Fisher and Kernel Fisher Discriminant Analysis: Tutorial

Benyamin Ghojogh BGHOJOGH@UWATERLOO.CA
Department of Electrical and Computer Engineering,
Machine Learning Laboratory, University of Waterloo, Waterloo, ON, Canada

Fakhri Karray KARRAY@UWATERLOO.CA
Department of Electrical and Computer Engineering,
Centre for Pattern Analysis and Machine Intelligence, University of Waterloo, Waterloo, ON, Canada

Mark Crowley MCROWLEY@UWATERLOO.CA
Department of Electrical and Computer Engineering,
Machine Learning Laboratory, University of Waterloo, Waterloo, ON, Canada

Abstract

This is a detailed tutorial paper which explains the Fisher discriminant Analysis (FDA) and kernel FDA. We start with projection and reconstruction. Then, one- and multi-dimensional FDA subspaces are covered. Scatters in two- and then multi-classes are explained in FDA. Then, we discuss on the rank of the scatters and the dimensionality of the subspace. A real-life example is also provided for interpreting FDA. Then, possible singularity of the scatter is discussed to introduce robust FDA. PCA and FDA directions are also compared. We also prove that FDA and linear discriminant analysis are equivalent. Fisher forest is also introduced as an ensemble of fisher subspaces useful for handling data with different features and dimensionality. Afterwards, kernel FDA is explained for both one- and multi-dimensional subspaces with both two- and multi-classes. Finally, some simulations are performed on AT&T face dataset to illustrate FDA and compare it with PCA.

1. Introduction

Assume we have a dataset of *instances* or *data points* $\{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^n$ with sample size n and dimensionality $\boldsymbol{x}_i \in \mathbb{R}^d$ and $\boldsymbol{y}_i \in \mathbb{R}^\ell$. The $\{\boldsymbol{x}_i\}_{i=1}^n$ are the input data to the model and the $\{\boldsymbol{y}_i\}_{i=1}^n$ are the observations (labels). We define $\mathbb{R}^{d\times n} \ni \boldsymbol{X} := [\boldsymbol{x}_1, \dots, \boldsymbol{x}_n]$ and $\mathbb{R}^{\ell\times n} \ni \boldsymbol{Y} := [\boldsymbol{y}_1, \dots, \boldsymbol{y}_n]$. We can also have an out-of-sample data point, $\boldsymbol{x}_t \in \mathbb{R}^d$, which is not in the training set. If there are n_t out-of-sample data points, $\{\boldsymbol{x}_{t,i}\}_1^{n_t}$, we define $\mathbb{R}^{d\times n_t} \ni \boldsymbol{X}_t := [\boldsymbol{x}_{t,1}, \dots, \boldsymbol{x}_{t,n_t}]$. Usually, the data points exist on a subspace or sub-manifold. Subspace or manifold learning tries to learn this sub-manifold (Ghojogh et al., 2019b).

Here, we consider the case where the observations $\{\boldsymbol{y}_i\}_{i=1}^n$ come from a discrete set so that the task is *classification*. Assume the dataset consists of c classes, $\{\boldsymbol{x}_i^{(1)}\}_{i=1}^{n_1}, \dots, \{\boldsymbol{x}_i^{(c)}\}_{i=1}^{n_c}$ where n_j denotes the sample size (cardinality) of the the j-th class.

We want to find a subspace (or sub-manifold) which separates the classes as much as possible while the data also become as spread as possible. Fisher Discriminant Analysis (FDA) (Friedman et al., 2009) pursues this goal. It was first proposed in (Fisher, 1936) by Sir. Ronald Aylmer Fisher (1890 – 1962) who was a genius in statistics. He proposed many important concepts in the modern statistics, such as variance (Fisher, 1919), FDA (Fisher, 1936), Fisher information (Frieden, 2004), Analysis of Variance (ANOVA) (Fisher, 1992), etc. The paper (Fisher, 1936), which proposed FDA, was the first paper introducing the well-known Iris flower dataset. Note that Fisher's work was mostly concentrating on the statistics in the area of genetics. Much of his work was about variance making no wonder for us why FDA is all about variance and scatters.

Kernel FDA (Mika et al., 1999; 2000) performs the goal of FDA in the feature space. The FDA and kernel FDA have had many different applications. Some examples for applications of FDA are face recognition (Fisherfaces) (Belhumeur et al., 1997; Etemad & Chellappa, 1997; Zhao et al., 1999), action recognition (Fisherposes) (Ghojogh et al., 2017; Mokari et al., 2018), and gesture recognition (Samadani et al., 2013). Some examples for applications

of kernel FDA are face recognition (kernel Fisherfaces) (Yang, 2002; Liu et al., 2004) and palmprint Recognition (Wang & Ruan, 2006).

In the literature, sometimes, FDA is referred to as Linear Discriminant Analysis (LDA) or Fisher LDA (FLDA). This is because FDA and LDA (Ghojogh & Crowley, 2019a) are equivalent although LDA is a classification method and not a subspace learning algorithm. In this paper, we will prove why they are equivalent.

2. Projection Formulation

2.1. Projection

Assume we have a data point $\boldsymbol{x} \in \mathbb{R}^d$. We want to project this data point onto the vector space spanned by p vectors $\{\boldsymbol{u}_1, \dots, \boldsymbol{u}_p\}$ where each vector is d-dimensional and usually $p \ll d$. We stack these vectors column-wise in matrix $\boldsymbol{U} = [\boldsymbol{u}_1, \dots, \boldsymbol{u}_p] \in \mathbb{R}^{d \times p}$. In other words, we want to project $\boldsymbol{x}$ onto the column space of $\boldsymbol{U}$, denoted by $\mathbb{C}\text{ol}(\boldsymbol{U})$. The projection of $\boldsymbol{x} \in \mathbb{R}^d$ onto $\mathbb{C}\text{ol}(\boldsymbol{U}) \in \mathbb{R}^p$ and then its representation in the $\mathbb{R}^d$ (its reconstruction) can be seen as a linear system of equations:

$$\mathbb{R}^d \ni \widehat{\boldsymbol{x}} := \boldsymbol{U}\boldsymbol{\beta}, \tag{1}$$

where we should find the unknown coefficients $\boldsymbol{\beta} \in \mathbb{R}^p$.

If the $\boldsymbol{x}$ lies in the $\mathbb{C}\text{ol}(\boldsymbol{U})$ or $\textbf{span}\{\boldsymbol{u}_1, \dots, \boldsymbol{u}_p\}$, this linear system has exact solution, so $\widehat{\boldsymbol{x}} = \boldsymbol{x} = \boldsymbol{U}\boldsymbol{\beta}$. However, if $\boldsymbol{x}$ does not lie in this space, there is no any solution $\boldsymbol{\beta}$ for $\boldsymbol{x} = \boldsymbol{U}\boldsymbol{\beta}$ and we should solve for projection of $\boldsymbol{x}$ onto $\mathbb{C}\text{ol}(\boldsymbol{U})$ or $\textbf{span}\{\boldsymbol{u}_1, \dots, \boldsymbol{u}_p\}$ and then its reconstruction. In other words, we should solve for Eq. (1). In this case, $\widehat{\boldsymbol{x}}$ and $\boldsymbol{x}$ are different and we have a residual:

$$\boldsymbol{r} = \boldsymbol{x} - \widehat{\boldsymbol{x}} = \boldsymbol{x} - \boldsymbol{U}\boldsymbol{\beta}, \tag{2}$$

which we want to be small. As can be seen in Fig. 1, the smallest residual vector is orthogonal to $\mathbb{C}\text{ol}(\boldsymbol{U})$; therefore:

$$\begin{aligned} \boldsymbol{x} - \boldsymbol{U}\boldsymbol{\beta} \perp \boldsymbol{U} &\implies \boldsymbol{U}^\top(\boldsymbol{x} - \boldsymbol{U}\boldsymbol{\beta}) = 0, \\ &\implies \boldsymbol{\beta} = (\boldsymbol{U}^\top \boldsymbol{U})^{-1}\boldsymbol{U}^\top \boldsymbol{x}. \end{aligned} \tag{3}$$

It is noteworthy that the Eq. (3) is also the formula of coefficients in linear regression (Friedman et al., 2009) where the input data are the rows of $\boldsymbol{U}$ and the labels are $\boldsymbol{x}$; however, our goal here is different.

Plugging Eq. (3) in Eq. (1) gives us:

$$\widehat{\boldsymbol{x}} = \boldsymbol{U}(\boldsymbol{U}^\top \boldsymbol{U})^{-1}\boldsymbol{U}^\top \boldsymbol{x}.$$

We define:

$$\mathbb{R}^{d \times d} \ni \boldsymbol{\Pi} := \boldsymbol{U}(\boldsymbol{U}^\top \boldsymbol{U})^{-1}\boldsymbol{U}^\top, \tag{4}$$

as "projection matrix" because it projects $\boldsymbol{x}$ onto $\mathbb{C}\text{ol}(\boldsymbol{U})$ (and reconstructs back). Note that $\boldsymbol{\Pi}$ is also referred to as the "hat matrix" in the literature because it puts a hat on top of $\boldsymbol{x}$.

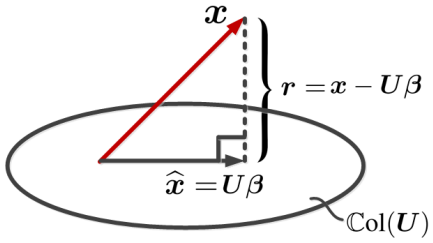

Figure 1. The residual and projection onto the column space of $\boldsymbol{U}$.

If the vectors $\{\boldsymbol{u}_1, \dots, \boldsymbol{u}_p\}$ are orthonormal (the matrix $\boldsymbol{U}$ is orthogonal), we have $\boldsymbol{U}^\top = \boldsymbol{U}^{-1}$ and thus $\boldsymbol{U}^\top \boldsymbol{U} = \boldsymbol{I}$. Therefore, Eq. (4) is simplified:

$$\boldsymbol{\Pi} = \boldsymbol{U}\boldsymbol{U}^\top. \tag{5}$$

So, we have:

$$\widehat{\boldsymbol{x}} = \boldsymbol{\Pi}\,\boldsymbol{x} = \boldsymbol{U}\boldsymbol{U}^\top \boldsymbol{x}. \tag{6}$$

2.2. Projection onto a Subspace

In subspace learning, the projection of a vector $\boldsymbol{x} \in \mathbb{R}^d$ onto the column space of $\boldsymbol{U} \in \mathbb{R}^{d \times p}$ (a p-dimensional subspace spanned by $\{\boldsymbol{u}_j\}_{j=1}^p$ where $\boldsymbol{u}_j \in \mathbb{R}^d$) is defined as:

$$\mathbb{R}^p \ni \widetilde{\boldsymbol{x}} := \boldsymbol{U}^\top \boldsymbol{x}, \tag{7}$$

$$\mathbb{R}^d \ni \widehat{\boldsymbol{x}} := \boldsymbol{U}\boldsymbol{U}^\top \boldsymbol{x} = \boldsymbol{U}\widetilde{\boldsymbol{x}}, \tag{8}$$

where $\widetilde{\boldsymbol{x}}$ and $\widehat{\boldsymbol{x}}$ denote the projection and reconstruction of $\boldsymbol{x}$, respectively.

If we have n data points, $\{\boldsymbol{x}_i\}_{i=1}^n$, which can be stored column-wise in a matrix $\boldsymbol{X} \in \mathbb{R}^{d \times n}$, the projection and reconstruction of $\boldsymbol{X}$ are defined as:

$$\mathbb{R}^{p \times n} \ni \widetilde{\boldsymbol{X}} := \boldsymbol{U}^\top \boldsymbol{X}, \tag{9}$$

$$\mathbb{R}^{d \times n} \ni \widehat{\boldsymbol{X}} := \boldsymbol{U}\boldsymbol{U}^\top \boldsymbol{X} = \boldsymbol{U}\widetilde{\boldsymbol{X}}, \tag{10}$$

respectively.

If we have an out-of-sample data point $\boldsymbol{x}_t$ which was not used in calculation of $\boldsymbol{U}$, the projection and reconstruction of it are defined as:

$$\mathbb{R}^p \ni \widetilde{\boldsymbol{x}}_t := \boldsymbol{U}^\top \boldsymbol{x}_t, \tag{11}$$

$$\mathbb{R}^d \ni \widehat{\boldsymbol{x}}_t := \boldsymbol{U}\boldsymbol{U}^\top \boldsymbol{x}_t = \boldsymbol{U}\widetilde{\boldsymbol{x}}_t, \tag{12}$$

respectively.

In case we have n_t out-of-sample data points, $\{\boldsymbol{x}_{t,i}\}_{i=1}^{n_t}$, which can be stored column-wise in a matrix $\boldsymbol{X}_t \in \mathbb{R}^{d \times n_t}$, the projection and reconstruction of $\boldsymbol{X}_t$ are defined as:

$$\mathbb{R}^{p \times n_t} \ni \widetilde{\boldsymbol{X}}_t := \boldsymbol{U}^\top \boldsymbol{X}_t, \tag{13}$$

$$\mathbb{R}^{d \times n_t} \ni \widehat{\boldsymbol{X}}_t := \boldsymbol{U}\boldsymbol{U}^\top \boldsymbol{X}_t = \boldsymbol{U}\widetilde{\boldsymbol{X}}_t, \tag{14}$$

respectively.

For the properties of the projection matrix $\boldsymbol{U}$, refer to (Ghojogh & Crowley, 2019c).

2.2.1. Projection onto a One-dimensional Subspace

Considering the data $\{\boldsymbol{x}_i\}_{i=1}^n$, the mean of data is:

$$\mathbb{R}^d \ni \boldsymbol{\mu}_x := \frac{1}{n}\sum_{i=1}^n \boldsymbol{x}_i, \tag{15}$$

and the centered data point $\boldsymbol{x}$ is:

$$\mathbb{R}^d \ni \breve{\boldsymbol{x}} := \boldsymbol{x} - \boldsymbol{\mu}_x. \tag{16}$$

The centered data $\boldsymbol{X}$ is:

$$\mathbb{R}^{d\times n} \ni \breve{\boldsymbol{X}} := \boldsymbol{X} - \boldsymbol{\mu}_x = \boldsymbol{X}\boldsymbol{H}, \tag{17}$$

where $\breve{\boldsymbol{X}} = [\breve{\boldsymbol{x}}_1, \dots, \breve{\boldsymbol{x}}_n] \in \mathbb{R}^{d\times n}$ and $\mathbb{R}^{n\times n} \ni \boldsymbol{H} := \boldsymbol{I} - (1/n)\boldsymbol{1}\boldsymbol{1}^\top$ is the centering matrix (see Appendix A in (Ghojogh & Crowley, 2019c)).

In Eq. (8), if $p = 1$, we are projecting $\boldsymbol{x}$ onto only one vector $\boldsymbol{u}$ and reconstruct it. If the data point is centered, the reconstruction is:

$$\widehat{\boldsymbol{x}} = \boldsymbol{u}\boldsymbol{u}^\top \breve{\boldsymbol{x}}.$$

The squared length (squared ℓ_2-norm) of this reconstructed vector is:

$$\begin{aligned}
&||\widehat{\boldsymbol{x}}||_2^2 = ||\boldsymbol{u}\boldsymbol{u}^\top \breve{\boldsymbol{x}}||_2^2 = (\boldsymbol{u}\boldsymbol{u}^\top \breve{\boldsymbol{x}})^\top (\boldsymbol{u}\boldsymbol{u}^\top \breve{\boldsymbol{x}}) \\
&= \breve{\boldsymbol{x}}^\top \boldsymbol{u} \underbrace{\boldsymbol{u}^\top \boldsymbol{u}}_{1} \boldsymbol{u}^\top \breve{\boldsymbol{x}} \overset{(a)}{=} \breve{\boldsymbol{x}}^\top \boldsymbol{u}\,\boldsymbol{u}^\top \breve{\boldsymbol{x}} \overset{(b)}{=} \boldsymbol{u}^\top \breve{\boldsymbol{x}}\,\breve{\boldsymbol{x}}^\top \boldsymbol{u},
\end{aligned} \tag{18}$$

where (a) is because $\boldsymbol{u}$ is a unit (normal) vector, i.e., $\boldsymbol{u}^\top\boldsymbol{u} = ||\boldsymbol{u}||_2^2 = 1$, and (b) is because $\breve{\boldsymbol{x}}^\top \boldsymbol{u} = \boldsymbol{u}^\top \breve{\boldsymbol{x}} \in \mathbb{R}$.

Suppose we have n data points $\{\boldsymbol{x}_i\}_{i=1}^n$ where $\{\breve{\boldsymbol{x}}_i\}_{i=1}^n$ are the centered data. The summation of the squared lengths of their projections $\{\widehat{\boldsymbol{x}}_i\}_{i=1}^n$ is:

$$\sum_{i=1}^n ||\widehat{\boldsymbol{x}}_i||_2^2 \overset{(18)}{=} \sum_{i=1}^n \boldsymbol{u}^\top \breve{\boldsymbol{x}}_i\, \breve{\boldsymbol{x}}_i^\top \boldsymbol{u} = \boldsymbol{u}^\top \Big(\sum_{i=1}^n \breve{\boldsymbol{x}}_i\, \breve{\boldsymbol{x}}_i^\top\Big)\boldsymbol{u}. \tag{19}$$

Considering $\breve{\boldsymbol{X}} = [\breve{\boldsymbol{x}}_1, \dots, \breve{\boldsymbol{x}}_n] \in \mathbb{R}^{d\times n}$, we have:

$$\begin{aligned}
\mathbb{R}^{d\times d} \ni \boldsymbol{S} &:= \sum_{i=1}^n (\boldsymbol{x}_i - \boldsymbol{\mu}_x)(\boldsymbol{x}_i - \boldsymbol{\mu}_x)^\top \overset{(16)}{=} \sum_{i=1}^n \breve{\boldsymbol{x}}_i\, \breve{\boldsymbol{x}}_i^\top \\
&= \breve{\boldsymbol{X}}\breve{\boldsymbol{X}}^\top \overset{(17)}{=} \boldsymbol{X}\boldsymbol{H}\boldsymbol{H}\boldsymbol{X}^\top,
\end{aligned} \tag{20}$$

where $\boldsymbol{S}$ is called the "covariance matrix" or "scatter matrix". If the data were already centered, we would have $\boldsymbol{S} = \boldsymbol{X}\boldsymbol{X}^\top$.

