Journal of Machine Learning Research 16 (2015) XX-XX Submitted 5/14; Revised 1/15; Published X/15

Linear Dimensionality Reduction: Survey, Insights, and Generalizations

John P. Cunningham JPC2181@COLUMBIA.EDU
Department of Statistics
Columbia University
New York City, USA

Zoubin Ghahramani ZOUBIN@ENG.CAM.AC.UK
Department of Engineering
University of Cambridge
Cambridge, UK

Editor: Gert Lanckriet

Abstract

Linear dimensionality reduction methods are a cornerstone of analyzing high dimensional data, due to their simple geometric interpretations and typically attractive computational properties. These methods capture many data features of interest, such as covariance, dynamical structure, correlation between data sets, input-output relationships, and margin between data classes. Methods have been developed with a variety of names and motivations in many fields, and perhaps as a result the connections between all these methods have not been highlighted. Here we survey methods from this disparate literature as optimization programs over matrix manifolds. We discuss principal component analysis, factor analysis, linear multidimensional scaling, Fisher's linear discriminant analysis, canonical correlations analysis, maximum autocorrelation factors, slow feature analysis, sufficient dimensionality reduction, undercomplete independent component analysis, linear regression, distance metric learning, and more. This optimization framework gives insight to some rarely discussed shortcomings of well-known methods, such as the suboptimality of certain eigenvector solutions. Modern techniques for optimization over matrix manifolds suggest a generic linear dimensionality reduction solver, which accepts as input data and an objective to be optimized, and returns, as output, an optimal low-dimensional projection of the data. This simple optimization framework further allows straightforward generalizations and novel variants of classical methods, which we demonstrate here by creating an orthogonal-projection canonical correlations analysis. More broadly, this survey and generic solver suggest that linear dimensionality reduction can move toward becoming a blackbox, objective-agnostic numerical technology.

1. Introduction

Linear dimensionality reduction methods have been developed throughout statistics, machine learning, and applied fields for over a century, and these methods have become indispensable tools for analyzing high dimensional, noisy data. These methods produce a low-dimensional linear mapping of the original high-dimensional data that preserves some feature of interest in the data. Accordingly, linear dimensionality reduction can be used for visualizing or exploring structure in data, denoising or compressing data, extracting meaningful feature spaces, and more. This abundance of methods, across a variety of data types and fields, suggests a great complexity to the space of linear dimensionality reduction techniques. As such, there has been little effort to consolidate our understanding. Here we survey a host of methods and investigate when a more general optimization framework can improve performance and extend the generality of these techniques.

We begin by defining linear dimensionality reduction (Section 2), giving a few canonical examples to clarify the definition. We then interpret linear dimensionality reduction in a simple optimization framework as a program with a problem-specific objective over orthogonal or unconstrained matrices. Section 3 surveys principal component analysis (PCA; Pearson (1901); Eckart and Young (1936)), multidimensional scaling (MDS; Torgerson (1952); Cox and Cox (2001); Borg and Groenen (2005)), Fisher's linear discriminant analysis (LDA; Fisher (1936); Rao (1948)), canonical correlations analysis (CCA; Hotelling (1936)), maximum autocorrelation factors (MAF; Switzer and Green (1984)), slow feature analysis (SFA; Wiskott and Sejnowski (2002); Wiskott (2003)), sufficient dimensionality reduction (SDR; Fukumizu et al. (2004); Adragni and Cook (2009)), locality preserving projections (LPP; He and Niyogi (2004); He et al. (2005)), undercomplete independent component analysis (ICA; e.g. Hyvarinen et al. (2001)), linear regression, distance metric learning (DML; Kulis (2012); Yang and Jin (2006)), probabilistic PCA (PPCA; Tipping and Bishop (1999); Roweis (1997); Theobald (1975)), factor analysis (FA; Spearman (1904)), several related methods, and important extensions such as kernel mappings and regularizations.

A common misconception is that many or all linear dimensionality reduction problems can be reduced to eigenvalue or generalized eigenvalue problems. Not only is this untrue in general, but it is also untrue for some very well-known algorithms that are typically thought of as generalized eigenvalue problems. The suboptimality of using eigenvector bases in these settings is rarely discussed and is one notable insight of this survey. Perhaps inherited from this eigenvalue misconception, a second common tendency is for practitioners to greedily choose the low-dimensional data: the first dimension is chosen to optimize the problem objective, and then subsequent dimensions are chosen to optimize the objective on a residual or reduced data set. The optimization framework herein shows the limitation of this view. More importantly, the framework also suggests a more generalized linear dimensionality reduction solver that encompasses all eigenvalue problems as well as many other important variants. In this survey we restate these algorithms as optimization programs over matrix manifolds that have a well understood geometry and a well developed optimization literature (Absil et al., 2008). This simple perspective leads to a generic algorithm for linear dimensionality reduction, suggesting that, like numerical optimization more generally, linear dimensionality reduction can become abstracted as a numerical technology for a range of problem-specific objectives. In all, this work: *(i)* surveys the literature on linear dimensionality reduction, *(ii)* gives insights to some rarely discussed shortcomings of traditional approaches, and *(iii)* provides a simple algorithmic template for generalizing to many more problem-specific techniques.

2. Linear dimensionality reduction as a matrix optimization program

We define linear dimensionality reduction as all methods with the problem statement:

Definition 1 (Linear Dimensionality Reduction) *Given n d-dimensional data points $X = [x_1, ..., x_n] \in \mathbb{R}^{d \times n}$ and a choice of dimensionality $r < d$, optimize some objective $f_X(\cdot)$ to produce a linear transformation $P \in \mathbb{R}^{r \times d}$, and call $Y = PX \in \mathbb{R}^{r \times n}$ the low-dimensional transformed data.*

Note that throughout this work we assume without loss of generality that data X is mean-centered, namely $X1 = 0$. To make this definition concrete, we briefly detail two widespread linear dimensionality reduction techniques: principal component analysis (PCA; Pearson (1901)) and canonical correlations analysis (CCA; Hotelling (1936)). PCA maximizes data variance captured by the low-dimensional projection, or equivalently minimizes the reconstruction error (under the ℓ_2-norm) of the projected data points with the original data, namely:

$$f_X(M) = ||X - MM^\top X||_F^2.$$

Here M is a matrix with r orthonormal columns. In the context of Definition 1, optimizing $f_X(M)$ produces an M such that $P = M^\top$, and the desired low-dimensional projection is $Y = M^\top X$. PCA is discussed in depth in Section 3.1.1.

We stress that the notation of M and P in Definition 1 is not redundant, but rather is required for other linear dimensionality reduction techniques where the linear transformation P does not equal the optimization variable M (as it does in PCA). Consider CCA, another classical linear dimensionality reduction technique that jointly maps two data sets $X_a \in \mathbb{R}^{d_a \times n}$ and $X_b \in \mathbb{R}^{d_b \times n}$ to $Y_a \in \mathbb{R}^{r \times n}$ and $Y_b \in \mathbb{R}^{r \times n}$, such that the sample correlation between Y_a and Y_b is maximized[1]. Under the additional constraints that Y_a and Y_b have uncorrelated variables ($Y_a Y_b^\top = \Lambda$, a diagonal matrix) and be individually uncorrelated with unit variance ($\frac{1}{n} Y_a Y_a^\top = \frac{1}{n} Y_b Y_b^\top = I$), a series of standard steps produces the well known objective:

$$f_X\left(M_a, M_b\right) = \frac{1}{r} \mathrm{tr}\left(M_a^\top (X_a X_a^\top)^{-1/2} X_a X_b^\top (X_b X_b^\top)^{-1/2} M_b\right),$$

as will be detailed in depth in Section 3.1.4. This objective is maximized when $M_a^\top$ and $M_b^\top$ are the left and right singular vectors of the matrix $(X_a X_a^\top)^{-1/2} X_a X_b^\top (X_b X_b^\top)^{-1/2}$. In the context of Definition 1, the low dimensional canonical variables Y_a are then related to the original data as $Y_a = P_a X_a \in \mathbb{R}^{r \times n}$, where $P_a = M_a^\top \left(X_a X_a^\top\right)^{-1/2}$ (and similar for Y_b). Since M_a has by definition orthonormal columns, CCA, by inclusion of the whitening term $\left(X_a X_a^\top\right)^{-1/2}$, does not represent an orthogonal projection of the data. Accordingly, CCA and PCA point out two key features of linear dimensionality reduction and Definition 1: first, that the objective function $f_X(\cdot)$ need not entirely define the linear mapping P to the low-dimensional space; and second, that not all linear dimensionality reduction methods need be orthogonal projections, or indeed projections at all.

[1] As a point of technical detail, note that the use of two data sets and mappings is only a notational convenience; writing the CCA projection as $Y = \begin{bmatrix} Y_a \\ Y_b \end{bmatrix} = \begin{bmatrix} P_a & 0 \\ 0 & P_b \end{bmatrix} \begin{bmatrix} X_a \\ X_b \end{bmatrix} = PX$, we see that CCA adheres precisely to Definition 1.

Note also that both PCA and CCA result in a matrix decomposition, and indeed a common approach to many linear dimensionality reduction methods is to attempt to cast the problem as an eigenvalue or generalized eigenvalue problem (Burges, 2010). This pursuit can be fruitful but is limited, often resulting in ad hoc or suboptimal algorithms. As a specific example, in many settings orthogonal projections of data are required for visualization and other basic needs. Can we create an *Orthogonal CCA*, where we seek orthogonal projections $Y_a = M_a^\top X_a$ for a matrix M_a with orthonormal columns (and similar for Y_b), such that the sample correlation between Y_a and Y_b is maximized? No known eigenvalue problem can produce this projection, so one tempting and common approach is to orthonormalize P_a and P_b found above from traditional CCA. We will show that this choice can be significantly suboptimal, and in later sections we will create Orthogonal CCA using a generic optimization program. Thus matrix decomposition approaches suggest an unfortunate limitation to the set of possible linear dimensionality reduction problems, and a broader framework is required to fully capture Definition 1 and linear dimensionality reduction.

2.1 Optimization framework for linear dimensionality reduction

All linear dimensionality reduction methods presented here can be viewed as solving an optimization program over a matrix manifold $\mathcal{M}$, namely:

$$\begin{aligned} \text{minimize} \quad & f_X(M) \\ \text{subject to} \quad & M \in \mathcal{M}. \end{aligned} \tag{1}$$

Given Definition 1, the intuition behind this optimization program should be apparent: the objective $f_X(\cdot)$ defines the feature of interest to be captured in the data, and the matrix manifold encodes some aspects of the linear mapping P such that $Y = PX$.[2]

All methods considered here specify M as one of two matrix forms. First, some methods are unconstrained optimizations over rank r linear mappings, implying the trivial manifold constraint of Euclidean space, which we denote as $M \in \mathbb{R}^{d\times r}$. In this case, optimization may be straightforward, and algorithms like expectation-maximization (Dempster et al., 1977) or standard first order solvers have been well used.

