Journal of Computer and Communications, 2022, 10, 95-112
https://www.scirp.org/journal/jcc
ISSN Online: 2327-5227
ISSN Print: 2327-5219

Adaptive Metric Learning for Dimensionality Reduction

Lihua Chen[1], Peiwen Wei[1], Zhongzhen Long[2], Yufeng Yu[1*]

[1]Department of Statistics, Guangzhou University, Guangzhou, China
[2]Shenzhen Securities Communication Co., Ltd., Shenzhen, China
Email: *yufengyu@gzhu.edu.cn

How to cite this paper: Chen, L.H., Wei, P.W., Long, Z.Z. and Yu, Y.F. (2022) Adaptive Metric Learning for Dimensionality Reduction. *Journal of Computer and Communications*, **10**, 95-112.
https://doi.org/10.4236/jcc.2022.1012008

Received: November 25, 2022
Accepted: December 27, 2022
Published: December 30, 2022

Abstract

Finding a suitable space is one of the most critical problems for dimensionality reduction. Each space corresponds to a distance metric defined on the sample attributes, and thus finding a suitable space can be converted to develop an effective distance metric. Most existing dimensionality reduction methods use a fixed pre-specified distance metric. However, this easy treatment has some limitations in practice due to the fact the pre-specified metric is not going to warranty that the closest samples are the truly similar ones. In this paper, we present an adaptive metric learning method for dimensionality reduction, called AML. The adaptive metric learning model is developed by maximizing the difference of the distances between the data pairs in cannot-links and those in must-links. Different from many existing papers that use the traditional Euclidean distance, we use the more generalized $l_{2,p}$-norm distance to reduce sensitivity to noise and outliers, which incorporates additional flexibility and adaptability due to the selection of appropriate p-values for different data sets. Moreover, considering traditional metric learning methods usually project samples into a linear subspace, which is overstrict. We extend the basic linear method to a more powerful nonlinear kernel case so that well capturing complex nonlinear relationship between data. To solve our objective, we have derived an efficient iterative algorithm. Extensive experiments for dimensionality reduction are provided to demonstrate the superiority of our method over state-of-the-art approaches.

Keywords

Adaptive Learning, Kernel Learning, Dimension Reduction, Pairwise Constraints

1. Introduction

Metric learning, which aims to supply a metric to measure the distance or simi-

larity between the data, is a vital issue in the field of computer vision or pattern recognition. Metric learning has enormously wide spectrum of applications, such as classification [1] [2], person re-identification [3] [4], object tracking [5] [6], image retrieval [7], feature reduction [8] [9] and clustering [10] [11] [12] [13]. It should be noted that the performance of all these applications depends in part on the effectiveness of metric learning. Although some classical distance metric methods are proposed, such as Euclidean distance and cosine similarity, they do not distinguish the features with different importance. A great distance metric method should be able to apprehend the characters of the data of interest and enhance the robustness.

According to different learning methods, distance metric learning can be divided into supervised learning, weakly-supervised learning and unsupervised learning. In unsupervised case, distance metric learning obtains the low-dimensional representation by performing dimensionality reduction of the original data [14] [15] [16] [17]. In supervised case, some distance metric learning based methods adopt the training sets with label information and solve the objective functions to obtain a metric matrix [18] [19] [20] [21] [22]. Neighbourhood component analysis (NCA) [23] is a well-known supervised distance metric method, it uses stochastic nearest neighbors to perform metric learning. Maximally collapsing metric learning (MCML) [24] is another well-known supervised distance metric method, which tries to collapse all examples in the same class to a single point and push examples in other classes infinitely far away. Recently significant attention has been dedicated to the matter of learning a metric in the weakly-supervised case by using pairwise constraints: must-links and cannot-links [25] [26] [27]. In particular, Xiang *et al.* [10] learn the metric matrix by minimizing the ratio of the squared l_2-norm of two matrices, which describe the distances between pairs of point in must-links and those in cannot-links. Liao *et al.* [28] propose a two-stage metric learning method by combining $l_{2,1}$-norm with LDA. To address the sensitive problem to outliers in [10], Liu *et al.* [26] introduced the not-squared l_2-norm instead of the squared l_2-norm in objective function. The above approaches try to produce a homogenous metric for all datasets. In fact, it is virtually impossible to find a metric that matches all training data. Recently, $l_{2,p}$-norm is used to replace l_2-norm or $l_{2,1}$-norm as distance metric for improving the robustness, such as DCM [29] and $l_{2,p}$-PCA [30], $l_{2,p}$-2-DPCA [31], and RDS [32]. In [32], robust discriminant subspace (RDS) learning model is developed for dimensionality reduction. It might be flexible to choose appropriate p in keeping with the data and thus obtains more robust metric learning performance.

Dimensionality reduction is a critical step in pattern recognition systems. It transfers the original data from a high-dimensional space to a low-dimensional space through mathematical transformations [33]. Many strategies are adopted for dimensionality reduction, which can be loosely classified into two categories: linear dimensionality reduction and nonlinear dimensionality reduction. Prin-

cipal component analysis (PCA) [34] and linear discriminant analysis (LDA) [35] are the two classical linear dimensionality reduction methods and have been commonly used in different areas due to their relative effectiveness and simplicity. Zhao *et al.* [36] combine PCA with LDA, and propose a joint framework to extract discriminant information. LDA employs l_2 -norm to formulate objective function and is sensitive to outliers. To deal with this problem, Zhao *et al.* [37] use $l_{2,1}$ -norm to develop a new LDA formulation for improving robustness. Also, some other linear dimensionality reduction approaches using local information of data have been proposed, such as locality preserving projections [38] and local linear embedding [39] [40]. These methods achieve significant discriminant performance to deal with linear problems. However, once the complexity and dimensionality of data increase, the distribution of data is typically nonlinear and additional feature information is probably going to be hidden within the nonlinear structure. The linear versions of PCA and LDA only retain the linear structure in learning subspace, which may result in poor perfor_mance. To compensate the shortcoming of linear dimensionality reduction methods, some researchers introduce kernel trick into PCA and LDA, and propose kernelized versions, such as kernel PCA [41] and kernel LDA [42].

In this paper, we propose an adaptive metric learning method for dimensionality reduction (AML). Inspired by [26], we formulate our objective function by using pairwise constraints: must-links and cannot-links. Different from [26], we construct the metric learning model for dimensionality reduction by minimizing the distances between pairs of points in must-links and maximizing those in cannot-links below the orthogonal constraint. Meanwhile, we use $l_{2,p}$ -norm to measure the similarities, which can enhance the robustness of the model to outliers. Moreover, we present a kernel version of AML (KAML) to deal with the nonlinear problems. The main contributions of this paper are summarized as follows:

- $l_{2,p}$ -norm, instead of l_2 -norm is used as distance metric in the proposed method, which is robust to outliers. Meanwhile, it might be flexibly chooses appropriate p in keeping with the data and thus obtains better metric learning performance.
- An extension of the presented method is proposed in the reproducing kernel Hilbert space. The nonlinear relationship between samples can be captured to improve the discriminative power.
- Developing an efficient algorithm to solve the proposed framework and discussing the efficiency of proposed approach in aspect of parameter sensitivity.
- Experiments on various clustering tasks show that the proposed method can achieve competitive performance compared to the state-of-the-art approaches.

The rest of this paper is organized as follows. In Section 2, we introduce the adaptive metric learning method for dimensionality reduction. In Section 3, we elaborate the details of an extension of the presented method. In Section 4, experiments are designed to validate the proposed method and analyse parameter

sensitivity. Finally, we conclude this paper in Section 5.

2. Linear Adaptive Distance Metric Learning

In this section, we present the linear adaptive metric learning model (AML) for dimensionality reduction and the effective optimization algorithm.

