. Howorth, S.E. Brenner, T.J.P. Hubbard, C. Chothia, and A.G. Murzin, "SCOP database in 2004: refinements integrate structure and sequence family data," Nuclear Acid Research, vol.32, pp.D226–D229, 2004.
[26] H. Lodhi, C. Saunders, J. Shawe-Taylor, N. Cristianini, and C. Watkins, "Text classification using string kernels," J. Mach. Learn. Res., vol.2, pp.419–444, March 2002.

Appendix A: Fitting of Subspaces

Weighted least square fitting of cluster component is a necessary step in fuzzy clustering. In this Appendix we give algorithms for weighted least squares fitting of vector and affine subspaces in a kernelizable manner (See Fig. A· 1).

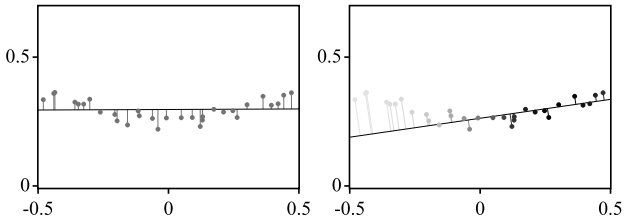
Fitting of Vector Subspaces. The square Euclidean distance from any input vector $\boldsymbol{x} \in \mathbb{R}^d$ to a linear subspace $\mathcal{S}_{\rm ss}(\boldsymbol{U})$ is given by

$$d(\boldsymbol{x}, \mathcal{S}_{\rm ss}(\boldsymbol{U})) = \min_{\boldsymbol{y}\in\mathcal{S}_{\rm ss}(\boldsymbol{U})} \|\boldsymbol{x}-\boldsymbol{y}\|^2 = \left\|(\boldsymbol{I}_d - \boldsymbol{U}\boldsymbol{U}^{\mathsf{T}})\boldsymbol{x}\right\|^2. \quad (\text{A}\cdot 1)$$

For a training set $\{\boldsymbol{x}_i\}_{i=1}^{\ell}$, it is known that a value of the parameter $\boldsymbol{U} \in \mathbb{O}^{d\times m}$ minimizing the unweighted mean square deviation is the matrix whose columns are the m major eigenvectors of

$$\sum_{i=1}^{\ell} \boldsymbol{x}_i \boldsymbol{x}_i^{\mathsf{T}}.$$

Our goal is to give a kernelizable solution to minimize the weighted mean square deviation



(a) Unweighted Least Squares (b) Weighted Least Squares

Fig. A· 1 Fitting of one-dimensional linear manifolds in a two-dimensional toy problem. Plots (a) and (b) are the fitting results using the unweighted least squares and the weighted least squares, respectively. The weights here are given by $\exp(-(x-0.3)^2/8)$. The darkness of each point indicates the value of the weight. It is observed in (b) that darker points tend to be nearer to the manifold.

$$J_{\mathrm{ss}}(\boldsymbol{U};\boldsymbol{v}) \equiv \frac{\sum_{i=1}^{\ell} v_i d(\boldsymbol{x}_i, \mathcal{S}_{\mathrm{ss}}(\boldsymbol{U}))}{\sum_{i=1}^{\ell} v_i}$$

with respect to $\boldsymbol{U}$, where $v_i \in \mathbb{R}_+$ is the weight of the ith example $\boldsymbol{x}_i$.

Let $\boldsymbol{X} \in \mathbb{R}^{d\times\ell}$ be the design matrix

$$\boldsymbol{X} \equiv [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_\ell].$$

For arbitrary vectors $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^d$, the function $K(\boldsymbol{x}, \boldsymbol{y})$ is given by

$$K(\boldsymbol{x}, \boldsymbol{y}) \equiv \langle \boldsymbol{x}, \boldsymbol{y} \rangle.$$

In kernel methods, various analyses are performed by rewriting the definition of $K(\boldsymbol{x}, \boldsymbol{y})$ in terms of a positive definite kernel. The inner-product matrix for the input data $\{\boldsymbol{x}_i\}_{i=1}^{\ell}$ is defined as the $\ell\times\ell$ matrix $\boldsymbol{K} \in \mathbb{S}^\ell$ whose entries are $K_{ij} \equiv K(\boldsymbol{x}_i, \boldsymbol{x}_j)$. Moreover, such matrices are always positive semidefinite. We also define the vector-valued function $\boldsymbol{k} : \mathbb{R}^d \mapsto \mathbb{R}^\ell$ as

$$\boldsymbol{k}(\boldsymbol{x}) \equiv [\langle \boldsymbol{x}_1, \boldsymbol{x} \rangle, \ldots, \langle \boldsymbol{x}_\ell, \boldsymbol{x} \rangle]^\mathsf{T}.$$

The following theorem gives the algorithm for the weighted least squares estimation of a vector subspace.

