A Least-Squares Framework for Component Analysis

Fernando De la Torre *Member, IEEE,*

***Abstract*— Over the last century, Component Analysis (CA) methods such as Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA), Canonical Correlation Analysis (CCA), Laplacian Eigenmaps (LE), and Spectral Clustering (SC) have been extensively used as a feature extraction step for modeling, clustering, classification, and visualization. CA techniques are appealing because many can be formulated as eigen-problems, offering great potential for learning linear and non-linear representations of data in closed-form. However, the eigen-formulation often conceals important analytic and computational drawbacks of CA techniques, such as solving generalized eigen-problems with rank deficient matrices (e.g., small sample size problem), lacking intuitive interpretation of normalization factors, and understanding commonalities and differences between CA methods.**

This paper proposes a unified least-squares framework to formulate many CA methods. We show how PCA, LDA, CCA, LE, SC, and their kernel and regularized extensions, correspond to a particular instance of least-squares weighted kernel reduced rank regression (LS-WKRRR). The LS-WKRRR formulation of CA methods has several benefits: (1) provides a clean connection between many CA techniques and an intuitive framework to understand normalization factors; (2) yields efficient numerical schemes to solve CA techniques; (3) overcomes the small sample size problem; (4) provides a framework to easily extend CA methods. We derive new weighted generalizations of PCA, LDA, CCA and SC, and several novel CA techniques.

***Index Terms*— Principal Component Analysis, Linear Discriminant Analysis, Canonical Correlation Analysis, k-means, Spectral Clustering, Reduced Rank Regression, Kernel Methods and Dimensionality Reduction.**

I. Introduction

Over the last century, Component Analysis (CA) methods [1] such as Principal Component Analysis (PCA) [2], [3], Linear Discriminant Analysis (LDA) [4], [5], Canonical Correlation Analysis (CCA) [6], Laplacian Eigenmaps (LE) [7], Locality Preserving Projections (LPP) [8], and Spectral Clustering (SC) [9] have been extensively used as a feature extraction step for modeling, classification, visualization and clustering problems. The aim of CA techniques is to decompose a signal into *relevant* components that are optimal for a given task (e.g., classification, visualization). These components, explicitly or implicitly (e.g., kernel methods), define the representation of the signal. CA techniques are appealing for two main reasons. Firstly, CA models typically have a small number of parameters, and therefore can be estimated using relatively few samples. CA techniques are especially useful to model high-dimensional data, because due to the *curse-of-dimensionality* learning models typically requires a large number of samples. Secondly, many CA techniques can be formulated as eigen-problems, offering great potential for efficient learning of linear and non-linear models without local minima. The use of eigen-solvers to address statistical problems dates back to the 1930s, and since then many numerically stable and efficient packages have been developed to solve eigen-problems. For these reasons, during the last century, many computer vision, computer graphics, signal processing, and statistical problems were framed as learning a low dimensional CA model.

Although CA methods have been widely used in many scientific disciplines, there is still a need for a better mathematical framework than the eigen-formulation to analyze and extend CA techniques. The least-squares unified framework proposed in this paper provides a tool for analyzing, generalizing, and developing efficient algorithms to solve many CA methods. This paper shows how Kernel PCA (KPCA), Kernel LDA (KLDA), Kernel CCA (KCCA), Normalized Cuts (Ncuts), and LE correspond to a particular instance of a least-squares weighted kernel reduced rank regression (LS-WKRRR) problem. This framework should provide researchers with a thorough understanding of a large number of existing CA techniques, and it may serve as a tool for dealing with novel CA problems as they arise. Preliminary versions of this work were published at [10], [11].

This paper recovers the spirit of three previously published papers seeking unified frameworks. Borga [12] showed how PCA, Partial Least Squares, CCA and Multiple Linear Regression can be formulated as generalized eigen-value problems (GEPs). To efficiently solve the GEP for high-dimensional data, Borga proposed to use a gradient-descent algorithm on a Rayleigh quotient. Roweis and Ghahramani [13] showed how a Linear Dynamical System (LDS) is the generative model for Hidden Markov Models, Kalman Filter, vector quantization, Factor Analysis, and mixture of Gaussians. By introducing non-linearities into the model, [13] demonstrated how Independent Component Analysis can also be cast as an extension of a LDS. Yan *et al.* [14] have proposed a unifying view of PCA, LPP, Isomap, and LDA using a graph theoretical formulation. Additionally, the authors proposed Marginal Fisher Analysis, a variant of non-parametric LDA [15].

This paper differs from previous research in that it unifies PCA, CCA, LDA, SC, LE, and their kernel and regularized extensions using the LS-WKRRR model. Moreover, we show that several extensions of the LS-WKRRR derive into novel techniques such as Dynamic Coupled Component Analysis (DCCA), Aligned Cluster Analysis (ACA), Canonical Time Warping (CTW), Filtered Component Analysis (FCA), Parameterized Kernel Principal Component Analysis (PaKPCA), Feature Selection for Subspace Analysis (FSSA) and Discriminative Cluster Analysis (DCA). In addition, we propose new weighted extensions for PCA, LDA, CCA, and SC.

The rest of the paper is organized as follows: Section II introduces the notation. Section III describes the LS-WKRRR

Manuscript received June 03, 2009;

F. De la Torre is with the Robotics Institute at Carnegie Mellon University. 211 Smith Hall Robotics Institute. Pittsburgh, PA 15213. E-mail: ftorre@cs.cmu.edu

problem and derives the coupled generalized eigenvalue system of equations that results from solving it. Section IV relates PCA, KPCA and weighted extensions to the LS-WKRRR. Section V shows how LDA, KLDA, CCA, KCCA and weighted extensions are a particular instance of LS-WKRRR. Section VI connects LS-WKRRR to non-linear embedding methods. Section VII shows the relationship between LS-WKRRR, k-means and SC. Section VIII describes extensions of CA methods derived from the LS-WKRRR framework. Section IX finalizes the paper with the conclusions.

II. Notation

Bold capital letters denote matrices (e.g., $\mathbf{D}$), bold lower-case letters represent column vectors (e.g., $\mathbf{d}$). All non-bold letters denote scalar variables. $\mathbf{d}_j$ is the j^{th} column of the matrix $\mathbf{D}$. d_{ij} denotes the scalar in the i^{th} row and j^{th} column of $\mathbf{D}$. $||\mathbf{d}||_2^2$ denotes the Euclidean squared norm of the vector $\mathbf{d}$. $tr(\mathbf{A}) = \sum_i a_{ii}$ is the trace of the matrix $\mathbf{A}$. $\mathbf{D} = diag(\mathbf{a})$ is an operator that transforms a vector $\mathbf{a}$ into a diagonal matrix $\mathbf{D}$ such that $d_{ii} = a_i$. $vec(\mathbf{A})$ is an operator which converts a matrix $\mathbf{A} \in \Re^{m \times n}$ into a column vector $\mathbf{a} \in \Re^{mn \times 1}$. $||\mathbf{A}||_F^2 = tr(\mathbf{A}^T\mathbf{A}) = tr(\mathbf{A}\mathbf{A}^T)$ designates the squared Frobenius norm of $\mathbf{A}$. $|\mathbf{A}|$ represents the determinant of the matrix $\mathbf{A}$. $\mathbf{1}_k \in \Re^{k \times 1}$ is a vector of ones. $\mathbf{I}_k$ denotes a $k \times k$ identity matrix. $\circ$ denotes the Hadamard product, $*$ represents the convolution, and $\otimes$ the Kronecker product. $\propto$ refers to "proportional to the maximization of". J_x denotes an error function for a standard formulation of a CA method, and E_x refers to the LS-WKRRR version.

III. A Generative Model for Component Analysis

This section introduces the formulation for the least-squares weighted kernel reduced rank regression (LS-WKRRR) problem. In the following sections, we will show how the LS-WKRRR is the generative model for many CA methods, including KPCA, KLDA, KCCA, LE, and Ncuts.

A. Least-Squares Weighted Kernel Reduced Rank Regression (LS-WKRRR)

Since its introduction in the early 1950s by Anderson [16], [17], the reduced-rank regression (RRR) model has inspired a wealth of diverse applications in several fields such as signal processing [18], [19] (also known as reduced-rank Wiener filtering), neural networks [20] (also known as asymmetric PCA), time series analysis [16], and computer vision [21]. This section extends previous work by introducing kernels and weights into the RRR framework, and it derives the system of GEPs resulting from solving the LS-WKRRR problem.

Learning a linear regression between two high-dimensional data sets is usually an ill-posed problem due to lack of training samples to constrain the regression parameters. Consider learning a regression between two high-dimensional data sets, $\mathbf{X} \in \Re^{x \times n}$ and $\mathbf{D} \in \Re^{d \times n}$, and let $\mathbf{T} \in \Re^{d \times x}$ be the regression matrix. The LS regression problem minimizes $\min_{\mathbf{T}} ||\mathbf{D} - \mathbf{T}\mathbf{X}||_F^2$. The optimal $\mathbf{T}$ can be found in closed-form as $\mathbf{T} = \mathbf{D}\mathbf{X}^T(\mathbf{X}\mathbf{X}^T)^{-1}$. If $rank(\mathbf{X}) < x$ the matrix $\mathbf{X}\mathbf{X}^T$ will be rank deficient. In this situation dimensionality reduction or regularization is often necessary. A common approach to learn the mapping is to independently learn low-dimensional models for $\mathbf{X}$ and $\mathbf{D}$ using PCA or KPCA, and then to learn a linear or non-linear mapping between the projections using a supervised learning technique. Applying PCA/KPCA separately to each set preserves the directions of maximum variance within the set, but these do not necessarily correspond to the direction of maximum covariation between sets [21]. That is, independently learning low-dimensional models may result in a loss of important detail relevant to the coupling between sets. The RRR model [16], [19], [20] finds a linear mapping, $\mathbf{T}$, that minimizes the LS error subject to a rank constraint on $\mathbf{T}$, effectively reducing the number of free parameters to estimate. The RRR model minimizes $||\mathbf{D} - \mathbf{T}\mathbf{X}||_F^2$ subject to $rank(\mathbf{T}) = k$.

The LS-WKRRR extends previous work on RRR in three aspects: (1) it explicitly parameterizes $\mathbf{T}$ as the outer product of two matrices of rank k, that is $\mathbf{T} = \mathbf{B}\mathbf{A}^T$, where $\mathbf{A} \in \Re^{x \times k}$ and $\mathbf{B} \in \Re^{d \times k}$, similar to [19]–[21]; (2) it incorporates non-linear regression. In the more general formulation, LS-WKRRR maps $\mathbf{D}$ and $\mathbf{X}$ to a feature space using kernel methods. That is, $\mathbf{\Gamma} = \boldsymbol{\phi}(\mathbf{D}) = [\ \boldsymbol{\phi}(\mathbf{d}_1)\ \boldsymbol{\phi}(\mathbf{d}_2)\ \cdots\ \boldsymbol{\phi}(\mathbf{d}_n)\ ] \in \Re^{d_d \times n}$ represents a mapping of $\mathbf{D}$. $\boldsymbol{\phi}$ denotes a mapping from the d dimensional input space to the feature space (d_d dimensions). Similarly, $\mathbf{\Upsilon} = \Phi(\mathbf{X}) = [\ \varphi(\mathbf{x}_1)\ \varphi(\mathbf{x}_2)\ \cdots\ \varphi(\mathbf{x}_n)\ ] \in \Re^{d_x \times n}$ denotes the mapping for $\mathbf{X}$. $\boldsymbol{\phi}$ and φ map the data to a (usually) higher dimensional space, where the data is more likely to behave linearly. (3) The LS-WKRRR incorporates different weights for the features $\mathbf{W}_r \in \Re^{d_d \times d_d}$, and samples $\mathbf{W}_c \in \Re^{n \times n}$.

The LS-WKRRR problem minimizes the following expression

$$\begin{aligned} E_0(\mathbf{A}, \mathbf{B}) = ||\mathbf{W}_r(\mathbf{\Gamma} - \mathbf{B}\mathbf{A}^T\mathbf{\Upsilon})\mathbf{W}_c||_F^2 = \\ tr(\mathbf{W}_c^T\mathbf{\Gamma}^T\mathbf{W}_r^T\mathbf{W}_r\mathbf{\Gamma}\mathbf{W}_c) - 2tr(\mathbf{W}_c^T\mathbf{\Gamma}^T\mathbf{W}_r^T\mathbf{W}_r\mathbf{B}\mathbf{A}^T\mathbf{\Upsilon}\mathbf{W}_c) \\ +tr(\mathbf{W}_c^T\mathbf{\Upsilon}^T\mathbf{A}\mathbf{B}^T\mathbf{W}_r^T\mathbf{W}_r\mathbf{B}\mathbf{A}^T\mathbf{\Upsilon}\mathbf{W}_c)\ , \end{aligned} \tag{1}$$

with respect to the regression matrices $\mathbf{A} \in \Re^{d_x \times k}$ and $\mathbf{B} \in \Re^{d_d \times k}$. $\mathbf{A}$ spans the subspace that preserves the correlation between $\mathbf{\Upsilon}$ and $\mathbf{\Gamma}$, and $\mathbf{B}$ spans the column space of $\mathbf{\Gamma}$. $\mathbf{W}_r \in \Re^{d_d \times d_d}$ is a matrix that weights the features (e.g., PCA) or classes (e.g., LDA). Similarly, $\mathbf{W}_c \in \Re^{n \times n}$ weights the importance of each sample. In the following, we will assume that the weighting matrices are symmetric and full rank. Eq. (1) is the **fundamental equation of CA methods**. In the rest of the manuscript, we will show how to relate many CA methods to this equation.