2.1. Objective Function Construction

Suppose we have a high-dimensional data set which consists of n samples $\left\{\boldsymbol{x}_i \in R^D\right\}_{i=1}^n$, and two sets of pairwise constraints are defined as:

$$\begin{cases} \mathcal{M} = \left\{\left(\boldsymbol{x}_i, \boldsymbol{x}_j\right) | \text{ if } \boldsymbol{x}_i \text{ and } \boldsymbol{x}_j \text{ are in the same class}\right\} \\ \mathcal{C} = \left\{\left(\boldsymbol{x}_i, \boldsymbol{x}_j\right) | \text{ if } \boldsymbol{x}_i \text{ and } \boldsymbol{x}_j \text{ are in the different class}\right\} \end{cases}$$

where $\mathcal{M}$ and $\mathcal{C}$ are named as must-links and cannot-links, respectively.

Because of the large amount of noise, outliers and redundant features in the high-dimensional data, it makes the algorithm performance degradation. A straightforward idea is to use a linear transformation $\boldsymbol{W} \in R^{D\times d}$ such that each sample $\boldsymbol{x}_i$ in D-dimensional space is mapped into $\boldsymbol{y}_i$ in d-dimensional space, as follows:

$$\boldsymbol{x}_i \in R^D \to \boldsymbol{y}_i = \boldsymbol{W}^{\mathrm{T}} \boldsymbol{x}_i \in R^d \left(d \ll D\right)$$

Under this transformation, the $l_{2,p}$-norm distance of the point pairs in $\mathcal{M}$ can be calculated as follows:

$$d_w = \sum_{\left(\boldsymbol{x}_i, \boldsymbol{x}_j\right) \in \mathcal{M}} \left\| \boldsymbol{W}^{\mathrm{T}} \left(\boldsymbol{x}_i - \boldsymbol{x}_j\right) \right\|_2^p \tag{1}$$

Correspondingly, for the point pairs in $\mathcal{C}$, we have

$$d_c = \sum_{\left(\boldsymbol{x}_i, \boldsymbol{x}_j\right) \in \mathcal{C}} \left\| \boldsymbol{W}^{\mathrm{T}} \left(\boldsymbol{x}_i - \boldsymbol{x}_j\right) \right\|_2^p \tag{2}$$

Our goal is to minimize the d_w and maximize d_c under the orthogonal constraint. The objective function can be formulated as:

$$\max_{\boldsymbol{W}^{\mathrm{T}}\boldsymbol{W}=\boldsymbol{I}} \sum_{\left(\boldsymbol{x}_i, \boldsymbol{x}_j\right) \in \mathcal{C}} \left\| \boldsymbol{W}^{\mathrm{T}} \left(\boldsymbol{x}_i - \boldsymbol{x}_j\right) \right\|_2^p - \sum_{\left(\boldsymbol{x}_i, \boldsymbol{x}_j\right) \in \mathcal{M}} \left\| \boldsymbol{W}^{\mathrm{T}} \left(\boldsymbol{x}_i - \boldsymbol{x}_j\right) \right\|_2^p \tag{3}$$

Note that some existing methods use side information to learn the distance metric. They adopt the l_2 or $l_{2,1}$-norm to measure the distance between two data points. Unlike these methods, the model in Equation (3) uses the $l_{2,p}$-norm as the distance metric function, which possesses flexibility in choosing the appropriate p according to different data and has stronger robustness.

2.2. Optimization Algorithm

The existence of $l_{2,p}$-norm makes it difficult to solve the objective function (3) directly. [30] adopts an iterative algorithm for solving the objective function in the form of $l_{2,p}$-norm. The similar approach is adopted in [43] to solve the LDA

minimization problem based on $l_{2,p}$-norm ($0 \le p \le 2$). Inspired by these papers, we investigate the problem and give an efficient iterative method for solving our objective function.

Let $d_{ij} = \left\| \boldsymbol{W}^{\mathrm{T}} \left(x_i - x_j \right) \right\|_2^{p-2}$, then Equation (3) can be transformed to

$$\max_{\boldsymbol{W}^{\mathrm{T}}\boldsymbol{W}=\boldsymbol{I}} \sum_{\left(\boldsymbol{x}_i,\boldsymbol{x}_j\right)\in\mathcal{C}} d_{ij} \left\| \boldsymbol{W}^{\mathrm{T}} \left(\boldsymbol{x}_i - \boldsymbol{x}_j \right) \right\|_2^2 - \sum_{\left(\boldsymbol{x}_i,\boldsymbol{x}_j\right)\in\mathcal{M}} d_{ij} \left\| \boldsymbol{W}^{\mathrm{T}} \left(\boldsymbol{x}_i - \boldsymbol{x}_j \right) \right\|_2^2 \tag{4}$$

Denote $\boldsymbol{L}_{\mathcal{M}}$ and $\boldsymbol{L}_{\mathcal{C}}$ as the indicator matrices of the must-links and cannot-links, respectively:

$$\begin{cases} \boldsymbol{L}_{\mathcal{M}}(i,j) = \boldsymbol{L}_{\mathcal{M}}(j,i) = 1, \left(\boldsymbol{x}_i, \boldsymbol{x}_j\right) \in \mathcal{M} \\ \boldsymbol{L}_{\mathcal{M}}(i,j) = \boldsymbol{L}_{\mathcal{M}}(j,i) = 0, \left(\boldsymbol{x}_i, \boldsymbol{x}_j\right) \notin \mathcal{M} \end{cases}$$

$$\begin{cases} \boldsymbol{L}_{\mathcal{C}}(i,j) = \boldsymbol{L}_{\mathcal{C}}(j,i) = 1, \left(\boldsymbol{x}_i, \boldsymbol{x}_j\right) \in \mathcal{C} \\ \boldsymbol{L}_{\mathcal{C}}(i,j) = \boldsymbol{L}_{\mathcal{C}}(j,i) = 0, \left(\boldsymbol{x}_i, \boldsymbol{x}_j\right) \notin \mathcal{C} \end{cases}$$

With the indicator matrices, the objective function (4) can be expressed as:

$$\max_{\boldsymbol{W}^{\mathrm{T}}\boldsymbol{W}=\boldsymbol{I}} \sum_{i,j} \boldsymbol{L}_{\mathcal{C}}(i,j) d_{ij} \left\| \boldsymbol{W}^{\mathrm{T}} \left(\boldsymbol{x}_i - \boldsymbol{x}_j \right) \right\|_2^2 - \sum_{i,j} \boldsymbol{L}_{\mathcal{M}}(i,j) d_{ij} \left\| \boldsymbol{W}^{\mathrm{T}} \left(\boldsymbol{x}_i - \boldsymbol{x}_j \right) \right\|_2^2 \tag{5}$$

We introduce two auxiliary variables $\hat{\boldsymbol{L}}_{\mathcal{C}}$ and $\hat{\boldsymbol{L}}_{\mathcal{M}}$ by defining

$$\hat{\boldsymbol{L}}_{\mathcal{C}}(i,j) = \boldsymbol{L}_{\mathcal{C}}(i,j) d_{ij} \tag{6}$$

and

$$\hat{\boldsymbol{L}}_{\mathcal{M}}(i,j) = \boldsymbol{L}_{\mathcal{M}}(i,j) d_{ij} \tag{7}$$

Moreover, we assume

$$\boldsymbol{L}_w = \mathrm{diag}\left(\mathrm{sum}\left(\hat{\boldsymbol{L}}_{\mathcal{M}} \right) \right) - \hat{\boldsymbol{L}}_{\mathcal{M}} \tag{8}$$

$$\boldsymbol{L}_c = \mathrm{diag}\left(\mathrm{sum}\left(\hat{\boldsymbol{L}}_{\mathcal{C}} \right) \right) - \hat{\boldsymbol{L}}_{\mathcal{C}} \tag{9}$$

