

Chapter 1

A Unification of Component Analysis Methods

F. De la Torre

*Robotics Institute. Carnegie Mellon University.
5000 Forbes av. 211 Smith Hall. Pittsburgh, PA 15213.
e-mail: ftorre@cs.cmu.edu*

Over the last century Component Analysis (CA) methods such as Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA), Canonical Correlation Analysis (CCA), k -means and Spectral Clustering (SC) have been extensively used as a feature extraction step for modeling, classification, visualization and clustering. CA techniques are appealing because many can be formulated as eigen-problems, offering great potential for learning linear and non-linear representations of data without local minima. However, the eigen-formulation often conceals important analytic and computational drawbacks of CA techniques, such as solving generalized eigen-problems with rank deficient matrices, lacking intuitive interpretation of normalization factors, and understanding relationships between CA methods.

This chapter proposes a unified framework to formulate many CA methods as a least-squares estimation problem. We show how PCA, LDA, CCA, k -means, spectral graph methods and kernel extensions correspond to a particular instance of a least squares weighted kernel reduced rank regression (LS-KRRR). The least-squares formulation allows better understanding of normalization factors, provides a clean framework to understand the communalities and differences between many CA methods, yields efficient optimization algorithms for many CA algorithms, suggest easy derivation for on-line learning methods, and provides an easier generalization of CA techniques. In particular, we derive the matrix expressions for weighted generalizations of PCA, LDA, SC and CCA (including kernel extensions).

1.1. Introduction

Over the last century, Component Analysis (CA) methods¹ such as Kernel Principal Component Analysis (KPCA),² Linear Discriminant Analysis (LDA),³ Canonical Correlation Analysis,⁴ and Spectral Clustering (SC)⁵ have been extensively used as a feature extraction step in modeling, clustering, classification and visualization 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 handle high-dimensional data due to the *curse-of-dimensionality*, that usually requires a large number of samples to build accurate models. 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 posed as problems of 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 chapter provides a tool for analyzing, generalizing, and developing efficient algorithms to solve many CA methods. This chapter shows how Kernel PCA, Kernel LDA, Kernel CCA, k -means, and Normalized Cuts 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 least-squares (LS) problems as they arise. Preliminary versions of this work were published in a technical report.⁶

This paper recovers the spirit of three previous published papers seeking unified frameworks. Borga⁷ showed how PCA, Partial Least Squares (PLS), Canonical Correlation Analysis (CCA) and Multiple Linear Regression (MLR) can be formulated as a generalized eigen-value problems (GEPs). He proposed a gradient-descent algorithm on the Rayleigh quotient to efficiently solve the GEP. Roweis and Ghahramani⁸ 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 nonlinearities in the model,⁸ showed how Independent Component Analysis (ICA) can also be cast as an extension of a LDS. Yan *et al.*⁹ have recently proposed a unifying view of PCA, LDA, LPP, Isomap, and LDA using a graph theoretical formulation. Additionally, the authors propose Marginal Fisher Analysis, a variant of non-parametric LDA.¹⁰ This work differs from previous works in that we unified PCA, CCA, LDA, SC, and kernel generalizations with a LS-WKRRR. In addition, we propose new weighted extensions for PCA, CCA, LDA, and SC.

The rest of the chapter is organized as follows: Section 1.2 introduces the notation and formulates the typical covariance matrices of CA methods using a compact matrix formulation. Section 1.3 introduces the LS-WKRRR problem and derives the coupled generalized eigenvalue system resulting from solving it. Section 1.4 relates PCA, KPCA and weighted extensions to the LS-WKRRR. Section 1.5 shows how LDA, KLDA, CCA, KCCA and weighted extensions are a particular instance of LS-WKRRR. Section 1.6 shows the relationship between LS-WKRRR, k -means and spectral clustering. Section 1.7 finalizes the chapter presenting the conclusions.

1.2. Covariance matrices in component analysis

Many CA methods can be formulated as generalized eigenvalue problems (GEPs). This section derives a compact matrix expression for most common covariance matrices used to solve CA methods through GEPs.

Let $\mathbf{D} \in \mathfrak{R}^{d \times n}$ (see notation ^a) 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:¹¹

$$\begin{aligned}\mathbf{S}_t &= \frac{1}{n-1} \sum_{j=1}^n (\mathbf{d}_j - \mathbf{m})(\mathbf{d}_j - \mathbf{m})^T = \frac{1}{n-1} \mathbf{D} \mathbf{P}_t \mathbf{D}^T \\ \mathbf{S}_w &= \frac{1}{n-1} \sum_{i=1}^c \sum_{\mathbf{d}_j \in C_i} (\mathbf{d}_j - \mathbf{m}_i)(\mathbf{d}_j - \mathbf{m}_i)^T = \frac{1}{n-1} \mathbf{D} \mathbf{P}_w \mathbf{D}^T \\ \mathbf{S}_b &= \frac{1}{n-1} \sum_{i=1}^c n_i (\mathbf{m}_i - \mathbf{m})(\mathbf{m}_i - \mathbf{m})^T = \frac{1}{n-1} \mathbf{D} \mathbf{P}_b \mathbf{D}^T\end{aligned}$$

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:

$$\mathbf{P}_t = \mathbf{I}_n - \frac{1}{n} \mathbf{1}_n \mathbf{1}_n^T \quad \mathbf{P}_w = \mathbf{I}_n - \mathbf{G}(\mathbf{G}^T \mathbf{G})^{-1} \mathbf{G}^T \quad \mathbf{P}_b = \mathbf{G}(\mathbf{G}^T \mathbf{G})^{-1} \mathbf{G}^T - \frac{1}{n} \mathbf{1}_n \mathbf{1}_n^T$$

$\mathbf{G} \in \mathfrak{R}^{n \times c}$ 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 the previous matrix expressions, it is straightforward to show that $\mathbf{S}_t = \mathbf{S}_w + \mathbf{S}_b$. The upper bounds on the ranks of the matrices are $\min(c-1, d)$, $\min(n-c, d)$, $\min(n-1, d)$ for \mathbf{S}_b , \mathbf{S}_w , and \mathbf{S}_t respectively.

1.3. 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 kernel PCA, kernel LDA, kernel CCA, k -means, and normalized cuts.

^aBold capital letters denote a matrix \mathbf{D} , bold lower-case letters a column vector \mathbf{d} . \mathbf{d}_j represents the j^{th} column of the matrix \mathbf{D} . All non-bold letters denote scalar variables. \mathbf{d}^j is a column vector that represents the j -th row of the matrix \mathbf{D} . d_{ij} denotes the scalar in the row i and column j of the matrix \mathbf{D} . $\mathbf{1}_k \in \mathfrak{R}^{k \times 1}$ is a vector of ones. $\mathbf{I}_k \in \mathfrak{R}^{k \times k}$ denotes the identity matrix. $\|\mathbf{d}\|_2^2$ denotes the norm of the vector \mathbf{d} . $\text{tr}(\mathbf{A}) = \sum_i a_{ii}$ is the trace of the matrix \mathbf{A} and $|\mathbf{A}|$ denotes the determinant. $\text{vec}(\mathbf{A})$ is a linear operator which converts a matrix $\mathbf{A} \in \mathfrak{R}^{m \times n}$ into a column vector $\mathbf{a} \in \mathfrak{R}^{mn \times 1}$. $\|\mathbf{A}\|_F^2 = \text{tr}(\mathbf{A}^T \mathbf{A}) = \text{tr}(\mathbf{A} \mathbf{A}^T)$ designates the Frobenius norm of a matrix. \circ denotes the Hadamard or point-wise product, and \otimes the Kronecker product.

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

Since its introduction in the early 1950s by Anderson^{12,13}, the reduced-rank regression (RRR) model has inspired a wealth of diverse applications in several fields such as signal processing^{14,15} (also known as reduced rank Wiener filtering), neural networks¹⁶ (also known as asymmetric PCA), time series analysis¹², and computer vision¹⁷. This section, extends previous work on RRR by introducing kernels and weights within a least-squares formulation, and it derives the system of GEPs resulting from solving it.

Given two data sets $\mathbf{X} \in \mathbb{R}^{x \times n}$ and $\mathbf{D} \in \mathbb{R}^{d \times n}$, the RRR model^{12,15,16} finds a linear mapping, $\mathbf{T} \in \mathbb{R}^{x \times d}$, that minimizes the LS error subject to rank constraints on \mathbf{T} . The RRR model minimizes $\|\mathbf{D} - \mathbf{TX}\|_F^2$ subject to $rank(\mathbf{T}) = k$. The rank constrain is typically needed when \mathbf{X} is high dimensional and the dimension of the features is larger than the samples ($x > n$).

