Robust Fast Subclass Discriminant Analysis

Kateryna Chumachenko
Department of Computing Sciences
Tampere University
Tampere, Finland
kateryna.chumachenko@tuni.fi

Alexandros Iosifidis
Department of Engineering
Aarhus University
Aarhus, Denmark
ai@eng.au.dk

Moncef Gabbouj
Department of Computing Sciences
Tampere University
Tampere, Finland
moncef.gabbouj@tuni.fi

***Abstract*—In this paper, we propose novel methods to address the challenges of dimensionality reduction related to potential outlier classes and imbalanced classes often present in data. In particular, we propose extensions to Fast Subclass Discriminant Analysis and Subclass Discriminant Analysis that allow to put more attention on uder-represented classes or classes that are likely to be confused with each other. Furthermore, the kernelized variants of the proposed algorithms are presented. The proposed methods lead to faster training time and improved accuracy as shown by experiments on eight datasets of different domains, tasks, and sizes.**

***Index Terms*—subclass discriminant analysis, subspace learning, dimensionality reduction**

I. Introduction

Dimensionality reduction has acquired an important role within modern machine learning techniques driven by the increase in the availability of high-dimensional data, such as images, videos, and sensor data. Subspace learning is one of the common approaches to dimensionality reduction, the goal of which is to find a subspace of the original data, projection onto which would satisfy a certain statistical criteria defined for the projected data while reducing the number of features.

Notable approaches in the area of subspace learning include Linear Discriminant Analysis (LDA) [1]–[3] and Principal Component Analysis (PCA) [4]. PCA is a basic unsupervised method that finds the projection space where the data would have the highest variance. LDA is a supervised method that seeks to find a subspace that would ensure a high between-class variance and a low within-class variance. However, LDA suffers from several limitations: first, the assumption of unimodality of each class generally does not hold in real-world data, resulting in decreased accuracy; second, the maximal dimensionality of the learnt space is limited by the number of classes; third, the solution relies on eigendecomposition which is computationally intensive in the cases of high-dimensional or large-scale data.

A step towards relaxing the limitations of LDA was taken by introducing Subclass Discriminant Analysis (SDA) [5] that relies on representing the data of each class with several subclasses. This resolves the unimodality assumption limitation and increases the maximal dimensionality of the projection space. However, SDA still relies on eigendecomposition hence being slow for large-scale and high-dimensional data. An approach to overcoming the speed limitation was recently proposed by introducing Fast Subclass Discriminant Analysis (fastSDA) along with its incremental solution [6], [7].

This work was supported by the Business Finland project 5G VIIMA.

LDA, SDA, and their variants assume that different classes contain a similar amount of discriminative information and thus equal attention is given to each of the classes when learning the projection matrix. Such a situation is, however, unrealistic in real-world scenarios where the data is likely to have outliers or an imbalance between classes. A straightforward example is the case where different classes have significantly different numbers of samples, resulting in implicitly biasing the model to learning the discriminative features of the larger class and performing poorly on the under-represented classes. Besides, the discriminative information of some classes might be more useful than that of the others even under the condition of balanced classes. For example, a class that lies far from the others in the original space will put more weight to learning the projection matrix, while it is less likely to be confused with other classes. Instead, we would like to pay more attention to the classes that can easily be confused.

Solutions overcoming the above-mentioned limitations have been proposed [8], [9]; however, they are mainly relying on the assumptions of LDA on unimodality of classes and utilize the computationally intensive eigendecomposition.

In this work, we propose a novel weighting approach to Fast Subclass Discriminant Analysis that would preserve the benefits of the method in terms of speed and relaxation of unimodality assumption, while accounting for potential class imbalance or presence of outlier classes. Besides, we show how the proposed weighting strategies can be incorporated into eigendecomposition-based SDA. We perform experiments on 8 datasets of different domains, tasks, and sizes, and the experimental results show the superiority of the proposed approaches compared to other methods.

II. Related Work

One of the classical approaches in supervised dimensionality reduction is Linear Discriminant Analysis (LDA) that represents the data of each class with a unimodal Gaussian distribution and seeks for the subspace in which the between-class scatter of the data would be maximized, while minimizing the within-class scatter. Projection onto such subspace results in compact classes lying far from each other, hence, leading to high discrimination between classes. This is achieved by optimizing the Fisher-Rao criterion [10]:

$$\mathcal{J}(\mathbf{W}) = \underset{\mathbf{W}^T\mathbf{W}=\mathbf{I}}{argmin} \frac{Tr(\mathbf{W}^T\mathbf{S}_w\mathbf{W})}{Tr(\mathbf{W}^T\mathbf{S}_b\mathbf{W})}, \quad (1)$$

where $Tr()$ denotes the trace operator.

