Graph Embedded Non-Parametric Mutual Information For Supervised Dimensionality Reduction

Dimitrios Bouzas, Nikolaos Arvanitopoulos, *Student Member, IEEE,* and Anastasios Tefas, *Member, IEEE.*

***Abstract*—In this paper we propose a novel algorithm for dimensionality reduction, which uses as a criterion the mutual information between the transformed data and their corresponding class labels. The mutual information is a powerful criterion which can be used as a proxy to the Bayes error rate. Furthermore, recent quadratic non-parametric implementations of mutual information are computationally efficient and do not require any prior assumptions about the class densities. We show that the quadratic non-parametric mutual information can be formulated as a kernel objective in the graph embedding framework. Moreover, we propose its linear equivalent as a novel linear dimensionality reduction algorithm. The derived methods are compared against the state-of-the-art dimensionality reduction algorithms with various classifiers and on various benchmark and real-life datasets. The experimental results show that non-parametric mutual information as an optimization objective for dimensionality reduction gives comparable and in most of the cases better results compared to other dimensionality reduction methods.**

***Index Terms*—Dimensionality Reduction, Mutual Information, Feature Extraction, Quadratic Mutual Information, Face Recognition, Graph Embedding Framework, Data Visualization.**

I. Introduction

The problem of dimensionality reduction has attracted the attention of a vast number of researchers in computer vision and pattern recognition. This is mainly attributed to the fact that in many systems, dimensionality reduction is the necessary preprocessing step to efficiently manipulate high-dimensional data or to denoise them [4]. A dimensionality reduction algorithm is an approach which, given a set of high-dimensional data of dimensionality d, maps them into a lower-dimensional space of dimensionality l, where $l \ll d$. For this to be accomplished, many *linear* and *non-linear* algorithms have been proposed. Among them, *Principal Component Analysis (PCA)* [18] and *Linear Discriminant analysis (LDA)* [9], [12] are the most popular linear ones in the categories of *unsupervised* and *supervised* dimensionality reduction respectively. By the term supervised dimensionality reduction we mean that the specific method uses the data class labels in order to achieve a mapping, linear or non-linear, such that the resulting mapped data which belong to different classes will be well separated [11]. Another linear approach that can be either supervised or unsupervised, is the *Locality preserving projections (LPP)* [15]. LPP tries to preserve the local relationships within the data samples and thus to reveal their overall structure. On the other side of dimensionality reduction lie the non-linear methods like *Laplacian Eigenmaps (LEM)* [3], *Locally Linear Embedding (LLE)* [27] and *ISOMAP* [32]. These methods try to find non-linear projections of the data that are more likely to detect their latent non-linear manifold structure. With the recent advent of the *kernel trick* [31] most of the linear methods can be reformulated as kernel ones. Kernels have been extensively used in the context of *Support Vectors Machines (SVM)* [6]. The main idea behind the kernels is to employ a linear mapping $\phi : \mathbb{R}^m \rightarrow \mathcal{H}$ in a high-dimensional *Hilbert space* $\mathcal{H}$ where in the original space this mapping will be non-linear [17]. Many linear methods have been extended to their kernel equivalent like *Kernel Principal Component Analysis (KPCA)* [30], *Kernel Fisher Discriminant (KFD)* [26], *Generalized Discriminant Analysis (GDA)* [1], *Complete Kernel Fisher Discriminant (CKFD)* [36] and many recent ones like *Kernel Orthogonal Neighborhood Preserving Projections (KONPP)* [22].

Recently, Shuicheng Yan et. al proposed a general framework for dimensionality reduction, the so called *Graph Embedding Framework* [35]. Graph embedding unifies most of the popular dimensionality reduction methods under a well defined framework while it gives the ability to formulate new ones. Under the context of the Graph Embedding Framework, we propose the formulation of a novel dimensionality reduction algorithm that uses as an optimization criterion the *Mutual Information (MI)* [7]. The intuition behind the use of the MI is the fact that it is a general criterion which can overcome limitations of the previous proposed methods. MI uses high-order statistics, and not just second-order ones as, for example, LDA. Furthermore, it can be can used as an alternative to the Bayes error rate, which is the optimal criterion for classification [33]. However, computing the MI is computationally inefficient, since probability density functions of variables are required and also high-dimensional numerical integration of thoses. In [33], an efficient *quadratic non-parametric* formulation of the MI (QMI) between the data feature vectors and their corresponding class labels is proposed, which gives better MI estimation in high-dimensional spaces with acceptable computational cost.

In our work we show that QMI can be integrated in the Graph Embedding Framework and hence, we are able to derive

D. Bouzas is with the Beta CAE systems S.A, Epanomi, Greece, GR-57500 (email: bouzas@beta-cae.gr)

N. Arvanitopoulos is with the School of Computer and Communication Sciences (IC), École Polytechnique Fédérale de Lausanne (EPFL), Switzerland. email: nick.arvanitopoulos@epfl.ch

A. Tefas is with the Department of Informatics, Aristotle University of Thessaloniki, Thessaloniki 54124, Greece. e-mail: tefas@aiia.csd.auth.gr

a closed mathematical form for the optimization of the corresponding objective criterion. In particular, we show that under the Graph Embedding Framework, QMI can be reformulated to produce a kernel dimensionality reduction method that we call *Kernel Quadratic Mutual Information (KQMI)*. Secondly, we derive the linear equivalent of this method, called *Linear Quadratic Mutual Information (LQMI)*. Under the perspective of the closed form optimization formula, we are given the opportunity to evaluate the use of the non-parametric MI as a dimensionality reduction objective. Furthermore, we compare the reformulated methods against the mainstream projective dimensionality reduction approaches available, with various classifiers and on several benchmark and real-life datasets. Our results show that our proposed algorithms attain comparable and in most of the cases better results than other state-of-the-art supervised dimensionality reductions methods.

Our paper is organized as follows. In Section II we describe the prior work of several researchers that form the basis of our proposed method. In Section III we give all the theoretical background of our work. In Section IV we illustrate our experimental results. Finally, in Section V we conclude our work.

II. Prior Work

In this Section we comment on the work of several researchers that supplied the motivation for the derivation of our work.

A. Graph Embedding Framework

Recent work has shown that many dimensionality reduction algorithms can be reformulated into the *Graph Embedding Framework* [35]. The Graph Embedding Framework is based on the introduction of the undirected weighted graph $\mathbf{G} = (\mathbf{X}, \mathbf{W})$, with vertex set the data points $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_n]^T \in \mathbb{R}^{n \times d}$ and similarity matrix $\mathbf{W} \in \mathbb{R}^{n \times n}$. The graph embedding of the graph $\mathbf{G}$ is, therefore, an algorithm to find the low-dimensional representation of the data that best preserve the relationships between the vertex pairs of $\mathbf{G}$. The graph $\mathbf{G}$ can be seen as an intrinsic graph. Furthermore, a penalty graph $\mathbf{G}^p = (\mathbf{X}, \mathbf{W}^p)$ can also be defined, such that the weight matrix of the graph penalizes specific characteristics of the data structure. For the one-dimensional case, assuming that $\mathbf{y} = [y_1, \ldots, y_n]^T$ is the vector containing the projections of each data sample $\mathbf{x}_i$, the graph criterion to be optimized is

$$\begin{aligned}\mathbf{y}^* &= \arg \min_{\mathbf{y}^\top \mathbf{C} \mathbf{y} = \epsilon} \sum_{i,j=1}^{n} \|y_i - y_j\|^2 W_{ij} \\ &= \arg \min_{\mathbf{y}^\top \mathbf{C} \mathbf{y} = \epsilon} \mathbf{y}^\top \mathbf{L} \mathbf{y}, \end{aligned} \tag{1}$$

where $\mathbf{L}$ is the graph Laplacian defined as $\mathbf{L} = \mathbf{D} - \mathbf{W}$ and $\mathbf{D}$ is the diagonal degree matrix defined as $D_{ii} = \sum_{j=1}^{n} W_{ij}, i = 1, \ldots, n$. $\mathbf{C}$ is a constraint matrix to avoid trivial solutions and is typically a diagonal matrix for scale normalization, or the graph Laplacian of $\mathbf{G}^p$, that is $\mathbf{C} = \mathbf{L}^p = \mathbf{D}^p - \mathbf{W}^p$ and ϵ is a constant. If we assume that the vector $\mathbf{y}$ is obtained by the linear projection $\mathbf{y} = \mathbf{X}\mathbf{w}$, where $\mathbf{w} \in \mathbb{R}^d$ is the projection vector, then the objective becomes

$$\begin{aligned}\mathbf{w}^* &= \arg \min_{\substack{\mathbf{w}^\top \mathbf{X}^\top \mathbf{C} \mathbf{X} \mathbf{w} = \epsilon \\ \mathbf{w}^\top \mathbf{w} = \epsilon}} \sum_{i,j=1}^{n} \|\mathbf{w}^\top \mathbf{x}_i - \mathbf{w}^\top \mathbf{x}_j\|^2 W_{ij} \\ &= \arg \min_{\substack{\mathbf{w}^\top \mathbf{X}^\top \mathbf{C} \mathbf{X} \mathbf{w} = \epsilon \\ \text{or } \mathbf{w}^\top \mathbf{w} = \epsilon}} \mathbf{w}^\top \mathbf{X}^\top \mathbf{L} \mathbf{X} \mathbf{w}. \end{aligned} \tag{2}$$

Following similar arguments to [30] in the case of PCA (see also (4) below) we see that the solution to the objective (2) should lie on the span of the data points $\mathbf{x}_1, \ldots, \mathbf{x}_n$, therefore it can be written as a linear combination of the form $\mathbf{w} = \sum_{i=1}^{n} \beta_i \mathbf{x}_i$.

The objective in (2), even though is computationally efficient to optimize, is not always optimal in terms of classification perfomance, especially when the underlying data are distributed in a highly non-linear way. A solution to this problem is to introduce the kernel extension of objective (2) to handle non-linearly distributed data by using the kernel trick [31]. The input data are mapped to a higher dimensional Hilbert space $\mathcal{H}$ using a map $\phi : \mathbf{x} \to \mathcal{H}$. In this new feature space, a linear projection algorithm is performed similar to (2). The key property of the kernel trick is that it is based only on inner products of data pairs defined by the kernel function $k(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)^\top \phi(\mathbf{x}_j)$. As in the linear case, the projection direction $\mathbf{w} \in \mathbb{R}^n$ lies in the span of $\phi(\mathbf{x}_1), \ldots, \phi(\mathbf{x}_n)$ (see also [30]), therefore it admits a representation of the form $\mathbf{w} = \sum_{i=1}^{n} \alpha_i \phi(\mathbf{x}_i)$. By defining the kernel matrix $\mathbf{K} \in \mathbb{R}^{n \times n}$ as $K_{ij} = \phi(\mathbf{x}_i)^\top \phi(\mathbf{x}_j)$, the objective in (2) can be written as

$$\begin{aligned}\boldsymbol{\alpha}^* &= \arg \min_{\substack{\boldsymbol{\alpha}^\top \mathbf{K}^\top \mathbf{C} \mathbf{K} \boldsymbol{\alpha} = \epsilon \\ \text{or } \boldsymbol{\alpha}^\top \mathbf{K} \boldsymbol{\alpha} = \epsilon}} \sum_{i,j=1}^{n} \|\boldsymbol{\alpha}^\top \mathbf{K}_i - \boldsymbol{\alpha}^\top \mathbf{K}_j\|^2 W_{ij} \\ &= \arg \min_{\substack{\boldsymbol{\alpha}^\top \mathbf{K}^\top \mathbf{C} \mathbf{K} \boldsymbol{\alpha} = \epsilon \\ \text{or } \boldsymbol{\alpha}^\top \mathbf{K} \boldsymbol{\alpha} = \epsilon}} \boldsymbol{\alpha}^\top \mathbf{K}^\top \mathbf{L} \mathbf{K} \boldsymbol{\alpha}. \end{aligned} \tag{3}$$

The solutions of (1), (2) and (3) can be obtained by solving the generalized eigenvalue problem

$$\mathbf{A}\mathbf{v} = \lambda \mathbf{B}\mathbf{v}, \tag{4}$$

where, $\mathbf{A} = \mathbf{L}, \mathbf{X}^\top \mathbf{L} \mathbf{X}, \mathbf{K}^\top \mathbf{L} \mathbf{K}$ and $\mathbf{B} = \mathbf{I}, \mathbf{C}, \mathbf{X}^\top \mathbf{C} \mathbf{X}, \mathbf{K}, \mathbf{K}^\top \mathbf{C} \mathbf{K}$, depending on the type of the problem.