The necessary conditions on $\mathbf{A}$ and $\mathbf{B}$ for the minimum of Eq. (1) are

$$\frac{\partial E_0}{\partial \mathbf{B}} = 2\mathbf{W}_r^2\mathbf{B}\mathbf{A}^T\mathbf{\Upsilon}\mathbf{W}_c^2\mathbf{\Upsilon}^T\mathbf{A} - 2\mathbf{W}_r^2\mathbf{\Gamma}\mathbf{W}_c^2\mathbf{\Upsilon}^T\mathbf{A} = \mathbf{0}, \tag{2}$$

$$\frac{\partial E_0}{\partial \mathbf{A}} = 2\mathbf{\Upsilon}\mathbf{W}_c^2\mathbf{\Gamma}^T\mathbf{W}_r^2\mathbf{B} - 2\mathbf{\Upsilon}\mathbf{W}_c^2\mathbf{\Upsilon}^T\mathbf{A}\mathbf{B}^T\mathbf{W}_r^2\mathbf{B} = \mathbf{0}. \tag{3}$$

Eq. (2) and Eq. (3) form a set of coupled equations that have solutions in terms of a GEP in either $\mathbf{A}$ or $\mathbf{B}$. Assuming that $\mathbf{A}^T\mathbf{\Upsilon}\mathbf{W}_c^2\mathbf{\Upsilon}^T\mathbf{A}$ is invertible and substituting the optimal $\mathbf{B} = \mathbf{\Gamma}\mathbf{W}_c^2\mathbf{\Upsilon}^T\mathbf{A}(\mathbf{A}^T\mathbf{\Upsilon}\mathbf{W}_c^2\mathbf{\Upsilon}^T\mathbf{A})^{-1}$ derived from Eq. (2) into Eq. (1), minimizing $E_0(\mathbf{A})$ w.r.t. $\mathbf{A}$ is equivalent to the maximization of

$$tr\big((\mathbf{A}^T\mathbf{\Upsilon}\mathbf{W}_c^2\mathbf{\Upsilon}^T\mathbf{A})^{-1}(\mathbf{A}^T\mathbf{\Upsilon}\mathbf{W}_c^2\mathbf{\Gamma}^T\mathbf{W}_r^2\mathbf{\Gamma}\mathbf{W}_c^2\mathbf{\Upsilon}^T\mathbf{A})\big). \tag{4}$$

Similarly, assuming that $(\mathbf{B}^T\mathbf{W}_r^2\mathbf{B})^{-1}$ and $(\mathbf{\Upsilon}\mathbf{W}_c^2\mathbf{\Upsilon}^T)^{-1}$ exist, substituting the optimal value of $\mathbf{A} = (\mathbf{\Upsilon}\mathbf{W}_c^2\mathbf{\Upsilon}^T)^{-1}\mathbf{\Upsilon}\mathbf{W}_c^2\mathbf{\Gamma}^T\mathbf{W}_r^2\mathbf{B}(\mathbf{B}^T\mathbf{W}_r^2\mathbf{B})^{-1}$ from Eq. (3) into Eq. (1), minimizing $E_0(\mathbf{B})$ is equivalent to the maximization of

$$\begin{aligned} tr\big((\mathbf{B}^T\mathbf{W}_r^2\mathbf{B})^{-1}(\mathbf{B}^T\mathbf{W}_r^2\mathbf{\Gamma}\mathbf{W}_c^2\mathbf{\Upsilon}^T \\ (\mathbf{\Upsilon}\mathbf{W}_c^2\mathbf{\Upsilon}^T)^{-1}\mathbf{\Upsilon}\mathbf{W}_c^2\mathbf{\Gamma}^T\mathbf{W}_r^2\mathbf{B})\big). \end{aligned} \tag{5}$$

Eq. (4) and Eq. (5) are quotient trace problems (one possible multidimensional extension of Rayleigh quotients). For a given pair $\mathbf{S}_1, \mathbf{S}_2$ of real symmetric matrices, the quotient trace problem optimizes

$$J(\mathbf{B}) = tr\big((\mathbf{B}^T\mathbf{S}_2\mathbf{B})^{-1}\mathbf{B}^T\mathbf{S}_1\mathbf{B}\big), \tag{6}$$

and the solution is given by the following GEP [15]

$$\mathbf{S}_1\mathbf{B} = \mathbf{S}_2\mathbf{B}\mathbf{\Lambda}, \tag{7}$$

where $\mathbf{\Lambda}$ is a diagonal matrix containing the generalized eigenvalues. The eigenvectors (columns of $\mathbf{B}$) are critical points of $J(\mathbf{B})$. The solution of Eq. (4) is unique up to an invertible transformation $\mathbf{R} \in \Re^{k\times k}$, that is, $E_0(\mathbf{AR}) = E_0(\mathbf{A})$. Similarly, Eq. (5) is invariant under $k \times k$ invertible linear transformations.

Recasting the CA eigen-formulation as a LS-WKRRR problem, Eq. (1), has a number of desirable benefits that will be illustrated throughout the paper:

1) Eq. (1) provides a unified expression for many CA methods. The commonalities and differences between the methods, as well as the intrinsic relationship, can be easily understood from Eq. (1). See Sections IV, V, VI and VII.
2) The Least-Squares (LS) formulation provides an alternative and simple framework to understand normalization factors in CA methods (e.g., normalization factors in spectral graph clustering in Section VII, weighting factors in PCA/LDA in Section IV).
3) Eq. (1) has a unique global minimum [22] and many numerical optimization methods are available to solve it (Section III.B). In general, algorithms that directly optimize the LS-WKRRR can be more efficient than eigen-solvers for large-scale problems. In addition, on-line versions can be easily derived.
4) Directly optimizing Eq. (1) solves the small sample size (SSS) problem of standard eigen-formulations.
5) The LS formulation allows many extensions of CA methods (Section VIII). It is unclear how to formulate these new extensions using eigen-formulations.

B. Computational Aspects of LS-WKRRR

This section reviews three methods to optimize the LS-WKRRR model.

1) Subspace Iteration: Standard numerical packages to solve GEPs (i.e. $\mathbf{S}_1\mathbf{B} = \mathbf{S}_2\mathbf{B}\Lambda$) are not well suited to solve Eq. (4) or Eq. (5) for high-dimensional data, especially when the number of samples is smaller than the number of features (SSS problem). In this case, directly minimizing the Rayleigh quotient $\frac{\mathbf{x}^T\mathbf{S}_1\mathbf{x}}{\mathbf{x}^T\mathbf{S}_2\mathbf{x}}$ with numerical methods (e.g., [12], [23]) can avoid the SSS problem. However, these methods rely on deflation procedures in order to obtain several eigenvectors and the deflation process often breaks down numerically [24] (especially when increasing the number of eigenvectors). To overcome these problems, this section reviews the subspace iteration method [24].

Given two covariance matrices, $\mathbf{S}_1 \in \Re^{d\times d}$ and $\mathbf{S}_2 \in \Re^{d\times d}$, and an initial random matrix $\mathbf{V}_0 \in \Re^{d\times q}$, the subspace iteration method alternates the following steps:

$$\mathbf{S}_1\hat{\mathbf{V}}_{t+1} = \mathbf{S}_2\mathbf{V}_t \ , \tag{8}$$

$$\mathbf{S} = \hat{\mathbf{V}}_{t+1}^T\mathbf{S}_1\hat{\mathbf{V}}_{t+1} \ , \qquad \mathbf{T} = \hat{\mathbf{V}}_{t+1}^T\mathbf{S}_2\hat{\mathbf{V}}_{t+1} \ , \tag{9}$$

$$\mathbf{SW} = \mathbf{TW}\mathbf{\Delta} \ , \tag{10}$$

$$\mathbf{V}_{t+1} = \hat{\mathbf{V}}_{t+1}\mathbf{W} \ , \qquad \hat{\mathbf{V}}_{t+1} = \hat{\mathbf{V}}_{t+1}/||\hat{\mathbf{V}}_{t+1}||_F.$$

The first step of the subspace iteration algorithm, Eq. (8), solves a linear system of equations to find $\hat{\mathbf{V}}_{t+1}$. In the second step, the covariances $\mathbf{S}_1$ and $\mathbf{S}_2$ are projected onto $\hat{\mathbf{V}}_{t+1}$, Eq. (9). In order to impose the constraints that $\mathbf{V}_{t+1}^T\mathbf{S}_1\mathbf{V}_{t+1} = \mathbf{\Lambda}$ and $\mathbf{V}_{t+1}^T\mathbf{S}_2\mathbf{V}_{t+1} = \mathbf{I}_q$, $\hat{\mathbf{V}}_{t+1}$ is transformed by $\mathbf{W}$. $\mathbf{W}$ results from solving the $q \times q$ GEP of Eq. (10). It can be shown that as t increases, $\mathbf{V}_{t+1}$ will converge to the eigenvectors of $\mathbf{S}_1\mathbf{B} = \mathbf{S}_2\mathbf{B}\mathbf{\Phi}$ and $\mathbf{\Delta}$ to the eigenvalues $\mathbf{\Phi}$ [24]. The convergence is achieved when $\frac{|\delta_i^{k+1} - \delta_i^k|}{\delta_i^{k+1}} < \epsilon \ \forall i$, where δ_i^k denotes the k-largest generalized eigenvalue, and ϵ is the convergence criterion. The subspace iteration algorithm converges linearly and the convergence rate is proportional to $\frac{|\delta_q|}{|\delta_{q+1}|}$ [24]. It is not critical that $\mathbf{V}_0$ has a projection onto the first q generalized eigenvectors, because numerical errors will provide such a projection.

2) Alternated Least Squares (ALS): ALS approaches alternate between solving for $\mathbf{A}$ with $\mathbf{B}$ fixed, and solving for $\mathbf{B}$ with $\mathbf{A}$ fixed. Each step can be computed in closed-form as

$$\mathbf{A}^{t+1} = (\mathbf{\Upsilon}\mathbf{W}_c^2\mathbf{\Upsilon}^T)^{-1}\mathbf{\Upsilon}\mathbf{W}_c^2\mathbf{\Gamma}^T\mathbf{W}_r^2\mathbf{B}^t({\mathbf{B}^t}^T\mathbf{W}_r^2\mathbf{B}^t)^{-1}, \tag{11}$$

$$\mathbf{B}^{t+1} = \mathbf{\Gamma}\mathbf{W}_c^2\mathbf{\Upsilon}^T\mathbf{A}^{(t+1)}({\mathbf{A}^{(t+1)}}^T\mathbf{\Upsilon}\mathbf{W}_c^2\mathbf{\Upsilon}^T\mathbf{A}^{(t+1)})^{-1}. \tag{12}$$

In the case of kernel methods, the ALS procedure needs to re-parameterize $\mathbf{B}$, see Section IV-B for more details.

3) Gradient descent and second-order methods: For large amounts of high-dimensional data, gradient descent and second-order algorithms (e.g., Newton, conjugate gradient) are typically more computationally efficient than eigensolvers [25], [26]. Eq. (2) and Eq. (3) suggest a simple gradient descent update:

$$\mathbf{A}^{t+1} = \mathbf{A}^t - \eta_a\frac{\partial E_0(\mathbf{A}^t)}{\partial \mathbf{A}} \ , \qquad \mathbf{B}^{t+1} = \mathbf{B}^t - \eta_b\frac{\partial E_0(\mathbf{B}^t)}{\partial \mathbf{B}}. \tag{13}$$

η_a and η_b in Eq. (13) can be estimated using a line search strategy [25], [27]. Alternatively, an upper bound on the diagonal of the Hessian matrix can be used [26], [28]. Recently, Buchanan and Fitzgibbon [25] showed how second-order algorithms such as the damped Newton algorithm on the joint matrix vec([$\mathbf{A}$; $\mathbf{B}$])) is more efficient than ALS or gradient descent algorithms to solve for $\mathbf{A}, \mathbf{B}$. Moreover, in the case of having missing data, the joint damped Newton algorithm is able to avoid local minima more often. Finally, it is important to notice that both the ALS and the gradient-based methods effectively solve the SSS problem unlike those that directly solve the GEP.

IV. PCA, KPCA, and Weighted Extensions

This section derives PCA, KPCA and weighted extensions as a particular case of the fundamental equation of CA methods, Eq. (1).

A. Principal Component Analysis (PCA)

PCA is one of the most popular dimensionality reduction techniques [1]–[3], [20]. The basic ideas behind PCA date back to Pearson in 1901 [2], and a more general procedure was described by Hotelling [3] in 1933. PCA finds an orthogonal subspace $\mathbf{B} \in \Re^{d\times k}$ that maximizes

$$J_1(\mathbf{B}) = tr(\mathbf{B}^T\mathbf{S}_t\mathbf{B}) \quad s.t. \quad \mathbf{B}^T\mathbf{B} = \mathbf{I}_k, \tag{14}$$

where $\mathbf{S}_t = \frac{1}{n-1}\mathbf{D}(\mathbf{I}_n - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^T)\mathbf{D}^T$ denotes the covariance matrix (see Appendix A). $\mathbf{B}$ is a basis for the principal subspace of $\mathbf{D}$, where d denotes the number of features, n the number of

samples, k the dimension of the subspace, and $k \leqslant min(n,d)$. PCA can be computed in closed-form by calculating the leading eigenvectors of the covariance matrix $\mathbf{S}_t$ [1], [20]. The PCA projections, $\mathbf{C} = \mathbf{B}^T\mathbf{D}(\mathbf{I}_n - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^T) \in \Re^{k\times n}$, are decorrelated, that is, $\mathbf{C}\mathbf{C}^T = \mathbf{\Lambda}$, where $\mathbf{\Lambda} \in \Re^{k\times k}$ is a diagonal matrix containing the eigenvalues of $\mathbf{S}_t$. During the paper, the trace quotients in standard CA methods or standard CA formulations will be denoted by J_x, whereas the LS-WKRRR or LS extensions will be denoted by E_x.

For large data sets of high-dimensional data (d and n are large), minimizing the least-squares error function is an efficient procedure (in both space and time) to compute the principal subspace of $\mathbf{D}$ [29], [30]. There exist several least-squares error functions such that the stationary points are solutions of PCA. Consider the fundamental equation of CA, Eq. (1), where $\mathbf{\Upsilon} = \mathbf{I}_n, \mathbf{W}_r = \mathbf{I}_d, \mathbf{W}_c = \mathbf{I}_n, \mathbf{\Gamma} = \mathbf{D}$, and $\mathbf{D}\mathbf{1}_n = \mathbf{0}$ (zero mean data):

$$E_1(\mathbf{B},\mathbf{A}) = ||\mathbf{D} - \mathbf{B}\mathbf{A}^T||_F^2. \tag{15}$$

In this case, Eq. (4) and Eq. (5) transform to

$$E_1(\mathbf{A}) \propto tr\big((\mathbf{A}^T\mathbf{A})^{-1}(\mathbf{A}^T\mathbf{D}^T\mathbf{D}\mathbf{A})\big), \tag{16}$$

$$E_1(\mathbf{B}) \propto tr\big((\mathbf{B}^T\mathbf{B})^{-1}(\mathbf{B}^T\mathbf{D}\mathbf{D}^T\mathbf{B})\big). \tag{17}$$

Recall that $\propto$ represents "proportional to the maximization of". The optimal $\mathbf{B}$ (primal problem) is given by the leading eigenvectors of the covariance matrix $\mathbf{D}\mathbf{D}^T \in \Re^{d\times d}$, that effectively maximizes Eq. (17). Similarly, the dual PCA formulation finds the matrix $\mathbf{A}$ that maximizes Eq. (16). The optimal $\mathbf{A}$ is given by the leading eigenvectors of the Gram matrix $\mathbf{D}^T\mathbf{D} \in \Re^{n\times n}$.

Eq. (15) can be solved directly with ALS, gradient descent [29] or second-order methods [25]. ALS approaches to solve Eq. (15), alternate between solving for $\mathbf{A}$ while $\mathbf{B}$ is fixed and vice versa [22], [26], [31], [32]. In the case of PCA, the ALS equations (Eq. (11) and Eq. (12)) can be solved with the following systems of linear equations: $\mathbf{D}^T\mathbf{B} = \mathbf{A}\mathbf{B}^T\mathbf{B}$ and $\mathbf{D}\mathbf{A} = \mathbf{B}\mathbf{A}^T\mathbf{A}$. This optimization is equivalent to the Expectation Maximization (EM) algorithm in probabilistic PCA (PPCA) [30], [33] when the noise becomes infinitesimal and equal in all directions. Once $\mathbf{A}$ and $\mathbf{B}$ are found, the unique PCA solution ($\hat{\mathbf{B}}$) can be obtained by finding an invertible transformation $\mathbf{R} \in \Re^{k\times k}$ that jointly diagonalizes $\hat{\mathbf{B}}^T\hat{\mathbf{B}}$ and $\hat{\mathbf{A}}^T\hat{\mathbf{A}}$, where $\hat{\mathbf{B}} = \mathbf{B}\mathbf{R}$ and $\hat{\mathbf{A}} = \mathbf{A}(\mathbf{R}^{-1})^T$. $\mathbf{R}$ has to satisfy the simultaneous diagonalization of $\mathbf{R}^T\mathbf{B}^T\mathbf{B}\mathbf{R} = \mathbf{I}$ and $\mathbf{R}^T(\mathbf{A}^T\mathbf{A})^{-1}\mathbf{R} = \mathbf{\Lambda}^{-1}$, where $\mathbf{\Lambda} \in \Re^{k\times k}$ is a diagonal matrix containing the eigenvalues of the covariance matrix $\mathbf{S}_t$. $\mathbf{R}$ can be computed by solving the $k \times k$ GEP $(\mathbf{A}^T\mathbf{A})^{-1}\mathbf{R} = \mathbf{B}^T\mathbf{B}\mathbf{R}\mathbf{\Lambda}^{-1}$.