Plugging Eq. (20) in Eq. (19) gives us:

$$\sum_{i=1}^n ||\widehat{\boldsymbol{x}}_i||_2^2 = \boldsymbol{u}^\top \boldsymbol{S}\boldsymbol{u}. \tag{21}$$

Note that we can also say that $\boldsymbol{u}^\top\boldsymbol{S}\boldsymbol{u}$ is the variance of the projected data onto PCA subspace. In other words, $\boldsymbol{u}^\top\boldsymbol{S}\boldsymbol{u} = \mathbb{V}\text{ar}(\boldsymbol{u}^\top\breve{\boldsymbol{X}})$. This makes sense because when some non-random thing (here $\boldsymbol{u}$) is multiplied to the random data (here $\breve{\boldsymbol{X}}$), it will have squared (quadratic) effect on variance, and $\boldsymbol{u}^\top\boldsymbol{S}\boldsymbol{u}$ is quadratic in $\boldsymbol{u}$.

Therefore, $\boldsymbol{u}^\top\boldsymbol{S}\boldsymbol{u}$ can be interpreted in two ways: (I) the squared length of reconstruction and (II) the variance of projection.

If we consider the n data points in the matrix $\boldsymbol{X} \in \mathbb{R}^{d\times n}$, the squared length of reconstruction of the centered data is:

$$\begin{aligned}
||\widehat{\boldsymbol{X}}||_F^2 &= ||\boldsymbol{u}\boldsymbol{u}^\top\breve{\boldsymbol{X}}||_F^2 = \textbf{tr}\big((\boldsymbol{u}\boldsymbol{u}^\top\breve{\boldsymbol{X}})^\top(\boldsymbol{u}\boldsymbol{u}^\top\breve{\boldsymbol{X}})\big) \\
&= \textbf{tr}(\breve{\boldsymbol{X}}^\top \boldsymbol{u} \underbrace{\boldsymbol{u}^\top\boldsymbol{u}}_{1} \boldsymbol{u}^\top\breve{\boldsymbol{X}}) \overset{(a)}{=} \textbf{tr}(\breve{\boldsymbol{X}}^\top\boldsymbol{u}\boldsymbol{u}^\top\breve{\boldsymbol{X}}) \\
&\overset{(b)}{=} \textbf{tr}(\boldsymbol{u}^\top\breve{\boldsymbol{X}}\breve{\boldsymbol{X}}^\top\boldsymbol{u}) \overset{(c)}{=} \boldsymbol{u}^\top\breve{\boldsymbol{X}}\breve{\boldsymbol{X}}^\top\boldsymbol{u} \overset{(20)}{=} \boldsymbol{u}^\top\boldsymbol{S}\boldsymbol{u},
\end{aligned}$$

where $\textbf{tr}(.)$ denotes the trace of matrix, (a) is because $\boldsymbol{u}$ is a unit vector, (b) is because of the cyclic property of the trace, and (c) is because $\boldsymbol{u}^\top\breve{\boldsymbol{X}}\breve{\boldsymbol{X}}^\top\boldsymbol{u}$ is a scalar. Hence, we have:

$$||\widehat{\boldsymbol{X}}||_F^2 = \boldsymbol{u}^\top\boldsymbol{S}\boldsymbol{u}. \tag{22}$$

2.2.2. Projection Onto a Multi-dimensional Subspace

In Eq. (10), if $p > 1$, we are projecting the data onto a subspace with dimensionality more than one (spanned by $\{\boldsymbol{u}_j\}_{j=1}^p$) and then reconstruct back. If the data $\boldsymbol{X}$ are assumed to be centered, the reconstruction is:

$$\widehat{\boldsymbol{X}} = \boldsymbol{U}\boldsymbol{U}^\top\breve{\boldsymbol{X}}.$$

The squared length (squared Frobenius Norm) of this reconstructed matrix is:

$$\begin{aligned}
||\widehat{\boldsymbol{X}}||_F^2 &= ||\boldsymbol{U}\boldsymbol{U}^\top\breve{\boldsymbol{X}}||_F^2 = \textbf{tr}\big((\boldsymbol{U}\boldsymbol{U}^\top\breve{\boldsymbol{X}})^\top(\boldsymbol{U}\boldsymbol{U}^\top\breve{\boldsymbol{X}})\big) \\
&= \textbf{tr}(\breve{\boldsymbol{X}}^\top \boldsymbol{U} \underbrace{\boldsymbol{U}^\top\boldsymbol{U}}_{\boldsymbol{I}} \boldsymbol{U}^\top\breve{\boldsymbol{X}}) \overset{(a)}{=} \textbf{tr}(\breve{\boldsymbol{X}}^\top\boldsymbol{U}\boldsymbol{U}^\top\breve{\boldsymbol{X}}) \\
&\overset{(b)}{=} \textbf{tr}(\boldsymbol{U}^\top\breve{\boldsymbol{X}}\breve{\boldsymbol{X}}^\top\boldsymbol{U}) \overset{(20)}{=} \textbf{tr}(\boldsymbol{U}^\top\boldsymbol{S}\boldsymbol{U}),
\end{aligned}$$

where (a) is because $\boldsymbol{U}$ is an orthogonal matrix (its columns are orthonormal) and (b) is because of the cyclic property of trace. Thus, we have:

$$||\widehat{\boldsymbol{X}}||_F^2 = \textbf{tr}(\boldsymbol{U}^\top\boldsymbol{S}\boldsymbol{U}). \tag{23}$$

3. Fisher Discriminant Analysis

3.1. One-dimensional Subspace

3.1.1. Scatters in Two-Class Case

Assume we have two classes, $\{\boldsymbol{x}_i^{(1)}\}_{i=1}^{n_1}$ and $\{\boldsymbol{x}_i^{(2)}\}_{i=1}^{n_2}$, where n_1 and n_2 denote the sample size of the first and second class, respectively, and $\boldsymbol{x}_i^{(j)}$ denotes the i-th instance of the j-th class.

If the data instances of the j-th class are projected onto a one-dimensional subspace (vector $\boldsymbol{u}$) by $\boldsymbol{u}^\top \boldsymbol{x}_i^{(j)}$, the mean and the variance of the projected data are $\boldsymbol{u}^\top \boldsymbol{\mu}_j$ and $\boldsymbol{u}^\top \boldsymbol{S}_j \boldsymbol{u}$, respectively, where $\boldsymbol{\mu}_j$ and $\boldsymbol{S}_j$ are the mean and covariance matrix (scatter) of the j-th class. The mean of the j-th class is:

$$\mathbb{R}^d \ni \boldsymbol{\mu}_j := \frac{1}{n_j} \sum_{i=1}^{n_j} \boldsymbol{x}_i^{(j)}. \tag{24}$$

According to Appendix A, after projection onto the one-dimensional subspace, the distance between the means of classes is:

$$\begin{aligned}
\mathbb{R} \ni d_B &:= (\boldsymbol{u}^\top \boldsymbol{\mu}_1 - \boldsymbol{u}^\top \boldsymbol{\mu}_2)^\top (\boldsymbol{u}^\top \boldsymbol{\mu}_1 - \boldsymbol{u}^\top \boldsymbol{\mu}_2) \\
&= (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^\top \boldsymbol{u}\boldsymbol{u}^\top (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2) \\
&\overset{(a)}{=} \textbf{tr}\big((\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^\top \boldsymbol{u}\boldsymbol{u}^\top (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)\big) \\
&\overset{(b)}{=} \textbf{tr}\big(\boldsymbol{u}^\top (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^\top \boldsymbol{u}\big) \\
&\overset{(c)}{=} \boldsymbol{u}^\top (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^\top \boldsymbol{u} \overset{(d)}{=} \boldsymbol{u}^\top \boldsymbol{S}_B \, \boldsymbol{u},
\end{aligned} \tag{25}$$

where (a) is because $(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^\top \boldsymbol{u}\boldsymbol{u}^\top (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)$ is a scalar, (b) is because of the cyclic property of trace, (c) is because $\boldsymbol{u}^\top (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^\top \boldsymbol{u}$ is a scalar, and (d) is because we define:

$$\mathbb{R}^{d \times d} \ni \boldsymbol{S}_B := (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^\top, \tag{26}$$

as the *between-scatter* of classes.

The Eq. (25) can also be interpreted according to Eq. (22): the d_B is the variance of projection of the class means or the squared length of reconstruction of the class means.

We saw that the variance of projection is $\boldsymbol{u}^\top \boldsymbol{S}_j \boldsymbol{u}$ for the j-th class. If we add up the variances of projections of the two classes, we have:

$$\begin{aligned}
\mathbb{R} \ni d_W &:= \boldsymbol{u}^\top \boldsymbol{S}_1 \boldsymbol{u} + \boldsymbol{u}^\top \boldsymbol{S}_2 \boldsymbol{u} = \boldsymbol{u}^\top (\boldsymbol{S}_1 + \boldsymbol{S}_2)\, \boldsymbol{u} \\
&\overset{(a)}{=} \boldsymbol{u}^\top \boldsymbol{S}_W \, \boldsymbol{u},
\end{aligned} \tag{27}$$

where:

$$\mathbb{R}^{d \times d} \ni \boldsymbol{S}_W := \boldsymbol{S}_1 + \boldsymbol{S}_2, \tag{28}$$

is the *within-scatter* of classes. According to Eq. (22), the d_W is the summation of projection variance of class instances or the summation of the reconstruction length of class instances.

3.1.2. Scatters in Multi-Class Case: Variant 1

Assume $\{\boldsymbol{x}_i^{(j)}\}_{i=1}^{n_j}$ are the instances of the j-th class where we have multiple number of classes. In this case, the *between-scatter* is defined as:

$$\mathbb{R}^{d \times d} \ni \boldsymbol{S}_B := \sum_{j=1}^{c} (\boldsymbol{\mu}_j - \boldsymbol{\mu})(\boldsymbol{\mu}_j - \boldsymbol{\mu})^\top, \tag{29}$$

where c is the number of classes and:

$$\mathbb{R}^d \ni \boldsymbol{\mu} := \frac{1}{\sum_{k=1}^{c} n_k} \sum_{j=1}^{c} n_j \, \boldsymbol{\mu}_j = \frac{1}{n} \sum_{i=1}^{n} \boldsymbol{x}_i, \tag{30}$$

is the weighted mean of means of classes or the total mean of data.

It is noteworthy that some researches define the between-scatter in a weighted way:

$$\mathbb{R}^{d \times d} \ni \boldsymbol{S}_B := \sum_{j=1}^{c} n_j (\boldsymbol{\mu}_j - \boldsymbol{\mu})(\boldsymbol{\mu}_j - \boldsymbol{\mu})^\top. \tag{31}$$

If we extend the Eq. (28) to c number of classes, the *within-scatter* is defined as:

$$\mathbb{R}^{d \times d} \ni \boldsymbol{S}_W := \sum_{j=1}^{c} \boldsymbol{S}_j \tag{32}$$

$$\overset{(20)}{=} \sum_{j=1}^{c} \sum_{i=1}^{n_j} (\boldsymbol{x}_i^{(j)} - \boldsymbol{\mu}_j)(\boldsymbol{x}_i^{(j)} - \boldsymbol{\mu}_j)^\top, \tag{33}$$

where n_j is the sample size of the j-th class.

In this case, the d_B and d_W are:

$$\mathbb{R} \ni d_B := \boldsymbol{u}^\top \boldsymbol{S}_B \, \boldsymbol{u}, \tag{34}$$

$$\mathbb{R} \ni d_W := \boldsymbol{u}^\top \boldsymbol{S}_W \, \boldsymbol{u}, \tag{35}$$

where $\boldsymbol{S}_B$ and $\boldsymbol{S}_W$ are Eqs. (29) and (33).

3.1.3. Scatters in Multi-Class Case: Variant 2

There is another variant for multi-class case in FDA. In this variant, the within-scatter is the same as Eq. (33). The between-scatter is, however, different.

The *total-scatter* is defined as the covariance matrix of the whole data, regardless of classes (Welling, 2005):

$$\mathbb{R}^{d \times d} \ni \boldsymbol{S}_T := \frac{1}{n} \sum_{i=1}^{n} (\boldsymbol{x}_i - \boldsymbol{\mu})(\boldsymbol{x}_i - \boldsymbol{\mu})^\top, \tag{36}$$

where the total mean $\boldsymbol{\mu}$ is the Eq. (30). We can also use the scaled total-scatter by dropping the $1/n$ factor. On the other hand, the total scatter is equal to the summation of the within- and between-scatters:

$$\boldsymbol{S}_T = \boldsymbol{S}_W + \boldsymbol{S}_B. \tag{37}$$

Therefore, the between-scatter, in this variant, is obtained as:

$$\boldsymbol{S}_B := \boldsymbol{S}_T - \boldsymbol{S}_W. \tag{38}$$

3.1.4. FISHER SUBSPACE: VARIANT 1

In FDA, we want to maximize the projection variance (scatter) of means of classes and minimize the projection variance (scatter) of class instances. In other words, we want to maximize d_B and minimize d_W. The reason is that after projection, we want the within scatter of every class to be small and the between scatter of classes to be large; therefore, the instances of every class get close to one another and the classes get far from each other. The two mentioned optimization problems are:

$$\underset{\boldsymbol{u}}{\text{maximize}} \quad d_B(\boldsymbol{u}), \tag{39}$$

$$\underset{\boldsymbol{u}}{\text{minimize}} \quad d_W(\boldsymbol{u}). \tag{40}$$

We can merge these two optimization problems as a regularized optimization problem:

$$\underset{\boldsymbol{u}}{\text{maximize}} \quad d_B(\boldsymbol{u}) - \alpha\, d_W(\boldsymbol{u}), \tag{41}$$

where $\alpha > 0$ is the regularization parameter. Another way of merging Eqs. (39) and (40) is:

$$\underset{\boldsymbol{u}}{\text{maximize}} \quad f(\boldsymbol{u}) := \frac{d_B(\boldsymbol{u})}{d_W(\boldsymbol{u})} = \frac{\boldsymbol{u}^\top \boldsymbol{S}_B\, \boldsymbol{u}}{\boldsymbol{u}^\top \boldsymbol{S}_W\, \boldsymbol{u}}, \tag{42}$$

where $f(\boldsymbol{u}) \in \mathbb{R}$ is referred to as the *Fisher criterion* (Xu & Lu, 2006). The Fisher criterion is a generalized Rayleigh-Ritz Quotient (see Appendix B):

$$f(\boldsymbol{u}) \overset{(158)}{=} R(\boldsymbol{S}_B, \boldsymbol{S}_W; \boldsymbol{u}). \tag{43}$$

According to Eq. (165) in Appendix B, the optimization in Eq. (42) is equivalent to:

$$\begin{aligned} &\underset{\boldsymbol{u}}{\text{maximize}} && \boldsymbol{u}^\top \boldsymbol{S}_B\, \boldsymbol{u} \\ &\text{subject to} && \boldsymbol{u}^\top \boldsymbol{S}_W\, \boldsymbol{u} = 1. \end{aligned} \tag{44}$$

The Lagrangian (Boyd & Vandenberghe, 2004) is:

$$\mathcal{L} = \boldsymbol{w}^\top \boldsymbol{S}_B\, \boldsymbol{w} - \lambda(\boldsymbol{w}^\top \boldsymbol{S}_W\, \boldsymbol{w} - 1),$$

where λ is the Lagrange multiplier. Equating the derivative of $\mathcal{L}$ to zero gives:

$$\begin{aligned} &\mathbb{R}^d \ni \frac{\partial \mathcal{L}}{\partial \boldsymbol{u}} = 2\, \boldsymbol{S}_B\, \boldsymbol{u} - 2\, \lambda\, \boldsymbol{S}_W\, \boldsymbol{u} \overset{\text{set}}{=} \boldsymbol{0} \\ &\implies 2\, \boldsymbol{S}_B\, \boldsymbol{u} = 2\, \lambda\, \boldsymbol{S}_W\, \boldsymbol{u} \implies \boldsymbol{S}_B\, \boldsymbol{u} = \lambda\, \boldsymbol{S}_W\, \boldsymbol{u}, \end{aligned} \tag{45}$$

which is a generalized eigenvalue problem $(\boldsymbol{S}_B, \boldsymbol{S}_W)$ according to (Ghojogh et al., 2019a). The $\boldsymbol{u}$ is the eigenvector with the largest eigenvalue (because the optimization is maximization) and the λ is the corresponding eigenvalue. The $\boldsymbol{u}$ is referred to as the *Fisher direction* or *Fisher axis*. The projection and reconstruction are according to Eqs. (9) and (10), respectively, where $\boldsymbol{u} \in \mathbb{R}^d$ is used instead of $\boldsymbol{U} \in \mathbb{R}^{d \times p}$. The out-of-sample projection and reconstruction are according to Eqs. (13) and (14), respectively, with $\boldsymbol{u}$ rather than $\boldsymbol{U}$.

One possible solution to the generalized eigenvalue problem $(\boldsymbol{S}_B, \boldsymbol{S}_W)$ is (Ghojogh et al., 2019a):

$$\begin{aligned} &\boldsymbol{S}_B\, \boldsymbol{u} = \lambda\, \boldsymbol{S}_W\, \boldsymbol{u} \implies \boldsymbol{S}_W^{-1} \boldsymbol{S}_B\, \boldsymbol{u} = \lambda\, \boldsymbol{u} \\ &\implies \boldsymbol{u} = \textbf{eig}(\boldsymbol{S}_W^{-1} \boldsymbol{S}_B), \end{aligned} \tag{46}$$

where **eig**(.) denotes the eigenvector of the matrix with the largest eigenvalue. Although the solution in Eq. (46) is a little dirty (Ghojogh et al., 2019a) because $\boldsymbol{S}_w$ might be singular and not invertible, but this solution is very common for FDA. In some researches, the diagonal of $\boldsymbol{S}_W$ is strengthened slightly to make it full rank and invertible (Ghojogh et al., 2019a):

$$\boldsymbol{u} = \textbf{eig}((\boldsymbol{S}_W + \varepsilon \boldsymbol{I})^{-1} \boldsymbol{S}_B), \tag{47}$$

where ε is a very small positive number, large enough to make $\boldsymbol{S}_W$ full rank.