Second, very often the matrix form will have an orthogonality constraint $\mathcal{M} = \{M \in \mathbb{R}^{d\times r} \ : \ M^\top M = I\}$, corresponding to orthogonal projections of the data X. In this case we write $\mathcal{M} = \mathcal{O}^{d\times r}$. As noted previously, the typical and often flawed approach is to attempt to cast these problems as eigenvalue problems. Instead, viewed through the lens of Equation 1, linear dimensionality reduction is simply an optimization program over a matrix manifold, and indeed there is a well-developed optimization literature for matrix manifolds (foundations include Luenberger (1972); Gabay (1982); Edelman et al. (1998); an excellent summary is Absil et al. (2008)).

As a primary purpose of this work is to survey linear dimensionality reduction, we first detail linear dimensionality reduction techniques using this optimization framework. We then implement a generic solver for programs of the form Equation 1, where $\mathcal{M}$ is the family of orthogonal matrices $\mathcal{O}^{d\times r}$. Thus we show the framework of Equation 1 to be not

[2] Note that several methods will require optimization over additional auxiliary unconstrained variables, which can be addressed algorithmically via a coordinate descent approach (alternating optimizations over the auxiliary variable and Equation 1) or some more nuanced scheme.

only conceptually simplifying, but also algorithmically simplifying. Instead of resorting to ad hoc (and often suboptimal) formulations for each new problem in linear dimensionality reduction, practitioners need only specify the objective $f_X(\cdot)$ and the high-dimensional data X, and these numerical technologies can produce the desired low-dimensional data. Section 4 validates this claim by applying this generic solver without change to different objectives $f_X(\cdot)$, both classic and novel. We require only the condition that $f_X(\cdot)$ be differentiable in M to enable simple gradient descent methods. However, this choice is a convenience of implementation and not a fundamental issue, and thus approaches for optimization of nondifferentiable objectives over nonconvex sets (here $\mathcal{O}^{d\times r}$) could be readily introduced to remove this restriction (for example, Boyd et al. (2011)).

3. Survey of linear dimensionality reduction methods

We now review linear dimensionality reduction techniques using the framework of Section 2, to understand the problem-specific objective and manifold constraint of each method.

3.1 Linear dimensionality reduction with orthogonal matrix constraints

Amongst all dimensionality reduction methods, the most widely used techniques are orthogonal projections. These methods owe their popularity in part due to their simple geometric interpretation as a low-dimensional view of high-dimensional data. This interpretation is of great comfort to many application areas, since these methods do not artificially create or exaggerate many types of structure in the data, as is possible with other models that encode strong prior assumptions.

3.1.1 Principal Component Analysis

Principal component analysis (PCA) was originally formulated by Pearson (1901) as a minimization of the sum of squared residual errors between projected data points and the original data $f_X(M) = ||X - MM^\top X||_F^2 = \sum_{i=1}^n ||x_i - MM^\top x_i||_2^2$. Modern treatments tend to favor the equivalent "maximizing variance" derivation (e.g., (Bishop, 2006)), resulting in the objective $-\mathrm{tr}(M^\top XX^\top M)$. We write PCA in the formulation of Equation 1 as:

$$\begin{array}{ll} \text{minimize} & ||X - MM^\top X||_F^2 \\ \text{subject to} & M \in \mathcal{O}^{d\times r}. \end{array} \tag{2}$$

Equation 2 leads to the familiar SVD solution: after summarizing the data by its sample covariance matrix $\frac{1}{n}XX^\top$, the decomposition $XX^\top = Q\Lambda Q^\top$ produces an optimal point $M = Q_r$, where Q_r denotes the columns of Q associated with the largest r eigenvalues of $XX^\top$ (Eckart and Young, 1936; Mirsky, 1960; Golub and Van Loan, 1996).

There are many noteworthy extensions to PCA. A first example is kernel PCA, which uses PCA on a feature space instead of the inputs themselves (Schölkopf et al., 1999), and indeed some dimensionality reduction methods and their kernelized counterparts can be considered together as kernel regression problems (De la Torre, 2012). While quite important for all machine learning methods, we consider kernelized methods orthogonal to much of the presentation here, since using this kernel mapping is a question of representation of data, not of the dimensionality reduction algorithm itself.

Second, there have been several probabilistic extensions to PCA, such as probabilistic PCA (PPCA, (Tipping and Bishop, 1999; Roweis, 1997)), extreme component analysis (XCA, (Welling et al., 2003)), and minor component analysis (MCA, (Williams and Agakov, 2002)). These algorithms all share a common purpose (modeling covariance) and the same coordinate system for projection (the principal axes of the covariance ellipsoid), even though they differ in the particulars of the projection and which basis is chosen from that coordinate system. We present PPCA as a separate algorithm below and leave the others as extensions of this core method.

Third, extensions have introduced outlier insensitivity via a different implicit noise model such as a Laplace observation model, leading to a few examples of robust PCA (Galpin and Hawkins, 1987; Baccini et al., 1996; Choulakian, 2006). An alternative approach to robust PCA is driven by the observation that a small number of highly corrupted observations can drastically influence standard PCA. Candes et al. (2011) takes this approach to robust PCA, considering the data as low-rank plus sparse noise. Their results have particular theoretical and practical appeal and connect linear dimensionality reduction to the substantial nuclear-norm minimization literature.

Fourth, PCA has been made sparse in several contexts (Zou et al., 2006; d'Aspremont et al., 2007, 2008; Journee et al., 2010), where the typical PCA objective is augmented with a lasso-type ℓ_1 penalty term, namely $f_X(M) = ||X - MM^\top X||_F^2 + \lambda||M||_1$, with penalty term λ and $||M||_1 = \sum_i \sum_j |M_{ij}|$. This objective does not admit an eigenvalue approach, and as a result several specialized algorithms have been proposed. Note however that this sparse objective is again simply a program over $\mathcal{O}^{d\times r}$ (albeit nondifferentiable).

Fifth, another class of popular extensions generalizes PCA to other exponential family distributions, beyond the implicit normal distribution of standard PCA (Collins et al., 2002; Mohamed et al., 2008). These methods, while important, result in nonlinear mappings of the data and thus fall outside the scope of Definition 1. Additionally, there are other nonlinear extensions to PCA; Chapter 12.6 of Hyvarinen et al. (2001) gives an overview.

3.1.2 Multidimensional Scaling

Multidimensional scaling (MDS; Torgerson (1952); Cox and Cox (2001); Borg and Groenen (2005)) is a class of methods and a large literature in its own right, but its connections to linear dimensionality reduction and PCA are so well-known that it warrants individual mention. PCA minimizes low-dimensional reconstruction error, but another sensible objective is to maximize the scatter of the projection, under the rationale that doing so would yield the most informative projection (this choice is sometimes called classical scaling). Defining our projected points $y_i = M^\top x_i$ for some $M \in \mathcal{O}^{d\times r}$, MDS seeks to maximize pairwise distances $\sum_i \sum_j ||y_i - y_j||^2$.

MDS leads to the seemingly novel optimization program (Equation 1) over the scatter objective $f_X(M) = \sum_i \sum_j ||M^\top x_i - M^\top x_j||^2$, which can be expanded as:

$$f_X(M) \propto \operatorname{tr}\left(M^\top XX^\top M\right) - 1^\top X^\top MM^\top X1 = \operatorname{tr}\left(M^\top X\left(I - \frac{1}{n}11^\top\right)X^\top M\right), \quad (3)$$

where we denote the vector of all ones as 1. Noting that X has zero mean by definition and thus $X(I - \frac{1}{n}11^\top) = X$, we see classical MDS is precisely the 'maximal variance' PCA objective $\operatorname{tr}(M^\top XX^\top M)$.

The equivalence of MDS and PCA in this special case is well-known (Cox and Cox, 2001; Borg and Groenen, 2005; Mardia et al., 1979; Williams, 2002), and indeed this particular example only scratches the surface of MDS, which is usually considered in much more general terms. Specifically, if we have available only pairwise distances $d_X(x_i, x_j)$, a more general MDS problem statement is to fit the low-dimensional data so as to preserve these pairwise distances as closely as possible in the least squares sense: minimizing $\sum_i \sum_j (d_X(x_i, x_j) - d_Y(y_i, y_j))^2$ is known as Kruskal-Shephard scaling, and the distance metrics can be arbitrary and different between the original and low-dimensional data. First, it is worth noting that least squares is by no means the only appropriate stress function on the distances d_X and d_Y; a Sammon mapping is another common choice (see for example Hastie et al. (2008), §14.8). Second, MDS does not generally require the data itself, but only the pairwise dissimilarities $d_{ij} = d_X(x_i, x_j)$, which is often a useful property. When the data is known, we see here that if we specify a low-dimensional orthogonal projection $Y = M^\top X$, then indeed this objective will result in a class of linear dimensionality reduction programs:

$$\begin{array}{ll} \text{minimize} & \displaystyle\sum_i \sum_j \left(d_X\left(x_i, x_j\right) - d_Y\left(M^\top x_i, M^\top x_j \right) \right)^2 \\ \text{subject to} & M \in \mathcal{O}^{d\times r}. \end{array} \tag{4}$$

Special approaches exist to solve this program on a case-by-case basis (Cox and Cox, 2001; Borg and Groenen, 2005). However, by broadly considering Equation 4 as an optimization over orthogonal projections, we again see the motivation for a generic numerical solver for this class of problems, obviating objective-specific methods.

Of course, the low-dimensional data Y need not be a linear mapping of X (indeed, in many cases the original points X are not even available). This more general form of MDS is used in a variety of nonlinear dimensionality reduction techniques, including prominently Isomap (Tenenbaum et al., 2000), as discussed below in Section 3.3.