$\mathrm{Sum}(.)$ is a vector that represents the sum of each row of the matrix. Then the optimization problem (5) becomes:

$$\begin{aligned} &\max_{\boldsymbol{W}^{\mathrm{T}}\boldsymbol{W}=\boldsymbol{I}} \sum_{i,j} \hat{\boldsymbol{L}}_{\mathcal{C}}(i,j) \left\| \boldsymbol{W}^{\mathrm{T}} \left(\boldsymbol{x}_i - \boldsymbol{x}_j \right) \right\|_2^2 - \sum_{i,j} \hat{\boldsymbol{L}}_{\mathcal{M}}(i,j) \left\| \boldsymbol{W}^{\mathrm{T}} \left(\boldsymbol{x}_i - \boldsymbol{x}_j \right) \right\|_2^2 \\ &= \max_{\boldsymbol{W}^{\mathrm{T}}\boldsymbol{W}=\boldsymbol{I}} \mathrm{Tr}\left(\boldsymbol{W}^{\mathrm{T}} \boldsymbol{X} \left(\boldsymbol{L}_c - \boldsymbol{L}_w \right) \boldsymbol{X}^{\mathrm{T}} \boldsymbol{W} \right) = \max_{\boldsymbol{W}^{\mathrm{T}}\boldsymbol{W}=\boldsymbol{I}} \mathrm{Tr}\left(\boldsymbol{W}^{\mathrm{T}} \boldsymbol{X} \boldsymbol{D} \boldsymbol{X}^{\mathrm{T}} \boldsymbol{W} \right) \end{aligned} \tag{10}$$

where $\boldsymbol{D} = \boldsymbol{L}_c - \boldsymbol{L}_w$.

By observing Equation (10), we can see that d_{ij} is not independent of the matrix of $\boldsymbol{W}$. Hence, we propose an effective iterative strategy to optimize d_{ij} and $\boldsymbol{W}$ alternately. That is, when d_{ij} is fixed, the $\boldsymbol{W}$ is updated; and then fixing $\boldsymbol{W}$, updating d_{ij}. Specifically, suppose that in $t+1$ iteration, d_{ij}^t is given, then we know matrix $\boldsymbol{D}$, so we can solve for $\boldsymbol{W}$ by maximizing Equation (10). After that, we use the updated $\boldsymbol{W}$ to update d_{ij}, and this iterative process is repeated until the algorithm converges. **Algorithm 1** summarizes the optimization procedure.

Algorithm 1. AML algorithm.

Input: Data set $\boldsymbol{X} \in R^{D\times n}$; The reduced dimension d, parameter p. **Output**: optimal transformation matrix $\boldsymbol{W} \in R^{D\times d}$.

1: Initialize $\boldsymbol{W}$ by using LDA [35];

2: **Repeat** step 3-step 5;

3: Update d_{ij}^{t} by using $d_{ij} = \left\| \boldsymbol{W}^{\mathrm{T}} \left(\boldsymbol{x}_i - \boldsymbol{x}_j \right) \right\|_2^{p-2}$;

4: Update $\boldsymbol{W}^{t}$ by using the eigenvalue decomposition of Equation (10);

5: $t = t+1$;

6: **Until** Convergence;

7: Return transformation matrix $\boldsymbol{W}$.

3. Nonlinear Adaptive Distance Metric Learning

In this section, we present the nonlinear adaptive metric learning model for dimensionality reduction and the effective optimization algorithm.

3.1. Objective Function Construction

Propelled by nonlinear generalization performance of kernel techniques in metric learning, we present a kernel version of AML, namely KAML in short. Essentially, KAML forms the distance metric in a reproducing kernel Hilbert space. That is, there exists a reproducing kernel Hilbert space H and a nonlinear map $\varphi : R^{D} \to H$, such that we have $\boldsymbol{x} \to \varphi(\boldsymbol{x})$. After playing out the nonlinear map, the nonlinear adaptive metric learning issue can be planned as the following maximization issue:

$$\max_{\boldsymbol{W}_{\varphi}} \sum_{(\boldsymbol{x}_i, \boldsymbol{x}_j)\in\mathcal{C}} \left\| \boldsymbol{W}_{\varphi}^{\mathrm{T}} \left(\varphi(\boldsymbol{x}_i) - \varphi(\boldsymbol{x}_j) \right) \right\|_2^{p} - \sum_{(\boldsymbol{x}_i, \boldsymbol{x}_j)\in\mathcal{M}} \left\| \boldsymbol{W}_{\varphi}^{\mathrm{T}} \left(\varphi(\boldsymbol{x}_i) - \varphi(\boldsymbol{x}_j) \right) \right\|_2^{p} \tag{11}$$

where $\boldsymbol{W}_{\varphi} = \left(\boldsymbol{W}_{\varphi 1}, \cdots, \boldsymbol{W}_{\varphi d} \right)$ is the transformation matrix in the feature space. Denoting $d_{ij}^{\varphi} = \left\| \boldsymbol{W}_{\varphi}^{\mathrm{T}} \left(\varphi(\boldsymbol{x}_i) - \varphi(\boldsymbol{x}_j) \right) \right\|_2^{p-2}$ and following the same algebraic calculus of AML, Equation (11) can be transformed to

$$\max_{\boldsymbol{W}_{\varphi}} \mathrm{Tr}\left(\boldsymbol{W}_{\varphi}^{\mathrm{T}} \varphi(\boldsymbol{X}) \boldsymbol{D}_{\varphi} \varphi^{\mathrm{T}}(\boldsymbol{X}) \boldsymbol{W}_{\varphi} \right) \tag{12}$$

where the construction of $\boldsymbol{D}_{\varphi}$ is similar to $\boldsymbol{D}$ in Equation (10).

According to the Representer Theorem [44], we define the coefficients as $\boldsymbol{\alpha}_k \left(k = 1, \cdots, d \right)$, $\boldsymbol{W}_{\varphi}$ can be represented as a linear combination of vectors $\varphi(\boldsymbol{x})$ in the space H:

$$\boldsymbol{W}_{\varphi k} = \sum_{k=1}^{n} \boldsymbol{\alpha}_{ki} \varphi(\boldsymbol{x}_i) = \varphi(\boldsymbol{x}) \boldsymbol{\alpha}_k \tag{13}$$

where the $\varphi(\boldsymbol{x}) = \left(\varphi(\boldsymbol{x}_1), \cdots, \varphi(\boldsymbol{x}_n) \right)$ and $\boldsymbol{\alpha}_k = \left(\boldsymbol{\alpha}_{k1}, \cdots, \boldsymbol{\alpha}_{kn} \right)^{\mathrm{T}}$.

Using the Gaussian kernel function to define the kernel similarity between the samples $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ as $K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \varphi(\boldsymbol{x}_i)^{\mathrm{T}} \varphi(\boldsymbol{x}_j) = \exp\left(-\left\| \boldsymbol{x}_i - \boldsymbol{x}_j \right\|_2^2 / \sigma \right)$, where σ is the kernel width, which control the radial range of the Gaussian kernel function. According to Equation (13), we get:

$$\boldsymbol{W}_{\varphi}^{\mathrm{T}}\varphi(\boldsymbol{X})=\boldsymbol{\alpha}^{\mathrm{T}}\boldsymbol{K} \tag{14}$$

where $\boldsymbol{\alpha}=(\boldsymbol{\alpha}_1,\cdots,\boldsymbol{\alpha}_d)^{\mathrm{T}}$ and $\boldsymbol{K}=\varphi(\boldsymbol{X}_k)^{\mathrm{T}}\varphi(\boldsymbol{X}_k)$ is the kernel matrix.

Combining Equation (14), problem Equation (12) can be rewritten as:

$$\max_{\boldsymbol{\alpha}}\mathrm{Tr}\left(\boldsymbol{\alpha}^{\mathrm{T}}\boldsymbol{K}\boldsymbol{D}_{\varphi}\boldsymbol{K}^{\mathrm{T}}\boldsymbol{\alpha}\right) \tag{15}$$

It can be seen that finding the transformation matrix $\boldsymbol{W}_{\varphi}$ is equivalent to find a coefficient matrix $\boldsymbol{\alpha}$.

