Subclass Graph Embedding and a Marginal Fisher Analysis Paradigm

A. Maronidis[a,*], A. Tefas[a], I. Pitas[a]

[a]*Department of Informatics, Aristotle University of Thessaloniki, P.O.Box 451, 54124 Thessaloniki, Greece*

Abstract

Subspace learning techniques have been extensively used for dimensionality reduction (DR) in many pattern classification problem domains. Recently, methods like Subclass Discriminant Analysis (SDA) and Clustering-based Discriminant Analysis (CDA), which use subclass information for the discrimination between the data classes, have attracted much attention. In parallel, important work has been accomplished on Graph Embedding (GE), which is a general framework unifying several subspace learning techniques. In this paper, GE has been extended in order to integrate subclass discriminant information resulting to the novel Subclass Graph Embedding (SGE) framework, which is the main contribution of our work. It is proven that SGE encapsulates a diversity of both supervised and unsupervised unimodal methods like Locality Preserving Projections (LPP), Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA). The theoretical link of SDA and CDA methods with SGE is also established. Along these lines, it is shown that SGE comprises a generalization of the typical GE framework including subclass DR methods. Moreover, it allows for an easy utilization of kernels for confronting non-linearly separable data. Employing SGE, in this paper a novel DR algorithm, which uses subclass discriminant information, called Subclass Marginal Fisher Analysis (SMFA) has been proposed. Through a series of experiments on various real-world datasets, it is shown that SMFA outperforms in most of the cases the state-of-the-art demonstrating the efficacy and power of SGE as a platform to develop new methods.

Keywords: Dimensionality Reduction, Subspace Learning, Graph Embedding, Subclass Structure

1. Introduction

Dimensionality reduction (DR) is an important process for achieving efficient pattern classification. In recent years, a variety of subspace learning algorithms for DR has been developed. Locality Preserving Projections (LPP) [1, 2] and Principal Component Analysis (PCA) [3] are two of the most popular unsupervised linear DR algorithms with a wide range of applications. Besides, supervised methods like Linear Discriminant Analysis (LDA) [4] have shown superior performance in many classification problems, since through the DR process

*Corresponding author (Tel.: 00306970081310)

Email addresses: amaronidis@iti.gr (A. Maronidis), tefas@aiia.csd.auth.gr (A. Tefas), pitas@aiia.csd.auth.gr (I. Pitas)

they aim at achieving data class discrimination. Usually though, there is a case where many data clusters appear inside the same class imposing the need to integrate this information in the DR approach. Along these lines, techniques such as Clustering Discriminant Analysis (CDA) [5] and Subclass Discriminant Analysis (SDA) [6] have been proposed. Both of them utilize a specific objective criterion that incorporates the data subclass information in an attempt to discriminate subclasses that belong to different classes, while they put no constraints to subclasses within the same class.

In parallel to the development of subspace learning techniques, a lot of work has been carried out in the graph theoretic approach to DR. Towards this direction, Graph Embedding (GE) has been built as a generalized framework, which unifies several existing DR methods and furthermore allows for developing novel algorithms. In [2, 7] the connection of LPP, PCA and LDA with the GE framework has been illustrated and in [7], employing GE, the authors propose Marginal Fisher Analysis (MFA). In addition, the ISOMAP [8], Locally Linear Embedding (LLE) [9] and Laplacian Eigenmaps (LE) [10] algorithms have also been interpreted within the GE framework [7].

Despite the intense activity around GE, no extension of GE has been proposed, in order to integrate subclass information. In this paper, this extension is proposed, leading to the novel Subclass Graph Embedding (SGE) framework, which is the main contribution of our work. It is also shown that a variety of unimodal DR algorithms are encapsulated within SGE. Particularly, it is proven that SGE constitutes a generalized framework that includes the classical GE. Finally, the kernelization of SGE is also presented. SGE attempts to optimize an intrinsic and a penalty criterion by preserving the subclass structure and simultaneously the local data geometry. This results to the corresponding intrinsic and penalty graph matrices both having a subclass block form. The local geometry may be modeled by any similarity or distance measure, while subclass structure may be extracted by any clustering algorithm. Choosing the appropriate subclass number per class or the similarity measure and its parameters, SGE becomes one of the well-known aforementioned algorithms.

In this paper, based on the SGE framework, a novel Subclass Marginal Fisher Analysis (SMFA) algorithm for supervised dimensionality reduction has been proposed. The new method exploits subclass information of the data and models the margins among classes using neighbourhood information among the samples. This combination provides SMFA with the ability to overcome the shortcomings stemming from the distribution constraints of the data leading to improved classification performance. Through an experimental comparison, it is shown that our method outperforms a number of state-of-the-art dimensionality reduction methods in terms of classification accuracy. The superiority of SMFA proves that SGE constitutes a platform for developing novel powerful DR methods.

The remainder of this paper is organized as follows. A literature review of related work is presented in Section 2. The well-known subspace learning algorithms LPP, PCA, LDA, CDA and SDA are presented in Section 3 in order to pave the way for their connection with SGE. The GE framework is described in Section 4, while the novel SGE framework along with its kernelization is presented in Section 5. The connection between

the SGE framework and the several subspace learning techniques, along with the details and the meaning of this connection is given in Section 6. The proposed SMFA method, based on SGE, is presented in Section 7. A comparison of the aforementioned methods on real-world datasets is presented in Section 8. Finally, conclusions are drawn in Section 9. Detailed derivation of the proposed approach is given in Appendices A and B.

2. Related Work

Although LDA proves to be an efficient method in many classification problems, it encounters some fundamental limitations. For instance, it suffers from the *small sample size* problem, which occurs when the number of the training samples is smaller than the data dimensionality. In this case, LDA fails to optimize its objective criterion, due to the singularity of the involved matrices. A solution to this problem has been provided in [11], where the authors propose the use of the pseudo-inverse of a matrix, in order to overcome matrix singularity. Another approach is the utilization of PCA as a preprocessing step to reduce data dimensionality and then, the application of LDA, resulting to the combined PCA + LDA method [4]. As has been clearly stated in [12], an additional problem appears, when some of the smallest eigenvalues of the within matrix correspond to noisy features of the data. A factorization that prunes the noisy bases of the within matrix and a correlation-based criterion have been proposed in [12] for solving these problems. In an indirect way to deal with the singularity problem, another method (2D-LDA), where the data are represented as matrices has been proposed in [13]. For overcoming the *small sample size* problem, regularization techniques have been also employed [14, 15].

Another strong limitation is that LDA postulates that the data class samples have multivariate Gaussian distribution, common covariance matrix and different means, for achieving the optimal discrimination in Bayesian terms [16]. In real problems though, the class data might not be normally distributed. Many extensions of LDA have been proposed in the literature for circumventing these limitations [17, 18, 19, 20]. Amongst the most effective methods towards this end is Marginal Fisher Analysis [7] designed based on the Graph Embedding framework. MFA uses adjacency information among the data samples and achieves to overcome the above-mentioned distribution limitations.

As already mentioned in the Introduction, CDA and SDA have been proposed for exploiting the potential subclass structure of the data. Along the same lines, a Mixture Subclass Discriminant Analysis (MSDA) method that modifies the objective function of SDA has been proposed in [21]. Moreover, the link between MSDA and the Gaussian mixture model has been accomplished using the Expectation-Maximization framework. In the same work, MSDA has further been extended in several ways so that the subclass separation problem is solved and nonlinearly separable subclass structure has been tackled using the kernel trick. In [22], a Multiple-Exemplar Discriminant Analysis (MEDA) method is presented. The classes are represented by some exemplar vectors. Using these exemplars, an objective criterion is constructed. In this vein, the subclass means can be used as exemplars, hence exploiting the subclass structure of the data.

Subspace learning and clustering have been treated together into an iterative process in [23]. Intra-cluster similarity and inter-cluster separability are enhanced using initial cluster estimation in the subspace-learning step. Then, affinity propagation is adopted for clustering the reduced data providing an updated clustering estimation. In [24], the authors combine global with local geometric structures using a regularization technique. The singularity problem is tackled by imposing penalty on parameters and the optimal parameter is chosen based on a model selection approach.

For conducting nonlinear DR, the application of the kernel trick to the linear approaches has been proposed [25]. The main idea is to firstly map the data from the initial space to a high-dimensional Hilbert space, where they might be linearly separable and then use a linear subspace method. This approach results to the kernelized versions of the linear techniques, that have already been developed: Kernel Principal Component Analysis (KPCA) [26], Kernel Discriminant Analysis (KDA) [27], Kernel Clustering Discriminant Analysis (KCDA) [28], Kernel Subclass Discriminant Analysis (KSDA) [29], etc.

A graph-based supervised DR method has been proposed in [30] for circumventing the problem of non-Gaussian distributed data. The importance degrees of the same-class and not-same-class vertices are encoded by the intrinsic and extrinsic graphs, respectively, based on a strictly monotonically decreasing function. Moreover, the kernel extension of the proposed approach is also presented. In [31], instead of predefining the neighbor parameters of the intrinsic and extrinsic graph matrices, the neighbor parameter selection is adaptively performed based on the different local manifold structure of different samples, enhancing in this way the intra-class similarity and inter-class separability.

A methodology that converts a set of graphs into a vector space has been presented in [32]. A novel prototype selection method from a class-labeled set of graphs is proposed. A dissimilarity metric between a pair of graphs is established and the dissimilarities of a graph from a set of prototypes are calculated providing an n-dimensional feature vector. Several deterministic algorithms are used to select the prototypes with the most discriminative power [32]. The flexibility of GE has been also combined with the generalization ability of the support vector machine classifier resulting to improved classification performance. In [33], the authors propose the substitution of the support vector machine kernel with sub-space or sub-manifold kernels, that are constructed based on the GE framework.

From the above review, it looks as though the several limitations stemming from the data distributions or the singularity of the involved matrices have been successfully addressed by dedicated methods. However, there is still enough space for improvement as the new methods introduce new limitations. For instance, subclass-based methods postulate that the data subclasses have Gaussian distributions, hence translating the problem from classes to subclasses. Moreover, although some of the above-mentioned techniques manage to deal with such limitations and optimally model the distributions of the training data, the generalization ability to the test data still remains an open challenge.

3. Subspace Learning Techniques

In this section, we provide some useful notation along with the mathematical formulation of the subspace learning techniques LPP, PCA, LDA, CDA and SDA, in order to allow their connection with the SGE framework. A brief description of these methods has been already given in the introduction. In the following analysis, we consider that each data sample denoted by $\mathbf{x}$ is an m-dimensional real vector, i.e., $\mathbf{x} \in \mathbb{R}^m$. We also denote by $\mathbf{y} \in \mathbb{R}^{m'}$ its projection $\mathbf{y} = \mathbf{V}^T\mathbf{x}$ to a new m'-dimensional space using a projection matrix $\mathbf{V} \in \mathbb{R}^{m \times m'}$.