B. Quadratic Mutual Information

Many ways have been proposed to accurately estimate the MI between data points and their respective class labels. Quadratic Mutual Information (QMI) [33] is an accurate estimation method for high-dimensional problems with acceptable computational cost. We assume a random variable X representing the data points $\mathbf{x}_i \in \mathbb{R}^d$ and a discrete random variable Y representing the class labels. Therefore, we have data pairs of the form $\{\mathbf{x}_i, y_i\}_{i=1}^n$. Let $p(\mathbf{x})$ be the probability density function of the data points and $P(Y)$ the class prior probabilities. The MI between the two random variables is defined as

$$\mathcal{I}(X, Y) = \sum_y \int_{\mathbf{x}} p(\mathbf{x}, y) \log \frac{p(\mathbf{x}, y)}{p(\mathbf{x}) P(y)} d\mathbf{x}. \tag{5}$$

The MI is a measure of dependence between random variables, in our case between the data points X and their class labels Y. The above equation can also be interpreted as a Kullback-Leibler divergence:

$$\mathcal{KL}(Q_1(\mathbf{x},y),Q_2(\mathbf{x},y))=\sum_y\int_{\mathbf{x}}Q_1(\mathbf{x},y)\log\frac{Q_1(\mathbf{x},y)}{Q_2(\mathbf{x},y)}d\mathbf{x}, \tag{6}$$

where $Q_1(\mathbf{x},y)=p(\mathbf{x},y)$ and $Q_2(\mathbf{x},y)=p(\mathbf{x})P(y)$. In [20, p.178], [21, Chapter 4], it is argued that if our goal is to find the distribution that maximizes the divergence and not to compute its absolute value, the axioms used in deriving divergence measures can be relaxed resulting in the same maximizing distribution. One such measure that satisfies the relaxed axioms is given by

$$D_\alpha(Q_1,Q_2)=\frac{1}{\alpha(\alpha-1)}\sum_{i=1}^{n}(q_{1,i}^{\alpha}-\alpha q_{1,i}q_{2,i}^{\alpha-1}+(\alpha-1)q_{2,i}^{\alpha}), \tag{7}$$

where $\alpha\neq 0,1$. Selecting $\alpha=2$ and extending the measure to continuous densities we arrive at the quadratic measure given by (up to a constant)

$$D_2(Q_1,Q_2)=\int_{\mathbf{x}}(Q_1(\mathbf{x})-Q_2(\mathbf{x}))^2d\mathbf{x}. \tag{8}$$

Another justification for using the quadratic divergence measure is given in [33], where it is shown that maximizing $D_2(Q_1,Q_2)$ is equivalent to maximizing a lower bound to $\mathcal{KL}(Q_1,Q_2)$. The MI can now be expressed in terms of the divergence between the joint density and the product of its marginals. Inserting these forms of the distributions into (8) leads to the *Quadratic Mutual Information measure (QMI)* between two continuous variables $\mathbf{x}_1,\mathbf{x}_2$:

$$\mathcal{I}_{\mathcal{Q}}(X_1,X_2)=\int_{\mathbf{x}_1}\int_{\mathbf{x}_2}(p(\mathbf{x}_1,\mathbf{x}_2)-p(\mathbf{x}_1)p(\mathbf{x}_2))^2d\mathbf{x}_1\mathbf{x}_2. \tag{9}$$

In our case, the QMI between the continuous variable X of the data points and the discrete random variable Y of the class labels is defined as

$$\begin{aligned}\mathcal{I}_{\mathcal{Q}}(X,Y)=&\sum_y\int_{\mathbf{x}}p(\mathbf{x},y)^2d\mathbf{x}+\sum_y\int_{\mathbf{x}}p(\mathbf{x})^2P(y)^2d\mathbf{x}\\&-2\sum_y\int_{\mathbf{x}}p(\mathbf{x},y)p(\mathbf{x})P(y)d\mathbf{x}.\end{aligned} \tag{10}$$

In [33], the above probability distributions are approximated using Parzen window estimators with a Gaussian kernel. Therefore, $p(\mathbf{x})$, $P(y_i=c)$ and $p(\mathbf{x},y_i=c)$ can be written as

$$\begin{aligned}p(\mathbf{x})&=\sum_y p(\mathbf{x},y)=\frac{1}{n}\sum_{c=1}^{C}\sum_{j=1}^{J_c}\mathcal{N}(\mathbf{x};\mathbf{x}_j,\sigma^2I)\\&=\frac{1}{n}\sum_{i=1}^{n}\mathcal{N}(\mathbf{x};\mathbf{x}_i,\sigma^2\mathbf{I}),\\P(y_i=c)&=\frac{J_c}{n},\\p(\mathbf{x}|y_i=c)&=\frac{1}{J_c}\sum_{i:y_i=c}\mathcal{N}(\mathbf{x};\mathbf{x}_i,\sigma^2\mathbf{I}),\end{aligned}$$

where in the above equations n is the total number of data points, C is the total number of classes, J_c is the number of samples of class c and $\mathcal{N}(\mathbf{x};\mathbf{m},\mathbf{\Sigma})$ denotes the Gaussian probability distribution function with mean vector $\mathbf{m}$ and covariance matrix $\mathbf{\Sigma}$, defined as

$$\mathcal{N}(\mathbf{x};\mathbf{m},\mathbf{\Sigma})=\frac{1}{\sqrt{2\pi|\mathbf{\Sigma}|}}\exp\left(-\frac{1}{2}(\mathbf{x}-\mathbf{m})^\top\mathbf{\Sigma}^{-1}(\mathbf{x}-\mathbf{m})\right).$$

By expanding equation (10) we get the following equation:

$$\mathcal{I}_Q(\mathbf{x},y)=V_{IN}+V_{ALL}-2V_{BTW}, \tag{11}$$

where

$$\begin{aligned}V_{IN}&=\sum_y\int_{\mathbf{x}}p(\mathbf{x},y)^2d\mathbf{x}\\&=\frac{1}{n^2}\sum_{c=1}^{C}\sum_{j:y_j=c}\sum_{k:y_k=c}\mathcal{N}(\mathbf{x}_j-\mathbf{x}_k;\mathbf{0},2\sigma^2\mathbf{I}),\end{aligned} \tag{12}$$

$$\begin{aligned}V_{ALL}&=\sum_y\int_{\mathbf{x}}p(\mathbf{x})^2P(y)^2d\mathbf{x}\\&=\frac{1}{n^2}\sum_{c=1}^{C}\left(\frac{J_c}{N}\right)^2\sum_{j=1}^{n}\sum_{k=1}^{n}\mathcal{N}(\mathbf{x}_j-\mathbf{x}_k;\mathbf{0},2\sigma^2\mathbf{I}),\end{aligned} \tag{13}$$

$$\begin{aligned}V_{BTW}&=\sum_y\int_{\mathbf{x}}p(\mathbf{x},y)p(\mathbf{x})P(y)d\mathbf{x}\\&=\frac{1}{n^2}\sum_{c=1}^{C}\frac{J_c}{n}\sum_{j=1}^{n}\sum_{k:y_k=c}\mathcal{N}(\mathbf{x}_l-\mathbf{x}_k;\mathbf{0},2\sigma^2\mathbf{I}).\end{aligned} \tag{14}$$

The above pairwise interactions between samples, V_{IN},V_{ALL},V_{BTW} can be interpreted as follows [33]:

- V_{IN} can be seen as interactions between pairs of samples inside each class.
- V_{ALL} consists of interactions between all pairs of samples, regardless of class membership.
- V_{BTW} consists of interactions between samples of each class against all other samples.

III. Graph Embedding of Quadratic Mutual Information

In this Section, we show that the QMI can be formulated into the Graph Embedding Framework and can be interpreted as a direct kernelization of a linear objective.

A. Formulation of KQMI and LQMI Algorithms

We assume our initial data points are centralized, that is they have zero mean. Otherwise, we subtract from each data sample the mean vector of the whole dataset. We define the centralized kernel matrix $\mathbf{K}\in\mathbb{R}^{n\times n}$ with elements $K_{ij}=(\widetilde{\mathbf{K}}-\mathbf{E}_n\widetilde{\mathbf{K}}-\widetilde{\mathbf{K}}\mathbf{E}_n+\mathbf{E}_n\widetilde{\mathbf{K}}\mathbf{E}_n)_{ij}$, where $\widetilde{K}_{ij}=\mathcal{N}(\mathbf{x}_i-\mathbf{x}_j;\mathbf{0},2\sigma^2\mathbf{I})$ and $\mathbf{E}_n$ the $n\times n$ matrix with all elements equal to $1/n$. The kernel matrix can also be written as $\mathbf{K}=\mathbf{\Phi}\mathbf{\Phi}^\top$, where $\mathbf{\Phi}\in\mathbb{R}^{n\times m}$ is the matrix of the mapped data points $\mathbf{X}\in\mathcal{X}$ into a Hilbert space $\mathcal{H}$ through the mapping $\phi:\mathcal{X}\to\mathcal{H}$ and m is

the unknown dimensionality of the feature space. We define $\mathbf{1} = [1, \ldots, 1]^\top \in \mathbb{R}^n$ and $\mathbf{1}_c \in \mathbb{R}^n$ with elements

$$[\mathbf{1}_c]_i = \begin{cases} 1 & \text{if } \mathbf{x}_i \in c \\ 0 & \text{else} \end{cases}.$$

Furthermore, we define the constants $C_{ALL} = \frac{1}{n^4}\sum_{c=1}^{C}(J_c)^2$, $C_{IN} = \frac{1}{n^2}$ and $C_{BTW,c} = \frac{J_c}{n^3}, \quad c = 1, \ldots, C$. With the above notation, the V_{ALL}, V_{IN} and V_{BTW} terms can be written as

$$V_{ALL} = C_{ALL}\text{tr}\{\mathbf{\Phi}^\top \mathbf{1}\mathbf{1}^\top \mathbf{\Phi}\}, \tag{15}$$

$$V_{IN} = C_{IN}\text{tr}\{\mathbf{\Phi}^\top \left(\sum_{c=1}^{C} \mathbf{1}_c \mathbf{1}_c^\top\right) \mathbf{\Phi}\}, \tag{16}$$

$$V_{BTW} = \text{tr}\{\mathbf{\Phi}^\top \mathbf{1} \left(\sum_{c=1}^{C} C_{BTW,c}\mathbf{1}_c^\top\right) \mathbf{\Phi}\}. \tag{17}$$

Using the above, the QMI between the data points and their respective class labels can be reformulated as

$$\mathcal{I}_\mathcal{Q}(\mathbf{x}, y) = \text{tr}\Bigg\{\mathbf{\Phi}^\top \Bigg(C_{ALL}\mathbf{1}\mathbf{1}^\top + C_{IN}\Bigg(\sum_{c=1}^{C}\mathbf{1}_c\mathbf{1}_c^\top\Bigg) - 2 \cdot \mathbf{1}\Bigg(\sum_{c=1}^{C} C_{BTW,c}\mathbf{1}_c^\top\Bigg)\Bigg)\mathbf{\Phi}\Bigg\}. \tag{18}$$

Detailed derivations of the expressions (15),(16),(17) can be found in Appendix A. In order to compute non-linear projections of the data points, we define the projection vectors $\mathbf{w}_i \in \mathbb{R}^m, i = 1, \ldots, n$. We can restrict these vectors to be in the range of $\mathbf{\Phi}$, since they belong to $\mathbb{R}^m$, which is the column space of $\mathbf{\Phi}$. Therefore, they can admit a representation of the form $\mathbf{w}_i = \sum_j^n \alpha_{ij}\phi(\mathbf{x}_j) = \mathbf{\Phi}^\top \boldsymbol{\alpha}_i$. By arranging them as columns of a matrix we can create the projection matrix $\mathbf{W} = \{\mathbf{w}_i\}_{i=1}^n = \{\mathbf{\Phi}^\top\boldsymbol{\alpha}_i\}_{i=1}^n = \mathbf{\Phi}^\top\mathbf{A} \in \mathbb{R}^{n\times n}$, where $\mathbf{A} = \{\boldsymbol{\alpha}_i\}_{i=1}^n \in \mathbb{R}^{n\times n}$. After the non-linear projection with the matrix $\mathbf{W}$ and by defining

$$\mathbf{M} = \Bigg(C_{ALL}\mathbf{1}\mathbf{1}^\top + C_{IN}\Bigg(\sum_{c=1}^{C}\mathbf{1}_c\mathbf{1}_c^\top\Bigg) - 2 \cdot \mathbf{1}\Bigg(\sum_{c=1}^{C} C_{BTW,c}\mathbf{1}_c^\top\Bigg)\Bigg), \tag{19}$$

we result in the following formulation of the QMI inspired graph embedding objective:

$$\mathcal{I}_\mathcal{Q}^\mathcal{P}(\mathbf{x}, y) = \text{tr}\{(\mathbf{\Phi W})^\top \mathbf{M}\mathbf{\Phi W}\} = \text{tr}\{\mathbf{A}^\top\mathbf{K}\mathbf{M}\mathbf{K}\mathbf{A}\}. \tag{20}$$

The above formulation uses the matrix $\mathbf{M}$ that represents the interactions between data samples in the graph embedding framework and has been extracted by the QMI formulation in (18) in order to connect the data samples after projection to an arbitrary Hilbert space (represented in $\mathbf{\Phi}^\top$) and dimensionality reduction (represented in $(\mathbf{\Phi W})^\top = \mathbf{A}^\top\mathbf{K}$). The optimal matrix $\mathbf{A}$ can be computed by solving the following optimization problem:

$$\mathbf{A}^* = \arg\max_{\mathbf{A}^\top\mathbf{K}\mathbf{A}=\mathbf{I}} \text{tr}\{\mathbf{A}^\top\mathbf{K}\mathbf{M}\mathbf{K}\mathbf{A}\}. \tag{21}$$