3.2. Optimization Algorithm

Inspired by [45], we use the eigenvectors decomposition of the matrix K:

$$\boldsymbol{K}=\boldsymbol{U}\boldsymbol{\Gamma}\boldsymbol{U}^{\mathrm{T}} \tag{16}$$

where $\boldsymbol{U}\boldsymbol{U}^{\mathrm{T}}=\boldsymbol{I}$. Combining Equation (15) and Equation (16), we further have:

$$\max_{\boldsymbol{\alpha}}\mathrm{Tr}\left(\boldsymbol{\Gamma}\boldsymbol{U}^{\mathrm{T}}\boldsymbol{\alpha}\right)^{\mathrm{T}}\boldsymbol{U}^{\mathrm{T}}\boldsymbol{D}_{\varphi}\boldsymbol{U}\left(\boldsymbol{\Gamma}\boldsymbol{U}^{\mathrm{T}}\boldsymbol{\alpha}\right) \tag{17}$$

Suppose $\boldsymbol{\beta}=\boldsymbol{\Gamma}\boldsymbol{U}^{\mathrm{T}}\boldsymbol{\alpha}$, we have:

$$\max_{\boldsymbol{\beta}}\boldsymbol{\beta}^{T}U^{T}D_{\varphi}U\boldsymbol{\beta} \tag{18}$$

The optimization problem (18) can be solved by the eigenvalue decomposition. $\boldsymbol{\beta}$ is composed of d eigenvectors that correspond to the d largest eigenvalues of the matrix $\boldsymbol{U}^{\mathrm{T}}\boldsymbol{D}_{\varphi}\boldsymbol{U}$, For a given $\boldsymbol{\beta}$, there exists $\boldsymbol{\alpha}$ satisfying Equation (17) in the form $\boldsymbol{\alpha}=\boldsymbol{U}\boldsymbol{\Gamma}^{-1}\boldsymbol{\beta}$. For clarity, the detailed procedure is summarized in **Algorithm 2**.

Algorithm 2. KAML algorithm.

Input: Data set $\boldsymbol{X}\in R^{D\times n}$; the reduced dimension d, parameter p, $\boldsymbol{\sigma}$. **Output**: Coefficient matrix $\boldsymbol{\alpha}\in R^{n\times d}$.

1: Construct the kernel matrix $\boldsymbol{K}$;

2: Decompose $\boldsymbol{K}$ by using eigenvectors decomposition;

3: Initialize $\boldsymbol{\alpha}$ by using KDA[42];

4: **Repeat** step 5-step 9;

5: Update $\boldsymbol{W}_{\varphi k}$ base on (13);

6: Update $d_{ij}^{\varphi}=\left\|\boldsymbol{W}_{\varphi}^{\mathrm{T}}\left(\varphi(\boldsymbol{x}_i)-\varphi(\boldsymbol{x}_j)\right)\right\|_2^{p-2}$;

7: Update $\boldsymbol{\beta}$ by using the eigenvalue decomposition of (18);

8: Update $\boldsymbol{\alpha}$ by $\boldsymbol{\alpha}=\boldsymbol{U}\boldsymbol{\Gamma}^{-1}\boldsymbol{\beta}$;

9: $t=t+1$;

10: **Until** convergence;

11: Return coefficient matrix $\boldsymbol{\alpha}$.

4. Experiments

In the experiments, six datasets, including three UCI datasets (wine, breast and the lonosphere dataset), two benchmark image datasets (coil-20 [46] and Caltech101 [47]), one palmprint dataset, are adopted to test the performance of the proposed AML and KAML on classification. To display the competitive perfor-

mance of our methods, we compare them with some existing methods: PCA [34], LDA [35], Discriminant component analysis (DCA) [7], Learning a Mahalanobis distance metric (MDL) [10], Large Margin Nearest Neighbor Classification (LMNN) [48], A new formulation of linear discriminant analysis for robust dimensionality reduction (RLDA) [37], MCML [24], NCA [23]. All these compared methods are implemented by following the original papers. The 1-NN classifier is applied to the projected samples for classification. All the compared methods are implemented in MATLAB (R2019a). The computer processor is Intel (R) Core (TM) i7-7700HQ CPU @ 2.80GHz 2.80 GHz, and the memory is 16-GB.

4.1. Data Description

The six databases include:

UCl databases: We evaluate our methods on three UCI databases, which include Breast, Ionosphere, and Wine databases. We choose half of the subjects for training and the other half for testing.

COIL-20 object dataset [46]: It contains 20 different objects and each object has 72 images taken at pose intervals of 5 degrees. All images have a pixel size of 64 × 64. In our experiments, we randomly choose $Ts\left(Ts=10,20,30\right)$ images of each subject for training, and the others are used for testing. **Figure 1(a)** shows some image examples on this dataset.

Caltech101 [47]: The Caltech101 dataset consists of a total of 9146 images and has 101 different object categories. Each category has about 40 to 800 images and each image is about 300 × 200 pixels. In this paper, these images are cropped and resized to 60 × 50 pixels. In our experiments, we randomly select $Ts\left(Ts=10,20,30\right)$ images of each subject for training, and the others are used for testing.

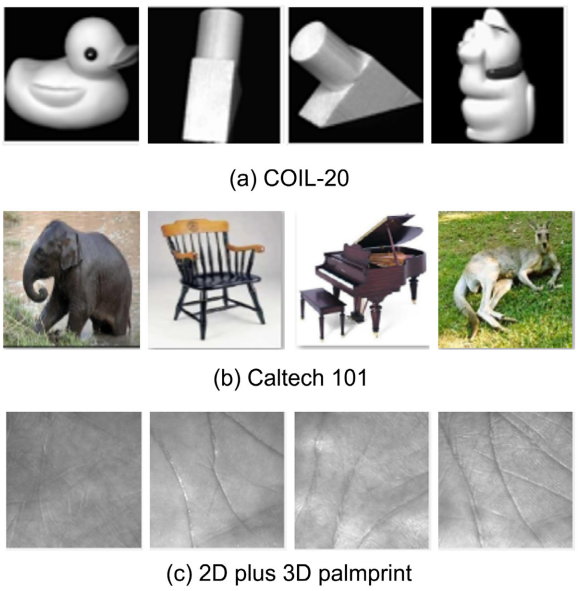

Figure 1. Some image examples on the COIL-20, Caltech 101 and 2D plus 3D palmprint databases.

2D plus 3D palmprint database [49]: The 2D plus 3D palmprint database contains 400 different palms with a total of 8000 samples, which means that there are two independent sessions of 10 palm samples per session. The interval between the two sessions is approximately 30 days. Each sample consists of a 2D region of interest and its corresponding 3D region of interest. All samples are cropped to a size of 64 × 64. In this paper, the performances of the algorithms are evaluated using 2D images. To point out various evaluation purposes, we divided the experiments into two groups. In each group, there are 10 palm print samples for each palm, two of which are randomly assigned for training and the other eight for testing.

4.2. Classification Results Comparisons

In this section, we show the classification results of various methods on different databases. The best combination of parameters is chosen for our methods. We repeat the experiments for 10 times and the average value is taken. The results are shown in Tables 1-4.

Table 1. Classification rates for different approaches on the UCI database (*mean* ± *std* (%)).

	Breast	Wine	Ionosphere
PCA	95.02 ± 1.20	95.45 ± 1.07	87.89 ± 1.17
LDA	95.42 ± 0.96	98.18 ± 2.02	88.63 ± 2.66
DCA	94.59 ± 1.33	96.93 ± 1.32	88.57 ± 3.06
MDL	95.39 ± 0.67	98.52 ± 1.32	89.37 ± 3.06
LMNN	94.56 ± 1.01	97.05 ± 1.71	88.57 ± 2.88
RLDA	93.58 ± 1.19	88.75 ± 2.36	89.77 ± 2.02
MCML	95.87 ± 0.92	97.95 ± 1.17	86.52 ± 2.93
NCA	94.93 ± 0.83	96.93 ± 1.61	88.12 ± 1.92
AML	95.70 ± 0.75	98.98 ± 0.98	91.37 ± 1.23
KAML	96.27 ± 0.67	95.23 ± 0.74	92.34 ± 1.29

Table 2. Classification rates for different approaches on the Coil-20 database (*mean* ± *std* (%)).