3.1.3 Linear Discriminant Analysis

Another natural problem-specific objective occurs when the data X has associated class labels, of which Fisher's linear discriminant analysis (LDA; Fisher (1936); Rao (1948); modern references include Fukunaga (1990); Bishop (2006)) is perhaps the most prominent example. The purpose of LDA is to project the data in such a way that separation between classes is maximized. To do so, LDA begins by partitioning the data covariance $XX^\top$ into covariance contributed within each of the c classes (Σ_W) and covariance contributed between the classes (Σ_B) , such that $XX^\top = \Sigma_W + \Sigma_B$ for:

$$\Sigma_W = \sum_{i=1}^{n} (x_i - \mu_{c_i})(x_i - \mu_{c_i})^\top \qquad\qquad \Sigma_B = \sum_{i=1}^{n} (\mu_{c_i} - \mu)(\mu_{c_i} - \mu)^\top, \tag{5}$$

where μ is the global data mean (here $\mu = 0$ by definition) and μ_{c_i} is the class mean associated with data point x_i. LDA seeks the projection that maximizes between-class variability $\operatorname{tr}\left(M^\top \Sigma_B M\right)$ while minimizing within-class variability $\operatorname{tr}\left(M^\top \Sigma_W M\right)$, leading to the optimization program:

$$\begin{aligned} &\text{maximize} && \frac{\operatorname{tr}\left(M^\top \Sigma_B M\right)}{\operatorname{tr}\left(M^\top \Sigma_W M\right)} \\ &\text{subject to} && M \in \mathcal{O}^{d\times r}. \end{aligned} \tag{6}$$

This objective appears very much like a generalized Rayleigh quotient, and is so for $r = 1$. In this special case, $M \in \mathcal{O}^{d\times 1}$ can be found as the top eigenvector of $\Sigma_W^{-1}\Sigma_B$, which can be seen by substituting $L = \Sigma_W^{1/2}M$ into Equation 6 above. This one-dimensional LDA projection is appropriate when there are $c = 2$ classes.

A common misconception is that LDA for higher dimensional projections $r > 1$ can be solved with a greedy selection of the top r eigenvectors of $\Sigma_W^{-1}\Sigma_B$. However, this is certainly not the case, as the top r eigenvectors of $\Sigma_W^{-1}\Sigma_B$ will not in general be orthogonal. The eigenvector solution solves the similar but not equivalent objective $\operatorname{tr}\left(\left(M^\top\Sigma_W M\right)^{-1}\left(M^\top\Sigma_B M\right)\right)$ over $M \in \mathbb{R}^{d\times r}$; these two objectives and a few others are nicely discussed in Chapter 10 of Fukunaga (1990). While each of these choices has its merits, in the common case that one seeks a projection of the original data, the orthogonal M produced by solving Equation 6 is more appropriate. Though rarely discussed, this misconception between the trace-of-quotient and the quotient-of-traces has been investigated in the literature (Yan and Tang, 2006; Shen et al., 2007).

The commonality of this misconception adds additional motivation for this work, to survey and consolidate a fragmented literature. Second, this misconception also points out the limitations of eigenvector approaches: even when considered the standard algorithm for a popular method, eigenvalue decompositions may in fact be an inappropriate choice. Third, as Equation 6 is a simple program over orthogonal projections, we see again the utility of a generic solver, an approach which should outperform traditional approaches (and indeed does, as Section 4 will show).

In terms of extensions, we note a few key constraints of classical LDA: each data point must be labeled with a class (no missing observations), each data point must be labeled with only one class (no mixed membership), and the class boundaries are modeled as linear. As a first extension, one might have incomplete class labels; Yu et al. (2006) extends LDA (with a probabilistic PCA framework; see Section 3.2.2) to the semi-supervised setting where not all points are labeled. Second, data points may represent a mixture of multiple features, such that one wants to extract a projection where one feature is most discriminable. Brendel et al. (2011) offers a possible solution by marginalizing covariances over each feature of interest. Third, Mika et al. (1999) has extended LDA to the nonlinear domain via kernelization, which has also been well used.

3.1.4 Canonical Correlations Analysis

Canonical correlation analysis (CCA) is a problem of joint dimensionality reduction: given two data sets $X_a \in \mathbb{R}^{d_a\times n}$ and $X_b \in \mathbb{R}^{d_b\times n}$, find low-dimensional mappings $Y_a = P_aX_a$ and $Y_b = P_bX_b$ that maximize the correlation between Y_a and Y_b, namely:

$$\rho\left(y_a, y_b\right) = \frac{E\left(y_a^\top y_b\right)}{\sqrt{E\left(y_a^\top y_a\right)E\left(y_b^\top y_b\right)}} = \frac{\operatorname{tr}\left(Y_aY_b^\top\right)}{\sqrt{\operatorname{tr}\left(Y_aY_a^\top\right)\operatorname{tr}\left(Y_bY_b^\top\right)}} = \frac{\operatorname{tr}\left(P_aX_aX_b^\top P_b^\top\right)}{\sqrt{\operatorname{tr}\left(P_aX_aX_a^\top P_a^\top\right)\operatorname{tr}\left(P_bX_bX_b^\top P_b^\top\right)}}. \tag{7}$$

CCA was originally derived in Hotelling (1936); more modern treatments include Muirhead (2005); Timm (2002); Hardoon et al. (2004); Hardoon and Shawe-Taylor (2009). This method in its classical form, which we call *Traditional CCA*, seeks to maximize $\rho\left(y_a, y_b\right)$ under the constraint that all variables are uncorrelated and of unit variance: $\frac{1}{n}Y_aY_a^\top = I$, $\frac{1}{n}Y_bY_b^\top = I$, and $Y_aY_b^\top = \Lambda$ for some diagonal matrix Λ. As an optimization program over P_a and P_b, Traditional CCA solves:

$$\begin{aligned} \text{maximize} \quad & \frac{\operatorname{tr}\left(P_aX_aX_b^\top P_b^\top\right)}{\sqrt{\operatorname{tr}\left(P_aX_aX_a^\top P_a^\top\right)\operatorname{tr}\left(P_bX_bX_b^\top P_b^\top\right)}} \\ \text{subject to} \quad & \frac{1}{n}P_aX_aX_a^\top P_a^\top = I \\ & \frac{1}{n}P_bX_bX_b^\top P_b^\top = I \\ & P_aX_aX_b^\top P_b^\top = \Lambda. \end{aligned} \tag{8}$$

Using the substitution $P_a = M_a^\top\left(X_aX_a^\top\right)^{-1/2}$ for $M_a \in \mathcal{O}^{d_a\times r}$ (and similar for P_b), Traditional CCA reduces to the well known objective:

$$\begin{aligned} \text{maximize} \quad & \operatorname{tr}\left(M_a^\top(X_aX_a^\top)^{-1/2}X_aX_b^\top(X_bX_b^\top)^{-1/2}M_b\right) \\ \text{subject to} \quad & M_a \in \mathcal{O}^{d_a\times r} \\ & M_b \in \mathcal{O}^{d_b\times r}. \end{aligned} \tag{9}$$

This objective is maximized when $M_a^\top$ is the top r left singular vectors and $M_b^\top$ is the top r right singular vectors of $(X_aX_a^\top)^{-1/2}X_aX_b^\top(X_bX_b^\top)^{-1/2}$. The linear transformations optimizing Equation 8 are then calculated as $P_a = M_a^\top(X_aX_a^\top)^{-1/2}$, and similar for P_b. This solution is provably optimal for any dimensionality r under the imposed constraints (Muirhead, 2005).

It is apparent by construction that P_a and P_b do not in general represent orthogonal projections (except when $X_aX_a^\top = I$ and $X_bX_b^\top = I$, respectively), and thus Traditional CCA is unsuitable for common settings (such as visualization of data in an orthogonal axis) where an orthogonal mapping is required. In these cases, a common heuristic approach is to orthogonalize P_a and P_b to produce orthogonal mappings of the data $Y_a = M_a^\top X_a$ and $Y_b = M_b^\top X_b$. This heuristic choice, however, produces suboptimal results for the original correlation objective of Equation 7 for all dimensions $r > 1$ (the $r = 1$ case is trivially an orthogonal projection), as the results will show.

Our approach addresses a desire for orthogonal projections directly: with the optimization framework of Equation 1, we can immediately write down a novel linear dimensionality reduction method that preserves Hotelling's original objective but is properly generalized to produce orthogonal projections. We call this method *Orthogonal CCA*, maximizing the correlation $\rho\left(y_a, y_b\right)$ objective directly over orthogonal matrices:

$$\begin{aligned} \text{maximize} \quad & \frac{\operatorname{tr}\left(M_a^\top X_aX_b^\top M_b\right)}{\sqrt{\operatorname{tr}\left(M_a^\top X_aX_a^\top M_a\right)\operatorname{tr}\left(M_b^\top X_bX_b^\top M_b\right)}} \\ \text{subject to} \quad & M_a \in \mathcal{O}^{d_a\times r} \\ & M_b \in \mathcal{O}^{d_b\times r}. \end{aligned} \tag{10}$$

The resulting low-dimensional mappings are then the orthogonal projections that we desire: $Y_a = M_a^\top X_a$ and $Y_b = M_b^\top X_b$. The optimization program of Equation 10 can not be solved with a known matrix decomposition, thus requiring a direct optimization approach. More importantly, we point out the meaningful difference between Traditional CCA and Orthogonal CCA: Traditional CCA whitens each data set X_a and X_b, and then orthogonally projects these whitened data into a common space such that correlation is maximized. Orthogonal CCA on the other hand preserves the covariance of the original data X_a and X_b, finding orthogonal projections where correlation is maximized without the initial whitening step. It is unsurprising then that these two methods should return different mappings, even when the Traditional CCA result is orthogonalized post hoc. Accordingly, CCA demonstrates the utility of considering linear dimensionality reduction in the framework of Equation 1; methods can be directly written down for the objective and projection of interest, without having to shoehorn the problem into an eigenvector decomposition.

3.1.5 Maximum Autocorrelation Factors

There are a number of linear dimensionality reduction methods that seek to preserve temporally interesting structure in the projected data. A first simple example is maximum autocorrelation factors (MAF; Switzer and Green (1984); Larsen (2002)). Suppose the high-dimensional data $X \in I\!R^{d\times n}$ has data points x_t for $t \in \{1, ..., n\}$, and that the index label t defines an order in the data. In such a setting, the structure of interest for the low-dimensional representation may have nothing to do with modeling data covariance (like PCA), but rather the appropriate description should include temporal structure.