The LS-WKRRR extends previous work on RRR on three aspects: (1) it explicitly parameterizes \mathbf{T} as the outer product of two matrices of rank k , that is $\mathbf{T} = \mathbf{AB}^T$, where $\mathbf{A} \in \mathbb{R}^{x \times k}$ and $\mathbf{B} \in \mathbb{R}^{d \times k}$, as has previously proposed¹⁵⁻¹⁷; (2) allows for non-linear regression. LS-WKRRR maps the input space of \mathbf{D} and \mathbf{X} to a feature space using kernel methods. That is, $\mathbf{\Gamma} = \phi(\mathbf{D}) = [\phi(\mathbf{d}_1) \phi(\mathbf{d}_2) \cdots \phi(\mathbf{d}_n)] \in \mathbb{R}^{d_d \times n}$ represents a mapping of \mathbf{D} . ϕ denotes a mapping from the d dimensional space to the feature space (d_d dimension). Similarly, $\mathbf{\Upsilon} = \varphi(\mathbf{X}) = [\varphi(\mathbf{x}_1) \varphi(\mathbf{x}_2) \cdots \varphi(\mathbf{x}_n)] \in \mathbb{R}^{d_x \times n}$ denotes the mapping for \mathbf{X} . ϕ, φ map the data to a (usually) higher dimensional space, where the data is more likely to behave linearly (if the right mapping is found). (3) The LS-WKRRR introduces weights for the features $\mathbf{W}_r \in \mathbb{R}^{d_d \times d_d}$ and samples $\mathbf{W}_c \in \mathbb{R}^{n \times n}$.

The LS-WKRRR problem minimizes the following expression:

$$E_0(\mathbf{A}, \mathbf{B}) = \|\mathbf{W}_r(\mathbf{\Gamma} - \mathbf{BA}^T\mathbf{\Upsilon})\mathbf{W}_c\|_F^2 \quad (1.1)$$

with respect to the regression matrices, $\mathbf{A} \in \mathbb{R}^{d_x \times k}$ and $\mathbf{B} \in \mathbb{R}^{d_d \times k}$. Typically \mathbf{A} corresponds to the projection matrix (e.g. LDA), and \mathbf{B} is a generative matrix for the column space of $\mathbf{\Gamma}$. $\mathbf{W}_r \in \mathbb{R}^{d_d \times d_d}$ is a matrix that weights the contributions of features (e.g. PCA) or classes (e.g. LDA). Similarly, $\mathbf{W}_c \in \mathbb{R}^{n \times n}$ weights the importance of each sample. In the following, we will assume that the weighting matrices are symmetric. The mappings ϕ and φ do not need to be explicitly computed and only the kernel between two samples needs to be defined. Kernel methods^{18,19} make use of the *kernel trick* to implicitly define the mapping by means of a kernel function. A given function is a kernel if, and only if, the value it produces for two vectors corresponds to a dot product in some Hilbert feature space. This is the well-known Representer Theorem: "Every positive definite, symmetric function is a kernel. For every kernel k , there is a function $\phi(\mathbf{x}) : k(\mathbf{d}_1, \mathbf{d}_2) = \langle \phi(\mathbf{d}_1), \phi(\mathbf{d}_2) \rangle$," where $\langle \rangle$ denotes dot product.

The necessary conditions on \mathbf{A} and \mathbf{B} for the critical points of eq. (1.1) are:

$$\frac{\partial E_0}{\partial \mathbf{B}} = \mathbf{W}_r^2 \mathbf{BA}^T \mathbf{\Upsilon} \mathbf{W}_c^2 \mathbf{\Upsilon}^T \mathbf{A} - \mathbf{W}_r^2 \mathbf{\Gamma} \mathbf{W}_c^2 \mathbf{\Upsilon}^T \mathbf{A} = \mathbf{0} \quad (1.2)$$

$$\frac{\partial E_0}{\partial \mathbf{A}} = \mathbf{\Upsilon} \mathbf{W}_c^2 \mathbf{\Gamma}^T \mathbf{W}_r^2 \mathbf{B} - \mathbf{\Upsilon} \mathbf{W}_c^2 \mathbf{\Upsilon}^T \mathbf{AB}^T \mathbf{W}_r^2 \mathbf{B} = \mathbf{0} \quad (1.3)$$

Eq. (1.2) and eq. (1.3) form a set of coupled equations that have solutions in terms of a GEP in either \mathbf{A} or \mathbf{B} . 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. (1.2) into eq. (1.1) leads to the following expression:

$$E_0(\mathbf{A}) \propto \text{tr}((\mathbf{A}^T\mathbf{\Upsilon}\mathbf{W}_c^2\mathbf{\Upsilon}^T\mathbf{A})^{-1}(\mathbf{A}^T\mathbf{\Upsilon}\mathbf{W}_c^2\mathbf{\Gamma}^T\mathbf{W}_r^2\mathbf{\Gamma}\mathbf{W}_c^2\mathbf{\Upsilon}^T\mathbf{A})) \quad (1.4)$$

Similarly, 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. (1.3) into eq. (1.1) leads to the following expression:

$$E_0(\mathbf{B}) \propto \text{tr}((\mathbf{B}^T\mathbf{W}_r^2\mathbf{B})^{-1}(\mathbf{B}^T\mathbf{W}_r^2\mathbf{\Gamma}\mathbf{W}_c^2\mathbf{\Upsilon}^T(\mathbf{\Upsilon}\mathbf{W}_c^2\mathbf{\Upsilon}^T)^{-1}\mathbf{\Upsilon}\mathbf{W}_c^2\mathbf{\Gamma}^T\mathbf{W}_r^2\mathbf{B})) \quad (1.5)$$

Eq. (1.4) and eq. (1.5) are the **fundamental equations** of CA methods. In the rest of the manuscript, we will show how to relate many CA methods to these two equations. Eq. (1.4) and eq. (1.5) are standard Rayleigh quotients (i.e. $J(\mathbf{B}) = \text{tr}((\mathbf{B}^T\mathbf{S}_1\mathbf{B})^{-1}\mathbf{B}^T\mathbf{S}_2\mathbf{B})$), and the solution is given by the following GEP, $\mathbf{S}_2\mathbf{B} = \mathbf{S}_1\mathbf{B}\lambda$.¹⁰ The solution of eq. (1.4) is unique up to an invertible transformation \mathbf{R} , that is, $E_o(\mathbf{A}\mathbf{R}) = E_o(\mathbf{A})$. Similarly for eq. (1.5).

Recasting CA eigenvalue problems as a LS-WKRRR problem (E_0), eq (1.1), has a number of desirable benefits that will be illustrated throughout the chapter:

- (1) E_0 provides a unifying expression for many CA methods. The commonalities and differences between the methods, as well as the intrinsic relationship, can be easily understood from eq. E_0 .
- (2) The Least-Squares (LS) formulation provides an alternative and simple framework to understand normalization factors in CA methods. For instance, normalization terms in spectral graph, or weighting factors in PCA/LDA.
- (3) The surface of E_0 , eq (1.1), has a unique local minimum²⁰, so most of the optimization-based algorithms will exhibit almost global convergence properties. In general, optimization theory provides a solid framework for convergence analysis, and many optimization methods are suitable (section 1.3.2).
- (4) Many numerical optimization methods (e.g. gradient descent, alternated least-squares) can be applied to solve E_0 . Directly optimizing E_0 solves the small sample size (SSS) problem of standard GEPs when dealing with high dimensional data. Moreover, optimization-based algorithms are more efficient for large-scale problems, especially if matrices are sparse. In addition, using the least-squares formulation one can easily derive extensions for online component analysis methods.
- (5) The least-squares formulation allows many straight-forward extensions of CA methods. It is unclear how to formulate these new extensions using an eigenvalue framework.

1.3.2. Computational Aspects of LS-WKRRR

This section proposes three optimization strategies to optimize the LS-WKRRR problem E_0 , eq (1.1).

1.3.2.1. *Subspace Iteration*

Standard numerical packages to solve GEP (i.e. $\mathbf{S}_1\mathbf{B} = \mathbf{S}_2\mathbf{B}\mathbf{\Lambda}$) are not well suited to solve eq. (1.4) or eq. (1.5) for high-dimensional data, especially when the number of samples is less than the number of features (small size sample problem). In this case, methods that use iterative schemes for minimizing the Rayleigh quotient $\frac{\mathbf{x}^T\mathbf{S}_1\mathbf{x}}{\mathbf{x}^T\mathbf{S}_2\mathbf{x}}$ ^{7,21} to obtain the biggest/smallest eigenvalue, rely on deflation procedures in order to obtain several eigenvectors. Such a deflation process often breaks down numerically (especially when increasing the number of eigenvectors)²². This section proposes a stable subspace iteration.