	$Ts = 10$	$Ts = 20$	$Ts = 30$
PCA	89.15 ± 1.25	94.60 ± 0.55	97.23 ± 0.64
LDA	91.04 ± 1.39	94.80 ± 1.05	96.48 ± 1.07
DCA	5.37 ± 0.65	90.24 ± 1.09	94.26 ± 0.77
MDL	85.80 ± 1.69	90.01 ± 1.48	91.56 ± 1.52
LMNN	90.56 ± 1.32	95.50 ± 0.44	98.56 ± 0.55
RLDA	88.53 ± 1.25	94.39 ± 0.61	97.27 ± 0.62
MCML	89.29 ± 1.22	94.40 ± 0.57	97.21 ± 0.54
NCA	81.87 ± 2.14	80.27 ± 4.34	70.70 ± 20.03
AML	93.07 ± 1.30	97.51 ± 0.78	99.45 ± 0.09
KAML	92.74 ± 1.39	95.83 ± 0.52	98.20 ± 0.56

Table 3. Classification rates for different approaches on the caltech101 database (*mean* ± *std* (%)).

	Ts = 10	*Ts* = 20	*Ts* = 30
PCA	46.93 ± 1.24	56.88 ± 0.70	62.76 ± 0.43
LDA	53.66 ± 1.02	49.85 ± 0.61	39.02 ± 24.20
DCA	23.76 ± 1.65	41.76 ± 2.62	51.08 ± 1.42
MDL	7.50 ± 1.50	13.89 ± 1.90	62.82 ± 1.60
LMNN	49.92 ± 0.95	56.81 ± 1.34	63.11 ± 1.43
RLDA	50.79 ± 0.72	58.05 ± 0.75	62.92 ± 0.52
MCML	23.78 ± 2.39	27.97 ± 1.98	31.31 ± 1.18
NCA	39.06 ± 0.72	42.01 ± 3.34	38.59 ± 2.11
AML	61.22 ± 0.88	68.95 ± 1.02	63.79 ± 0.43
KAML	14.67 ± 5.65	56.43 ± 0.73	60.24 ± 0.78

Table 4. Classification rates for different approaches on the 2D plus 3D palmprint database (*mean* ± *std* (%)).

	Session1	Session2
PCA	98.09 ± 0.28	98.70 ± 0.27
LDA	96.35 ± 0.37	97.68 ± 0.34
DCA	95.66 ± 0.44	97.00 ± 0.02
MDL	94.74 ± 0.68	96.48 ± 0.22
LMNN	98.59 ± 0.59	98.24 ± 1.10
RLDA	91.02 ± 0.75	92.71 ± 0.58
MCML	97.44 ± 0.31	98.31 ± 0.32
NCA	91.82 ± 1.82	91.95 ± 2.40
AML	98.63 ± 0.20	99.13 ± 0.23
KAML	96.34 ± 0.36	97.41 ± 0.37

Comparisons on the UCI database: The average and standard deviations of the classification accuracies are calculated and reported. The classification results are summarized in Table 1. It can be observed that the performance is significantly improved after performing distance metric learning. Compared to the classical dimensionality reduction methods PCA and LDA, which consider the Euclidean distance, our method achieves better classification rates. From the results of breast and ionosphere datasets, our KAML obtains classification rates of 96.27% and 92.34% respectively, which are higher than others. It indicates that the interesting data points in low dimensional space are more separable after mapping to the kernel space.

Comparisons on the COIL-20 object database: We report the average and standard deviations of the classification accuracies and the results are shown in Table 2. We can see that our linear model outperforms all compared approaches

and the accuracies are 93.07%, 97.51% and 99.45%, respectively. In addition, our nonlinear method also achieves good experimental results on the three subsets.

Comparisons on the Caltech101 database: The classification accuracies are reported in Table 3. We note that most methods do not perform well in classification when the number of training samples is small ($Ts=10$), but our linear method still maintains better results. Moreover, we can see that our linear model achieves stable classification accuracies of over 60% in all three subsets, especially in subset 2, which reaches 68%, much better than other methods. Although the classification effect of the proposed KAML is relatively poor when the training set size is $Ts=10$, the classification accuracy of KAML increases rapidly with the increase of sample size. It is possible that the classifier produces overfitting when the training set is too small. In addition, the performance of KAML is affected by the kernel function and kernel parameters in addition to the d and p values. Without knowing the sample distribution, we empirically choose Gaussian kernel function as our kernel function, and the value of kernel parameter σ is also selected according to the heuristic principle, which may not be the most suitable for Caltech101. However, compared with the classical supervised dimensionality reduction methods such as LDA and DCA, the classification effect of our method is better with the increase of sample size.

Comparisons on the 2D plus 3D palmprint database: The average classification accuracies and the standard deviations are calculated and reported. The results are listed in Table 4. We can see that with the exception of DCA, MDL and RLDA, all other algorithms can obtain the accuracies higher than 96%. The best performance is achieved by our linear method and the classification accuracies are 98.63% and 99.13%, respectively. It indicates that our distance metric model is effective in reducing dimensionality and thus improving classification performance.

4.3. Impact of Dimension Reduction

We investigate the effect of reduced dimension d on different datasets by different methods. On the three UCI datasets, we set d as $\{2,3,\cdots,7,8\}$. On the Coil20, Caltech101 and 2D plus 3D palmprint databases, d values are changed by $\{200,250,\cdots,450,500\}$. Each experiment is repeated ten times, and the mean value is recorded. The results are shown in Figure 2. Through observation, the following conclusions can be summarized:

- Not everyone of the methods can accomplish better outcomes when d is increased, which demonstrates that dimension reduction can successfully enhance classification performance.

- Some distance metric learning methods will cause performance degradation due to excessive information loss when only a small dimension is retained.

- Our approach performs better than the existing methods on small dimensions. Also, ideal outcomes are normally acquired on the small dimensions, which show that our model can effectively choose the important features in the data. Overall, our approach yields better results in most cases.