Alternatively, PCA can also be derived from a least-squares optimization problem by considering Eq. (1) with the following values [29]: $\mathbf{\Gamma} = \mathbf{D}, \mathbf{W}_r = \mathbf{I}_d, \mathbf{W}_c = \mathbf{I}_n, \mathbf{A} = \mathbf{B}$, that results in:

$$E_2(\mathbf{B}) = ||\mathbf{D} - \mathbf{B}(\mathbf{B}^T\mathbf{D})||_F^2 \;\; s.t. \;\; \mathbf{B}^T\mathbf{B} = \mathbf{I}_k \; . \tag{18}$$

However, Eq. (18) is more challenging to optimize because it is quartic in $\mathbf{B}$. Moreover, this formulation of PCA does not allow to incorporate robustness to intra-sample outliers [26].

B. Kernel Principal Component Analysis (KPCA)

Similar to PCA, KPCA [34] can be derived from Eq. (1), by lifting the original data samples, $\mathbf{D}$, to a feature space, $\mathbf{\Gamma} = \boldsymbol{\phi}(\mathbf{D})$. The kernelized version of Eq. (15) can be written as

$$E_3(\mathbf{B},\mathbf{A}) = ||\mathbf{\Gamma} - \mathbf{B}\mathbf{A}^T||_F^2. \tag{19}$$

Observe that in the case of kernel methods, it is (in general) not possible to directly solve the primal problem, Eq. (5). This is because the covariance in the feature space, $\mathbf{\Gamma}\mathbf{\Gamma}^T$, can be infinite dimensional. In the dual problem, $\mathbf{A}$ can be computed maximizing Eq. (4), that is

$$E_3(\mathbf{A}) \propto tr((\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{K}\mathbf{A}), \tag{20}$$

where $\mathbf{K} = \mathbf{\Gamma}^T\mathbf{\Gamma} \in \Re^{n\times n}$ is the kernel matrix. Each element $k_{ij} = k(\mathbf{d}_i, \mathbf{d}_j)$ of $\mathbf{K}$ represents the similarity between two samples by means of a kernel function. To center the kernel matrix in the feature space, the mean needs to be introduced into the formulation, i.e. $||\mathbf{\Gamma} - \boldsymbol{\mu}\mathbf{1}_n^T - \mathbf{B}\mathbf{A}^T||_F^2$, where $\boldsymbol{\mu} = \frac{1}{n}\mathbf{\Gamma}\mathbf{1}_n$. We omit the details in the interest of space.

For large amounts of data (large n) an ALS or down-hill approaches to computing KPCA can be computationally more convenient (see Section III-B.2). To apply the ALS method in the case of KPCA, a re-parameterization of $\mathbf{B}$ is needed. Recall that for KPCA, $\mathbf{B}$ can be expressed as a linear combination of the data in feature space $\mathbf{\Gamma}$ [35]; that is, $\mathbf{B} = \mathbf{\Gamma}\boldsymbol{\alpha}$, where $\boldsymbol{\alpha} \in \Re^{n\times k}$. Substituting this expression into Eq. (19) results in

$$E_3(\boldsymbol{\alpha},\mathbf{A}) = ||\mathbf{\Gamma}(\mathbf{I}_n - \boldsymbol{\alpha}\mathbf{A}^T)||_F^2. \tag{21}$$

Assuming that $\mathbf{K}$ is invertible, we can alternate between computing $\boldsymbol{\alpha}$ and $\mathbf{A}$ as

$$\boldsymbol{\alpha} = \mathbf{A}(\mathbf{A}^T\mathbf{A})^{-1} \;, \quad \mathbf{A} = (\boldsymbol{\alpha}^T\mathbf{K}\boldsymbol{\alpha})^{-1}\boldsymbol{\alpha}^T\mathbf{K} \; . \tag{22}$$

C. Weighted Extensions

In many situations it is convenient to weight differently features and/or samples. For instance, when modeling faces from images, it is likely that some pixels have more variance than others (e.g., pixels in the eye regions have more variance than pixels in the cheeks) and they should be weighted less in the model. Alternatively, we might be interested in weighing the influence of samples (e.g., reduce the influence of sample outliers in the subspace).

Eq. (4) and Eq. (5) provide a partial solution to the weighting problem. For instance, consider the weighted PCA case, with a matrix that weights rows ($\mathbf{W}_r$) and a matrix that weights columns ($\mathbf{W}_c$) in Eq. (1). The closed-form solutions for the weighted PCA are given by Eq. (4) and Eq. (5):

$$E_0(\mathbf{A}) \propto tr\big((\mathbf{A}^T\mathbf{W}_c^2\mathbf{A})^{-1}(\mathbf{A}^T\mathbf{W}_c^2\mathbf{D}^T\mathbf{W}_r^2\mathbf{D}\mathbf{W}_c^2\mathbf{A})\big) \;, \tag{23}$$

$$E_0(\mathbf{B}) \propto tr\big((\mathbf{B}^T\mathbf{W}_r^2\mathbf{B})^{-1}(\mathbf{B}^T\mathbf{W}_r^2\mathbf{D}\mathbf{W}_c^2\mathbf{D}^T\mathbf{W}_r^2\mathbf{B})\big). \tag{24}$$

Eq. (23) and Eq. (24) have a closed-form solution as a GEP. For $\mathbf{A}$, the GEP is $\mathbf{W}_c^2\mathbf{D}^T\mathbf{W}_r^2\mathbf{D}\mathbf{W}_c^2\mathbf{A} = \mathbf{W}_c^2\mathbf{A}\mathbf{\Lambda}_a$ and for $\mathbf{B}$ the GEP is $\mathbf{W}_r^2\mathbf{D}\mathbf{W}_c^2\mathbf{D}^T\mathbf{W}_r^2\mathbf{B} = \mathbf{W}_r^2\mathbf{B}\mathbf{\Lambda}_b$. The Generalized Singular Value Decomposition [36], [37] provides an alternative approach to solve the previous weighted PCA problem.

It is also possible to find a weighted KPCA solution for features and samples. Weighting the samples (i.e. $\mathbf{W}_c \neq \mathbf{I}_n$) directly translates to weighting the kernel matrix and results in solving the following GEP: $\mathbf{K}\mathbf{W}_c^2\mathbf{A} = \mathbf{A}\mathbf{\Lambda}_a$. If the weighting is in the feature space (i.e. $\mathbf{W}_r \neq \mathbf{I}_{d_d}$), the weighted KPCA problem can still be solved using the kernel trick [38].

In general, for an arbitrary set of weights for features or samples, the weighted PCA minimizes

$$E_4(\mathbf{A},\mathbf{B}) = ||\mathbf{W} \circ (\mathbf{D} - \mathbf{B}\mathbf{A}^T)||_F^2, \tag{25}$$

where $\circ$ denotes the Hadamard or pointwise product. In general, Eq. (25) does not have a closed-form solution in terms of GEP [31], [37]. Moreover, the problem of data factorization with arbitrary weights has several local minima depending on the structure of the weights [25], [39]. Minimization of Eq. (25), has been typically used to solve PCA with missing data [25], [31], [39] and outliers [26], [40]. Recently, Aguiar *et al.* [41] proposed a closed-form solution to the data factorization problem, when the missing data has a special structure.

V. LDA, KLDA, CCA, KCCA and Weighted extensions

This section relates LDA, KLDA, CCA and KCCA to Eq. (1), and derives weighted generalizations.

A. Linear Discriminant Analysis (LDA)

Let $\mathbf{D} \in \Re^{d\times n}$ be a matrix, where each column is a vectorized data sample from one of c classes. d denotes the number of features and n the number of samples. $\mathbf{G} \in \Re^{n\times c}$ is an indicator matrix such that $\sum_j g_{ij} = 1$, $g_{ij} \in \{0, 1\}$, and g_{ij} is 1 if $\mathbf{d}_i$ belongs to class j, and 0 otherwise. LDA, originally proposed by Fisher [4], [5] for the two-class case and later extended to the multi-class case [15], [42], computes a linear transformation ($\mathbf{A} \in \Re^{d\times k}$) of $\mathbf{D}$ that maximizes the Euclidean distance between the means of the classes ($\mathbf{S}_b$) while minimizing the within-class variance ($\mathbf{S}_w$). Trace quotients are among the most popular LDA optimization criteria [15]. For instance, LDA can be obtained by maximizing

$$J_2(\mathbf{A}) = tr((\mathbf{A}^T\mathbf{S}_1\mathbf{A})^{-1}\mathbf{A}^T\mathbf{S}_2\mathbf{A}), \tag{26}$$

where several combinations of $\mathbf{S}_1$ and $\mathbf{S}_2$ matrices lead to the same LDA solution (e.g., $\mathbf{S}_1 \in \{\mathbf{S}_w, \mathbf{S}_t, \mathbf{S}_w\}$ and $\mathbf{S}_2 \in \{\mathbf{S}_b, \mathbf{S}_b, \mathbf{S}_t\}$). In the case of high-dimensional data, the covariance matrices are likely to be rank-deficient due to lack of training samples, and standard eigen-solutions for LDA can be ill-conditioned. This is the well-known small sample size (SSS) problem. In recent years, many algorithms have been proposed to deal with the SSS problem, including PCA+LDA [43], [44], regularized LDA [45], and many other methods that explore several combinations of the Null and Range spaces of $\mathbf{S}_1$ and $\mathbf{S}_2$ [46]. See [47] for the analysis of the maximum in Eq. (26) as a function of the four fundamental spaces of $\mathbf{S}_1$ and $\mathbf{S}_2$.

LDA has been previously formulated as a regression problem for the two-class case [48], and extended to the multi-class case [45], [49], [50]. This section provides a simpler derivation of the relation between regression and LDA following our previous work [10]. In the following, we will assume zero mean data ($\mathbf{D1} = \mathbf{0}$). Consider Eq. (1), where $\mathbf{\Gamma} = \mathbf{G}^T$, $\mathbf{\Upsilon} = \mathbf{D}$, $\mathbf{W}_r = (\mathbf{G}^T\mathbf{G})^{-\frac{1}{2}}$, $\mathbf{W}_c = \mathbf{I}_n$, and $\mathbf{D1} = \mathbf{0}$:

$$E_5(\mathbf{A}, \mathbf{B}) = ||(\mathbf{G}^T\mathbf{G})^{-\frac{1}{2}}(\mathbf{G}^T - \mathbf{B}\mathbf{A}^T\mathbf{D})||_F^2. \tag{27}$$

In this case, Eq. (4) transforms to

$$E_5(\mathbf{A}) \propto tr((\mathbf{A}^T \underbrace{\mathbf{D}\mathbf{D}^T}_{\mathbf{S}_t}\mathbf{A})^{-1}\mathbf{A}^T \underbrace{\mathbf{D}\mathbf{G}(\mathbf{G}^T\mathbf{G})^{-1}\mathbf{G}^T\mathbf{D}^T}_{\mathbf{S}_b}\mathbf{A}). \tag{28}$$

$\mathbf{S}_t$ denotes the total covariance matrix and $\mathbf{S}_b$ the between-class covariance matrix (see Appendix A). Eq. (28) is one of the standard trace quotients for LDA. Recall that LDA is a supervised learning problem and the binary indicator matrix $\mathbf{G}$ is known. LDA can be understood as finding a linear mapping with RRR from the data samples ($\mathbf{D}$) to the labels ($\mathbf{G}$). The weighting factor $\mathbf{G}^T\mathbf{G}$ compensates for unequal number of samples between classes. Observe, that directly optimizing Eq. (28) (e.g., gradient descent) with respect to $\mathbf{A}$ and $\mathbf{B}$ in Eq. (27) avoids the SSS problem and can be numerically efficient for large amounts of high-dimensional data.

B. Kernel Linear Discriminant Analysis (KLDA)

KLDA [51] can also be derived from Eq. (1). Consider Eq. (1), where $\mathbf{\Gamma} = \mathbf{G}^T$, $\mathbf{\Upsilon} = \boldsymbol{\phi}(\mathbf{D})$, $\mathbf{W}_r = (\mathbf{G}^T\mathbf{G})^{-\frac{1}{2}}$, $\mathbf{W}_c = \mathbf{I}_n$:

$$E_6(\mathbf{A}, \mathbf{B}) = ||(\mathbf{G}^T\mathbf{G})^{-\frac{1}{2}}(\mathbf{G}^T - \mathbf{B}\mathbf{A}^T\mathbf{\Upsilon})||_F^2. \tag{29}$$

In this case, Eq. (4) has the following expression:

$$E_6(\mathbf{A}) \propto tr((\mathbf{A}^T\mathbf{\Upsilon}\mathbf{\Upsilon}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{\Upsilon}\mathbf{G}(\mathbf{G}^T\mathbf{G})^{-1}\mathbf{G}^T\mathbf{\Upsilon}^T\mathbf{A}). \tag{30}$$

Using the Mercer theorem [35], it can be shown that the solution to the KLDA problem can be expressed as $\mathbf{A} = \mathbf{\Upsilon}\boldsymbol{\alpha}$ [51]. Using this fact, the KLDA can be found as the solution of the following GEP, $\mathbf{K}\mathbf{G}(\mathbf{G}^T\mathbf{G})^{-1}\mathbf{G}^T\mathbf{K}^T\boldsymbol{\alpha} = \mathbf{K}^2\boldsymbol{\alpha}\mathbf{\Lambda}_{\boldsymbol{\alpha}}$, where $\mathbf{K} = \mathbf{\Upsilon}^T\mathbf{\Upsilon}$ is the kernel matrix. $\boldsymbol{\alpha}$ and $\mathbf{\Lambda}_{\boldsymbol{\alpha}}$ are the eigenvectors and eigenvalues of the GEP, respectively.

C. Canonical Correlation Analysis (CCA) and Kernel CCA

CCA is a technique to extract common features from a pair of multivariate data. CCA, first proposed by Hotelling in 1936 [6], identifies relationships between two sets of variables by finding the linear combination of the variables in the first set ($\mathbf{D} \in \Re^{d_d\times n}$) that are most highly correlated with a linear combination of the variables in the second set ($\mathbf{X} \in \Re^{d_x\times n}$).

Assuming zero mean data (i.e. $\mathbf{D1}_n = \mathbf{0}$, $\mathbf{X1}_n = \mathbf{0}$), CCA finds a combination of the original variables (i.e. $\mathbf{B}^T\mathbf{D}$ and $\mathbf{A}^T\mathbf{X}$) that maximize [6]:

$$J_3(\mathbf{A}, \mathbf{B}) = tr(\mathbf{B}^T\mathbf{S}^{\mathbf{DX}}\mathbf{A}) \quad s.t. \quad \mathbf{B}^T\mathbf{S}_t^{\mathbf{D}}\mathbf{B} = \mathbf{A}^T\mathbf{S}_t^{\mathbf{X}}\mathbf{A} = \mathbf{I}, \tag{31}$$

where $\mathbf{S}_t^{\mathbf{X}} = \frac{1}{n-1}\mathbf{X}\mathbf{X}^T, \mathbf{S}_t^{\mathbf{D}} = \frac{1}{n-1}\mathbf{D}\mathbf{D}^T$, and $\mathbf{S}^{\mathbf{DX}} = \frac{1}{n-1}\mathbf{D}\mathbf{X}^T$. The pair of canonical variates ($\mathbf{b}_i^T\mathbf{D}$, $\mathbf{a}_i^T\mathbf{X}$) is uncorrelated with other canonical variates of lower order. Each successive canonical variate pair achieves the maximum relationship orthogonal to the preceding pair. Observe that canonical correlations are invariant with respect to a full-rank affine transformation of $\mathbf{X}$ and $\mathbf{D}$. Eq. (31) has a closed-form solution as two symmetric GEPs [6], [52]:

$$(\mathbf{S}_t^{\mathbf{X}})^{-1}\mathbf{S}^{\mathbf{XD}}(\mathbf{S}_t^{\mathbf{D}})^{-1}\mathbf{S}^{\mathbf{DX}}\mathbf{A} = \mathbf{A}\mathbf{\Lambda}_a, \tag{32}$$

$$(\mathbf{S}_t^{\mathbf{D}})^{-1}\mathbf{S}^{\mathbf{DX}}(\mathbf{S}_t^{\mathbf{X}})^{-1}\mathbf{S}^{\mathbf{XD}}\mathbf{B} = \mathbf{B}\mathbf{\Lambda}_b. \tag{33}$$

The number of solutions (canonical variates) is given by $\min(d_x, d_d)$.

