Multi-view Discriminant Analysis with Posterior Probability Graph Weighting

Pei Jihong[1,2], Lin Xiaoan[2], Li Meihua[2], Zhao Yang[3*], and Yang Xuan[3]

[1] Guangdong Key Laboratory of Intelligent Information Processing, Shenzhen University, Shenzhen, China
jhpei@szu.edu.cn
[2] College of Electronics and Information Engineering, Shenzhen University, Shenzhen, China
[3] College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, China
zhaoyang1990@szu.edu.cn

Abstract. Multi-view Discriminant Analysis is a supervised multi-view learning method widely used in pattern recognition fields. When one view's data gets affected, the labels may contain errors, making the label reliability of that view's dataset questionable. To address the above issue, the Multi-view Discriminant Analysis with Posterior Probability Graph Weighting (MvDA-PPG) method is proposed for reliable multi-view learning tasks in unreliable labeling environments. This method optimizes the posterior probability matrix by introducing the sample posterior probability graph. Moreover, it reduces the impact of data from unreliably labeled views on the performance of multi-view learning by adjusting the weights of some views. Experimental results demonstrate that the KMvDA-PPG method helps reduce the influence of unreliable labels by applying posterior probability weighting to samples containing unreliable labeled views. This improves classification accuracy and robustness. The paper conducted experiments on three widely used datasets. Theoretical analysis and experimental results indicate that the improved KMvDA-PPG algorithm exhibits excellent classification recognition performance.

Keywords: Multi-view discriminant analysis, Unreliable labeling environments, Consistency constraints, Posterior probability Graph

1 Introduction

In the era of big data, we often encounter datasets with multiple views. These views may originate from different data sources, measurement methods, or involve information from different domains. Typically, these views possess certain structure and features. However, using a single view for classification recognition can lead to overfitting and underfitting problems. Therefore, utilizing multi-view

*Corresponding author. Email: zhaoyang1990@szu.edu.cn

learning methods to extract information from each view and combine them effectively has become an active research area.

In single-view learning, Linear Discriminant Analysis (LDA)[1][2] is a widely used supervised learning method. Uncorrelated Linear Discriminant Analysis (ULDA)[3] is an extension of LDA, and Sun et al. proposed obtaining the optimal projection matrix with minimal redundant information by applying ULDA. The Multi-view Uncorrelated Linear Discriminant Analysis (MULDA)[4] combines the advantages of ULDA and CCA by extracting uncorrelated features within each view and projecting them onto a shared subspace. Later, the Generalized Multiview Analysis (GMA)[5] was proposed to incorporate supervised information, resulting in a discriminative common subspace. However, GMA focuses only on intra-view discriminant information while ignoring inter-view discriminant information. To address this issue, the Multi-view Discriminant Analysis (MvDA)[6] was introduced with the objective of maximizing class-wise distances across all views and minimizing class-wise within-distances to find optimal projection directions that map data onto a shared subspace. However, MvDA is limited to linear spaces, prompting the development of Kernel Multi-view Discriminant Analysis (KMvDA)[7]. Yang et al. proposed a Robust Multi-view Discriminant Analysis with Viewpoint Consistency (RMvDA-VC)[8] to tackle noisy data problems. Peng et al. proposed "Deep Supervised Multi-View Learning With Graph Priors"[11], which constructs a discriminative similarity graph based on multi-view input labels and pairwise relationships as prior knowledge. Nevertheless, in multi-resolution image classification tasks, low-resolution images can be difficult to differentiate due to label noise, which may degrade classification performance. Li et al. introduced Weighted Multi-view Discriminant Analysis (WMvDA)[9], assigning different weights to each view to enhance MvDA's accuracy under unreliable label environments.

In this paper, Multi-view Discriminant Analysis with Posterior Probability Graph Optimization (MvDA-PPG) is proposed. The algorithm constructs and optimizes a sample posterior probability graph to more finely describe the categorical properties of samples, thereby reducing the impact of potentially unreliable labels on performance. It also decreases the weights of views with unreliable labels to minimize their impact on overall learning performance. This method is extended to the kernel space to find more optimal linear relationships in high-dimensional feature spaces, effectively handling non-linear data distributions in original spaces and further improving classification performance.

In summary, our contributions are as follows:

(1) We propose a Multi-view Discriminant Analysis with Posterior Probability Graph Weighting (MvDA-PPG). This method leverages sample posterior probability information to build a graph model that optimizes the posterior probability matrix, making it closer to the true label distribution. Based on view-wise data reliability and consistency assessments, we adjust each view's weight to ensure reliable views play a more significant role in model training.