Assume that there is an underlying r-dimensional temporal signal that is smooth, and that the remaining $d - r$ dimensions are noise with little temporal correlation (less smooth). MAF then seeks an orthogonal projection $P = M^\top$ for $M \in \mathcal{O}^{d\times r}$ so as to maximize correlation between adjacent points $y_t, y_{t+\delta}$, yielding the following objective:

$$f_X(M) = \rho(y_t, y_{t+\delta}) = \frac{E(y_t^\top y_{t+\delta})}{\sqrt{E(y_t^2)E(y_{t+\delta}^2)}} = \frac{E(x_t^\top MM^\top x_{t+\delta})}{E(x_t^\top MM^\top x_t)} = \frac{\mathrm{tr}(M^\top \Sigma_\delta M)}{\mathrm{tr}(M^\top \Sigma M)}, \tag{11}$$

where Σ is the empirical covariance of the data $E(x_t x_t^\top) = \frac{1}{n} XX^\top$ and Σ_δ is the symmetrized empirical cross-covariance of the data evaluated at a one-step time lag $\Sigma_\delta = \frac{1}{2}\left(E(x_{t+\delta}x_t^\top) + E(x_t x_{t+\delta}^\top)\right)$. This objective results in the linear dimensionality program:

$$\begin{aligned} \text{maximize} \quad & \frac{\mathrm{tr}(M^\top \Sigma_\delta M)}{\mathrm{tr}(M^\top \Sigma M)} \\ \text{subject to} \quad & M \in \mathcal{O}^{d\times r}, \end{aligned} \tag{12}$$

Note again the appearance of the quotient-of-traces objective (as in LDA and CCA). Indeed, the same heuristic (solving the trace-of-quotient problem) is typically applied to MAF, which results in the standard choice of the top eigenvectors of $\Sigma^{-1}\Sigma_\delta$ as the solution to Equation 12. Though correct in the $r = 1$ case, this misconception is incorrect for precisely the same reasons as above with LDA, and its use results in the same pitfalls. Directly solving the manifold optimization of Equation 1 presents a more straightforward option.

MAF can be seen as a method balancing the desire for cross-covariance (Σ_δ) of the data without overcounting data that has high power (the denominator containing Σ). Indeed, such methods have been invented with slight variations in various application areas (e.g., Cunningham and Yu (2014)). For example, one might simply ask to maximize the cross-covariance $E(y_t^\top y_{t+\delta})$ rather than the correlation itself. Doing so results in a simpler problem than Equation 12: maximize $\text{tr}(M^\top \Sigma_\delta M)$ for $M \in \mathcal{O}^{d\times r}$. In this case the eigenvector solution is optimal. Second, we may want to maximize (or minimize, as in Turner and Sahani (2007)) the squared distance between projected points; the objective then becomes $E(||y_{t+\delta} - y_t||^2)$, which through a similar set of steps produces the similar eigenvalue problem $\text{tr}\left(M^\top(\Sigma - \Sigma_\delta)M\right)$ for $M \in \mathcal{O}^{d\times r}$. This last choice is a discrete time analog of a more popular method—slow feature analysis—which we discuss in the next section. Third, one might want to specify a particular form of temporal structure in terms of a dynamics objective $f_X(M)$, and seek linear projections containing that structure. The advantage of such an approach is that one can specify a range of dynamical structures well beyond the statistics captured by an autocorrelation matrix. A recent simple example is Churchland et al. (2012), who sought a linear subspace of the data where linear dynamics were preserved, namely an M minimizing $f_X(M) = ||\dot{X} - MDM^\top X||_F^2$ for some dynamics matrix $D \in \mathbb{R}^{r\times r}$. This objective is but one simple choice of dynamical structure; given the canonical autonomous system $\dot{y} = g(y) + \epsilon$, one might similarly optimize $f_X(M) = ||M^\top \dot{X} - g(M^\top X)||_F^2$. Optimizing such a program finds the projection of the data that optimally expresses that dynamical feature of interest, without danger of artificially creating that structure based on a strong prior model (as is possible in state-space models like the Kalman filter (Kalman, 1960)).

3.1.6 Slow Feature Analysis

Similar in spirit to MAF, slow feature analysis (SFA; Wiskott and Sejnowski (2002); Wiskott (2003)) is a linear dimensionality reduction technique developed to seek invariant representations in object recognition problems. SFA assumes that measured data, such as pixels in a movie, can have rapidly changing values over time, whereas the identity, pose, or position of the underlying object should move much more slowly. Thus, recovering a slowly moving projection of data may produce a meaningful representation of the true object of interest. Accordingly, assuming access to derivatives $\dot{X} = [\dot{x}_1, ..., \dot{x}_n]$, SFA minimizes the trace of the covariance of the projection $\text{tr}(\dot{Y}\dot{Y}^\top) = \text{tr}(M^\top \dot{X}\dot{X}^\top M)$. This objective is PCA on the derivative data:

$$\begin{aligned} &\text{minimize} \quad && \text{tr}\left(M^\top \dot{X}\dot{X}^\top M\right) \\ &\text{subject to} \quad && M \in \mathcal{O}^{d\times r}. \end{aligned} \tag{13}$$

Linear SFA is the most straightforward case of the class of SFA methods. Several additional choices are typical in SFA implementations, including: *(i)* data points $x_t \in X$ are usually expanded nonlinearly via some feature mapping $h : \mathbb{R}^d \to \mathbb{R}^p$ for some $p > d$ (a typical choice is all monomials of degree one and two to capture linear and quadratic effects); and *(ii)* data are whitened to prevent the creation of structure due to the mapping $h(\cdot)$ alone before the application of the PCA-like program in Equation 13. A logical extension of this nonlinear feature space mapping is to consider a reproducing kernel Hilbert space mapping, as has indeed been done (Bray and Martinez, 2002). Turner and Sahani (2007) established the connections between SFA and linear dynamical systems, giving a probabilistic interpretation of SFA that also makes different and interesting connections of this method to PCA and its probabilistic counterpart (Section 3.2.2).

3.1.7 Sufficient Dimensionality Reduction

Consider a supervised learning problem with high dimensional covariates $X \in \mathbb{R}^{d \times n}$ and responses $Z \in \mathbb{R}^{\ell \times n}$. The concept behind sufficient dimensionality reduction is to find an orthogonal projection of the data $Y = M^\top X \in \mathbb{R}^{r \times n}$ such that the reduced-dimension points Y capture all statistical dependency between X and Z. Thus, sufficient dimensionality reduction (SDR) is a problem of feature selection that seeks an $M \in \mathcal{O}^{d \times r}$ which makes covariates and responses conditionally independent:

$$p_{Z|X}(z|x) = p_{Z|M^\top X}(z|M^\top x) \quad \Longleftrightarrow \quad Z \perp\!\!\!\perp X | M^\top X. \tag{14}$$

SDR is in fact a class of methods, as there are a number of ways one might derive an objective for such a conditional independence relationship. Particularly popular in machine learning is the use of kernel mappings to characterize the conditional independence relationship of Equation 14 (Fukumizu et al., 2004, 2009; Nilsson et al., 2007). The essential idea in these works is to map covariates X and responses Z into reproducing kernel Hilbert spaces, where it has been shown that, for universal kernels, cross-covariance operators can be used to determine conditional independence of X and Z (Fukumizu et al., 2004; Gretton et al., 2012, 2005). Such an approach induces a cost function on the projection:

$$f_X(M) = J(Z, M^\top X) := \mathrm{tr}\left(\bar{K}_Z \left(\bar{K}_{M^\top X} + n\epsilon I\right)^{-1}\right), \tag{15}$$

where $\bar{K}_Z = \left(I - \frac{1}{n}11^\top\right) K_Z \left(I - \frac{1}{n}11^\top\right)$ is the centered Gram matrix $K_Z = \{k(z_i, z_j)\}_{ij}$ (and similar for $\bar{K}_{M^\top X}$). Critically, this cost function is provably larger than $J(Z, X)$, with equality if and only if the desired conditional independence of Equation 14 holds. Thus, we have the following linear dimensionality reduction program:

$$\begin{array}{ll} \text{minimize} & \mathrm{tr}\left(\bar{K}_Z \left(\bar{K}_{M^\top X} + n\epsilon I\right)^{-1}\right) \\ \text{subject to} & M \in \mathcal{O}^{d \times r}. \end{array} \tag{16}$$

SDR has been extended to the unsupervised case (Wang et al., 2010) and has been implemented with other objectives such as the Hilbert-Schmidt independence criterion (Gretton et al., 2005). An important review of non-kernel SDR techniques is Adragni and Cook (2009), in addition to earlier work (Li, 1991).

3.1.8 Locality Preserving Projections

All methods considered thus far stipulate objectives based on global loss functions, which can be sensitive to outliers and can be significantly distorted by nonlinear structure in the data. A popular alternative throughout machine learning is to consider local neighborhood structure. In the case of dimensionality reduction, considering locality often amounts to constructing a neighborhood graph of the training data, and using that graph to define the loss function. Numerous *nonlinear* methods have been proposed along these lines (see Section 3.3), and this development has led to a few important *linear* methods that consider local structure. First, locality preserving projections (LPP) (He and Niyogi, 2004) is a direct linear interpretation of Laplacian Eigenmaps (Belkin and Niyogi, 2003). LPP begins by defining a graph with each data point $x_i \in I\!R^d$ as a vertex, connecting x_i and x_j with the edge $\delta_{i,j}$ if these points are in the same ϵ neighborhood (that is, $||x_i - x_j|| < \epsilon$). A kernel (typically the squared exponential kernel) is then used to weight the existing edges. The cost of the reconstruction $y_i = Px_i$ is then:

$$\sum_{i=1}^{n}\sum_{j=1}^{n}||Px_i - Px_j||_2^2 W_{ij} \quad , \quad \text{where} \quad W_{ij} = \delta_{i,j} \exp\left\{-\frac{1}{\tau}||x_i - x_j||_2^2\right\}. \tag{17}$$

Through a few standard steps (see for example Belkin and Niyogi (2003); He and Niyogi (2004)), this objective results in the linear dimensionality objective:

$$\begin{aligned} &\text{minimize} \quad \text{tr}\left(PXLX^\top P^\top\right) \\ &\text{subject to} \quad PXDX^\top P^\top = I, \end{aligned} \tag{18}$$

where the matrix D is diagonal with the column sums of W, namely $D_{ii} = \sum_j W_{ij}$, and $L = D - W$ is the Laplacian matrix. Note that this constraint set is sometimes called a *flag* matrix manifold. As in Traditional CCA (Section 3.1.4), LPP can be solved to produce the matrix P with columns equal to the generalized eigenvectors v_i satisfying $XLX^\top v_i = \lambda_i XDX^\top v_i$, by implicitly solving an orthogonally constrained optimization over $M = (XDX^\top)^{1/2}P^\top \in \mathcal{O}^{d\times r}$:

$$\begin{aligned} &\text{minimize} \quad \text{tr}\left(M^\top(XDX^\top)^{-\top/2}XLX^\top(XDX^\top)^{-1/2}M\right) \\ &\text{subject to} \quad M \in \mathcal{O}^{d\times r}. \end{aligned} \tag{19}$$

Note again that the resulting linear mapping $Y = PX = M^\top(XDX^\top)^{-\top/2}X$ is not an orthogonal projection.