In general, it is not clear how Eq. (1) can recover the canonical variates, because CCA treats both data sets $\mathbf{D}$ and $\mathbf{X}$ symmetrically, whereas LS-WKRRR only normalizes for the input $\mathbf{X}$. At this point, it is worth observing that if $\mathbf{X} = \mathbf{G}^T$ (the indicator matrix), the CCA solution of Eq. (33) is equivalent to the LDA solution, Eq. (28). In this case, using our matrix notation it is straightforward to show that Eq. (33) in CCA reduces to $\mathbf{S}_b^{\mathbf{D}}\mathbf{B} = \mathbf{S}_t^{\mathbf{D}}\mathbf{B}\mathbf{\Lambda}_b$ (assuming zero mean data). Using this fact, we can interpret LDA as CCA. LDA finds the linear subspace that maximally correlates $\mathbf{D}$ with $\mathbf{G}^T$. Using this observation, it is simple to relate CCA to the fundamental equation of CA, Eq. (1).

In order to treat all variables symmetrically, we introduce weights in the predicted variable, $\mathbf{D}$, and show that the CCA solution can be recovered using the fundamental equation of CA, Eq. (1). Consider Eq. (1), where $\mathbf{\Gamma} = \mathbf{D}$, $\mathbf{\Upsilon} = \mathbf{X}$, $\mathbf{W}_r = (\mathbf{D}\mathbf{D}^T)^{-\frac{1}{2}}$, and $\mathbf{W}_c = \mathbf{I}_n$:

$$E_7(\mathbf{A}, \mathbf{B}) = ||(\mathbf{D}^T\mathbf{D})^{-\frac{1}{2}}(\mathbf{D} - \mathbf{B}\mathbf{A}^T\mathbf{X})||_F^2. \quad (34)$$

After substituting these values into Eq. (4) results in

$$E_7(\mathbf{A}) \propto tr((\mathbf{A}^T\mathbf{S}_t^{\mathbf{X}}\mathbf{A})^{-1}\mathbf{A}^T\mathbf{S}^{\mathbf{XD}}(\mathbf{S}_t^{\mathbf{D}})^{-1}\mathbf{S}^{\mathbf{DX}}\mathbf{A}) \ , \quad (35)$$

which corresponds to the GEP for CCA, Eq. (32). Similarly, Eq. (5) corresponds to

$$E_7(\mathbf{B}) \propto tr((\mathbf{B}^T(\mathbf{D}\mathbf{D}^T)^{-1}\mathbf{B})^{-1}\mathbf{B}^T(\mathbf{D}\mathbf{D}^T)^{-1}\mathbf{X} \mathbf{D}^T(\mathbf{D}\mathbf{D}^T)^{-1}\mathbf{D}\mathbf{X}^T(\mathbf{D}\mathbf{D}^T)^{-1}\mathbf{B}) \ . \quad (36)$$

After a change of variable, $\mathbf{U} = \mathbf{B}(\mathbf{D}\mathbf{D}^T)^{-1}$ (assuming the inverse exist), Eq. (36) can be re-written as

$$E_7(\mathbf{U}) \propto tr((\mathbf{U}^T(\mathbf{S}_t^{\mathbf{D}})\mathbf{U})^{-1}\mathbf{U}^T\mathbf{S}^{\mathbf{DX}}(\mathbf{S}_t^{\mathbf{X}})^{-1}\mathbf{S}^{\mathbf{XD}}\mathbf{U}) \ , \quad (37)$$

which is the same solution provided by CCA, Eq. (33). As in KLDA, similar derivation can be done for the case of KCCA.

There exist other LS energy-based formulations of CCA that are worth mentioning. To treat all variables symmetrically, the minima of the following LS function corresponds to CCA:

$$E_8(\mathbf{A}, \mathbf{B}) = ||\mathbf{B}^T\mathbf{D} - \mathbf{A}^T\mathbf{X}||_F^2 \quad (38)$$
$$s.t. \ \ \mathbf{B}^T\mathbf{S}_t^{\mathbf{D}}\mathbf{B} = \mathbf{I}_d \ \ , \ \mathbf{A}^T\mathbf{S}_t^{\mathbf{X}}\mathbf{A} = \mathbf{I}_d,$$

Alternatively, CCA can also be recovered using an unweighted regression. [53], [54] have shown that the canonical variates minimize

$$E_9(\mathbf{A}, \mathbf{B}) = |\mathbf{D} - \mathbf{B}\mathbf{A}^T\mathbf{X}| \ \ s.t. \ \ \mathbf{A}^T\mathbf{S}_t^{\mathbf{X}}\mathbf{A} = \mathbf{I}_d,$$

where recall $|.|$ denotes determinant. This is equivalent to minimizing Eq. (1) if $\mathbf{\Gamma} = \mathbf{D}$, $\mathbf{\Upsilon} = \mathbf{X}$, $\mathbf{W}_r = \mathbf{I}$, $\mathbf{W}_c = \mathbf{I}$ using the determinant instead of the Frobenious norm as the loss function.

D. Weighted extensions

Similar to PCA and KPCA, there are possible weighted extensions for LDA and KLDA. Consider Eq. (4) and Eq. (5) where $\mathbf{\Gamma} = \mathbf{G}^T$ and $\mathbf{W}_r = (\mathbf{G}^T\mathbf{G})^{-\frac{1}{2}}$:

$$E_0(\mathbf{A}) \propto tr\big((\mathbf{A}^T\mathbf{\Upsilon}\mathbf{W}_c^2\mathbf{\Upsilon}^T\mathbf{A})^{-1} (\mathbf{A}^T\mathbf{\Upsilon}\mathbf{W}_c^2\mathbf{G}(\mathbf{G}^T\mathbf{G})^{-1}\mathbf{G}^T\mathbf{W}_c^2\mathbf{\Upsilon}^T\mathbf{A})\big) \ , \quad (39)$$

$$E_0(\mathbf{B}) \propto tr\big((\mathbf{B}^T(\mathbf{G}^T\mathbf{G})^{-1}\mathbf{B})^{-1}(\mathbf{B}^T(\mathbf{G}^T\mathbf{G})^{-1} \mathbf{G}^T\mathbf{W}_c^2\mathbf{\Upsilon}^T(\mathbf{\Upsilon}\mathbf{W}_c^2\mathbf{\Upsilon}^T)^{-1}\mathbf{\Upsilon}\mathbf{W}_c^2\mathbf{G}(\mathbf{G}^T\mathbf{G})^{-1}\mathbf{B}\big) \ . \quad (40)$$

Eq. (39) and Eq. (40) extend previous work on weighted LDA/CCA approaches by allowing us to weight the samples rather than the classes [55]. Similar expressions can be derived for weighted CCA and KCCA, changing $\mathbf{G}^T$ for $\mathbf{X}$.

VI. Non-Linear Embedding Methods

Recently, a large family of algorithms, such as ISOMAP [56], Local Linear Embedding (LLE) [57], Laplacian Eigenmaps (LE) [7] or Locality Preserving Projections (LPP) [8] have derived a compact low-dimensional non-linear embedding that preserves local geometric properties of underlying high-dimensional manifold of the data. In this section, we show how several nonlinear embedding methods can be formulated as a particular instance of LS-WKRRR.

A. Laplacian Eigenmaps (LE) and Locality Preserving Projection (LPP)

Laplacian Eigenmaps (LE) [7] is a non-linear embedding technique originally motivated by the need to visualize and analyze large amounts of multivariate data. The goal of LE is to find an embedding that preserves the local structure of nearby high-dimensional input patterns. LE exploits the Graph Laplacian of a neighborhood graph on the sample data $\mathbf{D}$. Each edge of the $n \times n$ neighborhood graph measures the affinity between two sample points $\mathbf{d}_i$ and $\mathbf{d}_j$. If the nodes i and j are connected (e.g., k-nearest neighbors or ε-neighborhoods), a variety of possible weights can be given, for instance: $w_{ij} = e^{-\frac{1}{2\sigma^2}||\mathbf{d}_i - \mathbf{d}_j||_2^2}$ or 1.

Given the weighted graph, $\mathbf{W} \in \Re^{n \times n}$, LE minimizes [7]

$$J_4(\mathbf{Y}) = \sum_{i=1}^n \sum_{j=1}^n w_{ij}||\mathbf{y}_i - \mathbf{y}_j||_2^2 = 2tr(\mathbf{Y}^T\mathbf{L}\mathbf{Y}) \quad s.t. \ \mathbf{Y}^T\mathbf{S}\mathbf{Y} = \mathbf{I}_k, \quad (41)$$

where $\mathbf{Y} \in \Re^{n \times k}$ is a matrix containing the low dimensional embedding. $\mathbf{S} \in \Re^{n \times n}$ is a diagonal matrix such that each entry is the sum of the rows of $\mathbf{W}$, i.e. $s_{ii} = \sum_j w_{ij}$. $\mathbf{L} = \mathbf{S} - \mathbf{W}$ is the Graph Laplacian. The constraint $\mathbf{Y}^T\mathbf{S}\mathbf{Y} = \mathbf{I}_k$ removes an arbitrary scaling factor in the embedding. Recall that solving Eq. (41) is equivalent to minimizing

$$J_5(\mathbf{Y}) = tr((\mathbf{Y}^T\mathbf{S}\mathbf{Y})^{-1}\mathbf{Y}^T\mathbf{L}\mathbf{Y}). \quad (42)$$

and maximizing

$$J_6(\mathbf{Y}) \propto tr((\mathbf{Y}^T\mathbf{S}\mathbf{Y})^{-1}\mathbf{Y}^T\mathbf{W}\mathbf{Y}). \quad (43)$$

The LE embedding can be found by solving the following GEP, $\mathbf{W}\mathbf{Y} = \mathbf{S}\mathbf{Y}\mathbf{\Lambda}$.

Locality Preserving Projections (LPP) [8], similar to LE, finds an embedding that preserves neighborhood structure of the data. Unlike LE, LPP parameterizes $\mathbf{Y}$ with a linear transformation of the data $\mathbf{Y} = \mathbf{D}^T\mathbf{B}$. Observe that with a linear parameterization it becomes natural to handle new test data out of the observed data set. LPP maximizes

$$J_7(\mathbf{B}) = tr((\mathbf{B}^T\mathbf{D}\mathbf{S}\mathbf{D}^T\mathbf{B})^{-1}\mathbf{B}^T\mathbf{D}\mathbf{W}\mathbf{D}^T\mathbf{B}), \quad (44)$$

LE can also be derived from the fundamental equation of CA. Consider Eq. (1), where, $\mathbf{\Upsilon} = \mathbf{I}_n$, $\mathbf{W}_r = \mathbf{I}_d$ and $\mathbf{W}_c = \mathbf{S}^{\frac{1}{2}}$:

$$E_{10}(\mathbf{B}, \mathbf{A}) = ||(\mathbf{\Gamma} - \mathbf{B}\mathbf{A}^T)\mathbf{S}^{\frac{1}{2}}||_F^2, \quad (45)$$

where $\mathbf{\Gamma} = \boldsymbol{\phi}(\mathbf{D})$. In this case, Eq. (4) translates into

$$E_{10}(\mathbf{A}) \propto tr((\mathbf{A}^T\mathbf{S}\mathbf{A})^{-1}\mathbf{A}^T\mathbf{S}\mathbf{K}\mathbf{S}\mathbf{A}), \quad (46)$$

where $\mathbf{K} = \mathbf{\Gamma}^T\mathbf{\Gamma}$ is the kernel matrix. LE can be achieved minimizing Eq. (45). In this case, the normalized kernel matrix will be $\mathbf{K} = \mathbf{S}^{-1}\mathbf{W}\mathbf{S}^{-1}$. Recall that not for all choices of $\mathbf{W}$, $\mathbf{K}$ will be strictly positive definite or might not have an explicit functional form. In case of computing LE using the unnormalized graph Laplacian, $\mathbf{L}$, it is easy to show that with $\mathbf{W}_c = \mathbf{I}_n$ and adding the mean in the feature space, LE is equivalent to compute KPCA in the pseudo-inverse of $\mathbf{L}$ [58]. Similar connection between KPCA and LE had been previously reported by [59].

LPP can also be derived by minimizing

$$E_{11}(\mathbf{B}, \mathbf{A}) = ||(\mathbf{\Gamma} - \mathbf{B}\mathbf{A}^T\mathbf{D})\mathbf{S}^{\frac{1}{2}}||_F^2. \quad (47)$$

LPP can be interpreted as a method to perform reduced rank regression from the input space to the feature space (assuming a positive definite kernel exists).

B. Local Linear Embedding (LLE) and Neighborhood Preserving Embedding

Local Linear Embedding (LLE) [57] finds an embedding of the data, $\mathbf{D}$, that preserves the local structure of nearby input patterns in the high-dimensional space. LLE builds the embedding by preserving the geometry of pair-wise relations between samples in the high-dimensional manifold. In the first step, LLE computes a weight matrix, $\mathbf{W} \in \Re^{n\times n}$, that contains the structural information of the embedding by minimizing

$$J_8(\mathbf{W}) = \sum_{i=1}^{n} ||\mathbf{d}_i - \sum_{j\in\aleph(i)} w_{ij}\mathbf{d}_j||_2^2 = ||\mathbf{D}(\mathbf{I}_n - \mathbf{W})||_F^2 \\ s.t. \quad \mathbf{W}\mathbf{1}_n = \mathbf{1}_n.$$

$\aleph(i)$ denotes the k-nearest neighbors of $\mathbf{d}_i$, and $\mathbf{W}$ is a matrix such that each column only has k (or less) non-zero values. The weight matrix, $\mathbf{W}$, can be computed by solving a linear system of equations [57]. Once $\mathbf{W}$ is calculated, LLE finds the embedding $\mathbf{Y}$ that minimizes

$$J_9(\mathbf{Y}) = \sum_{i=1}^{n} ||\mathbf{y}_i - \sum_j w_{ij}\mathbf{y}_j||_2^2 = ||\mathbf{Y}(\mathbf{I}_n - \mathbf{W})||_F^2 \quad (48) \\ s.t.\ \mathbf{Y}\mathbf{1}_n = \mathbf{0}\ and\ \mathbf{Y}\mathbf{Y}^T = \mathbf{I}_d\ .$$

Eq. (48) can be solved by finding the eigenvectors corresponding to the smallest nonzero eigenvalues of $\mathbf{M} = (\mathbf{I}_n - \mathbf{W})(\mathbf{I}_n - \mathbf{W})^T$.

Similar to LPP, Neighborhood Preserving Embedding [60] parameterizes $\mathbf{Y}$ with a linear transformation, that is $\mathbf{Y} = \mathbf{B}^T\mathbf{D}$, and the solution is given by the minimum of the GEP $\mathbf{D}\mathbf{M}\mathbf{D}^T\mathbf{B} = \mathbf{D}\mathbf{D}^T\mathbf{B}\mathbf{\Lambda}$. The matrix $\mathbf{M}$ is a discrete approximation of the Laplace Beltrami operator on the manifold [7], [60].

LLE can be interpreted as performing KPCA in a particular kernel matrix [58]. LLE computes the smallest eigenvectors of the matrix $\mathbf{M} = (\mathbf{I}_n - \mathbf{W})(\mathbf{I}_n - \mathbf{W})^T$, which is equivalent to finding the maximum eigenvalue of the identity matrix scaled by the maximum eigenvalue (λ_{max}) minus the original matrix, that is: $\hat{\mathbf{M}} = \lambda_{max}\mathbf{I}_n - \mathbf{M}$. The leading eigenvector of $\hat{\mathbf{M}}$ is $\mathbf{1}_n$ and projecting out this eigenvector is equivalent to the centering operation in feature space done by KPCA [58]. Ham *et al.* [58] have also shown that ISOMAP can also be interpreted as KPCA with special kernel matrices.