The constraint $\mathbf{A}^\top\mathbf{K}\mathbf{A} = \mathbf{I}$ is derived from the orthogonality constraint of the projection matrix $\mathbf{W}$, that is, $\mathbf{W}^\top\mathbf{W} = \mathbf{I} \Rightarrow (\mathbf{\Phi}^\top\mathbf{A})^\top(\mathbf{\Phi}^\top\mathbf{A}) = \mathbf{I} \Rightarrow \mathbf{A}^\top\mathbf{\Phi}\mathbf{\Phi}^\top\mathbf{A} = \mathbf{I} \Rightarrow \mathbf{A}^\top\mathbf{K}\mathbf{A} = \mathbf{I}$. In general, the matrix $\mathbf{M}$ is not symmetric due to the $C_{BTW,c}$ terms, unless the classes of the dataset are balanced. However, it is known that for every general square matrix $\mathbf{M} \in \mathbb{R}^{n\times n}$ and $\mathbf{x} \in \mathbb{R}^n$, it holds $\mathbf{x}^\top\mathbf{M}\mathbf{x} = \mathbf{x}^\top\mathbf{M}'\mathbf{x}$, where $\mathbf{M}' = (\mathbf{M} + \mathbf{M}^\top)/2$ the symmetrization of $\mathbf{M}$ [14]. By symmetrizing the matrix $\mathbf{M}$ in (19) we obtain the equivalent objective

$$\mathbf{A}^* = \arg\max_{\mathbf{A}^\top\mathbf{K}\mathbf{A}=\mathbf{I}} \text{tr}\{\mathbf{A}^\top\mathbf{K}\mathbf{M}'\mathbf{K}\mathbf{A}\}. \tag{22}$$

In order to make the problem well-posed (see also Section III-C), we additionally enforce the constraint that the embedding vectors have unit covariance, that is, $\frac{1}{n}(\mathbf{\Phi W})^\top(\mathbf{\Phi W}) = \mathbf{I} \Rightarrow \frac{1}{n}(\mathbf{\Phi}\mathbf{\Phi}^\top\mathbf{A})^\top(\mathbf{\Phi}\mathbf{\Phi}^\top\mathbf{A}) = \mathbf{I} \Rightarrow (\mathbf{K}\mathbf{A})^\top(\mathbf{K}\mathbf{A}) = n\mathbf{I} \Rightarrow \mathbf{A}^\top\mathbf{K}\mathbf{K}\mathbf{A} = n\mathbf{I}$. The final objective becomes

$$\mathbf{A}^* = \arg\max_{\mathbf{A}^\top\mathbf{K}\mathbf{A}=\mathbf{I}} \frac{\text{tr}\{\mathbf{A}^\top\mathbf{K}\mathbf{M}'\mathbf{K}\mathbf{A}\}}{\text{tr}\{\mathbf{A}^\top\mathbf{K}\mathbf{K}\mathbf{A}\}}. \tag{23}$$

The solution of the above optimization problem is given by the generalized eigenvalue problem

$$\mathbf{K}\mathbf{M}'\mathbf{K}\mathbf{U} = \mathbf{\Lambda}\mathbf{K}\mathbf{K}\mathbf{U} \Leftrightarrow \mathbf{M}'\mathbf{K}\mathbf{U} = \mathbf{\Lambda}\mathbf{K}\mathbf{U}, \tag{24}$$

where $\mathbf{\Lambda}$ is the diagonal matrix of the eigenvalues λ_i and $\mathbf{U}$ is the matrix whose columns contain the eigenvectors $\boldsymbol{\upsilon}_i$. To satisfy the constraint $\mathbf{A}^\top\mathbf{K}\mathbf{A} = \mathbf{I}$, each eigenvector $\boldsymbol{\upsilon}_i$ must be divided by $\sqrt{\boldsymbol{\upsilon}_i\mathbf{K}\boldsymbol{\upsilon}_i}$ in order to get normalized eigenvectors $\boldsymbol{\upsilon}_i$. It is important to note here that the matrix $\mathbf{M}'$ has rank $C-1$, therefore the product $\mathbf{K}\mathbf{M}'\mathbf{K}$ has maximum rank also $C-1$. As a result, using this formulation, we are able to utilize only $C-1$ eigenvectors that belong to the column space of $\mathbf{K}\mathbf{M}'\mathbf{K}$ and the maximum projection dimension is $C-1$. The optimal non-linear projection of the data points to dimension $l = 1, \ldots, C-1$ is given by the first l dominant eigenvectors $\boldsymbol{\upsilon}_i$, $i = 1, \ldots, l$, that is $\mathbf{A}^* = \{\boldsymbol{\upsilon}_i\}_{i=1}^l$. The matrix of the projected data points is given by $\mathbf{K}^\mathcal{P} = \mathbf{K}\mathbf{A}^*$.

Another more robust strategy to solve (23) is to apply an eigenvalue decomposition of the kernel matrix $\mathbf{K} = \mathbf{P}\mathbf{L}\mathbf{P}^\top$, where $\mathbf{P}$ is the matrix which contains in its columns the eigenvectors of $\mathbf{K}$ and $\mathbf{L}$ is the diagonal matrix that contains the eigenvalues of $\mathbf{K}$. The quotient becomes now

$$Q = \frac{\text{tr}\{\mathbf{A}^\top\mathbf{P}\mathbf{L}\mathbf{P}^\top\mathbf{M}'\mathbf{P}\mathbf{L}\mathbf{P}^\top\mathbf{A}\}}{\text{tr}\{\mathbf{A}^\top\mathbf{P}\mathbf{L}\mathbf{P}^\top\mathbf{P}\mathbf{L}\mathbf{P}^\top\mathbf{A}\}}. \tag{25}$$

By defining $\mathbf{B} = \mathbf{L}\mathbf{P}^\top\mathbf{A}$ and using the fact that $\mathbf{P}$ is orthonormal, the quotient becomes

$$Q = \frac{\text{tr}\{\mathbf{B}^\top\mathbf{P}^\top\mathbf{M}'\mathbf{P}\mathbf{B}\}}{\text{tr}\{\mathbf{B}^\top\mathbf{B}\}}. \tag{26}$$

This quotient is maximized by solving the eigenvalue problem

$$\mathbf{P}^\top\mathbf{M}'\mathbf{P}\mathbf{Z} = \mathbf{H}\mathbf{Z}, \tag{27}$$

where $\mathbf{Z}$ is the matrix containing the eigenvectors of $\mathbf{P}^\top\mathbf{M}'\mathbf{P}$ and $\mathbf{H}$ a diagonal matrix containing the corresponding eigenvalues. The optimal projection vectors can now be computed by $\mathbf{A}^* = \mathbf{P}^{-\top}\mathbf{L}^{-1}\mathbf{B} = \mathbf{P}\mathbf{L}^{-1}\mathbf{B}$.

The complete algorithm to compute the optimal non-linear projections produced by the QMI between the data points and

Algorithm 1 Kernel QMI

Input:

- Vector $\mathbf{y} = \{y_i | i = 1, \ldots, n\}$, $y_i \in \{1, \ldots, C\}$.
- Centralized data matrix $\mathbf{X} = (\{\mathbf{x}_i\}_{i=1}^n)^\top \in \mathbb{R}^{n \times m}$ with zero mean.

Output:

- Non-linearly projected data $\mathbf{K}^{\mathcal{P}} \in \mathbb{R}^{n \times d}$.

Step 1:

1: Calculate centralized kernel matrix $\mathbf{K} \in \mathbb{R}^{n \times n}$, $K_{ij} = (\widetilde{\mathbf{K}} - \mathbf{E}_n\widetilde{\mathbf{K}} - \widetilde{\mathbf{K}}\mathbf{E}_n + \mathbf{E}_n\widetilde{\mathbf{K}}\mathbf{E}_n)_{ij}$, where $\widetilde{K}_{ij} = \mathcal{N}(\mathbf{x}_i - \mathbf{x}_j, \sigma^2\mathbf{I})$ and $\mathbf{E}_n$ the $n \times n$ matrix with all elements equal to $1/n$.
2: Calculate matrix $\mathbf{M}$ using (19) and symmetrize it to obtain $\mathbf{M}'$.

Step 2:

1: Decompose the kernel matrix $\mathbf{K}$ into its eigenvectors and eigenvalues:
$$\mathbf{K} = \mathbf{P}\mathbf{L}\mathbf{P}^\top.$$
2: Solve the eigenvalue problem
$$\mathbf{P}^\top\mathbf{M}'\mathbf{P}\mathbf{Z} = \mathbf{H}\mathbf{Z}.$$
where $\mathbf{H}$ is the diagonal matrix containing the eigenvalues of $\mathbf{P}^\top\mathbf{M}'\mathbf{P}$ and $\mathbf{Z}$ is the matrix whose columns are the corresponding eigenvectors.
3: Sort the eigenvalues with descending order and arrange the corresponding eigenvectors into the matrix $\mathbf{B}$.
4: Compute the optimal projection vectors as
$$\mathbf{A} = \mathbf{P}\mathbf{L}^{-1}\mathbf{B}.$$

Step 3:

1: Select the first $d = 1, \ldots, C-1$ eigenvectors of $\mathbf{A}$ to create a new matrix $\mathbf{A}^*$and compute the resulting projected data as
$$\mathbf{K}^{\mathcal{P}} = \mathbf{K}\mathbf{A}^*.$$

their respective labels is given in Algorithm 1. The objective in equation (23) can be seen as a direct kernelization of a linear objective of the form

$$\mathbf{W}^* = \arg\max_{\mathbf{W}^\top\mathbf{W}=\mathbf{I}} \frac{\mathrm{tr}\{\mathbf{W}^\top\mathbf{X}^\top\mathbf{M}'\mathbf{X}\mathbf{W}\}}{\mathrm{tr}\{\mathbf{W}^\top\mathbf{X}^\top\mathbf{X}\mathbf{W}\}}, \qquad (28)$$

where $\mathbf{W} = \{\mathbf{w}_i\}_{i=1}^n \in \mathbb{R}^{d \times d}$ contains the projection vectors. The solution of the above optimization problem is given by the generalized eigenvalue problem

$$\mathbf{X}^\top\mathbf{M}'\mathbf{X}\mathbf{V} = \mathbf{\Gamma}\mathbf{X}^\top\mathbf{X}\mathbf{V} \Leftrightarrow (\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{M}'\mathbf{X}\mathbf{V} = \mathbf{\Gamma}\mathbf{V}, \qquad (29)$$

where $\mathbf{\Gamma}$ is the diagonal matrix of the eigenvalues γ_i of $(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{M}'\mathbf{X}$ and $\mathbf{V}$ is the matrix whose columns contain the eigenvectors $\mathbf{v}_i$ of $\mathbf{M}'\mathbf{X}$. To enforce the orthogonality constraint $\mathbf{W}^\top\mathbf{W} = \mathbf{I}$ each eigenvector $\mathbf{v}_i$ must be divided by $\|\mathbf{v}_i\|$ in order to get normalized eigenvectors $\mathbf{v}_i$. The optimal linear projection of the data points to dimension $l = 1, \ldots, d$ is given by the first l dominant eigenvectors $\mathbf{v}_i$, $i = 1, \ldots, l$, that is $\mathbf{W}^* = \{\mathbf{v}_i\}_{i=1}^l$. The matrix of the projected data points is given by $\mathbf{X}^{\mathcal{P}} = \mathbf{X}\mathbf{W}^*$. The algorithm to compute the optimal linear projections is described in Algorithm 2.

Algorithm 2 Linear QMI

Input:

- Vector $\mathbf{y} = \{y_i | i = 1, \ldots, n\}$, $y_i \in \{1, \ldots, C\}$.
- Centralized data matrix $\mathbf{X} = (\{\mathbf{x}_i\}_{i=1}^n)^\top \in \mathbb{R}^{n \times m}$ with zero mean.

Output:

- Linearly projected data $\mathbf{X}^{\mathcal{P}} \in \mathbb{R}^{n \times d}$.

Step 1:

1: Calculate matrix $\mathbf{M}$ as in (19) and symmetrize it to obtain $\mathbf{M}'$.

Step 2:

1: Solve the generalized eigenvalue problem
$$X^\top\mathbf{M}'\mathbf{X}\mathbf{V} = \mathbf{\Gamma}\mathbf{X}^\top\mathbf{X}\mathbf{V} \Leftrightarrow$$
$$\Leftrightarrow (\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{M}'\mathbf{X}\mathbf{V} = \mathbf{\Gamma}\mathbf{V},$$
where $\mathbf{\Gamma}$ is the diagonal matrix containing the eigenvalues of $(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{M}'\mathbf{X}$ and $\mathbf{V}$ is the matrix whose columns are the corresponding eigenvectors.
2: Sort the eigenvalues with descending order and arrange the corresponding eigenvectors into the matrix $\mathbf{W}$.