In an attempt to preserve the local data geometry after the projection to the space of reduced dimensionality, LPP solves the following minimization problem:

$$\min tr\{\sum_{qp} \left(\mathbf{y}_q - \mathbf{y}_p\right) S_{qp} \left(\mathbf{y}_q - \mathbf{y}_p\right)^T\}, \tag{1}$$

where $tr\{\cdot\}$ denotes the matrix trace, $\mathbf{y}_q$ is the projection of $\mathbf{x}_q$ and S_{qp} is a value that expresses the similarity between $\mathbf{x}_q$ and $\mathbf{x}_p$. The values S_{qp}, for every pair of vectors (q, p), construct the *affinity matrix* $\mathbf{S}$.

There are many possible ways for defining the affinity matrix [34]. One way is to use the *Gaussian similarity function* defined as:

$$S_{qp} = S(\mathbf{x}_q, \mathbf{x}_p) = \exp\left(-\frac{d^2(\mathbf{x}_q, \mathbf{x}_p)}{\sigma^2}\right), \tag{2}$$

where $d(\mathbf{x}_q, \mathbf{x}_p)$ is a distance metric (e.g., Euclidean) and σ^2 is a parameter (variance) that determines the distance scale.

PCA is a statistical learning technique that seeks the directions, where the projected data scatter is maximized [35, 36]. Strictly speaking, it attempts to project the vector $\mathbf{x}$ to the values $y_i = \mathbf{v}_i^T\mathbf{x}$, looking for those projection vectors $\mathbf{v}_i$ that maximize the scatter of y_i. This maximization problem is resolved by performing eigenanalysis on the covariance matrix of the mean centered data $E[(\mathbf{x} - \boldsymbol{\mu})(\mathbf{x} - \boldsymbol{\mu})^T]$, where $\boldsymbol{\mu}$ is the mean vector of the data samples. The transformation matrix $\mathbf{V}$ consists of those orthonormal eigenvectors that correspond to the m' largest eigenvalues, where $m' \ll m$.

LDA, CDA and SDA attempt to minimize a Fisher-Rao's criterion [37]:

$$J(\mathbf{v}) = \frac{\mathbf{v}^T\mathbf{S}_W\mathbf{v}}{\mathbf{v}^T\mathbf{S}_B\mathbf{v}}, \tag{3}$$

where $\mathbf{S}_W$ is called the *within* and $\mathbf{S}_B$ the *between* scatter matrix. These matrices are symmetric and positive semi-definite. The minimization of the ratio (3) leads to the following generalized eigenvalue decomposition problem to find the optimal discriminant projection eigenvectors:

$$\mathbf{S}_W\mathbf{v} = \lambda\mathbf{S}_B\mathbf{v}. \tag{4}$$

The eigenvalues λ_i of the above eigenproblem are by definition positive or zero:

$$0 \leq \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_m \,. \tag{5}$$

Let $\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_m$ be the corresponding eigenvectors. Then the projection $\mathbf{y} = \mathbf{V}^T\mathbf{x}$, from the initial space to the new space of reduced dimensionality employs the projection matrix $\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_{m'}]$ whose columns are the eigenvectors $\mathbf{v}_i$, $i = 1, \ldots, m'$ and $m' \ll m$.

LDA seeks for a low-dimensional space, such that when data vectors $\mathbf{x}$ are projected, their classes are well discriminated [4]. Let us denote the total number of classes by c, the mean vector of the i-th class by $\boldsymbol{\mu}^i$, the mean vector of the whole data set by $\boldsymbol{\mu}$, the number of samples belonging to the i-th class by n_i and the q-th vector of the i-th class by $\mathbf{x}_q^i$. The objective of LDA is to find the projection vectors $\mathbf{v}$ that minimize the Fisher ratio (3), where $\mathbf{S}_W^{(LDA)}$ is the *within-class* and $\mathbf{S}_B^{(LDA)}$ the *between-class* scatter matrix defined as [4]:

$$\mathbf{S}_W^{(LDA)} = \sum_{i=1}^{c} \sum_{q=1}^{n_i} \left(\mathbf{x}_q^i - \boldsymbol{\mu}^i\right)\left(\mathbf{x}_q^i - \boldsymbol{\mu}^i\right)^T \,, \tag{6}$$

$$\mathbf{S}_B^{(LDA)} = \sum_{i=1}^{c} \left(\boldsymbol{\mu}^i - \boldsymbol{\mu}\right)\left(\boldsymbol{\mu}^i - \boldsymbol{\mu}\right)^T \,. \tag{7}$$

CDA, like LDA, looks for a linear transform that effectively separates the projected data of each class. Its difference with LDA is that CDA makes use of the potential subclass structure. Let us denote the total number of subclasses inside the i-th class by d_i and, for the j-th subclass of the i-th class, the number of its samples by n_{ij}, its q-th sample by $\mathbf{x}_q^{ij}$ and its mean vector by $\boldsymbol{\mu}^{ij}$. CDA attempts to minimize (3), where $\mathbf{S}_W^{(CDA)}$ is the *within-subclass* and $\mathbf{S}_B^{(CDA)}$ the *between-subclass* scatter matrix, defined as [5]:

$$\mathbf{S}_W^{(CDA)} = \sum_{i=1}^{c} \sum_{j=1}^{d_i} \sum_{q=1}^{n_{ij}} \left(\mathbf{x}_q^{ij} - \boldsymbol{\mu}^{ij}\right)\left(\mathbf{x}_q^{ij} - \boldsymbol{\mu}^{ij}\right)^T \,, \tag{8}$$

$$\mathbf{S}_B^{(CDA)} = \sum_{i=1}^{c-1} \sum_{l=i+1}^{c} \sum_{j=1}^{d_i} \sum_{h=1}^{d_l} \left(\boldsymbol{\mu}^{ij} - \boldsymbol{\mu}^{lh}\right)\left(\boldsymbol{\mu}^{ij} - \boldsymbol{\mu}^{lh}\right)^T \,. \tag{9}$$

The difference between SDA and CDA mainly lies on the definition of the within scatter matrix, while the between scatter matrix of SDA is a modified version of that of CDA. The exact definitions of the two matrices are:

$$\mathbf{S}_W^{(SDA)} = \sum_{q=1}^{n} \left(\mathbf{x}_q - \boldsymbol{\mu}\right)\left(\mathbf{x}_q - \boldsymbol{\mu}\right)^T \,, \tag{10}$$

$$\mathbf{S}_B^{(SDA)} = \sum_{i=1}^{c-1} \sum_{l=i+1}^{c} \sum_{j=1}^{d_i} \sum_{h=1}^{d_l} p_{ij} p_{lh} \left(\boldsymbol{\mu}^{ij} - \boldsymbol{\mu}^{lh}\right)\left(\boldsymbol{\mu}^{ij} - \boldsymbol{\mu}^{lh}\right)^T \,, \tag{11}$$

where $p_{ij} = \frac{n_{ij}}{n}$ is the relative frequency of the j-th cluster of the i-th class [6]. It is worth mentioning that $\mathbf{S}_W^{(SDA)}$ is actually the total covariance matrix of the data.

The previously described DR methods can be seen under a common prism, since their basic calculation element towards the construction of the corresponding optimization criteria is the similaritiy among the samples. Thus we can unify them in a common framework if we consider that the samples form a graph and we set criteria on the similarities between the nodes of this graph. In the following section we describe in detail this approach.

4. Graph Embedding

In this section, the problem of dimensionality reduction is described from a graph theoretic perspective. Before we proceed further into the analysis of this approach, let us briefly give some basic graph notation which will be used subsequently.

4.1. Graph Notation

A graph $G = \{\mathcal{X}, \mathcal{E}\}$ consists of a vertex set $\mathcal{X}$ and an edge set $\mathcal{E}$ such that $\mathcal{E} \subseteq \mathcal{X}^2$. A graph is called weighted, when the above edges are characterized by some weights, constituting the weight-matrix $\mathbf{W}$. This matrix provides a good graph description of the form $G = \{\mathcal{X}, \mathbf{W}\}$. A non-zero value in $\mathbf{W}$ indicates that there is a connection between the corresponding vertices. Also, a zero value in $\mathbf{W}$ indicates that there is no connection between the corresponding vertices. A graph is called undirected, when the weight-matrix $\mathbf{W}$ is symmetric. Otherwise, it is called directed [38].

In our analysis, we shall use only undirected graphs. The degree matrix is defined as the diagonal matrix $\mathbf{D}$ which has at position (q, q) the value $D_{qq} = \sum_p W_{qp}$. This value is the summation of all values of $\mathbf{W}$ across the q-th row or column, since $\mathbf{W}$ is symmetric. In fact, the sum is calculated over the adjacent vertices to vertex $\mathbf{x}_q$, as, for all the other vertices, the weights are zero: $W_{qp} = 0$. Therefore, the elements of $\mathbf{D}$ give an indication of the adjacency of the q-th vertex to the rest of the vertices. Finally, the Laplacian matrix $\mathbf{L}$ is defined as $\mathbf{L} = \mathbf{D} - \mathbf{W}$ [7].

4.2. Graph Embedding

In the GE framework, the set of the data samples to be projected in a low dimensionality space is represented by two graphs, namely, the *intrinsic* $G_{int} = \{\mathcal{X}, \mathbf{W}_{int}\}$ and the *penalty* $G_{pen} = \{\mathcal{X}, \mathbf{W}_{pen}\}$ graph, where $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n\}$ is the set of the data samples in both graphs. The intrinsic graph models the similarity connections between every pair of data samples that have to be reinforced after the projection. The penalty graph contains the connections between the data samples that must be suppressed after the projection. For both of the above matrices these connections might have negative values. A negative value causes the opposite results, i.e., a negative value in the intrinsic matrix means that the corresponding data samples should diverge and a

negative value in the penalty matrix means that the corresponding data samples should converge after the projection. Choosing the values of both the intrinsic and the penalty graph matrices, may lead in either supervised, unsupervised or semi-supervised DR algorithms. Now, the problem of DR could be interpreted in another way. It is desirable to project the initial data to the new low dimensional space, such that the geometrical structure of the data is preserved. The corresponding objective function for optimization is:

$$\underset{tr\{\mathbf{YBY}^T\}=d}{\operatorname{argmin}} \; J(\mathbf{Y}), \tag{12}$$

$$J(\mathbf{Y}) = \frac{1}{2} tr\{\sum_q \sum_p (\mathbf{y}_q - \mathbf{y}_p)\mathbf{W}_{int}(q,p)(\mathbf{y}_q - \mathbf{y}_p)^T\}, \tag{13}$$

where $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_n]$ are the projected vectors, d is a constant, $\mathbf{B}$ is a constraint matrix, defined to remove an arbitrary scaling factor in the embedding and $\mathbf{W}_{int}(q,p)$ is the value of $\mathbf{W}_{int}$ at position (q,p). The structure of the objective function (13) postulates that, the larger the value $\mathbf{W}_{int}(q,p)$ is, the smaller the distance between the projections of the data samples $\mathbf{x}_q$ and $\mathbf{x}_p$ has to be. By using some simple algebraic manipulations, equation (13) becomes:

$$J(\mathbf{Y}) = tr\{\mathbf{YL}_{int}\mathbf{Y}^T\}, \tag{14}$$

where $\mathbf{L}_{int} = \mathbf{D}_{int} - \mathbf{W}_{int}$ is the intrinsic Laplacian matrix. The Laplacian matrix $\mathbf{L}_{pen} = \mathbf{D}_{pen} - \mathbf{W}_{pen}$ of the penalty graph is often used as the constraint matrix $\mathbf{B}$. Thus, the above optimization problem becomes:

$$\operatorname{argmin} \frac{tr\{\mathbf{YL}_{int}\mathbf{Y}^T\}}{tr\{\mathbf{YL}_{pen}\mathbf{Y}^T\}}. \tag{15}$$

The optimization of the above objective function is achieved by solving the generalized eigenproblem:

$$\mathbf{L}_{int}\mathbf{v} = \lambda \mathbf{L}_{pen}\mathbf{v}, \tag{16}$$

keeping the eigenvectors, which correspond to the smallest eigenvalues.