VII. K-means and spectral clustering

This section relates LS-WKRRR to k-means, Spectral Clustering and proposes a new clustering method, Discriminative Cluster Analysis (DCA).

A. k-means

k-means clustering [61], [62] splits a set of n objects into c groups by minimizing the within-cluster variation. That is, k-means clustering finds the partition of the data that is a local optimum of the following energy function [10], [63]–[65]:

$$J_{10}(\mathbf{b}_1, ..., \mathbf{b}_c) = \sum_{i=1}^{c}\sum_{j\in C_i} ||\mathbf{d}_j - \mathbf{b}_i||_2^2, \quad (49)$$

where $\mathbf{d}_j$ is a vector representing the j^{th} data point, and $\mathbf{b}_i$ is the geometric centroid of the data points for i^{th} cluster. Eq. (49) can be rewritten in matrix form as [10]:

$$E_{12}(\mathbf{B}, \mathbf{A}) = ||\mathbf{D} - \mathbf{B}\mathbf{A}^T||_F^2 = tr(\mathbf{S}_w) \quad (50) \\ s.t. \quad \mathbf{A}\mathbf{1}_c = \mathbf{1}_n \quad and \quad a_{ij} \in \{0, 1\},$$

where $\mathbf{A} \in \Re^{n\times c}$ is the indicator matrix and $\mathbf{B} \in \Re^{d\times c}$ is the matrix of centroids. Recall that the equivalence between the k-means error function Eq. (49) and Eq. (50) is only valid if $\mathbf{A}$ strictly satisfies the constraints. Observe that Eq. (50) can be derived from the fundamental equation of CA, Eq. (1), where $\mathbf{\Upsilon} = \mathbf{I}_n, \mathbf{W}_r = \mathbf{I}_d, \mathbf{W}_c = \mathbf{I}_n, \mathbf{\Gamma} = \mathbf{D}$.

The k-means algorithm performs coordinate descent in $E_{12}(\mathbf{B}, \mathbf{A})$. Given the actual value of the centroids, $\mathbf{B}$, the first step finds for each data point $\mathbf{d}_j$, the $\mathbf{a}^j$ (j^{th} row of $\mathbf{A}$) such that one of the columns is one and the rest 0, while minimizing Eq. (50). The second step optimizes over $\mathbf{B} = \mathbf{D}\mathbf{A}(\mathbf{A}^T\mathbf{A})^{-1}$, which is equivalent to computing the mean of each cluster.

After optimizing over $\mathbf{B}$, Eq. (50) can be rewritten as: $E_{12}(\mathbf{A}) = ||\mathbf{D} - \mathbf{D}\mathbf{A}(\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T||_F^2 = tr(\mathbf{D}^T\mathbf{D}) - tr((\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{D}^T\mathbf{D}\mathbf{A}) \geq \sum_{i=c+1}^{min(d,n)} \lambda_i$, where λ_i are the eigenvalues of $\mathbf{D}^T\mathbf{D}$. The continuous solution of $\mathbf{A}$ lies in the $c-1$ subspace spanned by the first $c-1$ eigenvectors with largest eigenvalues of $\mathbf{D}^T\mathbf{D}$ [63], [64]. In this case, the error E_{12} is equal to the sum of the residual eigenvalues, i.e. $E_{12} = \sum_{i=c+1}^{min(d,n)} \lambda_i$. This is the spectral relaxation of the k-means algorithm.

B. Normalized Cuts (Ncuts)

Recently, spectral graph methods for clustering have arisen as a solid approach to data clustering, and have grown in popularity [64]–[68]. Spectral clustering arises from concepts in spectral graph theory, where the connection between graphs and matrices provides powerful tools to tackle graph theoretical and linear algebra problems.

Spectral clustering, constructs a weighted graph, $M(\mathbf{W}, Q)$, with n nodes $Q = [q_1, ..., q_n]$, where the i^{th} node represents the sample $\mathbf{d}_i$, and each weighted edge, w_{ij}, measures the similarity between two samples, $\mathbf{d}_i$ and $\mathbf{d}_j$. Once the affinity matrix $\mathbf{W} \in \Re^{n\times n}$ is computed, the clustering problem can be seen as a graph cut problem [69], where the goal is to find a partition of the graph that minimizes a particular cost function. A popular cost function is

$$cut(M) = \sum_{q_i\in R, q_j\in Q-R} w_{ij}, \quad (51)$$

where q_i denotes the i^{th} node of the Graph M, Q represents all nodes and R is a subset of the nodes. Finding the optimal cut is an NP complete problem, and spectral graph methods use relaxations to find an approximate solution. However, minimization of this objective function, Eq. (51), favors partitions containing isolated nodes, and better measures such as Ncuts [66] or ratio-cuts [70] have been proposed. Ncuts [66] finds a low dimensional embedding better suited for clustering by computing the eigenvector with the second smallest eigenvalue of the normalized Laplacian $\mathbf{S}^{-\frac{1}{2}}\mathbf{L}\mathbf{S}^{-\frac{1}{2}}$, where $\mathbf{L} = \mathbf{S} - \mathbf{W} \in \Re^{n\times n}$, and $\mathbf{S}$ is a diagonal matrix whose elements are the sum of the rows of $\mathbf{W}$, that is, $s_{ii} = \sum_j w_{ij}$. See [68], [71]–[74] for a comparison of different spectral clustering algorithms.

Recently, [65], [75] established the connection between kernel k-means and Ncuts. In this section, we follow a simpler derivation of the same idea with our compact matrix notation [10], and relate it to KPCA [34]. Consider Eq. (1), where $\mathbf{\Gamma} = \boldsymbol{\phi}(\mathbf{D})$, $\mathbf{\Upsilon} = \mathbf{I}_n, \mathbf{W}_r = \mathbf{I}_d$, $\mathbf{W}_c = diag(\mathbf{\Gamma}^T\mathbf{\Gamma}\mathbf{1}_n)^{-\frac{1}{2}}$, the weighted kernelized version of k-means, Eq. (50), is:

$$E_{13}(\mathbf{B}, \mathbf{A}) = ||(\mathbf{\Gamma} - \mathbf{B}\mathbf{A}^T)\mathbf{W}_c||_F^2. \quad (52)$$

Recall that the weight matrix $\mathbf{W}_c$ weights each sample (columns of $\mathbf{\Gamma}$) differently. In this case minimizing Eq. (52) is equivalent to maximizing Eq. (4), that is:

$$E_{13}(\mathbf{A}) \propto tr((\mathbf{A}^T\mathbf{W}_c^2\mathbf{A})^{-1}\mathbf{A}^T\mathbf{W}_c^2\mathbf{K}\mathbf{W}_c^2\mathbf{A}), \quad (53)$$

where $\mathbf{K} = \mathbf{\Gamma}^T\mathbf{\Gamma}$ is the standard affinity matrix in spectral graph methods. After a change of variable $\mathbf{Z} = \mathbf{A}^T\mathbf{W}_c$, Eq. (53) can be expressed as:

$$E_{13}(\mathbf{Z}) \propto tr((\mathbf{Z}\mathbf{Z}^T)^{-1}\mathbf{Z}\mathbf{W}_c\mathbf{K}\mathbf{W}_c\mathbf{Z}^T). \quad (54)$$

Eq. (54) is the same expression used in NCuts [66], considering $\mathbf{W}_c = \mathbf{S}^{-\frac{1}{2}}$ and $\mathbf{K} = \mathbf{\Gamma}^T\mathbf{\Gamma} \in \Re^{n\times n}$. Once again, with a LS view of Ncuts, the connection with KPCA (without centering the data in the feature space) becomes evident. Moreover, the LS formulation is more general because it allows for different kernels and weights. For instance, the weight matrix could be used to reject the influence of a pair of data points with unknown similarity (i.e., missing data).

Typically, after the embedding is found, there are several multiway cut algorithms to cluster the data in the embedded space [71], [76]. See [68], [73] for a review of rounding methods and more advanced rounding strategies. In related work, Rahimi and Recht [77] showed how Ncuts [66], originally presented as a graph-theoretic algorithm, can be framed as a regression problem. They also pointed out the problems of sensitivity of Ncuts to outliers. Zass and Shashua [72] showed the importance of normalizing the affinity matrix in spectral clustering. Important connections have also been made between clustering and manifold learning. Bengio *et al.* [59] also showed the connection between the continuous formulation of spectral embedding and KPCA through learning eigenfunctions.

C. Discriminative Cluster Analysis (DCA)

The k-means algorithm is a widely used technique for clustering due to its easiness of programming and good performance; however, k-means suffers from several drawbacks: it is sensitive to initial conditions, only optimal for hyper-spherical clusters and does not remove undesirable features for clustering. A common approach to clustering high-dimensional data with k-means is to project the data onto the space spanned by the principal components. Clustering in the space of principal components has been shown to be equivalent to the spectral relaxation of k-means [64]. However, the space of principal components does not necessarily improve the separability of the clusters. This section describes Discriminative Cluster Analysis (DCA) [10] that computes a low dimensional discriminative space that encourages cluster separability, and provides a more natural solution to the rounding problem in spectral clustering. DCA simultaneously performs dimensionality reduction and clustering, improving efficiency and clustering performance in comparison with generative approaches (e.g., PCA). Recently, Ding and Li [78], Bach and Harchaoui [79], and Ye *et al.* [80] have further shown advantages of discriminative clustering methods versus generative approaches.

Consider again the LS formulation for LDA, Eq. (27), and assume zero mean data ($\mathbf{D}\mathbf{1}_n = \mathbf{0}$):

$$E_{14}(\mathbf{B}, \mathbf{A}) = ||(\mathbf{G}^T\mathbf{G})^{-\frac{1}{2}}(\mathbf{G}^T - \mathbf{B}\mathbf{A}^T\mathbf{D})||_F^2, \quad (55)$$

where recall that $\mathbf{G} \in \Re^{n\times c}$ is an indicator matrix such that $\sum_j g_{ij} = 1$, $g_{ij} \in \{0, 1\}$, and g_{ij} is 1 if $\mathbf{d}_i$ belongs to class j, and 0 otherwise. After substituting the optimal $\mathbf{B}$ value, Eq. (55) can be rewritten as Eq. (4)

$$E_{14}(\mathbf{A}) \propto tr\big((\mathbf{A}^T\mathbf{D}\mathbf{D}^T\mathbf{A})^{-1}(\mathbf{A}^T\mathbf{D}\mathbf{G}(\mathbf{G}^T\mathbf{G})^{-1}\mathbf{G}^T\mathbf{D}^T\mathbf{A})\big). \quad (56)$$

Eq. (56) is the basis for DCA. In LDA, $\mathbf{G}$ is given (supervised problem), but in clustering (unsupervised) $\mathbf{G}$ is not known. Based upon this observation, the goal of DCA is to jointly optimize over the clustering variable, $\mathbf{G}$, and the dimensionality reduction matrix, $\mathbf{A}$. In the first step, given an initial estimate of a local similarity matrix, $\mathbf{G}(\mathbf{G}^T\mathbf{G})^{-1}\mathbf{G}^T$, DCA optimizes $\mathbf{A}$ finding a low dimensional projection well suited for clustering (i.e. the samples that belong to the same class are grouped together and the means of the classes are far from each other). Later, DCA performs a "soft" clustering in this discriminative embedding. The result of the clustering is feedback into the dimensionality reduction step, and this procedure is repeated until convergence. See [10] for more details on the optimization.

Finally, it is worth pointing out that the optimization problem in Eq. (56) is similar in spirit to recent work on clustering with non-negative matrix factorization [65], [81], [82]. However, DCA optimizes a discriminative criterion rather than a generative one, and simultaneously optimizes dimensionality reduction and clustering.

VIII. Least-Squares Extensions of CA Methods

In previous sections, we have related many CA methods to the LS-WKRRR problem. This section relates the fundamental equation of CA, Eq. (1), with other CA methods such as Non-negative Matrix Factorization (NMF), Probabilistic PCA (PPCA), and Regularized LDA (RLDA), and proposes new extensions of CA methods such as Dynamic Coupled Component Analysis (DCCA), Aligned Cluster Analysis (ACA), Canonical Time Warping (CTW), Filtered Component Analysis (FCA), Parameterized Kernel Principal Component Analysis (PaKPCA), and Feature Selection for Subspace Analysis (FSSA). These extensions typically involve adding extra constraints on $\mathbf{A}$ or $\mathbf{B}$, additional terms or new operators into the LS-WKRRR framework. It is important to notice that the LS formulation proposed in this paper allows several of these extensions, and it is unclear how they could be derived from an eigen-formulation.

A. Non-negative Matrix Factorization (NMF)

Early work on NMF was performed in the area of chemometrics, known under the name of "self modeling curve resolution" [83]. It followed by work on positive matrix factorization done by Pattero and Tapper [84] and Shen and Israel [85] proposing positive extensions of Factor Analysis and PCA with application to environmental problems. Later, Lee and Seung [82] further investigated the properties of the fitting algorithm for two types of factorizations, and applied it to visual data. Recently, Ding

et al. [81] have shown the relation between NMF and spectral clustering.

The main difference between standard NMF and PCA is that NMF constrains the matrices $\mathbf{A}$ and $\mathbf{B}$ to be non-negative, i.e. all elements must be equal to or greater than zero. As PCA, NMF can also be derived from Eq. (1), imposing positive constraints on $\mathbf{A}$ and $\mathbf{B}$. That is,

$$E_{15}(\mathbf{B},\mathbf{A}) = ||\mathbf{D} - \mathbf{B}\mathbf{A}^T||_F^2 \quad s.t. \quad \mathbf{A} \geqslant 0 \quad \mathbf{B} \geqslant 0 \tag{57}$$

Unfortunately NMF does not have a closed-form solution as a GEP. Most successful approaches to optimize NMF make use of bound optimization algorithms. See [65], [82] for more details.

B. Probabilistic PCA (PPCA)

Probabilistic PCA (PPCA) [13], [33], [86] is a probabilistic extension of PCA and a general case of Factor Analysis (FA) [52]. This section shows how the maximum-likelihood estimation for the parameters of the PPCA model, when noise is isotropic, can also be obtained from a least squares formulation.

Let us assume zero mean data and consider Eq. (1) with the following values: $\mathbf{\Gamma} = \mathbf{S}_t, \mathbf{W}_r = \mathbf{I}_d, \mathbf{W}_c = \mathbf{I}_n, \mathbf{A} = \mathbf{U}\mathbf{\Lambda}^{\frac{1}{2}}, \mathbf{B} = \mathbf{U}\mathbf{\Lambda}^{\frac{1}{2}}$. After adding an extra term $\sigma^2\mathbf{I}_d$ to the factorization, Eq. (1) results in

$$E_{16}(\mathbf{U},\mathbf{\Lambda},\sigma^2) = ||\mathbf{S}_t - \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T - \sigma^2\mathbf{I}_d||_F^2, \tag{58}$$

where $\mathbf{U} \in \Re^{d\times k}$ and $\mathbf{\Lambda} \in \Re^{k\times k}$ is a diagonal matrix. The necessary conditions for the minima of Eq. (58) w.r.t $\mathbf{U}, \sigma^2$ are

$$\mathbf{U}\mathbf{\Lambda} = (\mathbf{S}_t - \sigma^2\mathbf{I}_d)\mathbf{U},$$
$$\sigma^2 = \frac{tr(\mathbf{S}_t - \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T)}{tr(\mathbf{I}_d)} = \frac{tr(\mathbf{S}_t - \mathbf{U}\hat{\mathbf{\Lambda}}\mathbf{U}^T)}{d-k}.$$

The optimal $\mathbf{U}$ corresponds to the leading eigenvectors of $\mathbf{S}_t$. Observe that $\mathbf{S}_t$ and $(\mathbf{S}_t - \sigma^2\mathbf{I}_d)$ have the same eigenvectors and the eigenvalues of $(\mathbf{S}_t - \sigma^2\mathbf{I}_d)$ are the eigenvalues of $\mathbf{S}_t$ minus σ^2. $\hat{\mathbf{\Lambda}}$ represents the k first eigenvalues of the covariance matrix $\mathbf{S}_t$, and $\mathbf{\Lambda} = \hat{\mathbf{\Lambda}} - \sigma^2\mathbf{I}_k$ the first k eigenvalues of $(\mathbf{S}_t - \sigma^2\mathbf{I}_d)$. σ^2 corresponds to the average residual eigenvalue. These are the same expressions as the maximum-likelihood estimation for PPCA [13], [33], [86].