Related to LPP, neighborhood preserving embedding (NPE) (He et al., 2005) has a largely parallel motivation. NPE is a linear analogue to locally linear embedding (Roweis and Saul, 2000) that produces a different linear approximation to the Laplace Beltrami operator, resulting in the objective:

$$\begin{aligned} &\text{minimize} \quad \text{tr}\left(M^\top(XX^\top)^{-\top/2}X(I-W)^\top(I-W)X^\top(XX^\top)^{-1/2}M\right) \\ &\text{subject to} \quad M \in \mathcal{O}^{d\times r}, \end{aligned} \tag{20}$$

where the matrix W is the same in LPP. We group these methods together due to their similarity in motivation and resulting objective.

3.2 Linear dimensionality reduction with unconstrained objectives

All methods reviewed so far involve orthogonal mappings, but several methods simplify further to an unconstrained optimization over matrices $M \in I\!R^{d\times r}$. We describe those linear dimensionality reduction methods here.

3.2.1 Undercomplete Independent Component Analysis

Independent Component Analysis (ICA; Hyvarinen et al. (2001)) is a massively popular class of methods that is often considered alongside PCA and other simple linear transformations. ICA specifies the usual data $X \in I\!\!R^{d \times n}$ as a mixture of unknown and independent sources $Y \in I\!\!R^{r \times n}$. Note the critical difference between the independence requirement and the uncorrelatedness of PCA and other methods: for each source data point $y = [y^1, ..., y^r]^\top \in I\!\!R^r$ (one column of Y), independence implies $p(y) \approx \prod_{j=1}^r p\left(y^j\right)$, where the $p\left(y^j\right)$ are the univariate marginals of the low dimensional data (sources).

ICA finds the demixing matrix P such that we recover the independent sources as $Y = PX$. The vast majority of implementations and presentations of ICA deal with the dimension preserving case of $r = d$, and indeed most widely used algorithms require this parity. In this case, ICA is not a dimensionality reduction method.

Our case of interest for dimensionality reduction is the ‘undercomplete’ case where $r < d$, in which case $Y = PX$ is a linear dimensionality reduction method according to Definition 1. Interestingly, the most common approach to undercomplete ICA is to preprocess the mixed data X with PCA (e.g., Joho et al. (2000)), reducing the data to r dimensions, and running a standard square ICA algorithm. That said, there are a number of principled approaches to undercomplete ICA, including (Stone and Porrill, 1998; Zhang et al., 1999; Amari, 1999; De Ridder et al., 2002; Welling et al., 2004). All of these models necessarily involve a probabilistic model, required by the independence of the sources. As an implementation detail, note that observations X are whitened as a preprocessing step.

With this model, authors have maximized the log-likehood of a generative model (De Ridder et al., 2002) or minimized the mutual information between the sources (Stone and Porrill, 1998; Amari, 1999; Zhang et al., 1999), each of which requires an approximation technique. Welling et al. (2004) describes an exact algorithm for maximizing the log-likelihood of a product of experts objective:

$$f_X(M) \quad = \quad \frac{1}{n}\sum_{i=1}^{n} \log p(x_i) \quad \propto \quad \frac{1}{2}\log|M^\top M| + \frac{1}{n}\sum_{i=1}^{n}\sum_{k=1}^{r} \log p_\theta\left(m_k^\top x_i\right), \tag{21}$$

where m_k are the (unconstrained) columns of M, and $p_\theta(\cdot)$ is a likelihood distribution (an “expert”) parameterized by some θ_k. Thus this undercomplete ICA, as an optimization program like Equation 1, is a simple unconstrained maximization of $f_X(M)$ over $M \in I\!\!R^{d \times r}$.

Extensions of ICA are numerous. Insomuch as undercomplete ICA is a special case of ICA, many of these extensions will also be applicable in the undercomplete case; see the reference Hyvarinen et al. (2001).

3.2.2 Probabilistic PCA

One often-noted shortcoming of PCA is that it partitions data into orthogonal signal (the r-dimensional projected subspace) and noise (the $(d - r)$-dimensional nullspace of $M^\top$). Furthermore, PCA lacks an explicit generative model. Probabilistic PCA (PPCA; Tipping and Bishop (1999); Roweis (1997); Theobald (1975)) adds a prior to PCA to address both these potential concerns, treating the high-dimensional data to be a linear mapping of the low-dimensional data (plus noise). If we stipulate some latent independent, identically distributed r-dimensional data $y_i \sim \mathcal{N}(0, I_r)$ for $i \in \{1, ..., n\}$, and we presume that the high-dimensional data is a noisy linear mapping of that low-dimensional data $x_i|y_i \sim \mathcal{N}(My_i, \sigma_\epsilon^2 I)$ for some given or estimated noise parameter σ_ϵ^2. This model yields a natural objective with the total (negative log) data likelihood, namely:

$$f_X(M) \;=\; -\log p(X|M) \;\;\propto\;\; \log|MM^\top + \sigma_\epsilon^2 I| + \operatorname{trace}\Big((MM^\top + \sigma_\epsilon^2 I)^{-1}XX^\top\Big). \quad (22)$$

Mapping this onto our dimensionality reduction program, we want to minimize the negative log likelihood $f_X(M)$ over an arbitrary matrix $M \in \mathbb{R}^{d\times r}$. Appendix A of Tipping and Bishop (1999) shows that this objective can be minimized in closed form as $M = U_r(S_r - \sigma_\epsilon^2 I)^{\frac{1}{2}}$ where $\frac{1}{n}XX^\top = USU^\top$ is the singular value decomposition of the empirical covariance, and U_r denotes the first r columns of U (ordered by the singular values). Tipping and Bishop (1999) also show that the noise parameter σ_ϵ^2 can be solved in closed form, resulting in a closed-form maximum likelihood solution to the parameters of PPCA. This closed-form obviates a more conventional expectation-maximization (EM) approach (Dempster et al., 1977), though in practice EM is still used with the Sherman-Morrison-Woodbury matrix inversion lemma for computational advantage when $d \gg r$. Under this statistical model, the low-dimensional mapping of the observed data is the mean of the posterior $p(Y|X)$, which also corresponds to the MAP estimator: $Y = M^\top(MM^\top + \sigma_\epsilon^2 I)^{-1}X$, which again fits the form of linear dimensionality reduction $Y = PX$.

As with PCA, there are a number of noteworthy extensions to PPCA. Ulfarsson and Solo (2008) add an ℓ_2 regularization term to the PPCA objective. This regularization can be viewed as placing a Gaussian shrinkage prior $p(M)$ on the entries of M, though the authors termed this choice more as a penalty term to drive a sparse solution. A different choice of regularization is found in “Directed” PCA (Kao and Van Roy, 2013), where a trace penalty on the inverse covariance matrix is added. Finally, more generally, several of the extensions noted in Section 3.1.1 are also applicable to the probabilistic version.

3.2.3 Factor Analysis

Factor analysis (FA; Spearman (1904)) has become one of the most widely used statistical methods, in particular in psychology and behavioral sciences. FA is a more general case of a PPCA model: the observation noise is fit per observation rather than across all observations, resulting in the following conditional data likelihood: $x_i|y_i \sim \mathcal{N}(My_i, D)$ for a diagonal matrix D, where the matrix M is typically termed factor loadings. This choice can be viewed as a means to add scale invariance to each measurement, at the cost of losing rotational invariance across observations. Following the same steps as in PPCA, we arrive at the linear dimensionality reduction program:

$$\text{minimize} \quad \log|MM^\top + D| + \operatorname{trace}\Big((MM^\top + D)^{-1}XX^\top\Big) \quad (23)$$

which results in a similar linear dimensionality reduction mapping $Y = PX$ for $P = M^\top(MM^\top + D)^{-1}$. Unlike PPCA, FA has no known closed-form solution, and thus an expectation-maximization algorithm (Dempster et al., 1977) or direct gradient method is typically used to find a (local) optimum of the log likelihood. Extensions similar to those for PPCA have been developed for FA (see for example Kao and Van Roy (2013)).

3.2.4 Linear Regression

Linear regression is one of the most basic and popular tools for statistical modeling. Though not typically considered a linear dimensionality reduction method, this technique maps d-dimensional data onto an r-dimensional hyperplane defined by the number of independent variables. Considering d-dimensional data X as being partitioned into inputs and outputs $X = [X_{in}; X_{out}]$ for inputs $X_{in} \in I\!R^{r \times n}$ and outputs $X_{out} \in I\!R^{(d-r)\times n}$, linear regression fits $X_{out} \approx MX_{in}$ for some parameters $M \in I\!R^{(d-r)\times r}$. The standard choice for fitting such a model is to minimize a simple sum-of-squared-errors objective $f_X(M) = ||X_{out} - MX_{in}||_F^2$, which leads to the least squares solution $M = X_{out}X_{in}^\top(X_{in}X_{in}^\top)^{-1}$. In the form of Equation 1, linear regression is:

$$\text{minimize} \quad ||X_{out} - MX_{in}||_F^2 \tag{24}$$

This model produces a regressed data set $\hat{X} = [X_{in}; MX_{in}] = [I; M]X_{in}$. Note that $[I; M]$ has rank r (the data lie on a r-dimensional subspace) and thus Definition 1 applies. To find the dimensionality reduction mapping P, we simply take the SVD $[I; M] = USV^\top$ and set $P = [SV^\top \; 0]$ where 0 is the $(d-r)\times(d-r)$ matrix of zeroes. The low dimensional mapping of the original data X then takes the standard form $Y = PX$. Chapter 3 of Hastie et al. (2008) gives a thorough introduction to linear regression and points out (Equation 3.46) that the least squares solution can be viewed as mapping the output X_{out} in a projected basis. Adragni and Cook (2009) point out linear regression as a dimensionality reduction method in passing while considering the case of sufficient dimensionality reduction (see SDR, Section 3.1.7, for more detail).

An important extension to linear regression is regularization for bias-variance tradeoff, runtime performance, or interpretability of results. The two most popular include adding an ℓ_2 (ridge or Tikhonov regression) or an ℓ_1 penalty (lasso), resulting in the objective:

$$\text{minimize} \quad ||X_{out} - MX_{in}||_F^2 + \lambda||M||_p \tag{25}$$

for some penalty λ. While the ℓ_2 case can be solved in closed form as an augmented least squares, the ℓ_1 case requires a quadratic program (Tibshirani, 1996); though the simple quadratic program formulation scales poorly (Boyd et al., 2011; Bach et al., 2011). Regardless, both methods produce an analogous form as in standard linear regression, resulting in a linear dimensionality reduction $Y = PX$ for $P = [SV^\top \; 0]$ as above.