Such formulation is rather simplistic in a way that it relies on an unrealistic assumption of unimodality of data which is rarely present in real-world problems. Besides, the potential dimensionality of the projection space is limited by the rank of the between-class scatter matrix which is equal to $C-1$, where C is the number of classes.

A. Subclass Discriminant Analysis

Subclass Discriminant Analysis was proposed as an extension to LDA that would make it more suitable for real-world data where the unimodality assumption does not hold. SDA represents the data of each class with several subclasses obtained using a certain clustering algorithm and defines the new between-class and within-class scatter matrices based on the distances between subclass means. This generally results in better performance and allows to increase the potential dimensionality of a subspace to $\sum_i d_i - 1$, where d_i is the number of subclasses in class i. The minimization of the within-class scatter matrix $\mathbf{S}_w$ in (1) is equivalent to the minimization of the total scatter $\mathbf{S}_t$ given that the between-class scatter $\mathbf{S}_b$ is maximized, as $\mathbf{S}_t = \mathbf{S}_b + \mathbf{S}_w$. Therefore, SDA criterion can be formulated using the following matrices:

$$\mathbf{S}_t = \sum_{q=1}^{N} (\mathbf{x}_q - \boldsymbol{\mu})(\mathbf{x}_q - \boldsymbol{\mu})^T, \quad (2)$$

$$\mathbf{S}_b = \sum_{i=1}^{C-1} \sum_{l=i+1}^{C} \sum_{j=1}^{d_i} \sum_{h=1}^{d_l} p_{ij} p_{lh} (\boldsymbol{\mu}_{ij} - \boldsymbol{\mu}_{lh})(\boldsymbol{\mu}_{ij} - \boldsymbol{\mu}_{lh})^T, \quad (3)$$

where $\boldsymbol{\mu}$ is the mean of data, i and l are class labels, j and h are subclass labels, p_{ij} and p_{lh} are the subclass priors, $p_{ij} = \frac{N_{ij}}{N}$, where N_{ij} is the number of samples in subclass j of class i and N is the total number of samples in the dataset. The solution is given by the generalized eigendecomposition problem

$$\mathbf{S}_t \mathbf{w} = \lambda \mathbf{S}_b \mathbf{w}, \quad (4)$$

and the obtained eigenvectors $[\mathbf{w}_1, \mathbf{w}_2, ..., \mathbf{w}_d]$ that correspond to d minimal eigenvalues form a projection matrix $\mathbf{W}$ which can be used for projecting the data to the d-dimensional discriminant subspace as follows: $\mathbf{y}_i = \mathbf{W}^T\mathbf{x}_i$, where $\mathbf{x}_i$ is a data sample represented in the original space and $\mathbf{y}_i$ is the projected data sample in the discriminant space.

In order to obtain the kernelized variant of the algorithm it is beneficial to consider the graph embedding-based formulation of SDA [11], [12]. Assuming that the data is centered at its mean, $\mathbf{S}_t = \mathbf{X}\mathbf{X}^T$, $\mathbf{S}_b = \mathbf{X}\mathbf{L}_b\mathbf{X}^T$, where $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N] \in \mathbb{R}^D$, and $\mathbf{L}_b$ is a Laplacian matrix defined as:

$$\mathbf{L}_b(i,j) = \begin{cases} \frac{N-N_{c_i}}{N^2 N_{ch}}, & \text{if } z_i = z_j = h, c_i = c_j \\ 0, & \text{if } z_i \neq z_j,\, c_i = c_j \\ -\frac{1}{N^2}, & \text{if } c_i \neq c_j \end{cases}, \quad (5)$$

where c_i is the class label of $\mathbf{x}_i$, and z_i is the subclass label of $\mathbf{x}_i$, N_c is the number of samples in class c and N_{ch} is the number of samples in subclass h of class c. The kernelized formulation [13] can then be defined as: $\mathbf{S}_{kt} = \mathbf{\Phi}\mathbf{\Phi}^T$, $\mathbf{S}_{kb} = \mathbf{\Phi}\mathbf{L}_b\mathbf{\Phi}^T$, where $\mathbf{\Phi} = [\phi(\mathbf{x}_1), \phi(\mathbf{x}_2), ..., \phi(\mathbf{x}_N)]$ is the data representation in the kernel space. In this case, it is assumed that data is centered in $\mathcal{F}$.

Thus, the solution to KSDA is given by the generalized eigendecomposition problem $\mathbf{L}_b\mathbf{K}\mathbf{a} = \lambda\mathbf{K}\mathbf{a}$, where $\mathbf{K} = \mathbf{\Phi}^T\mathbf{\Phi}$.