Given two covariance matrices, $\mathbf{S}_1 \in \mathbb{R}^{d \times d}$ and $\mathbf{S}_2 \in \mathbb{R}^{d \times d}$, and an initial random matrix $\mathbf{V}_0 \in \mathbb{R}^{d \times q}$, the subspace iteration method²² alternates the following steps:

$$\mathbf{S}_1 \hat{\mathbf{V}}_{t+1} = \mathbf{S}_2 \mathbf{V}_t \tag{1.6}$$

$$\mathbf{S} = \hat{\mathbf{V}}_{t+1}^T \mathbf{S}_1 \hat{\mathbf{V}}_{t+1} \quad \mathbf{T} = \hat{\mathbf{V}}_{t+1}^T \mathbf{S}_2 \hat{\mathbf{V}}_{t+1} \tag{1.7}$$

$$\mathbf{S}\mathbf{W} = \mathbf{T}\mathbf{W}\mathbf{\Delta} \tag{1.8}$$

$$\mathbf{V}_{t+1} = \hat{\mathbf{V}}_{t+1} \mathbf{W} \quad \hat{\mathbf{V}}_{t+1} = \hat{\mathbf{V}}_{t+1} / \|\hat{\mathbf{V}}_{t+1}\|_F^2$$

The first step, eq. (1.6), of the subspace iteration algorithm solves a linear system of equations to find $\hat{\mathbf{V}}_{t+1}$. In the second step, the data is projected onto the estimated subspace, eq. (1.7). In order to impose the constraints that $\mathbf{V}_{t+1}^T \mathbf{S}_1 \mathbf{V}_{t+1} = \mathbf{\Lambda}$ and $\mathbf{V}_{t+1}^T \mathbf{S}_2 \mathbf{V}_{t+1} = \mathbf{I}_q$, a normalization is done by solving the following $q \times q$ generalized eigenvalue problem, $\mathbf{S}\mathbf{W} = \mathbf{T}\mathbf{W}\mathbf{\Delta}$, eq. (1.8). It can be shown²² that as t increases, \mathbf{V}_{t+1} will converge to the eigenvectors of $\mathbf{S}_1\mathbf{B} = \mathbf{S}_2\mathbf{B}\mathbf{\Lambda}$ and $\mathbf{\Delta}$ to the eigenvalues $\mathbf{\Lambda}$. 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. The subspace iteration algorithm converges linearly and the convergence rate is proportional to $\frac{|\delta_q|}{|\delta_{q+1}|}$. It is not critical that \mathbf{V}_0 does not have a projection onto the first q generalized eigenvectors, because numerical errors will provide such a projection.

The computationally expensive part of the subspace iteration algorithm is to solve the linear system of equations (especially for high-dimensional data), i.e. eq. (1.6). To regularize the solution and improve efficiency, we approximate the covariance as $\mathbf{S}_1 \approx \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T + \sigma^2\mathbf{I}_d$, where $\mathbf{U} \in \mathbb{R}^{d \times k}$. The parameters σ^2 , \mathbf{U} and $\mathbf{\Lambda}$ can be estimated by minimizing:

$$E(\mathbf{U}, \mathbf{\Lambda}, \sigma^2) = \|\mathbf{S}_1 - \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T - \sigma^2\mathbf{I}_d\|_F^2. \tag{1.9}$$

The optimal parameters correspond to: $\sigma^2 = \text{tr}(\mathbf{S}_1 - \mathbf{U}\hat{\mathbf{\Lambda}}\mathbf{U}^T)/d - k$, $\mathbf{\Lambda} = \hat{\mathbf{\Lambda}} - \sigma^2\mathbf{I}_d$, where $\hat{\mathbf{\Lambda}}$ are the eigenvalues of the \mathbf{S}_1 and $\mathbf{U} \in \mathbb{R}^{d \times k}$ the first k eigenvectors.

Once the factorization is done, inverting \mathbf{S}_1 can be done efficiently using the matrix inversion lemma²³ $(\mathbf{U}\mathbf{\Lambda}\mathbf{U}^T + \sigma^2\mathbf{I}_d)^{-1} = \frac{1}{\sigma^2}(\mathbf{I}_d - \frac{1}{\sigma^2}\mathbf{U}(\mathbf{\Lambda}^{-1} + \frac{\mathbf{I}_k}{\sigma^2})^{-1}\mathbf{U}^T)$.

1.3.2.2. *Alternated Least Squares (ALS)*

Solving the GEP resulting from the LS-WKRRR with a subspace iteration method or standard eigen-packages might be computationally intensive if the weighted covariance matrices do not have any special structure (e.g. sparse). For large amounts of high dimensional

data previous approaches might not be efficient in either space nor time, and Alternated Least-Squares (ALS) approaches might be more convenient.

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:

$$\mathbf{A} = (\mathbf{\Upsilon}\mathbf{W}_c^2\mathbf{\Upsilon}^T)^{-1}\mathbf{\Upsilon}\mathbf{W}_c^2\mathbf{\Gamma}^T\mathbf{W}_r^2\mathbf{B}(\mathbf{B}^T\mathbf{W}_r^2\mathbf{B})^{-1} \quad (1.10)$$

$$\mathbf{B} = \mathbf{\Gamma}\mathbf{W}_c^2\mathbf{\Upsilon}^T\mathbf{A}(\mathbf{A}^T\mathbf{\Upsilon}\mathbf{W}_c^2\mathbf{\Upsilon}^T\mathbf{A})^{-1} \quad (1.11)$$

After a few iterations, ALS strategies have shown slower convergence than gradient descent strategies.^{24,25} In the case of kernel methods, the ALS procedure needs to re-parameterize \mathbf{B} , see section 1.4.2 for more details.

1.3.2.3. Gradient descent

For large amounts of high dimensional data, gradient descent approaches can provide a less computationally demanding alternative^{24,25}. Eq. (1.2) and eq. (1.3) suggests a simple gradient descent scheme:

$$\mathbf{A}^{t+1} = \mathbf{A}^t - \eta_a \frac{\partial E_0(\mathbf{A}^t)}{\partial \mathbf{A}} \quad \mathbf{B}^{t+1} = \mathbf{B}^t - \eta_b \frac{\partial E_0(\mathbf{B}^t)}{\partial \mathbf{B}} \quad (1.12)$$

A major problem with the update of eq. (1.12) is determining the optimal η . η can be found with a line search strategy,^{24,26} or as an estimate on upper bound on the diagonal of the Hessian matrix.^{25,27} Recently, Buchanan and Fitzgibbon²⁴ showed how a damped Newton algorithm on the joint pair \mathbf{A}, \mathbf{B} (i.e. $\text{vec}([\mathbf{A}; \mathbf{B}])$) is more efficient than alternated least-squares 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 algorithm and the gradient descent algorithm will effectively solve the SSS problem of many common CA methods. This is another advantage of using optimization techniques on a LS formulation rather than solving the resulting eigen-problem. Moreover, the computational cost of the algorithms will be less than standard eigen-decompositions, $O(n^3)$ or $O(d^3)$ for \mathbf{A} and \mathbf{B} , respectively.

1.4. PCA, KPCA, and weighted extensions

This section derives PCA, KPCA and weighted extensions as a particular case of the fundamental equation of CA methods (E_0), eq. (1.1).

1.4.1. Principal Component Analysis (PCA)

PCA is one of the most popular dimensionality reduction techniques^{1,16,28,29}. The basic ideas behind PCA date back to Pearson in 1901,²⁸ and a more general procedure was described by Hotelling²⁹ in 1933. PCA finds an orthogonal subspace (\mathbf{B}) that best preserves the covariance (\mathbf{S}_t) of the data \mathbf{D} . PCA maximizes:

$$J_1(\mathbf{B}) = \text{tr}(\mathbf{B}^T\mathbf{S}_t\mathbf{B}) \quad \text{s.t.} \quad \mathbf{B}^T\mathbf{B} = \mathbf{I}_k \quad (1.13)$$

where $\mathbf{B} \in \mathfrak{R}^{d \times k}$, being d the number of features, n number of samples, and k the dimension of the subspace. Typically $k \leq \min(n, d)$. The columns of \mathbf{B} form an orthonormal basis that spans the principal subspace of \mathbf{D} . PCA can be computed in closed-form by calculating the leading eigenvectors of the covariance matrix \mathbf{S}_t ^{1,16}. The PCA projections, $\mathbf{C} = \mathbf{B}^T \mathbf{D} (\mathbf{I}_n - \frac{1}{n} \mathbf{1}_n \mathbf{1}_n^T) \in \mathfrak{R}^{k \times n}$, are decorrelated, that is $\mathbf{C} \mathbf{C}^T = \mathbf{\Lambda}$, where $\mathbf{\Lambda} \in \mathfrak{R}^{k \times k}$ is a diagonal matrix containing the eigenvalues of \mathbf{S}_t .

For large data sets of high dimensional data (d and n are large), minimizing a least-squares error function^{30,31} is an efficient procedure (in both space and time) to compute the principal subspace of \mathbf{D} . There exist several least-squares error functions such that the stationary points are solutions of PCA. Consider the fundamental equation of CA, eq. (1.1), where $\mathbf{\Upsilon} = \mathbf{I}_d$, $\mathbf{W}_r = \mathbf{I}_d$, $\mathbf{W}_c = \mathbf{I}_n$, $\mathbf{\Gamma} = \mathbf{D}$, and $\mathbf{D}\mathbf{1} = \mathbf{0}$ (zero mean data):

$$E_1(\mathbf{B}, \mathbf{A}) = \|\mathbf{D} - \mathbf{B}\mathbf{A}^T\|_F^2 \quad (1.14)$$

In this case, eq. (1.4) and eq. (1.5) transform to:

$$E_1(\mathbf{A}) \propto \text{tr}((\mathbf{A}^T \mathbf{A})^{-1} (\mathbf{A}^T \mathbf{D}^T \mathbf{D} \mathbf{A})) \quad (1.15)$$

$$E_1(\mathbf{B}) \propto \text{tr}((\mathbf{B}^T \mathbf{B})^{-1} (\mathbf{B}^T \mathbf{D} \mathbf{D}^T \mathbf{B})) \quad (1.16)$$

The \mathbf{B} that maximizes eq. (1.16) is given by the leading eigenvectors of covariance matrix $(\mathbf{D} \mathbf{D}^T)$. Similarly, the optimal \mathbf{A} corresponds to the eigenvectors of the Gram matrix $(\mathbf{D}^T \mathbf{D})$. Observe that the primal and dual formulation of PCA lead to a clean and direct connection with the estimates of the regression matrices \mathbf{A} and \mathbf{B} .