(2) On the basis of the unified framework proposed in Contribution (1), a new multi-view learning algorithm is further designed. The algorithm not only inher-

its the core idea of the framework, but also optimizes the processing of nonlinear data. We introduce Kernel Multi-view Discriminant Analysis with Posterior Probability Graph Weighting for unreliable label environments (KMvDA-PPG). This method effectively addresses non-linear data distribution issues in original spaces by finding optimal linear relationships in high-dimensional feature spaces, further enhancing classification performance.

2 Related Work

Multi-view Discriminant Analysis[6] extends traditional Linear Discriminant Analysis (LDA)[2] to multi-view data. Its core idea is to project multiple views of data into a shared latent subspace while maximizing class-wise distances and minimizing class-wise within-distances within that subspace.

Given c classes, there are v views of datasets $X_j \in \mathbf{R}^{\mathbf{d_j}} (j = 1, 2, ..., j, ..., v)$, where d_j represents the feature dimension size of the j-th view. The projection results can be represented as $y_{ijk} = W_j^T x_{ijk}$ $(i = 1, \cdots, c; j = 1, \cdots, v; k = 1, \cdots, n_{ij})$, where x_{ijk} denotes the k-th training sample in the i-th class of the j-th view, and W_j represents the projection matrix for the j-th view. The optimization target function of the MvDA algorithm can be represented as follows:

$$\max_{W_1, W_2, \cdots, W_v} \frac{Tr\left(S_B^y\right)}{Tr\left(S_W^y\right)}, \tag{1}$$

where S_W^y represents the within-class scatter matrix, and S_B^y represents the between-class scatter matrix.

In MvDA[6], view consistency constraints were introduced to obtain the optimization objective function of Multi-view Discriminant Analysis with view consistency:

$$\max_{W_1, W_2, \cdots, W_v} \frac{Tr\left(S_B^y\right)}{Tr\left(S_W^y\right) + \lambda \sum_{j,l=1}^{v} ||\beta_j - \beta_l||_2^2}, \tag{2}$$

where λ is the balance parameter. In view consistency constraints $\sum_{j,l=1}^{v} ||\beta_j - \beta_l||_2^2$, β_j is the feature matrix extracted from the projection matrix W_j.

The between-class scatter matrix mainly measures the differences between centers of different categories, while the within-class scatter matrix reflects the differences between samples within the same category. By optimizing the ratio of within-class and between-class scatter matrices, MvDA can find a projection direction that best distinguishes different categories, significantly improving classification and recognition accuracy. However, in environments with unreliable labels, views containing unreliable labels will affect the projection matrix. Moreover, within the same view, the importance of information contained in each sample varies. Unlike MvDA which does not consider the different importance degrees of samples to their respective categories while calculating between-class and within-class scatter matrices, this paper proposes a method that introduces posterior probability matrix as sample weights, thereby reducing the impact of unreliable labels on model training and enhancing model robustness.

3 Multi-view Discriminant Analysis with Posterior Probability Graph Weighting

This section introduces the general framework, target function construction, and optimization process of Multi-view Discriminant Analysis with Posterior Probability Graph Weighting in unreliable label environments, as well as its extension to kernel methods.

3.1 Overview

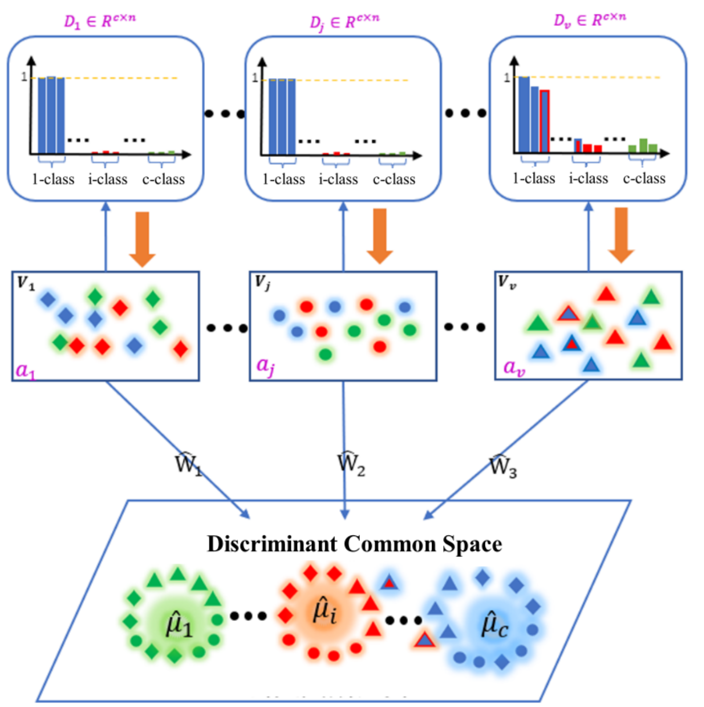

Fig. 1. Algorithm Diagram of Multi-view Discriminant Analysis with Posterior Probability Graph Weighting