Theorem 1. *Define a symmetric matrix* $\tilde{\boldsymbol{K}} \equiv \boldsymbol{D}_v^{1/2}\boldsymbol{K}\boldsymbol{D}_v^{1/2}$. *Assume that* $r \geq m$, *where* $r \equiv \mathrm{rank}(\tilde{\boldsymbol{K}})$. *Denote the m major eigenvectors of* $\tilde{\boldsymbol{K}}$ *by* $\lambda_1, \ldots, \lambda_m \in \mathbb{R}_{++}$ *and the corresponding eigenvalues by* $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_m \in \mathbb{R}^\ell$. *Let* $\boldsymbol{\lambda} \equiv [\lambda_1, \ldots, \lambda_m]^\mathsf{T}$ *and* $\boldsymbol{B} \equiv [\boldsymbol{b}_1, \ldots, \boldsymbol{b}_m]$, *and define*

$$\hat{\boldsymbol{U}}_{\mathrm{ss}} \equiv \boldsymbol{X}\boldsymbol{D}_v^{1/2}\boldsymbol{B}\mathrm{diag}(\boldsymbol{\lambda})^{-1/2}. \tag{A· 2}$$

Then

$$\hat{\boldsymbol{U}}_{\mathrm{ss}} \in \operatorname*{arg\,min}_{\boldsymbol{U}\in\mathbb{O}^{d\times m}} J_{\mathrm{ss}}(\boldsymbol{U};\boldsymbol{v}).$$

Proof. The weighted mean deviation can be written as

$$J_{\mathrm{ss}}(\boldsymbol{U};\boldsymbol{v}) = \frac{1}{2}\mathrm{tr}(\boldsymbol{X}\boldsymbol{D}_{\boldsymbol{v}}\boldsymbol{X}^\mathsf{T}) - \frac{1}{2}\mathrm{tr}(\boldsymbol{U}^\mathsf{T}\boldsymbol{X}\boldsymbol{D}_{\boldsymbol{v}}\boldsymbol{X}^\mathsf{T}\boldsymbol{U}). \tag{A· 3}$$

Hence, the columns of the optimal $\boldsymbol{U}$ are the m major eigenvectors of a symmetric matrix $\boldsymbol{X}\boldsymbol{D}_{\boldsymbol{v}}\boldsymbol{X}^\mathsf{T}$. Moreover, each column lies in the span of the input vectors with positive weights. Therefore, there exists a matrix $\boldsymbol{A} \in \mathbb{R}^{\ell\times m}$ such that $\boldsymbol{U} = \boldsymbol{X}\boldsymbol{D}_v^{1/2}\boldsymbol{A}$. Substituting this to $J_{\mathrm{ss}}(\boldsymbol{U};\boldsymbol{v})$, the second term in (A· 3) becomes

$$\begin{aligned}
&\frac{1}{2}\mathrm{tr}(\boldsymbol{U}^\mathsf{T}\boldsymbol{X}\boldsymbol{D}_{\boldsymbol{v}}\boldsymbol{X}^\mathsf{T}\boldsymbol{U}) \\
=&\frac{1}{2}\mathrm{tr}(\boldsymbol{A}^\mathsf{T}\boldsymbol{D}_v^{1/2}\boldsymbol{X}^\mathsf{T}\boldsymbol{X}\boldsymbol{D}_{\boldsymbol{v}}\boldsymbol{X}^\mathsf{T}\boldsymbol{X}\boldsymbol{D}_v^{1/2}\boldsymbol{A}) \\
=&\frac{1}{2}\mathrm{tr}(\boldsymbol{A}^\mathsf{T}\tilde{\boldsymbol{K}}^2\boldsymbol{A})
\end{aligned}$$