Alternated least-squares (ALS) approaches to solve eq. (1.14), alternate between solving for \mathbf{A} while \mathbf{B} is fixed and vice versa^{20,25,32,33}. In the case of PCA, the ALS equation (eq. (1.10) and eq. (1.11)) reduce to solve the following systems of linear equations:

$$\mathbf{D}^T \mathbf{B} = \mathbf{A} \mathbf{B}^T \mathbf{B} \quad (1.17)$$

$$\mathbf{D} \mathbf{A} = \mathbf{B} \mathbf{A}^T \mathbf{A} \quad (1.18)$$

This optimization is equivalent to the Expectation Maximization (EM) algorithm in probabilistic PCA (PPCA)^{31,34} 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 $\mathbf{R} \in \mathfrak{R}^{k \times k}$ 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 following simultaneous diagonalization:

$$\mathbf{R}^T \mathbf{B}^T \mathbf{B} \mathbf{R} = \mathbf{I} \quad \mathbf{R}^T (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{R} = \mathbf{\Lambda}^{-1}$$

where $\mathbf{\Lambda} \in \mathfrak{R}^{k \times k}$ is a diagonal matrix containing the eigenvalues of \mathbf{S}_t . \mathbf{R} can be computed by solving the following $k \times k$ 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 E_0 , eq. (1.1), with the following values³⁰: $\mathbf{\Gamma} = \mathbf{D}$, $\mathbf{W}_r = \mathbf{I}_d$, $\mathbf{W}_c = \mathbf{I}_n$, $\mathbf{A} = \mathbf{B}$:

$$E_2(\mathbf{B}) = \|\mathbf{D} - \mathbf{B}(\mathbf{B}^T \mathbf{D})\|_F^2 \quad \text{s.t.} \quad \mathbf{B}^T \mathbf{B} = \mathbf{I}_k \quad (1.19)$$

However, eq. (1.19) 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 outliers²⁵.

1.4.2. Kernel Principal Component Analysis (KPCA)

Similar to PCA, KPCA can be derived from E_0 , eq. (1.1), by lifting the original data samples, \mathbf{D} , to a feature space, that is, $\mathbf{\Gamma} = \phi(\mathbf{D})$. The kernelized version of eq. (1.14) can be written as:

$$E_3(\mathbf{B}, \mathbf{A}) = \|\mathbf{\Gamma} - \mathbf{B}\mathbf{A}^T\|_F^2 \quad (1.20)$$

The optimal \mathbf{A} can be obtained from one of the fundamental equations of CA, eq. (1.4):

$$E_3(\mathbf{A}) \propto \text{tr}((\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{K} \mathbf{A}) \quad (1.21)$$

where $\mathbf{K} = \mathbf{\Gamma}^T \mathbf{\Gamma} \in \mathfrak{R}^{n \times n}$ is the standard 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. Observe that in the case of kernel methods, it is (in general) not possible to directly solve eq. (1.5), because the covariance in the input space, $\mathbf{\Gamma}\mathbf{\Gamma}^T$, can be infinite dimensional.

The computational cost of the eigen-decomposition of \mathbf{K} is $O(n^3)$ (no sparsity is assumed), where n is the number of samples. For large amounts of data (large n) an ALS or gradient-descent approach to computing KPCA is computationally more convenient (see Section 1.3.2.2). 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 $\mathbf{\Gamma}$ ¹⁸; that is, $\mathbf{B} = \mathbf{\Gamma}\boldsymbol{\alpha}$, where $\boldsymbol{\alpha} \in \mathfrak{R}^{n \times k}$. Substituting this expression into eq. (1.20) results in:

$$E_3(\boldsymbol{\alpha}, \mathbf{A}) = \|\mathbf{\Gamma}(\mathbf{I}_n - \boldsymbol{\alpha}\mathbf{A}^T)\|_F^2 \quad (1.22)$$

Assuming that \mathbf{K} is invertible, similarly to the ALS-PCA, we can alternate between computing $\boldsymbol{\alpha}$ and $\mathbf{A} \in \mathfrak{R}^{k \times n}$ as:

$$\boldsymbol{\alpha} = \mathbf{A}^T (\mathbf{A}\mathbf{A}^T)^{-1} \mathbf{A} = (\boldsymbol{\alpha}^T \mathbf{K} \boldsymbol{\alpha})^{-1} \boldsymbol{\alpha}^T \mathbf{K} \quad (1.23)$$

The computational cost of each iteration is $O(n^2k)$.

1.4.3. Weighted Extensions

In many situations, it is convenient to weight the 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 could be weighted less in the model. It could also be the case that some face images are outliers, and we would like to reduce their influence in the subspace.

Eq. (1.4) and eq. (1.5) 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 E_0 , eq. (1.1). The closed-form solutions for the weighted PCA is given by the fundamental equations of CA, eq. (1.4) and eq. (1.5):

$$E_0(\mathbf{A}) \propto \text{tr}((\mathbf{A}^T \mathbf{W}_c^2 \mathbf{A})^{-1} (\mathbf{A}^T \mathbf{W}_c^2 \mathbf{D}^T \mathbf{W}_r^2 \mathbf{D} \mathbf{W}_c^2 \mathbf{A})) \quad (1.24)$$

$$E_0(\mathbf{B}) \propto \text{tr}((\mathbf{B}^T \mathbf{W}_r^2 \mathbf{B})^{-1} (\mathbf{B}^T \mathbf{W}_r^2 \mathbf{D} \mathbf{W}_c^2 \mathbf{D}^T \mathbf{W}_r^2 \mathbf{B})) \quad (1.25)$$

Previous equations have a closed-form solution as a GEP. For \mathbf{A} , the GEP is $\mathbf{D}^T \mathbf{W}_r^2 \mathbf{D} \mathbf{W}_c^2 \mathbf{A} = \mathbf{A} \mathbf{\Lambda}_a$ and for the dual problem in \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 Valued Decomposition (GSVD) provides^{35,36} an alternative approach to solve the 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) directly translates to weighting the kernel matrix $\mathbf{K} \mathbf{W}_c^2 \mathbf{A} = \mathbf{A} \mathbf{\Lambda}_a$. If the weighting is in the feature space (e.g. Mahalanobis in feature space), the weighting can still be taken into account using the kernel trick.³⁷

In general, for an arbitrary set of weights, the weighted PCA minimizes:

$$E_4(\mathbf{A}, \mathbf{B}) = \|\mathbf{W} \circ (\mathbf{D} - \mathbf{B} \mathbf{A}^T)\|_F^2 \quad (1.26)$$

\circ denotes the Hadamard or pointwise product. Observe that there is no closed-form solution in terms of GEP for the solution of eq. (1.26)^{32,36}. Moreover, the problem of data factorization with arbitrary weights has several local minima depending on the structure of the weights^{24,38}. Minimization of E_4 , eq. (1.26), has been typically used to solve PCA with missing data^{24,32,38} or outliers in PCA^{25,39} or LDA.⁴⁰ Recently, Aguiar et al.⁴¹ have proposed a closed-form solution to the data factorization problem, when the missing data has a special structure.

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

This section relates LDA, KLDA, CCA and KCCA to the LS-WKRRR problem of E_0 , eq. (1.1), and derives weighted generalizations.

1.5.1. Linear Discriminant Analysis (LDA)

Let $\mathbf{D} \in \mathfrak{R}^{d \times n}$ 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. $\mathbf{G} \in \mathfrak{R}^{n \times c}$ 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 Fisher³ for the two-class case and later extended to the multi-class case,¹⁰ computes a linear transformation ($\mathbf{A} \in \mathfrak{R}^{d \times k}$) of \mathbf{D} that maximizes the Euclidian distance between the means of the classes (\mathbf{S}_b) while minimizing the within-class variance (\mathbf{S}_w). Rayleigh-like quotients are among the most popular LDA optimization criteria¹⁰. For instance, LDA can be obtained by maximizing:

$$J_2(\mathbf{A}) = \text{tr}((\mathbf{A}^T \mathbf{S}_1 \mathbf{A})^{-1} \mathbf{A}^T \mathbf{S}_2 \mathbf{A}) \quad (1.27)$$

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 $\mathbf{S}_2 \in \{\mathbf{S}_b, \mathbf{S}_b, \mathbf{S}_t\}$). The Rayleigh quotient of eq. (1.27) has a

closed-form solution in terms of a GEP, $\mathbf{S}_2 \mathbf{A} = \mathbf{S}_1 \mathbf{A} \Lambda_a$,¹⁰ where Λ_a are the eigenvalues. In the case of high-dimensional data (e.g. images), 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^{42,43}, regularized LDA⁴⁴⁻⁴⁶, and many other methods that explore several combinations of the Null and Range spaces of the matrices \mathbf{S}_1 and \mathbf{S}_2 ⁴⁷. See⁴⁸ for a unifying review of the optimal solution of eq. (1.27) based on the analysis 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 (Fisher Discriminant),⁴⁹ and extended to the multi-class case using several approximations of the covariances^{44,50}. Recently,⁵¹ extended⁴⁴ by finding the optimal indicator matrix \mathbf{G} that corresponds to LDA. This section provides a simpler proof of the relation between regression and LDA using a convenient matrix formulation.⁵² In the following, we will assume zero mean data ($\mathbf{D}\mathbf{1} = \mathbf{0}$).