B. Fast Subclass Discriminant Analysis

In Fast Subclass Discriminant Analysis [6], the slow eigendecomposition step of SDA is substituted with a much faster process. This process is based on the creation of target vector matrix of random values with the same structure as the one of the eigenvectors of the between-class Laplacian matrix. The Laplacian matrix is a block matrix, hence, the structure of its eigenvectors can be inferred from the labels of the data. The target vector creation is followed by the orthogonalization of the resulting matrix. The algorithm can be described as follows:

1) Creation of the between-class Laplacian matrix (5)
2) Generation of target vector matrix $\mathbf{T}$ as in [6]
3) Regression of $\mathbf{X}$ to $\mathbf{T}$:

$$\mathbf{W} = (\mathbf{X}\mathbf{X}^T + \delta\mathbf{I})^{-1}\mathbf{X}\mathbf{T}^T, \quad (6)$$

where δ is the regularization parameter.

4) Orthogonalization of $\mathbf{W}$ such that $\mathbf{W}^T\mathbf{W} = \mathbf{I}$

Equivalently, for the kernel case, the steps 3-4 are the regression of $\mathbf{\Phi}$ to $\mathbf{T}$, i.e., $\mathbf{A} = (\mathbf{K}\mathbf{K}^T + \delta\mathbf{I})^{-1}\mathbf{K}\mathbf{T}^T$, and orthogonalization of $\mathbf{A}$ such that $\mathbf{A}^T\mathbf{K}\mathbf{A} = \mathbf{I}$. Besides, inversion based on Cholesky decomposition is applied for further speeding-up the process [14].

III. Weighted Fast Subclass Discriminant Analysis

LDA, SDA, and fastSDA assume that the data of each class contain the same amount of discriminative information and do not account for possible imbalance between the classes or the presence of outliers. In this section we propose two strategies to overcome these limitations based on re-weighting the contribution of samples to the solution.

In fastSDA, the solution is given by solving the regression problem in (6). In order to put more attention to certain classes during the learning of the projection matrix, each sample in $\mathbf{X}$ can be multiplied by the weight of a corresponding class. The re-weighting of samples is achieved by solving for $\mathbf{T} = \mathbf{\Omega}\mathbf{X}^T\mathbf{W}$. The solution is then given by

$$\mathbf{W} = (\mathbf{X}\mathbf{\Omega}\mathbf{X}^T + \delta\mathbf{I})^{-1}\mathbf{X}\mathbf{\Omega}\mathbf{T}^T, \quad (7)$$

where $\mathbf{\Omega}$ is an $N \times N$ diagonal weight matrix. Similarly, for kernelized fastSDA:

$$\mathbf{A} = (\mathbf{K}\mathbf{\Omega}\mathbf{K}^T + \delta\mathbf{I})^{-1}\mathbf{K}\mathbf{\Omega}\mathbf{T}^T. \quad (8)$$

The re-weighting of SDA can be achieved using the definition based on the Graph Embedding framework and weighting the total scatter as follows:

$$J(v) = \underset{\mathbf{W}}{argmax} \frac{Tr(\mathbf{W}^T \mathbf{X} L_b \mathbf{X}^T \mathbf{W})}{Tr(\mathbf{W}^T \mathbf{X} \mathbf{\Omega} \mathbf{X}^T \mathbf{W}}. \quad (9)$$

The solution is then given by the generalized eigendecomposition problem:

$$\mathbf{X} \mathbf{L}_b \mathbf{X}^T w = \lambda \mathbf{X} \mathbf{\Omega} \mathbf{X}^T w. \quad (10)$$

Similarly, in the kernelized formulation, the solution can be obtained as:

$$\mathbf{K} \mathbf{L}_b \mathbf{K}^T a = \lambda \mathbf{K} \mathbf{\Omega} \mathbf{K}^T a. \quad (11)$$

A. Prior weighted fastSDA for imbalanced classes

In real-world classification problems, the training data are often imbalanced, i.e., some classes have more samples than others. Such scenario biases the model towards learning the discriminative features of better represented class and can result in reduced performance on under-represented classes during inference. This limitation can be addressed by re-weighting the samples according to their number in the dataset. In other words, we would like to add more weight to the discriminative information present in under-represented classes to compensate for the low quantity of their samples.