Another important extension, particularly given the present subject of dimensionality reduction, is principal components regression and partial least squares (Hastie et al., 2008). Principal components regression uses PCA to preprocess the input variables $X_{in} \in I\!R^{r\times n}$ down to a reduced $\tilde{X}_{in} \in I\!R^{\tilde{r}\times n}$, where $\tilde{r}$ is chosen by computational constraints, cross-validation, or similar. Standard linear regression is then run on the resulting components. This two-stage method (first PCA, then regression) can produce deeply suboptimal results, a shortcoming which to some extent is answered by partial least squares. Partial least squares is another classical method that trades off covariance of X_{in} (as in the PCA step of principal components regression) and predictive power (as in linear regression). Indeed, partial least squares has been shown to be a compromise between linear regression and principal components regression, using the framework of continuum regression (Stone and Brooks, 1990). Even still, the partial least squares objective is heuristic and is carried out on r dimensions in a greedy fashion. Bakır et al. (2004) approached the rank-r linear regression problem directly, writing the objective in the form of Equation 1 as:

$$\begin{array}{ll} \text{minimize} & ||X_{out} - M_{out} S M_{in}^\top X_{in}||_F^2 \\ \text{subject to} & M_{out} \in \mathcal{O}^{d_{out} \times r} \\ & M_{in} \in \mathcal{O}^{d_{in} \times r}, \end{array} \tag{26}$$

where S is a nonnegative diagonal matrix, and the optimization program is over the variables $\{M_{in}, M_{out}, S\}$. This method can again be solved as an example of Equation 1.

3.2.5 Distance Metric Learning

Distance metric learning (DML) is an important class of machine learning methods that is typically motivated by the desire to improve a classification method. Numerous algorithms—canonical examples include k-nearest neighbors and support vector machines—calculate distances between training points, and the performance of these algorithms can be improved substantially by a judicious choice of distance metric between these points. Many objectives have been proposed to learn these distance metrics; a seminal work is Xing et al. (2002), and thorough surveys of this literature include Kulis (2012); Yang and Jin (2006); Yang (2007).

In the linear case, to generalize beyond Euclidean distance, distance metric learning seeks a Mahalanobis distance $d_M(x_i, x_j) = ||M^\top x_i - M^\top x_j||_2 = ||x_i - x_j||_{MM^\top}$ that improves some objective on training data. When $M \in I\!R^{d \times d}$ is full rank, this approach is not a dimensionality reduction. However, as is often noted in that literature, a lower rank $M \in I\!R^{d \times r}$ for $r < d$ implies a linear mapping of the data to some reduced space where classification (or another objective) is hopefully improved, thus implicitly defining a linear dimensionality reduction method.

Numerous methods have been introduced in the DML literature. Here for clarity we survey one representative method in depth and incorporate other popular approaches from this literature thereafter. Large margin nearest neighbors (LMNN; Weinberger et al. (2005); Torresani and Lee (2006); Weinberger and Saul (2009)) assumes labeled data: (x_i, z_i), such that $z_i \in \{1, ..., C\}$ for the C data classes. LMNN typically begins by identifying a target neighbor set $\eta(i)$ for each data point x_i, which, in the absence of side information, is simply the k nearest neighbors belonging to the same class z_i as point x_i. The key intuition behind LMNN is that a distance metric $d_M(x_i, x_j)$ is desired such that target neighbors are pulled closer together than any points belonging to a different class, ideally with a large margin. Accordingly, LMNN optimizes the following objective:

$$f_X(M) = \sum_{i=1}^{n} \sum_{j \in \eta(i)} \left(d_M(x_i, x_j)^2 + \lambda \sum_{\ell=1}^{n} \mathbb{1}(z_i \neq z_\ell) \left[1 + d_M(x_i, x_j)^2 - d_M(x_i, x_\ell)^2 \right]_+ \right), \tag{27}$$

where $\mathbb{1}(\cdot)$ is the indicator function for the class labels z_i, z_ℓ, and $[\cdot]_+$ is the hinge loss. Intuitively, the first term of the right hand side pulls target neighbors closer together, while the second term penalizes (with weight λ) any points x_ℓ that are closer to x_i than its target neighbors x_j (plus some margin), and have a different label ($z_i \neq z_\ell$).

As a dimensionality reduction technique, this objective is readily optimized over $M \in \mathbb{R}^{d\times r}$, to produce a low dimensional mapping of the data $Y = M^\top X$. Beyond LMNN, other prominent methods explore slightly different objectives with similar motivations. Examples include relevant component analysis for DML (Bar-Hillel et al., 2003), neighborhood component analysis (Goldberger et al., 2004), collapsing classes (Globerson and Roweis, 2005), discriminative component analysis (Peltonen et al., 2007), latent coincidence analysis (Der and Saul, 2012), and an online, large-scale method (Chechik et al., 2009). Many of these works also offer kerneled extensions for nonlinear DML.

3.3 Scope Limitations

Definition 1 limits our scope and excludes a number of algorithms that could be considered dimensionality reduction methods. Here we consider four prominent cases that fall outside the definition of linear dimensionality reduction.

3.3.1 Nonlinear Manifold Methods

The most obvious methods to exclude from linear dimensionality reduction are nonlinear manifold methods, the most popular of which include Local Linear Embedding (Roweis and Saul, 2000), Isomap (Tenenbaum et al., 2000), Laplacian eigenmaps (Belkin and Niyogi, 2003), maximum variance unfolding (Weinberger and Saul, 2006) and diffusion maps (Coifman and Lafon, 2006). These methods seek a nonlinear manifold by using local neighborhoods, geodesic distances, or other graph theoretic considerations. Thus, while these methods are an important contribution to dimensionality reduction, they do not produce low-dimensional data as $Y = PX$ for any P. It is worth noting that some of these problems, such as Laplacian eigenmaps, do involve a generalized eigenvector problem in their derivation, though typically those eigenproblems are the direct solution to a stated objective and not the heuristic that is more often seen in the linear setting (and that motivates the use of direct optimization). A concise introduction to nonlinear manifold methods is given in Zhao et al. (2007), an extensive comparative review is Van der Maaten et al. (2009), and a probabilistic perspective on many spectral methods is given in Lawrence (2012).

3.3.2 Nonparametric Methods

One might also consider classical methods from linear systems theory, like Kalman filtering or smoothing (Kalman, 1960), as linear dimensionality reduction methods. Even more generally, nonparametric methods like Gaussian Processes (Rasmussen and Williams, 2006) also bear some similarity. The key distinction with these algorithms is that our definition of linear dimensionality is parametric: $P \in \mathbb{R}^{r\times d}$ is a fixed mapping and does not change across the dataset or some other index. Certainly any nonparametric method violates this restriction, as by definition the transformation mapping must grow with the number of data points. In the Kalman filter, for example, the mapping (which is indeed linear) between each point x_i and its low-dimensional projection y_i changes with each data point (based on all previous data), so in fact this method is also a nonparametric mapping that grows with the number of data points n. This same argument applies to most state-space models and subspace identification methods, including the linear quadratic regulator, linear quadratic

Gaussian control, and similar. Hence these other classic methods also fall outside the scope of linear dimensionality reduction.

3.3.3 Matrix Factorization Problems

A few methods discussed in this work have featured matrix factorizations, and indeed there are many other methods that involve such a decomposition in areas like indexing and collaborative filtering. This general class certainly bears similarity to dimensionality reduction, in that it uses a lower dimensional set of factors to reconstruct noisy or missing high-dimensional data (for example, classical latent semantic indexing is entirely equivalent to PCA (Deerwester et al., 1990)). A common factorization objective is to find $H \in I\!R^{d \times r}$ and $Y \in I\!R^{r \times n}$ such that the product HY reasonably approximates X according to some criteria. The critical difference between these methods and linear dimensionality reduction is that these methods do not in general yield a sensible linear mapping $Y = PX$, but rather the inverse mapping from low-dimension to high-dimension. While this may seem a trivial and invertible distinction, it is not: specifics of the method often imply that the inverse mapping is nonlinear or ill-defined. To demonstrate why this general class of problem falls outside the scope of linear dimensionality reduction, we detail two popular examples: nonnegative matrix factorization and matrix factorization as used in collaborative filtering.

Nonnegative matrix factorization (NMF; Lee and Seung (1999), sometimes called multinomial PCA (Buntine, 2002)), solves the objective $f_X(H,Y) = ||X - HY||$ for a nonnegative linear basis $H \in I\!R_{+}^{d \times r}$ and a nonnegative low-dimensional mapping $Y \in I\!R_{+}^{r \times n}$. The critical difference with our construction is that NMF is not linear: there is no P such that $Y = PX$ for all points x_i. If we are given H and a test point x_i, we must do the nonlinear solve $y_i = \text{argmin}_{y \geq 0} ||x_i - Hy||_2$. A simple counterexample is to take an existing point x_j and its nonnegative projection y_j (which we assume is not zero). If we then test on $-x_j$, certainly we can not get $-y_j$ as a valid nonnegative projection.

A second example is the broad class of matrix factorization problems as used in collaborative filtering, which includes weighted low-rank approximations (Srebro and Jaakkola, 2003), maximum margin matrix factorization (Srebro et al., 2004; Rennie and Srebro, 2005), probabilistic matrix factorization (Mnih and Salakhutdinov, 2007), and more. As above, collaborative filtering algorithms approximate data X with a low-dimensional factor model HY. However, the goal of collaborative filtering is to fill in the missing entries of X (e.g., to make movie or product recommendations), and indeed the data matrix X is usually missing the vast majority of its entries. Thus, not only is there no explicit dimensionality reduction $Y = PX$, but that operation is not even well defined for missing data.

More broadly, there has been a longstanding literature in linear algebra of low rank approximations and matrix nearness problems, often called Procrustes problems (Higham, 1989; Li and Hu, 2011; Ruhe, 1987; Schonemann, 1966). These optimization programs have the objective $f_X(M) = ||X - M||$ for some norm (often a unitarily invariant norm, most commonly the Frobenius norm) and some constrained, low-rank matrix M. PCA would be an example, considering X as the data (or the covariance) and M as the r-rank approximation thereof. While a few linear dimensionality reduction methods can be written as Procrustes problems, not all can, and thus nothing general can be claimed about the connection between Procrustes problems and the scope of this work.