In a future section, we will cover robust FDA which tackles this problem. On the other hand, the generalized eigenvalue problem $(\boldsymbol{S}_B, \boldsymbol{S}_W)$ has a rigorous solution (Ghojogh et al., 2019a; Wang, 2015) which does not require non-singularity of $\boldsymbol{S}_W$.

Another way to solve the optimization in Eq. (42) is taking derivative from the Fisher criterion:

$$\begin{aligned} &\mathbb{R}^d \ni \frac{\partial f(\boldsymbol{u})}{\partial \boldsymbol{u}} = \frac{1}{(\boldsymbol{u}^\top \boldsymbol{S}_W\, \boldsymbol{u})^2} \times \\ &\quad \Big[(\boldsymbol{u}^\top \boldsymbol{S}_W\, \boldsymbol{u})(2 \boldsymbol{S}_B \boldsymbol{u}) - (\boldsymbol{u}^\top \boldsymbol{S}_B\, \boldsymbol{u})(2 \boldsymbol{S}_W \boldsymbol{u})\Big] \overset{\text{set}}{=} \boldsymbol{0} \\ &\overset{(a)}{\implies} \boldsymbol{S}_B\, \boldsymbol{u} = \frac{\boldsymbol{u}^\top \boldsymbol{S}_B\, \boldsymbol{u}}{\boldsymbol{u}^\top \boldsymbol{S}_W\, \boldsymbol{u}}\, \boldsymbol{S}_W\, \boldsymbol{u}, \end{aligned} \tag{48}$$

where (a) is because $\boldsymbol{u}^\top \boldsymbol{S}_W\, \boldsymbol{u}$ is a scalar. The Eq. (48) which is a generalized eigenvalue problem $(\boldsymbol{S}_B, \boldsymbol{S}_W)$ (Ghojogh et al., 2019a) with $\boldsymbol{u}$ and $(\boldsymbol{u}^\top \boldsymbol{S}_B\, \boldsymbol{u})/(\boldsymbol{u}^\top \boldsymbol{S}_W\, \boldsymbol{u})$ as the eigenvector with the largest eigenvalue (because the optimization is maximization) and the corresponding eigenvalue, respectively. Therefore, *the Fisher criterion is the eigenvalue of the Fisher direction.*

3.1.5. FISHER SUBSPACE: VARIANT 2

Another way to find the FDA direction is to consider another version of Fisher criterion. According to Eq. (38) for $\boldsymbol{S}_B$, the Fisher criterion becomes (Welling, 2005):

$$\begin{aligned} f(\boldsymbol{u}) &= \frac{\boldsymbol{u}^\top \boldsymbol{S}_B\, \boldsymbol{u}}{\boldsymbol{u}^\top \boldsymbol{S}_W\, \boldsymbol{u}} \overset{(38)}{=} \frac{\boldsymbol{u}^\top (\boldsymbol{S}_T - \boldsymbol{S}_W)\, \boldsymbol{u}}{\boldsymbol{u}^\top \boldsymbol{S}_W\, \boldsymbol{u}} \\ &= \frac{\boldsymbol{u}^\top \boldsymbol{S}_T\, \boldsymbol{u} - \boldsymbol{u}^\top \boldsymbol{S}_W\, \boldsymbol{u}}{\boldsymbol{u}^\top \boldsymbol{S}_W\, \boldsymbol{u}} = \frac{\boldsymbol{u}^\top \boldsymbol{S}_T\, \boldsymbol{u}}{\boldsymbol{u}^\top \boldsymbol{S}_W\, \boldsymbol{u}} - 1. \end{aligned} \tag{49}$$

The -1 is a constant and is dropped in the optimization; therefore:

$$\begin{aligned} \underset{\boldsymbol{u}}{\text{maximize}} \quad & \boldsymbol{u}^\top \boldsymbol{S}_T \, \boldsymbol{u} \\ \text{subject to} \quad & \boldsymbol{u}^\top \boldsymbol{S}_W \, \boldsymbol{u} = 1, \end{aligned} \tag{50}$$

whose solution is similarly obtained as:

$$\boldsymbol{S}_T \, \boldsymbol{u} = \lambda \, \boldsymbol{S}_W \, \boldsymbol{u}, \tag{51}$$

which is a generalized eigenvalue problem $(\boldsymbol{S}_T, \boldsymbol{S}_W)$ according to (Ghojogh et al., 2019a).

3.2. Multi-dimensional Subspace

In case the Fisher subspace is the span of several Fisher directions, $\{\boldsymbol{u}_j\}_{j=1}^p$ where $\boldsymbol{u}_j \in \mathbb{R}^d$, the d_B and d_W are defined as:

$$\mathbb{R} \ni d_B := \textbf{tr}(\boldsymbol{U}^\top \boldsymbol{S}_B \, \boldsymbol{U}), \tag{52}$$

$$\mathbb{R} \ni d_W := \textbf{tr}(\boldsymbol{U}^\top \boldsymbol{S}_W \, \boldsymbol{U}), \tag{53}$$

where $\mathbb{R}^{d \times p} \ni \boldsymbol{U} = [\boldsymbol{u}_1, \dots, \boldsymbol{u}_p]$. In this case, maximizing the *Fisher criterion* is:

$$\underset{\boldsymbol{U}}{\text{maximize}} \quad f(\boldsymbol{U}) := \frac{d_B(\boldsymbol{U})}{d_W(\boldsymbol{U})} = \frac{\textbf{tr}(\boldsymbol{U}^\top \boldsymbol{S}_B \, \boldsymbol{U})}{\textbf{tr}(\boldsymbol{U}^\top \boldsymbol{S}_W \, \boldsymbol{U})}. \tag{54}$$

The Fisher criterion $f(\boldsymbol{U})$ is a generalized Rayleigh-Ritz Quotient (see Appendix B). According to Eq. (165) in Appendix B, the optimization in Eq. (54) is equivalent to:

$$\begin{aligned} \underset{\boldsymbol{U}}{\text{maximize}} \quad & \textbf{tr}(\boldsymbol{U}^\top \boldsymbol{S}_B \, \boldsymbol{U}) \\ \text{subject to} \quad & \boldsymbol{U}^\top \boldsymbol{S}_W \, \boldsymbol{U} = \boldsymbol{I}. \end{aligned} \tag{55}$$

The Lagrangian (Boyd & Vandenberghe, 2004) is:

$$\mathcal{L} = \textbf{tr}(\boldsymbol{U}^\top \boldsymbol{S}_B \, \boldsymbol{U}) - \textbf{tr}\big(\boldsymbol{\Lambda}^\top (\boldsymbol{U}^\top \boldsymbol{S}_W \, \boldsymbol{U} - \boldsymbol{I})\big),$$

where $\boldsymbol{\Lambda} \in \mathbb{R}^{d \times d}$ is a diagonal matrix whose diagonal entries are the Lagrange multipliers. Equating the derivative of $\mathcal{L}$ to zero gives:

$$\begin{aligned} &\mathbb{R}^{d \times p} \ni \frac{\partial \mathcal{L}}{\partial \boldsymbol{U}} = 2 \, \boldsymbol{S}_B \, \boldsymbol{U} - 2 \, \boldsymbol{S}_W \, \boldsymbol{U} \boldsymbol{\Lambda} \overset{\text{set}}{=} \boldsymbol{0} \\ &\implies 2 \, \boldsymbol{S}_B \, \boldsymbol{U} = 2 \, \boldsymbol{S}_W \, \boldsymbol{U} \boldsymbol{\Lambda} \implies \boldsymbol{S}_B \, \boldsymbol{U} = \boldsymbol{S}_W \, \boldsymbol{U} \boldsymbol{\Lambda}, \end{aligned} \tag{56}$$

which is a generalized eigenvalue problem $(\boldsymbol{S}_B, \boldsymbol{S}_W)$ according to (Ghojogh et al., 2019a). The columns of $\boldsymbol{U}$ are the eigenvectors sorted by largest to smallest eigenvalues (because the optimization is maximization) and the diagonal entries of $\boldsymbol{\Lambda}$ are the corresponding eigenvalues. The columns of $\boldsymbol{U}$ are referred to as the *Fisher directions* or *Fisher axes*. The projection and reconstruction are according to Eqs. (9) and (10), respectively. The out-of-sample projection and reconstruction are according to Eqs. (13) and (14), respectively.

One possible solution to the generalized eigenvalue problem $(\boldsymbol{S}_B, \boldsymbol{S}_W)$ is (Ghojogh et al., 2019a):

$$\begin{aligned} &\boldsymbol{S}_B \, \boldsymbol{U} = \boldsymbol{S}_W \, \boldsymbol{U} \boldsymbol{\Lambda} \implies \boldsymbol{S}_W^{-1} \boldsymbol{S}_B \, \boldsymbol{U} = \boldsymbol{U} \boldsymbol{\Lambda} \\ &\implies \boldsymbol{U} = \textbf{eig}(\boldsymbol{S}_W^{-1} \boldsymbol{S}_B), \end{aligned} \tag{57}$$

where $\textbf{eig}(.)$ denotes the eigenvectors of the matrix stacked column-wise. Again, we can have (Ghojogh et al., 2019a):

$$\boldsymbol{U} = \textbf{eig}((\boldsymbol{S}_W + \varepsilon \boldsymbol{I})^{-1} \boldsymbol{S}_B), \tag{58}$$

Another way to solve the optimization in Eq. (54) is taking derivative from the Fisher criterion:

$$\begin{aligned} &\mathbb{R}^{d \times p} \ni \frac{\partial f(\boldsymbol{U})}{\partial \boldsymbol{U}} = \frac{1}{\big(\textbf{tr}(\boldsymbol{U}^\top \boldsymbol{S}_W \, \boldsymbol{U})\big)^2} \times \\ &\Big[\textbf{tr}(\boldsymbol{U}^\top \boldsymbol{S}_W \, \boldsymbol{U})(2 \boldsymbol{S}_B \boldsymbol{U}) - \textbf{tr}(\boldsymbol{U}^\top \boldsymbol{S}_B \, \boldsymbol{U})(2 \boldsymbol{S}_W \boldsymbol{U})\Big] \overset{\text{set}}{=} \boldsymbol{0} \\ &\overset{(a)}{\implies} \boldsymbol{S}_B \, \boldsymbol{U} = \frac{\textbf{tr}(\boldsymbol{U}^\top \boldsymbol{S}_B \, \boldsymbol{U})}{\textbf{tr}(\boldsymbol{U}^\top \boldsymbol{S}_W \, \boldsymbol{U})} \boldsymbol{S}_W \, \boldsymbol{U}, \end{aligned} \tag{59}$$

where (a) is because $\textbf{tr}(\boldsymbol{U}^\top \boldsymbol{S}_W \, \boldsymbol{U})$ is a scalar. The Eq. (59) which is a generalized eigenvalue problem $(\boldsymbol{S}_B, \boldsymbol{S}_W)$ (Ghojogh et al., 2019a) with columns of $\boldsymbol{U}$ as the eigenvectors and $(\boldsymbol{u}_j^\top \boldsymbol{S}_B \, \boldsymbol{u}_j)/(\boldsymbol{u}_j^\top \boldsymbol{S}_W \, \boldsymbol{u}_j)$ as the j-th largest eigenvalue (because the optimization is maximization).

Again, another way to find the FDA directions is to consider another version of Fisher criterion. According to Eq. (38) for $\boldsymbol{S}_B$, the Fisher criterion becomes (Welling, 2005):

$$f(\boldsymbol{U}) = \frac{\textbf{tr}\big(\boldsymbol{U}^\top (\boldsymbol{S}_T - \boldsymbol{S}_W) \, \boldsymbol{U}\big)}{\textbf{tr}(\boldsymbol{U}^\top \boldsymbol{S}_W \, \boldsymbol{U})} = \frac{\textbf{tr}(\boldsymbol{U}^\top \boldsymbol{S}_T \, \boldsymbol{U})}{\textbf{tr}(\boldsymbol{U}^\top \boldsymbol{S}_W \, \boldsymbol{U})} - 1. \tag{60}$$

The -1 is a constant and is dropped in the optimization; therefore:

$$\begin{aligned} \underset{\boldsymbol{U}}{\text{maximize}} \quad & \textbf{tr}(\boldsymbol{U}^\top \boldsymbol{S}_T \, \boldsymbol{U}) \\ \text{subject to} \quad & \boldsymbol{U}^\top \boldsymbol{S}_W \, \boldsymbol{U} = \boldsymbol{I}, \end{aligned} \tag{61}$$

whose solution is similarly obtained as:

$$\boldsymbol{S}_T \, \boldsymbol{U} = \boldsymbol{S}_W \, \boldsymbol{U} \boldsymbol{\Lambda}, \tag{62}$$

which is a generalized eigenvalue problem $(\boldsymbol{S}_T, \boldsymbol{S}_W)$ according to (Ghojogh et al., 2019a).

3.3. Discussion on Dimensionality of the Fisher Subspace

In general, the rank of a covariance (scatter) matrix over the d-dimensional data with sample size n is at most $\min(d, n-1)$. The d is because the covariance matrix is a $d \times d$ matrix and the n is because we iterate over n data instances for calculating the covariance matrix. The -1 is because of subtracting the mean in calculation of the

covariance matrix. For clarification, assume we only have one instance which becomes zero after removing the mean. This makes the covariance matrix a zero matrix.

According to Eq. (33), the rank of the $\boldsymbol{S}_W$ is at most $\min(d, n-1)$ because all the instances of all the classes are considered. Hence, the rank of $\boldsymbol{S}_W$ is also at most $\min(d, n-1)$. According to Eq. (29), the rank of the $\boldsymbol{S}_B$ is at most $\min(d, c-1)$ because we have c iterations in its calculation.

In Eq. (57), we have $\boldsymbol{S}_W^{-1}\boldsymbol{S}_B$ whose rank is:

$$\begin{aligned}&\textbf{rank}(\boldsymbol{S}_W^{-1}\boldsymbol{S}_B) \leq \min\big(\textbf{rank}(\boldsymbol{S}_W^{-1}), \textbf{rank}(\boldsymbol{S}_B)\big)\\ &\leq \min\big(\min(d, n-1), \min(d, c-1)\big)\\ &= \min(d, n-1, c-1) \overset{(a)}{=} c-1,\end{aligned} \tag{63}$$

where (a) is because we usually have $c < d, n$. Therefore, the rank of $\boldsymbol{S}_W^{-1}\boldsymbol{S}_B$ is limited because of the rank of $\boldsymbol{S}_B$ which is at most $c-1$.

According to Eq. (57), the $c-1$ leading eigenvalues will be valid and the rest are zero or very small. Therefore, the p, which is the dimensionality of the Fisher subspace, is at most $c-1$. The $c-1$ leading eigenvectors are considered as the Fisher directions and the rest of eigenvectors are invalid and ignored.

4. Interpretation of FDA: The Example of a Man with Weak Eyes

In this section, we interpret the FDA using a real-life example in order to better understand the essence of Fisher's method. Consider a man which has two eye problems: (1) he is color-blind and (2) his eyes are also very weak.

Suppose there are two sets of balls with red and blue colors. The man wants to discriminate the balls into red and blue classes; however, he needs help because of his eye problems.

First, consider his color-blindness. In order to help him, we separate the balls into two sets of red and blue. In other words, we increase the distances of the balls with different colors to give him a clue that which balls belong to the same class. This means that we are increasing the between-scatter of the two classes to help him.

Second, consider his very weak eyes. although the balls with different colors are almost separated, everything is blue to him. Thus, we put the balls of the same color closer to one another. In other words, we decrease the within-scatter of every class. In this way, the man sees every class as almost one blurry ball so he can discriminate the classes better.

Recall Eq. (57) which includes $\boldsymbol{S}_W^{-1}\boldsymbol{S}_B$. The $\boldsymbol{S}_B$ implies that we want to increase the between-scatter as we did in the first help. The $\boldsymbol{S}_W^{-1}$ implies that we want to decrease the within-scatter as done in the second help to the man.

In conclusion, FDA increases the between-scatter and decreases the within-scatter (collapses each class (Globerson & Roweis, 2006)), at the same time, for better discrimination of the classes.

5. Robust Fisher Discriminant Analysis

Robust FDA (RFDA) (Deng et al., 2007; Guo & Wang, 2015), has also addressed the problem of singularity (or close to singularity) of $\boldsymbol{S}_W$. In RFDA, the $\boldsymbol{S}_W$ is decomposed using eigenvalue decomposition (Ghojogh et al., 2019a):

$$\boldsymbol{S}_W = \boldsymbol{\Phi}^\top \boldsymbol{\Lambda} \boldsymbol{\Phi}, \tag{64}$$

where $\boldsymbol{\Phi}$ and $\boldsymbol{\Lambda} = \textbf{diag}([\lambda_1, \dots, \lambda_d]^\top)$ include the eigenvectors and eigenvalues of $\boldsymbol{S}_W$, respectively. The eigenvalues are sorted as $\lambda_1 \geq \dots \geq \lambda_d$ and the eigenvectors (columns of $\boldsymbol{\Phi}$) are sorted accordingly. If $\boldsymbol{S}_W$ is close to singularity, the first d' eigenvalues are valid and the rest $(d-d')$ eigenvalues are either very small or zero. The appropriate d' is obtained as:

$$d' := \arg\min_m \left(\frac{\sum_{j=1}^m \lambda_j}{\sum_{k=1}^d \lambda_k} \geq 0.98 \right). \tag{65}$$

In RFDA, the $(d-d')$ invalid eigenvalues are replaced with λ_*:

$$\mathbb{R}^{d\times d} \ni \boldsymbol{\Lambda}' := \textbf{diag}([\lambda_1, \dots, \lambda_{d'}, \lambda_*, \dots, \lambda_*]^\top), \tag{66}$$

where (Deng et al., 2007):

$$\lambda_* := \frac{1}{d-d'} \sum_{j=d'+1}^{d} \lambda_j. \tag{67}$$

Hence, the $\boldsymbol{S}_W$ is replaced with $\boldsymbol{S}'_W$:

$$\mathbb{R}^{d\times d} \ni \boldsymbol{S}'_W := \boldsymbol{\Phi}^\top \boldsymbol{\Lambda}' \boldsymbol{\Phi}, \tag{68}$$

and the robust Fisher directions are the eigenvectors of the generalized eigenvalue problem $(\boldsymbol{S}_B, \boldsymbol{S}'_W)$ (Ghojogh et al., 2019a).