Consider E_0 , eq. (1.1), 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 $\mathbf{D}\mathbf{1} = \mathbf{0}$, E_0 transforms to:

$$E_5(\mathbf{A}, \mathbf{B}) = \|(\mathbf{G}^T \mathbf{G})^{-\frac{1}{2}}(\mathbf{G}^T - \mathbf{B}\mathbf{A}^T \mathbf{D})\|_F^2 \quad (1.28)$$

Considering $\mathbf{C} = \mathbf{A}^T \mathbf{D}$ and after eliminating \mathbf{B} , eq. (1.28) can be re-written as:

$$E_5(\mathbf{A}) = \|(\mathbf{G}^T \mathbf{G})^{-\frac{1}{2}} \mathbf{G}^T (\mathbf{I}_n - \mathbf{C}^T (\mathbf{C}\mathbf{C}^T)^{-1} \mathbf{C})\|_F^2 \propto \quad (1.29)$$

$$tr((\underbrace{\mathbf{A}^T \mathbf{D}\mathbf{D}^T \mathbf{A}}_{\mathbf{S}_t})^{-1} \mathbf{A}^T \underbrace{\mathbf{D}\mathbf{G}(\mathbf{G}^T \mathbf{G})^{-1} \mathbf{G}^T \mathbf{D}^T \mathbf{A}}_{\mathbf{S}_b}) \quad (1.30)$$

Eq. (1.30) is one of the standard Rayleigh quotients for LDA. Eq. (1.30) can also be derived from one of the fundamental equations of CA, eq. (1.4). Recall that LDA is a supervised learning problem and the binary indicator matrix $\mathbf{G} \in \mathbb{R}^{c \times n}$ is given. LDA can be understood as using RRR from the data samples (\mathbf{D}) to the labels (\mathbf{G}), weighted by $\mathbf{G}^T \mathbf{G}$ to compensate for unequal number of samples in each class. Observe, that directly optimizing eq. (1.30) (e.g. gradient descent) with respect to \mathbf{A} and \mathbf{B} in eq. 1.28 avoids the small sample size (SSS) problem and can be a numerically convenient algorithm for large amounts of high dimensional data.

1.5.2. Kernel Linear Discriminant Analysis (KLDA)

Kernel Linear Discriminant Analysis (KLDA)⁵³ can also be derived from E_0 , eq. (1.1). Consider E_0 , eq. (1.1), where $\mathbf{\Gamma} = \mathbf{G}^T$, $\mathbf{\Upsilon} = \varphi(\mathbf{D})$, $\mathbf{W}_r = (\mathbf{G}^T \mathbf{G})^{-\frac{1}{2}}$, $\mathbf{W}_c = \mathbf{I}_n$:

$$E_6(\mathbf{A}, \mathbf{B}) = \|(\mathbf{G}^T \mathbf{G})^{-\frac{1}{2}}(\mathbf{G}^T - \mathbf{B}\mathbf{A}^T \varphi(\mathbf{D}))\|_F^2 \quad (1.31)$$

In this case, eq. (1.4) translates to the following expression:

$$E_6(\mathbf{A}) \propto tr((\mathbf{A}^T \mathbf{\Upsilon} \mathbf{\Upsilon}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{\Upsilon} \mathbf{G} (\mathbf{G}^T \mathbf{G})^{-1} \mathbf{G}^T \mathbf{\Upsilon}^T \mathbf{A}) \quad (1.32)$$

Using the Mercer theorem,^{18,19} and assuming that the dimension of the feature space is larger than the number of samples, it can be shown that the solution to the KLDA problem can be expressed as $\mathbf{A} = \mathbf{Y}\boldsymbol{\alpha}$.⁵³ Using this fact, the KLDA can be found as the solution of the following GEP, $\mathbf{K}\mathbf{G}(\mathbf{G}^T\mathbf{G})^{-1}\mathbf{G}^T\mathbf{K}^T\boldsymbol{\alpha} = \mathbf{K}^2\boldsymbol{\alpha}\Lambda_{\boldsymbol{\alpha}}$, where $\mathbf{K} = \mathbf{Y}^T\mathbf{Y}$ is the kernel matrix and $\boldsymbol{\alpha}$ the eigenvectors of the GEP.

1.5.3. Canonical Correlation Analysis (CCA) and Kernel CCA

Canonical correlation analysis (CCA) is a technique to extract common features from a pair of multivariate data. CCA, first proposed by Hotelling in 1936,⁴ identifies relationships between two sets of variables by finding the linear combinations of the variables in the first set ($\mathbf{D} \in \mathfrak{R}^{d_d \times n}$) that are most highly correlated with the linear combinations of the variables in the second set ($\mathbf{X} \in \mathfrak{R}^{d_x \times n}$). CCA has been used for matching sets of images in problems such as activity recognition from video,⁵⁴ robot navigation,⁵⁵ and pose estimation.⁵⁶

Assuming zero mean data (i.e. $\mathbf{D}\mathbf{1}_n = \mathbf{0}$, $\mathbf{X}\mathbf{1}_n = \mathbf{0}$), CCA finds a combination of the original variables (i.e. $\hat{\mathbf{B}}^T\mathbf{D}$ and $\hat{\mathbf{A}}^T\mathbf{X}$) that maximize:⁴

$$J_3(\hat{\mathbf{A}}, \hat{\mathbf{B}}) = \text{tr}(\hat{\mathbf{B}}^T\mathbf{S}^{\mathbf{D}\mathbf{X}}\hat{\mathbf{A}}) \quad \text{s.t.} \quad \hat{\mathbf{B}}^T\mathbf{S}_t^{\mathbf{D}}\hat{\mathbf{B}} = \hat{\mathbf{A}}^T\mathbf{S}_t^{\mathbf{X}}\hat{\mathbf{A}} = \mathbf{I} \quad (1.33)$$

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 $\mathbf{S}^{\mathbf{D}\mathbf{X}} = \frac{1}{n-1}\mathbf{D}\mathbf{X}^T$. The pair of canonical variates ($\hat{\mathbf{b}}_i^T\mathbf{D}$, $\hat{\mathbf{a}}_i^T\mathbf{X}$) 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. (1.33) has a closed-form solution as two symmetric GEPs:^{4,57}

$$(\mathbf{S}_t^{\mathbf{X}})^{-1}\mathbf{S}^{\mathbf{X}\mathbf{D}}(\mathbf{S}_t^{\mathbf{D}})^{-1}\mathbf{S}^{\mathbf{D}\mathbf{X}}\hat{\mathbf{A}} = \hat{\mathbf{A}}\Lambda_{\hat{\mathbf{a}}} \quad (1.34)$$

$$(\mathbf{S}_t^{\mathbf{D}})^{-1}\mathbf{S}^{\mathbf{D}\mathbf{X}}(\mathbf{S}_t^{\mathbf{X}})^{-1}\mathbf{S}^{\mathbf{X}\mathbf{D}}\hat{\mathbf{B}} = \hat{\mathbf{B}}\Lambda_{\hat{\mathbf{b}}} \quad (1.35)$$

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

Borga⁷ proposed a unified eigen-framework for PCA, CCA, and Partial Least-Squares (PLS).⁷ showed that the canonical factors can be obtained as the critical points of the following Rayleigh quotient:⁷

$$J_4(\mathbf{U}) = \text{tr}\left(\left(\mathbf{U}^T \begin{pmatrix} \mathbf{0} & \mathbf{S}^{\mathbf{X}\mathbf{D}} \\ \mathbf{S}^{\mathbf{D}\mathbf{X}} & \mathbf{0} \end{pmatrix} \mathbf{U}\right)^{-1} \left(\mathbf{U}^T \begin{pmatrix} \mathbf{S}_t^{\mathbf{X}} & \mathbf{0} \\ \mathbf{0} & \mathbf{S}_t^{\mathbf{D}} \end{pmatrix} \mathbf{U}\right)\right) \quad \text{s.t.} \quad \mathbf{U} = \begin{bmatrix} \hat{\mathbf{A}} \\ \hat{\mathbf{B}} \end{bmatrix} \quad (1.36)$$

Previous eigen-formulation of CCA, eq. (1.34), eq. (1.35) and eq. (1.36), is appealing from an analytical viewpoint; however, solving the eigen-system is not a numerically convenient method for large amount of high-dimensional data.⁷ Alternatively, $\hat{\mathbf{A}}$ and $\hat{\mathbf{B}}$ can be obtained by performing gradient descent on eq. (1.36),⁷ or performing SVD on $(\mathbf{S}_t^{\mathbf{D}})^{-\frac{1}{2}}\mathbf{S}^{\mathbf{D}\mathbf{X}}(\mathbf{S}_t^{\mathbf{X}})^{-\frac{1}{2}}$.^{57,58}