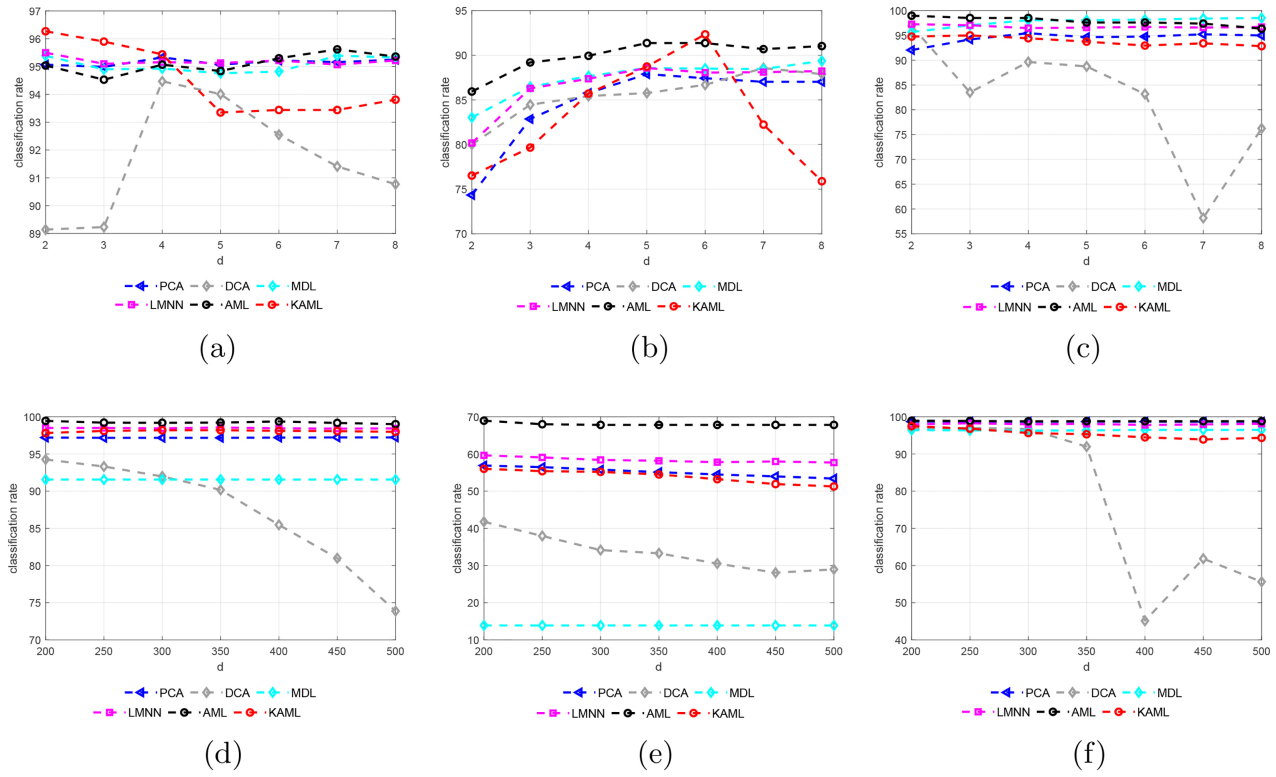

Figure 2. Classification rates of the different methods with different value of d on the various databases. (a) Breast. (b) Ionosphere. (c) Wine. (d) Coil20. (e) Caltech101. (f) 2D plus 3D palmprint.

4.4. Parameter Sensitivity Analysis

To comprehend the impact of the parameters p and σ on the results of the classification experiments, we discuss one of the parameters while the other is fixed. As can be seen from **Figure 3** and **Figure 4**, both p and σ have great influence on the final classification accuracy. We fist analyse the impact of p. From the experimental results, we observe that the changes of p affect the classification results. Especially in the classification accuracies on the Caltech101 and Breast databases, the performances of AML and KAML are incredibly impacted by p. It can be seen that the best results of each database are acquired at various p, which implies that it is not fitting to use the same distance metric for each data set. Moreover, the results further demonstrate that the effective of our adaptive metric learning strategy.

In addition, we empirically take the values of σ changing by $\left\{2^{-6}, 2^{-5}, \cdots, 2^{5}, 2^{6}\right\}$. As listed in **Figure 4**, the classification accuracies of most databases are not high when $\sigma < 2^{0}$. For example, in **Figure 4(e)**, the results of $\sigma < 2^{0}$ are almost close to 0 on the Caltech101 database, but increasing rapidly when $\sigma > 2^{0}$. In addition, with the change of σ, the classification results of all databases have the conspicuous peaks. Specifically, it could achieve the best performance of the proposed KAML when the values of parameter σ are set as 2^{3}, 2^{2}, 2^{2}, 2^{2}, 2^{3} and 2^{6} for Breast, Ionosphere, Wine, Coil-20, Caltech101 and 2D plus 3D palmprint databases, respectively. The above perception is extremely helpful for parameter

selection, that is, the parameter value can be approximately determined by finding which range of results is better.

4.5. Comparison of the Speed of the Algorithms

In this section, we conduct experiments to compare the speed of the proposed methods with the compared algorithms. To this end, we choose three datasets to test the speed of the algorithms. For a fair comparison, we set the d value as 2 in all experiments. The experiments are independently repeated 10 times and the average runtime is shown on Table 5. It can be seen that running speeds of PCA, LDA, DCA and RLDA are fast because they can directly obtain the projection matrix by the eigenvalue (or generalized eigenvalue) decomposition method. Our methods are slower than these algorithms. The reasons are: 1) AML and KAML work on pair-wise distances; 2) We need to adopt an alternate iterative strategy to optimize d_{ij} and $\boldsymbol{W}$. In addition, the running speed of MCML is the slowest. The main reason is the optimization of the objective function using the projected gradient method, which makes it necessary to recalculate the conditional distribution of each training point on other points at each iteration.

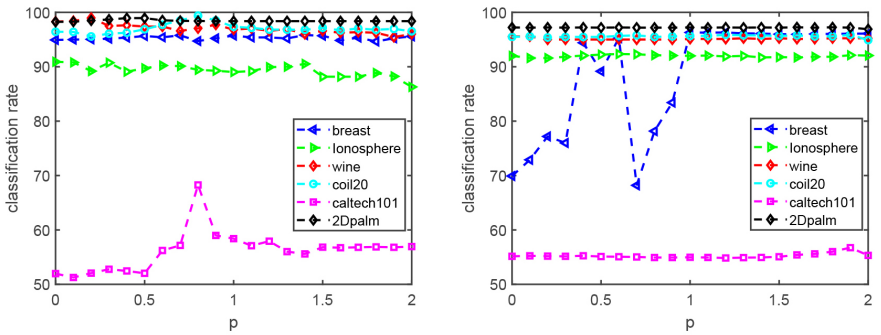

Figure 3. Classification rates of the proposed AML and KAML with different value of p. (a) AML. (b) KAML.

Table 5. Test time on UCI databases(s).

Methods	Breast	Ionosphere	Wine
PCA	0.018	0.006	0.002
LDA	0.044	0.015	0.004
DCA	0.026	0.010	0.008
MDL	0.263	0.025	0.017
LMNN	0.018	0.026	0.013
RLDA	0.029	0.015	0.008
MCML	18.862	22.956	3.230
NCA	5.644	2.497	0.198
AML	0.151	0.049	0.013
KAML	0.339	0.065	0.014