C. Regularized LDA (RLDA)

The LDA solution is typically ill-posed when the number of samples is smaller than the number of features, i.e. the SSS problem. In order to transform the ill-posed problem into a well-posed one, a regularization term is often added. Several research papers have addressed the benefits of Regularized LDA (RLDA), see [45], [46] for a review. This section shows how to derive RLDA from a LS-WKRRR formulation.

Consider regularizing Eq. (1), as

$$E_{17}(\mathbf{A},\mathbf{B}) = ||\mathbf{W}_r(\mathbf{\Gamma} - \mathbf{B}\mathbf{A}^T\mathbf{\Upsilon})\mathbf{W}_c||_F^2 + \lambda||\mathbf{W}_r\mathbf{B}\mathbf{A}^T||_F^2. \tag{59}$$

The necessary conditions on $\mathbf{B}$ for the minimum of Eq. (59) are

$$\frac{\partial E_{17}}{\partial \mathbf{B}} = \mathbf{W}_r^2\mathbf{B}\mathbf{A}^T\mathbf{\Upsilon}\mathbf{W}_c^2\mathbf{\Upsilon}^T\mathbf{A} - \mathbf{W}_r^2\mathbf{\Gamma}\mathbf{W}_c^2\mathbf{\Upsilon}^T\mathbf{A} + \lambda\mathbf{W}_r^2\mathbf{B}\mathbf{A}^T\mathbf{A} = \mathbf{0}. \tag{60}$$

Substituting the optimal $\mathbf{B} = \mathbf{\Gamma}\mathbf{W}_c^2\mathbf{\Upsilon}^T\mathbf{A}\big(\mathbf{A}^T(\mathbf{\Upsilon}\mathbf{W}_c^2\mathbf{\Upsilon}^T + \lambda\mathbf{I}_d)\mathbf{A}\big)^{-1}$ of Eq. (60) into Eq. (59) leads to the following expression:

$$E_{17}(\mathbf{A}) \propto tr\big((\mathbf{A}^T(\mathbf{\Upsilon}\mathbf{W}_c^2\mathbf{\Upsilon}^T + \lambda\mathbf{I}_d)\mathbf{A})^{-1} (\mathbf{A}^T\mathbf{\Upsilon}\mathbf{W}_c^2\mathbf{\Gamma}^T\mathbf{W}_r^2\mathbf{\Gamma}\mathbf{W}_c^2\mathbf{\Upsilon}^T\mathbf{A})\big). \tag{61}$$

Eq. (61) is the generalized weighted expression for RLDA. If we consider $\mathbf{\Gamma} = \mathbf{G}^T$, $\mathbf{W}_r = (\mathbf{G}^T\mathbf{G})^{-\frac{1}{2}}$, $\mathbf{\Upsilon} = \mathbf{D}$ and $\mathbf{W}_c = \mathbf{I}_n$ in Eq. (59), Eq. (61) is equivalent to standard RLDA [45], [46].

D. Dynamic Coupled Component Analysis (DCCA)

This section describes Dynamic Coupled Component Analysis (DCCA) [21], an extension of the fundamental equation of CA, Eq. (1), that generalizes CCA to learn correlations between two time series $\mathbf{D} \in \Re^{d_d\times n}$ and $\mathbf{X} \in \Re^{d_x\times n}$. DCCA minimizes

$$E_{18}(\mathbf{B}_1,\mathbf{B}_2,\mathbf{A}) = ||\mathbf{D} - \mathbf{B}_1\mathbf{A}||_F^2 + \lambda_1||\mathbf{A} - \mathbf{B}_2^T\mathbf{X}||_F^2 + \lambda_2\sum_{i=1}^n||\mathbf{a}_i - \mathbf{R}\mathbf{a}_{i-1}||_2^2, \tag{62}$$

where $\mathbf{A} \in \Re^{k\times n}$ contains the projected coefficients from the dataset $\mathbf{X}$ that are maximally correlated with $\mathbf{D}$. $\mathbf{B}_1 \in \Re^{d_d\times k}$ spans the column space of $\mathbf{D}$, and $\mathbf{B}_2 \in \Re^{d_x\times k}$ is a basis that preserves the correlations between $\mathbf{D}$ and $\mathbf{X}$. $\mathbf{R} \in \Re^{k\times k}$ is a matrix that couples the coefficients $\mathbf{a}_i$ over time and temporally regularizes the solution. Eq. (62) is similar to CCA, Eq. (34), with three main differences: (1) the dynamic term couples the correlations through time, (2) it provides an uncertainty value for the coefficients $\mathbf{A}$ controlled by λ_1, (3) the predicted variable ($\mathbf{D}$) is not normalized.

E. Aligned Cluster Analysis (ACA)

This section describes Aligned Cluster Analysis (ACA) [87], an extension of kernel k-means and SC for time series clustering and embedding. Given a sequence $\mathbf{D} = [\mathbf{d}_1,\cdots,\mathbf{d}_n] \in \Re^{d\times n}$ with n samples, ACA decomposes $\mathbf{D}$ into m disjointed segments, each of which corresponds to one of k temporal clusters. The i^{th} segment, $\mathbf{Z}_i = [\mathbf{d}_{s_i},\cdots,\mathbf{d}_{s_{i+1}-1}] = \mathbf{D}_{[s_i,s_{i+1})} \in \Re^{d\times n_i}$, is composed of samples that begin at position s_i and end at $s_{i+1}-1$. The length of the segment is constrained as $n_i = s_{i+1} - s_i \leq n_{\max}$. $n_{\max}$ represents the maximum length of the segment and it controls the temporal granularity of the factorization. An indicator matrix $\mathbf{A} \in \{0,1\}^{k\times m}$ assigns each segment to a cluster; $a_{ci} = 1$ if $\mathbf{Z}_i$ belongs to cluster c.

ACA combines kernel k-means or SC with Dynamic Time Alignment Kernel (DTAK) [88] to achieve temporal clustering by minimizing

$$E_{19}(\mathbf{A},\mathbf{B},\mathbf{s}) = ||[\boldsymbol{\phi}(\mathbf{Z}_1)\ \cdots\ \boldsymbol{\phi}(\mathbf{Z}_m)] - \mathbf{B}\mathbf{A}||_F^2 = \sum_{c=1}^k\sum_{i=1}^m a_{ci}\underbrace{||\boldsymbol{\phi}(\mathbf{Z}_i) - \mathbf{b}_c||_2^2}_{dist_c(\mathbf{Z}_i)}, \tag{63}$$

where $dist_c(\mathbf{Z}_i)$ refers to the distance between the i^{th} segment and the center of class c. $\boldsymbol{\phi}(\cdot)$ is a mapping such that, $\tau_{ij} = \boldsymbol{\phi}(\mathbf{Z}_i)^T\boldsymbol{\phi}(\mathbf{Z}_j)$ is the DTAK. Observe that if we remove the variable $\mathbf{s}$, ACA is equivalent to kernel k-means and SC, that is, $||\boldsymbol{\phi}(\mathbf{D}) - \mathbf{B}\mathbf{A}||_F^2$ s.t. $\mathbf{A}^T\mathbf{1}_k = \mathbf{1}_n$ and $a_{ij} \in \{0,1\}$. There are two main differences between kernel k-means and ACA: (1) ACA defines a distance between segments, whereas kernel k-means only defines distances between samples, (2) a new set of variables,

$\mathbf{s}$, is introduced to optimize over the start and end for each of the segments. ACA is iteratively minimized with an alternating strategy, using Dynamic Programming to optimize over $\mathbf{s}$ and kernel k-means or SC to solve for $\mathbf{G}$. See [87] for more details.

F. Canonical Time Warping (CTW)

This section reviews Canonical Time Warping (CTW) [89], an extension of CCA for spatio-temporal alignment of two signals $\mathbf{D} \in \Re^{d_d \times n_d}$ and $\mathbf{X} \in \Re^{d_x \times n_x}$ with different number of samples and features.

Dynamic time warping (DTW) has been a frequent approach to align time series. DTW minimizes the following least-squares error function [89]

$$E_{20}(\mathbf{W}_x, \mathbf{W}_y) = \|\mathbf{D}\mathbf{W}_d^T - \mathbf{X}\mathbf{W}_x^T\|_F^2, \tag{64}$$

where $\mathbf{W}_d \in \{0,1\}^{m \times n_d}$ and $\mathbf{W}_x \in \{0,1\}^{m \times n_x}$ are binary matrices such that the sum of the rows is 1. $\mathbf{W}_d$ and $\mathbf{W}_x$ can only replicate samples of the original signal $\mathbf{D}$ and $\mathbf{X}$. Observe that Eq. (64) is similar to the CCA's objective, Eq. (38). CCA applies linear transformations to the rows (features) while DTW replicates columns (samples or time).

A major limitation of DTW is that it does not have a feature weighting mechanism to remove irrelevant dimensions for alignment. Moreover, it is unclear how to use DTW to align two datasets with a different number of features (e.g., video and motion capture data). In order to add a feature weighting mechanism, CTW adds a linear transformation in the feature space as CCA does. CTW combines DTW (Eq. 64) and CCA (Eq. 38) by minimizing

$$E_{21}(\mathbf{W}_x, \mathbf{W}_d, \mathbf{V}_x, \mathbf{V}_d) = \|\mathbf{V}_d^T\mathbf{D}\mathbf{W}_d^T - \mathbf{V}_x^T\mathbf{X}\mathbf{W}_x^T\|_F^2, \tag{65}$$

where $\mathbf{V}_x \in \Re^{d_x \times b}$, $\mathbf{V}_d \in \Re^{d_d \times b}$, $b \leq \min(d_x, d_d)$ project the sequences in the same coordinate system. On the other hand, $\mathbf{W}_x$ and $\mathbf{W}_d$ stretch the signals in time. Similar to CCA, to make CTW invariant to translation, rotation and scaling, we impose the following constraints: (i) $\mathbf{X}\mathbf{W}_x^T\mathbf{1}_m = \mathbf{0}_{d_x}$, $\mathbf{D}\mathbf{W}_d^T\mathbf{1}_m = \mathbf{0}_{d_d}$, (ii) $\mathbf{V}_x^T\mathbf{X}\mathbf{D}_x\mathbf{X}^T\mathbf{V}_x = \mathbf{V}_d^T\mathbf{D}\mathbf{D}_d\mathbf{D}^T\mathbf{V}_d = \mathbf{I}_b$ and (iii) $\mathbf{V}_x^T\mathbf{X}\mathbf{W}\mathbf{D}^T\mathbf{V}_d$ to be a diagonal matrix, where $\mathbf{D}_x = \mathbf{W}_x^T\mathbf{W}_x$, $\mathbf{D}_d = \mathbf{W}_d^T\mathbf{W}_d$ and $\mathbf{W} = \mathbf{W}_x^T\mathbf{W}_d$. CTW extends previous work on CCA by adding temporal alignment and generalizes DTW by allowing a feature selection and dimensionality reduction mechanism for signals of different dimensions. More details on CTW are given in [89].

G. Filtered Component Analysis (FCA)

Multiband representations of images (e.g., [90], [91]) have proven to be useful in many computer vision problems such as robust image matching [91], visual learning [90] and detection [92]. Learning image filters can also be casted in the LS-WKRRR framework.

Given a set of training images, $\mathbf{D} \in \Re^{d \times n}$, where each sample $\mathbf{d}_i$ is an image, the aim of Filtered Component Analysis (FCA) [92] is to find a set of filters $\mathbf{B}^{1,\cdots,F}$ that decorrelate the spatial statistics of $\mathbf{D}$. Consider Eq. (1), where $\boldsymbol{\Upsilon} = \mathbf{I}_n$, $\mathbf{W}_r = \mathbf{I}_d$, $\mathbf{W}_c = \mathbf{I}_n$, $\boldsymbol{\Gamma} = \mathbf{D}$, $\mathbf{A} = \mathbf{I}_d$, $\mathbf{D}\mathbf{1} = \mathbf{0}$, and the subtraction operator is replaced by a convolution (denoted by $*$)

$$E_{22}(\mathbf{B}) = \sum_{i=1}^{n} ||\mathbf{d}_i * \mathbf{B}||_2^2 \ . \tag{66}$$

Without imposing any constraints on $\mathbf{B} \in \Re^{f_x \times f_y}$ (filter coefficients), maximizing $E_2 2$ w.r.t. $\mathbf{B}$ is unbounded. To avoid this trivial solution, we impose that the sum of squared coefficients is 1, i.e. $||\mathbf{B}||_F^2 = 1$. In this case, Eq. (66) has a solution as a GEP given by the leading eigenvectors of $\mathbf{Q}$, where $\mathbf{Q} = \sum_{i=1}^{n}\sum_{(x,y)} \mathbf{d}_i^{(x,y)}{\mathbf{d}_i^{(x,y)}}^T$. (x, y) represents the domain where the convolution is valid and $\mathbf{d}_i^{(x,y)}$ is a patch of size (f_x, f_y) centered at the coordinates (x, y) of the image $\mathbf{d}_i$. The matrix $\mathbf{Q}$ can be computed efficiently in space or frequency using the autocorrelation function of $\mathbf{d}_i$ or the integral image.

To learn a filter bank $(\mathbf{B}^{1,\cdots,F})$ in a discriminative manner, FCA maximizes

$$E_{23} = \sum_{f=1}^{F}\Big(\sum_{i=1}^{n}||\mathbf{d}_i * \mathbf{B}^f||_2^2 - \lambda\sum_{j=1}^{n_2}||\mathbf{d}_j^n * \mathbf{B}^f||_2^2\Big) \ , \tag{67}$$

where $\mathbf{d}_i$ is the i^{th} sample of the positive class and $\mathbf{d}_j^n$ denotes the j^{th} sample of the negative class (e.g., background). Eq. (67) can be solved in closed-form as a GEP, see [92] for more details.

It is interesting to consider the analogy with PCA. PCA computes the leading eigenvectors of $\mathbf{S}_t = \sum_{i=1}^{n}\mathbf{d}_i\mathbf{d}_i^T$ whereas FCA computes the leading eigenvectors of $\mathbf{Q} = \sum_{i=1}^{n}\sum_{(x,y)}\mathbf{d}_i^{(x,y)}{\mathbf{d}_i^{(x,y)}}^T$. While PCA finds the directions of maximum variation of the covariance matrix, FCA finds the directions of maximum variation of the sum of all overlapping patches. PCA decorrelates the signal with the covariance of the data, whereas FCA decorrelates the spatial statistics. Adding non-linear layers within the convolutional architecture [93] can extract higher-order moments of the signal in a discriminative manner.

H. Parameterized Kernel Principal Component Analysis (PaKPCA)

Learning a subspace invariant to possible normalizations of the data is of interest in many statistical problems, for instance, learning a model of visual data invariant to geometric transformations (e.g., rotation, scale). Several researchers have proposed learning visual appearance models invariant to geometric transformations [94]–[99]. This section describes Parameterized Kernel Principal Component Analysis (PaKPCA), an extension of Eq. (1), to learn a non-linear shape and appearance model invariantly to rigid and non-rigid geometric transformations [100].