In general, it is not clear how E_0 , eq. (1.1), can recover the canonical variates, because CCA treats both data sets \mathbf{D} and \mathbf{X} symmetrically, whereas LS-WKRRR only normalizes for \mathbf{X} . At this point, it is worth observing that if $\mathbf{X} = \mathbf{G}$ (the indicator matrix), the CCA

solution of eq. (1.35) is equivalent to the LDA solution, eq. (1.27). Using our matrix notation, it is straightforward to show that, in this case, eq. (1.35) in CCA reduces to $\mathbf{S}_w^{\mathbf{D}}\mathbf{B} = \mathbf{S}_t^{\mathbf{D}}\mathbf{B}\mathbf{A}$ (assuming zero mean data). Using this fact, we can interpret LDA as CCA. LDA finds the optimal linear subspace that makes \mathbf{D} best correlated with the label matrix \mathbf{G} . Similar reasoning can be done for the case of Kernel CCA. Using this observation, it is simple to relate CCA to the fundamental equation of CA, eq (1.1). In order to treat all the variables symmetrically, we introduce weights in the predicted variable (\mathbf{D}) (as LDA), and show that the CCA solution can be recovered using the fundamental equation of CA, E_0 . Consider E_0 , eq. (1.1), where $\mathbf{\Gamma} = \mathbf{D}$, $\mathbf{\Upsilon} = \mathbf{X}$, $\mathbf{W}_r = (\mathbf{D}\mathbf{D}^T)^{-\frac{1}{2}}$, and $\mathbf{W}_c = \mathbf{I}_n$.

$$E_7(\mathbf{A}, \mathbf{B}) = \|(\mathbf{D}^T\mathbf{D})^{-\frac{1}{2}}(\mathbf{D} - \mathbf{B}\mathbf{A}^T\mathbf{X})\|_F^2 \tag{1.37}$$

After substituting these values into one of the fundamental equations of CA, eq. (1.4), results in:

$$E_7(\mathbf{A}) \propto tr((\mathbf{A}^T\mathbf{S}_t^{\mathbf{X}}\mathbf{A})^{-1}\mathbf{A}^T\mathbf{S}^{\mathbf{X}\mathbf{D}}(\mathbf{S}_t^{\mathbf{D}})^{-1}\mathbf{S}^{\mathbf{D}\mathbf{X}}\mathbf{A}) \tag{1.38}$$

which corresponds to the GEP for CCA, eq. (1.34). Similarly, eq. (1.5) derives in:

$$E_7 \propto tr((\mathbf{B}^T(\mathbf{D}\mathbf{D}^T)^{-1}\mathbf{B})^{-1}\mathbf{B}^T(\mathbf{D}\mathbf{D}^T)^{-1}\mathbf{X}\mathbf{D}^T(\mathbf{D}\mathbf{D}^T)^{-1}\mathbf{D}\mathbf{X}^T(\mathbf{D}\mathbf{D}^T)^{-1}\mathbf{B}) \tag{1.39}$$

After a change of variable, $\mathbf{U} = \mathbf{B}(\mathbf{D}\mathbf{D}^T)^{-1}$, eq. (1.39) can be re-written as:

$$E_7(\mathbf{U}) \propto tr((\mathbf{U}^T(\mathbf{S}_t^{\mathbf{D}})\mathbf{U})^{-1}\mathbf{U}^T\mathbf{S}^{\mathbf{D}\mathbf{X}}(\mathbf{S}_t^{\mathbf{X}})^{-1}\mathbf{S}^{\mathbf{X}\mathbf{D}}\mathbf{U}) \tag{1.40}$$

which is the same solution provided by CCA, eq. (1.35).

There exist other least-squares formulations of CCA that are worth mentioning. To treat all the variables symmetrically, a LS function can be obtained by minimizing:

$$E_8(\mathbf{B}, \mathbf{A}) = \|\mathbf{B}^T\mathbf{D} - \mathbf{A}^T\mathbf{X}\|_F^2 \text{ s.t. } \mathbf{B}^T\mathbf{S}_t^{\mathbf{D}}\mathbf{B} = \mathbf{I}_d \quad \mathbf{A}^T\mathbf{S}_t^{\mathbf{X}}\mathbf{A} = \mathbf{I}_d \tag{1.41}$$

Assuming $\mathbf{S}_t^{\mathbf{D}} = \mathbf{D}\mathbf{D}^T$ is invertible, after optimizing w.r.t. the optimal $\mathbf{A} = \mathbf{S}_t^{\mathbf{D}^{-1}}\mathbf{D}\mathbf{X}^T\mathbf{B}$, eq. (1.41) transforms to:

$$E_8(\mathbf{B}) = tr(\mathbf{B}\mathbf{S}_t^{\mathbf{D}}\mathbf{B}) - tr(\mathbf{B}^T\mathbf{S}^{\mathbf{D}\mathbf{X}}\mathbf{S}_t^{\mathbf{X}^{-1}}\mathbf{S}^{\mathbf{X}\mathbf{D}}\mathbf{B}) \text{ s.t. } \mathbf{B}^T\mathbf{D}\mathbf{D}^T\mathbf{B} = \mathbf{I}_d \tag{1.42}$$

Changing $\mathbf{U} = \mathbf{S}^{\mathbf{D}^{\frac{1}{2}}}\mathbf{B}$, and using the cyclic permutation property of traces, it can be shown that the eigen-problem corresponds with the CCA solution, eq. (1.35), that is:

$$E_8(\mathbf{U}) = -tr(\mathbf{U}^T\mathbf{S}_t^{\mathbf{D}^{-\frac{1}{2}}}\mathbf{S}^{\mathbf{X}\mathbf{D}}\mathbf{S}_t^{\mathbf{X}^{-1}}\mathbf{S}^{\mathbf{D}\mathbf{X}}\mathbf{S}_t^{\mathbf{D}^{-\frac{1}{2}}}\mathbf{U}) \text{ s.t. } \mathbf{U}^T\mathbf{U} = \mathbf{I}_d \tag{1.43}$$

It is interesting to point out that CCA can also be recovered using an unweighted regression. Yohai and Garcia⁵⁹ and Tso⁶⁰ have shown that the canonical variates minimize:

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

where $|\cdot|$ denotes determinant. This is equivalent to minimizing eq. (1.1) 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 trace as the loss function.

1.5.4. Weighted extensions

Similarly to PCA and KPCA, for LDA and KLDA there are possible weighted extensions, consider eq. (1.4) and eq. (1.5) when $\Gamma = \mathbf{G}^T$ and $\mathbf{W}_r = (\mathbf{G}^T \mathbf{G})^{-\frac{1}{2}}$:

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

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

Previous equations extend work on weighted LDA approaches by allowing to weight the samples rather than the classes.⁶¹ Similar expressions can be derived for weighted CCA and KCCA, exchanging \mathbf{G}^T for \mathbf{X} , and we omit them in the interest of space.

1.6. K-means and spectral clustering

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

1.6.1. k-means

k-means^{62,63} is one of the most popular unsupervised learning algorithms to solve the clustering problem. *k*-means clustering 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:^{52,64-66}

$$J_5(\mathbf{b}_1, \dots, \mathbf{b}_c) = \sum_{i=1}^c \sum_{j \in C_i} \|\mathbf{d}_j - \mathbf{b}_i\|_2^2 \quad (1.44)$$

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 class i . Eq. (1.44) can be rewritten in matrix form⁵² as:

$$E_{10}(\mathbf{B}, \mathbf{A}) = \|\mathbf{D} - \mathbf{B}\mathbf{A}^T\|_F^2 = tr(\mathbf{S}_w) \quad s.t. \quad \mathbf{A}\mathbf{1}_c = \mathbf{1}_n \quad and \quad a_{ij} \in \{0, 1\} \quad (1.45)$$

where $\mathbf{A} \in \mathbb{R}^{n \times c}$ is the indicator matrix and $\mathbf{B} \in \mathbb{R}^{d \times c}$ is the matrix of centroids. Recall that the equivalence between the *k*-means error function eq. (1.44) and eq. (1.45) is only valid if \mathbf{A} strictly satisfies the constraints. Observe that eq. (1.45) can be derived from the fundamental equation of CA, E_0 , eq. (1.1), where $\Upsilon = \mathbf{I}_d$, $\mathbf{W}_r = \mathbf{I}_d$, $\mathbf{W}_c = \mathbf{I}_n$, $\Gamma = \mathbf{D}$.

The *k*-means algorithm performs coordinate descent in $E_{10}(\mathbf{B}, \mathbf{A})$. Given the actual value of the centroids, \mathbf{B} , the first step finds for each data point \mathbf{d}_j , the \mathbf{a}^j such that one of the columns is one and the rest 0, while minimizing eq. (1.45). Recall that \mathbf{a}^j refers to a column vector with the j^{th} row of \mathbf{A} . 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.

Eliminating \mathbf{B} , eq. (1.45) can be rewritten as:

$$E_{10}(\mathbf{A}) = \|\mathbf{D} - \mathbf{D}\mathbf{A}(\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T\|_F^2 =$$

$$tr(\mathbf{D}^T \mathbf{D}) - tr((\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{D}^T \mathbf{D} \mathbf{A}) \geq \sum_{i=c+1}^{\min(d,n)} \lambda_i \quad (1.46)$$

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 $\mathbf{D}^T \mathbf{D}$.^{64,65} In this case, the error E_{10} is equal to the sum of the residual eigenvalues, i.e. $E_{10} = \sum_{i=c+1}^{\min(d,n)} \lambda_i$. Observe that $\text{tr}((\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{D}^T \mathbf{D} \mathbf{A})$ is a Rayleigh quotient optimization problem with a closed-form eigenvalue solution. This is the spectral relaxation of the k -means algorithm.