Table 1: Summary of Linear Dimensionality Reduction Methods

Method	Objective $f_X(M)$	Manifold $\mathcal{M}$	Mapping $Y = PX$
PCA (§3.1.1)	$\lvert\lvert X - MM^\top X\rvert\rvert_F^2$	$\mathcal{O}^{d\times r}$	$M^\top X$
MDS (§3.1.2)	$\sum_{i,j} \left(d_X(x_i, x_j) - d_Y(M^\top x_i, M^\top x_j)\right)^2$	$\mathcal{O}^{d\times r}$	$M^\top X$
LDA (§3.1.3)	$\frac{\mathrm{tr}(M^\top \Sigma_B M)}{\mathrm{tr}(M^\top \Sigma_W M)}$	$\mathcal{O}^{d\times r}$	$M^\top X$
Traditional CCA (§3.1.4)	$\mathrm{tr}\left(M_a^\top (X_a X_a^\top)^{-1/2} X_a X_b^\top (X_b X_b^\top)^{-1/2} M_b\right)$	$\mathcal{O}^{d_a\times r} \times \mathcal{O}^{d_b\times r}$	$M_a^\top \left(X_a X_a^\top\right)^{-1/2} X_a$, $M_b^\top \left(X_b X_b^\top\right)^{-1/2} X_b$
Orthogonal CCA (§3.1.4)	$\frac{\mathrm{tr}\left(M_a^\top X_a X_b^\top M_b\right)}{\sqrt{\mathrm{tr}\left(M_a^\top X_a X_a^\top M_a\right)\mathrm{tr}\left(M_b^\top X_b X_b^\top M_b\right)}}$	$\mathcal{O}^{d_a\times r} \times \mathcal{O}^{d_b\times r}$	$M_a^\top X_a$, $M_b^\top X_b$
MAF (§3.1.5)	$\frac{\mathrm{tr}(M^\top \Sigma_\delta M)}{\mathrm{tr}(M^\top \Sigma M)}$	$\mathcal{O}^{d\times r}$	$M^\top X$
SFA (§3.1.6)	$\mathrm{tr}(M^\top \dot{X}\dot{X}^\top M)$	$\mathcal{O}^{d\times r}$	$M^\top X$
SDR (§3.1.7)	$\mathrm{tr}\left(\bar{K}_Z \left(\bar{K}_{M^\top X} + n\epsilon I\right)^{-1}\right)$	$\mathcal{O}^{d\times r}$	$M^\top X$
LPP (§3.1.8)	$\mathrm{tr}\left(M^\top (XDX^\top)^{-\top/2} XLX^\top (XDX^\top)^{-1/2} M\right)$	$\mathcal{O}^{d\times r}$	$M^\top (XDX^\top)^{-\top/2} X$
UICA (§3.2.1)	$\frac{1}{2}\log\lvert M^\top M\rvert + \frac{1}{n}\sum_{i=1}^n \sum_{k=1}^r \log f_\theta\left(m_k^\top x_n\right)$	$\mathbb{R}^{d\times r}$	$M^\top X$
PPCA (§3.2.2)	$\log\lvert MM^\top + \sigma^2 I\rvert + \mathrm{tr}\left(XX^\top (MM^\top + \sigma^2 I)^{-1}\right)$	$\mathbb{R}^{d\times r}$	$M^\top (MM^\top + \sigma^2 I)^{-1} X$
FA (§3.2.3)	$\log\lvert MM^\top + D\rvert + \mathrm{tr}\left(XX^\top (MM^\top + D)^{-1}\right)$	$\mathbb{R}^{d\times r}$	$M^\top (MM^\top + D)^{-1} X$
LR (§3.2.4)	$\lvert\lvert X_{out} - MX_{in}\rvert\rvert_F^2 + \lambda\lvert\lvert M\rvert\rvert_p$	$\mathbb{R}^{d\times r}$	$SV^\top X_{in}$ for $M = USV^\top$
DML (§3.2.5)	$\sum_{i,j\in\eta(i)} \Big\{ d_M(x_i, x_j)^2 + \lambda \sum_\ell \mathbb{1}(z_i \neq z_\ell)$ $\left[1 + d_M(x_i, x_j)^2 - d_M(x_i, x_\ell)^2\right]_+ \Big\}$	$\mathbb{R}^{d\times r}$	$M^\top X$

3.4 Summary of the framework

Table 1 offers a consolidated summary of these methods. Considering linear dimensionality reduction through the lens of a constrained matrix optimizaton enables a few key insights. First, as is the primary purpose of this paper, this framework surveys and consolidates the space of linear dimensionality reduction methods. It clarifies that linear dimensionality reduction goes well beyond PCA and can require much more than simply eigenvalue decompositions, and also that many of these methods bear significant resemblance to each other in spirit and in detail. Second, this consolidated view suggests that, since optimization programs over well-understood matrix manifolds address a significant subclass of these methods, an objective-agnostic solver over matrix manifolds may provide a useful generic solver for linear dimensionality reduction techniques.

4. Results

All methods considered here have specified $\mathcal{M}$ as either unconstrained matrices or matrices with orthonormal columns, variables in the space $I\!R^{d \times r}$. In the unconstrained case, numerous standard optimizers can and have been brought to bear to optimize the objective $f_X(M)$. In the orthogonal case, we have also claimed that the very well-understood geometry of the manifold of orthogonal matrices enables optimization over these manifolds. Pursuing such approaches is critical to consolidating and extending dimensionality reduction, as orthogonal projections $Y = M^\top X$ for $M \in \mathcal{O}^{d \times r}$ are arguably the most natural formulation of linear dimensionality reduction: one seeks a low-dimensional view of the data where some feature is optimally preserved.

The matrix family $\mathcal{O}^{d \times r}$ is precisely the real Stiefel manifold, which is a compact, embedded submanifold of $I\!R^{d \times r}$. In our context, this means that many important intuitions of optimization can be carried over onto the Stiefel manifold. Notably, with a differentiable objective function $f_X(M)$ and its gradient $\nabla_M f$, one can carry out standard first order optimization via a projected gradient method, where the unconstrained gradient is mapped onto the Stiefel manifold for gradient steps and linesearches. Second order techniques also exist, with some added complexity. The foundations of these techniques are Luenberger (1972); Gabay (1982), both of which build on classic and straightforward results from differential geometry. More recently, Edelman et al. (1998) sparked significant interest in optimization over matrix manifolds. Some relevant examples include Manton (2002, 2004); Fiori (2005); Nishimori and Akaho (2005); Abrudan et al. (2008); Ulfarsson and Solo (2008); Srivastava and Liu (2005); Rubinshtein and Srivastava (2010); Varshney and Willsky (2011). Indeed, some of these works have been in the machine learning community (Fiori, 2005; Ulfarsson and Solo, 2008; Varshney and Willsky, 2011), and some have made the connection of geometric optimization methods to PCA (Srivastava and Liu, 2005; Ulfarsson and Solo, 2008; Rubinshtein and Srivastava, 2010; Varshney and Willsky, 2011). The basic geometry of this manifold, as well as optimization over Riemannian manifolds, has been often presented and is now fairly standard. For completeness, we include a primer on this topic in Appendix A. There, as a motivating example, we derive the tangent space, the projection operation, and a retraction operation for the Stiefel manifold. Appendix A then includes Algorithm 1, which uses these objects to present an optimization routine that performs gradient descent over the Stiefel manifold. For a thorough treatment, we refer the interested reader to the excellent summary of much of this modern work (Absil et al., 2008).

One important technical note warrants mention here. The Stiefel manifold is the manifold of all ordered r-tuples of orthonormal vectors in $I\!R^d$, but in some cases the dimensionality reduction objective $f_X(\cdot)$ evaluates only the subspace (orthonormal basis) implied by M, not the particular choice and order of the orthonormal vectors in M. This class of objective functions is precisely those functions $f_X(M)$ such that, for any $r \times r$ orthogonal matrix R, $f_X(M) = f(MR)$. The implied constraint in these cases is the manifold of rank-r subspaces in $I\!R^d$, which corresponds to the real Grassmann manifold $\mathcal{G}^{d \times r}$ (another very well understood manifold). As a clarifying example, note that the PCA objective is redundant on the Stiefel manifold: if we want the highest variance r-dimensional projection of our data, the parameterization of those r dimensions is arbitrary, and indeed $f(M) = ||X - MM^\top X||_F^2 = f(MR)$ for any orthogonal R. If one is particularly interested in ranked eigenvectors, there are standard numerical tricks to break this equivalence and produce an ordered result: for example, maximizing $\mathrm{tr}(AM^\top XX^\top M)$ over the Stiefel manifold, where A is any diagonal matrix with ordered elements ($A_{11} > ... > A_{rr}$). From the perspective of optimization and linear dimensionality reduction, the difference between the Grassmann and Stiefel manifold is one of identifiability. Since there is an uncountable set of Stiefel points corresponding to a single Grassmann point, it seems sensible for many reasons to optimize over the Grassmann manifold when possible (though, as our results will show, this distinction empirically mattered very little). Indeed, most of the optimization literature noted above also deals with the Grassmann case, and the techniques are similar. Conveniently, an objective $f_X(M)$ can be quickly tested for the true implied manifold by comparing values of $f_X(MR)$ for various R. Because the end result is still a matrix $M \in \mathcal{O}^{d\times r}$ (which happens to be in a canonical form in the Grassmann case), this fact truly is an implementation detail of the algorithm, not a fundamental distinction between different linear dimensionality reduction methods. Thus, we present our results as agnostic to this choice, and we empirically revisit the question of identifiability at the end of this section.

To demonstrate the effectiveness of these optimization techniques, we implemented a variety of linear dimensionality reduction methods with several solvers: first order steepest descent methods over the Stiefel and Grassmann manifolds, and second order trust region methods over the Stiefel and Grassmann manifolds (Absil et al., 2008). We implemented these methods in MATLAB, both natively for first order methods, and using the excellent `manopt` software library (Boumal et al., 2014) for first and second order methods. All of these solvers accept, as input, data X and any function that evaluates a differential objective $f_X(M)$ and its gradient $\nabla_M f$ at any point $M \in \mathcal{O}^{d\times r}$, and return, as output, an orthogonal M that corresponds to a (local) optimum of the objective $f_X(M)$.