Step 3:

1: Select the first $d = 1, \ldots, C-1$ eigenvectors of $\mathbf{W}$ to create a new matrix $\mathbf{W}^*$and compute the resulting projected data as
$$\mathbf{X}^{\mathcal{P}} = \mathbf{X}\mathbf{W}^*.$$

B. Graph derivation of KQMI

We now derive the intrinsic graph that corresponds to the proposed dimensionality reduction algorithm, which is based on the non-parametric MI. In the following we assume that the data in the matrix $\mathbf{X}$ are sorted according to their class labels. By defining $\alpha_c = C_{IN} + C_{ALL} - 2C_{BTW,c}$ and $\beta_{c,c'} = C_{ALL} - C_{BTW,c} - C_{BTW,c'}$, the matrix $\mathbf{M}'$ has the form

$$\mathbf{M}' = \begin{pmatrix} \begin{bmatrix} \alpha_1 & \cdots & \alpha_1 \\ \vdots & \ddots & \vdots \\ \alpha_1 & \cdots & \alpha_1 \end{bmatrix} & \cdots & \begin{bmatrix} \beta_{1,C} & \cdots & \beta_{1,C} \\ \vdots & \ddots & \vdots \\ \beta_{1,C} & \cdots & \beta_{1,C} \end{bmatrix} \\ \vdots & \ddots & \vdots \\ \begin{bmatrix} \beta_{1,C} & \cdots & \beta_{1,C} \\ \vdots & \ddots & \vdots \\ \beta_{1,C} & \cdots & \beta_{1,C} \end{bmatrix} & \cdots & \begin{bmatrix} \alpha_C & \cdots & \alpha_C \\ \vdots & \ddots & \vdots \\ \alpha_C & \cdots & \alpha_C \end{bmatrix} \end{pmatrix}, \qquad (30)$$

where each submatrix containing the α_c, $c = 1, \ldots, C$ entries has dimensions $J_c \times J_c$ and each submatrix containing the $\beta_{c,c'}$, $c = 1, \ldots, C$, $c' = 1, \ldots, C$ entries has dimensions $J_c \times J_{c'}$.

By defining a graph with weight matrix $\mathbf{W}$ as

$$\mathbf{W} = \begin{pmatrix} \begin{bmatrix} 0 & \cdots & -\alpha_1 \\ \vdots & \ddots & \vdots \\ -\alpha_1 & \cdots & 0 \end{bmatrix} & \cdots & \begin{bmatrix} -\beta_{1,C} & \cdots & -\beta_{1,C} \\ \vdots & \ddots & \vdots \\ -\beta_{1,C} & \cdots & -\beta_{1,C} \end{bmatrix} \\ \vdots & \ddots & \vdots \\ \begin{bmatrix} -\beta_{1,C} & \cdots & -\beta_{1,C} \\ \vdots & \ddots & \vdots \\ -\beta_{1,C} & \cdots & -\beta_{1,C} \end{bmatrix} & \cdots & \begin{bmatrix} 0 & \cdots & -\alpha_C \\ \vdots & \ddots & \vdots \\ -\alpha_C & \cdots & 0 \end{bmatrix} \end{pmatrix}, \qquad (31)$$

the matrix $\mathbf{M}'$ is the Laplacian matrix of this graph and can be written as $\mathbf{M}' = \mathbf{D} - \mathbf{W}$, where $\mathbf{D} = \text{diag}(\mathbf{M}')$. In Appendix B it is shown that the property $\sum_{j=1}^{n} W_{ij} = D_{ii},\ i = 1, \ldots, n$ holds. The above weight matrix $\mathbf{W}$ corresponds to a fully connected graph, where data samples that belong to the same class are connected with edges of non-positive weights $-\alpha_c = 2C_{BTW,c} - C_{IN} - C_{ALL}$ and data samples belonging to different classes are connected with edges of non-negative weights $-\beta_{c,c'} = C_{BTW,c} + C_{BTW,c'} - C_{ALL}$.

C. Discussion

One of the key contributions of the proposed approach is to interpret the QMI in the graph embedding framework in order to derive novel closed form dimensionality reduction objectives. To do so, someone has to consider either to maximize the QMI between the samples and the labels after the dimensionality reduction, which is what Torrkola did, giving an iterative algorithm in his work [33], or to use the information provided by the QMI formulation in order to build new dimensionality reduction criteria that can be solved in closed form, which is what we propose. If someone wants to follow the second approach he can observe that in the QMI formulation given in (18) we have two elements that define the measure. One is the matrix $\mathbf{\Phi}$ that corresponds to the data samples in the feature space and has been derived by a specific kernel used in Parzen estimation (i.e., the Gaussian kernel), to define sample similarity. The other element is the matrix $\mathbf{M}$ in (19) that represents the links between the data samples in the intrinsic graph that forms the QMI measure. Thus, using different kernels in Parzen estimation will result in different forms of $\mathbf{\Phi}$ with the same matrix $\mathbf{M}$ that defines the graph connections. Moreover, someone can directly use the matrix $\mathbf{M}$ in the graph embedding framework and change the feature map using other well-known kernels. The simplest form of $\mathbf{\Phi}$ is the original data matrix $\mathbf{X}$. If we consider that the data samples are the outcome of a projection to lower dimension using $\mathbf{W}$ then the data matrix $\mathbf{X}$ is replaced by $\mathbf{XW}$. Proceeding from the linear case to the nonlinear case, the projection matrix $\mathbf{W}$ is a linear combination of the samples in the feature space given in $\mathbf{\Phi}$, that is $\mathbf{W} = \{\mathbf{w}_i\}_{i=1}^{n} = \{\mathbf{\Phi}^\top \boldsymbol{\alpha}_i\}_{i=1}^{n} = \mathbf{\Phi}^\top \mathbf{A} \in \mathbb{R}^{n \times n}$. In all cases, the dimensionality reduction objectives depend on a constant graph embedding matrix $\mathbf{M}$ that represents the QMI sample interactions and a feature matrix $\mathbf{\Phi}$ that represents the samples in the corresponding Hilbert space and can be considered to correspond to a specific kernel used in Parzen estimation. Finding the kernel for the Parzen estimation that corresponds to the well-known kernels used in projecting samples in RKHS is out the scope of the paper and will be considered in future research.

For the derivation of KQMI in Section III-A we enforce an additional constraint of unit covariance of the embedding vectors. This modification was done for two reasons. First, this covariance condition implies that the projected points $\mathbf{\Phi W}$ will be differenct from each other, because of the orthogonality of the columns of $\mathbf{\Phi W}$ [22]. This is similar to what PCA does. Second, we also conducted experiments without this constraint, however they were slightly worse than the ones we present in Section IV using the additional constraint.

Furthermore, another logical question arises of why to use the MI as a criterion for Dimensionality Reduction. It is known that the Bayes error rate is the optimal criterion for classification and it can take the form [33]

$$E(X) = \int_{\mathbf{x}} p(\mathbf{x})(1 - \max_i p(y_i|\mathbf{x}))d\mathbf{x}. \tag{32}$$

The above criterion needs the computation of class posterior probabilities and numerical integration of those. This is a difficult problem given only one training dataset. Several approximations have been proposed which use parametric estimation of class-conditional densities followed by numerical optimization [13], [29]. For example, LDA assumes all classes to be Gaussian with a single shared covariance matrix. The key difference of MI with the already proposed approximations is the fact that it accounts for high-order statistics and not only the second order. Another major property is that MI bounds the Bayes error rate. An upper bound of the form $E(X) \leq \frac{1}{2}(H(Y) - I(X,Y))$ is given in [16]. Furthermore, in [8] a lower bound involving the Bayes error rate and the MI is proved. Both bounds are minimized when the MI between classes and data points is maximized. That means that we can use the MI as an alternative criterion of the Bayes error rate.

IV. Experimental Results

In this Section we illustrate the experimental results obtained by comparing our proposed methods against several dimensionality reduction methods available.

A. Experimental Results on Benchmark Datasets

We compared our proposed dimensionality reduction methods against other competitive methods (i.e., LDA, PCA, supervised LPP, GDA, KCFD, supervised KLPP, KPCA) using 20 benchmark datasets from the UCI [10], and Statlog [25], repositories. The characteristics of each dataset can be seen in Table I. Let us note here that the methods LPP and KLPP can be either unsupervised or supervised. In this evaluation, we have implemented the supervised versions, since our goal is to compare the supervised KQMI with as many supervised methods as possible. In these versions of the algorithms, we compute the KNN graph by connecting with edges the points that belong to the same class and they are among the k nearest neighbors of each other.

All the features of each dataset were scaled to the interval $[-1, +1]$. To evaluate the test error on the different experiments, we used 10-fold cross validation. In each fold, we first compute the eigenvectors for each method based only on the fold's training set and then we project the feature vectors of the fold's training and test sets on the acquired eigenvectors. As classifiers we used the *Nearest Class Centroid (NCC)* classifier, the *K Nearest Neighbours (KNN)* classifier [12] with $K = 3$ and the LIBSVM's [5] SVM classifier with *linear* and *Radial Basis Function (RBF)* kernels. We set the cost variable C for both linear and RBF SVM to $C = 100$ and for the RBF SVM we set the RBF kernel's σ to $\sigma = 1$. For the kernel dimensionality reduction methods we used RBF kernel with $\sigma = 1$. The experimental results for the NCC

TABLE II: Classification error rates of Linear and Kernel Dimensionality Reduction methods with Nearest Class Centroid Classifier. In parentheses the dimensionality of the final projected features is shown. The last row shows the number of wins for each method across all datasets.

	Kernel Methods					Linear Methods			
Dataset	**KQMI**	**GDA**	**KLPP**	**CKFD**	**KPCA**	**LQMI**	**LDA**	**PCA**	**LPP**
Australian	15.51 % (1)	24.20 % (1)	16.52 % (1)	20.44 % (1)	**14.49 % (14)**	14.20 % (1)	14.20 % (1)	**13.77 % (4)**	14.20 % (1)
Balance	**13.12 % (1)**	16.47 % (2)	13.44 % (1)	13.74 % (1)	23.83 % (4)	**25.76 % (2)**	29.89 % (2)	26.23 % (4)	**25.76 % (2)**
Breast Cancer	**3.23 % (1)**	4.25 % (1)	3.66 % (1)	3.81 % (1)	3.36 % (4)	3.82 % (1)	3.82 % (1)	**3.52 % (1)**	3.82 % (1)
Dermatology	**12.84 % (5)**	14.80 % (5)	14.51 % (5)	17.53 % (10)	17.26 % (34)	**2.98 % (5)**	3.53 % (5)	3.27 % (17)	**2.98 % (5)**
Diabetes	**24.75 % (1)**	29.05 % (1)	26.70 % (1)	27.22 % (1)	27.74 % (7)	23.57 % (1)	**23.44 % (1)**	26.83 % (7)	**23.44 % (1)**
Ecoli	**19.13 % (7)**	19.44 % (7)	19.38 % (7)	19.67 % (7)	26.61 % (5)	22.92 % (5)	**22.09 % (7)**	24.65 % (7)	22.10 % (6)
German Numer	**28.10 % (1)**	28.30 % (1)	28.30 % (1)	28.50 % (2)	40.40 % (23)	**27.60 % (1)**	**27.60 % (1)**	29.50 % (22)	**27.60 % (1)**
Glass	32.87 % (5)	32.80 % (5)	**29.53 % (5)**	30.42 % (5)	52.97 % (8)	**38.63 % (5)**	43.60 % (5)	55.26 % (8)	43.29 % (5)
Heart	**19.23 % (1)**	22.59 % (1)	22.59 % (1)	21.11 % (2)	20.37 % (13)	**15.56 % (1)**	**15.56 % (1)**	18.52 % (11)	**15.56 % (1)**
Ionosphere	8.81 % (1)	6.52 % (1)	**5.39 % (1)**	6.52 % (2)	26.14 % (19)	**12.75 % (1)**	16.44 % (1)	25.27 % (8)	**12.75 % (1)**
Iris	**2.67 % (2)**	4.00 % (2)	3.33 % (2)	4.66 % (2)	6.66 % (4)	**2.00 % (1)**	**2.00 % (1)**	5.33 % (1)	**2.00 % (1)**
Liver Disorders	**28.94 % (1)**	32.97 % (1)	29.50 % (1)	29.50 % (1)	41.98 % (4)	37.40 % (1)	**37.12 % (1)**	40.51 % (6)	**37.12 % (1)**
Segment	3.72 % (6)	**2.46 % (6)**	4.37 % (6)	4.16 % (6)	15.45 % (19)	**9.74 % (6)**	35.58 % (6)	16.02 % (11)	11.99 % (5)
Sonar	13.03 % (1)	**12.53 % (1)**	**12.53 % (1)**	26.56 % (2)	21.65 % (40)	24.70 % (1)	**24.22 % (1)**	29.36 % (11)	**24.22 % (1)**
Soy	**5.23 % (2)**	6.56 % (2)	5.90 % (2)	11.17 % (3)	14.04 % (27)	**5.88 % (2)**	8.16 % (2)	22.90 % (30)	**5.88 % (2)**
Thyroid	3.25 % (2)	3.70 % (2)	**2.79 % (2)**	3.27 % (2)	3.70 % (3)	**4.22 % (1)**	4.68 % (1)	6.52 % (2)	4.68 % (1)
Vehicle	20.32 % (3)	**18.45 % (3)**	21.28 % (3)	19.75 % (3)	47.16 % (18)	**21.28 % (3)**	**21.28 % (3)**	56.03 % (15)	22.23 % (3)
Vowel	1.01 % (10)	**0.51 % (10)**	3.43 % (10)	4.85 % (8)	41.92 % (10)	39.29 % (6)	**38.89 % (6)**	49.09 % (10)	39.39 % (6)
Wine	**0.56 % (2)**	**0.56 % (2)**	**0.56 % (2)**	1.11 % (2)	1.64 % (7)	1.67 % (2)	1.70 % (2)	4.45 % (2)	**1.11 % (2)**
Zoo	**30.63 % (6)**	31.74 % (6)	**30.63 % (6)**	39.01 % (12)	49.38 % (15)	**24.88 % (5)**	25.79 % (1)	28.81 % (5)	25.79 % (1)
Rank	**12**	5	6	0	1	**12**	9	2	11

TABLE III: Classification error rates of Linear and Kernel Dimensionality Reduction methods with K Nearest Neighbours Classifier ($K = 3$). In parentheses the dimensionality of the final projected features is shown. The last row shows the number of wins for each method across all datasets.