The weighting strategy for imbalanced classes can be defined based on the inverse prior probability of classes. Thus, under-represented classes should obtain higher weights than over-represented ones. For the q^{th} sample in $\mathbf{X}$:

$$\mathbf{\Omega}_{cl}(q,q) = \frac{N}{N_i}, \quad (12)$$

where N and N_i are the total number of samples and number of samples in class i, which is the class of q^{th} sample, respectively.

Given the subclass information of samples, such a weighting scheme can further be extended to subclass-based weights as follows:

$$\mathbf{\Omega}_{sub}(q,q) = \frac{N}{N_{ih}}, \quad (13)$$

where N and N_{ih} are the total number of samples and number of samples in subclass of q^{th} sample.

In certain cases it is beneficial to utilize the combined weighting of the samples by using both class and subclass priors:

$$\mathbf{\Omega}(q,q) = \gamma \frac{N}{N_i} + \beta \frac{N_i}{N_{ih}}, \quad (14)$$

where γ and β are the hyperparameters defining the balance between subclass and class information.

B. Relevance weighted fastSDA for outlier classes

In some scenarios, despite the fact that the classes are balanced in the number of samples, certain classes present more relevant information than others. For example, in the presence of an outlier class/subclass, i.e., the one lying far from other subclasses and classes, it is reasonable to assume that this class is less likely to be confused with others, while it affects the learning of a weight matrix more than the classes lying closer to each other. In turn, we would like to pay more attention to the classes lying closer to each other as those are more likely to be confused in the projection space. Thus, the strategy to overcome such situation would be to put less weight to the classes/subclasses that are likely to be outliers.

A weighting technique based on scaled pairwise class distances in LDA was proposed in [8] and we further extend it to our subclass-based problem. Thus we employ a weighting scheme based on the pairwise distances of classes and subclasses scaled by the number of samples in the corresponding class/subclass. Note that although here we rely on the Euclidean distance, any other distance metric can be used for calculation of $D^{cl}_{i,j}$ and $D^{sub}_{ih,jl}$ [8]. For class i and subclass h the class-based relevance r_i is defined as:

$$r_i = \sum_{j=1, j \neq i}^{C} \frac{1}{D^{cl}_{i,j}} \quad (15)$$

$$D^{cl}_{i,j} = \sqrt{(\mu_i - \mu_j)^T (\mu_i - \mu_j)}, \quad (16)$$

where μ_i is the mean of class i. Similarly, the relevance weight based on pairwise subclass distance can be defined as:

$$s_{ih} = \sum_{j=1, j \neq i}^{C} \sum_{l=1}^{z_j} \frac{1}{D^{sub}_{ih,jl}} \quad (17)$$

$$D^{sub}_{ih,jl} = \sqrt{(\mu_{ih} - \mu_{jl})^T (\mu_{ih} - \mu_{jl})}, \quad (18)$$

where μ_{ih} is the mean of h^{th} subclass of i^{th} class. The weight matrices can be then defined as follows:

$$\mathbf{\Omega}_{cl}(q,q) = r_i \frac{N_i}{N}, \quad (19)$$

$$\mathbf{\Omega}_{sub}(q,q) = s_{ih} \frac{N_{ih}}{N}. \quad (20)$$

Finally, the combination of relevance-based weighting and prior-based weighting can be utilized:

$$\mathbf{\Omega}_(q,q) = \gamma r_i \frac{N_i}{N} + \beta s_{ih} \frac{N_{ih}}{N}, \quad (21)$$

where γ and β are the hyperparameters.

IV. Experiments

We compare the performance of the proposed methods with that of related methods on 8 datasets of different domains, tasks, sizes, and feature representations. We compare the performance of fastSDA, SDA, relevance-weighted fastSDA and SDA, LDA and relevance-weighted LDA on several classification datasets. To assess the performance of the proposed methods on imbalanced datasets we artificially introduce imbalance to some datasets and compare the accuracy and training time of fastSDA, SDA, prior-weighted fastSDA and SDA, LDA and relevance-weighted LDA.

For the evaluation of relevance-weighted approach we consider 8 datasets of different tasks. The first dataset is the Cohn-Kanade dataset [15] that contains 245 facial images of different people with different facial expressions of the 7 classes. All images were flattened to obtain 1200-dimensional vectors. For the task of digit recognition, Semeion dataset

TABLE I

CLASSIFICATION RESULTS FOR IMBALANCED DATASETS: ACCURACY/NUMBER OF CLUSTERS PER CLASS/TIME IN SEC.