1.6.2. Normalized Cuts

Recently, spectral graph methods for clustering have arisen as a solid approach to data clustering, and have grown in popularity.⁶⁵⁻⁶⁹ 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, similarly to Laplacian Eigenmaps, constructs a weighted graph, $M(\mathbf{W}, Q)$, with n nodes $Q = [q_1, \dots, q_n]$, where node i represents a 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 adjacency or affinity matrix (i.e. $\mathbf{W} \in \mathfrak{R}^{n \times n}$) is computed, the clustering problem can be seen as a graph cut problem,⁷⁰ where the goal is to find a partition of the graph that minimizes a particular cost function. A popular cost function is:

$$\text{cut}(M) = \sum_{q_i \in R, q_j \in Q-R} w_{ij} \quad (1.47)$$

where q_i denotes the i node of the Graph M , Q represents all the 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. (1.47), favors partitions containing isolated nodes, and better measures such as normalized cuts⁶⁷ or ratio-cuts⁷¹ have been proposed. Normalized cuts⁶⁷ 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 $\mathbf{L} = \mathbf{S} - \mathbf{W} \in \mathfrak{R}^{n \times n}$, and \mathbf{S} is a diagonal matrix whose elements are the sum of the rows of \mathbf{W} , that is, $s_{ii} = \sum_j w_{ij}$. Ratio-cuts⁷¹ computes the second eigenvector of \mathbf{L} . See^{69,72-74} for a comparison of different spectral clustering algorithms.

Recently,^{66,75} established the connection between kernel k -means and normalized cuts, by means of kernel methods. In this section, we follow a simpler derivation of the same idea with our compact matrix notation, and linked to kernel PCA.⁵² Consider E_0 , eq. (1.1), where $\mathbf{\Gamma} = \phi(\mathbf{D})$, $\mathbf{\Upsilon} = \mathbf{I}_n$, $\mathbf{W}_r = \mathbf{I}_d$, $\mathbf{W}_c = \text{diag}(\mathbf{\Gamma}^T \mathbf{\Gamma} \mathbf{I}_n)^{-\frac{1}{2}}$, the weighted kernelized version of k -means, eq. (1.45), is:

$$E_{11}(\mathbf{B}, \mathbf{A}) = \|(\mathbf{\Gamma} - \mathbf{B} \mathbf{A}^T) \mathbf{W}_c\|_F^2 \quad (1.48)$$

Recall that the weight matrix \mathbf{W}_c weights each sample (columns of $\mathbf{\Gamma}$) differently. In this case, minimizing eq. (1.48) is equivalent to maximizing eq. (1.4), that is:

$$E_{11}(\mathbf{A}) \propto \text{tr}((\mathbf{A}^T \mathbf{W}_c^2 \mathbf{A})^{-1} \mathbf{A}^T \mathbf{W}_c^2 \mathbf{\Gamma}^T \mathbf{\Gamma} \mathbf{W}_c^2 \mathbf{A}) \quad (1.49)$$

where $\mathbf{W} = \mathbf{\Gamma}^T \mathbf{\Gamma}$ is the standard affinity matrix in spectral graph methods. After a change of variable $\mathbf{Z} = \mathbf{A}^T \mathbf{W}_c$, eq. (1.49) can be expressed as:

$$E_{11}(\mathbf{Z}) \propto \text{tr}((\mathbf{Z}\mathbf{Z}^T)^{-1} \mathbf{Z}\mathbf{W}_c \mathbf{\Gamma}^T \mathbf{\Gamma} \mathbf{W}_c \mathbf{Z}^T) \quad (1.50)$$

Eq. (1.50) is the same expression used in Normalized Cuts (Ncuts),^{67,69} considering $\mathbf{W}_c = \mathbf{S}^{-\frac{1}{2}}$ and $\mathbf{W} = \mathbf{\Gamma}^T \mathbf{\Gamma} \in \mathbb{R}^{n \times n}$, where \mathbf{W} is the affinity matrix in Ncuts. Once again, with a LS view of Ncuts, the connection with kernel-PCA becomes evident, and normalization factors are easier to interpret. Moreover, the LS formulation is more general since 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 (directly splitting the samples into c clusters) to cluster in the embedded space^{72,76}. See⁷⁷ for a review of rounding methods and more advanced rounding strategies. In related work, Rahimi and Recht⁷⁸ showed how Normalized Cuts⁶⁷, originally presented as a graph-theoretic algorithm, can be framed as a regression problem, and also point out the problems of sensitivity to outliers. Zass and Shashua⁷³ showed the importance of normalization of the affinity matrix in spectral clustering. Important connections have also been made between clustering and manifold learning. Recently,⁷⁹ showed the connection between the continuous formulation of spectral embedding and Kernel PCA through learning eigenfunctions. Finally, similar relations could be derived for other spectral graph methods such as Ratio-cuts⁷¹ or MinMaxCut⁸⁰.

1.7. Conclusions

In this chapter, we have shown that the LS-WKRRR is a generative model for several CA methods. In particular, we have shown how the fundamental equation of CA E_0 , eq. (1.1), relates to PCA, LDA, CCA, k -means, spectral methods, and kernel extensions. We have derived the coupled symmetric system of eigen-equations to solve E_0 , and showed several alternatives to solve the resulting GEP. The LS formulation of CA has several advantages: (1) provides a clean connection between many CA techniques. It allows understanding the communalities and differences between several CA methods, as well as the intrinsic relationships, (2) helps to understand *normalization* factors in CA methods, (3) suggests new optimization strategies, (4) yields efficient optimization algorithms to solve CA techniques that avoid typical problems when the covariance matrices are rank deficient (e.g. small size sample problem); (5) allows many straight-forward extensions of CA methods (e.g. online learning versions). We have derived weighted extensions for PCA, LDA, CCA, and kernel extensions. Further work must be done to address the equivalence between methods when covariance matrices are rank-deficient and not invertible, and to relate other extensions of CA (e.g. Partial Least-Squares, maximum variance unfolding) to this framework.

Acknowledgment

This material is based upon work supported by the U.S. Naval Research Laboratory under Contract No. N00173-07-C-2040. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the U.S. Naval Research Laboratory. Thanks to Minyoung Kim, Karim Abou-Moustafa, Zaid Harchaoui, Chris Ding and Jordi Soler for helpful comments.

References

1. I. T. Jolliffe, *Principal Component Analysis*. (New York: Springer-Verlag, 1986).
2. B. Schölkopf, A. Smola, and K. Muller, Nonlinear component analysis as a kernel eigenvalue problem., *Neural Computation*. **10**, 1299–1319, (1998).
3. R. A. Fisher, The statistical utilization of multiple measurements, *Annals of Eugenics*. **8**, 376–386, (1938).
4. H. Hotelling, Relations between two sets of variates, *Biometrika*. **28**, 321–377, (1936).
5. B. Mohar, *Some applications of Laplace eigenvalues of graphs*. In G. Hahn and G. Sabidussi (Eds.), *Graph Symmetry: Algebraic Methods and Applications (Vol. NATO ASI Ser. C 497, p. 225-275)*. (Kluwer, 1997).
6. F. de la Torre. A least-squares unified view of pca, lda, cca and spectral graph methods. In *tech. report CMU-RI-TR-08-29, Robotics Institute, Carnegie Mellon University, May, (2008)*.
7. M. Borga. Learning multidimensional signal processing. In *PhD Dissertation. Linköping University, Sweden, (1998)*.
8. S. Roweis and Z. Ghahramani, A unifying review of linear gaussian models., *Neural Computation*. **11**(2), 305–345, (1999).
9. S. Yan, D. Xu, B. Zhang, and H. Zhang, Graph embedding: A general framework for dimensionality reduction, *PAMI*. **29**(1), 40–51, (2007).
10. K. Fukunaga, *Introduction to Statistical Pattern Recognition, Second Edition*. (Academic Press. Boston, MA, 1990).
11. F. de la Torre and T. Kanade. Multimodal oriented discriminant analysis. In *International Conference on Machine Learning*, pp. 177–184, (2005).
12. T. W. Anderson, Estimating linear restrictions on regression coefficients for multivariate normal distributions, *Ann. Math. Statist.* **12**, 327–351, (1951).
13. T. W. Anderson, *An Introduction to Multivariate Statistical Analysis*. (2nd ed. Wiley, New York, 1984).
14. S. S. Haykin, *Adaptive filter theory*. (Prentice-Hall, 1996).
15. L. Scharf, *The SVD and reduced rank signal processing. SVD and Signal Processing, II*. (elsevier, 1991).
16. K. I. Diamantaras, *Principal Component Neural Networks (Theory and Applications)*. (John Wiley & Sons, 1996).
17. F. de la Torre and M. J. Black. Dynamic coupled component analysis. In *Computer Vision and Pattern Recognition*, pp. 643–650, (2001).
18. B. Scholkopf and A. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. (MIT Press, 2002).
19. J. Shawe-Taylor and N. Cristianini, *Kernel Methods for Pattern Analysis*. (Cambridge University Press, 2004).
20. P. Baldi and K. Hornik, Neural networks and principal component analysis: Learning from examples without local minima, *Neural Networks*. **2**, 53–58, (1989).