From the orthonormality of $\boldsymbol{U}$, $\boldsymbol{A}$ must satisfy $\boldsymbol{A}^\mathsf{T}\tilde{\boldsymbol{K}}\boldsymbol{A} = \boldsymbol{I}_m$. Hence, the optimal matrix $\boldsymbol{A}$ is the solution of the following constraint optimization problem:

$$\begin{aligned}
&\max\ \mathrm{tr}(\boldsymbol{A}^\mathsf{T}\tilde{\boldsymbol{K}}^2\boldsymbol{A}) \\
&\quad \mathrm{wrt}\ \boldsymbol{A} \in \mathbb{R}^{\ell\times m} \\
&\mathrm{subj\ to}\ \boldsymbol{A}^\mathsf{T}\tilde{\boldsymbol{K}}\boldsymbol{A} = \boldsymbol{I}_m.
\end{aligned}$$

Furthermore, the columns of $\boldsymbol{A}$ should be proportional to the m major eigenvectors of $\tilde{\boldsymbol{K}}$. Thus, we have

$$\forall h \in \mathbb{N}_m,\ \exists t_h \in \mathbb{R}_+ :\ \boldsymbol{a}_h = t_h\boldsymbol{b}_h.$$

Now, the columns of $\boldsymbol{U}$ must be unit vectors. And since $\forall h \in \mathbb{N}_m$,

$$\begin{aligned}
1 = \|\boldsymbol{u}_h\|^2 &= \|\boldsymbol{X}\boldsymbol{D}_v^{1/2}t_h\boldsymbol{b}_h\|^2 \\
&= t_h^2\mathrm{tr}(\boldsymbol{X}\boldsymbol{D}_v^{1/2}\boldsymbol{b}_h\boldsymbol{b}_h^\mathsf{T}\boldsymbol{D}_v^{1/2}\boldsymbol{X}^\mathsf{T}) \\
&= t_h^2\mathrm{tr}(\boldsymbol{b}_h^\mathsf{T}\tilde{\boldsymbol{K}}\boldsymbol{b}_h) = t_h^2\lambda_h,
\end{aligned}$$

then $t_h = \lambda_h^{-1/2}$. Therefore, an optimal value of $\boldsymbol{u}_h$ is obtained such that

$$\forall h \in \boldsymbol{N}_m\ :\ \boldsymbol{u}_h = \lambda_h^{-1/2}\boldsymbol{X}\boldsymbol{D}_v^{1/2}\boldsymbol{b}_h,$$

and the conclusion follows. □

To kernelize the method, input vectors are embedded in the inner product. Nevertheless, the input vectors remain explicitly in the result of Theorem 1. Fortunately in many applications, what are important are the distances from arbitrary vectors to the manifold, and not the basis vectors in $\boldsymbol{U}$.

Corollary 2.

$$d(\boldsymbol{x}, \mathcal{S}_{\mathrm{ss}}(\hat{\boldsymbol{U}}_{\mathrm{ss}})) = K(\boldsymbol{x}, \boldsymbol{x}) - \left\|\mathrm{diag}(\boldsymbol{\lambda}')^{-1/2}\boldsymbol{A}\boldsymbol{D}_v^{1/2}\boldsymbol{k}(\boldsymbol{x})\right\|^2.$$

The equation holds by substituting the result of Theorem 1 into Eq. (A· 1).

Thus, distances to the vector subspace spanned by the columns of the matrix $\hat{\boldsymbol{U}}_{\mathrm{ss}}$ do not include any input vector explicitly, enabling us to kernelize the distance to the vector subspace even if the training data are weighted.

Note that the optimal solution is not unique since the set of eigenvectors of a symmetric matrix is not unique. For

instance, if $\boldsymbol{u}$ is an eigenvector corresponding to an eigenvalue λ, then so is the negative of $\boldsymbol{u}$. Hence, in our problem, $\boldsymbol{UR}$, where $\boldsymbol{R} \in \mathbb{O}^{m\times m}$, is also an optimal solution whenever $\boldsymbol{U}$ is an optimal solution. The set of optimal solutions are further expanded when the mth largest eigenvalue of $\tilde{\boldsymbol{K}}$ is equal to its $(m+1)$st largest eigenvalue.

We now build the relationship between Theorem 1 and some classical results. If inputs are given by vectors (e.g. inputs are not given by structured kernels), the non-kernelized method can be used to obtain the vector subspace linear to the input vectors. Furthermore, if $d \ll \ell$, the non-kernelized method is faster: The algorithm in Theorem 1 requires eigen-decomposition of an $\ell \times \ell$ symmetric matrix. The following result shows that the optimal basis vectors of $\boldsymbol{U}$ are also obtained by eigen-decomposition of a $d \times d$ matrix.