6. Comparison of FDA and PCA Directions

The FDA directions capture the directions where the instances of different classes fall apart and the instances in one class fall close to each other. On the other hand, the PCA directions capture the directions where the data have maximum variance (spread) regardless of the classes (Ghojogh & Crowley, 2019c). In some datasets, the FDA and PCA are orthogonal and in some datasets, they are parallel. Other cases between these two extreme cases can happen for some datasets. This depends on the spread of classes in the dataset. Figure 2 shows these cases for some two-dimensional datasets.

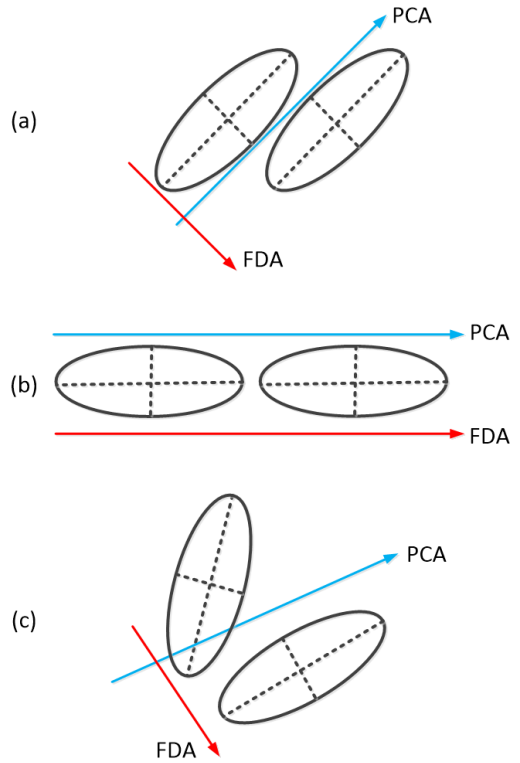

Figure 2. Comparison of FDA and PCA directions for two-dimensional data with two classes: (a) a case where FDA and PCA are orthogonal, (b) a case where FDA and PCA are equivalent (parallel), and (c) a case between the two extreme cases of (a) and (b).

Moreover, considering the Eq. (38) for $\boldsymbol{S}_B$, the Fisher criterion becomes Eqs. (49) and (60) for one-dimensional and multi-dimensional Fisher subspaces, respectively. In these equations, the -1 is a constant and is dropped in the optimization. This has an important message about FDA: *the Fisher direction is maximizing the total variance (spread) of data, as also done in PCA, while at the same time, it minimizes the within-scatters of classes (by making use of the class labels).* In other words, the optimization of FDA is equivalent to (we repeat Eq. (61) here):

$$\begin{aligned} \underset{\boldsymbol{U}}{\text{maximize}} \quad & \textbf{tr}(\boldsymbol{U}^\top \boldsymbol{S}_T \boldsymbol{U}) \\ \text{subject to} \quad & \boldsymbol{U}^\top \boldsymbol{S}_W \boldsymbol{U} = \boldsymbol{I}, \end{aligned} \tag{69}$$

while the optimization of the PCA is (Ghojogh & Crowley, 2019c):

$$\begin{aligned} \underset{\boldsymbol{U}}{\text{maximize}} \quad & \textbf{tr}(\boldsymbol{U}^\top \boldsymbol{S}_T \boldsymbol{U}) \\ \text{subject to} \quad & \boldsymbol{U}^\top \boldsymbol{U} = \boldsymbol{I}. \end{aligned} \tag{70}$$

The solutions to Eqs. (69) and (70) are the generalized eigenvalue problem $(\boldsymbol{S}_T, \boldsymbol{S}_W)$ and the eigenvalue problem for $\boldsymbol{S}_T$, respectively (Ghojogh et al., 2019a).

7. FDA $\overset{?}{=}$ LDA

The FDA is also referred to as Linear Discriminant Analysis (LDA) and Fisher LDA (FLDA). Note that FDA is a manifold (subspace) learning method and LDA (Ghojogh & Crowley, 2019a) is a classification method. However, LDA can be seen as a metric learning method (Ghojogh & Crowley, 2019a) and as metric learning is a manifold learning method (see Appendix A), there is a connection between FDA and LDA.

We know that FDA is a projection-based subspace learning method. Consider the projection vector $\boldsymbol{u}$. According to Eq. (7), the projection of data $\boldsymbol{x}$ is:

$$\boldsymbol{x} \mapsto \boldsymbol{u}^\top \boldsymbol{x}, \tag{71}$$

which can be done for all the data instances of every class. Thus, the mean and the covariance matrix of the class are transformed as:

$$\boldsymbol{\mu} \mapsto \boldsymbol{u}^\top \boldsymbol{\mu}, \tag{72}$$

$$\boldsymbol{\Sigma} \mapsto \boldsymbol{u}^\top \boldsymbol{\Sigma}\, \boldsymbol{u}, \tag{73}$$

because of characteristics of mean and variance.

According to Eq. (42), the Fisher criterion is the ratio of the between-class variance, σ_b^2, and within-class variance, σ_w^2:

$$f := \frac{\sigma_b^2}{\sigma_w^2} = \frac{(\boldsymbol{u}^\top \boldsymbol{\mu}_2 - \boldsymbol{u}^\top \boldsymbol{\mu}_1)^2}{\boldsymbol{u}^\top \boldsymbol{\Sigma}_2\, \boldsymbol{u} + \boldsymbol{u}^\top \boldsymbol{\Sigma}_1\, \boldsymbol{u}} = \frac{\big(\boldsymbol{u}^\top (\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1)\big)^2}{\boldsymbol{u}^\top (\boldsymbol{\Sigma}_2 + \boldsymbol{\Sigma}_1)\, \boldsymbol{u}}. \tag{74}$$

The FDA maximizes the Fisher criterion:

$$\underset{\boldsymbol{u}}{\text{maximize}} \quad \frac{\big(\boldsymbol{u}^\top (\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1)\big)^2}{\boldsymbol{u}^\top (\boldsymbol{\Sigma}_2 + \boldsymbol{\Sigma}_1)\, \boldsymbol{u}}, \tag{75}$$

which can be restated as:

$$\begin{aligned} \underset{\boldsymbol{u}}{\text{maximize}} \quad & \big(\boldsymbol{u}^\top (\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1)\big)^2, \\ \text{subject to} \quad & \boldsymbol{u}^\top (\boldsymbol{\Sigma}_2 + \boldsymbol{\Sigma}_1)\, \boldsymbol{u} = 1, \end{aligned} \tag{76}$$

according to Rayleigh-Ritz quotient method (Croot, 2005). The Lagrangian (Boyd & Vandenberghe, 2004) is:

$$\mathcal{L} = \big(\boldsymbol{u}^\top (\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1)\big)^2 - \lambda \big(\boldsymbol{u}^\top (\boldsymbol{\Sigma}_2 + \boldsymbol{\Sigma}_1)\, \boldsymbol{u} - 1\big),$$

where λ is the Lagrange multiplier. Equating the derivative of $\mathcal{L}$ to zero gives:

$$\begin{aligned} & \frac{\partial \mathcal{L}}{\partial \boldsymbol{u}} = 2\, (\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1)^2\, \boldsymbol{u} - 2\, \lambda\, (\boldsymbol{\Sigma}_2 + \boldsymbol{\Sigma}_1)\, \boldsymbol{u} \overset{\text{set}}{=} \boldsymbol{0} \\ & \implies (\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1)^2\, \boldsymbol{u} = \lambda\, (\boldsymbol{\Sigma}_2 + \boldsymbol{\Sigma}_1)\, \boldsymbol{u}, \end{aligned}$$

which is a generalized eigenvalue problem $\big((\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1)^2, (\boldsymbol{\Sigma}_2 + \boldsymbol{\Sigma}_1)\big)$ according to (Ghojogh et al., 2019a).

The projection vector is the eigenvector of $(\boldsymbol{\Sigma}_2 + \boldsymbol{\Sigma}_1)^{-1}(\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1)^2$; therefore, we can say:

$$\boldsymbol{u} \propto (\boldsymbol{\Sigma}_2 + \boldsymbol{\Sigma}_1)^{-1}(\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1)^2. \tag{77}$$

On the other hand, in LDA, the decision function is (Ghojogh & Crowley, 2019a):

$$\begin{aligned} &2\left(\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1)\right)^\top \boldsymbol{x} \\ &+ (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2) = 0, \end{aligned} \tag{78}$$

Moreover, in LDA, the covariance matrices are assumed to be equal (Ghojogh & Crowley, 2019a): $\boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}_2 = \boldsymbol{\Sigma}$. Therefore, in LDA, the Eq. (77) becomes (Ghojogh & Crowley, 2019a):

$$\boldsymbol{u} \propto (2\,\boldsymbol{\Sigma})^{-1}(\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1)^2 \propto \boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1)^2. \tag{79}$$

According to Eq. (71), we have:

$$\boldsymbol{u}^\top \boldsymbol{x} \propto \left(\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1)^2\right)^\top \boldsymbol{x}. \tag{80}$$

Comparing Eq. (80) with Eq. (78) shows that LDA and FDA are equivalent up to a scaling factor $(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)$. Note that this term is multiplied as an exponential factor before taking logarithm to obtain Eq. (78), so this term a scaling factor (see (Ghojogh & Crowley, 2019a) for more details). It should be noted that in manifold (subspace) learning, the scale does not matter because all the distances scale similarly. Hence, we can say that LDA and FDA are equivalent:

$$\text{LDA} \equiv \text{FDA}. \tag{81}$$

Therefore, *the two subspaces of FDA and LDA are the same subspace.* This sheds light to why LDA and FDA are used interchangeably in the literature.

Note that LDA assumes *one* (and not several) Gaussian for every class (Ghojogh & Crowley, 2019a) and so does the FDA because they are equivalent. That is why FDA faces problem for multi-modal data (Sugiyama, 2007).

8. Fisher Forest

If the data include several different types of data which may even have different dimensionality. Some examples of these types of data are different key-poses in action human action recognition or different facial expressions that a face can have. In this case, we can use the *Fisher forest* (Ghojogh & Mohammadzade, 2017). Note that forest here does not imply an ensemble of trees but means an ensemble of the Fisher subspaces.

Let the number of the data types be z and let the dimensionality of the ℓ-th data type be $d_{|\ell}$. We usually have a dataset $\{\boldsymbol{x}_i\}_{i=1}^n$ where $\boldsymbol{x}_i \in \mathbb{R}^d$. Every type of data is the whole dataset but having only a subset of the features, i.e., $\{\boldsymbol{x}_{i|\ell}\}_{i=1}^n$ where $\boldsymbol{x}_{i|\ell} \in \mathbb{R}^{d_{|\ell}}$. The features of the ℓ-th type are a subset of the features of the dataset, i.e., $d_{|\ell} \leq d$. Note that, we do not necessarily have the same value for $d_{|\ell}$ in all the data types. The i-th instance of the j-th class having the ℓ-th type is denoted by $\boldsymbol{x}_{i|\ell}^{(j)}$.

For example, in the key-poses of human action, the important of skeletal joints can be different in various key-poses (Ghojogh & Mohammadzade, 2017). Thus, some joints are taken in a specific key-joint and some other are taken in another key-pose. Note that a key-pose can have five key-joints but another key-pose can have three key-joints. Another example is using different landmarks for different facial expressions; for example, eye-brows, lips, and chin for wondering but just lips for smiling. As can be seen, Fisher forest can be useful for handling the data types with different features and even dimensionality.

The between- and within-scatters for the ℓ-th data types (for all $\ell \in \{1, \dots, z\}$) are defined as (Ghojogh & Mohammadzade, 2017):

$$\mathbb{R}^{d_\ell \times d_\ell} \ni \boldsymbol{S}_{B|\ell} := \sum_{j=1}^{c} n_j (\boldsymbol{\mu}_{j|\ell} - \boldsymbol{\mu}_{|\ell})(\boldsymbol{\mu}_{j|\ell} - \boldsymbol{\mu}_{|\ell})^\top, \tag{82}$$

$$\mathbb{R}^{d_\ell \times d_\ell} \ni \boldsymbol{S}_{W|\ell} := \sum_{j=1}^{c} \sum_{i=1}^{n_j} (\boldsymbol{x}_{i|\ell}^{(j)} - \boldsymbol{\mu}_{j|\ell})(\boldsymbol{x}_{i|\ell}^{(j)} - \boldsymbol{\mu}_{j|\ell})^\top, \tag{83}$$

where:

$$\mathbb{R}^{d_{|\ell}} \ni \boldsymbol{\mu}_{j|\ell} := \frac{1}{n_j} \sum_{i=1}^{n_j} \boldsymbol{x}_{i|\ell}^{(j)}, \tag{84}$$

$$\mathbb{R}^{d_{|\ell}} \ni \boldsymbol{\mu}_{|\ell} := \frac{1}{n} \sum_{i=1}^{n} \boldsymbol{x}_{i|\ell}. \tag{85}$$

The ℓ-th Fisher subspace is spanned by the eigenvectors of $\boldsymbol{S}_{W|\ell}^{-1} \boldsymbol{S}_{B|\ell}$.

Hence, z Fisher subspaces are trained. In the test phase, the data instance is projected onto every subspace. If we want to classify the data instance in the projected subspaces, we will have z classification results after projection onto these z subspaces. We can then use majority voting for a final classification of the data instance (Ghojogh & Mohammadzade, 2017). The effectiveness of the majority voting can be explained because of ensemble learning (Polikar, 2012; Ghojogh & Crowley, 2019b). We can also normalize the distances in the subspaces of Fisher forest for the sake of classification (see (Ghojogh & Mohammadzade, 2017) for more details).

9. Kernel Fisher Discriminant Analysis

9.1. Kernels and Hilbert Space

Suppose that $\boldsymbol{\phi} : \boldsymbol{x} \to \mathcal{H}$ is a function which maps the data $\boldsymbol{x}$ to Hilbert space (feature space). The $\boldsymbol{\phi}$ is called *pulling function*. In other words, $\boldsymbol{x} \mapsto \boldsymbol{\phi}(\boldsymbol{x})$. Let t denote the dimensionality of the feature space, i.e., $\boldsymbol{\phi}(\boldsymbol{x}) \in \mathbb{R}^t$ while $\boldsymbol{x} \in \mathbb{R}^d$. Note that we usually have $t \gg d$.

If $\mathcal{X}$ denotes the set of points, i.e., $\boldsymbol{x} \in \mathcal{X}$, the kernel of two vectors $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ is $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ and is defined as (Hofmann et al., 2008; Herbrich, 2001):

$$k(\boldsymbol{x}_1, \boldsymbol{x}_2) := \boldsymbol{\phi}(\boldsymbol{x}_1)^\top \boldsymbol{\phi}(\boldsymbol{x}_2), \tag{86}$$

which is a measure of *similarity* between the two vectors because the inner product captures similarity.

We can compute the kernel of two matrices $\boldsymbol{X}_1 \in \mathbb{R}^{d \times n_1}$ and $\boldsymbol{X}_2 \in \mathbb{R}^{d \times n_2}$ and have a *kernel matrix* (also called *Gram matrix*):

$$\mathbb{R}^{n_1 \times n_2} \ni \boldsymbol{K}(\boldsymbol{X}_1, \boldsymbol{X}_2) := \boldsymbol{\Phi}(\boldsymbol{X}_1)^\top \boldsymbol{\Phi}(\boldsymbol{X}_2), \tag{87}$$

where $\boldsymbol{\Phi}(\boldsymbol{X}_1) := [\boldsymbol{\phi}(\boldsymbol{x}_1), \dots, \boldsymbol{\phi}(\boldsymbol{x}_n)] \in \mathbb{R}^{t \times n_1}$ is the matrix of mapped $\boldsymbol{X}_1$ to the feature space. The $\boldsymbol{\Phi}(\boldsymbol{X}_2) \in \mathbb{R}^{t \times n_2}$ is defined similarly. We can compute the kernel matrix of dataset $\boldsymbol{X} \in \mathbb{R}^{d \times n}$ over itself:

$$\mathbb{R}^{n \times n} \ni \boldsymbol{K}_x := \boldsymbol{K}(\boldsymbol{X}, \boldsymbol{X}) = \boldsymbol{\Phi}(\boldsymbol{X})^\top \boldsymbol{\Phi}(\boldsymbol{X}), \tag{88}$$

where $\boldsymbol{\Phi}(\boldsymbol{X}) := [\boldsymbol{\phi}(\boldsymbol{x}_1), \dots, \boldsymbol{\phi}(\boldsymbol{x}_n)] \in \mathbb{R}^{t \times n}$ is the pulled (mapped) data.

Note that in kernel methods, the pulled data $\boldsymbol{\Phi}(\boldsymbol{X})$ are usually not available and merely the kernel matrix $\boldsymbol{K}(\boldsymbol{X}, \boldsymbol{X})$, which is the inner product of the pulled data with itself, is available.

There exist different types of kernels. Some of the most well-known kernels are:

$$\text{Linear: } k(\boldsymbol{x}_1, \boldsymbol{x}_2) = \boldsymbol{x}_1^\top \boldsymbol{x}_2 + c_1, \tag{89}$$

$$\text{Polynomial: } k(\boldsymbol{x}_1, \boldsymbol{x}_2) = (c_1 \boldsymbol{x}_1^\top \boldsymbol{x}_2 + c_2)^{c_3}, \tag{90}$$

$$\text{Gaussian: } k(\boldsymbol{x}_1, \boldsymbol{x}_2) = \exp\Big(-\frac{||\boldsymbol{x}_1 - \boldsymbol{x}_2||_2^2}{2\sigma^2}\Big), \tag{91}$$

$$\text{Sigmoid: } k(\boldsymbol{x}_1, \boldsymbol{x}_2) = \tanh(c_1 \boldsymbol{x}_1^\top \boldsymbol{x}_2 + c_2), \tag{92}$$

where c_1, c_2, c_3, and σ are scalar constants. The Gaussian and Sigmoid kernels are also called Radial Basis Function (RBF) and hyperbolic tangent, respectively. Note that the Gaussian kernel can also be written as $\exp\big(-\gamma||\boldsymbol{x}_1 - \boldsymbol{x}_2||_2^2\big)$ where $\gamma > 0$.

It is noteworthy to mention that in the RBF kernel, the dimensionality of the feature space is infinite. The reason lies in the Maclaurin series expansion (Taylor series expansion at zero) of this kernel:

$$\exp(-\gamma r) \approx 1 - \gamma r + \frac{\gamma^2}{2!} r^2 - \frac{\gamma^3}{3!} r^3 + \dots,$$

where $r := ||\boldsymbol{x}_1 - \boldsymbol{x}_2||_2^2$, which is infinite dimensional with respect to r.