Methods	WEATHER		LSD		MONKS2		CALT7-WM		HWD-KAR		HWD-ZER		SEMEION		PIMA	
$\mathbf{fastSDA}$	**92.50/2**	**0.064**	62.33/3	0.003	62.50/2	0.001	35.85/2	0.002	**94.50/2**	**0.003**	80.10/3	0.002	80.14/1	0.003	55.93/2	0.001
$\mathbf{fastSDA_{cl}}$	90.42/1	0.086	**62.83/4**	**0.009**	66.11/2	0.001	36.53/3	0.003	**94.50/3**	**0.004**	79.80/3	0.005	80.14/4	0.006	53.74/5	0.001
$\mathbf{fastSDA_{sub}}$	90.42/1	0.086	62.67/4	0.010	64.86/2	0.001	35.92/2	0.003	94.19/1	0.006	81.10/3	0.005	80.01/4	0.007	**57.36/4**	**0.001**
$\mathbf{fastSDA_{clsub}}$	90.21/1	0.097	62.33/4	0.013	**67.08/2**	**0.001**	35.37/1	0.004	94.29/2	0.006	**81.20/3**	**0.006**	80.29/4	0.007	56.65/4	0.001
$\mathbf{SDA}$	90.83/1	1.020	59.01/1	0.076	65.97/2	0.002	36.73/1	0.019	93.20/1	0.027	79.01/2	0.030	77.29/1	0.037	53.79/3	0.002
$\mathbf{SDA_{cl}}$	92.29/1	1.123	60.67/1	0.074	63.75/2	0.002	**38.91/2**	**0.021**	93.41/1	0.031	78.80/1	0.027	76.71/1	0.037	52.91/4	0.002
$\mathbf{SDA_{sub}}$	92.29/1	1.135	60.67/1	0.075	62.64/2	0.002	35.58/2	0.021	93.41/1	0.029	78.80/1	0.026	76.71/1	0.038	52.20/5	0.002
$\mathbf{SDA_{clsub}}$	92.29/1	1.132	60.67/1	0.078	63.89/2	0.002	37.21/2	0.021	93.41/1	0.029	79.01/3	0.029	76.71/1	0.035	51.48/5	0.002
$\mathbf{LDA}$	56.25	1.123	36.51	0.095	60.56	0.010	19.32	0.025	41.81	0.029	29.91	0.029	30.43	0.031	52.20	0.017
$\mathbf{LDA_{rel}}$	37.29	1.165	61.33	0.105	55.97	0.014	37.28	0.027	93.31	0.035	78.20	0.035	**80.57**	**0.033**	52.20	0.018
Kernelized formulations																
$\mathbf{fastSDA}$	93.13/1	0.007	61.17/3	0.113	61.80/2	0.001	46.19/1	0.017	95.91/1	0.032	81.91/1	0.028	**93.29/1**	**0.014**	57.31/3	0.002
$\mathbf{fastSDA_{cl}}$	93.33/1	0.011	57.67/4	0.238	62.08/2	0.002	46.26/4	0.036	96.01/1	0.059	81.11/1	0.051	**93.29/1**	**0.022**	**62.61/5**	**0.002**
$\mathbf{fastSDA_{sub}}$	93.33/1	0.013	56.33/4	0.241	**62.51/2**	**0.001**	46.05/1	0.041	96.11/2	0.058	81.11/1	0.056	**93.29/1**	**0.023**	59.78/3	0.002
$\mathbf{fastSDA_{clsub}}$	93.33/1	0.04	55.83/4	0.246	61.81/3	0.001	**46.46/1**	**0.036**	96.21/1	0.062	81.41/1	0.066	**93.29/1**	**0.025**	61.21/3	0.003
$\mathbf{SDA}$	93.75/4	0.043	**63.50/2**	**1.197**	57.08/1	0.001	45.97/1	0.172	96.21/4	0.231	**83.61/3**	**0.234**	91.43/2	0.101	56.76/2	0.011
$\mathbf{SDA_{cl}}$	93.54/1	0.041	61.67/2	1.193	50.12/3	0.001	34.49/1	0.166	**96.31/1**	**0.236**	80.41/1	0.222	74.01/1	0.098	50.77/2	0.016
$\mathbf{SDA_{sub}}$	**93.96/1**	**0.044**	59.51/5	1.175	53.47/3	0.001	32.45/1	0.163	**96.31/1**	**0.236**	80.41/1	0.235	74.01/1	0.098	50.77/2	0.015
$\mathbf{SDA_{clsub}}$	**93.96/1**	**0.050**	58.67/4	1.185	52.36/3	0.001	34.97/1	0.153	**96.31/1**	**0.226**	81.11/1	0.241	74.86/1	0.099	50.77/2	0.013

TABLE II

CLASSIFICATION RESULTS OF RELEVANCE-WEIGHTED METHODS: ACCURACY/NUMBER OF CLUSTERS PER CLASS/TIME IN SEC.