As shown in Fig 1, MvDA-PPG utilizes the similarity between different samples to construct a posterior probability graph matrix for views with unreliable labels. It uses the distributional information provided by the posterior probability matrix as sample weights to correct for unreliable labels. Then, it assigns different weights to each view based on the importance of information in different views. Based on the weights of different views, it constructs weighted multi-view between-class and within-class scatter matrices and projects samples from v different views into the same latent subspace using v linear transformations. After projection, samples of the same class are grouped as closely together as possible, while samples of different classes are moved as far apart from each other as possible.

MvDA-PPG projects samples from different views into a common discriminant subspace by applying a set of linear transformations. Each view corresponds to one transformation. Here, D_j is the posterior probability matrix for the j-th view. Different colors represent different classes, and different shapes represent different views.

3.2 Model of MvDA-PPG

Define a set $X_j = \{x_{ijk} | i = 1, \cdots, c; k = 1, \cdots, n_{ij}; j = 1, \cdots, v\}$, where the projection result is denoted as $Y_j = \{W_j^T x_{ijk} | i = 1, \cdots, c; k = 1, \cdots, n_{ij}; j = 1, \cdots, v\}$. Here, x_{ijk} represents the k-th sample of the i-th class in the j-th view. Let c be the number of classes, v the number of views, and n_{ij} the number of samples in the i-th class for the j-th view. Then, the optimization objective function of the MvDA-PPG algorithm can be represented as follows:

$$\max_{\widehat{W}_1, \widehat{W}_2, \cdots, \widehat{W}_v} \frac{Tr\left(\hat{S}_B^y\right)}{Tr\left(\hat{S}_W^y\right) + \lambda \sum_{j,l=1}^{v} ||\beta_j - \beta_l||_2^2}, \tag{3}$$

where $\hat{S}_W^y$ and $\hat{S}_B^y$ denote the weighted within-class and between-class scatter matrices obtained using multiple views' data, respectively. β_j is the feature matrix extracted from the projection matrix $W_j = X_j \beta_j$. Weighted within-class scatter matrix $\hat{S}_W^y$ and weighted between-class scatter matrix $\hat{S}_B^y$ are defined as follows:

$$\hat{S}_W^y = \sum_{i=1}^{c} \sum_{j=1}^{v} a_j \sum_{k=1}^{n_{ij}} \left(y_{ijk} - \hat{\mu}_i\right) \left(y_{ijk} - \hat{\mu}_i\right)^T, \tag{4}$$

$$\hat{S}_B^y = \sum_{i=1}^{c} n_i \left(\hat{\mu}_i - \hat{\mu}\right)(\hat{\mu}_i - \hat{\mu})^T. \tag{5}$$

In Eq 4, a_j is the weight of the j-th view, and $\hat{\mu}$ and $\hat{\mu}_i$ represent the weighted mean of all samples and the weighted mean of the i-th class after projection, which can be calculated as follows:

$$\hat{\mu}_i = \sum_{j=1}^{v} a_j \frac{1}{n_{ij}} \sum_{t=1}^{n} d_j^{it} y_{ijt}, \tag{6}$$

$$\hat{\mu} = \frac{1}{c} \sum_{i=1}^{c} \sum_{j=1}^{v} a_j \frac{1}{n_{ij}} \sum_{t=1}^{n} d_j^{it} y_{ijt}. \tag{7}$$

MvDA-PPG finds a projection direction by solving the optimization Eq 3, maximizing between-class scatter and minimizing within-class scatter, while ensuring that the main projection vector passes through the densest reliable data points. To suppress the influence of unreliable labeled samples, we introduce sample posterior probability graphs and adjust the weights of unreliable labeled views to penalize the sample space.

The posterior probability matrix $D_j \in \mathbf{R^{c\times n}}$ provides the probability of each sample belonging to different classes. It is used as the weighted value for samples to obtain the weighted class centers $\hat{\mu}_i$ and overall sample center $\hat{\mu}$. For views containing unreliable labels, the posterior probability matrix D_j can be obtained through the following optimization problem:

$$\min_{D_j} ||X_j - X_j G_j||_F^2 + \alpha ||L_j - D_j||_F^2, \quad s.t.\ D_j \geq 0\,, \tag{8}$$

where $G_j = D_j^T D_j B_j$ is the posterior probability graph matrix, $D_j = (D_j^1, D_j^2, ..., D_j^n)$ is the posterior probability matrix, and Matrix B_j is the normalized matrix of correlation matrix $R = D_j^T D_j$. D_j^k indicates the probability distribution of the k-th sample belonging to different categories. α is a balancing parameter. $L_j \in \mathbf{R^{c\times n}}$ is the label matrix after one-hot encoding of the original labels.