9.2. One-dimensional Subspace

9.2.1. Scatters in Two-Class Case

The Eq. (26) in the feature space is:

$$\mathbb{R}^{t \times t} \ni \boldsymbol{\Phi}(\boldsymbol{S}_B) := \big(\boldsymbol{\phi}(\boldsymbol{\mu}_1) - \boldsymbol{\phi}(\boldsymbol{\mu}_2)\big)\big(\boldsymbol{\phi}(\boldsymbol{\mu}_1) - \boldsymbol{\phi}(\boldsymbol{\mu}_2)\big)^\top, \tag{93}$$

where the mean of the j-th class in the feature space is:

$$\mathbb{R}^t \ni \boldsymbol{\phi}(\boldsymbol{\mu}_j) := \frac{1}{n_j} \sum_{i=1}^{n_j} \boldsymbol{\phi}(\boldsymbol{x}_i^{(j)}). \tag{94}$$

According to the representation theory (Alperin, 1993), any solution (direction) $\boldsymbol{\phi}(\boldsymbol{u}) \in \mathcal{H}$ must lie in the span of "all" the training vectors mapped to $\mathcal{H}$, i.e., $\boldsymbol{\Phi}(\boldsymbol{X}) = [\boldsymbol{\phi}(\boldsymbol{x}_1), \dots, \boldsymbol{\phi}(\boldsymbol{x}_n)] \in \mathbb{R}^{t \times n}$ (usually $t \gg d$). Note that $\mathcal{H}$ denotes the Hilbert space (feature space). Therefore, we can state that:

$$\mathbb{R}^t \ni \boldsymbol{\phi}(\boldsymbol{u}) = \sum_{i=1}^{n} \theta_i \, \boldsymbol{\phi}(\boldsymbol{x}_i) = \boldsymbol{\Phi}(\boldsymbol{X})\, \boldsymbol{\theta}, \tag{95}$$

where $\mathbb{R}^n \ni \boldsymbol{\theta} := [\theta_1, \dots, \theta_n]^\top$ is the unknown vector of coefficients, and $\boldsymbol{\phi}(\boldsymbol{u}) \in \mathbb{R}^t$ is the pulled Fisher direction to the feature space. The pulled directions can be put together in $\mathbb{R}^{t \times p} \ni \boldsymbol{\Phi}(\boldsymbol{U}) := [\boldsymbol{\phi}(\boldsymbol{u}_1), \dots, \boldsymbol{\phi}(\boldsymbol{u}_p)]$:

$$\mathbb{R}^{t \times p} \ni \boldsymbol{\Phi}(\boldsymbol{U}) = \boldsymbol{\Phi}(\boldsymbol{X})\, \boldsymbol{\Theta}, \tag{96}$$

where $\boldsymbol{\Theta} := [\boldsymbol{\theta}_1, \dots, \boldsymbol{\theta}_p] \in \mathbb{R}^{n \times p}$.

The d_B in the feature space is:

$$\mathbb{R} \ni d_B := \boldsymbol{\phi}(\boldsymbol{u})^\top \boldsymbol{\Phi}(\boldsymbol{S}_B)\, \boldsymbol{\phi}(\boldsymbol{u}) \tag{97}$$

$$\overset{(a)}{=} \boldsymbol{\theta}^\top \boldsymbol{\Phi}(\boldsymbol{X})^\top \big(\boldsymbol{\phi}(\boldsymbol{\mu}_1) - \boldsymbol{\phi}(\boldsymbol{\mu}_2)\big) \big(\boldsymbol{\phi}(\boldsymbol{\mu}_1) - \boldsymbol{\phi}(\boldsymbol{\mu}_2)\big)^\top \boldsymbol{\Phi}(\boldsymbol{X})\, \boldsymbol{\theta}, \tag{98}$$

where (a) is because of Eqs. (93). and (95).

For the j-th class (here $j \in \{1, 2\}$), we have:

$$\boldsymbol{\theta}^\top \boldsymbol{\Phi}(\boldsymbol{X})^\top \boldsymbol{\phi}(\boldsymbol{\mu}_j) \overset{(95)}{=} \sum_{i=1}^{n} \theta_i \, \boldsymbol{\phi}(\boldsymbol{x}_i)^\top \boldsymbol{\phi}(\boldsymbol{\mu}_j)$$

$$\overset{(94)}{=} \frac{1}{n_j} \sum_{i=1}^{n} \sum_{k=1}^{n_j} \theta_i \, \boldsymbol{\phi}(\boldsymbol{x}_i)^\top \boldsymbol{\phi}(\boldsymbol{x}_k^{(j)})$$

$$\overset{(86)}{=} \frac{1}{n_j} \sum_{i=1}^{n} \sum_{k=1}^{n_j} \theta_i \, k(\boldsymbol{x}_i, \boldsymbol{x}_k^{(j)}) = \boldsymbol{\theta}^\top \boldsymbol{m}_j, \tag{99}$$

where $\boldsymbol{m}_j \in \mathbb{R}^n$ whose i-th entry is:

$$\boldsymbol{m}_j(i) := \frac{1}{n_j} \sum_{k=1}^{n_j} k(\boldsymbol{x}_i, \boldsymbol{x}_k^{(j)}). \tag{100}$$

Hence, Eq. (98) becomes:

$$d_B \overset{(99)}{=} \boldsymbol{\theta}^\top (\boldsymbol{m}_1 - \boldsymbol{m}_2)(\boldsymbol{m}_1 - \boldsymbol{m}_2)^\top \boldsymbol{\theta} = \boldsymbol{\theta}^\top \boldsymbol{M} \boldsymbol{\theta}, \tag{101}$$

where:

$$\mathbb{R}^{n \times n} \ni \boldsymbol{M} := (\boldsymbol{m}_1 - \boldsymbol{m}_2)(\boldsymbol{m}_1 - \boldsymbol{m}_2)^\top, \tag{102}$$

is the *between-scatter* in kernel FDA. Hence, the Eq. (98) becomes:

$$d_B = \boldsymbol{\phi}(\boldsymbol{u})^\top \boldsymbol{\Phi}(\boldsymbol{S}_B)\, \boldsymbol{\phi}(\boldsymbol{u}) = \boldsymbol{\theta}^\top \boldsymbol{M} \boldsymbol{\theta}. \tag{103}$$

The Eq. (33) in the feature space is:

$$\mathbb{R}^{t \times t} \ni \boldsymbol{\Phi}(\boldsymbol{S}_W) := \sum_{j=1}^{c} \sum_{i=1}^{n_j} \big(\boldsymbol{\phi}(\boldsymbol{x}_i^{(j)}) - \boldsymbol{\phi}(\boldsymbol{\mu}_j)\big)\big(\boldsymbol{\phi}(\boldsymbol{x}_i^{(j)}) - \boldsymbol{\phi}(\boldsymbol{\mu}_j)\big)^\top. \tag{104}$$

The d_W in the feature space is:

$$\begin{aligned}
\mathbb{R} \ni d_W &:= \boldsymbol{\phi}(\boldsymbol{u})^\top \boldsymbol{\Phi}(\boldsymbol{S}_W)\, \boldsymbol{\phi}(\boldsymbol{u}) \\
&\overset{(a)}{=} \Big(\sum_{\ell=1}^{n} \theta_\ell\, \boldsymbol{\phi}(\boldsymbol{x}_\ell)^\top\Big)\Big(\sum_{j=1}^{c}\sum_{i=1}^{n_j} \big(\boldsymbol{\phi}(\boldsymbol{x}_i^{(j)}) - \boldsymbol{\phi}(\boldsymbol{\mu}_j)\big) \\
&\qquad \big(\boldsymbol{\phi}(\boldsymbol{x}_i^{(j)}) - \boldsymbol{\phi}(\boldsymbol{\mu}_j)\big)^\top\Big)\Big(\sum_{k=1}^{n} \theta_k\, \boldsymbol{\phi}(\boldsymbol{x}_k)\Big) \\
&= \sum_{j=1}^{c}\sum_{\ell=1}^{n}\sum_{i=1}^{n_j}\sum_{k=1}^{n} \Big(\theta_\ell\, \boldsymbol{\phi}(\boldsymbol{x}_\ell)^\top \big(\boldsymbol{\phi}(\boldsymbol{x}_i^{(j)}) - \boldsymbol{\phi}(\boldsymbol{\mu}_j)\big) \\
&\qquad \big(\boldsymbol{\phi}(\boldsymbol{x}_i^{(j)}) - \boldsymbol{\phi}(\boldsymbol{\mu}_j)\big)^\top \theta_k\, \boldsymbol{\phi}(\boldsymbol{x}_k)\Big) \\
&\overset{(94)}{=} \sum_{j=1}^{c}\sum_{\ell=1}^{n}\sum_{i=1}^{n_j}\sum_{k=1}^{n} \Big(\theta_\ell\, \boldsymbol{\phi}(\boldsymbol{x}_\ell)^\top \big(\boldsymbol{\phi}(\boldsymbol{x}_i^{(j)}) - \frac{1}{n_j}\sum_{e=1}^{n_j} \boldsymbol{\phi}(\boldsymbol{x}_e^{(j)})\big) \\
&\qquad \big(\boldsymbol{\phi}(\boldsymbol{x}_i^{(j)}) - \frac{1}{n_j}\sum_{z=1}^{n_j} \boldsymbol{\phi}(\boldsymbol{x}_z^{(j)})\big)^\top \theta_k\, \boldsymbol{\phi}(\boldsymbol{x}_k)\Big) \\
&\overset{(86)}{=} \sum_{j=1}^{c}\sum_{\ell=1}^{n}\sum_{i=1}^{n_j}\sum_{k=1}^{n} \Big(\theta_\ell\, k(\boldsymbol{x}_\ell, \boldsymbol{x}_i^{(j)}) - \frac{1}{n_j}\sum_{e=1}^{n_j} \theta_\ell\, k(\boldsymbol{x}_\ell, \boldsymbol{x}_e^{(j)})\Big) \\
&\qquad \Big(\theta_k\, k(\boldsymbol{x}_i^{(j)}, \boldsymbol{x}_k) - \frac{1}{n_j}\sum_{z=1}^{n_j} \theta_k\, k(\boldsymbol{x}_z^{(j)}, \boldsymbol{x}_k)\Big) \\
&\overset{(b)}{=} \sum_{j=1}^{c}\sum_{\ell=1}^{n}\sum_{i=1}^{n_j}\sum_{k=1}^{n} \Big(\theta_\ell\, k(\boldsymbol{x}_\ell, \boldsymbol{x}_i^{(j)}) - \frac{1}{n_j}\sum_{e=1}^{n_j} \theta_\ell\, k(\boldsymbol{x}_\ell, \boldsymbol{x}_e^{(j)})\Big) \\
&\qquad \Big(\theta_k\, k(\boldsymbol{x}_k, \boldsymbol{x}_i^{(j)}) - \frac{1}{n_j}\sum_{z=1}^{n_j} \theta_k\, k(\boldsymbol{x}_k, \boldsymbol{x}_z^{(j)})\Big) \\
&= \sum_{j=1}^{c}\sum_{\ell=1}^{n}\sum_{i=1}^{n_j}\sum_{k=1}^{n} \Big(\theta_\ell\, \theta_k\, k(\boldsymbol{x}_\ell, \boldsymbol{x}_i^{(j)})\, k(\boldsymbol{x}_k, \boldsymbol{x}_i^{(j)}) \\
&\quad - \frac{2\,\theta_\ell\, \theta_k}{n_j}\sum_{z=1}^{n_j} k(\boldsymbol{x}_\ell, \boldsymbol{x}_i^{(j)})\, k(\boldsymbol{x}_k, \boldsymbol{x}_z^{(j)}) \\
&\quad + \frac{\theta_\ell\, \theta_k}{n_j^2}\sum_{e=1}^{n_j}\sum_{z=1}^{n_j} k(\boldsymbol{x}_\ell, \boldsymbol{x}_e^{(j)})\, k(\boldsymbol{x}_k, \boldsymbol{x}_z^{(j)})\Big) \\
&= \sum_{j=1}^{c}\sum_{\ell=1}^{n}\sum_{i=1}^{n_j}\sum_{k=1}^{n} \Big(\theta_\ell\, \theta_k\, k(\boldsymbol{x}_\ell, \boldsymbol{x}_i^{(j)})\, k(\boldsymbol{x}_k, \boldsymbol{x}_i^{(j)}) \\
&\quad - \frac{\theta_\ell\, \theta_k}{n_j}\sum_{z=1}^{n_j} k(\boldsymbol{x}_\ell, \boldsymbol{x}_i^{(j)})\, k(\boldsymbol{x}_k, \boldsymbol{x}_z^{(j)})\Big) \\
&= \sum_{j=1}^{c}\Big(\sum_{\ell=1}^{n}\sum_{i=1}^{n_j}\sum_{k=1}^{n} \Big(\theta_\ell\, \theta_k\, k(\boldsymbol{x}_\ell, \boldsymbol{x}_i^{(j)})\, k(\boldsymbol{x}_k, \boldsymbol{x}_i^{(j)})\Big) \\
&\quad - \sum_{\ell=1}^{n}\sum_{i=1}^{n_j}\sum_{k=1}^{n} \Big(\frac{\theta_\ell\, \theta_k}{n_j}\sum_{z=1}^{n_j} k(\boldsymbol{x}_\ell, \boldsymbol{x}_i^{(j)})\, k(\boldsymbol{x}_k, \boldsymbol{x}_z^{(j)})\Big)\Big) \\
&\overset{(c)}{=} \sum_{j=1}^{c} \big(\boldsymbol{\theta}^\top \boldsymbol{K}_j \boldsymbol{K}_j^\top \boldsymbol{\theta} - \boldsymbol{\theta}^\top \boldsymbol{K}_j \frac{1}{n_j} \mathbf{1}\mathbf{1}^\top \boldsymbol{K}_j^\top \boldsymbol{\theta}\big) \\
&= \sum_{j=1}^{c} \boldsymbol{\theta}^\top \boldsymbol{K}_j \big(\boldsymbol{I} - \frac{1}{n_j}\mathbf{1}\mathbf{1}^\top\big) \boldsymbol{K}_j^\top \boldsymbol{\theta} \\
&\overset{(d)}{=} \sum_{j=1}^{c} \boldsymbol{\theta}^\top \boldsymbol{K}_j \boldsymbol{H}_j \boldsymbol{K}_j^\top \boldsymbol{\theta} = \boldsymbol{\theta}^\top \Big(\sum_{j=1}^{c} \boldsymbol{K}_j \boldsymbol{H}_j \boldsymbol{K}_j^\top\Big) \boldsymbol{\theta},
\end{aligned}$$

where (a) is because of Eqs. (104) and (95), (b) is because $k(\boldsymbol{x}_1, \boldsymbol{x}_2) = k(\boldsymbol{x}_2, \boldsymbol{x}_1) \in \mathbb{R}$, and (c) is because $\boldsymbol{K}_j \in \mathbb{R}^{n \times n_j}$ is the kernel matrix of the whole training data and the training data of the j-th class. The (a, b)-th element of $\boldsymbol{K}_j$ is:

$$\boldsymbol{K}_j(a, b) := k(\boldsymbol{x}_a, \boldsymbol{x}_b^{(j)}). \tag{105}$$

The (d) is because:

$$\mathbb{R}^{n_j \times n_j} \ni \boldsymbol{H}_j := \boldsymbol{I} - \frac{1}{n_j}\boldsymbol{1}\boldsymbol{1}^\top, \tag{106}$$

is the *centering matrix* (see Appendix A in (Ghojogh & Crowley, 2019c)).

We define:

$$\mathbb{R}^{n \times n} \ni \boldsymbol{N} := \sum_{j=1}^{c} \boldsymbol{K}_j \boldsymbol{H}_j \boldsymbol{K}_j^\top, \tag{107}$$

as the *within-scatter* in kernel FDA. Hence, the d_W becomes:

$$d_W = \boldsymbol{\phi}(\boldsymbol{u})^\top \boldsymbol{\Phi}(\boldsymbol{S}_W)\, \boldsymbol{\phi}(\boldsymbol{u}) = \boldsymbol{\theta}^\top \boldsymbol{N} \boldsymbol{\theta}. \tag{108}$$

The kernel Fisher criterion is:

$$f(\boldsymbol{\theta}) := \frac{d_B(\boldsymbol{\theta})}{d_W(\boldsymbol{\theta})} = \frac{\boldsymbol{\phi}(\boldsymbol{u})^\top \boldsymbol{\Phi}(\boldsymbol{S}_B)\, \boldsymbol{\phi}(\boldsymbol{u})}{\boldsymbol{\phi}(\boldsymbol{u})^\top \boldsymbol{\Phi}(\boldsymbol{S}_W)\, \boldsymbol{\phi}(\boldsymbol{u})} = \frac{\boldsymbol{\theta}^\top \boldsymbol{M} \boldsymbol{\theta}}{\boldsymbol{\theta}^\top \boldsymbol{N} \boldsymbol{\theta}}, \tag{109}$$

where the $\boldsymbol{\theta} \in \mathbb{R}^n$ is the *kernel Fisher direction*.

Similar to the solution of Eq. (42), the solution to maximization of Eq. (109) is:

$$\boldsymbol{M}\boldsymbol{\theta} = \lambda\, \boldsymbol{N}\boldsymbol{\theta}, \tag{110}$$

which is a generalized eigenvalue problem $(\boldsymbol{M}, \boldsymbol{N})$ according to (Ghojogh et al., 2019a). The $\boldsymbol{\theta}$ is the eigenvector with the largest eigenvalue (because the optimization is maximization) and the λ is the corresponding eigenvalue. The $\boldsymbol{\theta}$ is the *kernel Fisher direction* or *kernel Fisher axis*.

Again, one possible solution to the generalized eigenvalue problem $(\boldsymbol{M}, \boldsymbol{N})$ is (Ghojogh et al., 2019a):

$$\boldsymbol{\theta} = \textbf{eig}(\boldsymbol{N}^{-1}\boldsymbol{M}), \tag{111}$$

or (Ghojogh et al., 2019a):

$$\boldsymbol{\theta} = \textbf{eig}((\boldsymbol{N} + \varepsilon \boldsymbol{I})^{-1}\boldsymbol{M}), \tag{112}$$

where **eig**$(.)$ denotes the eigenvector of the matrix with the largest eigenvalue.