Methods	WEATHER		LSD		MONKS2		CALT7-CE		CALT7-GI		HWD-PIX		COHN-KAN		MSDI	
$\mathbf{fastSDA}$	95.73/1	0.070	82.93/3	0.007	64.24/4	0.001	91.65/2	0.003	93.69/1	0.007	95.95/4	0.004	65.31/1	0.019	46.29/2	0.019
$\mathbf{fastSDA_{cl}}$	96.80/2	0.107	83.32/2	0.059	**65.46/4**	**0.003**	**92.19/1**	**0.026**	**94.78/1**	**0.037**	**96.01/4**	**0.074**	**66.12/1**	**0.028**	46.45/1	0.219
$\mathbf{fastSDA_{sub}}$	**96.98/1**	**0.081**	83.38/3	0.029	**65.46/5**	**0.002**	91.86/2	0.018	94.71/1	0.026	95.20/4	0.039	64.08/1	0.026	46.79/1	0.142
$\mathbf{fastSDA_{clsub}}$	96.80/1	0.086	**83.41/3**	**0.064**	**65.46/5**	**0.006**	91.59/2	0.028	94.65/2	0.044	95.45/4	0.074	64.89/1	0.028	**46.89/1**	**0.221**
$\mathbf{SDA}$	95.20/1	1.058	80.97/1	0.148	62.42/5	0.001	91.52/1	0.047	93.08/1	0.091	94.01/1	0.077	60.40/1	0.170	44.52/1	0.778
$\mathbf{SDA_{cl}}$	96.62/1	1.045	81.69/1	0.143	63.03/5	0.003	91.45/1	0.064	90.02/1	0.109	94.00/1	0.093	64.08/1	0.169	44.66/1	0.863
$\mathbf{SDA_{sub}}$	96.62/1	1.054	81.69/1	0.188	63.66/5	0.002	91.45/1	0.051	90.02/1	0.097	94.01/1	0.087	64.08/1	0.255	44.66/1	0.784
$\mathbf{SDA_{clsub}}$	96.80/1	1.063	81.69/1	0.190	63.66/5	0.004	91.45/1	0.063	90.02/1	0.104	94.00/1	0.098	62.86/1	0.172	44.66/1	0.877
$\mathbf{LDA}$	71.56	1.525	53.35	0.131	57.58	0.003	68.19	0.073	45.05	0.164	41.65	0.096	38.78	0.191	28.62	0.695
$\mathbf{LDA_{rel}}$	38.49	1.581	82.23	0.146	55.15	0.009	92.16	0.079	**94.77**	**0.171**	94.95	0.101	45.71	0.194	45.943	0.759
Kernelized formulations																
$\mathbf{fastSDA}$	96.89/1	0.011	81.89/1	0.339	64.24/4	0.001	90.84/1	0.023	94.78/2	0.023	98.45/5	0.048	56.74/1	0.002	44.55/3	1.734
$\mathbf{fastSDA_{cl}}$	**96.98/1**	**0.031**	76.64/4	0.827	**65.46/3**	**0.004**	**93.68/1**	**0.061**	95.53/1	0.062	98.40/1	0.125	56.74/1	0.009	47.83/1	4.433
$\mathbf{fastSDA_{sub}}$	96.89/1	0.022	79.78/4	0.775	63.64/3	0.003	93.55/1	0.051	95.73/1	0.051	98.45/5	0.102	**57.14/1**	**0.008**	47.81/1	4.345
$\mathbf{fastSDA_{clsub}}$	96.89/1	0.031	81.87/4	0.884	64.85/5	0.004	92.27/1	0.064	96.27/1	0.068	**98.60/5**	**0.186**	55.51/1	0.011	**48.22/1**	**4.446**
$\mathbf{SDA}$	97.16/1	0.083	83.16/3	3.839	63.03/5	0.002	92.81/1	0.199	96.34/3	0.198	98.35/5	0.421	56.74/2	0.002	47.68/1	22.83
$\mathbf{SDA_{cl}}$	96.89/1	0.093	83.16/1	3.909	64.85/1	0.004	92.54/1	0.219	**96.41/1**	**0.216**	98.25/1	0.461	55.11/1	0.011	47.87/5	22.74
$\mathbf{SDA_{sub}}$	96.89/1	0.085	**83.25/2**	**3.983**	64.85/1	0.002	92.54/1	0.211	**96.41/1**	**0.197**	98.25/1	0.443	55.11/1	0.008	47.48/2	22.71
$\mathbf{SDA_{clsub}}$	96.89/1	0.091	83.18/1	3.912	61.82/3	0.004	92.06/1	0.218	**96.41/1**	**0.218**	98.30/1	0.454	55.11/1	0.011	47.76/5	23.22