This approach leads to the optimal projection of the given data samples. In order to achieve the out of sample projection, the linearization [7] of the above approach should be used. If we employ $\mathbf{y} = \mathbf{V}^T\mathbf{x}$, the objective function (13) becomes:

$$\underset{tr\{\mathbf{V}^T\mathbf{XL}_{pen}\mathbf{X}^T\mathbf{V}\}=d}{\operatorname{argmin}} \; J(\mathbf{V}), \tag{17}$$

$$J(\mathbf{V}) = \frac{1}{2} tr\{\mathbf{V}^T \left(\sum_q \sum_p (\mathbf{x}_q - \mathbf{x}_p)\mathbf{W}_{int}(q,p)(\mathbf{x}_q - \mathbf{x}_p)^T \right) \mathbf{V}\}, \tag{18}$$

where $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n]$. By using simple algebraic manipulations, we have:

$$J(\mathbf{V}) = tr\{\mathbf{V}^T \mathbf{X} \mathbf{L}_{int} \mathbf{X}^T \mathbf{V}\}. \tag{19}$$

Similarly to the straight approach, the optimal eigenvectors are given by solving the generalized eigenproblem:

$$\mathbf{X} \mathbf{L}_{int} \mathbf{X}^T \mathbf{v} = \lambda \mathbf{X} \mathbf{L}_{pen} \mathbf{X}^T \mathbf{v}. \tag{20}$$

It can be easily shown [7], that if the intrinsic matrix $\mathbf{W}_{int}$ takes the form: $\mathbf{W}_{int}(q, p) = \frac{1}{n}$, $\forall (q, p)$ and the penalty Laplacian matrix equals the identity matrix, $\mathbf{L}_{pen} = \mathbf{I}$, GE becomes identical to PCA. Now, let us denote the set of the samples that belong to the i-th class by C_i and define the intrinsic matrix as follows:

$$\mathbf{W}_{int}(q, p) = \begin{cases} \frac{1}{n_i} & \text{, if } \mathbf{x}_q, \mathbf{x}_p \in C_i \\ 0 & \text{, otherwise} \end{cases}. \tag{21}$$

Let us also choose the penalty matrix to be:

$$\mathbf{W}_{pen}(q, p) = \begin{cases} \frac{n_i - n}{n n_i} & \text{, if } \mathbf{x}_q, \mathbf{x}_p \in C_i \\ \frac{1}{n} & \text{, otherwise} \end{cases}. \tag{22}$$

Then, GE becomes identical to LDA [2].

5. Subclass Graph Embedding

5.1. Linear Subclass Graph Embedding

In this section we propose a GE extension in a way that allows the exploitation of subclass information. In the following analysis, it is assumed that the subclass labels are known. We attempt to minimize the scatter of the data samples within the same subclass, while separating data samples from subclasses that belong to different classes. Finally, we are not concerned about samples that belong to different subclasses of the same class.

Usually, in real-world problems, local geometry of the data is related to the global supervised structure. Samples that belong to the same class or subclass, should be "sufficiently close" to each other. SGE actually exploits this fact. It simultaneously handles supervised and unsupervised information. As a consequence, it combines the global labeling information with the local geometrical characteristics of the data samples. This is achieved by weighing the above connections with the similarities of the data samples. The *Gaussian similarity function*, defined in (2), has been used in this paper.

Let us denote as $\mathbf{P}$ an affinity matrix. Without limiting the generality, we assume that this matrix has block form, depending on the subclass and the class of the data samples. Using the linearized approach, we attempt to optimize a more general discrimination criterion. We consider again that $\mathbf{y} = \mathbf{V}^T \mathbf{x}$ is the projection of $\mathbf{x}$ to the

new subspace. $\mathbf{P}^{ij}(q,p)$ is the value of $\mathbf{P}$ at position (q,p) of the submatrix that contains the j-th subclass of the i-th class. Then, the proposed criterion is:

$$\min J(\mathbf{Y}), \tag{23}$$

$$J(\mathbf{Y}) = \frac{1}{2} tr\{\sum_{i=1}^{c}\sum_{j=1}^{d_i}\sum_{q=1}^{n_{ij}}\sum_{p=1}^{n_{ij}}\left(\mathbf{y}_q^{ij}-\mathbf{y}_p^{ij}\right)\mathbf{P}^{ij}(q,p)\left(\mathbf{y}_q^{ij}-\mathbf{y}_p^{ij}\right)^T\} \tag{24}$$

$$= \frac{1}{2} tr\{\mathbf{V}^T\left(\sum_{i=1}^{c}\sum_{j=1}^{d_i}\sum_{q=1}^{n_{ij}}\sum_{p=1}^{n_{ij}}\left(\mathbf{x}_q^{ij}-\mathbf{x}_p^{ij}\right)\mathbf{P}^{ij}(q,p)\left(\mathbf{x}_q^{ij}-\mathbf{x}_p^{ij}\right)^T\right)\mathbf{V}\} \tag{25}$$

$$= tr\{\mathbf{V}^T\mathbf{X}\left(\mathbf{D}_{int}-\mathbf{W}_{int}\right)\mathbf{X}^T\mathbf{V}\} \tag{26}$$

$$= tr\{\mathbf{V}^T\mathbf{X}\mathbf{L}_{int}\mathbf{X}^T\mathbf{V}\}. \tag{27}$$

The derivation of (27) can be found in Appendix A. The matrix $\mathbf{W}_{int}$ is block diagonal with blocks that correspond to each class and is given by:

$$\mathbf{W}_{int} = \begin{pmatrix} \mathbf{W}_{int}^1 & & & \\ & \mathbf{W}_{int}^2 & & 0 \\ 0 & & \ddots & \\ & & & \mathbf{W}_{int}^c \end{pmatrix}. \tag{28}$$

$\mathbf{W}_{int}^i$ are block diagonal submatrices, with blocks that correspond to the subclasses and are given by:

$$\mathbf{W}_{int}^i = \begin{pmatrix} \mathbf{P}^{i1} & & & \\ & \mathbf{P}^{i2} & & 0 \\ 0 & & \ddots & \\ & & & \mathbf{P}^{id_i} \end{pmatrix}. \tag{29}$$

$\mathbf{P}^{ij}$ is the submatrix of $\mathbf{P}$ that corresponds to the data of the j-th cluster of the i-th class. By looking carefully at the form of $\mathbf{W}_{int}$, it is clear that the degree intrinsic matrix $\mathbf{D}_{int}$ has values

$$\mathbf{D}_{int}(\sum_{s=0}^{i-1}\sum_{t=0}^{j-1} n_{st} + q, \sum_{s=0}^{i-1}\sum_{t=0}^{j-1} n_{st} + q) = \sum_p \mathbf{P}^{ij}(q,p), \tag{30}$$

where p runs over the indices of the j-th cluster of i-th class.

In parallel, we demand to maximize a criterion, which encodes the similarities among the centroid vectors of the subclasses. Let the value Q_{ij}^{lh} express the similarity between the centroid vectors $\boldsymbol{\mu}^{ij}$ and $\boldsymbol{\mu}^{lh}$. The more

similar two centroids that belong to different classes are, the further apart their projections $\mathbf{m}^{ij} = \mathbf{V}^T \boldsymbol{\mu}^{ij}$ have to be from each other:

$$\max G(\mathbf{m}^{ij}), \tag{31}$$

$$G(\mathbf{m}^{ij}) = tr\{\sum_{i=1}^{c-1} \sum_{l=i+1}^{c} \sum_{j=1}^{d_i} \sum_{h=1}^{d_l} \left(\mathbf{m}^{ij} - \mathbf{m}^{lh}\right) Q_{ij}^{lh} \left(\mathbf{m}^{ij} - \mathbf{m}^{lh}\right)^T\} \tag{32}$$

$$= tr\{\mathbf{V}^T \left(\sum_{i=1}^{c-1} \sum_{l=i+1}^{c} \sum_{j=1}^{d_i} \sum_{h=1}^{d_l} \left(\boldsymbol{\mu}^{ij} - \boldsymbol{\mu}^{lh}\right) Q_{ij}^{lh} \left(\boldsymbol{\mu}^{ij} - \boldsymbol{\mu}^{lh}\right)^T\right) \mathbf{V}\} \tag{33}$$

$$= tr\{\mathbf{V}^T \mathbf{X} \left(\mathbf{D}_{pen} - \mathbf{W}_{pen}\right) \mathbf{X}^T \mathbf{V}\} \tag{34}$$

$$= tr\{\mathbf{V}^T \mathbf{X} \mathbf{L}_{pen} \mathbf{X}^T \mathbf{V}\}, \tag{35}$$

as derived in Appendix B. The block matrix $\mathbf{W}_{pen}$ in (34) consists of block submatrices:

$$\mathbf{W}_{pen} = \begin{pmatrix} \mathbf{W}_{pen}^{1,1} & \mathbf{W}_{pen}^{1,2} & \cdots & \mathbf{W}_{pen}^{1,c} \\ \mathbf{W}_{pen}^{2,1} & \mathbf{W}_{pen}^{2,2} & \cdots & \mathbf{W}_{pen}^{2,c} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{W}_{pen}^{c,1} & \mathbf{W}_{pen}^{c,2} & \cdots & \mathbf{W}_{pen}^{c,c} \end{pmatrix}. \tag{36}$$

It is obvious that $\mathbf{W}_{pen}$ consists of the submatrices $\mathbf{W}_{pen}^{i,j}$, where the ones on the main block diagonal are given by:

$$\mathbf{W}_{pen}^{i,i} = \begin{pmatrix} \mathbf{W}^{i1} & & & \\ & \mathbf{W}^{i2} & & 0 \\ 0 & & \ddots & \\ & & & \mathbf{W}^{id_i} \end{pmatrix}, \tag{37}$$

where $\mathbf{W}^{ij}$ corresponds to the j-th subclass of the i-th class and is given by:

$$\mathbf{W}^{ij} = -\frac{\left(\sum_{\omega \neq i} \left(\sum_{t=1}^{d_\omega} Q_{ij}^{\omega t}\right)\right)}{\left(n_{ij}\right)^2} \mathbf{e}^{n_{ij}} \left(\mathbf{e}^{n_{ij}}\right)^T, \tag{38}$$

where $\mathbf{e}^{n_{ij}} = [\overbrace{11\cdots1}^{n_{ij}\text{-times}}]^T$. Respectively, the off-diagonal submatrices of $\mathbf{W}_{pen}$ are given by:

$$\mathbf{W}_{pen}^{i,l} = \begin{pmatrix} \mathbf{W}_{i1}^{l1} & \mathbf{W}_{i1}^{l2} & \cdots & \mathbf{W}_{i1}^{ld_l} \\ \mathbf{W}_{i2}^{l1} & \mathbf{W}_{i2}^{l2} & \cdots & \mathbf{W}_{i2}^{ld_l} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{W}_{id_i}^{l1} & \mathbf{W}_{id_i}^{l2} & \cdots & \mathbf{W}_{id_i}^{ld_l} \end{pmatrix}, i \neq l, \tag{39}$$

where:

$$\mathbf{W}_{ij}^{lh} = \frac{Q_{ij}^{lh}}{n_{ij}n_{lh}} \mathbf{e}^{n_{ij}} (\mathbf{e}^{n_{lh}})^T . \tag{40}$$

It can be easily shown that $\mathbf{D} = \mathbf{0}$, so that $\mathbf{L}_{pen} = -\mathbf{W}_{pen}$.