We parameterize an image $\mathbf{d} \in \Re^{d \times 1}$ with a geometric transformation $\mathbf{f}(\mathbf{x}, \mathbf{r})$ [101], [102], $\mathbf{d}(\mathbf{f}(\mathbf{x}, \mathbf{r}))$. In the case of an affine transformation $\mathbf{f}(\mathbf{x}, \mathbf{r})$ is

$$\mathbf{f}(\mathbf{x}, \mathbf{r}) = \begin{pmatrix} r_1 \\ r_2 \end{pmatrix} + \begin{pmatrix} r_3 & r_4 \\ r_5 & r_6 \end{pmatrix}\begin{pmatrix} x \\ y \end{pmatrix}, \tag{68}$$

where $\mathbf{r} = (r_1, r_2, r_3, r_4, r_5, r_6)$ are the affine parameters and $\mathbf{x} = (x_1, y_1, \cdots, x_n, y_n)$ is a vector containing the coordinates of the pixels of a given image region. Alternatively, non-rigid motion can be added in a straightforward manner by changing the definition of $\mathbf{f}$. Consider $\mathbf{f}(\mathbf{B}^s\mathbf{c}^s, \mathbf{r}) = \mathbf{f}(\sum_{l=1}^{k} c_l^s\mathbf{b}_l^S, \mathbf{r})$, where $\mathbf{B}^s$ is a non-rigid shape basis learned by computing PCA on a set of registered shapes [103]. The coefficient $\mathbf{c}^s$ represents the non-rigid parameters, and $\mathbf{r}$ denotes the rigid parameters. In this case, $\mathbf{f}(\mathbf{B}^s\mathbf{c}^s, \mathbf{r})$ will model rigid and non-rigid motion.

Consider Eq. (1), with the following values: $\boldsymbol{\Gamma} = \mathbf{D}, \mathbf{W}_r = \mathbf{I}_d, \mathbf{W}_c = \mathbf{I}_n$, and parameterize $\mathbf{d}_i$ with a geometric transforma-

tion $\mathbf{f}(\mathbf{B}^s\mathbf{c}^s, \mathbf{r}_i)$

$$E_{24}(\mathbf{C}^S, \mathbf{A}, \mathbf{B}, \mathbf{R}) = \sum_{i=1}^{n} ||\boldsymbol{\phi}\big(\mathbf{d}_i(\mathbf{f}(\mathbf{B}^s\mathbf{c}_i^s, \mathbf{r}_i))\big) - \mathbf{B}\mathbf{a}_i||_2^2 \quad (69)$$

with the constraints that $\mathbf{B}^T\mathbf{B} = \mathbf{I}_k$. Observe that the previous equation is equivalent to KPCA if $\mathbf{f}(\mathbf{B}^s\mathbf{c}_i^s, \mathbf{r}_i) = \mathbf{x}$ (image coordinates), that is, if the geometric transformation does not change the position of the coordinates in the image. Unlike KPCA, Eq. (69) is optimized w.r.t. the subspace $\mathbf{B}, \mathbf{A}$, non-rigid coefficients $\mathbf{C}^S$ and rigid coefficients $\mathbf{R}$. For more information see [100].

I. Feature Selection for Subspace Analysis (FSSA)

A relatively unexplored problem in subspace analysis is how to select a subset of features that minimize the distance to a given subspace. Recently, Roig *et al.* [104] proposed an extension of Eq. (1) to solve a relaxed version of the subspace feature selection problem. Consider Eq. (1), where $\boldsymbol{\Upsilon} = \mathbf{I}_d, \mathbf{W}_r = \mathbf{I}_k, \mathbf{W}_c = 1, \boldsymbol{\Gamma} = vec(\mathbf{PD})$, and after subtracting the mean $\boldsymbol{\mu}$ results in:

$$E_{25}(\mathbf{P}, \mathbf{a}) = ||vec(\mathbf{PD}) - \boldsymbol{\mu} - \mathbf{B}\mathbf{a}||_2^2 \quad (70)$$
$$s.t.\ p_{ij} \in \{0,1\}, \quad \textstyle\sum_j p_{ij} = 1\ \forall\ i, \quad \sum_i p_{ij} = \{0,1\}\ \forall\ j$$

where $\mathbf{D} \in \Re^{d\times r}$ is a matrix in which each row contains r features, and d denotes instances of a given feature. For instance, in the case of the features being two dimensional landmarks $r = 2$ for the (x, y), or in the case of the SIFT descriptor [105] $r = 128$. $\mathbf{P} \in \Re^{k\times d}$ is an indicator matrix such that $\sum_j p_{ij} = 1\ \forall i$, $p_{ij} \in \{0, 1\}$, and p_{ij} is 1 if the feature i belongs to the subset of k points that minimize the distance to the subspace. The sum of the columns of $\mathbf{P}$ can be either 0 or 1, that is: $\sum_i p_{ij} = \{0, 1\}\ \ \forall j$.

The objective of the optimization is to simultaneously find the subset of k features (selected by $\mathbf{P}$) and the subspace coefficients ($\mathbf{a}$) that minimize the error E_{25} in Eq. (70). To reduce the number of parameters, [104] computed the optimal value of $\mathbf{a} = \mathbf{B}^T(vec(\mathbf{PD}) - \boldsymbol{\mu})$ and after substituting this expression into Eq. (70) resulted in

$$E_{25}(\mathbf{P}) = ||(\mathbf{I}_{rk} - \mathbf{B}\mathbf{B}^T)(vec(\mathbf{PD}) - \boldsymbol{\mu})||_2^2 \propto$$
$$-\tfrac{1}{2}vec(\mathbf{P})^T\mathbf{Q}vec(\mathbf{P}) + \mathbf{b}^T vec(\mathbf{P}), \quad (71)$$

where $\mathbf{Q} = (\mathbf{D} \otimes \mathbf{I}_k)\mathbf{H}^T\mathbf{H}(\mathbf{D} \otimes \mathbf{I}_k)^T \in \Re^{kn\times kn}$, $\mathbf{H} = (\mathbf{I}_{rk} - \mathbf{B}\mathbf{B}^T) \in \Re^{kr\times kr}$, and $\mathbf{b} = (\mathbf{D} \otimes \mathbf{I}_k)\mathbf{H}^T\mathbf{H}\boldsymbol{\mu} \in \Re^{kn\times 1}$. [104] proposed a quadratic programming solution to minimize Eq. (71), and showed how it outperforms greedy [106], and naive gradient-descent approaches [104].

IX. Conclusions and Future Work

This paper shows that the LS-WKRRR is the generative model for several CA methods. In particular, we have shown how the fundamental equation of CA, Eq. (1), relates to PCA, LDA, CCA, LE, k-means, spectral methods, and their regularized and kernel extensions. We have derived the coupled system of eigen-equations that results from finding the critical points of Eq. (1), and suggested several numerical optimization schemes. The LS formulation of CA has several advantages:

- allows understanding the commonalities and differences between several CA methods, as well as the intrinsic relationships,
- helps to understand normalization factors in CA methods,
- suggests new optimization strategies for CA methods,
- avoids numerical problems of existing eigen-methods for rank deficient matrices (e.g., SSS problem),
- allows many extensions of CA methods.

We have derived weighted extensions for PCA, LDA, CCA, and kernel extensions. In addition, we have shown that several extensions of the LS-WKRRR derive into novel techniques such as DCA, DCCA, ACA, CTW, FCA, PaKPCA, and FSSA.

There exists a number of other CA techniques that are closely related to the fundamental equation of CA. An approximation to Independent Component Analysis can be derived from Eq. (15), by imposing that the coefficients $\mathbf{A}$ follow distributions with heavy tails (i.e. high kurtosis) [107]. Williams [108] showed that metric Multidimensional Scaling can be interpreted as KPCA if the kernel function is isotropic. Other techniques such as Partial Least Squares [12] or Probabilistic Latent Semantic Analysis [109] also have close connections to Eq. (1). In the LS formulation, we have implicitly assumed that the error follows an isotropic Gaussian distribution. Extensions to more complex noise models that follow the exponential family distribution have lead to the Exponential family PCA [110], [111], that can be seen as a extension of Eq. (1) changing the Frobenious norm for other metric. In Eq. (15) both regression matrices $\mathbf{A}$ and $\mathbf{B}$ are deterministic. On the other hand, Latent variable models (LVM) [13], [112] (e.g., Factor Analysis, PPCA, mixtures of Gaussians) incorporate a distribution in some of the variables, and are considered the probabilistic extensions of CA. Finally, Eq. (15) can be interpreted as a matrix factorization technique. Tensor factorization methods [113], [114] can also be considered as a generalization of PCA to more than two dimensions and can be formulated as extensions of Eq. (1). Formulating LVM and tensor factorization methods as extensions of Eq. (1) can benefit from the same advantages of the LS framework discussed in this paper.

Appendix

A: Covariance matrices in component analysis

Many CA methods can be formulated as generalized eigenvalue problems (GEPs). This appendix derives a compact matrix expression for most common covariance matrices in CA.

Let $\mathbf{D} \in \Re^{d\times n}$ be a matrix where each column is a vectorized data sample from one of c classes. d denotes the number of features and n number of samples. Some of the most common CA covariance matrices can be conveniently expressed in matrix form as [115]:

$$\mathbf{S}_t = \frac{1}{n-1}\sum_{j=1}^{n}(\mathbf{d}_j - \mathbf{m})(\mathbf{d}_j - \mathbf{m})^T = \frac{1}{n-1}\mathbf{D}\mathbf{P}_t\mathbf{D}^T,$$

$$\mathbf{S}_w = \frac{1}{n-1}\sum_{i=1}^{c}\sum_{\mathbf{d}_j\in C_i}(\mathbf{d}_j - \mathbf{m}_i)(\mathbf{d}_j - \mathbf{m}_i)^T = \frac{1}{n-1}\mathbf{D}\mathbf{P}_w\mathbf{D}^T,$$

$$\mathbf{S}_b = \frac{1}{n-1}\sum_{i=1}^{c}n_i(\mathbf{m}_i - \mathbf{m})(\mathbf{m}_i - \mathbf{m})^T = \frac{1}{n-1}\mathbf{D}\mathbf{P}_b\mathbf{D}^T,$$

where $\mathbf{m} = \frac{1}{n}\mathbf{D}\mathbf{1}_n$ is the mean vector, $\mathbf{m}_i$ is the mean vector for class i, n_i denotes the number of samples for class i, and $\mathbf{P}_i$ are projection matrices (i.e. $\mathbf{P}_i^T = \mathbf{P}_i$ and $\mathbf{P}_i^2 = \mathbf{P}_i$) with the

following expressions:

$$\mathbf{P}_t = \mathbf{I}_n - \tfrac{1}{n}\mathbf{1}_n\mathbf{1}_n^T \ , \qquad \mathbf{P}_w = \mathbf{I}_n - \mathbf{G}(\mathbf{G}^T\mathbf{G})^{-1}\mathbf{G}^T \ ,$$
$$\mathbf{P}_b = \mathbf{G}(\mathbf{G}^T\mathbf{G})^{-1}\mathbf{G}^T - \tfrac{1}{n}\mathbf{1}_n\mathbf{1}_n^T.$$

$\mathbf{G} \in \Re^{n\times c}$ is an indicator matrix such that $\sum_j g_{ij} = 1$, $g_{ij} \in \{0,1\}$, and g_{ij} is 1 if $\mathbf{d}_i$ belongs to class j, and 0 otherwise. $\mathbf{S}_b$ is the between-class covariance matrix and represents the average distance between the means of the classes. $\mathbf{S}_w$ is the within-class covariance matrix that contains information about the average compactness of each class. $\mathbf{S}_t$ is the total covariance matrix. Using this matrix expressions, it is straightforward to show that $\mathbf{S}_t = \mathbf{S}_w + \mathbf{S}_b$. The upper bounds on the ranks of $\mathbf{S}_b, \mathbf{S}_w$, and $\mathbf{S}_t$ are $\min(c-1,d), \min(n-c,d), \min(n-1,d)$, respectively.

B: Abbreviations

ACA: Aligned Cluster Analysis, **ALS**: Alternated Least-Squares, **CA**: Component Analysis, **CCA**: Canonical Correlation Analysis, **DCA**: Discriminative Cluster Analysis, **DCCA**: Dynamic Coupled Component Analysis, **DTW**: Dynamic Time Warping, **FCA**: Filtered Component Analysis, **GEPs**: Generalized Eigenvalue Problems, **KCCA**: Kernel Canonical Correlation Analysis, **KLDA**: Kernel Linear Discriminant Analysis, **KPCA**: Kernel Principal Component Analysis, **LDA**: Linear Discriminant Analysis, **LE**: Laplacian Eigenmap, **LLE**: Local Linear Embedding, **LPP**: Locality Preserving Projections, **LS**: Least-Squares, **LS-WKRRR**: Least-Squares Weighted Kernel Reduced Rank Regression, **NMF**: Non-negative Matrix Factorization, **Ncuts**: Normalized Cuts, **PaKPCA**: Parameterized Kernel Principal Component Analysis, **PCA**: Principal Component Analysis, **PPCA**: Probabilistic PCA, **SC**: Spectral Clustering, **SSS**: Small Sample Size.

Acknowledgements

This material is based upon work partially supported by the U.S. Naval Research Laboratory under Contract No. N00173-07-C-2040 and by National Science Foundation under Grant CPS-0931999. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the U.S. Naval Research Laboratory or the National Science Foundation. Thanks to Louis-Philippe Morency, Chris Ding, Andrew Fitzgibbon, Feng Zhou, Tomas Simon, Minyoung Kim, Karim Abou-Moustafa, Zaid Harchaoui, Jordi Soler for valuable comments. Thanks to the anonymous reviewers who provided helpful feedback on an earlier draft of this manuscript.

References

[1] I. T. Jolliffe, *Principal Component Analysis*. New York: Springer-Verlag, 1986.

[2] K. Pearson, "On lines and planes of closest fit to systems of points in space," *The London, Edinburgh and Dublin Philosophical Magazine and Journal*, vol. 6, pp. 559–572, 1901.

[3] H. Hotelling, "Analysis of a complex of statistical variables into principal components," *Journal of Educational Psychology*, vol. 24, no. 6, pp. 417–441, 1933.

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

[5] ——, "The use of multiple measurements in taxonomic problems," *Annals of Eugenics*, vol. 7, pp. 179–188, 1936.

[6] H. Hotelling, "Relations between two sets of variates," *Biometrika*, vol. 28, pp. 321–377, 1936.

[7] M. Belkin and P. Niyogi, "Laplacian eigenmaps for dimensionality reduction and data representation," *Neural Computation*, vol. 15, no. 6, pp. 1373–1396, 2003.

[8] X. He and P. Niyogi, "Locality preserving projections," in *Neural Information Processing Systems*, 2003.

[9] B. Mohar, "Some applications of laplace eigenvalues of graphs," *Graph Symmetry: Algebraic Methods and Applications*, pp. 225–275, 1997.

[10] F. De la Torre and T. Kanade, "Discriminative cluster analysis," in *International Conference on Machine Learning*, 2006.

[11] F. De la Torre, "A least-squares unified view of PCA, LDA, CCA and spectral graph methods," CMU-RI-TR-08-29, Robotics Institute, Carnegie Mellon University, Tech. Rep., May 2008.

[12] M. Borga, "Learning multidimensional signal processing," in *PhD Dissertation. Linkping University, Sweden*, 1998.

[13] S. Roweis and Z. Ghahramani, "A unifying review of linear gaussian models." *Neural Computation*, vol. 11, no. 2, pp. 305–345, 1999.

[14] S. Yan, D. Xu, B. Zhang, and H. Zhang, "Graph embedding: A general framework for dimensionality reduction," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 29, no. 1, pp. 40–51, 2007.

[15] K. Fukunaga, *Introduction to Statistical Pattern Recognition, Second Edition*. Academic Press. Boston, MA, 1990.