Let $Y_j = X_j D_j^T D_j B_j$, then Y is represented as:

$$\left[y_j^1, y_j^2, ..., y_j^n\right] = \left[\sum_{k=1}^{n} x_j^k G_{k1}, \sum_{k=1}^{n} x_j^k G_{k2}, ..., \sum_{k=1}^{n} x_j^k G_{kn}\right], \tag{9}$$

where y_j is the j-th element in Y. y_j represents the weighted sum of the relevance between the k-th sample and its posterior probability with other samples, and G_{ki} denotes the relevance between the k-th sample and its posterior probability with other samples. Specifically, for sample x_j^m, if there exists another sample x_j^n that belongs to the same category as x_j^m, then their relevance $[R]_{mn}$ will be relatively large. This larger relevance makes a greater contribution in the weighted sum process, making the smoothed result more similar to the original sample x_j^m. Therefore, this method reduces the error caused by considering only sample a and improves overall accuracy and robustness.

Using the update formula with multiplication to optimize the objective Eq 8, we obtain the following update formula:

$$N = 2D_j^{(t)} B_j X_j^T X_j + D_j^{(t)} X_j^T X_j B_j^T + \alpha L_j, \tag{10}$$

$$D = D_j^{(t)} X_j^T X_j D_j^{(t)^T} D_j^{(t)} B_j B_j^T + D_j^{(t)} B_j B_j^T D_j^{(t)^T} D_j^{(t)} X_j^T X_j + \alpha D_j^{(t)}, \tag{11}$$

$$D_j^{(t+1)} = D_j^{(t)} \odot (N/D), \tag{12}$$

$$D_j^{(t+1)} = D_j^{(t+1)} \cdot diag\left[\left[\left[D_j^{(t+1)}\right]^T \mathbf{1}_n\right]^T\right]^{-1}, \tag{13}$$

where $\odot$ denotes element-wise multiplication. Eq 13 is for normalizing D_j obtained in each iteration. The posterior probability matrix D_j provides the probability of each sample belonging to different classes, which can be used as reliability weights for sample categories.

Upon simplifying Eq 4, the view-weighted within-class scatter matrix $\hat{S}_W^y$ can be expressed in matrix form:

$$\hat{S}_W^y = \left[\widehat{W}_1^T, ..., \widehat{W}_v^T\right] \begin{bmatrix} \widehat{\Psi}_{11} & \cdots & \widehat{\Psi}_{1v} \\ \vdots & \ddots & \vdots \\ \widehat{\Psi}_{1v} & \cdots & \widehat{\Psi}_{vv} \end{bmatrix} \begin{bmatrix} \widehat{W}_1 \\ \vdots \\ \widehat{W}_v \end{bmatrix} = \widehat{W}^T \widehat{\Psi} \widehat{W}, \tag{14}$$

where matrix $\hat{\Psi}$ is the weighted within-class scatter coefficient matrix, with each block $\widehat{\Psi}_{jl}$ having the following form:

$$\hat{\Psi}_{jl} = \begin{cases} \sum_{i=1}^{c}(a_j \sum_{k=1}^{n_{ij}} x_{ijk} x_{ijk}^T - a_j a_j n_{ij} \mu_{ij}^{x1} \mu_{ij}^{x2^T} \\ -a_j a_j n_{ij} \mu_{ij}^{x2} \mu_{ij}^{x1^T} + a_j a_j n_{ij} \mu_{ij}^{x2} \mu_{ij}^{x2^T}), & j = l \\ \\ \sum_{i=1}^{c} \left(a_j a_l n_{ij} \mu_{ij}^{x2} \mu_{ii}^{x2^T} - a_j a_l n_{ij} \mu_{ij}^{x1} \mu_{ii}^{x2^T} - a_j a_l n_{ij} \mu_{ij}^{x2} \mu_{ii}^{x1^T} \right). & j \neq l \end{cases} \tag{15}$$