5.2. Physical Meaning

By looking carefully at the intrinsic graph matrix $\mathbf{W}_{int}$ given in (28) and (29), it becomes clear that it has a block diagonal form, where each block corresponds to a different subclass. The values of these subclass-blocks are positive, imposing a "favorable" connection between the data samples within the same subclass. The zeros out of the block diagonal put no constraint to the corresponding data samples. Similar remarks should also be pointed regarding the penalty graph matrix, whose off diagonal blocks have positive values, corresponding to data samples which belong to different classes. These connections are unfavorable and have to be suppressed. The blocks in the main block diagonal, have also a block diagonal form given in (37). These blocks correspond to the several subclasses and have negative values. This fact induces a favorable connection between the data samples within the same subclass, since they lie in the penalty matrix. Finally, the zeros of $\mathbf{W}_{pen}$ indicating the data samples, which belong to the same class but to different subclasses, do not impose any constraint. Of course, the above discussion absolutely fits the motivations that lead to the utilized SGE criteria.

5.3. Kernel Subclass Graph Embedding

In this section, the kernelization of SGE is presented. Kernels are widely used in classification problems, where the data are not linearly separable and in unsupervised learning when the data lie on a nonlinear manifold. Let us denote by $\mathcal{X}$ the initial data space, by $\mathcal{F}$ a Hilbert space and by f the non-linear mapping function from $\mathcal{X}$ to $\mathcal{F}$. The main idea is to firstly map the original data from the initial space into another high-dimensional Hilbert space and then perform linear subspace analysis in that space. If we denote by $m_{\mathcal{F}}$ the dimensionality of the Hilbert space, then the above procedure is described as:

$$\mathcal{X} \ni \mathbf{x}_q \rightarrow \mathbf{y}_q = f(\mathbf{x}_q) = \begin{pmatrix} \sum_{p=1}^{n} a_{1p} k(\mathbf{x}_q, \mathbf{x}_p) \\ \vdots \\ \sum_{p=1}^{n} a_{m_{\mathcal{F}}p} k(\mathbf{x}_q, \mathbf{x}_p) \end{pmatrix} \in \mathcal{F}, \tag{41}$$

where k is the kernel function. From the above equation it is obvious that

$$\mathbf{Y} = \mathbf{A}^T \mathbf{K}\,, \tag{42}$$

where $\mathbf{K}$ is the Gram matrix, which has at position (q, p) the value $K_{qp} = k(\mathbf{x}_q, \mathbf{x}_p)$ and

$$\mathbf{A} = [\mathbf{a}_1 \cdots \mathbf{a}_{m_{\mathcal{F}}}] = \begin{pmatrix} a_{11} & \cdots & a_{m_{\mathcal{F}}1} \\ \vdots & \ddots & \vdots \\ a_{1n} & \cdots & a_{m_{\mathcal{F}}n} \end{pmatrix} \tag{43}$$

is the map coefficient matrix. Consequently, the final SGE optimization becomes:

$$\operatorname{argmin} \frac{tr\{\mathbf{A}^T \mathbf{K} \mathbf{L}_{int} \mathbf{K} \mathbf{A}\}}{tr\{\mathbf{A}^T \mathbf{K} \mathbf{L}_{pen} \mathbf{K} \mathbf{A}\}}\,. \tag{44}$$

Similarly to the linear case, in order to find the optimal projections, we resolve the generalized eigenproblem:

$$\mathbf{K}\mathbf{L}_{int}\mathbf{K}\mathbf{a} = \lambda \mathbf{K}\mathbf{L}_{pen}\mathbf{K}\mathbf{a}\,, \tag{45}$$

keeping the eigenvectors that correspond to the smallest eigenvalues.

5.4. Subclass Extraction

From the above discussion, the need for efficient data clustering, is evident. A variety of clustering methods has been proposed in the literature. Techniques such as K-means and Expectation-Maximization (EM) [39] have been used for extracting clusters in a database. It is well-known that there is no method that outperforms the rest in all cases. A relatively new technique relying on spectral graph theory [40], called Spectral Clustering (SC), has also been proposed for data clustering. It has been shown that SC often outperforms traditional clustering algorithms such as K-Means [41]. However, the use of this method has certain limitations, described in [42]. SC can be used for the estimation of the correct number of subclasses within each class [41]. Another potential advantage of SC is that it uses the Gram matrix, which is also used by Kernel SGE. Therefore, when combining SC with Kernel SGE, the Gram matrix has to be calculated once, hence reducing the computational load. In this paper, a multiscale Spectral Clustering (MSC) approach, proposed in [43] has been used, in order to extract clusters within each class of the data at different scales.

6. SGE as a General Dimensionality Reduction Framework

From the previous analysis, a correspondence can be established between a specific criterion –which has to be optimized– and a specific graph matrix. In this subsection, it is shown that SGE is a generalized framework that can be used for subspace learning, since all the standard approaches are specific cases of SGE. Let us use the *Gaussian similarity function* (2), in order to construct the affinity matrix.

In the following analysis, we initially let the variance of Gaussian σ^2 to infinity. Hence,

$$S(\mathbf{x}_q, \mathbf{x}_p) = 1,\ \forall (q,p) \in \{1, 2, \cdots, n\}^2 .$$

Let the intrinsic matrix elements be:

$$\mathbf{P}^{ij}(q,p) = \begin{cases} \frac{S(\mathbf{x}_q, \mathbf{x}_p)}{n_{ij}} = \frac{1}{n_{ij}} & \text{, if } \mathbf{x}_q, \mathbf{x}_p \in C_{ij} \\ 0 & \text{, otherwise} \end{cases} , \tag{46}$$

where C_{ij} is the set of the samples that belong to the j-th subclass of the i-th class.

Obviously, (25) becomes the within-subclass criterion of CDA. Thus, in this case, $\mathbf{W}_{int}$ is the intrinsic graph matrix of CDA. Let also:

$$Q_{ij}^{lh} = S(\boldsymbol{\mu}^{ij}, \boldsymbol{\mu}^{lh}) = 1, \forall\, i, j, h, l \tag{47}$$

the penalty matrix elements. Then, (33) becomes the between-subclass criterion of CDA. Thus, $\mathbf{W}_{pen}$ is the penalty graph matrix of CDA and the connection between CDA and GE has been established.

Let us consider that each data sample constitutes its own class, i.e., $c = n$, $d_i = 1$ and $n_i = 1$, $\forall i \in \{1, 2, \cdots, c\}$. Thus, each class-block of the penalty graph matrix reduces to a single element of the matrix. Obviously, each data sample coincides with the mean of its class. By setting:

$$Q_{i1}^{l1} = \frac{S(\boldsymbol{\mu}^i, \boldsymbol{\mu}^l)}{n} = \frac{1}{n},\ \forall\, (i,l) \in \{1, 2, \cdots, c\}^2 , \tag{48}$$

then:

$$-\frac{\left(\sum_{\omega \neq i} \left(\sum_{t=1}^{d_\omega} Q_{i1}^{\omega t}\right)\right)}{(n_i)^2} = -\sum_{\omega \neq i} \left(\frac{1}{n}\right) = \frac{1}{n} - 1 . \tag{49}$$

These values lie on the main diagonal of the penalty graph matrix. Regarding the off diagonal elements we have:

$$\frac{Q_{i1}^{l1}}{n_i n_l} = \frac{1}{n} . \tag{50}$$

It can be easily shown that the degree penalty matrix is $\mathbf{D} = \mathbf{0}$, so that $\mathbf{L}_{pen} = -\mathbf{W}_{pen}$. Obviously, $\mathbf{L}_{pen} = \mathbf{I} - \frac{1}{n}\mathbf{e}^n (\mathbf{e}^n)^T$ and $\mathbf{X}\mathbf{L}_{pen}\mathbf{X}^T$ becomes the covariance matrix $\mathbf{C}$ of the data. By using as intrinsic graph matrix the identity matrix, SGE becomes identical to PCA:

$$\text{argmin}\, \frac{tr\{\mathbf{V}^T \mathbf{X}\mathbf{L}_{int}\mathbf{X}^T \mathbf{V}\}}{tr\{\mathbf{V}^T \mathbf{X}\mathbf{L}_{pen}\mathbf{X}^T \mathbf{V}\}} = \text{argmin}\, \frac{tr\{\mathbf{V}^T \mathbf{I}\mathbf{V}\}}{tr\{\mathbf{V}^T \mathbf{C}\mathbf{V}\}} \tag{51}$$

leading to the following generalized eigenproblem:

$$\mathbf{I}\mathbf{v} = \lambda \mathbf{C}\mathbf{v} , \tag{52}$$

Table 1: Dimensionality Reduction Using SGE Framework

	$\mathbf{P}(\mathbf{L}_{int})$	$Q(\mathbf{L}_{pen})$	σ^2	c	d_i	d
LPP	$\mathbf{P}^{11}(q,p) = \exp\left(-\frac{d^2(\mathbf{x}_q,\mathbf{x}_p)}{\sigma^2}\right), \forall \mathbf{x}_q, \mathbf{x}_p$	$Q_{11}^{11} = \frac{n^2}{1-n}\left(\mathbf{L}_{pen} = \mathbf{I}\right)$	σ^2	1	1	1
PCA	$\mathbf{L}_{int} = \mathbf{I}$	$Q_{i1}^{l1} = \frac{1}{n}$	∞	n	1	n
LDA	$\mathbf{P}^{i1}(q,p) = \frac{1}{n_i}, \mathbf{x}_q, \mathbf{x}_p \in c_i$	$Q_{i1}^{l1} = \frac{n_i n_l}{n}$	∞	c	1	c
CDA	$\mathbf{P}^{ij}(q,p) = \frac{1}{n_{ij}}, \mathbf{x}_q, \mathbf{x}_p \in c_{ij}$	$Q_{ij}^{lh} = 1$	∞	c	d_i	d
SDA	$\mathbf{L}_{int} = \mathbf{I} - \frac{1}{n}\mathbf{e}^n\left(\mathbf{e}^n\right)^T$	$Q_{ij}^{lh} = \frac{n_{ij}n_{lh}}{n}$	∞	c	d_i	d

solved by keeping the smallest eigenvalues, or by setting $\mu = \frac{1}{\lambda}$, since $\lambda \neq 0$, this leads to:

$$\mathbf{C}\mathbf{v} = \mu\mathbf{I}\mathbf{v}\,, \tag{53}$$

solved by keeping the greatest eigenvalues, which is obviously the PCA solution.