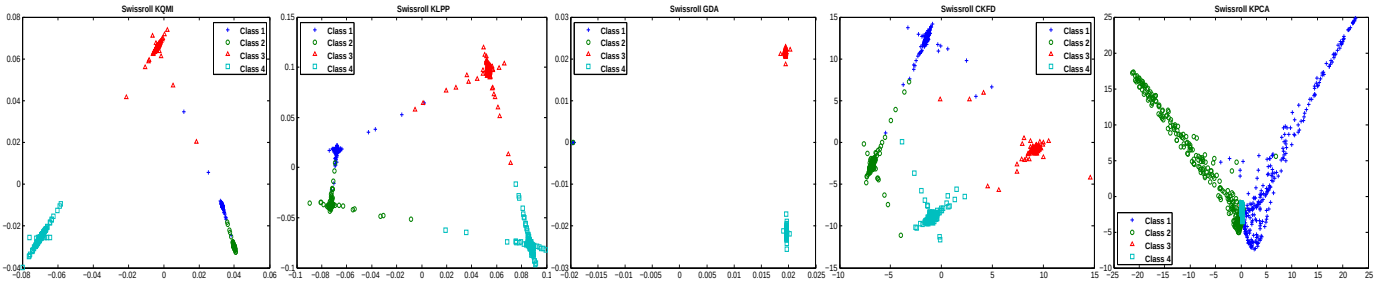
	Kernel Methods					Linear Methods			
Dataset	**KQMI**	**GDA**	**KLPP**	**CKFD**	**KPCA**	**LQMI**	**LDA**	**PCA**	**LPP**
Australian	16.54 % (1)	25.07 % (1)	16.25 % (1)	20.44 % (1)	**15.81 % (13)**	**15.21 % (1)**	**15.21 % (1)**	15.22 % (8)	15.94 % (1)
Balance	**3.20 % (1)**	8.31 % (1)	3.52 % (1)	4.15 % (1)	12.95 % (4)	11.20 % (2)	**10.72 % (2)**	20.79 % (4)	11.52 % (2)
Breast Cancer	3.81 % (1)	4.11 % (1)	3.52 % (1)	3.96 % (1)	**2.78 % (6)**	3.66 % (1)	3.66 % (1)	**2.64 % (5)**	3.66 % (1)
Dermatology	12.84 % (5)	14.80 % (5)	13.95 % (5)	12.33 % (10)	**10.12 % (21)**	3.28 % (5)	3.28 % (5)	**2.44 % (17)**	3.82 % (1)
Diabetes	28.38 % (1)	28.12 % (1)	27.47 % (1)	28.13 % (1)	**25.92 % (8)**	25.77 % (1)	26.82 % (1)	**25.14 % (7)**	27.08 % (1)
Ecoli	18.04 % (7)	20.53 % (7)	18.28 % (7)	18.62 % (7)	**17.21 % (6)**	18.83 % (3)	**18.55 % (5)**	19.67 % (6)	19.68 % (5)
German Numer	30.00 % (1)	**28.30 % (1)**	29.40 % (1)	28.60 % (2)	29.80 % (21)	29.60 % (1)	29.80 % (1)	**28.50 % (21)**	29.30 % (1)
Glass	**31.64 % (5)**	35.26 % (5)	31.69 % (5)	32.58 % (6)	34.49 % (6)	**34.06 % (5)**	36.36 % (3)	**29.92 % (5)**	39.30 % (5)
Heart	**18.89 % (1)**	22.59 % (1)	21.85 % (1)	20.00 % (2)	23.33 % (3)	**18.89 % (1)**	**18.89 % (1)**	20.00 % (6)	**18.89 % (1)**
Ionosphere	9.54 % (1)	6.51 % (1)	5.97 % (1)	7.08 % (2)	**4.81 % (9)**	14.22 % (1)	18.98 % (1)	**10.76 % (11)**	14.78 % (1)
Iris	**2.67 % (2)**	3.33 % (2)	**2.67 % (2)**	4.67 % (2)	4.67 % (3)	**2.00 % (1)**	**2.00 % (1)**	4.67 % (4)	**2.00 % (1)**
Liver Disorders	34.50 % (1)	37.08 % (1)	34.77 % (1)	**33.00 % (1)**	40.82 % (5)	38.62 % (1)	**38.00 % (1)**	35.35 % (6)	38.60 % (1)
Segment	2.90 % (6)	**2.68 % (6)**	2.94 % (10)	3.81 % (6)	4.68 % (19)	**3.03 % (6)**	7.84 % (5)	3.59 % (11)	3.85 % (6)
Sonar	**12.53 % (1)**	**12.53 % (1)**	**12.53 % (1)**	12.56 % (2)	25.49 % (13)	24.72 % (1)	22.29 % (1)	**12.46 % (21)**	22.29 % (1)
Soy	6.55 % (2)	**5.90 % (2)**	8.81 % (2)	7.20 % (3)	10.44 % (28)	**5.23 % (2)**	7.17 % (2)	5.56 % (27)	5.55 % (2)
Thyroid	**1.88 % (2)**	2.32 % (2)	2.79 % (2)	**1.88 % (2)**	2.32 % (3)	3.27 % (2)	3.72 % (2)	**2.81 % (1)**	3.72 % (2)
Vehicle	19.75 % (3)	18.09 % (3)	**17.85 % (3)**	18.34 % (4)	33.80 % (18)	24.02 % (3)	**23.43 (3)**	29.66 % (15)	24.61 % (3)
Vowel	0.61 % (10)	**0.40 % (10)**	2.02 % (10)	2.12 % (9)	4.85 % (9)	3.23 % (9)	3.03 % (10)	**2.83 % (10)**	3.43 % (9)
Wine	1.67 % (2)	**0.56 % (2)**	**0.56 % (2)**	1.11 % (2)	1.67 % (5)	**1.11 % (2)**	**1.11 % (2)**	1.70 % (7)	**1.11 % (2)**
Zoo	**30.63 % (6)**	31.74 % (6)	39.63 % (6)	31.74 % (12)	35.36 % (10)	**24.88 % (5)**	25.79 % (1)	26.90 % (4)	25.79 % (1)
Rank	**7**	6	4	2	6	8	8	**9**	3

TABLE IV: Classification error rates of Linear and Kernel Dimensionality Reduction methods with Linear Kernel SVM Classifier ($C = 100$). In parentheses the dimensionality of the final projected features is shown. The last row shows the number of wins for each method across all datasets.

	Kernel Methods					Linear Methods			
Dataset	**KQMI**	**GDA**	**KLPP**	**CKFD**	**KPCA**	**LQMI**	**LDA**	**PCA**	**LPP**
Australian	15.22 % (1)	24.20 % (1)	16.38 % (1)	20.44 % (1)	**14.78 % (14)**	**14.20 % (1)**	**14.20 % (1)**	14.49 % (4)	**14.20 % (1)**
Balance	**3.68 % (1)**	10.08 % (1)	3.84 % (1)	**3.68 % (1)**	8.95 % (4)	**8.32 % (1)**	**8.32 % (2)**	**8.32 % (4)**	**8.32 % (1)**
Breast Cancer	**3.23 % (1)**	4.10 % (1)	3.66 % (1)	4.10 % (1)	**3.23 % (1)**	3.23 % (1)	3.23 % (1)	**2.64 % (5)**	3.23 % (1)
Dermatology	12.84 % (5)	14.80 % (5)	13.95 % (5)	13.67 % (10)	**11.17 % (33)**	3.28 % (5)	3.27 % (5)	**1.36 % (21)**	3.26 % (5)
Diabetes	22.79 % (1)	31.25 % (1)	26.18 % (1)	27.09 % (1)	**22.40 % (8)**	**23.32 % (1)**	**23.32 % (1)**	23.45 % (7)	**23.32 % (1)**
Ecoli	**19.10 % (7)**	26.06 % (7)	19.36 % (6)	20.50 % (7)	22.10 % (6)	**19.89 % (6)**	20.17 % (6)	20.17 % (7)	20.18 % (6)
German Numer	**27.70 % (1)**	28.30 % (1)	28.00 % (1)	28.20 % (2)	28.90 % (24)	23.10 % (1)	23.10 % (1)	**23.00 % (24)**	23.10 % (1)
Glass	31.69 % (5)	46.89 % (5)	30.80 % (4)	**30.40 % (5)**	35.14 % (7)	**35.99 % (1)**	36.94 % (5)	36.19 % (7)	37.83 % (5)
Heart	**20.00 % (1)**	22.59 % (1)	21.48 % (1)	20.37 % (1)	**20.00 % (6)**	**15.93 % (1)**	**15.93 % (1)**	**15.93 % (12)**	**15.93 % (1)**
Ionosphere	**5.08 % (1)**	6.53 % (1)	5.69 % (1)	6.49 % (1)	6.54 % (28)	13.31 % (1)	35.89 % (1)	**10.22 % (19)**	13.03 % (1)
Iris	**3.33 % (2)**	4.00 % (2)	**3.33 % (2)**	4.00 % (2)	4.00 % (2)	3.33 % (1)	**2.67 % (1)**	4.67 % (4)	3.33 % (1)
Liver Disorders	**27.49 % (1)**	42.01 % (1)	28.64 % (1)	30.66 % (1)	42.02 % (1)	31.33 % (1)	**30.75 % (1)**	30.76 % (6)	31.04 % (1)
Segment	3.16 % (6)	**2.51 % (6)**	3.38 % (6)	3.77 % (6)	7.45 % (13)	5.46 % (6)	85.71 % (1)	**4.29 % (13)**	5.93 % (6)
Sonar	**12.53 % (1)**	**12.53 % (1)**	13.03 % (1)	12.56 % (2)	25.49 % (13)	24.22 % (1)	24.22 % (1)	**19.73 % (1)**	23.75 % (1)
Soy	5.89 % (2)	5.56 % (1)	**4.59 % (2)**	6.22 % (3)	19.56 % (33)	5.88 % (2)	12.02 % (2)	**3.58 % (30)**	5.88 % (2)
Thyroid	**1.86 % (2)**	8.33 % (2)	2.31 % (2)	1.88 % (2)	2.32 % (3)	3.72 % (2)	3.72 % (2)	**2.79 % (1)**	3.27 % (1)
Vehicle	19.85 % (3)	18.45 % (3)	**16.80 % (3)**	18.34 % (4)	33.80 % (18)	21.29 % (3)	21.16 % (3)	**19.40 % (18)**	21.05 % (03)
Vowel	0.61 % (10)	**0.51 % (10)**	2.73 % (10)	2.12 % (9)	4.84 % (9)	19.60 % (10)	19.29 % (10)	**18.88 % (10)**	19.70 % (10)
Wine	1.11 % (2)	**0.56 % (2)**	**0.56 % (2)**	1.11 % (3)	1.64 % (5)	1.70 % (2)	**1.11 (2)**	1.70 % (5)	2.26 % (2)
Zoo	**30.63 % (6)**	31.74 % (6)	**30.63 % (6)**	**30.63 % (12)**	38.15 % (12)	**24.89 % (5)**	25.79 % (1)	26.90 % (4)	25.79 % (3)
Rank	**11**	4	5	3	4	7	7	**12**	4

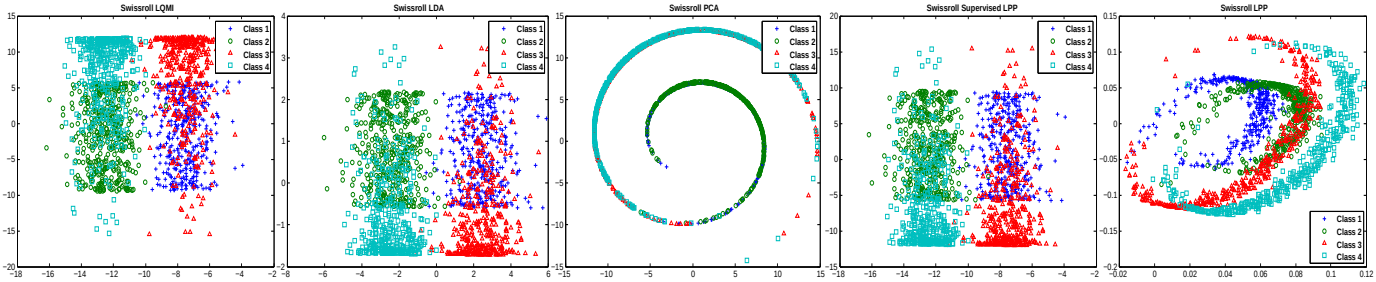
TABLE V: Classification error rates of Linear and Kernel Dimensionality Reduction methods with RBF kernel SVM Classifier ($C = 100$, $\sigma = 1$) In parentheses the dimensionality of the final projected features is shown. The last row shows the number of wins for each method across all datasets.