Similarly, the view-weighted between-class scatter matrix $\hat{S}_B^y$ can be expressed as follows:

$$\hat{S}_B^y = \left[\widehat{W}_1^T, ..., \widehat{W}_v^T\right] \begin{bmatrix} \widehat{B}_{11} & \cdots & \widehat{B}_{1v} \\ \vdots & \ddots & \vdots \\ \widehat{B}_{1v} & \cdots & \widehat{B}_{vv} \end{bmatrix} \begin{bmatrix} \widehat{W}_1 \\ \vdots \\ \widehat{W}_v \end{bmatrix} = \widehat{W}^T \widehat{B} \widehat{W}, \tag{16}$$

where matrix $\hat{B}$ is the weighted within-class scatter coefficient matrix, with each block $\hat{B}_{jl}$ having the following form:

$$\hat{B}_{jl} = \sum_{i=1}^{c} n_i a_j a_l \mu_{il}^{x2} \mu_{il}^{x2^T} - \frac{n_i}{c} \sum_{i=1}^{c} a_j \mu_{ij}^{x2} \sum_{i=1}^{c} a_l \mu_{il}^{x2^T}. \tag{17}$$

In Eq 3, consistency constraints across views $\sum_{j,l=1}^{v} ||\beta_j - \beta_l||_2^2 = Tr\left(\widehat{W}^T M \widehat{W}\right)$, where matrix M is:

$$M = \begin{bmatrix} M_{11} & \cdots & M_{1v} \\ \vdots & \ddots & \vdots \\ M_{v1} & \cdots & M_{vv} \end{bmatrix}. \tag{18}$$

The elements in matrix M are:

$$M_{jl} = \begin{cases} 2(v-1) P_j^T P_j, & j = l \\ -2 P_j^T P_l, & j \neq l \end{cases}, \tag{19}$$

where $P_j = \left(X_j^T X_j\right)^{-1} X_j^T$. Therefore, its optimization problem can be rewritten as follows:

$$\max_{\widehat{W}_1,\widehat{W}_2,\cdots,\widehat{W}_v} \frac{Tr\left(\widehat{W}^T \hat{B} \widehat{W}\right)}{Tr\left(\widehat{W}^T \left(\widehat{\Psi} + \lambda M\right) \widehat{W}\right)}. \tag{20}$$

Thus, Eq 20 can be solved using generalized eigenvalue decomposition:

$$\hat{B}\widehat{W} = \gamma\left(\widehat{\Psi} + \lambda M\right)\widehat{W}. \tag{21}$$

Solving Eq 21 yields n eigenvalues and their corresponding eigenvectors. The top d largest eigenvalues' eigenvectors are selected as projection directions, and the samples are projected into a shared d-dimensional subspace.

3.3 Kernel MvDA-PPG

To extend MvDA-PPG to non-linear spaces, we map the original image data X using $\varphi\colon \mathbf{R^d} \to \mathcal{F}$. In the kernel feature space, samples are represented as $\varphi(X_j)$, and the projection matrix is given by $W_j^T = \xi_j^T \varphi(X_j)^T$. Therefore, when the model in Eq 3 is extended to the kernel space, we obtain a kernel Multi-view Discriminant Analysis with Posterior Probability Graph Weighting, which can be formulated as follows:

$$\max_{\hat{\xi}_1,\hat{\xi}_2,\cdots,\hat{\xi}_v} \frac{Tr\left(\hat{\xi}^T \hat{B} \hat{\xi}\right)}{Tr\left(\hat{\xi}^T \left(\hat{\Psi} + \lambda M\right) \hat{\xi}\right)}. \tag{22}$$

In the kernel space, the mean of original data and weighted class mean are represented as:

$$\mu_{ij}^{x1} = \frac{1}{n_{ij}} \sum_{k=1}^{n_{ij}} \varphi\left(x_{ijk}\right), \tag{23}$$

$$\mu_{ij}^{x2} = \frac{1}{n_{ij}} \sum_{t=1}^{n} d_j^{it} \varphi\left(x_{ijt}\right). \tag{24}$$

$\hat{\Psi}$ and $\hat{B}_{jl}$ are block matrices, and the blocks $\hat{\Psi}_{jl}$ and $\hat{B}_{jl}$ can be simplified as follows:

$$\widehat{\Psi}_{jl} = \begin{cases} \sum_{i=1}^{c}\big(a_j \sum_{k=1}^{n_{ij}} \varphi\left(x_{ijk}\right)\varphi\left(x_{ijk}\right)^T - a_j a_j n_{ij} \mu_{ij}^{x1} \mu_{ij}^{x2^T} \\ -a_j a_j n_i \mu_{ij}^{x2} \mu_{ij}^{xT} + a_j a_j n_{ij} \mu_{ij}^{x2} \mu_{ij}^{x2^T} \big), & j = l \\ \\ \sum_{i=1}^{c} \left(a_j a_l n_{ij} \mu_{ij}^{x2} \mu_{il}^{x2^T} - a_j a_l n_{ij} \mu_{ij}^{x1} \mu_{il}^{x2^T} - a_j a_l n_{ij} \mu_{ij}^{x2} \mu_{il}^{x1^T} \right), & j \neq l \end{cases} \tag{25}$$

$$\hat{B}_{jl} = \sum_{i=1}^{c} n_i a_j a_l \mu_{il}^{x2} \mu_{il}^{x2^T} - \frac{n_i}{c} \sum_{i=1}^{c} a_j \mu_{ij}^{x2} \sum_{i=1}^{c} a_l \, \mu_{il}^{x2^T}. \tag{26}$$