Now, consider that every class consists of a unique subclass, thus $d_i = 1, \forall i \in \{1, 2, \ldots, c\}$. If we set:

$$\mathbf{P}(q,p) = \begin{cases} \frac{S(\mathbf{x}_q,\mathbf{x}_p)}{n_i} = \frac{1}{n_i} & \text{, if } \mathbf{x}_q, \mathbf{x}_p \in C_i \\ 0 & \text{, otherwise} \end{cases}, \tag{54}$$

then the intrinsic graph matrix becomes that of LDA. Furthermore, if we set:

$$Q_{i1}^{l1} = \frac{n_i n_l}{n}\,, \forall\,(i,l) \in \{1, \ldots, c\}^2 \tag{55}$$

then

$$-\frac{\left(\sum_{\omega \neq i}\left(\sum_{t=1}^{d_\omega} Q_{i1}^{\omega t}\right)\right)}{(n_i)^2} = \frac{n_i - n}{n n_i} \tag{56}$$

and

$$\frac{Q_{i1}^{l1}}{n_i n_l} = \frac{1}{n}\,. \tag{57}$$

These are the values of the penalty graph matrix of LDA. So, by taking the Laplacians of the above matrices, we end up to the LDA algorithm.

Let us now reject the assumption that the variance of Gaussian tends to infinity. Consider that there is only one class which contains the whole set of the data, i.e., $c = 1$. Also consider that there are no subclasses within this unique class, i.e., $d_1 = 1$. In this case the intrinsic graph matrix becomes equal to $\mathbf{P}$. Thus, by setting $\mathbf{P}$ equal to the affinity matrix $\mathbf{S}$, the intrinsic Laplacian matrix becomes that of LPP.

We observe that by utilizing the identity matrix as the penalty Laplacian matrix, obviously we get the LPP algorithm. Since we consider a unique class, which contains a unique subclass, from (36) and (37) we have that $\mathbf{W}_{pen} = \mathbf{W}^{11}$. The values of $\mathbf{W}^{11}$ are given from (38), which in this case reduces to:

$$\mathbf{W}^{11} = -\frac{Q_{11}^{11}}{n^2}\mathbf{e}^n\left(\mathbf{e}^n\right)^T \ . \tag{58}$$

If we set:

$$Q_{11}^{11} = \frac{n^2}{1-n}\,, \tag{59}$$

then $\mathbf{W}_{pen} = \mathbf{W}^{11} = \frac{1}{n-1}\mathbf{e}^n\left(\mathbf{e}^n\right)^T$. Consequently,

$$\mathbf{L}_{pen} = \begin{pmatrix} 1 & \frac{1}{1-n} & \cdots & \frac{1}{1-n} \\ \frac{1}{1-n} & 1 & \cdots & \frac{1}{1-n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{1}{1-n} & \frac{1}{1-n} & \cdots & 1 \end{pmatrix}. \tag{60}$$

Thus, if we make the assumption that the number of the data-samples becomes very large, then asymptotically we have $\mathbf{L}_{pen} = \mathbf{I}$.

Finally, to complete the analysis, if we consider as the intrinsic Laplacian matrix, the matrix

$$\mathbf{L}_{int} = \mathbf{I} - \frac{1}{n}\mathbf{e}^n\left(\mathbf{e}^n\right)^T \tag{61}$$

and if we set:

$$Q_{ij}^{lh} = \frac{n_{ij}n_{lh}}{n}\,, \tag{62}$$

in (38) and (40), SGE becomes identical to SDA. The parameters that determine the connection of the several methods with SGE are summarized in Table 1.

7. Subclass Marginal Fisher Analysis

Having established the connection of a set of state-of-the-art DR methods with the SGE framework, we are now at the position to propose a novel algorithm for dimensionality reduction. Motivated by the well-known Marginal Fisher Analysis (MFA) method presented in [7], we propose Subclass Marginal Fisher Analysis (SMFA) employing the SGE framework. The new method combines the power of subclass methods with the agility of the classical MFA to overcome the limitation of the intraclass Gaussian distribution assumption. The intrinsic graph matrix characterizes the intra-subclass compactness, while the penalty graph matrix characterizes

the inter-class separability. Both graph matrices are built using neighbouring information of the graph nodes. More specifically, the intrinsic graph matrix is defined as:

$$\mathbf{P}^{ij}(q,p) = \begin{cases} 1 & \text{, if } p \in \mathcal{N}_{k_{int}}(q) \text{ or } q \in \mathcal{N}_{k_{int}}(p) \\ 0 & \text{, otherwise} \end{cases} , \tag{63}$$

where $\mathcal{N}_{k_{int}}(q)$ denotes the index set of the k_{int} nearest neighbours of the q-th sample in subclass j of class i. Recall that $\mathbf{P}^{ij}(q,p)$ is the value of the intrinsic matrix at the (q,p) position of the j-th subclass belonging to the i-th class. The penalty graph matrix is defined as:

$$\mathbf{W}^{i,l}_{pen}(p,q) = \begin{cases} 1 & \text{, if } i \neq l \text{ and } \left(p \in \mathcal{M}_{k_{pen}}(q) \text{ or } q \in \mathcal{M}_{k_{pen}}(p)\right) \\ 0 & \text{, otherwise} \end{cases} , \tag{64}$$

where $\mathcal{M}_{k_{pen}}(q)$ denotes the set of samples that belong to the k_{pen} nearest neighbours of q outside the class of q. Recall that $\mathbf{W}^{i,l}_{pen}(p,q)$ is the value of the penalty matrix at the (q,p) position, where q belongs to the i-th class and p belongs to the l-th class. It is worth noting that in contrast to the intrinsic graph matrix, the values of the penalty graph matrix depend on the class information regardless of the subclass labels. In this way we avoid to put constraints between subclasses belonging to the same class offering better generalization chances.

The proposed SMFA algorithm inherits all the advantages of the typical MFA method. More specifically, there is no assumption on the data distribution, since the intra-subclass compactness is encoded by the nearest neighbours of the data belonging to the same subclass and the inter-class separability is modelled using the margins among the classes. Moreover, the functionality of SMFA is based on two parameters, i.e., k_{int} and k_{pen}, which appropriately adjusted may lead to avoiding potential overfitting therefore offering huge generalization power to the method. Also, the available projection dimensionality using SMFA is determined by k_{pen}, which almost always is much larger than that of LDA, CDA and SDA. Finally, SMFA is capable of handling and leveraging potential subclass structure of the data, which in many cases may boost its performance. In the following section, the superiority of SMFA over a number of previously presented state-of-the-art DR methods in terms of classification accuracy is demonstrated through a series of experiments.

8. Experimental Results

We conducted classification experiments on several real-world datasets using the proposed linear and kernel SGE framework. For validating the performance of the algorithms, the *5-fold cross-validation* procedure has been used. For extracting automatically the subclass structure, we have utilized the MSC technique [43], keeping the most plausible partition for each dataset. For classifying the data, the Nearest Centroid (NC) classifier has been used with LPP, PCA, LDA and MFA algorithms, while the Nearest Cluster Centroid (NCC) [44] has been used with CDA, SDA and SMFA algorithms. In NCC, the cluster centroids are calculated and the test sample is assigned to the class of the nearest cluster centroid. NC and NCC were selected because they provide

Table 2: Cross Validation Classification Accuracies (%) of Linear Methods on Several Real-World Datasets

DATASET	LPP	PCA	LDA	MFA	CDA	SDA	SMFA
FER-AIIA	40.9(3)	31.0(120)	64.6(6)	72.6(10)	73.2	**75.5(11)**	72.6(12)
BU	39.4(298)	38.1(49)	51.6(6)	**52.4(6)**	49.1(16)	52.3(15)	49.3(11)
JAFFE	46.8(18)	37.6(39)	53.2(6)	**61.5(14)**	40.0(15)	54.1(6)	44.9(20)
KANADE	34.2(92)	43.3(46)	**67.1(6)**	66.3(19)	59.7(7)	**67.1(5)**	63.8(9)
MNIST	71.1(259)	79.9(135)	84.6(9)	82.8(38)	84.8(15)	85.1(14)	**85.3(40)**
SEMEION	53.6(99)	83.2(55)	88.2(9)	86.9(8)	89.2(19)	**89.4(19)**	87.5(10)
XM2VTS	95.7(54)	92.0(86)	70.5(1)	97.7(4)	98.1(3)	97.4(2)	**98.4(4)**
IONOSPHERE	**84.6(23)**	72.3(15)	78.9(1)	76.0(12)	80.6(2)	83.4(2)	84.3(26)
MONK 1	66.7(3)	68.3(5)	50.8(1)	71.7(2)	70.0(4)	74.2(3)	**78.3(2)**
MONK 2	56.0(1)	53.3(4)	52.0(1)	58.7(2)	54.2(1)	54.0(2)	**60.7(1)**
MONK 3	77.2(5)	80.9(4)	49.4(1)	81.6(1)	74.6(2)	66.3(2)	**86.1(5)**
PIMA	61.8(1)	63.5(6)	56.5(1)	74.4(1)	60.5(3)	73.5(3)	**74.9(1)**
SPECIFIC RANK	5.1	5.8	5.0	3.0	4.0	2.7	2.3
OVERALL RANK	9.0	9.8	8.5	5.0	6.6	5.0	4.0

the optimal classification solutions in Bayesian terms, thus proving whether the DR methods have reached the goal described by their specific criterion.

In the following paragraphs, we briefly present the datasets that have been used along with the performance rates of the various subspace learning methods.

8.1. Classification experiments

For the classification experiments, we have used diverse publicly available datasets offered for various classification problems. More specifically, FER-AIIA, BU, JAFFE and KANADE were used for facial expression recognition, XM2VTS for face frontal view recognition, while MNIST and SEMEION for optical digit recognition. Finally, IONOSPHERE, MONK and PIMA were used in order to further extend our experimental study to diverse data classification problems.