[16] T. W. Anderson, "Estimating linear restrictions on regression coefficients for multivariate normal distributions," *Ann. Math. Statist.*, vol. 12, pp. 327–351, 1951.

[17] ——, *An Introduction to Multivariate Statistical Analysis*. 2nd ed. Wiley, New York, 1984.

[18] S. S. Haykin, *Adaptive filter theory*. Prentice-Hall, 1996.

[19] L. Scharf, "The SVD and reduced rank signal processing," *Signal Processing*, vol. 25, no. 2, pp. 113–133, 2002.

[20] K. I. Diamantaras, *Principal Component Neural Networks (Theory and Applications)*. John Wiley & Sons, 1996.

[21] F. De la Torre and M. J. Black, "Dynamic coupled component analysis," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2001.

[22] P. Baldi and K. Hornik, "Neural networks and principal component analysis: Learning from examples without local minima," *Neural Networks*, vol. 2, pp. 53–58, 1989.

[23] H. Murase and S. K. Nayar, "Visual learning and recognition of 3D objects from appearance," *International Journal of Computer vision*, vol. 1, no. 14, pp. 5–24, 1995.

[24] K. J. Bathe and E. Wilson, *Numerical Methods in Finite Element Analysis*. Englewood Cliffs, NJ: Prentice-Hall.

[25] A. Buchanan and A. Fitzgibbon, "Damped newton algorithms for matrix factorization with missing data," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2005.

[26] F. De la Torre and M. J. Black, "A framework for robust subspace learning," *International Journal of Computer Vision.*, vol. 54, pp. 117–142, 2003.

[27] R. Fletcher, *Practical methods of optimization*. John Wiley and Sons., 1987.

[28] A. Blake and A. Zisserman, *Visual Reconstruction*. Massachusetts: MIT Press series, 1987.

[29] E. Oja, "A simplified neuron model as principal component analyzer," *Journal of Mathematical Biology*, vol. 15, pp. 267–273, 1982.

[30] S. Roweis, "EM algorithms for PCA and SPCA," in *Neural Information Processing Systems*, 1997.

[31] K. R. Gabriel and S. Zamir, "Lower rank approximation of matrices by least squares with any choice of weights," *Technometrics, Vol. 21, pp.*, vol. 21, pp. 489–498, 1979.

[32] H. Shum, K. Ikeuchi, and R. Reddy, "Principal component analysis with missing data and its application to polyhedral object modeling," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 17, no. 9, pp. 855–867, 1995.

[33] M. Tipping and C. M. Bishop, "Probabilistic principal component analysis," *Journal of the Royal Statistical Society B*, vol. 61, pp. 611–622, 1999.

[34] B. Schölkopf, A. J. Smola, and K.-R. Müller, "Nonlinear component analysis as a kernel eigenvalue problem," *Neural Computation*, vol. 10, no. 5, pp. 1299–1319, 1998.

[35] B. Schölkopf and A. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press series, 2002.

[36] M. Irani and P. Anandan, "Factorization with uncertainty," in *European Conference on Computer Vision*, 2000.

[37] M. J. Greenacre, *Theory and Applications of Correspondence Analysis*. London: Academic Press, 1984.

[38] I. Tsang and J. Kwok, "Distance metric learning with kernels," in *International Conference on Artificial Neural Networks*, 2003.

[39] R. Hartley and F. Schaffalitzky, "Powerfactorization: an approach to affine reconstruction with missing and uncertain data," in *Australia-Japan Advance Workshop on Computer Vision*, 2003.

[40] D. Skocaj and A. Leonardis, "Weighted and robust incremental method for subspace learning," in *International Conference on Computer Vision*, 2003.

[41] P. Aguiar, M. Stosic, and J. Xavier, "Spectrally optimal factorization of incomplete matrices," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

[42] C. Rao, "The utilization of multiple measurements in problems of biological classification," *Journal of the Royal Statistical Society - Series B*, vol. 10, no. 2, pp. 159–203, 1948.

[43] A. Martinez and A. Kak, "PCA versus LDA," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 23, no. 2, pp. 228–233, 2003.

[44] P. Belhumeur, J. Hespanha, and D. Kriegman, "Eigenfaces vs. Fisherfaces: Recognition using class specific linear projection," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 711–720, 1997.

[45] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference and Prediction.* Springer, 2001.

[46] J. Ye, "Characterization of a family of algorithms for generalized discriminant analysis on undersampled problems," *The Journal of Machine Learning Research*, vol. 6, no. 1, pp. 483 – 502, September 2005.

[47] S. Zhang and T. Sim, "Discriminant subspace analysis: A Fukunaga-Koontz approach," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 29, no. 10, pp. 1732–1745, 2007.

[48] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*. John Wiley and Sons Inc., 2001.

[49] P. Gallinari, S. Thiria, F. Badran, and F. Fogelman-Soulie, "On the relations between discriminant analysis and multilayer perceptrons," *Neural Networks*, vol. 4, pp. 349–360, 1991.

[50] J. Ye, "Least squares linear discriminant analysis," in *International Conference on Machine Learning*, 2007.

[51] S. Mika, "Kernel fisher discriminants," in *PhD thesis, University of Technology, Berlin*, 2002.

[52] K. Mardia, J. Kent, and J. Bibby, *Multivariate Analysis*. Academic Press, London, 1979.

[53] V. J. Yohai and M. S. Garcia, "Canonical variables as optimal predictors." *The Annals of Statistics*, vol. 8, no. 4, pp. 865–869, 1980.

[54] M. Tso, "Reduced-rank regression and canonical analysis." *Journal of the Royal Statistical Society. Series B.*, vol. 43, no. 2, pp. 183–189, 1981.

[55] M. Loog, R. Duin, and R. Hacb-Umbach, "Multiclass linear dimension reduction by weighted pairwise fisher criteria," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 7, no. 23, p. 762766, 2001.

[56] J. B. Tenenbaum, V. de Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 5500, no. 290, pp. 2319–2323, 2000.

[57] S. Roweis and L. Saul, "Nonlinear dimensionality reduction by locally linear embedding." *Science*, vol. 290, no. 5500, pp. 2323–2326, 2000.

[58] J. Ham, D. Lee, S. Mika, and B. Schölkopf, "A kernel view of the dimensionality reduction of manifolds," in *International Conference on Machine Learning*, 2004.

[59] Y. Bengio, P. Vincent, J. Paiement, P. Vincent, and M. Ouimet, "Learning eigenfunctions links spectral embedding and kernel PCA," *Neural Computation*, no. 16, pp. 2197–2219, 2004.

[60] X. He, D. Cai, S. Yan, and H. Zhang, "Neighborhood preserving embedding," in *International Conference on Computer Vision*, 2005.

[61] J. B. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Fifth Berkeley Symposium on Mathematical Statistics and Probability. Berkeley, University of California Press.*, 1967, pp. 281–297.

[62] A. K. Jain, *Algorithms for clustering data*. Prentice Hall, 1988.

[63] H. Zha, C. Ding, M. Gu, X. He, and H. Simon., "Spectral relaxation for k-means clustering," in *Neural Information Processing Systems*, 2001.

[64] C. Ding and X. He, "k-means clustering via principal component analysis," in *International Conference on Machine Learning*, 2004.

[65] R. Zass and A. Shashua, "A unifying approach to hard and probabilistic clustering." in *International Conference on Computer Vision*, 2005.

[66] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888–905, Aug. 2000.

[67] A. Y. Ng, M. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," in *Neural Information Processing Systems*, 2002.

[68] S. Yu and J. Shi, "Multiclass spectral clustering," in *International Conference on Computer Vision*, 2003.

[69] F. K. Chung, *Spectral Graph Theory*. Providence: CBMS Regional Conference Series in Mathemtics, vol 92, American Mathematical Society, 1997.

[70] L. Hagen and A. Kahng, "New spectral methods for ratio cut partitioning and clustering." *IEEE. Trans. on Computed Aided Desgin*, no. 11, pp. 1074–1085, 1992.

[71] D. Verma and M. Meila, "Comparison of spectral clustering methods." in *Neural Information Processing Systems*, 2003.

[72] R. Zass and A. Shashua, "Doubly stochastic normalization for spectral clustering," in *Neural Information Processing Systems*, 2006.

[73] D. Tolliver, "Spectral rounding and image segmentation," CMU-RI-TR-06-44, Robotics Institute, Carnegie Mellon University, Tech. Rep., August 2006.

[74] M. Filippone, F. Camastra, F. Masulli, and S. Rovetta, "A survey of kernel and spectral methods for clustering," *Pattern Recognition*, vol. 41, no. 1, pp. 176–190, 2008.

[75] I. S. Dhillon, Y. Guan, and B. Kulis, "Weighted graph cuts without eigenvectors: A multilevel approach," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 29, no. 11, pp. 1944–1957, 2007.

[76] Y. Weiss, "Segmentation using eigenvectors: a unifying view," in *International Conference on Computer Vision*, 1999.

[77] A. Rahimi and B. Recht, "Clustering with normalized cuts is clustering with a hyperplane," in *Statistical Learning in Computer Vision*, 2004.

[78] C. Ding and T. Li., "Adaptive dimension reduction using discriminant analysis and k-means clustering," in *International Conference on Machine Learning*, 2007.

[79] F. Bach and Z. Harchaoui, "Diffrac : a discriminative and flexible framework for clustering," in *Neural Information Processing Systems*, 2007.

[80] J. Ye, Z. Zhao, and M. Wu., "Discriminative k-means for clustering," in *Neural Information Processing Systems*, 2007.

[81] C. Ding, X. He, and H. Simon, "On the equivalence of nonnegative matrix factorization and spectral clustering." in *SIAM International Conference on Data Mining*, 2005.

[82] D. Lee and H. Seung, "Algorithms for non-negative matrix factorization," in *Neural Information Processing Systems*, 2000.

[83] W. H. Lawton and E. A. Sylvestre, "Self modeling curve resolution," *Technometrics*, vol. 13, no. 3, pp. 617–633, 1971.

[84] P. Paatero and U. Tapper, "Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values." *Envirometrics*, vol. 5, pp. 111–126, 1994.

[85] J. Shena and G. Israela, "A receptor model using a specific non-negative transformation technique for ambient aerosol," *Atmospheric Environment*, vol. 23, no. 10, pp. 2289–2298, 1989.

[86] B. Moghaddam and A. Pentland, "Probabilistic visual learning for object representation," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 137–143, July 1997.

[87] F. Zhou, F. De la Torre, and J. K. Hodgins, "Aligned cluster analysis for temporal segmentation of human motion," in *IEEE Automatic Face and Gesture Recognition*, 2008.

[88] H. Shimodaira, K.-I. Noma, M. Nakai, and S. Sagayama., "Dynamic time-alignment kernel in support vector machine," in *Neural Information Processing Systems*, 2001.

[89] F. Zhou and F. De la Torre, "Canonical time warping," in *Neural Information Processing Systems*, 2009.

[90] H. Bischof, H. Wildenauer, and A. Leonardis, "Illumination insensitive recognition using eigenspaces," *Computer Vision and Image Understanding*, vol. 1, no. 95, pp. 86 – 104, 2004.

[91] W. T. Freeman and E. H. Adelson, "The design and use of steerable filters," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 13, pp. 891–906, 1991.

[92] F. De la Torre, A. Collet, J. Cohn, and T. Kanade, "Filtered component analysis to increase robustness to local minima in appearance models," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2007.

[93] Y. LeCun and Y. Bengio, "Convolutional networks for images, speech, and time-series." *M. A. Arbib, editor, The Handbook of Brain Theory and Neural Networks. MIT Press*, 1995.

[94] B. J. Frey and N. Jojic, "Transformation-invariant clustering using the em algorithm," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 25, no. 1, pp. 1–17, 2003.

[95] F. De la Torre and M. J. Black, "Robust parameterized component analysis: theory and applications to 2d facial appearance models," *Computer Vision and Image Understanding*, vol. 91, pp. 53 – 71, 2003.

[96] E. G. Learned-Miller, "Data driven image models through continuous joint alignment," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 28, no. 2, pp. 236–250, 2006.

[97] M. Cox, S. Lucey, S. Sridharan, and J. Cohn, "Least squares congealing for unsupervised alignment of images," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2008.
[98] S. Baker, I. Matthews, and J. Schneider, "Automatic construction of active appearance models as an image coding problem," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 26, no. 10, pp. 1380 – 1384, October 2004.
[99] I. Kookinos and A. Yuille, "Unsupervised learning of object deformation models," in *International Conference on Computer Vision*, 2007.
[100] F. De la Torre and M. Nguyen, "Parameterized kernel principal component analysis: Theory and applications to supervised and unsupervised image alignment," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2008.
[101] M. J. Black and A. D. Jepson, "Eigentracking: Robust matching and tracking of objects using view-based representation," *International Journal of Computer Vision*, vol. 26, no. 1, pp. 63–84, 1998.
[102] T. F. Cootes, G. J. Edwards, and C. J. Taylor, "Active appearance models," *IEEE transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 6, pp. 681–685, 2001.
[103] T. Cootes and C. Taylor, "Statistical models of appearance for computer vision." in *Tech. Report. University of Manchester*, 2001.
[104] G. Roig, X. Boix, and F. De la Torre, "Feature selection for subspace image matching," in *2nd IEEE International Workshop on Subspace Methods*, 2009.
[105] D. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.
[106] B. Moghaddam, G. A, Y. Weiss, and S. Avidan, "Sparse regression as a sparse eigenvalue problem," *Information Theory and Applications Workshop*, 2008.
[107] B. Olshausen and D. Field, "Sparse coding with an overcomplete basis set: A strategy employed by V1?" *Vision Research*, no. 37, pp. 3311–3325, 1997.
[108] C. Williams, "On a connection between kernel PCA and metric multidimensional scaling," in *Neural Information Processing Systems*, 2001.
[109] C. Ding, T. Li, and W. Peng, "On the equivalence between non-negative matrix factorization and probabilistic latent semantic indexing," *Computational Statistics and Data Analysis*, vol. 52, pp. 3913–3927, 2008.
[110] M. Collins, S. Dasgupta, and R. Schapire, "A generalization of principal components analysis to the exponential family," in *Neural Information Processing Systems*, 2002.
[111] G. Gordon, "Generalized linear models," in *Neural Information Processing Systems*, 2002.
[112] B. S. Everitt, *An Introduction to Latent Variable Models*. London: Chapman and Hall, 1984.
[113] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," *SIAM Review*, vol. 51, pp. 455–500, 2009.
[114] J. R. Magnus and H. Neudecker, *Matrix Differential Calculus with Applications in Statistics and Econometrics*. John Wiley, 1999.
[115] F. De la Torre and T. Kanade, "Multimodal oriented discriminant analysis," in *International Conference on Machine Learning*, 2005.

Fernando De la Torre received his B.Sc. degree in Telecommunications, M.Sc. and Ph. D degrees in Electronic Engineering, respectively, in 1994, 1996 and 2002, from La Salle School of Engineering in Ramon Llull University, Barcelona, Spain. In 1997 and 2000 he became Assistant and Associate Professor in the Department of Communications and Signal Theory in La Salle School of Engineering. In 2002 he was a post doctoral researcher at Brown university (Providence, RI) and Gatsby Neuroscience Unit (London). Since 2005 he is Research Faculty in the Robotics Institute at Carnegie Mellon University. His research interests are in the fields of Computer Vision and Machine Learning. Currently, he is directing the component analysis lab (http://ca.cs.cmu.edu) and the human sensing lab (http://humansensing.cs.cmu.edu). Dr. De la Torre has co-organized the first workshop on component analysis methods for modeling, classification and clustering problems in computer vision in conjunction with CVPR'07 and the workshop on human sensing from video in conjunction with CVPR'06. He has also given several tutorials at international conferences on the use and extensions of component analysis methods.