Eq 22 can be solved using generalized eigenvalue decomposition:

$$\hat{B}\hat{\hat{\xi}} = \gamma\big(\hat{\Psi} + \lambda M\big)\hat{\hat{\xi}}. \tag{27}$$

Solving Eq 27 yields n eigenvalues and their corresponding eigenvectors. The first d largest eigenvalues' eigenvectors are then chosen as the projection directions, and samples are projected into a shared d-dimensional subspace.

Table 1 outlines the steps for kernel Multi-view Discriminant Analysis with Posterior Probability Graph Weighting in unreliable label environments.

Table 1. kernel Multi-view Discriminant Analysis with Posterior Probability Graph Weighting (KMvDA-PPG)

Algorithm 1: KMvDA-PPG.
Input: Datasets X and corresponding labels; Parameters a_j with $\sum_{j=1}^{v} a_j = 1$; Gaussian kernel parameter σ balancing parameter α. **Output:** Projection matrices $\xi = \xi_1, \xi_j, \cdots, \xi_v$ and Classification accuracy of test samples. **Step 1:** Preprocess data, randomly divide into training and testing sets, initialize models. **Step 2:** Construct posterior probability relevance graph using Eq 12. **Step 3:** Optimize using Eq 13 to obtain posterior probability matrix D_j. **Step 4:** Calculate sample numbers n_{ij} and kernel function mappings K_j. **Step 5:** Compute nonlinear mapped within-class scatter matrix $\hat{\psi}$ using Eq 25. **Step 6:** Compute nonlinear mapped between-class scatter matrix $\hat{B}$ using Eq 26. **Step 7:** Solve for projection matrices ξ using Eq 27. **Step 8:** Project training and test samples onto a common space using projection matrices. **Step 9:** Classify test samples using k-nearest neighbor(KNN) classifier. **Step 10:** Calculate classification accuracy of test samples.

4 Experiments

In this section, we demonstrate the classification performance of the proposed Kernel Multi-view Discriminant Analysis with Posterior Probability Graph Weighting in unreliable label environments (KMvDA-PPG). First, we introduce the datasets and parameter settings. Then, we compare this algorithm with other multi-view classification algorithms and analyze the classification results. Next, we conduct experiments using different kernel functions to validate the advantage of the chosen Gaussian kernel function. Finally, we perform ablation studies to verify the necessity of each module in the algorithm.

4.1 Datasets and Experimental Settings

This paper utilizes publicly available datasets: Coil-20, MNIST and MFD. Coil-20 dataset contains 20 objects with 1440 grayscale images taken at various angles, resized to 32×32, 16×16, and 8×8 pixels for three different views. MNIST dataset comprises 70,000 handwritten digits, resized to 28×28, 14×14, and 7×7 pixels for three views. MFD dataset mainly consists of handwritten digits totaling 2000 samples. In the experiment, the first view uses a 216-dimensional contour-related feature, the second view uses a 64-dimensional KL transformation coefficient feature, and the third view uses pixel features, with each image downsized to 7×8 pixels.

In the experiment, the datasets were first processed to obtain the required three views for the experiments. Then, all views of the dataset were preprocessed into standardized data format. Next, the dataset was randomly divided into training and testing sets, with randomly selected 80% of total samples used for training and the rest for testing.

For the balancing parameter α, Gaussian kernel parameter σ, and each view's weight a_j, they are set as follows: The label constraint balancing parameter α is suggested to be traversed within the (0,1) range. The Gaussian kernel function parameter σ is suggested to be traversed within the [2,4] range, with optimal values found through multiple experiments. Each view's weight parameter a_j is traversed within the (0,1) range, with $\sum_{j=1}^{v} a_j = 1$. By considering the importance of different views and assigning weights accordingly, optimal values were found to better control the weight size of each view participating in training. The bold font is used to denote the optimal performance, while the values enclosed in parentheses represent the standard deviation of the replicated experiments.