[16] is considered containing 1593 instances of handwritten digits produced by 80 persons, represented by 16×16 binarized images flattened to 256×1 vectors. Another dataset of handwritten digits is considered [17], [18] with 2000 instances of handwritten digits of 10 classes represented by 240-dimensional pixel averages in 2×3 windows. The MONKS2 dataset [19] describes certain physical properties of robots with the goal of prediction of one of the two robot types based on these properties and contains 169 samples. The Landsat Satellite Dataset [20] consists of multi-spectral values of pixels of satellite images of different types of soil of 5 classes. The dataset contains 4435 36-dimensional samples. The Million Song Dataset with Images (MSDI) [21] poses a music genre classification task for 15 different genres. We consider a subset of 7468 instances described with 200-dimensional audio spectrograms. The Weather [22] is a dataset of images of 4 types of weather conditions, resulting in 1125 samples. 2048-dimensional features extracted from the pre-last layer of ResNet-50 [23] pre-trained on ImageNet were utilized. Caltech-101 [24] is an image classification dataset, and a subset of 7 classes is considered as described in [18], resulting in 1474 instances. 254-dimensional CENTRIST features and 512-dimensional GIST features are considered.

For evaluation of the proposed prior-weighted approaches on the imbalanced class problems, we introduce artificial imbalance to some of the above-mentioned datasets with different imbalance ratios resulting in the following numbers of samples: Semeion dataset - {120, 56, 120, 56, 120, 56, 120, 56, 120, 56}, LSD dataset - {760, 80, 760, 80, 760, 80}, Weather dataset - {240, 96, 96, 280}; Handwritten digits dataset - {160, 80, 160, 80, 160, 80, 160, 80, 160, 80}; Caltech-7 dataset - {348, 638, 42, 27, 28, 51, 45}; Monks2 dataset - {84, 34}. The imbalance was introduced to the training data only, while keeping the test data balanced. Two different feature representations are considered for Handwritten digits dataset: 47-dimensional Zernike moments and 64-

dimensional Karhunen-Love coeficients. For Caltech-7 dataset 40-dimensional wavelet moments are considered. Besides, the Pima Indians Diabetes dataset [25] was considered, containing data on different medical properties of patients with the goal of prediction whether the patient has diabetes. The dataset contains 500 samples in class 1 and 67 samples in class 2.

For training, 50% of the data is used for training, 30% for validation, and 20% for testing. Validation set is used for hyperparameter tuning, and the regularization parameter used for regularization of singular matrices in fastSDA, SDA, fastKSDA, KSDA and their weighted variants is chosen from the set $\{10^{-2}, 10^{-1}, 1, 10, 100, 100\}$ and both β and γ are selected as 0.5 in weighted methods, i.e., subclass and class information is taken into account equally. Subclass labels are obtained using k-means clustering in the original space and are the same for all the subclass-based methods as well as the kernelized variants. In kernelized formulations, we use RBF kernel with σ set to the mean distance between the training vectors. Classification in the projection space is achieved with k-nearest neighbors classifier with k = 5. Data is normalized prior to training. Dimensionality of the projected space is determined by the rank of the between-class scatter matrix and is set to $Cz - 1$ for all subclass-based methods, and $C - 1$ for LDA, where C is the number of classes and z is the number of subclasses per class. The number of subclasses from 1-5 were evaluated and the best result is reported.

The results for prior-weighted methods on imbalanced datasets and for relevance-weighted methods are shown in Tab. 1 and Tab. 2, respectively. We report the accuracy along with the number of clusters per class and the training time in seconds. Here $fastSDA$ and SDA refer to the original unweighted methods, $fastSDA_{cl}$ and SDA_{cl}, $fastSDA_{sub}$ and SDA_{sub}, and $fastSDA_{clusb}$ and SDA_{clsub} refer to the weighted methods with weights based on class information as in (12) and (19), subclass information as in (13) and (20), and both class and subclass information as in (14) and (21), respectively.

As can be observed, the best accuracy is generally obtained by weighted fastSDA or weighted SDA in both linear and kernel formulations. Besides, almost in all the cases the weighted version of fastSDA result in better accuracy than the original fastSDA while still giving a speed improvement compared to SDA. Thus, the use of re-weighting schemes can potentially improve the accuracy at almost no cost in terms of training time.