The projection and reconstruction of the training data point $\boldsymbol{x}_i$ and the out-of-sample data point $\boldsymbol{x}_t$ are:

$$\begin{aligned}
\mathbb{R} \ni \phi(\widetilde{\boldsymbol{x}}_i) &= \boldsymbol{\phi}(\boldsymbol{u})^\top \boldsymbol{\phi}(\boldsymbol{x}_i) \overset{(95)}{=} \boldsymbol{\theta}^\top \boldsymbol{\Phi}(\boldsymbol{X})^\top \boldsymbol{\phi}(\boldsymbol{x}_i) \\
&= \boldsymbol{\theta}^\top \boldsymbol{k}(\boldsymbol{X}, \boldsymbol{x}_i), \qquad (113) \\
\mathbb{R}^t \ni \boldsymbol{\phi}(\widehat{\boldsymbol{x}}_i) &= \boldsymbol{\phi}(\boldsymbol{u}) \boldsymbol{\phi}(\boldsymbol{u})^\top \boldsymbol{\phi}(\boldsymbol{x}_i) \\
&\overset{(95)}{=} \boldsymbol{\Phi}(\boldsymbol{X})\, \boldsymbol{\theta}\boldsymbol{\theta}^\top \boldsymbol{k}(\boldsymbol{X}, \boldsymbol{x}_i), \qquad (114) \\
\mathbb{R} \ni \phi(\widetilde{\boldsymbol{x}}_t) &= \boldsymbol{\theta}^\top \boldsymbol{k}(\boldsymbol{X}, \boldsymbol{x}_t), \qquad (115) \\
\mathbb{R}^t \ni \boldsymbol{\phi}(\widehat{\boldsymbol{x}}_t) &= \boldsymbol{\Phi}(\boldsymbol{X})\, \boldsymbol{\theta}\boldsymbol{\theta}^\top \boldsymbol{k}(\boldsymbol{X}, \boldsymbol{x}_t). \qquad (116)
\end{aligned}$$

However, in reconstruction expressions, the $\boldsymbol{\Phi}(\boldsymbol{X})$ is not necessarily available; therefore, in kernel FDA, similar to kernel PCA (Ghojogh & Crowley, 2019c), *reconstruction cannot be done.* For the whole training and out-of-sample data, the projections are:

$$\begin{aligned}
\mathbb{R}^{1 \times n} \ni \boldsymbol{\Phi}(\widetilde{\boldsymbol{X}}) &= \boldsymbol{\theta}^\top \boldsymbol{K}(\boldsymbol{X}, \boldsymbol{X}), \qquad (117) \\
\mathbb{R}^{1 \times n_t} \ni \boldsymbol{\Phi}(\widetilde{\boldsymbol{X}}_t) &= \boldsymbol{\theta}^\top \boldsymbol{K}(\boldsymbol{X}, \boldsymbol{X}_t). \qquad (118)
\end{aligned}$$

9.2.2. Scatters in Multi-Class Case: Variant 1

In multi-class case for kernel FDA, the *within-scatter* is the same as in the two-class case, which is Eq. (107) and d_W is also Eq. (108). However, the between-scatter is different. The between-scatter, Eq. (29), in the feature space is:

$$\begin{aligned}
\mathbb{R}^{t \times t} \ni \boldsymbol{\Phi}(\boldsymbol{S}_B) :=& \\
&\sum_{j=1}^{c} \big(\boldsymbol{\phi}(\boldsymbol{\mu}_j) - \boldsymbol{\phi}(\boldsymbol{\mu})\big)\big(\boldsymbol{\phi}(\boldsymbol{\mu}_j) - \boldsymbol{\phi}(\boldsymbol{\mu})\big)^\top, \qquad (119)
\end{aligned}$$

where the total mean in the feature space is:

$$\mathbb{R}^t \ni \boldsymbol{\phi}(\boldsymbol{\mu}) := \frac{1}{\sum_{k=1}^{c} n_k} \sum_{j=1}^{c} n_j\, \boldsymbol{\phi}(\boldsymbol{\mu}_j) = \frac{1}{n} \sum_{i=1}^{n} \boldsymbol{\phi}(\boldsymbol{x}_i), \tag{120}$$

The d_B in the feature space is:

$$\begin{aligned}
\mathbb{R} \ni d_B &:= \boldsymbol{\phi}(\boldsymbol{u})^\top \boldsymbol{\Phi}(\boldsymbol{S}_B)\, \boldsymbol{\phi}(\boldsymbol{u}) \\
&\overset{(a)}{=} \sum_{j=1}^{c} \boldsymbol{\theta}^\top \boldsymbol{\Phi}(\boldsymbol{X})^\top \big(\boldsymbol{\phi}(\boldsymbol{\mu}_j) - \boldsymbol{\phi}(\boldsymbol{\mu})\big) \\
&\qquad \big(\boldsymbol{\phi}(\boldsymbol{\mu}_j) - \boldsymbol{\phi}(\boldsymbol{\mu})\big)^\top \boldsymbol{\Phi}(\boldsymbol{X})\, \boldsymbol{\theta}, \qquad (121)
\end{aligned}$$

where (a) is because of Eqs. (119) and (95). We have:

$$\begin{aligned}
\boldsymbol{\theta}^\top \boldsymbol{\Phi}(\boldsymbol{X})^\top \boldsymbol{\phi}(\boldsymbol{\mu}) &\overset{(95)}{=} \sum_{i=1}^{n} \theta_i\, \boldsymbol{\phi}(\boldsymbol{x}_i)^\top \boldsymbol{\phi}(\boldsymbol{\mu}) \\
&\overset{(120)}{=} \frac{1}{n} \sum_{i=1}^{n} \sum_{k=1}^{n} \theta_i\, \boldsymbol{\phi}(\boldsymbol{x}_i)^\top \boldsymbol{\phi}(\boldsymbol{x}_k) \\
&\overset{(86)}{=} \frac{1}{n} \sum_{i=1}^{n} \sum_{k=1}^{n} \theta_i\, k(\boldsymbol{x}_i, \boldsymbol{x}_k) = \boldsymbol{\theta}^\top \boldsymbol{m}_*, \qquad (122)
\end{aligned}$$

where $\boldsymbol{m}_* \in \mathbb{R}^n$ whose i-th entry is:

$$\boldsymbol{m}_*(i) := \frac{1}{n} \sum_{k=1}^{n} k(\boldsymbol{x}_i, \boldsymbol{x}_k). \tag{123}$$

According to Eqs. (99) and (122), the Eq. (121) becomes:

$$d_B = \boldsymbol{\theta}^\top \sum_{j=1}^{c} (\boldsymbol{m}_j - \boldsymbol{m}_*)(\boldsymbol{m}_j - \boldsymbol{m}_*)^\top \boldsymbol{\theta} = \boldsymbol{\theta}^\top \boldsymbol{M} \boldsymbol{\theta}, \tag{124}$$

where:

$$\mathbb{R}^{n\times n} \ni \boldsymbol{M} := \sum_{j=1}^{c} (\boldsymbol{m}_j - \boldsymbol{m}_*)(\boldsymbol{m}_j - \boldsymbol{m}_*)^\top, \quad (125)$$

is the *between-scatter* in kernel FDA. Similar to Eq. (31), some researches consider the following instead:

$$\mathbb{R}^{n\times n} \ni \boldsymbol{M} := \sum_{j=1}^{c} n_j\, (\boldsymbol{m}_j - \boldsymbol{m}_*)(\boldsymbol{m}_j - \boldsymbol{m}_*)^\top. \quad (126)$$

Hence, the Eq. (121) becomes:

$$d_B = \boldsymbol{\phi}(\boldsymbol{u})^\top \boldsymbol{\Phi}(\boldsymbol{S}_B)\, \boldsymbol{\phi}(\boldsymbol{u}) = \boldsymbol{\theta}^\top \boldsymbol{M} \boldsymbol{\theta}, \quad (127)$$

where $\boldsymbol{M}$ here is Eq. (125) or (126).

The Fisher direction is again Eq. (109) and the solution is again the generalized eigenvalue problem $(\boldsymbol{M}, \boldsymbol{N})$ according to (Ghojogh et al., 2019a).

9.2.3. Scatters in Multi-Class Case: Variant 2

Again, in the second version of multi-class case for kernel FDA, the *within-scatter* is the same as in the two-class case, which is Eq. (107) and d_W is also Eq. (108).

For the between scatter in in the second version, we start with the Eqs. (49) and (50). We kernelize the objective function of the Eq. (50):

$$d_T := \boldsymbol{\phi}(\boldsymbol{u})^\top \boldsymbol{\Phi}(\boldsymbol{S}_T)\, \boldsymbol{\phi}(\boldsymbol{u}), \quad (128)$$

where total-scatter, Eq. (36), is pulled as:

$$\mathbb{R}^{t\times t} \ni \boldsymbol{\Phi}(\boldsymbol{S}_T) := \sum_{k=1}^{n} \big(\boldsymbol{\phi}(\boldsymbol{x}_k) - \boldsymbol{\phi}(\boldsymbol{\mu})\big)\big(\boldsymbol{\phi}(\boldsymbol{x}_k) - \boldsymbol{\phi}(\boldsymbol{\mu})\big)^\top. \quad (129)$$

According to Eqs. (95), (128), and (129), we have:

$$d_T = \sum_{k=1}^{n} \boldsymbol{\theta}^\top \boldsymbol{\Phi}(\boldsymbol{X})^\top \big(\boldsymbol{\phi}(\boldsymbol{x}_k) - \boldsymbol{\phi}(\boldsymbol{\mu})\big) \big(\boldsymbol{\phi}(\boldsymbol{x}_k) - \boldsymbol{\phi}(\boldsymbol{\mu})\big)^\top \boldsymbol{\Phi}(\boldsymbol{X})\, \boldsymbol{\theta}.$$

According to Eq. (122), we have:

$$\boldsymbol{\theta}^\top \boldsymbol{\Phi}(\boldsymbol{X})^\top \boldsymbol{\phi}(\boldsymbol{\mu}) = \boldsymbol{\theta}^\top \boldsymbol{m}_*, \quad (130)$$

where $\boldsymbol{m}_*$ is Eq. (123). On the other hand, we have:

$$\boldsymbol{\theta}^\top \boldsymbol{\Phi}(\boldsymbol{X})^\top \boldsymbol{\phi}(\boldsymbol{x}_k) \overset{(95)}{=} \sum_{i=1}^{n} \theta_i\, \boldsymbol{\phi}(\boldsymbol{x}_i)^\top \boldsymbol{\phi}(\boldsymbol{x}_k) \overset{(86)}{=} \sum_{i=1}^{n} \theta_i\, k(\boldsymbol{x}_i, \boldsymbol{x}_k) = \boldsymbol{\theta}^\top \boldsymbol{g}_k, \quad (131)$$

where $\boldsymbol{g}_k \in \mathbb{R}^n$ whose i-th entry is:

$$\boldsymbol{g}_k(i) := k(\boldsymbol{x}_i, \boldsymbol{x}_k). \quad (132)$$

Hence:

$$d_T = \sum_{k=1}^{n} \boldsymbol{\theta}^\top (\boldsymbol{g}_k - \boldsymbol{m}_*)(\boldsymbol{g}_k - \boldsymbol{m}_*)^\top \boldsymbol{\theta} = \boldsymbol{\theta}^\top \boldsymbol{G}\, \boldsymbol{\theta}, \quad (133)$$

where:

$$\mathbb{R}^{n\times n} \ni \boldsymbol{G} := \sum_{k=1}^{n} (\boldsymbol{g}_k - \boldsymbol{m}_*)(\boldsymbol{g}_k - \boldsymbol{m}_*)^\top. \quad (134)$$

The denominator of the Fisher criterion in the feature space is again the Eq. (108).

The optimization will be similar to Eq. (50) but in the feature space:

$$\begin{aligned} &\underset{\boldsymbol{\theta}}{\text{maximize}} && \boldsymbol{\theta}^\top \boldsymbol{G}\, \boldsymbol{\theta} \\ &\text{subject to} && \boldsymbol{\theta}^\top \boldsymbol{N}\, \boldsymbol{\theta} = 1, \end{aligned} \quad (135)$$

whose solution is similarly obtained as:

$$\boldsymbol{G}\, \boldsymbol{\theta} = \lambda\, \boldsymbol{N}\, \boldsymbol{\theta}, \quad (136)$$

which is a generalized eigenvalue problem $(\boldsymbol{G}, \boldsymbol{N})$ according to (Ghojogh et al., 2019a).

9.3. Multi-dimensional Subspace

In the previous section, the one-dimensional kernel Fisher subspace was discussed. In multi-dimensional kernel Fisher subspace, the within- and between-scatters are the same but the fisher criterion is different. According to Eq. (96), the d_B and d_W are:

$$d_B = \textbf{tr}\big(\boldsymbol{\phi}(\boldsymbol{U})^\top \boldsymbol{\Phi}(\boldsymbol{S}_B)\, \boldsymbol{\phi}(\boldsymbol{U})\big) = \textbf{tr}(\boldsymbol{\Theta}^\top \boldsymbol{M} \boldsymbol{\Theta}), \quad (137)$$

$$d_W = \textbf{tr}\big(\boldsymbol{\phi}(\boldsymbol{U})^\top \boldsymbol{\Phi}(\boldsymbol{S}_W)\, \boldsymbol{\phi}(\boldsymbol{U})\big) = \textbf{tr}(\boldsymbol{\Theta}^\top \boldsymbol{N} \boldsymbol{\Theta}), \quad (138)$$

where $\mathbb{R}^{n\times p} \ni \boldsymbol{\Theta} = [\boldsymbol{\theta}_1, \dots, \boldsymbol{\theta}_p]$ and $\boldsymbol{M} \in \mathbb{R}^{n\times n}$ and $\boldsymbol{N} \in \mathbb{R}^{n\times n}$ are the between- and within-scatters, respectively, determined for either two-class or multi-class case.

The Fisher criterion becomes:

$$f(\boldsymbol{\Theta}) := \frac{d_B(\boldsymbol{\Theta})}{d_W(\boldsymbol{\Theta})} = \frac{\textbf{tr}\big(\boldsymbol{\phi}(\boldsymbol{U})^\top \boldsymbol{\Phi}(\boldsymbol{S}_B)\, \boldsymbol{\phi}(\boldsymbol{U})\big)}{\textbf{tr}\big(\boldsymbol{\phi}(\boldsymbol{U})^\top \boldsymbol{\Phi}(\boldsymbol{S}_W)\, \boldsymbol{\phi}(\boldsymbol{U})\big)} = \frac{\textbf{tr}(\boldsymbol{\Theta}^\top \boldsymbol{M} \boldsymbol{\Theta})}{\textbf{tr}(\boldsymbol{\Theta}^\top \boldsymbol{N} \boldsymbol{\Theta})}, \quad (139)$$

where the columns of $\boldsymbol{\Theta}$ are the *kernel Fisher directions*.

Similar to Eq. (54), the solution to maximization of this criterion is:

$$\boldsymbol{M}\boldsymbol{\Theta} = \boldsymbol{N}\boldsymbol{\Theta}\boldsymbol{\Lambda}, \quad (140)$$

Figure 3. The projection directions (ghost faces) of FDA and PCA for the first four classes of facial AT&T dataset.

the generalized eigenvalue problem $(\boldsymbol{M}, \boldsymbol{N})$ according to (Ghojogh et al., 2019a). The columns of $\boldsymbol{\Theta}$ are the eigenvectors sorted from the largest to smallest eigenvalues (because the optimization is maximization) and the diagonal entries of $\boldsymbol{\Lambda}$ are the corresponding eigenvalues.

Again, we can have another variant of kernel FDA for the multi-dimensional sub-space where the optimization is (similar to Eq. (135)):

$$\begin{aligned} &\underset{\boldsymbol{\Theta}}{\text{maximize}} && \textbf{tr}(\boldsymbol{\Theta}^\top \boldsymbol{G}\, \boldsymbol{\Theta}) \\ &\text{subject to} && \boldsymbol{\Theta}^\top \boldsymbol{N}\, \boldsymbol{\Theta} = \boldsymbol{I}, \end{aligned} \tag{141}$$

whose solution is similarly obtained as:

$$\boldsymbol{G}\boldsymbol{\Theta} = \boldsymbol{N}\boldsymbol{\Theta}\boldsymbol{\Lambda}, \tag{142}$$

which is a generalized eigenvalue problem $(\boldsymbol{G}, \boldsymbol{N})$ according to (Ghojogh et al., 2019a).

As mentioned before, in kernel FDA, we do not have reconstruction. The projection of the training data point $\boldsymbol{x}_i$ and the out-of-sample data point $\boldsymbol{x}_t$ are:

$$\begin{aligned} \mathbb{R}^p \ni \boldsymbol{\phi}(\widetilde{\boldsymbol{x}}_i) &= \boldsymbol{\Phi}(\boldsymbol{U})^\top \boldsymbol{\phi}(\boldsymbol{x}_i) \overset{(96)}{=} \boldsymbol{\Theta}^\top \boldsymbol{\Phi}(\boldsymbol{X})^\top \boldsymbol{\phi}(\boldsymbol{x}_i) \\ &= \boldsymbol{\Theta}^\top \boldsymbol{k}(\boldsymbol{X}, \boldsymbol{x}_i), \end{aligned} \tag{143}$$

$$\mathbb{R}^p \ni \boldsymbol{\phi}(\widetilde{\boldsymbol{x}}_t) = \boldsymbol{\Theta}^\top \boldsymbol{k}(\boldsymbol{X}, \boldsymbol{x}_t). \tag{144}$$

For the whole training and out-of-sample data, the projections are:

$$\mathbb{R}^{p \times n} \ni \boldsymbol{\Phi}(\widetilde{\boldsymbol{X}}) = \boldsymbol{\Theta}^\top \boldsymbol{K}(\boldsymbol{X}, \boldsymbol{X}), \tag{145}$$

$$\mathbb{R}^{p \times n_t} \ni \boldsymbol{\Phi}(\widetilde{\boldsymbol{X}}_t) = \boldsymbol{\Theta}^\top \boldsymbol{K}(\boldsymbol{X}, \boldsymbol{X}_t). \tag{146}$$

9.4. Discussion on Dimensionality of the Kernel Fisher Subspace

According to Eq. (107), the rank of the $\boldsymbol{N}$ is at most $\min(n, c)$ because the matrix is $n \times n$ and its calculation includes c iterations. Hence, the rank of $\boldsymbol{N}^{-1}$ is also at most $\min(n, c)$. According to Eq. (126), the rank of the $\boldsymbol{M}$ is at most $\min(n, c-1)$ because the matrix is $n \times n$, we have c iterations in its calculation, and -1 is because of subtracting the mean (refer to the explanation in Section 3.3).