4.2 Image Classification Results in Unreliable Labeling Environment

To validate the performance of our algorithm, we compared the proposed MvDA-PPG and KMvDA-PPG with previous methods WMvDA[9], KWMvDA[9], MvDA[6], RMvDA[8], DMVTSVM[10], and GMLDA[5] on MNIST, Coil-20, MFD datasets in terms of classification accuracy and standard deviation. All methods were used with optimal parameter settings. The experiments mainly investigate the relationship between the number of unreliable labels for a specific class in a view and recognition rate. During training, we randomly select some data from the low-resolution view of one class and set them as unreliable labels. The results are shown in Table 2, Table 3 and Table 4, where PUL represents the proportion of unreliable labels in the low-resolution view.

The experimental results indicate that our proposed KMvDA-PPG method demonstrates high classification accuracy across different datasets and varying numbers of unreliable labels. This highlights the ability of KMvDA-PPG to extract more feature information, performing well in environments with unreliable labels.

As the number of unreliable labels increases, classification accuracy gradually decreases, demonstrating a significant impact on classification recognition. How-

Table 2. Different PUL in Unreliable Labels Environment on MNIST Dataset

PUL	0.1	0.2	0.4
MvDA[6]	77.85(3.77)	76.55(3.97)	71.80(5.24)
GMLDA[5]	77.30(7.06)	76.37(5.70)	70.88(2.56)
RMvDA[8]	81.37(6.70)	77.75(5.35)	72.12(3.56)
DMVTSVM[10]	83.50(2.69)	82.95(2.97)	79.00(2.64)
WMvDA[9]	79.20(4.15)	76.88(5.50)	73.83(6.26)
KWMvDA[9]	84.93(3.15)	83.03(2.50)	79.38(3.06)
MvDA-PPG	79.58(2.36)	78.00(2.56)	74.78(1.83)
KMvDA-PPG	**85.03(2.52)**	**83.25(2.76)**	**79.88(2.01)**

Table 3. Different PUL in Unreliable Labels Environment on Coil-20 Dataset

PUL	0.1	0.2	0.4
MvDA[6]	87.53(5.50)	87.30(5.51)	85.73(6.09)
GMLDA[5]	77.37(5.73)	74.70(7.25)	70.00(8.54))
RMvDA[8]	91.53(4.50)	90.38(5.15)	88.73(2.94)
DMVTSVM[10]	92.98(3.20)	92.85(3.24)	92.12(3.40)
WMvDA[9]	91.20(4.20)	91.23(3.87)	89.43(5.30)
KWMvDA[9]	98.62(1.60)	97.63(3.15)	95.87(2.32)
MvDA-PPG	98.57(0.72)	97.56(0.82)	95.80(1.24)
KMvDA-PPG	**98.76(1.32)**	**97.80(0.72)**	**95.95(1.26)**

Table 4. Different PUL in Unreliable Labels Environment on MFD Dataset

PUL	0.1	0.2	0.4
MvDA[6]	90.30(1.62)	88.48(2.37)	80.25(3.07)
GMLDA[5]	90.60(1.78)	87.58(1.63)	79.28(2.42)
RMvDA[8]	90.35(1.28)	87.52(1.70)	88.20(2.56)
DMVTSVM[10]	92.57(1.35)	90.95(0.83)	89.12(0.97)
WMvDA[9]	92.48(1.34)	91.53(1.11)	88.20(1.26)
KWMvDA[9]	95.15(1.24)	93.48(1.43)	89.70(1.24)
MvDA-PPG	93.10(1.50)	91.75(1.82)	**89.92(1.35)**
KMvDA-PPG	**95.38(1.59)**	**93.63(1.84)**	89.83(1.46)

ever, our proposed KMvDA-PPG method maintains high classification precision compared to other methods, especially when compared to MvDA. Moreover, it has a relatively low standard deviation, indicating more stable performance across multiple experiments.

In the KMvDA-PPG method, we primarily employ the RBF kernel function to map original images into a high-dimensional RKHS space, enabling samples to exhibit linearity in this high-dimensional space. We evaluated the impact of linear kernel functions, RBF kernel functions, and polynomial kernel functions on image classification performance using MNIST, Coil-20, MFD datasets, as shown in Table 5.

Table 5. Classification Accuracy of Different Kernel Functions in the KMvDA-PPG Method on Three Datasets

	MNIST	Coil-20	MFD
Linear kernel	54.75(3.60)	61.33(3.24)	89.40(2.20)
Polynomial kernel	80.25(3.76)	97.40(2.55)	92.27(1.48)
RBF kernel	**85.03(2.52)**	**97.80(0.72)**	**93.63(1.84)**

The experimental results show that the classification performance of KMvDA-PPG using RBF kernel function outperforms those using linear and polynomial kernels. This indicates that there exists non-linearity in images, making it difficult for linear methods to extract certain features. Nonlinear mapping enables samples to exhibit linearity in high-dimensional space, allowing more information extraction for superior recognition. To validate the effectiveness of KMvDA-PPG for classification tasks proposed in this paper, experiments were conducted comparing KNN classifiers, SVM, and DNN classifiers. The results using MFD dataset for WMvDA, MvDA-PPG, KWMvDA, and KMvDA-PPG methods are shown in Table 6, the PUL parameter is set to 0.2.