V. Conclusion

In this paper we proposed weighting schemes for Subclass Discriminant Analysis and Fast Subclass Discriminant Analysis for improving the robustness of the algorithms in the setting where there are potential outlier classes or the number of samples in each class is imbalanced. The results of extensive experiments on 9 datasets show that the proposed extensions result in improved accuracy while preserving the fast speed of fastSDA.

References

[1] J. Ye, "Least squares linear discriminant analysis," International Conference on Machine Learning, pp. 1087—1093, 2007.

[2] A. Iosifidis, A. Tefas, and I. Pitas, "Kernel reference discriminant analysis," Pattern Recognition Letters, pp. 85—91, 2014.

[3] A. Iosifidis, A. Tefas, and I. Pitas, "On the optimal class representation in linear discriminant analysis," IEEE Transactions on Neural Networks and Learning Systems, vol. 9, pp. 1491—1497, 2013.

[4] R. Duda, P. Hart, and D. Stork, Pattern Classification, 2nd Edition, Wiley, 2000.

[5] M. Zhu and A. Martinez, "Subclass discriminant analysis," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 28, 2006.

[6] K. Chumachenko, J. Raitoharju, A. Iosifidis, and M. Gabbouj, "Speed-up and multi-view extensions to subclass discriminant analysis," arXiv preprint arXiv:1905.00794, 2019.

[7] K. Chumachenko, J. Raitoharju, M. Gabbouj, A. Iosifidis, "Incremental fast subclass discriminant analysis," arXiv preprint arXiv:2002.04348, 2020.

[8] E. Tang, P. Suganthan, X. Yao, and A. Qin, "Linear dimensionality reduction using relevance weighted LDA," Pattern Recognition, vol. 8.4, pp. 485–493, 2005.

[9] L. Xu, A. Iosifidis, and M. Gabbouj, "Weighted linear discriminant analysis based on class saliency information," IEEE International Conference on Image Processing, pp. 2306–2310, 2018.

[10] R. Fisher, "The statistical utilization of multiple measurements," Annals of Eugenics, vol. 8, pp. 376-–386, 1938.

[11] A. Maronidis, A. Tefas, I. Pitas, "Subclass graph embedding and a marginal fisher analysis paradigm", Pattern Recognition, vol. 48, pp. 4024—4035, 2015.

[12] Y. Shuicheng, X. Dong, B. Zhang, H. Zhang, Q. Yang, and S. Lin, "Graph embedding and extensions: a general framework for dimensionality reduction," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 29, pp. 40—51, 2007.

[13] B. Chen, L. Yuan, H. Liu, and Z. Bao, "Kernel subclass discriminant analysis," Neurocomputing, vol. 71, pp. 455—458, 2007.

[14] A. Ruhe, Matrix algorithms volume 1: Basic decompositions, 2000.

[15] T. Kanade, J. Cohn, and Y. Tian, "Comprehensive database for facial expression analysis," IEEE International Conference on Automatic Face and Gesture Recognition, pp. 46—53, 2000.

[16] M. Buscema, "Metanet: the theory of independent judges," Substance Use & Misuse, vol. 33, pp. 439—461, 1998.

[17] M. van Breukelen, R. Duian, D. Tax, and J. den Hartog, "Handwritten digit recognition by combined classifiers," Kybernetika, vol. 34, pp. 381—386, 1998.

[18] F. Li, F. Nie, H. Huang, and J. Huang, "Large-scale multi-view spectral clustering via bipartite graph," AAAI Conference on Artificial Intelligence, pp. 2750—2756, 2015.

[19] J. Wnek and R. Michalski, "Comparing symbolic and subsymbolic learning: three studies," Machine Learning: A Multistrategy Approach vol. 4, 1993.

[20] A. Srinivasan, "Statlog (Landsat Satellite) Data Set", 1993, UCI Machine Learning Repository.

[21] S. Oramas, F. Barbieri, and O. Nieto, "Multimodal deep learning for music genre classification," Transactions of the International Society for Music Information Retrieval, vol. 1, pp. 4—21, 2018.

[22] A. Gbeminiyi, "Multi-class weather dataset for image classification, " Mendeley Data, 2018.

[23] K. He, X. Zhang, S. Ren, and J. Sun. "Deep residual learning for image recognition," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 770—778, 2016.

[24] L. Fei-Fei, R. Fergus, and P. Perona, "One-shot learning of object categories," IEEE Transactions on Pattern Recognition and Machine Intelligence, vol. 8, pp. 594-–611, 2006.

[25] J. Smith, J. Everhart, W. Dickson, W. Knowler, and R. Johannes, "Using the ADAP learning algorithm to forecast the onset of diabetes mellitus," Proceedings of the Symposium on Computer Applications and Medical Care, pp. 261-–265, 1988.