In our experiments, for performing DR we have used both the linear and the RBF kernel approach. The maximal dimensionality of the reduced space is determined by the rank of the corresponding matrices utilized by the discriminant analysis methods. Moreover, as already mentioned, LPP is a parametric method regarding the variance of *Gaussian similarity function*, when constructing the affinity matrix. Thus, looking for the optimal variance, in order to achieve the best classification results, makes the comparison very complex. In this paper, for the sake of simplicity and relying on some empirical studies of ours, this parameter was allowed to take values in the range $[0.1 \cdot \hat{E}(d_{ij}), 2.0 \cdot \hat{E}(d_{ij})]$, with step $0.1 \cdot \hat{E}(d_{ij})$, where $\hat{E}$ denotes the sample mean and d_{ij} is the Euclidean distance between i, j samples. Similarly, MFA and SMFA both depend on k_{int} and k_{pen} parameters. Since, to the best of our knowledge, there is no study on the optimal values of these parameters in the literature and as an exhaustive grid search is impossible for computational purposes, we experimented with several indicative values and the best results obtained are reported.

Table 3: Cross Validation Classification Accuracies (%) of Kernel Methods on Several Real-World Datasets

DATASET	KLPP	KPCA	KDA	KMFA	KCDA	KSDA	KSMFA
FER-AIIA	50.2(252)	41.5(29)	54.9(6)	**61.3(9)**	56.1(12)	53.5(12)	56.7(39)
BU	**52.7(317)**	35.9(290)	46.6(6)	44.4(29)	41.0(13)	48.0(14)	39.9(18)
JAFFE	28.8(98)	25.9(58)	42.4(6)	**47.8(6)**	36.1(18)	46.3(5)	34.1(13)
KANADE	32.7(99)	33.2(88)	44.3(6)	**46.6(6)**	40.0(6)	38.5(6)	45.8(7)
MNIST	81.4(299)	64.5(155)	86.0(9)	86.4(21)	83.4(19)	85.2(15)	**86.7(34)**
SEMEION	83.8(99)	77.4(77)	95.3(9)	90.0(11)	94.1(19)	**95.9(19)**	94.9(20)
XM2VTS	71.3(297)	74.7(56)	61.3(1)	78.7(31)	71.5(3)	57.3(4)	**81.2(4)**
IONOSPHERE	83.7(23)	70.3(2)	92.9(1)	92.3(1)	**93.1(1)**	92.9(1)	92.6(1)
MONK 1	63.3(2)	**72.5(1)**	55.8(1)	60.0(1)	58.3(4)	61.7(3)	70.8(4)
MONK 2	54.8(1)	59.8(3)	69.7(1)	70.8(2)	78.7(1)	54.5(1)	**79.7(2)**
MONK 3	62.5(2)	**79.2(5)**	51.7(1)	**79.2(2)**	67.5(2)	58.3(1)	73.3(2)
PIMA	50.7(3)	**67.5(4)**	48.9(1)	54.0(3)	52.5(3)	52.9(1)	56.2(3)
SPECIFIC RANK	5.3	5.0	4.3	2.8	3.9	4.1	2.6
OVERALL RANK	10.2	10.0	8.1	6.3	8.2	8.3	6.1

The cross-validation classification accuracy rates for the several subspace learning methods over the utilized datasets, are summarized in Tables 2 and 3 for the linear and the kernel methods, respectively. The optimal dimensionality of the projected space that returned the above results is depicted in parenthesis. For each dataset, the best performance rate among linear and kernel methods separately is highlighted with bold, while the best overall performance rate among all methods, both linear and kernel, is surrounded by a rectangle.

For ranking the methods in terms of classification performance we further conducted a post-hoc Bonferroni test [45] for each pair of methods. The performance of pairwise methods is significantly different, if the corresponding average ranks differ by at least the critical difference $CD = q_\alpha \sqrt{\frac{j(j+1)}{6T}}$ [46], where j is the number of methods compared, T is the number of data sets and critical values q_α can be found in [47]. In our comparisons we set $\alpha = 0.05$. The ranking has been performed including both linear and kernel methods in the comparison, as well as separately for the linear and kernel methods. The classification performance rank of each method is referred to in the last two rows of Tables 2 and 3. Specific Rank denotes the method rank independently for the linear and the kernel methods. Overall rank refers to the rank of each method among both the linear and the kernel methods. The ranking results are also illustrated in Fig. 1 left and right, for the linear and kernel methods, respectively. The vertical axis in both figures depicts the various methods, while the horizontal axis depicts the performance ranking. The circles indicate the mean rank and the intervals around them indicate the confidence interval as this is determined by the CD value. Overlapping intervals between two methods indicate that there is not a statistically significant difference between the corresponding ranks.

The first remark from Tables 2,3 and Fig. 1 is that SMFA and KSMFA outperform the rest methods in the linear and kernel case, respectively. Although their superiority is not statistically significant over all remaining methods, undoubtedly these two methods offer a strong potential to improve the performance or the state-of-the-art in many classification domains. In addition, it is interesting to observe the robustness of SMFA and

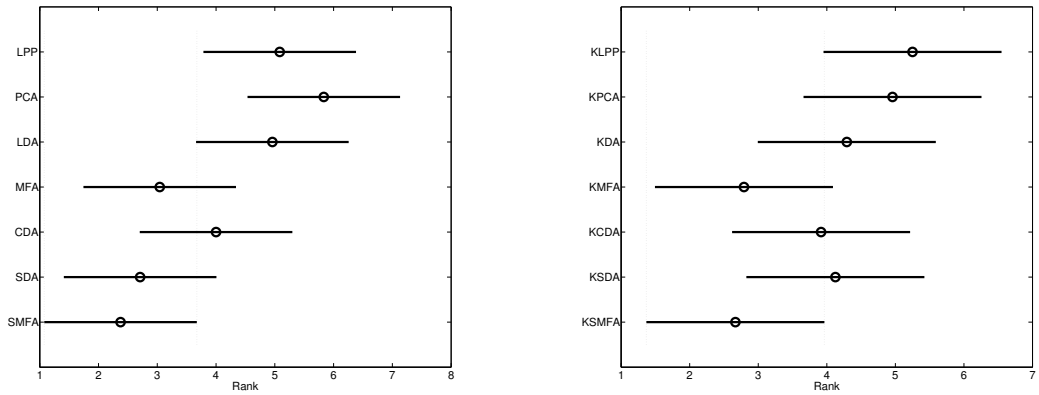

Figure 1: Ranking of Various Methods After Pairwise Post-Hoc Bonferroni Tests on Real Data. (Left: Linear Methods, Right: Kernel Methods)

MFA along with their kernel counterparts across the datasets. This observation combined with the fact that both these methods rely on the same motivations shows the advantage gained by encoding the data distributions using neighbouring information between the samples towards overcoming the several limitations previously presented in this paper, offering at the same time great generalization chances.

As a general remark, the superiority of subclass methods against unimodal ones is evident, with MFA and KMFA being vivid exceptions. The top overall performance is shown by SMFA followed by SDA and MFA, while the worst performance is shown by KLPP. More specifically, on the one hand, SDA, MFA and KMFA display on average the best performance in facial expression recognition problems. On the other hand, in optical digit recognition, face frontal view recognition and the remaining classification problems, SMFA and KSMFA clearly have on average the optimal performance.

In comparing linear with kernel methods, a simple calculation yields mean overall rank equal to 6.84 for the linear methods and 8.17 for the kernel ones. Although the difference between the two approaches (i.e., linear and kernel) is significant, we must admit that there is ample space for improving the kernel results by varying the RBF parameter, as the selection of this parameter is not trivial and may easily lead to over-fitting. Actually, the top performance rates presented in this paper have been obtained by testing indicative values of the above parameter. As a matter of fact, it is interesting to observe that the use of kernels proves to be beneficial for some methods in certain datasets, while deteriorates the performance of others. For instance, from Tables 2 and 3, the use of kernels boosts the performance of PCA in three out of the four last datasets (i.e., MONK 1, MONK 3 and PIMA), while this is not the case for example in XM2VTS. There are two main reasons for this. Firstly, while some datasets contain linearly separable classes, others need some kernel to obtain this linearity. The second reason is that in our experiments, for relaxing the computational complexity, we have used the same kernel values per dataset across all methods and there is no fact advocating that the same value constitutes the

optimal parameter for each method.

9. Conclusions

In this paper, data subclass information has been incorporated within Graph Embedding (GE) leading to a novel Subclass Graph Embedding (SGE) framework, which constitutes the main contribution of our work. In particular, it has been shown that SGE comprises a generalization of GE, encapsulating a number of state-of-the-art unimodal subspace learning techniques already integrated within GE. Besides, the connection of SGE with subspace learning algorithms that use subclass information in the embedding process has been also analytically proven. The physical meaning of the graphs involved in SGE has been described providing some intuition. Also the kernelization of SGE has been presented.

The contribution of this paper has been enriched by a novel Subclass Marginal Fisher Analysis (SMFA) dimensionality reduction method, which has been designed employing SGE. The functionality of SMFA is based on adjacency information of data samples within the same subclass as well as the proximity of "marginal" samples belonging to different classes. In this way, the new method combines the flexibility of neighbourhood modelling methods like MFA with the modularity offered by subclass information towards overcoming inherent limitations stemming from the data distributions, offering at the same moment great generalization chances.

Through an extensive experimental study, it has been shown that SMFA outperforms a number of state-of-the-art subspace learning methods in many real-world datasets pertaining to various classification domains. Similar remarks could be also drawn for KSMFA. In addition, the experimental results highlight the superiority in terms of classification performance of subclass-based methods against unimodal ones. Most importantly, through the superiority of SMFA and generally of subclass-based methods, SGE has gained credibility as a powerful generalized platform for designing novel dimensionality reduction methods.

Although the performance of the proposed method is impressive, there is yet space for exploring new methods employing SGE, either by designing completely new methods or by modifying SMFA. Experimenting on this direction is encompassed in our future plans. Moreover, in order to reinforce even more the outcomes of this paper and to provide more credibility to SGE, in the near future we intend to extend our current experimental study to more datasets from additional classification domains.

Acknowledgments

The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement no 248434 (MOBISERV).

References

[1] X. He, P. Niyogi, Locality preserving projections, in: S. Thrun, L. K. Saul, B. Schölkopf (Eds.), NIPS, MIT Press, 2003.

[2] X. He, S. Yan, Y. Hu, P. Niyogi, H. Zhang, Face recognition using laplacianfaces, IEEE Trans. Pattern Anal. Mach. Intell 27 (3) (2005) 328–340.

[3] I. Jolliffe, Principal Component Analysis, Springer Verlag, 1986.

[4] D. J. Kriegman, J. P. Hespanha, P. N. Belhumeur, Eigenfaces vs. fisherfaces: Recognition using class-specific linear projection, in: ECCV, 1996, pp. I:43–58.

[5] X. W. Chen, T. S. Huang, Facial expression recognition: A clustering-based approach, Pattern Recognition Letters 24 (9-10) (2003) 1295–1302.

[6] M. L. Zhu, A. M. Martinez, Subclass discriminant analysis, IEEE Trans. Pattern Analysis and Machine Intelligence 28 (8) (2006) 1274–1286.

[7] S. Yan, D. Xu, B. Zhang, H.-J. Zhang, Q. Yang, S. Lin, Graph embedding and extensions: A general framework for dimensionality reduction, Pattern Analysis and Machine Intelligence, IEEE Transactions on 29 (1) (2007) 40–51.