Corollary 3. *The m columns of $\hat{\boldsymbol{U}}_{\mathrm{ss}}$ defined in Theorem 1 are the m major eigenvectors of*

$$\boldsymbol{M} = \sum_{i=1}^{\ell} v_i \boldsymbol{x}_i \boldsymbol{x}_i^{\mathsf{T}}.$$

Any set of the m major eigenvectors of $\boldsymbol{M}$ minimizes $J_{\mathrm{ss}}(\boldsymbol{U};\boldsymbol{v})$.

Fitting of Affine Subspaces. In a similar manner as before, we give a method for fitting affine subspaces using weighted least squares. For any $\boldsymbol{x} \in \mathbb{R}^d$,

$$\begin{aligned} d(\boldsymbol{x}, \mathcal{S}_{\mathrm{sa}}(\boldsymbol{U},\boldsymbol{\mu})) &= \min_{\boldsymbol{y}\in\mathcal{S}_{\mathrm{sa}}(\boldsymbol{U},\boldsymbol{\mu})} \|\boldsymbol{x}-\boldsymbol{y}\|^2 \\ &= \left\|(\boldsymbol{I}_d - \boldsymbol{U}\boldsymbol{U}^{\mathsf{T}})(\boldsymbol{x}-\boldsymbol{\mu})\right\|^2. \end{aligned} \tag{A·4}$$

Given the weight $v_i \in \mathbb{R}_+$ for each data $\boldsymbol{x}_i$, we wish to find the affine subspace minimizing

$$J_{\mathrm{sa}}(\boldsymbol{U},\boldsymbol{\mu};\boldsymbol{v}) \equiv \frac{\sum_{i=1}^{\ell} v_i d(\boldsymbol{x}, \mathcal{S}_{\mathrm{sa}}(\boldsymbol{U},\boldsymbol{\mu}))}{\sum_{i=1}^{\ell} v_i}.$$

Without loss of generality, we can assume that $\sum_{i=1}^{\ell} v_i = 1$ since the set of optimal solutions is invariant of the scalar product of vectors in $\mathbb{R}^\ell$. Let us denote the training input vectors shifted with $\boldsymbol{Xv} \in \mathbb{R}^d$ by

$$\forall i \in \mathbb{N}_\ell : \quad \bar{\boldsymbol{x}}_i \equiv \boldsymbol{x}_i - \boldsymbol{Xv},$$

and the shifted design matrix by

$$\bar{\boldsymbol{X}} \equiv [\bar{\boldsymbol{x}}_1, \ldots, \bar{\boldsymbol{x}}_\ell].$$

Previously, we defined the functions $K(\cdot,\cdot)$ and $\boldsymbol{k}(\cdot)$, and a matrix $\boldsymbol{K}$ to represent inner products of input vectors. For the shifted input vectors, we introduce two functions, $\bar{K} : \mathbb{R}^d \times \mathbb{R}^d \mapsto \mathbb{R}$ and $\bar{\boldsymbol{k}} : \mathbb{R}^d \mapsto \mathbb{R}^\ell$, such that

$$\begin{aligned} \bar{K}(\boldsymbol{x},\boldsymbol{y}) &\equiv \langle \boldsymbol{x}-\boldsymbol{Xv}, \boldsymbol{y}-\boldsymbol{Xv}\rangle, \\ \bar{\boldsymbol{k}}(\boldsymbol{x}) &\equiv [\langle \bar{\boldsymbol{x}}_1, \boldsymbol{x}-\boldsymbol{Xv}\rangle, \ldots, \langle \bar{\boldsymbol{x}}_\ell, \boldsymbol{x}-\boldsymbol{Xv}\rangle]^{\mathsf{T}}. \end{aligned}$$