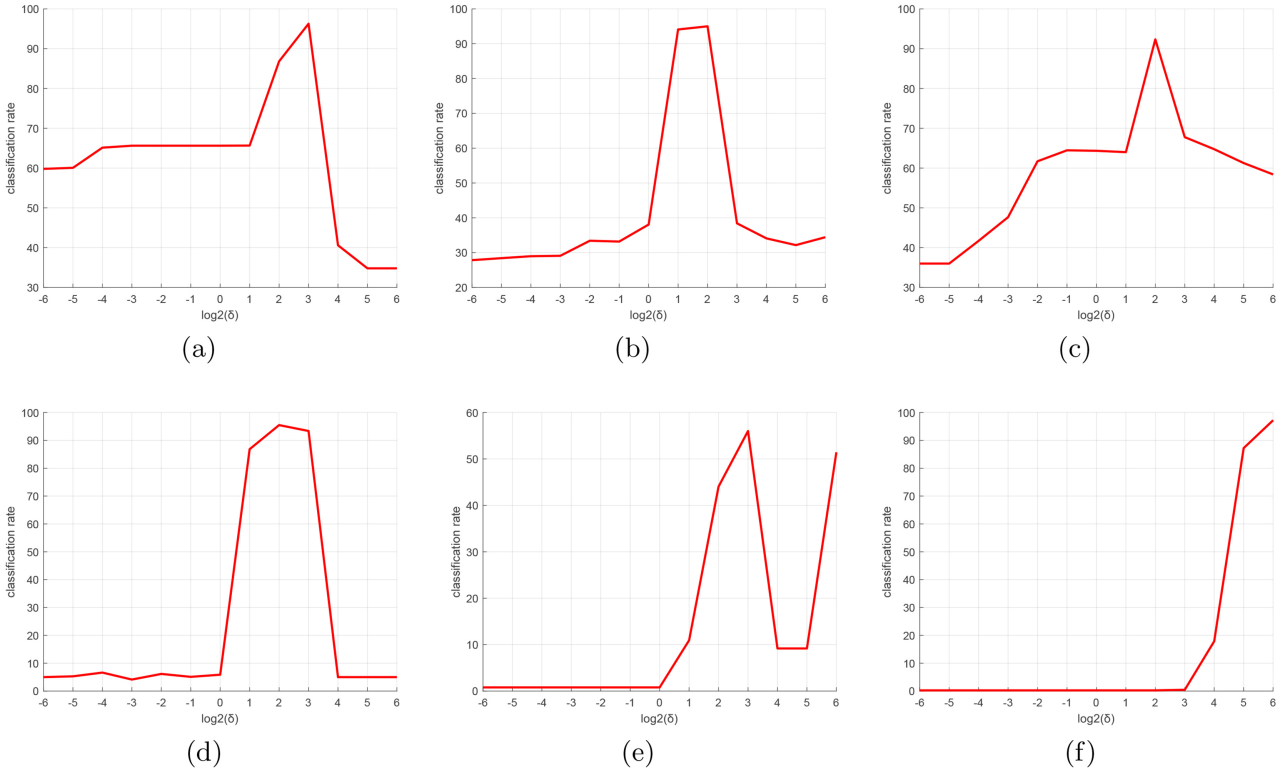

Figure 4. Classification rates of the proposed KAML with different value of σ on the different databases. (a) Breast. (b) Ionosphere. (c) Wine. (d) Coil20. (e) Caltech101. (f) 2D plus 3D palmprint.

5. Conclusions

In this paper, we present an adaptive metric learning method (AML), which integrates the $l_{2,p}$-norm distance metric and pairwise constraints into a unified framework. Specifically, the proposed method learns the distance or similarity based on $l_{2,p}$-norm rather than the traditional Euclidean, which enhances the flexibility and adaptability of metric learning. Meanwhile, we introduce the kernel technique to AML and propose a nonlinear metric learning method for dimensionality reduction. Furthermore, we present an effective optimization approach to deal with the new objective function. Extensive experiments show that the proposed approach could achieve competitive performance on metric learning tasks.

It is worth noting that the method in this paper can only manually select the parameter p for different data sets when determining the p value of the $l_{2,p}$-norm, and the size of the p value is determined by the performance of the algorithm at different p value. However, this is still not representative enough. Therefore, the effect of p value on the performance of the algorithm and the determination of p value remain key issues for future research. Can our distance metric model automatically adjust the parameter p according to the features of different data sets? If the answer is yes, how to design the scheme? In addition, the different initialization methods will result in different performances. How to select an effective initialization method is also a key issue. Our future work will

focus on these topics.

Acknowledgements

This work was supported by Science and Technology Planning Project of Guangzhou under Grant 202102020699.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] Barkalov, K., Shtanyuk, A. and Sysoyev, A. (2022) A Fast kNN Algorithm Using Multiple Space-Filling Curves. *Entropy*, **24**, 767. https://doi.org/10.3390/e24060767

[2] Qi, Q., Rong, J., Zhu, S. and Lin, Y. (2014) An Integrated Framework for High Dimensional Distance Metric Learning and Its Application to Fine-Grained Visual Categorization.

[3] Ren, Y.T. and Huang, Z.C. (2022) Distribution Probability-Based Self-Adaption Metric Learning for Person Re-Identification. *IET Computer Vision*, **16**, 376-387. https://doi.org/10.1049/cvi2.12094

[4] Zhang, J.A., Wang, Q. and Yuan, Y. (2019) Metric Learning by Simultaneously Learning Linear Transformation Matrix and Weight Matrix for Person Re-Identification. *IET Computer Vision*, **13**, 428-434. https://doi.org/10.1049/iet-cvi.2018.5402

[5] Liu, L.N., Lu, H.C. and Mei, X. (2016) Joint Learning Hash Codes and Distance Metric for Visual Tracking. 2016 *IEEE International Conference on Image Processing*, Phoenix, 25-28 September 2016, 1709-1713. https://doi.org/10.1109/ICIP.2016.7532650

[6] Li, X., Shen, C.H., Shi, Q.F., Dick, A. and Van Anton, D.H. (2012) Non-Sparse Linear Representations for Visual Tracking with Online Reservoir Metric Learning. 2012 *IEEE International Conference on Image Processing*, Orlando, 30 September-3 October 2012, 1760-1767.

[7] Hoi, S.C.H., Liu, W., Lyu, M.R. and Ma, W.Y. (2006) Learning Distance Metrics with Contextual Constraints for Image Retrieval. *IEEE Computer Society Conference on Computer Vision & Pattern Recognition*, New York, 17-22 June 2006, 2072-2078.

[8] Wang, F. and Sun, J.M. (2015) Survey on Distance Metric Learning and Dimensionality Reduction in Data Mining. *Data Mining and Knowledge Discovery*, **29**, 534-564. https://doi.org/10.1007/s10618-014-0356-z

[9] Wu, P.C., *et al.* (2016) Online Multi-Modal Distance Metric Learning with Application to Image Retrieval. *IEEE Transactions on Knowledge and Data Engineering*, **28**, 454-467. https://doi.org/10.1109/TKDE.2015.2477296

[10] Xiang, S., Nie, F. and Zhang, C. (2008) Learning a Mahalanobis Distance Metric for Data Clustering and Classification. *Pattern Recognition*, **41**, 3600-3612. https://doi.org/10.1016/j.patcog.2008.05.018

[11] Dong, M.Z., Wang, Y.J., Yang, X.C. and Xue, J.H. (2019) Learning Local Metrics and Influential Regions for Classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **42**, 1522-1529. https://doi.org/10.1109/TPAMI.2019.2914899

[12] Zhang, X.T., *et al.* (2020) Balancing Large Margin Nearest Neighbours for Imbalanced Data. *The Journal of Engineering*, **13**, 316-321. https://doi.org/10.1049/joe.2019.1178

[13] Nguyen, B. and De Baets, B. (2019) Kernel-Based Distance Metric Learning for Supervised *k*-Means Clustering. *IEEE Transactions on Neural Networks and Learning Systems*, **30**, 3084-3095. https://doi.org/10.1109/TNNLS.2018.2890021

[14] Heidari, N., Moslehi, Z., Mirzaei, A. and Safayani, M. (2018) Bayesian Distance Metric Learning for Discriminative Fuzzy c-Means Clustering. *Neurocomputing*, **319**, 21-33. https://doi.org/10.1016/j.neucom.2018.08.071

[15] Ye, J., Zhao, Z. and Liu, H. (2007) Adaptive Distance Metric Learning for Clustering. 2007 *IEEE Conference on Computer Vision and Pattern Recognition*, Minneapolis, 17-22 June 2007, 1-7. https://doi.org/10.1109/CVPR.2007.383103

[16] Dutta, U.K., Harandi, M. and Sekhar, C.C. (2020) Unsupervised Metric Learning with Synthetic Examples. *Proceedings of the AAAI Conference on Artificial Intelligence*, **34**, 3834-3841. https://doi.org/10.1609/aaai.v34i04.5795

[17] Wang, S.J., Xie, D.Y., Chen, F. and Gao, Q.X. (2018) Dimensionality Reduction by LPP-L21. *IET Computer Vision*, **12**, 659-665. https://doi.org/10.1049/iet-cvi.2017.0302

[18] Xu, Y., Lang, H. and Chai, X. (2019) Distribution Discrepancy Maximization Metric Learning for Ship Classification in Synthetic Aperture Radar Images. 2019 *IEEE International Geoscience and Remote Sensing Symposium*, Yokohama, 28 July-2 August 2019, 1208-1211. https://doi.org/10.1109/IGARSS.2019.8899173

[19] Wells, J.R., Aryal, S. and Ting, K.M. (2020) Simple Supervised Dissimilarity Measure: Bolstering iForest-Induced Similarity with Class Information without Learning. *Knowledge and Information Systems*, **62**, 3203-3216. https://doi.org/10.1007/s10115-020-01454-3

[20] Duan, Y., Liu, M. and Dong, M. (2020) A Metric-Learning-Based Nonlinear Modeling Algorithm and Its Application in Key-Performance-Indicator Prediction. *IEEE Transactions on Knowledge and Data Engineering*, **67**, 7073-7082. https://doi.org/10.1109/TIE.2019.2935979

[21] Dagnew, T.M. and Castellani, U. (2015) Supervised Learning of Diffusion Distance to Improve Histogram Matching. *International Workshop on Similarity-Based Pattern Recognition*, Copenhagen, 12-14 October 2015, 28-37. https://doi.org/10.1007/978-3-319-24261-3_3

[22] De, *et al.* (2017) Supervised Distance Metric Learning through Maximization of the Jeffrey Divergence. *Pattern Recognition*, **64**, 215-225. https://doi.org/10.1016/j.patcog.2016.11.010

[23] Goldberger, J. (2004) Neighbourhood Component Analysis. *Proceedings of the* 17*th International Conference on Neural Information Processing Systems*, Vancouver, December 2004, 513-520.