	Kernel Methods					Linear Methods			
Dataset	**KQMI**	**GDA**	**KLPP**	**CKFD**	**KPCA**	**LQMI**	**LDA**	**PCA**	**LPP**
Australian	**15.08 % (1)**	24.10 % (1)	16.24 % (1)	20.43 % (2)	19.85 % (1)	13.47 % (1)	13.77 % (1)	14.50 % (4)	**13.32 % (1)**
Balance	**3.20 % (1)**	10.08 % (1)	3.67 % (1)	4.48 % (1)	31.37 % (3)	8.32 % (1)	8.32 % (1)	**5.27 % (4)**	8.32 % (1)
Breast Cancer	**3.37 % (1)**	4.11 % (1)	3.66 % (1)	3.96 % (1)	**3.37 % (1**	2.78 % (1)	3.66 % (1)	**2.64 % (5)**	2.78 % (1)
Dermatology	**12.84 % (5)**	14.80 % (5)	13.95 % (5)	21.30 % (5)	16.64 % (4)	**3.28 % (5)**	12.32 % (5)	5.20 % (7)	**3.28 % (5)**
Diabetes	**22.79 % (1)**	28.65 % (1)	26.05 % (1)	27.22 % (1)	33.86 % (8)	23.84 % (1)	**23.57 % (1)**	25.14 % (7)	23.58 % (1)
Ecoli	**19.10 % (7)**	22.16 % (7)	19.93 % (7)	29.04 % (3)	41.15 % (2)	**19.09 % (7)**	26.33 % (6)	19.13 % (7)	19.89 % (7)
German Numer	**27.90 % (1)**	28.30 % (1)	28.70 % (1)	28.50 % (2)	30.30 % (23)	**22.60 % (1)**	22.90 % (1)	28.40 % (21)	**22.60 % (1)**
Glass	32.21 % (5)	32.31 % (5)	**30.35 % (4)**	34.31 % (5)	52.30 % (6)	34.67 % (4)	37.17 % (3)	**26.99 % (5)**	36.94 % (4)
Heart	**19.26 % (1)**	22.59 % (1)	23.33 % (1)	20.00 % (2)	23.33 % (3)	**15.92 % (1)**	18.89 % (1)	20.00 % (6)	16.30 % (1)
Ionosphere	**5.36 % (1)**	6.81 % (1)	5.97 % (1)	7.10 % (2)	22.14 % (3)	13.05 % (1)	35.89 % (1)	**5.38 % (12)**	13.05 % (1)
Iris	**3.33 % (2)**	4.00 % (2)	**3.33 % (2)**	6.67 % (2)	8.67 % (3)	3.33 % (1)	**2.67 % (1)**	5.33 % (4)	3.33 % (1)
Liver Disorders	27.47 % (1)	40.85 % (1)	28.64 % (1)	31.24 % (1)	38.81 % (4)	30.47 % (1)	30.75 % (1)	**29.22 % (6)**	30.47 % (1)
Segment	2.86 % (6)	**2.51 % (6)**	3.33 %(6)	6.23 % (4)	18.18 % (3)	3.59 % (6)	85.71 % (1)	**2.99 % (11)**	5.10 % (6)
Sonar	16.46 % (1)	**12.53 % (1)**	13.03 % (1)	13.01 % (2)	36.57 % (4)	24.22 % (1)	30.81 % (1)	**13.87 % (10)**	23.75 (1)
Soy	5.90 % (2)	6.23 % (2)	**4.92 % (2)**	9.16 % (3)	32.51 % (3)	5.57 % (2)	11.71 % (2)	5.57 % (9)	**5.56 % (2)**
Thyroid	2.32 % (2)	4.18 % (2)	2.79 % (2)	**1.86 % (2)**	7.81 % (2)	3.72 % (2)	3.72 % (2)	**2.79 % (1)**	3.72 % (1)
Vehicle	19.97 % (3)	18.33 % (3)	**16.80 % (3)**	19.62 % (4)	53.80 % (3)	21.76 % (3)	27.34 (3)	**19.87 % (17)**	20.81 % (3)
Vowel	0.61 % (10)	**0.51 % (10)**	2.42 % (10)	6.46 % (5)	55.05 % (3)	1.01 % (10)	2.02 % (8)	**0.40 % (10)**	1.01 % (10)
Wine	1.11 % (2)	**0.56 % (2)**	**0.56 % (2)**	2.22 % (3)	4.38 % (2)	**1.11 % (2)**	**1.11 % (2)**	**1.11 % (13)**	1.67 % (2)
Zoo	**30.63 % (6)**	31.74 % (6)	**30.63 % (6)**	46.10 % (3)	48.63 % (8)	**25.79 % (5)**	**25.79 % (1)**	28.81 % (3)	**25.79 % (2)**
Rank	**11**	4	6	1	1	6	4	**11**	5



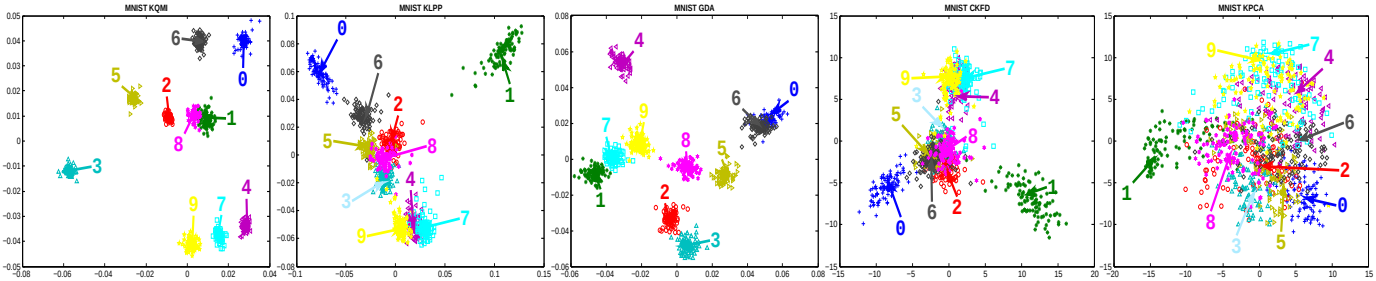
(a) KQMI (b) KLPP (c) GDA (d) CKFD (e) KPCA

Fig. 1: 2D Projections of the Swiss Role dataset using Kernel Dimensionality Reduction Methods.



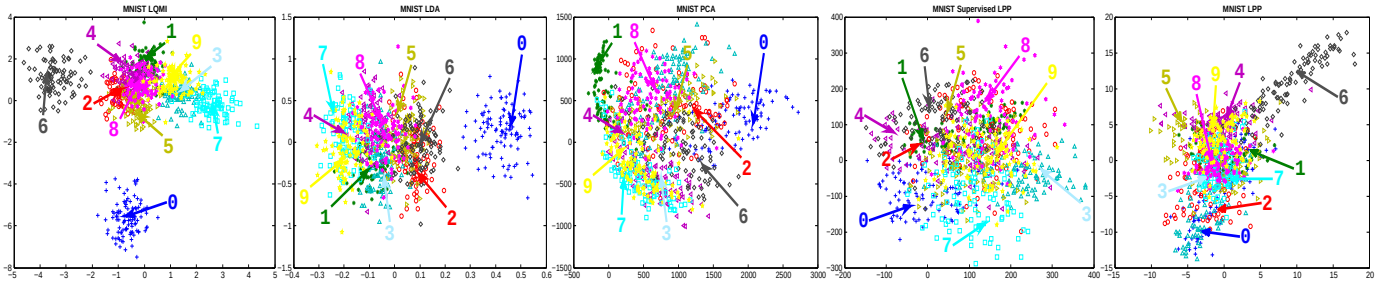
(a) LQMI (b) LDA (c) PCA (d) Supervised LPP (e) LPP

Fig. 2: 2D Projections of the Swiss Role dataset using Linear Dimensionality Reduction Methods.



(a) KQMI (b) KLPP (c) GDA (d) CKFD (e) KPCA

Fig. 3: 2D Projections of the MNIST dataset using Kernel Dimensionality Reduction Methods.



(a) LQMI (b) LDA (c) PCA (d) Supervised LPP (e) LPP

Fig. 4: 2D Projections of the MNIST dataset using Linear Dimensionality Reduction Methods.

TABLE I: Benchmark Datasets' Characteristics.

Dataset	Library	Samples	Attributes	Classes
Australian	Statlog	690	14	2
Balance	UCI	625	4	3
Breast Cancer	UCI	683	10	2
Dermatology	UCI	366	34	6
Diabetes	UCI	768	8	2
Ecoli	UCI	336	7	8
German Numer	Statlog	1000	24	2
Glass	UCI	214	9	6
Heart	Statlog	270	13	2
Ionosphere	UCI	351	34	2
Iris	UCI	150	4	3
Liver Disorders	UCI	214	9	7
Segment	Statlog	2310	19	7
Sonar	UCI	208	60	2
Soy	UCI	307	35	3
Thyroid	UCI	215	5	3
Vehicle	Statlog	846	18	4
Vowel	UCI	990	10	11
Wine	UCI	178	13	3
Zoo	UCI	101	17	7

classifier can be seen in Table II, those of the KNN classifier can be seen in Table III, for the Linear SVM classifier in Table IV and for the RBF SVM classifier in Table V. In each column we illustrate the minimum classification error attained by each classifier for the specific dimensionality reduction method and in the parentheses the number of dimensions this error has been achieved. In the last row of each one of these tables we can see the rankings (i.e., the number of winning datasets) of each dimensionality reduction method. In the case of the NCC classifier (i.e., Table II) we can see that KQMI obtains better performance than the other kernel methods. In the case of the linear projection algorithms the LQMI method displays comparable results with the supervised LPP method while both methods seem to outperform LDA. For the KNN classifier (i.e., Table III) we can see that KQMI and GDA display almost the same performance, obtaining though better results than CKFD and KLPP. It is interesting to note here the relatively good performance of the KPCA method, which for the KNN classifier wins the same number of datasets as the supervised GDA method. For the same classifier and for the linear projection algorithms the LQMI and LDA exhibit the same performance winning however less sets than the unsupervised PCA. These poor results for the linear supervised methods can be mainly attributed to the inadequate number of samples per class, or due to the fact that the training data might non-uniformly sample the underlying distribution [24]. Moreover, PCA achieves, in most of the cases, better performance in higher dimensions than the supervised methods which are constrained, due to rank deficiency issues, to less than $C-1$ dimensions. Another reason for these good results for the unsupervised PCA method might be the *Vapnik-Chervonenkis dimension (VC)* [34] of the classifier in use. That is, the powerful KNN and RBF SVM classifiers with high VC dimension seem to favour more the unsupervised PCA than the other supervised methods. This does not hold for the weaker NCC classifier which seems to favour more the supervised approaches. In Tables IV and V we can see that for the kernel methods KQMI becomes the dominant method when the SVM classifier is used. For the same classifier but for the linear methods we can see again the previously mentioned superiority of the PCA method against the supervised ones.

B. Visualization

We also conducted visualization experiments where we used the artificially created swissroll dataset and a subset of 1000 samples from the MNIST handwritten digits dataset. The swissroll dataset was created to test out various dimensionality reduction algorithms. The idea that lies behind the creation of this dataset is to create several points in $\mathbb{R}^2$, and then map them to $\mathbb{R}^3$ with some smooth function. The resulting 3D dataset can then be used to test how well a dimensionality method maps the 3D manifold back to the 2D space. Here, our purpose is to visualize how well the classes of each dataset are separated in the 2D projective space. For the swissroll the 2D projections of the kernel dimensionality reduction methods (i.e., KQMI, GDA, supervised KLPP, CKFD and KPCA) can be seen in Figure 1 and those of the linear dimensionality reduction methods (i.e., LQMI, LDA, PCA and supervised LPP) can be seen in Figure 2.

Comparing Figures 1 and 2 we observe that the linear methods fail to achieve a mapping in which the dataset's classes are well separated in the 2D space, while the supervised kernel methods (i.e., KQMI, GDA, KLPP and CKFD) give a relatively good mapping in terms of class separability. In this setting GDA performs best with KQMI and KLPP giving similar results, even though KQMI produces more compact clusters with less linearly non-separable points than KLPP. Moreover, from the above figures we can verify that the

unsupervised methods, both kernel and linear, fail to give a good mapping in which the classes are well separated, a fact to be expected due to their unsupervised nature.

In Figures 3 and 4 we can see the 2D visualization of the MNIST dataset with the kernel and linear reduction methods respectively. We can see how nicely the KQMI and GDA methods produce a 2D projection subspace in which the classes are very well separated, with the KQMI attaining a better separation of the classes than that of the GDA. It could be counter-intuitive the fact that the distance between digits 1 and 8 is much smaller than that between 1 and 7. However, let us note here that the objective function takes into account all the classes simultaneously and thus, if a specific projection worsens the separability between two specific classes whereas it enhances the separability among all the other classes, it is expected to be selected as an optimal projection. In the specific case we can see that indeed the separability of all the classes is enhanced compared to the one obtained with the KLPP projections except the separability between digits 1 and 8 which is worse. That is, the separability between 1 and 8 is the prize to be payed in order to enhance the other 44 pairwise separabilities. The situation changes in the cases of the CKFD and KLPP methods where as we can see although we can moderately discriminate relatively good the classes, many of them seem to coincide. All the linear methods fail to give a 2D projection where the classes are well separated. However, we can mention here that LQMI gives fairly better 2D projection in terms of class discrimination than all the other linear methods.