[8] J. B. Tenenbaum, V. de Silva, J. C. Langford, A global geometric framework for nonlinear dimensionality reduction., Science 290 (5500) (2000) 2319–2323.

[9] S. T. Roweis, L. K. Saul, Nonlinear dimensionality reduction by locally linear embedding., Science 290 (5500) (2000) 2323–2326.

[10] M. Belkin, P. Niyogi, Laplacian eigenmaps and spectral techniques for embedding and clustering., Advances in Neural Information Processing Systems (NIPS) 14 (2001) 585–591.

[11] J. Ye, R. Janardan, C. H. Park, H. Park, An optimization criterion for generalized discriminant analysis on undersampled problems., IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) 26 (8) (2004) 982–994.

[12] M. Zhu, A. M. Martínez, Pruning noisy bases in discriminant analysis, IEEE Transactions on Neural Networks 19 (1) (2008) 148–157.

[13] W. J. Krzanowski, P. Jonathan, W. V. McCarthy, M. R. Thomas, General interest section: Discriminant analysis with singular covariance matrices: Methods and applications to spectroscopic data., Applied Statistics 44 (1) (1995) 101–115.

[14] J. H. Friedman, Regularized discriminant analysis, Journal of the American Statistical Association 84 (405) (1989) 165–175.

[15] M. Kyperountas, A. Tefas, I. Pitas, Weighted piecewise lda for solving the small sample size problem in face verification, IEEE Transactions on Neural Networks 18 (2) (2007) 506–519.

[16] O. C. Hamsici, A. M. Martinez, Bayes optimality in linear discriminant analysis, IEEE Trans. Pattern Analysis and Machine Intelligence 30 (4) (2008) 647–657.

[17] T. Hastie, A. Buja, R. Tibshirani, Penalized discriminant analysis, Annals of Statistics 23 (1995) 73–102.

[18] G. Baudat, F. Anouar, Generalized discriminant analysis using a kernel approach, Neural Computation 12 (10) (2000) 2385–2404.

[19] M. Loog, R. P. W. Duin, R. Haeb-Umbach, Multiclass linear dimension reduction by weighted pairwise fisher criteria., IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) 23 (7) (2001) 762–766.

[20] G. Goudelis, S. Zafeiriou, A. Tefas, I. Pitas, Class-specific kernel-discriminant analysis for face verification, IEEE Transactions on Information Forensics and Security 2 (3-2) (2007) 570–587.

[21] N. Gkalelis, V. Mezaris, I. Kompatsiaris, Mixture subclass discriminant analysis, Signal Processing Letters, IEEE 18 (5) (2011) 319–322.

[22] S. K. Zhou, R. Chellappa, Multiple-exemplar discriminant analysis for face recognition., International Conference on Pattern Recognition (ICPR) (4) (2004) 191–194.

[23] X. Wu, X. Chen, X. Li, L. Zhou, J. Lai, Adaptive subspace learning: an iterative approach for document clustering, Neural Computing and Applications 1–10.

[24] X. Shu, Y. Gao, H. Lu, Efficient linear discriminant analysis with locality preserving for face recognition, Pattern Recognition 45 (5) (2012) 1892–1898.

[25] K.-R. Müller, S. Mika, G. Rätsch, S. Tsuda, B. Schölkopf, An introduction to kernel-based learning algorithms., IEEE Transactions on Neural Networks 12 (2) (2001) 181–202.

[26] B. Schölkopf, A. J. Smola, K.-R. Muller, Kernel principal component analysis., in: Proceedings of the International Conference on Artificial Neural Networks (ICANN-1997), 1997, pp. 583–588.

[27] M.-H. Yang, Kernel eigenfaces vs. kernel fisherfaces: Face recognition using kernel methods., in: FGR, IEEE Computer Society, 2002, pp. 215–220.

[28] B. Ma, H. Y. Qu, H. S. Wong, Kernel clustering-based discriminant analysis, Pattern Recognition 40 (1) (2007) 324–327.

[29] D. You, O. C. Hamsici, A. M. Martínez, Kernel optimization in discriminant analysis, IEEE Trans. Pattern Anal. Mach. Intell. 33 (3) (2011) 631–638.

[30] Y. Cui, L. Fan, A novel supervised dimensionality reduction algorithm: Graph-based fisher analysis, Pattern Recognition 45 (4) (2012) 1471–1481.

[31] J. Shi, Z. Jiang, H. Feng, Adaptive graph embedding discriminant projections, Neural Processing Letters (2013) 1–16.

[32] E. Zare Borzeshi, M. Piccardi, K. Riesen, H. Bunke, Discriminative prototype selection methods for graph embedding, Pattern Recognition.

[33] G. Arvanitidis, A. Tefas, Exploiting graph embedding in support vector machines, in: Machine Learning for Signal Processing (MLSP), 2012 IEEE International Workshop on, IEEE, 2012, pp. 1–6.

[34] X. He, P. Niyogi, Locality preserving projections., Advances in Neural Information Processing Systems (NIPS).

[35] M. A. Turk, A. P. Pentland, Face recognition using eigenfaces., in: Proceedings IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), 1992, pp. 586–590.

[36] I. Jolliffe, Principal Component Analysis., Springer Verlag, 1986.

[37] R. A. Fisher, The statistical utilization of multiple measurements., Annals of Eugenics 8 (1938) 376–386.

[38] R. Diestel, Graph theory., Vol. 173, Springer-Verlag, 2005.

[39] G. J. McLachlan, T. Krishnan, The EM algorithm and extensions., 2nd Edition, Wiley series in probability and statistics, Wiley, Hoboken, NJ, 2008.

[40] Doob, Spectral graph theory., in: J. L. Gross, J. Yellen (Eds.), Handbook of Graph Theory, CRC Press, 2004, 2004.

[41] U. von Luxburg, A tutorial on spectral clustering, Statistics and Computing 17 (4) (2007) 395–416.

[42] U. von Luxburg, O. Bousquet, M. Belkin, Limits of spectral clustering., in: Advances in Neural Information Processing Systems (NIPS), Vol. 17, MIT Press, 2005, pp. 857–864.

[43] A. Azran, Z. Ghahramani, Spectral methods for automatic multiscale data clustering., in: IEEE Computer Vision and Pattern Recognition (CVPR) (1), IEEE Computer Society, 2006, pp. 190–197.

[44] A. Maronidis, A. Tefas, I. Pitas, Frontal view recognition using spectral clustering and subspace learning methods., in: W. D. K. I. Diamantaras, L. S. Iliadis (Eds.), ICANN (1), Vol. 6352 of Lecture Notes in Computer Science, Springer, 2010, pp. 460–469.

[45] O. J. Dunn, Multiple comparisons among means, Journal of American Statistical Association 56 (293) (1961) 52–64.

[46] H. Chen, P. Tino, X. Yao, Probabilistic classification vector machines., IEEE Transactions on Neural Networks 20 (6) (2009) 901–914.

[47] J. Demsar, Statistical comparisons of classifiers over multiple data sets, Journal of Machine Learning Research 7 (2006) 1–30.

Appendix A.

In this Appendix, the derivation of eq. (27) from eq. (25) is given.

$$\frac{1}{2}tr\{\mathbf{V}^T\left(\sum_{i=1}^{c}\sum_{j=1}^{d_i}\sum_{q=1}^{n_{ij}}\sum_{p=1}^{n_{ij}}\left(\mathbf{x}_q^{ij}-\mathbf{x}_p^{ij}\right)\mathbf{P}^{ij}(q,p)\left(\mathbf{x}_q^{ij}-\mathbf{x}_p^{ij}\right)^T\right)\mathbf{V}\} = \tag{25}$$

$$tr\{\mathbf{V}^T\mathbf{A}\mathbf{V}\}, \tag{A.1}$$

where $\mathbf{A}$ is given by

$$\mathbf{A} = \sum_{i=1}^{c}\sum_{j=1}^{d_i}\sum_{q=1}^{n_{ij}}\sum_{p=1}^{n_{ij}}\mathbf{x}_q^{ij}\mathbf{P}^{ij}(q,p)\left(\mathbf{x}_q^{ij}\right)^T - \sum_{i=1}^{c}\sum_{j=1}^{d_i}\sum_{q=1}^{n_{ij}}\sum_{p=1}^{n_{ij}}\mathbf{x}_q^{ij}\mathbf{P}^{ij}(q,p)\left(\mathbf{x}_p^{ij}\right)^T =$$

$$\sum_{i=1}^{c}\sum_{j=1}^{d_i}\sum_{q=1}^{n_{ij}}\mathbf{x}_q^{ij}\overbrace{\left(\sum_{p=1}^{n_{ij}}\mathbf{P}^{ij}(q,p)\right)}^{f(q,j,i)}\left(\mathbf{x}_q^{ij}\right)^T - \sum_{i=1}^{c}\sum_{j=1}^{d_i}\sum_{q=1}^{n_{ij}}\sum_{p=1}^{n_{ij}}\mathbf{x}_q^{ij}\mathbf{P}^{ij}(q,p)\left(\mathbf{x}_p^{ij}\right)^T =$$

$$\sum_{i=1}^{c}\sum_{j=1}^{d_i}\sum_{q=1}^{n_{ij}}\mathbf{x}_q^{ij}f(q,j,i)\left(\mathbf{x}_q^{ij}\right)^T - \sum_{i=1}^{c}\sum_{j=1}^{d_i}\sum_{q=1}^{n_{ij}}\sum_{p=1}^{n_{ij}}\mathbf{x}_q^{ij}\mathbf{P}^{ij}(q,p)\left(\mathbf{x}_p^{ij}\right)^T =$$

$$\sum_{i=1}^{c}\sum_{l=1}^{c}\sum_{j=1}^{d_i}\sum_{h=1}^{d_l}\sum_{q=1}^{n_{ij}}\sum_{p=1}^{n_{lh}}\mathbf{x}_q^{ij}\left[\delta_{il}\delta_{jh}\left(\delta_{qp}f(q,j,i)-\mathbf{P}^{ij}(q,p)\right)\right]\left(\mathbf{x}_p^{lh}\right)^T =$$

$$\sum\mathbf{x}_q^{ij}\left[\delta_{il}\delta_{jh}\left(\delta_{qp}f(q,j,i)-\mathbf{P}^{ij}(q,p)\right)\right]\left(\mathbf{x}_p^{lh}\right)^T,$$

where δ_{il} denotes the true delta function. From now on by $\sum$ we will denote the above six-fold summation. Using the following relation,

$$\mathbf{XLX}^T = \sum_{s=1}^{n}\sum_{t=1}^{n} \mathbf{x}_s L_{st} \mathbf{x}_t^T = \sum \mathbf{x}_q^{ij} L_{qp}\left(i,l,j,h\right) \mathbf{x}_p^{lh}\,, \tag{A.2}$$

where $L_{qp}\left(i,l,j,h\right)$ is the value of $\mathbf{L}$ at position (a,b), where a corresponds to the q-th sample of the j-th cluster of the i-th class and b corresponds to the p-th sample of the h-th cluster of the l-th class, then it can be easily shown that

$$\mathbf{A} = \mathbf{X}\left(\mathbf{D}_{int} - \mathbf{W}_{int}\right)\mathbf{X}^T = \mathbf{XL}_{int}\mathbf{X}^T\,,$$

where the values of $\mathbf{W}_{int}$ are those given by (28) and (29). Thus, (A.1) becomes

$$tr\{\mathbf{V}^T\mathbf{XL}_{int}\mathbf{X}^T\mathbf{V}\}\,,$$

which is essentially eq. (27).