[24] Globerson, A. and Roweis, S.T. (2005) Metric Learning by Collapsing Classes. *NIPS* 2005, Vancouver, 5-8 December 2005, 451-458.

[25] Huang, K., Jin, R., Xu, Z. and Liu, C.L. (2020) Robust Metric Learning by Smooth Optimization. AUAI Press, Arlington.

[26] Liu, K., Brand, L., Wang, H. and Nie, F.P. (2019) Learning Robust Distance Metric with Side Information via Ratio Minimization of Orthogonally Constrained L21-Norm Distances. *Proceedings of the* 28*th International Joint Conference on Artificial Intelligence* (*IJCAI*-19), Macao, 10-16 August 2019, 3008-3014.

https://doi.org/10.24963/ijcai.2019/417

[27] Jiao, L.M., Geng, X.J. and Pan, Q. (2019) BP *k* NN: *k*-Nearest Neighbor Classifier with Pairwise Distance Metrics and Belief Function Theory. *IEEE Access*, **7**, 48935-48947. https://doi.org/10.1109/ACCESS.2019.2909752

[28] Liao, S., Gao, Q., Yang, Z., Fang, C., Nie, F. and Han, J. (2018) Discriminant Analysis via Joint Euler Transform and $L_{2,1}$-Norm. *IEEE Transactions on Image Processing*, **27**, 5668-5682. https://doi.org/10.1109/TIP.2018.2859589

[29] Yu, Y.F., Xu, G.X., Huang, K.K., Zhu, H., Chen, L. and Wang, H. (2020) Dual Calibration Mechanism Based $L_{2,p}$-Norm for Graph Matching. *IEEE Transactions on Circuits and Systems for Video Technology*, **31**, 2343-2358. https://doi.org/10.1109/TCSVT.2020.3023781

[30] Wang, Q., Gao, Q., Gao, X. and Nie, F. (2018) $L_{2,p}$-Norm Based PCA for Image Recognition. *IEEE Transactions on Image Processing*, **27**, 1336-1346. https://doi.org/10.1109/TIP.2017.2777184

[31] Mi, J.X., Zhang, Y.N., Li, Y. and Shu, Y. (2020) Generalized Two-Dimensional PCA Based on $L_{2,p}$-Norm Minimization. *International Journal of Machine Learning and Cybernetics*, **11**, 1-18. https://doi.org/10.1007/s13042-020-01127-1

[32] Fu, L., Li, Z., Ye, Q., Yin, H. and Yang, G. (2020) Learning Robust Discriminant Subspace Based on Joint $L_{2,p}$- and $L_{2,s}$-Norm Distance Metrics. *IEEE Transactions on Neural Networks and Learning Systems*, **33**, 130-144.

[33] Ma, Y., Niyogi, P., Sapiro, G. and Vidal, R. (2011) Dimensionality Reduction via Subspace and Submanifold Learning. *IEEE Signal Processing Magazine*, **28**, 14-126. https://doi.org/10.1109/MSP.2010.940005

[34] Turk, M. (1991) Eigenfaces for Recognition. *Journal of Cognitive Neuroscience*, **3**, 71-86. https://doi.org/10.1162/jocn.1991.3.1.71

[35] Belhumeur, P.N., Hespanha, J.P. and Kriegman, D.J. (1997) Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **19**, 711-720. https://doi.org/10.1109/34.598228

[36] Zhao, X.W., Guo, J., Nie, F.P., Chen, L., Li, Z.H. and Zhang, H.X. (2020) Joint Principal Component and Discriminant Analysis for Dimensionality Reduction. *IEEE Transactions on Neural Networks and Learning Systems*, **31**, 433-444. https://doi.org/10.1109/TNNLS.2019.2904701

[37] Zhao, H., Wang, Z. and Nie, F. (2018) A New Formulation of Linear Discriminant Analysis for Robust Dimensionality Reduction. *IEEE Transactions on Knowledge &; Data Engineering*, **31**, 629-640.

[38] Gao, Y.L., Zhong, S.X., Hu, K.L. and Pan, J.Y. (2020) Robust Locality Preserving Projections Using Angle-Based Adaptive Weight Method. *IET Computer Vision*, **14**, 605-613. https://doi.org/10.1049/iet-cvi.2019.0403

[39] Yu, W.W., Zhang, M. and Shen, Y. (2019) Learning a Local Manifold Representation Based on Improved Neighborhood Rough Set and LLE for Hyperspectral Dimensionality Reduction. *Signal Processing*, **164**, 20-29. https://doi.org/10.1016/j.sigpro.2019.05.034

[40] Niu, G. and Ma, Z.M. (2017) Tensor Local Linear Embedding with Global Subspace Projection Optimisation. *IET Computer Vision*, **16**, 241-254. https://doi.org/10.1049/cvi2.12083

[41] Schlkopf, B., Smola, A. and Mller, K. (1998) Nonlinear Component Analysis as a Kernel Eigenvalue Problem. *Neural Computation*, **10**, 1299-1319.

https://doi.org/10.1162/089976698300017467

[42] Deng, C., He, X. and Han, J. (2011) Speed Up Kernel Discriminant Analysis. *The VLDB Journal*, **20**, 21-33. https://doi.org/10.1007/s00778-010-0189-3

[43] Tao, H., Hou, C., Nie, F., Jiao, Y. and Yi, D. (2016) Effective Discriminative Feature Selection with Nontrivial Solution. *IEEE Transactions on Neural Networks and Learning Systems*, **27**, 796-808. https://doi.org/10.1109/TNNLS.2015.2424721

[44] Shawetaylor, J. (2005) Kernel Methods for Pattern Analysis.

[45] Baudat, G. and Anouar, F. (2000) Generalized Discriminant Analysis Using a Kernel Approach. *Neural Computation*, **12**, 2385-2404. https://doi.org/10.1162/089976600300014980

[46] Kai, Y., Tong, Z. and Gong, Y. (2009) Nonlinear Learning Using Local Coordinate Coding. *Advances in Neural Information Processing Systems* 22: 23*rd Annual Conference on Neural Information Processing Systems* 2009, Vancouver, 7-10 December 2009, 2223-2231.

[47] Li, F.F., Fergus, R. and Perona, P. (2004) Learning Generative Visual Models from Few Training Examples: An Incremental Bayesian Approach Tested on 101 Object Categories. *Conference on Computer Vision & Pattern Recognition Workshop*, Washington DC, 27 June-2 July 2004, 59-70.

[48] Weinberger, K.Q. and Saul, L.K. (2009) Distance Metric Learning for Large Margin nearest Neighbor Classification. *Journal of Machine Learning Research*, **10**, 207-244.

[49] HK-PolyU 2D + 3D Palmprint Database. http://www.comp.polyu.edu.hk/biometrics/2D_3D_Palmprint.htm