C. Face Recognition and Facial Expression Recognition

We also tested our methods on two face recognition (i.e., ORL and YALE) and two facial expression recognition datasets (i.e., JAFFE and KANADE). The characteristics of each dataset can be seen in Table VI. We normalized all the feature vectors of each dataset to unit length and for the linear methods only, we preprocessed the datasets with PCA in order to hold 99 % of the initial dataset variance. As can be seen in Table VI, for all the datasets the number of the dataset samples n is less than the number of dataset attributes m. This situation is known as the *under-sampled size* problem. For datasets where the under-sampled size problem exists, in many cases occur singularities that in turn result in very bad performance of the eigenvalue analysis algorithm. One of the solutions to overcome the under-sampled size problem is to perform an initial PCA step on the data and fall to the dimension where its associated eigenvalue is greater than some threshold [37]. For the evaluation of the generalization error we used the same procedure as in IV-A. In the sequel we give the characteristics of the real-life datasets used in our evaluation.

TABLE VI: Face Recognition and Facial Expression Recognition Datasets' Characteristics.

Dataset	Samples	Attributes	Classes
ORL	400	1024	40
YALE	165	1024	15
JAFFE	213	1200	7
KANADE	704	1200	7

ORL [28]: The dataset contains 40 individuals and 10 different images for each individual, including variations in facial expression and pose. In Figure 7 we can see a male and a female subject from the ORL dataset.

Fig. 7: Two subjects from the ORL Face Recognition dataset.

YALE [2]: The dataset contains 165 gray-scale images of 15 individuals. There are 11 images per subject, one per different facial expression or configuration: center-light, with glasses, happy, left-light, without glasses, normal, right-light, sad, sleepy, surprised, and wink. Figure 8 shows a subject from the YALE dataset.

Fig. 8: A subject from the YALE Face Recognition dataset.

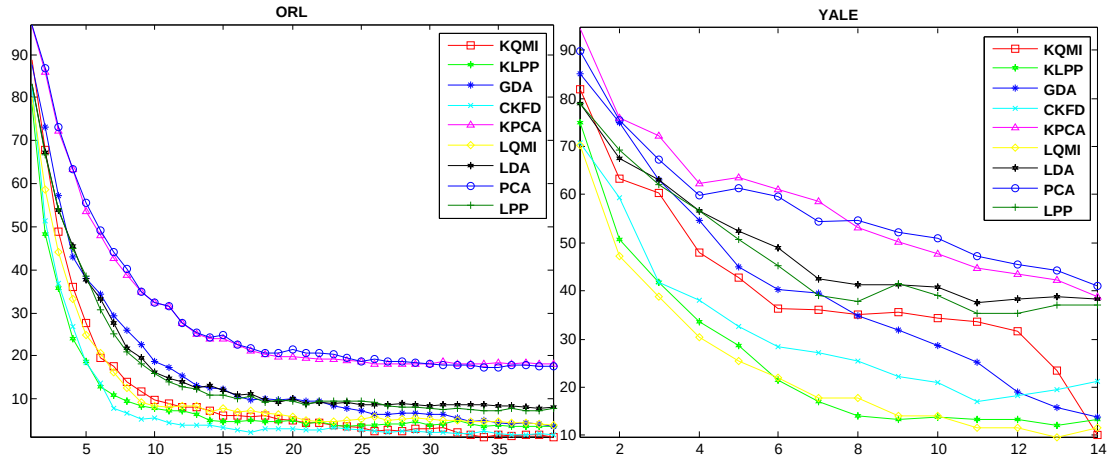
JAFFE [23]: The dataset contains 213 images of 7 facial expressions (6 basic facial expressions + 1 neutral) posed by 10 Japanese female models. Each image has been labeled with one of 6 emotion adjectives (i.e., fear, anger, disgust, happiness, surprise, sadness). In Figure 9 we see a subject from the JAFFE dataset.

Fig. 9: A subject from the JAFFE Facial Expression Recognition dataset.

KANADE [19]: The dataset contains 704 images of 7 facial expressions (6 basic facial expressions + 1 neutral). Each image has been rated on 6 emotion adjectives (i.e., fear, anger, disgust, happiness, surprise, sadness). The variety of subjects covers different races, ages and genders. The database is collected under controlled illumination and background. In Figure 10 various subjects from the KANADE dataset are given.

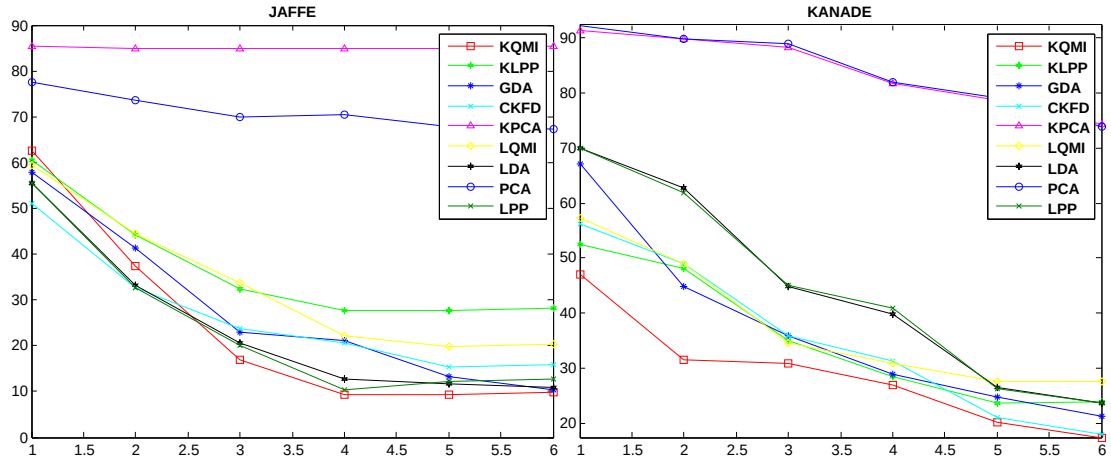
Fig. 10: Various subjects from the KANADE Facial Expression Recognition dataset.

In Tables VII and VIII we show the classification errors on the above mentioned datasets attained by the various dimensionality reduction methods using the NCC and KNN classifiers respectively. We also conducted experiments using as classifiers the Linear and RBF SVM. However, for these



(a) ORL (b) YALE

Fig. 5: Results on ORL and YALE Face Recognition Datasets. The horizontal axis shows the dimension of the projected features for several dimensions. The vertical axis shows the corresponding classification error rate.



(a) JAFFE (b) KANADE

Fig. 6: Results on JAFFE and KANADE Facial Expression Recognition Datasets. The horizontal axis shows the dimension of the projected features for several dimensions. The vertical axis shows the corresponding classification error rate.

TABLE VII: Classification error rates of Linear and Kernel Dimensionality Reduction methods with Nearest Class Centroid Classifier. In parentheses the dimensionality of the final projected features is shown.

	Kernel Methods					**Linear Methods**			
Dataset	**KQMI**	**GDA**	**KLPP**	**CKFD**	**KPCA**	**LQMI**	**LDA**	**PCA**	**LPP**
ORL	**1.25 % (34)**	3.75 % (39)	3.50 % (24)	1.75 % (36)	18.00 % (28)	**4.00 % (39)**	7.75 % (38)	14.00 % (142)	7.25 % (34)
YALE	**10.22 % (14)**	12.14 % (14)	13.39 % (14)	17.01 % (11)	27.71 % (62)	**9.74 % (13)**	37.65 % (11)	29.63 % (67)	35.34 % (12)
JAFFE	**9.29 % (4)**	10.37 % (6)	27.78 % (4)	15.34 % (5)	48.48 % (148)	19.76% (5)	10.81 % (6)	52.20 % (93)	**10.31 % (4)**
KANADE	**17.46 % (6)**	21.27 % (6)	23.73 % (5)	17.88 % (12)	48.30 % (130)	24.67 % (6)	**23.69 % (6)**	46.02 % (175)	23.71 % (6)

TABLE VIII: Classification error rates of Linear and Kernel Dimensionality Reduction methods with K Nearest Neighbors Classifier ($K = 3$). In parentheses the dimensionality of the final projected features is shown.

	Kernel Methods					**Linear Methods**			
Dataset	**KQMI**	**GDA**	**KLPP**	**CKFD**	**KPCA**	**LQMI**	**LDA**	**PCA**	**LPP**
ORL	**1.50 % (38)**	3.75 % (39)	1.75 % (34)	**1.50 % (36)**	10.00 % (45)	**6.50 % (35)**	8.25 % (26)	8.50 % (242)	**6.50 % (34)**
YALE	12.55 % (14)	13.91 % (14)	**9.16 % (14)**	17.12 % (24)	35.52 % (32)	35.95 % (13)	37.72 % (14)	20.07 % (102)	**11.24 % (4)**
JAFFE	**8.92 % (6)**	10.37 % (6)	15.01 % (5)	13.95 % (12)	15.07 % (86)	11.78 % (13)	**10.81 % (6)**	15.03 % (14)	11.24 % (4)
KANADE	**17.32 % (6)**	21.27 % (6)	20.46 % (6)	17.76 % (12)	42.03 % (17)	**23.98 % (6)**	24.69 % (6)	42.32 % (18)	25.12 % (6)

classifiers the dominant method was PCA, and all the remaining methods attained equivalent performance. Therefore, we omit the results of these experiments.

In Figures 5a and 5b we illustrate the diagrams of the classification error using the NCC classifier in the dimensions $1-(C-1)$ for the ORL and YALE datasets respectively. In Figures 6a and 6b we can see the resulting classification errors using the NCC classifier on the JAFFE and KANADE datasets attained in the dimensions $1-(C-1)$ as well.

From the results in Table VII we can see that, in the case of the NCC classifier, KQMI obtains better performance in most of the datasets than all the other kernel methods. In the linear case for the NCC classifier, the LQMI wins more datasets than the other methods. For the KNN classifier in Table VIII, we can see that both KQMI and LQMI exhibit slightly worse performance than that attained for the NCC classifier, winning however most of the datasets.

In Figures 5 and 6 we also see that our proposed methods, in most of the cases, converge faster to their best performance compared to the other methods and hence, they display better performance in dimensions lower than $C-1$.

V. Conclusion

In this paper we proposed a novel supervised dimensionality reduction method based on the maximization of a non-parametric Mutual Information criterion between the feature vectors and their respective class labels. We formulated the Quadratic Mutual Information as a kernel objective function that can be directly optimized inside the Graph Embedding Framework. We also derived the linear equivalent of this kernel method and we compared both methods to several state-of-the-art kernel and linear dimensionality reduction methods. From the experimental results we can conclude that the proposed methods obtain comparable and in most cases even better classification accuracy than the state-of-the-art.

Appendix A
Mutual Information Terms

In this Appendix we derive analytically the expressions for the V_{ALL}, V_{IN} and V_{BTW} terms. We have

$$\begin{aligned}V_{ALL} &= \frac{1}{n^2}\left(\sum_{c=1}^{C}\left(\frac{J_c}{n}\right)^2\right)\sum_{j=1}^{n}\sum_{k=1}^{n}\mathbf{K}_{jk} = \left(\frac{1}{n^4}\sum_{c=1}^{C}(J_c)^2\right)\mathbf{1}^\top\mathbf{K}\mathbf{1}\\ &= C_{ALL}\mathbf{1}^\top\mathbf{K}\mathbf{1} = C_{ALL}\mathrm{tr}\{\mathbf{K}\mathbf{1}\mathbf{1}^\top\} = C_{ALL}\mathrm{tr}\{\mathbf{\Phi}\mathbf{\Phi}^\top\mathbf{1}\mathbf{1}^\top\}\\ &= C_{ALL}\mathrm{tr}\{\mathbf{\Phi}^\top\mathbf{1}\mathbf{1}^\top\mathbf{\Phi}\},\end{aligned}$$

$$\begin{aligned}V_{IN} &= \frac{1}{n^2}\sum_{c=1}^{C}\sum_{j,y_j=c}\sum_{k,y_k=c}\mathbf{K}_{jk} = C_{IN}\sum_{c=1}^{C}\mathbf{1}_c^\top\mathbf{K}\mathbf{1}_c\\ &= C_{IN}\sum_{c=1}^{C}\mathrm{tr}\{\mathbf{K}\mathbf{1}_c\mathbf{1}_c^\top\} = C_{IN}\sum_{c=1}^{C}\mathrm{tr}\{\mathbf{\Phi}\mathbf{\Phi}^\top\mathbf{1}_c\mathbf{1}_c^\top\}\\ &= C_{IN}\mathrm{tr}\{\sum_{c=1}^{C}\mathbf{\Phi}\mathbf{\Phi}^\top\mathbf{1}_c\mathbf{1}_c^\top\} = C_{IN}\mathrm{tr}\{\mathbf{\Phi}\mathbf{\Phi}^\top\left(\sum_{c=1}^{C}\mathbf{1}_c\mathbf{1}_c^\top\right)\}\\ &= C_{IN}\mathrm{tr}\{\mathbf{\Phi}^\top\left(\sum_{c=1}^{C}\mathbf{1}_c\mathbf{1}_c^\top\right)\mathbf{\Phi}\},\end{aligned}$$

$$\begin{aligned}V_{BTW} &= \frac{1}{n^3}\sum_{c=1}^{C}J_c\sum_{k,y_k=c}\sum_{j=1}^{n}\mathbf{K}_{kj} = \sum_{c=1}^{C}C_{BTW,c}\mathbf{1}_c^\top\mathbf{K}\mathbf{1}\\ &= \sum_{c=1}^{C}\mathrm{tr}\{C_{BTW,c}\mathbf{K}\mathbf{1}\mathbf{1}_c^\top\} = \sum_{c=1}^{C}\mathrm{tr}\{C_{BTW,c}\mathbf{\Phi}\mathbf{\Phi}^\top\mathbf{1}\mathbf{1}_c^\top\}\\ &= \mathrm{tr}\{\sum_{c=1}^{C}C_{BTW,c}\mathbf{\Phi}\mathbf{\Phi}^\top\mathbf{1}\mathbf{1}_c^\top\} = \mathrm{tr}\{\mathbf{\Phi}\mathbf{\Phi}^\top\mathbf{1}\left(\sum_{c=1}^{C}C_{BTW,c}\mathbf{1}_c^\top\right)\}\\ &= \mathrm{tr}\{\mathbf{\Phi}^\top\mathbf{1}\left(\sum_{c=1}^{C}C_{BTW,c}\mathbf{1}_c^\top\right)\mathbf{\Phi}\}.\end{aligned}$$