Then the shifted inner product matrix is given by $\bar{\boldsymbol{K}} \in \mathbb{S}^\ell$, where $\bar{K}_{i,j} = \langle \bar{\boldsymbol{x}}_i, \bar{\boldsymbol{x}}_j\rangle$. The following properties of the functions $\bar{K}(\cdot)$ and $\bar{\boldsymbol{k}}(\cdot)$, and matrix $\bar{\boldsymbol{K}}$, are useful for kernelization:

$$\begin{aligned} \bar{K}(\boldsymbol{x},\boldsymbol{y}) &= K(\boldsymbol{x},\boldsymbol{y}) - \langle \boldsymbol{k}(\boldsymbol{x}) + \boldsymbol{k}(\boldsymbol{y}), \boldsymbol{v}\rangle + \boldsymbol{v}^{\mathsf{T}}\boldsymbol{K}\boldsymbol{v}, \\ \bar{\boldsymbol{k}}(\boldsymbol{x}) &= \boldsymbol{k}(\boldsymbol{x}) - \langle \boldsymbol{k}(\boldsymbol{x}), \boldsymbol{v}\rangle \mathbf{1}_\ell - \bar{\boldsymbol{K}}\boldsymbol{v} + \langle \boldsymbol{v}, \bar{\boldsymbol{K}}\boldsymbol{v}\rangle \mathbf{1}_\ell, \\ \bar{\boldsymbol{K}} &= (\boldsymbol{I} - \mathbf{1}_\ell \boldsymbol{v}^{\mathsf{T}})\boldsymbol{K}(\boldsymbol{I} - \boldsymbol{v}\mathbf{1}_\ell^{\mathsf{T}}). \end{aligned}$$

Finally, we have the following weighted least squares estimation and the distance to the optimal affine manifold in an analytic and kernelizable form.

Theorem 4. *Define a symmetric matrix $\bar{\boldsymbol{K}} \equiv \boldsymbol{D}_{\boldsymbol{v}}^{1/2}\bar{\boldsymbol{K}}\boldsymbol{D}_{\boldsymbol{v}}^{1/2}$. Suppose $r \geq m$, where $\bar{r} \equiv \mathrm{rank}(\bar{\boldsymbol{K}})$. Denote the m major eigenvectors of $\bar{\boldsymbol{K}}$ by $\bar{\lambda}_1, \ldots, \bar{\lambda}_m \in \mathbb{R}_{++}$ and the corresponding eigenvalues by $\bar{\boldsymbol{b}}_1, \ldots, \bar{\boldsymbol{b}}_m \in \mathbb{R}^\ell$. Let $\bar{\boldsymbol{\lambda}} \equiv [\bar{\lambda}_1, \ldots, \bar{\lambda}_m]^{\mathsf{T}}$ and $\bar{\boldsymbol{B}} \equiv [\bar{\boldsymbol{b}}_1, \ldots, \bar{\boldsymbol{b}}_m]$, and define*

$$\hat{\boldsymbol{U}}_{\mathrm{sa}} \equiv \boldsymbol{X}\boldsymbol{D}_{\boldsymbol{v}}^{1/2}\bar{\boldsymbol{B}}\mathrm{diag}(\bar{\boldsymbol{\lambda}})^{-1/2} \text{ and } \hat{\boldsymbol{\mu}}_{\mathrm{sa}} \equiv \boldsymbol{Xv}.$$

Then

$$(\hat{\boldsymbol{U}}_{\mathrm{sa}}, \hat{\boldsymbol{\mu}}_{\mathrm{sa}}) \in \operatorname*{arg\,min}_{\boldsymbol{U}\in\mathbb{O}^{d\times m}, \boldsymbol{\mu}\in\mathbb{R}^d} J_{\mathrm{sa}}(\boldsymbol{U},\boldsymbol{\mu};\boldsymbol{v}).$$

Proof. Substituting (A·4) into $J_{\mathrm{sa}}(\boldsymbol{U},\boldsymbol{\mu};\boldsymbol{v})$ gives us

$$J_{\mathrm{sa}}(\boldsymbol{U},\boldsymbol{\mu};\boldsymbol{v}) = \frac{1}{2}\sum_{i=1}^{\ell} v_i \left\|(\boldsymbol{I}_d - \boldsymbol{U}\boldsymbol{U}^{\mathsf{T}})(\boldsymbol{x}_i - \boldsymbol{\mu})\right\|^2, \tag{A·5}$$

with the assumption that $\sum_{i=1}^{\ell} v_i = 1$. To minimize this, we set

$$\frac{\partial J_{\mathrm{sa}}(\boldsymbol{U},\boldsymbol{\mu})}{\partial \boldsymbol{\mu}} = (\boldsymbol{I}_d - \boldsymbol{U}\boldsymbol{U}^{\mathsf{T}})\sum_{i=1}^{\ell} v_i(\boldsymbol{\mu} - \boldsymbol{x}_i) = 0,$$

and obtain the solutions

$$\forall \boldsymbol{\beta} \in \mathbb{R}^m, \qquad \boldsymbol{\mu} = \boldsymbol{Xv} + \boldsymbol{U}\boldsymbol{\beta}. \tag{A·6}$$