Appendix B.

In this Appendix, the derivation of eq. (35) from eq. (33) is given.

$$tr\{\mathbf{V}^T\left(\sum_{i=1}^{c-1}\sum_{l=i+1}^{c}\sum_{j=1}^{d_i}\sum_{h=1}^{d_l}\left(\boldsymbol{\mu}^{ij}-\boldsymbol{\mu}^{lh}\right)Q_{ij}^{lh}\left(\boldsymbol{\mu}^{ij}-\boldsymbol{\mu}^{lh}\right)^T\right)\mathbf{V}\} = \tag{33}$$

$$tr\{\mathbf{V}^T\mathbf{BV}\}\,, \tag{B.1}$$

where $\mathbf{B} = \mathbf{B}_1 + \mathbf{B}_2 + \mathbf{B}_3 + \mathbf{B}_4$ and $\mathbf{B}_k$ are given below.

$$\mathbf{B}_1 = \sum_{i=1}^{c-1}\sum_{l=i+1}^{c}\sum_{j=1}^{d_i}\sum_{h=1}^{d_l}\boldsymbol{\mu}^{ij}Q_{ij}^{lh}\left(\boldsymbol{\mu}^{ij}\right)^T =$$

$$\sum_{i=1}^{c-1}\sum_{l=i+1}^{c}\sum_{j=1}^{d_i}\boldsymbol{\mu}^{ij}\overbrace{\left(\sum_{h=1}^{d_l}Q_{ij}^{lh}\right)}^{f(i,j,l)}\left(\boldsymbol{\mu}^{ij}\right) =$$

$$\sum_{i=1}^{c-1}\sum_{j=1}^{d_i}\boldsymbol{\mu}^{ij}\overbrace{\left(\sum_{l=i+1}^{c}f(i,j,l)\right)}^{g(i,j)}\left(\boldsymbol{\mu}^{ij}\right) = \sum_{i=1}^{c-1}\sum_{j=1}^{d_i}\boldsymbol{\mu}^{ij}g(i,j)\left(\boldsymbol{\mu}^{ij}\right)^T\,, \tag{B.2}$$

where

$$g(i,j) = \left(\sum_{\omega=i+1}^{c}\sum_{t=1}^{d_\omega}Q_{ij}^{\omega t}\right),\ 1 \le i \le c-1,\ 1 \le j \le d_i\,.$$

We extend $g(i,j)$ to $i = c$ as

$$g'(i,j) = \begin{cases} 0 & , i = c \\ g(i,j) & , i < c \end{cases}$$

Hence, (B.2) becomes

$$\sum_{i=1}^{c-1}\sum_{j=1}^{d_i} \boldsymbol{\mu}^{ij} g(i,j) \left(\boldsymbol{\mu}^{ij}\right)^T = \sum_{i=1}^{c}\sum_{l=1}^{c}\sum_{j=1}^{d_i}\sum_{h=1}^{d_l} \boldsymbol{\mu}^{ij} \left(\delta_{il}\delta_{jh}\right) g'(i,j) \left(\boldsymbol{\mu}^{lh}\right)^T =$$

$$\sum \mathbf{x}_q^{ij} \frac{\delta_{il}\delta_{jh} g'(i,j)}{\left(n_{ij}\right)^2} \left(\mathbf{x}_p^{lh}\right)^T .$$

$$\mathbf{B}_2 = -\sum_{i=1}^{c-1}\sum_{l=i+1}^{c}\sum_{j=1}^{d_i}\sum_{h=1}^{d_l} \boldsymbol{\mu}^{ij} Q_{ij}^{lh} \left(\boldsymbol{\mu}^{lh}\right)^T =$$

$$\sum \mathbf{x}_q^{ij} \frac{\rho(i,l,j,h)}{n_{ij} n_{lh}} \left(\mathbf{x}_p^{lh}\right)^T ,$$

where

$$\rho(i,l,j,h) = \begin{cases} 0 & , (i = c) \vee (i < c \wedge l \leq i) \\ -Q_{ij}^{lh} & , \text{otherwise} \end{cases}$$

$$\mathbf{B}_3 = -\sum_{i=1}^{c-1}\sum_{l=i+1}^{c}\sum_{j=1}^{d_i}\sum_{h=1}^{d_l} \boldsymbol{\mu}^{lh} Q_{ij}^{lh} \left(\boldsymbol{\mu}^{ij}\right)^T =$$

$$\sum \mathbf{x}_q^{lj} \frac{\rho(i,l,j,h)}{n_{ij} n_{lh}} \left(\mathbf{x}_p^{ij}\right)^T =$$

$$-\sum \mathbf{x}_q^{ij} \frac{\rho(l,i,h,j)}{n_{ij} n_{lh}} \left(\mathbf{x}_p^{lh}\right)^T .$$

$$\mathbf{B}_4 = \sum_{i=1}^{c-1}\sum_{l=i+1}^{c}\sum_{j=1}^{d_i}\sum_{h=1}^{d_l} \boldsymbol{\mu}^{lh} Q_{ij}^{lh} \left(\boldsymbol{\mu}^{lh}\right)^T =$$

$$\sum_{i=1}^{c-1}\sum_{l=i+1}^{c}\sum_{h=1}^{d_l} \boldsymbol{\mu}^{lh} \overbrace{\left(\sum_{j=1}^{d_i} Q_{ij}^{lh}\right)}^{f(i,l,h)} \left(\boldsymbol{\mu}^{lh}\right)^T =$$

$$\sum_{i=1}^{c-1}\sum_{l=i+1}^{c} \overbrace{\sum_{h=1}^{d_l} \boldsymbol{\mu}^{lh} f(i,l,h) \left(\boldsymbol{\mu}^{lh}\right)^T}^{t(i,l)} = \sum_{i=1}^{c-1}\sum_{l=i+1}^{c} t(i,l) = \sum_{i=1}^{c-1} d(i) , \tag{B.3}$$

where

$$d(i) = t(i, i+1) + t(i, i+2) + \cdots + t(i, c)\,.$$

(B.3) becomes

$$\sum_{i=1}^{c-1} d(i) = d(1) + d(2) + \cdots + d(c-1) =$$

$$[t(1,2) + t(1,3) + \cdots + t(1,c)] + [t(2,3) + \cdots + t(2,c)] + \cdots + [t(c-1,c)] =$$

$$[t(1,2)] + [t(1,3) + t(2,3)] + \cdots + [t(1,c) + \cdots + t(c-1,c)] =$$

$$\sum_{l=2}^{c}\sum_{i=1}^{l-1} t(i,l) = \sum_{l=2}^{c}\sum_{i=1}^{l-1}\sum_{h=1}^{d_l} \boldsymbol{\mu}^{lh} f(i,l,h)\left(\boldsymbol{\mu}^{lh}\right)^T =$$

$$\sum_{l=2}^{c}\sum_{h=1}^{d_l} \boldsymbol{\mu}^{lh} \overbrace{\left(\sum_{i=1}^{l-1} f(i,l,h)\right)}^{e(l,h)} \left(\boldsymbol{\mu}^{lh}\right)^T = \sum_{l=2}^{c}\sum_{h=1}^{d_l} \boldsymbol{\mu}^{lh} e(l,h)\left(\boldsymbol{\mu}^{lh}\right)^T\,, \tag{B.4}$$

where

$$e(l,h) = \sum_{\omega=1}^{i-1}\sum_{t=1}^{d_\omega} Q_{\omega t}^{lh},\ 2 \le l \le c,\ 1 \le h \le d_l\,.$$

We extend $e(l,h)$ to $l = 1$ as

$$e'(l,h) = \begin{cases} 0 & , l = 1 \\ e(l,h) & , l \ge 1 \end{cases}$$

Hence, (B.4) becomes

$$\sum_{l=1}^{c}\sum_{i=1}^{c}\sum_{h=1}^{d_l}\sum_{j=1}^{d_i} \boldsymbol{\mu}^{lh} \delta_{li}\delta_{hj} e'(l,h)\left(\boldsymbol{\mu}^{ij}\right)^T =$$

$$\sum \mathbf{x}_q^{ij} \frac{\delta_{il}\delta_{jh} e'(i,j)}{\left(n_{ij}\right)^2}\left(\mathbf{x}_p^{lh}\right)^T\,.$$

Summarizing the above we have that

$$\mathbf{B}_1 + \mathbf{B}_2 + \mathbf{B}_3 + \mathbf{B}_4 =$$

$$\sum \mathbf{x}_q^{ij}\left(\frac{\delta_{il}\delta_{jh}[g'(i,j) + e'(i,l)]}{\left(n_{ij}\right)^2} + \frac{[\rho(i,l,j,h) + \rho(l,i,h,j)]}{n_{ij}n_{lh}}\right)\left(\mathbf{x}_p^{lh}\right)^T\,.$$

By definition of g' and e' we have that

$$g'(i,j) + e'(i,j) = \sum_{\omega=i+1}^{c} \sum_{t=1}^{d_\omega} Q_{ij}^{\omega t} + \sum_{\omega=1}^{i-1} \sum_{t=1}^{d_\omega} Q_{ij}^{\omega t} =$$

$$\sum_{\omega \neq i} \sum_{t=1}^{d_\omega} Q_{ij}^{\omega t} = \sum_{\omega \neq i} f(\omega, i, j)\,.$$

Also, it can be easily shown that

$$\rho(i,l,j,h) + \rho(l,i,h,j) = \begin{cases} 0 & i = l \\ Q_{ij}^{lh} & i \neq l \end{cases}\,.$$

Finally, **B** becomes equal to

$$tr\{\mathbf{V}^T \left(\sum \mathbf{x}_q^{ij} \frac{\delta_{il}\delta_{jh}\left(\sum_{\omega \neq i} \sum_{t=1}^{d_\omega} Q_{ij)}^{\omega t}\right) - \delta_{(i \neq l)} Q_{ij}^{lh}}{n_{ij} n_{lh}} \left(\mathbf{x}_p^{lh}\right)^T \right) \mathbf{V}\}\,.$$

In the same manner as in Appendix A, using (A.2) it can be shown that

$$\mathbf{B} = \mathbf{X}\left(\mathbf{D}_{pen} - \mathbf{W}_{pen}\right)\mathbf{X}^T = \mathbf{X}\mathbf{L}_{pen}\mathbf{X}^T\,, \tag{35}$$

where the values of $\mathbf{W}_{pen}$ are those given by eq. (36) – (40). Thus, eq. (B.2) becomes

$$tr\{\mathbf{V}^T \mathbf{X}\mathbf{L}_{pen}\mathbf{X}^T \mathbf{V}\}\,,$$

which is essentially eq.(35).