In Eq. (111), we have $\boldsymbol{N}^{-1}\boldsymbol{M}$ whose rank is:

$$\begin{aligned} &\textbf{rank}(\boldsymbol{N}^{-1}\boldsymbol{M}) \leq \min\big(\textbf{rank}(\boldsymbol{N}^{-1}), \textbf{rank}(\boldsymbol{M})\big) \\ &\leq \min\big(\min(n, c), \min(n, c-1)\big) \\ &= \min(n, c, c-1) \overset{(a)}{=} c - 1, \end{aligned} \tag{147}$$

where (a) is because we usually have $c < n$. Therefore, the rank of $\boldsymbol{N}^{-1}\boldsymbol{M}$ is limited because of the rank of $\boldsymbol{M}$ which is at most $c-1$.

According to Eq. (111), the $c-1$ leading eigenvalues will be valid and the rest are zero or very small. Therefore, the p, which is the dimensionality of the kernel Fisher subspace, is at most $c-1$. The $c-1$ leading eigenvectors are considered as the kernel Fisher directions and the rest of eigenvectors are invalid and ignored.

10. Simulations

For the simulations, we used the AT&T face dataset which includes 400 images, 40 subjects, and 10 images per subject. The images of every person have different poses and expressions. For better visualization of separation of classes in the projection subspace, we only used the images of the first four subjects. The dataset, except for reconstruction experiments, was standardized so that its mean and variance became zero and one, respectively.

10.1. Visualization of the Projection Directions

First, we used the entire 40 images for training FDA, kernel FDA, PCA, and kernel PCA where the used kernels are linear, Radial Basis Function (RBF), and cosine kernels. As we have four classes, the number of FDA directions is three. The three FDA directions and the top ten PCA directions for the used dataset are shown in Fig. 3. As can be seen, the projection directions of a facial dataset are some facial features which are like ghost faces. That is why the facial projection directions are also referred to as *ghost faces*. The ghost faces in FDA and PCA are also referred to as *Fisherfaces* (Belhumeur et al., 1997; Etemad & Chellappa, 1997; Zhao et al., 1999) and *eigenfaces* (Turk &

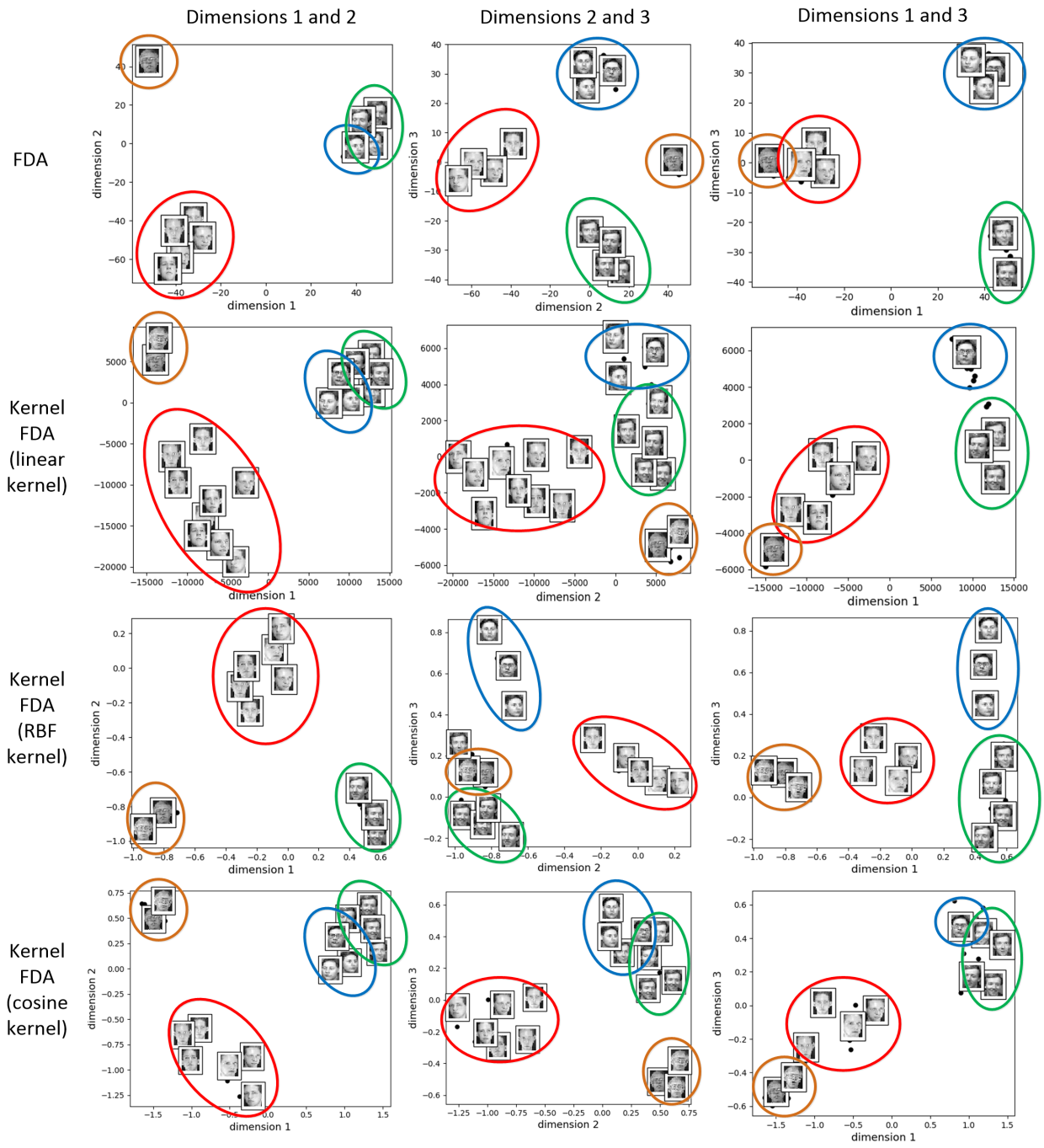

Figure 4. The projection of the first four classes of AT&T dataset onto FDA and kernel FDA subspaces where the used kernels were linear, RBF, and cosine kernels.

Pentland, 1991a;b), respectively. In Fig. 3, the projection directions have captured different facial features which discriminate the data with respect to the maximum variance in PCA and maximum class separation and minimum within class scatter in FDA. The captured features are eyes, eyeglasses, nose, cheeks, chin, lips, eyebrows, and hair, which are the most important facial features. This figure does not include projection directions of kernel FDA and kernel PCA because in kernel FDA, the projection directions are n-dimensional and not d dimensional, and in kernel PCA, the projection directions are not available (see (Ghojogh & Crowley, 2019c)). Note that the face recognition using kernel FDA and kernel PCA are referred to as *kernel Fisherfaces* (Yang, 2002; Liu et al., 2004) and *kernel eigenfaces*

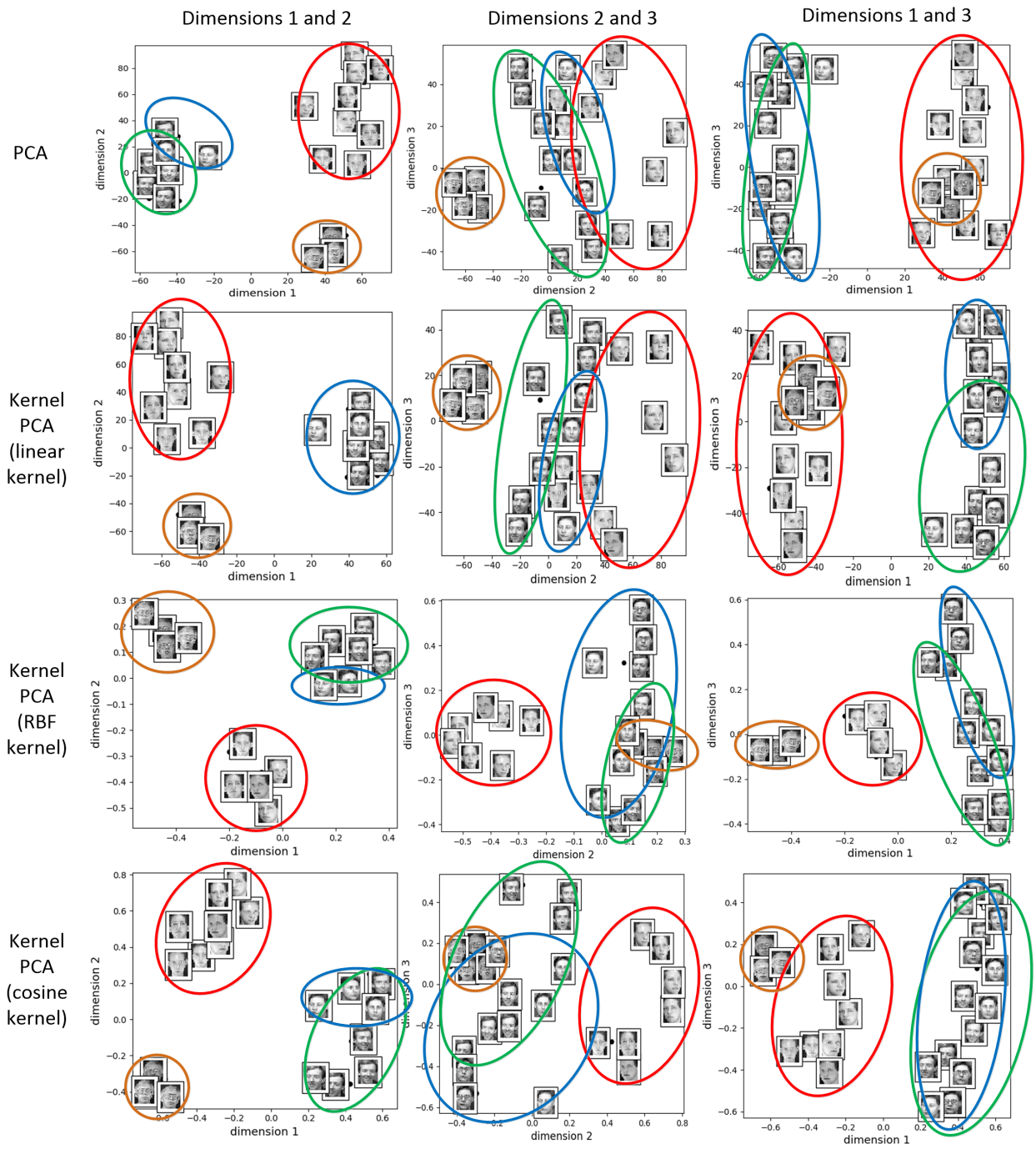

Figure 5. The projection of the first four classes of AT&T dataset onto PCA and kernel PCA subspaces where the used kernels were linear, RBF, and cosine kernels.

(Yang et al., 2000), respectively.

10.2. Projection of the Training Data

The projection of the images onto FDA and kernel FDA subspaces are shown in Fig. 4 where the linear, Radial Basis Function (RBF), and cosine kernels were used. The projection of the images using PCA and kernel PCA are also shown in Fig. 5. As can be seen, the FDA and kernel FDA subspaces have separated the classes much better than the PCA and kernel PCA subspaces. This is because the FDA and kernel FDA make use of the class labels in order to separate the classes in the subspace while the PCA and kernel

PCA only capture the variance (spread) of data regardless of class labels.

10.3. Reconstruction of Images

Figure 6 illustrates the reconstruction of some of the training images. The FDA has used its three projection directions for reconstruction. For reconstruction in PCA, we once used the top three PCA directions and one used the whole d PCA directions.

As can be seen in this figure, the reconstruction of PCA is much better than FDA. This makes sense because PCA is the best linear method for reconstruction having the least squared error (see (Ghojogh & Crowley, 2019c)). However, the responsibility of FDA is not reconstruction but separation of the classes. Thus, the FDA directions try to separate the classes as much as possible and do not *necessarily* care for a good reconstruction. Recall Fig. 2 which shows different cases for FDA and PCA directions. According to this figure, even in some datasets, FDA direction is orthogonal to PCA direction which is the best direction for reconstruction. It is noteworthy that reconstruction cannot be done in kernel FDA, so as in kernel PCA (see (Ghojogh & Crowley, 2019c)) as was mentioned before. Moreover, note that reconstruction can be done in FDA also for the out-of-sample data. Here, for the sake of brevity, we do not provide simulation for it.

10.4. Out-of-sample Projection

We took the first six images of each of the first four subjects in the AT&T dataset as the training images and the rest as the test (out-of-sample) images. The projection of the training and the out-of-sample images onto FDA and kernel FDA (using linear, RBF, and cosine kernels) are shown in Fig. 7. This figure shows that projection of out-of-sample images have been properly carried on in FDA and kernel FDA.

11. Conclusion

This paper was a tutorial paper introducing FDA and kernel FDA in detail. Various concepts about FDA, such as rank of scatters, dimensionality of the subspace, an example for interpretation, robust FDA, equivalency to LDA, and Fisher forest were explained and discussed. Both cases of two- and multi-classes were covered for FDA and kernel FDA. Finally, some simulations were performed to validate the theory in practice and compare to the unsupervised PCA method.

Acknowledgment

The authors hugely thank Prof. Ali Ghodsi (see his great online courses (Ghodsi, 2017; 2015)), Prof. Mu Zhu, Prof. Hoda Mohammadzade, and other professors whose courses have partly covered the materials mentioned in this tutorial paper.

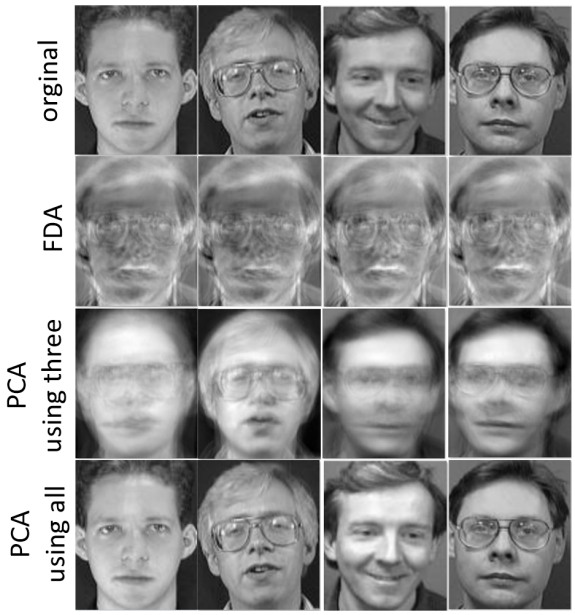

Figure 6. The reconstruction of four sample faces of AT&T datasets in FDA and PCA.

A. Metric Learning

The general form of metric (Peltonen et al., 2004) is usually defined as a form similar to Mahalanobis distance (McLachlan, 1999; De Maesschalck et al., 2000). The metric is:

$$||\boldsymbol{x}_i - \boldsymbol{x}_j||_{\boldsymbol{A}} := (\boldsymbol{x}_i - \boldsymbol{x}_j)^\top \boldsymbol{A}\, (\boldsymbol{x}_i - \boldsymbol{x}_j), \tag{148}$$

where:

$$\boldsymbol{A} = \boldsymbol{U}\boldsymbol{U}^\top \succeq 0, \tag{149}$$

to have a valid distance metric. Most of the metric learning algorithms (Kulis et al., 2013) are optimization problems where $\boldsymbol{A}$ is unknown to make data points in same class (similar pairs) closer to each other, and points in different classes far apart from each other. We have:

$$\begin{aligned} ||\boldsymbol{x}_i - \boldsymbol{x}_j||_{\boldsymbol{A}} &\overset{(149)}{=} (\boldsymbol{x}_i - \boldsymbol{x}_j)^\top \boldsymbol{U}\boldsymbol{U}^\top (\boldsymbol{x}_i - \boldsymbol{x}_j) && (150)\\ &= (\boldsymbol{U}^\top \boldsymbol{x}_i - \boldsymbol{U}^\top \boldsymbol{x}_j)^\top (\boldsymbol{U}^\top \boldsymbol{x}_i - \boldsymbol{U}^\top \boldsymbol{x}_j), && (151) \end{aligned}$$

so this metric is equivalent to projection of data with projection matrix $\boldsymbol{U}$ and then using Euclidean distance in the embedded space (Peltonen et al., 2004). Therefore, Metric learning can be considered as a feature extraction (Ghojogh et al., 2019b) and manifold learning method (Alipanahi et al., 2008; Globerson & Roweis, 2006).

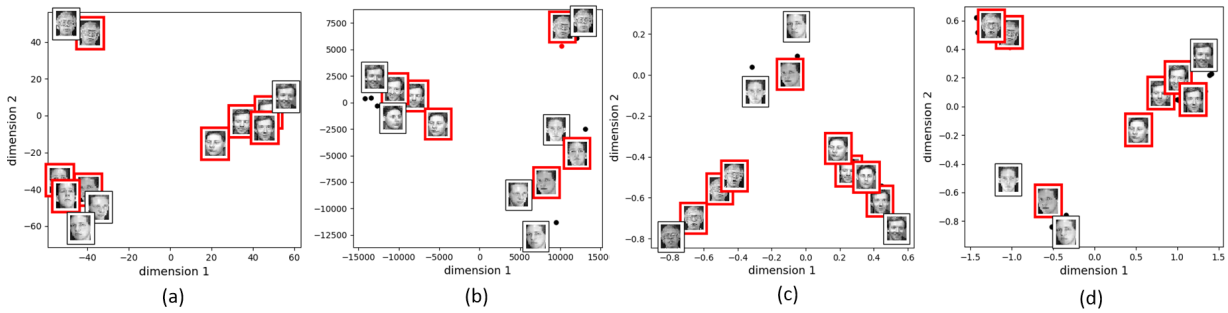

Figure 7. The first two dimensions of the projection of both training and out-of-sample instances in the first four classes of AT&T dataset onto subspaces of (a) FDA, (b) kernel FDA using linear kernel, (c) kernel FDA using RBF kernel, and (d) kernel FDA using cosine kernel.