Table 6. Classification Accuracy of Different Classifiers on MFD Dataset

	WMvDA	MvDA-PPG	KWMvDA	KMvDA-PPG
SVM classifier	88.95(2.80)	89.00(3.06)	90.25(2.65)	90.40(2.80)
DNN classifier	90.98(1.50)	81.27(2.06)	93.40(2.34)	93.27(1.78)
KNN classifier	**91.53(1.11)**	**91.75(1.82)**	**93.48(1.43)**	**93.63(1.84)**

Upon comparison of the classification results of KNN classifier, SVM classifier, and DNN classifier on MFD dataset, it was observed that KMvDA-PPG method outperformed other classifiers when using KNN classifier for classification. Additionally, both KNN classifier and DNN classifier are easy to implement and require no training, thus verifying the suitability of choosing KNN classifier.

5 Conclusion

This paper proposes two novel methods: Multi-view Discriminant Analysis with Posterior Probability Graph Weighting (MvDA-PPG) and Kernel Multi-view Discriminant Analysis with Posterior Probability Graph Weighting (KMvDA-PPG). Initially, for views with unreliable labels, a sample posterior probability graph matrix is constructed by leveraging the similarity among different samples, aiming to optimize the posterior probability matrix. In contrast, for the remaining views, the one-hot encoding of sample labels is utilized as their posterior probability matrix. Subsequently, weights are allocated to different views in accordance with the significance of view information. Furthermore, reconstructed Reproducing Kernel Hilbert Space (RKHS) multi-view within-class and between-class scatter matrices are established. The optimal projection direction is then determined based on these scatter matrices. Eventually, classification tasks are executed within the common subspace. Experimental results on the MNIST, Coil-20, and MFD datasets demonstrate that the proposed KMvDA-PPG effectively tackles the problem of unreliable labels in multi-view learning. By integrating posterior probability graph optimization, view weight adjustment, and the kernel method, it enhances the classification performance and robustness of the model in comparison to other existing methods.

6 Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (62201355, 62071303), Guangdong Basic and Applied Basic Research Foundation (2024A1515010977), Shenzhen Science and Technology Projection (JCYJ20220531102407018), Guangdong Provincial Key Laboratory (2023B1212060076), Guangdong Province Key Laboratory of Popular High Performance Computers (2017B030314073).

References

1. Friedman, M. and Kandel, A.: Introduction to pattern recognition: statistical, structural, neural, and fuzzy logic approaches. World Scientific, Singapore (1999)
2. Wang, H., Lu, X., Hu, Z. et al.: Fisher discriminant analysis with L1-norm. IEEE Transactions on Cybernetics 44(6), 828–842 (2013)
3. Zhang, X. and Chu, D.: Sparse uncorrelated linear discriminant analysis. In: International Conference on Machine Learning, pp. 45–52. PMLR (2013)
4. Yang, M. and Sun, S.: Multi-view uncorrelated linear discriminant analysis with applications to handwritten digit recognition. In: 2014 International Joint Conference on Neural Networks (IJCNN), pp. 4175–4181. IEEE (2014)
5. Sharma, A., Kumar, A., Daume, H. et al.: Generalized multiview analysis: A discriminative latent space. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition, pp. 2160–2167. IEEE (2012)
6. Kan, M., Shan, S., Zhang, H. et al.: Multi-view discriminant analysis. IEEE Transactions on Pattern Analysis and Machine Intelligence 38(1), 188–194 (2015)

7. Li, X., Gui, J., Li, P.: Randomized kernel multi-view discriminant analysis. arXiv preprint arXiv:2004.01143 (2020)
8. Yang, X.F., Li, C.N., Shao, Y.H.: Robust multi-view discriminant analysis with view-consistency. Information Sciences 596, 153–168 (2022)
9. Li, M., Zhao, Y., Pei, J. et al.: Weighted consistent multi-view discriminant analysis in unreliable labeling environment. In: International Conference on AI Logic and Applications, pp. 271–280. Springer Nature Singapore, Singapore (2023)
10. Xie, X.J., Li, Y., Sun, S.: Deep multi-view multiclass twin support vector machines. Information Fusion 91, 80–92 (2023)
11. Hu, P., Zhen, L., Peng, X., Zhu, H., Lin, J., Wang, X., Peng, D.: Deep Supervised Multi-View Learning With Graph Priors. IEEE Transactions on Image Processing 33, 123–133 (2024)