21. H. Murase and S. K. Nayar, Visual learning and recognition of 3D objects from appearance, *International Journal of Computer vision*. **1**(14), 5–24, (1995).
22. K. J. Bathe and E. Wilson, *Numerical Methods in Finite Element*. (Prentice-Hall. Englewood Cliffs, NJ., 1976).
23. G. Golub and C. F. V. Loan, *Matrix Computations*. (2nd ed. The Johns Hopkins University Press, 1989).
24. A. Buchanan and A. Fitzgibbon. Damped newton algorithms for matrix factorization with missing data. In *IEEE Conference on Computer Vision and Pattern Recognition*, (2005).
25. F. de la Torre and M. J. Black, A framework for robust subspace learning, *International Journal of Computer Vision*. **54**, 117–142, (2003).
26. R. Fletcher, *Practical methods of optimization*. (John Wiley and Sons., 1987).
27. A. Blake and A. Zisserman, *Visual Reconstruction*. (MIT Press series, Massachusetts, 1987).
28. K. Pearson, On lines and planes of closest fit to systems of points in space, *The London, Edinburgh and Dublin Philosophical Magazine and Journal*. **6**, 559–572, (1901).
29. H. Hotelling, Analysis of a complex of statistical variables into principal components, *Journal of Educational Psychology*. **24**, (1933).
30. E. Oja, A simplified neuron model as principal component analyzer, *Journal of Mathematical Biology*. **15**, 267–273, (1982).
31. S. Roweis. EM algorithms for PCA and SPCA. In *NIPS*, pp. 626–632, (1997).
32. K. R. Gabriel and S. Zamir, Lower rank approximation of matrices by least squares with any choice of weights, *Technometrics*, Vol. 21, pp. 21, 489–498, (1979).
33. H. Shum, K. Ikeuchi, and R. Reddy, Principal component analysis with missing data and its application to polyhedral object modeling, *Pattern Analysis and Machine Intelligence*. **17**(9), 855–867, (1995).
34. M. Tipping and C. M. Bishop, Probabilistic principal component analysis, *Journal of the Royal Statistical Society B*. **61**, 611–622, (1999).
35. M. Irani and P. Anandan. Factorization with uncertainty. In *European Conference on Computer Vision*, pp. 539–553, (2000).
36. M. J. Greenacre, *Theory and Applications of Correspondence Analysis*. (Academic Press, London, 1984).
37. I. Tsang and J. Kwok. Distance metric learning with kernels. In *Proceedings of the International Conference on Artificial Neural Networks*, pp. 126–129 (June, 2003).
38. 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).
39. D. Skocaj and A. Leonardis. Weighted and robust incremental method for subspace learning. In *International Conference on Computer Vision ICCV*, pp. 1494–1501, (2003).
40. S. Fidler, D. Skocaj, and A. Leonardis, Combining reconstructive and discriminative subspace methods for robust classification and regression by subsampling, *IEEE Transactions on Pattern Analysis and Machine Intelligence*. **28**(3), 337–350, (2006).
41. P. Aguiar, M. Stosic, and J. Xavier. Spectrally optimal factorization of incomplete matrices. In *IEEE Computer Vision and Pattern Recognition*, (2008).
42. A. Martinez and A. Kak, Pca versus lda, *IEEE Transactions on Pattern Analysis and Machine Intelligence*. **23**(2), 228–233, (2003).
43. P. Belhumeur, J. Hespanha, and D. Kriegman, Eigenfaces vs. fisherfaces: Recognition using class specific linear projection, *IEEE Transactions on Pattern Analysis and Machine Intelligence*. (19), 711–720, (1997).
44. T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference and Prediction*. (Springer, 2001).
45. W. Zhao. Discriminant component analysis for face recognition. In *ICPR*, pp. 818–821, (2000).
46. J. Hoffbeck and D. Landgrebe, Covariance matrix estimation and classification with limited

- training data, *IEEE Transactions on Pattern Analysis and Machine Intelligence*. **18**(7), 763–767, (1996).
47. J. Ye, Characterization of a family of algorithms for generalized discriminant analysis on undersampled problems, *The Journal of Machine Learning Research*. **6**(1), 483 – 502 (September, 2005).
 48. S. Zhang and T. Sim, Discriminant subspace analysis: A fukunaga-koontz approach., *PAMI*. **29**, 1732–1745, (2007).
 49. R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*. (John Wiley and Sons Inc., 2001).
 50. P. Gallinari, S. Thiria, F. Badran, and F. Fogelman-Soulie, On the relations between discriminant analysis and multilayer perceptrons, *Neural Networks*. **4**, 349–360, (1991).
 51. J. Ye. Least squares linear discriminant analysis. In *ICML*, pp. 1087–1093, (2007).
 52. F. de la Torre and T. Kanade. Discriminative cluster analysis. In *International Conference on Machine Learning*, vol. 148, pp. 241 – 248, New York, NY, USA (June, 2006). ACM Press.
 53. S. Mika. Kernel fisher discriminants. In *PhD thesis, University of Technology, Berlin*, (2002).
 54. T.-K. Kim, J. Kittler, and R. Cipolla., Discriminative learning and recognition of image set classes using canonical correlations, *IEEE Trans. on PAMI*. **29**(6), 1005–1018, (2007).
 55. D. Skocaj and A. Leonardis. Appearance-based localization using cca. In *Computer Vision Winter Workshop*, (2000).
 56. T. Melzer, M. Reiter, and H. Bischof, Appearance models based on kernel canonical correlation analysis, *Pattern Recognition*. **36**(9), 1961–1971, (2003).
 57. K. Mardia, J. Kent, and J. Bibby, *Multivariate Analysis*. (Academic Press, London, 1979).
 58. S. Cherry, Singular value decomposition analysis and canonical correlation analysis., *J. Climate*. (9), 2003–2009, (1997).
 59. V. J. Yohai and M. S. Garcia, Canonical variables as optimal predictors., *The Annals of Statistics*. **8**(4), 865–869, (1980).
 60. M. Tso, Reduced-rank regression and canonical analysis., *Journal of the Royal Statistical Society. Series B*. **43**(2), 183–189, (1981).
 61. M. Loog, R. Duin, and R. Hacb-Umbach, Multiclass linear dimension reduction by weighted pairwise fisher criteria, *IEEE Transactions on Pattern Analysis and Machine Intelligence*. **7**(23), 762766, (2001).
 62. J. B. MacQueen. Some methods for classification and analysis of multivariate observations. In *5-th Berkeley Symposium on Mathematical Statistics and Probability*. Berkeley, University of California Press., pp. 1:281–297, (1967).
 63. A. K. Jain, *Algorithms For Clustering Data*. (Prentice Hall, 1988).
 64. H. Zha, C. Ding, M. Gu, X. He, and H. Simon. Spectral relaxation for k-means clustering. In *Neural Information Processing Systems*, pp. 1057–1064, (2001).
 65. C. Ding and X. He. K-means clustering via principal component analysis. In *International Conference on Machine Learning*, vol. 1, pp. 225–232, (2004).
 66. R. Zass and A. Shashua. A unifying approach to hard and probabilistic clustering. In *International Conference on Computer Vision. Beijing*, (2005).
 67. J. Shi and J. Malik, Normalized cuts and image segmentation, *IEEE Transactions on Pattern Analysis and Machine Intelligence*. **22**(8), 888–905 (Aug., 2000).
 68. A. Y. Ng, M. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *Neural Information Processing Systems*, number 14, (2002).
 69. S. Yu and J. Shi. Multiclass spectral clustering. In *Proceedings of International Conference on Computer Vision*, pp. 335–336 (Nice, France, October, 2003).
 70. F. K. Chung, *Spectral Graph Theory*. (CBMS Regional Conference Series in Mathematics, vol 92, American Mathematical Society., Providence, 1997).
 71. L. Hagen and A. Kahng, New spectral methods for ratio cut partitioning and clustering., *IEEE. Trans. on Computed Aided Desgin*. (11), 1074–1085, (1992).

72. D. Verma and M. Meila. Comparison of spectral clustering methods. In *NIPS*, (2003).
73. R. Zass and A. Shashua. Doubly stochastic normalization for spectral clustering. In *Neural Information Processing Systems*, (2006).
74. M. Filippone, F. Camastra, F. Masulli, and S. Rovetta, A survey of kernel and spectral methods for clustering, *Pattern Recognition*. **41**(1), 176–190, (2008).
75. I. S. Dhillon, Y. Guan, and B. Kulis. A unified view of kernel k-means, spectral clustering and graph partitioning. In *UTCS Tech. Report TR-04-25*, (2004).
76. Y. Weiss. Segmentation using eigenvectors: a unifying view. In *ICCV*, (1999).
77. D. Tolliver. Spectral rounding and image segmentation. Technical report, doctoral dissertation, tech. report CMU-RI-TR-06-44, Robotics Institute, Carnegie Mellon University (August, 2006).
78. A. Rahimi and B. Recht. Clustering with normalized cuts is clustering with a hyperplane. In *Statistical Learning in Computer Vision*, (2004).
79. Y. Bengio, P. Vincent, J. Paiement, P. Vincent, and M. Ouimet, Learning eigenfunctions links spectral embedding and kernel pca, *Neural Computation*. (16), 2197–2219, (2004).
80. C. Ding, X. He, H. Zha, M. Gu, and H. Simon. A min-max cut algorithm for graph partitioning and data clustering. In *Proc. IEEE International Conference on Data Mining*, (2001).