B. Rayleigh-Ritz Quotient

The *Rayleigh-Ritz quotient* or *Rayleigh quotient* is defined as (Parlett, 1998; Croot, 2005):

$$\mathbb{R} \ni R(\boldsymbol{A}, \boldsymbol{x}) := \frac{\boldsymbol{x}^\top \boldsymbol{A}\, \boldsymbol{x}}{\boldsymbol{x}^\top \boldsymbol{x}}, \tag{152}$$

where $\boldsymbol{A}$ is a symmetric matrix and $\boldsymbol{x}$ is a non-zero vector:

$$\boldsymbol{A} = \boldsymbol{A}^\top, \quad \boldsymbol{x} \neq \boldsymbol{0}. \tag{153}$$

One of the properties of the Rayleigh-Ritz quotient is:

$$R(\boldsymbol{A}, c\boldsymbol{x}) = R(\boldsymbol{A}, \boldsymbol{x}), \tag{154}$$

where c is a scalar. The proof is that:

$$\begin{aligned} R(\boldsymbol{A}, c\boldsymbol{x}) = \frac{(c\boldsymbol{x})^\top \boldsymbol{A}\, c\boldsymbol{x}}{(c\boldsymbol{x})^\top c\boldsymbol{x}} &\overset{(a)}{=} \frac{c\boldsymbol{x}^\top \boldsymbol{A}\, c\boldsymbol{x}}{c\boldsymbol{x}^\top c\boldsymbol{x}} \\ &\overset{(b)}{=} \frac{c^2}{c^2} \times \frac{\boldsymbol{x}^\top \boldsymbol{A}\, \boldsymbol{x}}{\boldsymbol{x}^\top \boldsymbol{x}} \overset{(152)}{=} R(\boldsymbol{A}, \boldsymbol{x}), \end{aligned}$$

where (a) and (b) are because c is a scalar.

Because of the Eq. (154), the optimization of the Rayleigh-Ritz quotient has an equivalent (Croot, 2005):

$$\begin{aligned} &\underset{\boldsymbol{x}}{\text{minimize/maximize}} \quad R(\boldsymbol{A}, \boldsymbol{x}) \overset{(a)}{=} \\ &\underset{\boldsymbol{x}}{\text{minimize/maximize}} \quad R(\boldsymbol{A}, \boldsymbol{x}) \\ &\text{subject to} \quad ||\boldsymbol{x}||_2 = 1, \overset{(b)}{=} \\ &\underset{\boldsymbol{x}}{\text{minimize/maximize}} \quad \boldsymbol{x}^\top \boldsymbol{A}\, \boldsymbol{x} \\ &\text{subject to} \quad ||\boldsymbol{x}||_2 = 1, \end{aligned} \tag{155}$$

where (a) is because if we define $\boldsymbol{y} := (1/||\boldsymbol{x}||_2)\, \boldsymbol{x}$, the Rayleigh-Ritz quotient is:

$$R(\boldsymbol{A}, \boldsymbol{y}) = \frac{\boldsymbol{y}^\top \boldsymbol{A}\, \boldsymbol{y}}{\boldsymbol{y}^\top \boldsymbol{y}} = \frac{1/||\boldsymbol{x}||_2^2}{1/||\boldsymbol{x}||_2^2} \times \frac{\boldsymbol{x}^\top \boldsymbol{A}\, \boldsymbol{x}}{\boldsymbol{x}^\top \boldsymbol{x}} = R(\boldsymbol{A}, \boldsymbol{x}), \tag{156}$$

and:

$$||\boldsymbol{y}||_2^2 = \frac{1}{||\boldsymbol{x}||_2^2} \times ||\boldsymbol{x}||_2^2 = 1 \implies ||\boldsymbol{y}||_2 = 1. \tag{157}$$

Thus, we have $R(\boldsymbol{A}, \boldsymbol{y})$ subject to $||\boldsymbol{y}||_2 = 1$. Changing the dummy variable $\boldsymbol{y}$ to $\boldsymbol{x}$ gives the Eq. (155). The (b) notices $\boldsymbol{x}^\top \boldsymbol{x} = 1$ because of the constraint $||\boldsymbol{x}||_2 = 1$.

Note that the constraint in Eq. (155) can be equal to any constant which is proved similarly. Moreover, note that the value of constant in the constraint is not important because it will be removed after taking derivative from the Lagrangian in optimization (Boyd & Vandenberghe, 2004).

The *generalized Rayleigh-Ritz quotient* or *generalized Rayleigh quotient* is defined as (Parlett, 1998; Ghojogh et al., 2019a):

$$\mathbb{R} \ni R(\boldsymbol{A}, \boldsymbol{B}; \boldsymbol{x}) := \frac{\boldsymbol{x}^\top \boldsymbol{A}\, \boldsymbol{x}}{\boldsymbol{x}^\top \boldsymbol{B}\, \boldsymbol{x}}, \tag{158}$$

where $\boldsymbol{A}$ and $\boldsymbol{B}$ are symmetric matrices and $\boldsymbol{x}$ is a non-zero vector:

$$\boldsymbol{A} = \boldsymbol{A}^\top, \quad \boldsymbol{B} = \boldsymbol{B}^\top, \quad \boldsymbol{x} \neq \boldsymbol{0}. \tag{159}$$

If the symmetric $\boldsymbol{B}$ is positive definite:

$$\boldsymbol{B} \succ 0, \tag{160}$$

it has a Cholesky decomposition:

$$\boldsymbol{B} = \boldsymbol{C}\boldsymbol{C}^\top, \tag{161}$$

where $\boldsymbol{C}$ is a lower triangular matrix. In case $\boldsymbol{B} \succ 0$, the generalized Rayleigh-Ritz quotient can be converted to a Rayleigh-Ritz quotient:

$$R(\boldsymbol{A}, \boldsymbol{B}; \boldsymbol{x}) = R(\boldsymbol{D}, \boldsymbol{C}^\top \boldsymbol{x}), \tag{162}$$

where:

$$\boldsymbol{D} := \boldsymbol{C}^{-1} \boldsymbol{A} \boldsymbol{C}^{-\top}. \tag{163}$$

The proof is:

$$\begin{aligned}\text{RHS} = R(\boldsymbol{D}, \boldsymbol{C}^\top \boldsymbol{x}) &\overset{(152)}{=} \frac{(\boldsymbol{C}^\top \boldsymbol{x})^\top \boldsymbol{D} (\boldsymbol{C}^\top \boldsymbol{x})}{(\boldsymbol{C}^\top \boldsymbol{x})^\top (\boldsymbol{C}^\top \boldsymbol{x})} \\ &\overset{(163)}{=} \frac{\boldsymbol{x}^\top \boldsymbol{C}\boldsymbol{C}^{-1} \boldsymbol{A} (\boldsymbol{C}\boldsymbol{C}^{-1})^\top \boldsymbol{x}}{\boldsymbol{x}^\top (\boldsymbol{C}\boldsymbol{C}^\top) \boldsymbol{x}} \overset{(a)}{=} \frac{\boldsymbol{x}^\top \boldsymbol{A}\, \boldsymbol{x}}{\boldsymbol{x}^\top \boldsymbol{B}\, \boldsymbol{x}} \\ &\overset{(158)}{=} R(\boldsymbol{A}, \boldsymbol{B}; \boldsymbol{x}) = \text{LHS}, \quad \text{Q.E.D.},\end{aligned}$$

where RHS and LHS are short for right and left hand sides and (a) is because of Eq. (161) and $\boldsymbol{C}\boldsymbol{C}^{-1} = \boldsymbol{I}$ because $\boldsymbol{C}$ is a square matrix.

Similarly, one of the properties of the generalized Rayleigh-Ritz quotient is:

$$R(\boldsymbol{A}, \boldsymbol{B}; c\boldsymbol{x}) = R(\boldsymbol{A}, \boldsymbol{B}; \boldsymbol{x}), \tag{164}$$

where c is a scalar. The proof is that:

$$\begin{aligned}R(\boldsymbol{A}, \boldsymbol{B}; c\boldsymbol{x}) = \frac{(c\boldsymbol{x})^\top \boldsymbol{A}\, c\boldsymbol{x}}{(c\boldsymbol{x})^\top \boldsymbol{B}\, c\boldsymbol{x}} &\overset{(a)}{=} \frac{c\boldsymbol{x}^\top \boldsymbol{A}\, c\boldsymbol{x}}{c\boldsymbol{x}^\top \boldsymbol{B}\, c\boldsymbol{x}} \\ &\overset{(b)}{=} \frac{c^2}{c^2} \times \frac{\boldsymbol{x}^\top \boldsymbol{A}\, \boldsymbol{x}}{\boldsymbol{x}^\top \boldsymbol{B}\, \boldsymbol{x}} \overset{(158)}{=} R(\boldsymbol{A}, \boldsymbol{B}; \boldsymbol{x}),\end{aligned}$$

where (a) and (b) are because c is a scalar.

Because of the Eq. (164), the optimization of the generalized Rayleigh-Ritz quotient has an equivalent:

$$\begin{aligned}&\underset{\boldsymbol{x}}{\text{minimize/maximize}} \;\; R(\boldsymbol{A}, \boldsymbol{B}; \boldsymbol{x}) \equiv \\ &\underset{\boldsymbol{x}}{\text{minimize/maximize}} \quad \boldsymbol{x}^\top \boldsymbol{A}\, \boldsymbol{x} \\ &\text{subject to} \qquad\qquad\;\; \boldsymbol{x}^\top \boldsymbol{B}\, \boldsymbol{x} = 1,\end{aligned} \tag{165}$$

for a similar reason that we provided for the Rayleigh-Ritz quotient. the constraint can be equal to any constant because in the derivative of Lagrangian, the constant will be dropped.

References

Alipanahi, Babak, Biggs, Michael, and Ghodsi, Ali. Distance metric learning vs. Fisher discriminant analysis. In *Proceedings of the 23rd national conference on Artificial intelligence*, volume 2, pp. 598–603, 2008.

Alperin, Jonathan L. *Local representation theory: Modular representations as an introduction to the local representation theory of finite groups*, volume 11. Cambridge University Press, 1993.

Belhumeur, Peter N, Hespanha, João P, and Kriegman, David J. Eigenfaces vs. Fisherfaces: Recognition using class specific linear projection. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (7):711–720, 1997.

Boyd, Stephen and Vandenberghe, Lieven. *Convex optimization*. Cambridge university press, 2004.

Croot, Ernie. The Rayleigh principle for finding eigenvalues. Technical report, Georgia Institute of Technology, School of Mathematics, 2005. Online: http://people.math.gatech.edu/∼ecroot/notes_linear.pdf, Accessed: March 2019.

De Maesschalck, Roy, Jouan-Rimbaud, Delphine, and Massart, Désiré L. The Mahalanobis distance. *Chemometrics and intelligent laboratory systems*, 50(1):1–18, 2000.

Deng, Weihong, Hu, Jiani, Guo, Jun, and Zhang, Honggang. Robust discriminant analysis of gabor feature for face recognition. In *Fourth International Conference on Fuzzy Systems and Knowledge Discovery (FSKD 2007)*, volume 3, pp. 248–252. IEEE, 2007.

Etemad, Kamran and Chellappa, Rama. Discriminant analysis for recognition of human face images. *Journal of the Optical Society of America A*, 14(8):1724–1733, 1997.

Fisher, Ronald A. Xv.the correlation between relatives on the supposition of mendelian inheritance. *Earth and Environmental Science Transactions of the Royal Society of Edinburgh*, 52(2):399–433, 1919.

Fisher, Ronald A. The use of multiple measurements in taxonomic problems. *Annals of eugenics*, 7(2):179–188, 1936.

Fisher, Ronald Aylmer. Statistical methods for research workers. In *Breakthroughs in statistics*, pp. 66–70. Springer, 1992.

Frieden, B Roy. *Science from Fisher information: a unification*. Cambridge University Press, 2004.

Friedman, Jerome, Hastie, Trevor, and Tibshirani, Robert. *The elements of statistical learning*, volume 2. Springer series in statistics New York, NY, USA, 2009.

Ghodsi, Ali. Classification course, department of statistics and actuarial science, university of Waterloo. Online Youtube Videos, 2015. Accessed: January 2019.

Ghodsi, Ali. Data visualization course, department of statistics and actuarial science, university of Waterloo. Online Youtube Videos, 2017. Accessed: January 2019.

Ghojogh, Benyamin and Crowley, Mark. Linear and quadratic discriminant analysis: Tutorial. *arXiv preprint arXiv:1906.02590*, 2019a.

Ghojogh, Benyamin and Crowley, Mark. The theory behind overfitting, cross validation, regularization, bagging, and boosting: Tutorial. *arXiv preprint arXiv:1905.12787*, 2019b.

Ghojogh, Benyamin and Crowley, Mark. Unsupervised and supervised principal component analysis: Tutorial. *arXiv preprint arXiv:1906.03148*, 2019c.

Ghojogh, Benyamin and Mohammadzade, Hoda. Automatic extraction of key-poses and key-joints for action recognition using 3d skeleton data. In *2017 10th Iranian Conference on Machine Vision and Image Processing (MVIP)*, pp. 164–170. IEEE, 2017.

Ghojogh, Benyamin, Mohammadzade, Hoda, and Mokari, Mozhgan. Fisherposes for human action recognition using kinect sensor data. *IEEE Sensors Journal*, 18(4): 1612–1627, 2017.

Ghojogh, Benyamin, Karray, Fakhri, and Crowley, Mark. Eigenvalue and generalized eigenvalue problems: Tutorial. *arXiv preprint arXiv:1903.11240*, 2019a.

Ghojogh, Benyamin, Samad, Maria N, Mashhadi, Sayema Asif, Kapoor, Tania, Ali, Wahab, Karray, Fakhri, and Crowley, Mark. Feature selection and feature extraction in pattern analysis: A literature review. *arXiv preprint arXiv:1905.02845*, 2019b.

Globerson, Amir and Roweis, Sam T. Metric learning by collapsing classes. In *Advances in neural information processing systems*, pp. 451–458, 2006.

Guo, Ming and Wang, Zhelong. A feature extraction method for human action recognition using body-worn inertial sensors. In *2015 IEEE 19th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, pp. 576–581. IEEE, 2015.

Herbrich, Ralf. *Learning kernel classifiers: theory and algorithms*. Mit Press, 2001.

Hofmann, Thomas, Schölkopf, Bernhard, and Smola, Alexander J. Kernel methods in machine learning. *The annals of statistics*, pp. 1171–1220, 2008.

Kulis, Brian et al. Metric learning: A survey. *Foundations and Trends® in Machine Learning*, 5(4):287–364, 2013.

Liu, Qingshan, Lu, Hanqing, and Ma, Songde. Improving kernel Fisher discriminant analysis for face recognition. *IEEE transactions on circuits and systems for video technology*, 14(1):42–49, 2004.

McLachlan, Goeffrey J. Mahalanobis distance. *Resonance*, 4(6):20–26, 1999.

Mika, Sebastian, Rätsch, Gunnar, Weston, Jason, Schölkopf, Bernhard, and Müller, Klaus-Robert. Fisher discriminant analysis with kernels. In *Proceedings of the 1999 IEEE signal processing society workshop on Neural networks for signal processing IX*, pp. 41–48. IEEE, 1999.

Mika, Sebastian, Rätsch, Gunnar, Weston, Jason, Schölkopf, Bernhard, Smola, Alex J, and Müller, Klaus-Robert. Invariant feature extraction and classification in kernel spaces. In *Advances in neural information processing systems*, pp. 526–532, 2000.

Mokari, Mozhgan, Mohammadzade, Hoda, and Ghojogh, Benyamin. Recognizing involuntary actions from 3d skeleton data using body states. *Scientia Iranica*, 2018.

Parlett, Beresford N. The symmetric eigenvalue problem. *Classics in Applied Mathematics*, 20, 1998.

Peltonen, Jaakko, Klami, Arto, and Kaski, Samuel. Improved learning of Riemannian metrics for exploratory analysis. *Neural Networks*, 17(8-9):1087–1100, 2004.

Polikar, Robi. Ensemble learning. In *Ensemble machine learning*, pp. 1–34. Springer, 2012.

Samadani, Ali-Akbar, Ghodsi, Ali, and Kulić, Dana. Discriminative functional analysis of human movements. *Pattern Recognition Letters*, 34(15):1829–1839, 2013.

Sugiyama, Masashi. Dimensionality reduction of multimodal labeled data by local fisher discriminant analysis. *Journal of machine learning research*, 8(May):1027–1061, 2007.

Turk, Matthew and Pentland, Alex. Eigenfaces for recognition. *Journal of cognitive neuroscience*, 3(1):71–86, 1991a.

Turk, Matthew A and Pentland, Alex P. Face recognition using eigenfaces. In *Computer Vision and Pattern Recognition, 1991. Proceedings CVPR'91., IEEE Computer Society Conference on*, pp. 586–591. IEEE, 1991b.

Wang, Ruye. Generalized eigenvalue problem. http://fourier.eng.hmc.edu/e161/lectures/algebra/node7.html, 2015. Accessed: January 2019.

Wang, Yanxia and Ruan, Qiuqi. Kernel fisher discriminant analysis for palmprint recognition. In *18th International Conference on Pattern Recognition (ICPR'06)*, volume 4, pp. 457–460. IEEE, 2006.

Welling, Max. Fisher linear discriminant analysis. Technical report, Department of Computer Science, University of Toronto, 2005.

Xu, Yong and Lu, Guangming. Analysis on Fisher discriminant criterion and linear separability of feature space. In *2006 International Conference on Computational Intelligence and Security*, volume 2, pp. 1671–1676. IEEE, 2006.

Yang, M-H, Ahuja, Narendra, and Kriegman, David. Face recognition using kernel eigenfaces. In *Image processing, 2000. proceedings. 2000 international conference on*, volume 1, pp. 37–40. IEEE, 2000.

Yang, Ming-Hsuan. Kernel Eigenfaces vs. kernel Fisherfaces: Face recognition using kernel methods. In *Proceedings of the fifth IEEE international conference on automatic face and gesture recognition*, pp. 215–220, 2002.

Zhao, Wenyi, Chellappa, Rama, and Phillips, P Jonathon. *Subspace linear discriminant analysis for face recognition*. Citeseer, 1999.