The result of Theorem 4 follows when $\boldsymbol{\beta} = \mathbf{0}_m$. Substituting (A·6) into (A·5) yields

$$J_{\mathrm{sa}}(\boldsymbol{U}, \boldsymbol{Xv}; \boldsymbol{v}) = \frac{1}{2}\sum_{i=1}^{\ell} v_i \left\|(\boldsymbol{I}_d - \boldsymbol{U}\boldsymbol{U}^{\mathsf{T}})\bar{\boldsymbol{x}}_i\right\|^2,$$

which is equal to $J_{\mathrm{ss}}(\boldsymbol{U};\boldsymbol{v})$ when the $\bar{\boldsymbol{x}}_i$'s are treated as input vectors. By applying Theorem 1, the theorem is established. □

Corollary 5.

$$\begin{aligned} &d(\boldsymbol{x}, \mathcal{S}_{\mathrm{sa}}(\hat{\boldsymbol{U}}_{\mathrm{sa}}, \hat{\boldsymbol{\mu}}_{\mathrm{sa}})) \\ &= \bar{K}(\boldsymbol{x},\boldsymbol{x}) - \left\|\mathrm{diag}(\boldsymbol{\lambda})^{-1/2}\boldsymbol{B}\boldsymbol{D}_{\boldsymbol{v}}^{1/2}\bar{\boldsymbol{k}}(\boldsymbol{x})\right\|^2. \end{aligned}$$

The equation follows when the result of Theorem 4 is substituted in Eq. (A·4).

For completeness, we give the non-kernelized form in the following corollary.

Corollary 6. *The m columns in* $\hat{\boldsymbol{U}}_{\mathrm{sa}}$ *defined in Theorem 4 are m major eigenvectors of*

$$\bar{\boldsymbol{M}} = \sum_{i=1}^{\ell} v_i (\boldsymbol{x}_i - \hat{\boldsymbol{\mu}}_{\mathrm{sa}})(\boldsymbol{x}_i - \hat{\boldsymbol{\mu}}_{\mathrm{sa}})^{\mathsf{T}}.$$

Any set of m major eigenvectors of $\bar{\boldsymbol{M}}$ *minimizes* $J_{\mathrm{sa}}(\boldsymbol{U}, \hat{\boldsymbol{\mu}}_{\mathrm{sa}}; \boldsymbol{v})$.

Raissa Relator received the B.S. and M.S. degrees in Mathematics from the University of the Philippines Diliman in 2005 and 2008, respectively. She is currently pursuing a Ph.D. in Computer Science at the Graduate School of Science and Engineering, Gunma University. Her research interests include pattern recognition, machine learning, and bioinformatics.

Tsuyoshi Kato received the B.E., M.E., and Ph.D. degree from Tohoku University, Sendai, Japan, in 1998, 2000, and 2003, respectively. From 2003 to 2005, he was with the National Institute of Advanced Industrial Science Technology (AIST) as a postdoctoral fellow in the Computational Biology Research Center (CBRC) in Tokyo. From 2005 to 2008, he was an assistant professor at the Graduate School of Frontier Sciences, University of Tokyo. From 2008 to 2010, he was an associate professor at the Center for Informational Biology, Ochanomizu University. He then moved back to Graduate School of Frontier Sciences, University of Tokyo, and as of present, he is an associate professor at the Graduate School of Science and Engineering, Gunma University. His current scientific interests include pattern recognition, computer vision and bioinformatics. He is a member of IEICE Japan.

Takuma Tomaru received the B.E. and M.E. from Gunma University, Gunma, Japan, in 2009 and 2011, respectively. He was engaged in research on pattern recognition problems, especially sound recognition. He currently works for Taiyo Yuden Co., Ltd..

Naoya Ohta received the B.E. from Tokyo University of Agriculture and Technology in 1983, the M.E. from Tokyo Institute Technology in 1985, and the PhD degree from the University of Tokyo in 1998. From 1987, he worked for NEC Corporation as a researcher in the Central Research Laboratories. He was a research affiliate of the Media Laboratory of MIT from 1991 to 1992. In 1994, he moved to Gunma University and is currently a professor at the Graduate School of Science and Engineering, Gunma University. His scientific interests include image processing, computer vision, and pattern recognition.