Appendix B
QMI Graph Formulation

In this Appendix we show that $\sum_{j=1}^{n}W_{ij} = D_{ii},\ i = 1,\ldots,n$ for the QMI graph.

$$\begin{aligned}&(1-J_c)\alpha_c - \sum_{\substack{c'=1,\\ c'\neq c}}^{C}\beta_{c,c'}J_{c'} = (1-J_c)(C_{IN}+C_{ALL}-2C_{BTW,c})-\\ &\quad-\sum_{\substack{c'=1,\\ c'\neq c}}^{C}(C_{ALL}-C_{BTW,c}-C_{BTW,c'})J_{c'}\\ &= \alpha_c - J_cC_{IN} - J_cC_{ALL} + 2J_cC_{BTW,c}-\\ &\quad-\sum_{\substack{c'=1,\\ c'\neq c}}^{C}C_{ALL}J_{c'} + \sum_{\substack{c'=1,\\ c'\neq c}}^{C}C_{BTW,c}J_{c'} + \sum_{\substack{c'=1,\\ c'\neq c}}^{C}C_{BTW,c'}J_{c'}\\ &= \alpha_c - \frac{J_c}{n^2} - J_cC_{ALL} + 2\frac{J_c^2}{n^3} - C_{ALL}(n-J_c)+\\ &\quad + C_{BTW,c}(n-J_c) + \sum_{\substack{c'=1,\\ c'\neq c}}^{C}\frac{J_{c'}^2}{n^3}\\ &= \alpha_c - \frac{J_c}{n^2} - J_cC_{ALL} + 2\frac{J_c^2}{n^3} - nC_{ALL} + J_cC_{ALL}+\\ &\quad + nC_{BTW,c} - J_cC_{BTW,c} + \sum_{\substack{c'=1,\\ c'\neq c}}^{n_c}\frac{J_{c'}^2}{n^3}\\ &= \alpha_c - \frac{J_c}{n^2} - J_cC_{ALL} + 2\frac{J_c^2}{n^3} - nC_{ALL} + J_cC_{ALL}+\\ &\quad + \frac{J_c}{n^2} - \frac{J_c^2}{n^3} + \sum_{\substack{c'=1,\\ c'\neq c}}^{C}\frac{J_{c'}^2}{n^3}\\ &= \alpha_c + \frac{J_c^2}{n^3} - nC_{ALL} + \sum_{\substack{c'=1,\\ c'\neq c}}^{C}\frac{J_{c'}^2}{n^3}\\ &= \alpha_c - nC_{ALL} + \sum_{c'=1}^{C}\frac{J_{c'}^2}{n^3} = \alpha_c - nC_{ALL} + nC_{ALL} = \alpha_c.\end{aligned}$$

References

[1] K. Bache and M. Lichman. UCI machine learning repository, 2013.

[2] G. Baudat and F. Anouar. Generalized Discriminant Analysis Using a Kernel Approach. *Neural Computation*, 12(10):2385–2404, 2000.

[3] P.N. Belhumeur, J.P. Hespanha, and D.J. Kriegman. Eigenfaces vs. Fisherfaces: recognition using class specific linear projection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 19(7):711–720, jul 1997.

[4] Mikhail Belkin and Partha Niyogi. Laplacian Eigenmaps for Dimensionality Reduction and Data Representation. *Neural Computation*, 15(6):1373–1396, jun 2003.

[5] Mikio L. Braun, Joachim Buhmann, and Klaus-Robert Müller. Denoising and Dimension Reduction in feature space. In B. Schölkopf, J. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems 19*, pages 185–192, Cambridge, MA, 2007. MIT Press.

[6] Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A library for support vector machines, 2011. Software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm.

[7] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20:273–297, 1995.

[8] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. Wiley Series in Telecommunications and Signal Processing. Wiley-Interscience, 2nd edition, jul. 2006.

[9] R. M. Fano. *Transmission of Information: A Statistical theory of Communications*. MIT Press, 1961.

[10] Sir Ronald Aylmer Fisher. The Use of Multiple Measurements in Taxonomic Problems. *Annals of Eugenics*, (7):179–188, 1936.

[11] Kenji Fukumizu, Francis R. Bach, and Michael I. Jordan. Dimensionality Reduction for Supervised Learning with Reproducing Kernel Hilbert Spaces. *Journal of Machine Learning Research*, (5):73–99, jan 2004.

[12] Keinosuke Fukunaga. *Introduction to Statistical Pattern Recognition*. Academic Press, 2nd edition, oct 1990.

[13] Xuan Guorong, Chai Peiqi, and Wu Minhui. Bhattacharyya distance feature selection. In *Pattern Recognition, 1996., Proceedings of the 13th International Conference on*, volume 2, pages 195–199 vol.2, 1996.

[14] S.W. Hadley, F. Rendl, and H. Wolkowicz. Symmetrization of Nonsymmetric Quadratic Assignment Problems and the Hoffman-Wielandt Inequality. Technical report, Sixth Haifa Conference on Matrix Theory, 1996.

[15] Xiaofei He and Partha Niyogi. Locality Preserving Projections. In Sebastian Thrun, Lawrence Saul, and Bernhard Schölkopf, editors, *Advances in Neural Information Processing Systems 16*, Cambridge, MA, 2004. MIT Press.

[16] M.E. Hellman and J. Raviv. Probability of error, equivocation, and the Chernoff bound. *Information Theory, IEEE Transactions on*, 16(4):368–372, 1970.

[17] Thomas Hofmann, Bernhard Schölkopf, and Alexander J. Smola. Kernel methods in machine learning. *Annals of Statistics*, 36(3):1171–1220, jun 2008.

[18] I. Jolliffe. *Principal Component Analysis.* Encyclopedia of Statistics in Behavioral Science. Ltd John Wiley & Sons, oct. 2005.

[19] Takeo Kanade, Yingli Tian, and Jeffrey F. Cohn. Comprehensive Database for Facial Expression Analysis. *Automatic Face and Gesture Recognition, IEEE International Conference on*, 0:46, 2000.

[20] J. Kapur. *Measures of Information and their Applications*. Wiley, 1994.

[21] J. Kapur and H. Kesavan. *Entropy Optimization principles with Applications*. Academic Press, 1992.

[22] E. Kokiopoulou and Y. Saad. Orthogonal Neighborhood Preserving Projections: A Projection-Based Dimensionality Reduction Technique. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 29(12):2143–2156, 2007.

[23] Michael J. Lyons, Shigeru Akamatsu, Miyuki Kamachi, and Jiro Gyoba. Coding Facial Expressions with Gabor Wavelets. In *Proceedings, Third IEEE International Conference on Automatic Face and Gesture Recognition*, Nara Japan, IEEE Computer Society, pages 14–16, apr 1998.

[24] A.M. Martinez and A.C. Kak. PCA versus LDA. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 23(2):228–233, feb 2001.

[25] D. Michie, D.J. Spiegelhalter, and C.C. Taylor, editors. *Machine Learning, Neural and Statistical Classication*. Ellis Horwood Series in Artificial Intelligence. Prentice Hall, 1994.

[26] S. Mika, G. Ratsch, J. Weston, B. Schölkopf, and K.R. Müller. Fisher discriminant analysis with kernels. In *Neural Networks for Signal Processing IX, 1999. Proceedings of the 1999 IEEE Signal Processing Society Workshop*, pages 41–48, aug. 1999.

[27] Sam T. Roweis and Lawrence K. Saul. Nonlinear Dimensionality Reduction by Locally Linear Embedding. *Science*, 290(5500):2323–2326, 2000.

[28] Ferdinando Samaria and Andy Harter. Parameterisation of a Stochastic Model for Human face Identification. In *Proceedings of 2nd IEEE Workshop on Applications of Computer Vision, Sarasota FL*, dec 1994.

[29] George Saon and Mukund Padmanabhan. Minimum Bayes Error Feature Selection for Continuous Speech Recognition. In *Advances in Neural Information Processing Systems 13*, pages 800–806. MIT Press, 2001.

[30] Bernhard Schölkopf, Alexander Smola, and Klaus-Robert Müller. Nonlinear Component Analysis as a Kernel Eigenvalue Problem. *Neural Computation*, 10(5):1299–1319, 1998.

[31] Bernhard Schölkopf and Alexander J. Smola. *Learning With Kernels Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, dec. 2001.

[32] Joshua B. Tenenbaum, Vin de Silva, and John C. Langford. A Global Geometric Framework for Nonlinear Dimensionality Reduction. *Science*, 290(5500):2319–2323, 2000.

[33] Kari Torkkola. Feature Extraction by Non-Parametric Mutual Information Maximization. *Journal of Machine Learning Research*, 3:1415–1438, March 2003.

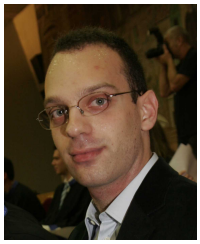
[34] Vladimir N. Vapnik. *Statistical Learning Theory*. Wiley-Interscience, 1998.

[35] Shuicheng Yan, Dong Xu, Benyu Zhang, Hong-Jiang Zhang, Qiang Yang, and S. Lin. Graph Embedding and Extensions: A General Framework for Dimensionality Reduction. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 29(1):40–51, jan. 2007.

[36] Jian Yang, A.F. Frangi, Jing-Yu Yang, David Zhang, and Zhong Jin. KPCA plus LDA: a complete kernel Fisher discriminant framework for feature extraction and recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 27(2):230–244, feb. 2005.

[37] Jian Yang and Jing yu Yang. Why can LDA be performed in PCA transformed space? *Pattern Recognition*, 36(2):563–566, 2003.

Dimitrios Bouzas obtained his BSc in informatics in 2010 and his MSc in digital media in 2012 with highest honors from Aristotle University of Thessaloniki, Greece. He is currently working as a researcher and developer at the geometry division of Beta CAE systems S.A. His research interests include Machine Learning, Computational Geometry, Computer Graphics, Numerical Analysis, Mesh Generation Algorithms, Computational Fluid Dynamics, Stress Analysis and Computer Aided Engineering.



Nikolaos Arvanitopoulos obtained his BSc in informatics from Aristotle University of Thessaloniki, Greece in 2009 and his MSc in visual computing from Saarland University, Saabrucken, Germany in 2011. He is currently pursuing his PhD studies in the Image and Visual Representation Group in the School of Computer and Communication Sciences at EPFL. His research interests include Machine Learning, Image Processing and Handwriting Recognition. He is a student member of the IEEE.

Anastasios Tefas received the B.Sc. in informatics in 1997 and the Ph.D. degree in informatics in 2002, both from the Aristotle University of Thessaloniki, Greece. Since 2013 he has been an Assistant Professor at the Department of Informatics, Aristotle University of Thessaloniki. From 2008 to 2012, he was a Lecturer at the same University. From 2006 to 2008, he was an Assistant Professor at the Department of Information Management, Technological Institute of Kavala. From 2003 to 2004, he was a temporary lecturer in the Department of Informatics, University of Thessaloniki. From 1997 to 2002, he was a researcher and teaching assistant in the Department of Informatics, University of Thessaloniki. Dr. Tefas participated in 12 research projects financed by national and European funds. He has co-authored 42 journal papers, 122 papers in international conferences and contributed 7 chapters to edited books in his area of expertise. Over 2220 citations have been recorded to his publications and his H-index is 24 according to Google scholar. His current research interests include computational intelligence, pattern recognition, statistical machine learning, digital signal and image processing and computer vision.