4.1 Example of eigenvector suboptimality

We have cautioned throughout the above survey about the suboptimality of heuristic eigenvector solutions. Figure 1 demonstrates this suboptimality for LDA (Section 3.1.3). In each panel (A and B), we simulated data of dimensionality $d = 3$, with $n = 3000$ points, corresponding to 1000 points in each of 3 clusters (shown in black, blue, and red). Data in each cluster were normally distributed with random means (normal with standard deviation 5/2) and random covariance (uniformly distributed orientation and exponentially distributed eccentricity with mean 5). In the left subpanel of panel A, we then calculated the $r = 2$ dimensional projection by orthogonalizing the top two eigenvectors of the matrix $\Sigma_W^{-1}\Sigma_B$ ('Heuristic LDA'). In the right subpanel, we directly optimized the objective of Equation 6 over $\mathcal{O}^{d\times r}$ ('Orthogonal LDA'). We calculate the normalized improvement of the manifold method as:

$$-\frac{\left(f_X\left(M^{(orth)}\right) - f_X\left(M^{(eig)}\right)\right)}{\left|f_X\left(M^{(eig)}\right)\right|}. \tag{28}$$

Throughout the results we will call the results of traditional eigenvector approaches $M^{(eig)}$ and the results of our manifold solver $M^{(orth)}$. Figure 1A shows an example where both the heuristic and manifold optimization methods return qualitatively similar results, and indeed the numerical improvement (0.02) reflects that indeed this heuristic is by no means wildly

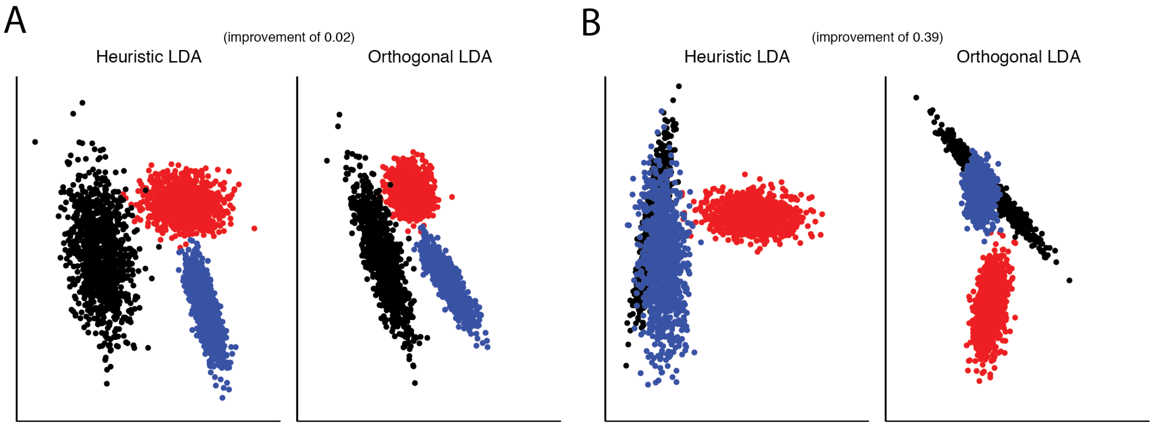

Figure 1: Cautionary example of differences in objectives for LDA. Panel A shows a dataset that offers only marginal performance gain by using manifold optimization (Orthogonal LDA, right subpanel of panel A) rather than the traditional eigenvector heuristic (Heuristic LDA, left subpanel). Panel B shows a dataset that has a stark difference between the two methods. The measured performance difference (see Equation 28) is shown.

inappropriate for the stated objective. Indeed, we know it to be correct for $r = 1$. Figure 1B shows a particularly telling example: both methods distinguish the red cluster easily, whereas the heuristic method confounds the black and blue clusters, while the optimization approach offers better separability, which indeed correlates with improvement on the stated objective of Equation 6. It is critical to clarify the distinction between these two methods: the heuristic and orthogonal solutions are indeed optimal, but for *different* objectives, as discussion in Section 3.1.3. Thus, the purpose of this cautionary example is to highlight the importance of optimizing the intended objective, and the freedom to choose that objective without a tacit connection to a generalized eigenvalue problem. These goals can be directly and generically achieved with the optimization framework of Equation 1.

4.2 Performance improvement

Here we seek to demonstrate the quantitative improvements available by directly optimizing an objective, rather than resorting to an eigenvector heuristic. First we implemented PCA (Section 3.1.1) using both methods. We ran PCA on 20 random data sets for each dimensionality $d \in \{4, 8, 16, ..., 1024\}$, each time projecting onto $r = 3$ dimensions. Data were normally distributed with random covariance (exponentially distributed eccentricity with mean 2). We calculated $f_X\left(M^{(eig)}\right)$ and $f_X\left(M^{(orth)}\right)$ from Equation 2, and we calculated the normalized improvement of the manifold method as above in Equation 28. Since the eigenvector decomposition is provably optimal for PCA, our method should demonstrate no improvement. Indeed, Figure 2 (purple trace) shows the distribution of normalized improvements for PCA is entirely 0 in panel A. We then repeated this analysis for a fixed data dimensionality $d = 100$ (generating data as above), now ranging the projected dimensionality $r \in \{1, 2, 5, 10, 20, 40, 80\}$. These results are shown in Figure 2B, and again, the optimization approach recovers the known PCA optima precisely. This confirmatory result

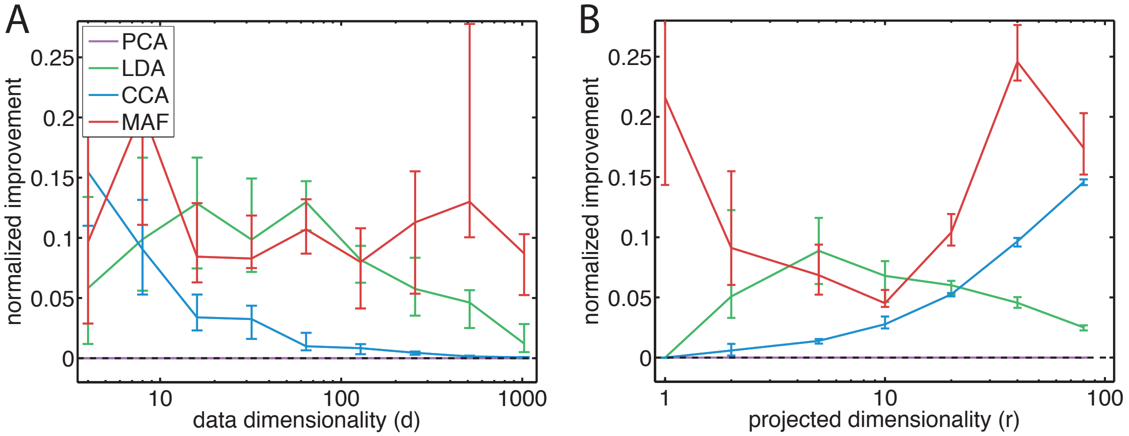

Figure 2: Performance comparison between heuristic solvers and direct optimization of linear dimensionality reduction objectives. The vertical axis denotes normalized improvement of the optimization program over traditional approaches. Panel A shows dimensionality reduction over a range of data dimensionalities $d \in \{4, 8, 16, ..., 1024\}$, with $r = 3$ projected dimensions. Panel B shows dimensionality reduction over a range of projected dimensionalities $r \in \{1, 2, 5, 10, 20, 40, 80\}$, with $d = 100$ data dimensions. Each method was run 20 times independently for each choice of (d, r), and the error bars represent median performance improvement and the central 50th percentile of the distribution.

also shows, pleasingly, that there is no empirical downside (in terms of accuracy) to using manifold optimization.

We next repeated the same experiment for LDA (Section 3.1.3). We generated data with 1000 data points in each of d classes, where within class data was generated according to a normal distribution with random covariance (uniformly distributed orientation and exponentially distributed eccentricity with mean 5), and each class mean vector was randomly chosen (normal with standard deviation $5/d$). We compared the suboptimal LDA heuristic $M^{(eig)}$ (orthogonalizing the top r eigenvectors of $\Sigma_W^{-1}\Sigma_B$) to the direct optimization of $f_X(M) = \text{tr}(M^\top \Sigma_B M)/\text{tr}(M^\top \Sigma_W M)$, which produced $M^{(orth)}$. Unlike in PCA, Figure 2 (green traces) shows that directly addressing the LDA objective produces significant performance improvements. The green trace is plotted at the median, and the error bars show the median 50% of the distribution of performance improvements across both data dimensionality d (panel A) and projected dimensionality r (panel B).

We next implemented Traditional CCA and Orthogonal CCA as introduced in Section 3.1.4, which yield the blue performance distributions shown in Figure 2A and B. Dataset X_a was generated by a random linear transformation of a latent dataset Z (iid standard normal points with dimensionality of $d/2$; the random linear transformation had the same distribution), plus noise, and dataset X_b was generated by a different random linear transformation of the same latent Z, plus noise. Again we see significant improvement of direct Orthogonal CCA over orthgonalizing Traditional CCA, when evaluated under the correlation objective of Equation 10. First, we note that to be conservative in this case we omit the denominator term from the improvement metric (Equation 28); that is, we do not normalize CCA improvements. CCA has a correlation objective, which is already a normalized quantity, and thus renormalizing would increase these improvements. More importantly, it is essential to note that we do not claim any suboptimality of Hotelling's Traditional CCA in solving Equation 8. Rather, it is the subsequent heuristic choice of orthogonalizing the resulting mapping that is problematic. In other words, we show that if one seeks an orthogonal projection of the data, as is often desired in practice, one should do so directly. Our CCA results demonstrate the substantial underperformance of eigenvector heuristics in this case, and our generic solver allows a direct solution without conceptual difficulty.

Finally, we implemented MAF as introduced in Section 3.1.5, where we generated data by a random linear transformation (uniformly distributed entries on $[0, d^{-1/2}]$) of d dimensions of univariate random temporal functions, which we generated with cubic splines with four randomly located knots (uniformly distributed in the domain, standard normally distributed in range), plus noise. MAF is another method that has been solved using an eigenvector heuristic, and the performance improvement is shown in red in Figure 2.

In total, Figure 2 offers some key points of interpretation. First, note that no data lie in the negative halfplane (see black dashed line atop the purple line at 0). Though unsurprising, this is an important confirmation that the optimization program performs unambiguously better than or equal to heuristic methods. Second, methods other than PCA produce approximately 10% improvement using direct optimization, a significant improvement that suggests the broad use of this optimization framework. Third, a natural question for these nonconvex programs is that of local optima. We found that, across a wide range of choices for d and r, nearly all methods converged to the same optimal value whether started at a random M or started at the heuristic point $M^{(eig)}$. Deeper characterization of local optima should be highly dependent on the particular objective and is beyond the scope of this work. Third, we note that methods sometimes have performance equal to the heuristic method; indeed $M^{(eig)}$ is sometimes a local optimum. We found empirically that larger r makes this less likely, and larger d makes this more likely.

A significant point of interpretation is that of size of average performance. We stress that these data sets were not carefully chosen to demonstrate effect. Indeed, we are able to adversarially choose data to create much larger performance improvements, and similarly we can choose datasets that demonstrate no effect. Thus, one should not infer from Figure 2 that, for example, Orthogonal CCA fundamentally has increasing benefit over the heuristic approach with increasing r (or decreasing benefit with increasing d). Instead, we encourage the takeaway of this performance figure to be that one should always optimize the objective of interest directly, rather than resorting to a reasonable but theoretically unsound eigenvector heuristic, as the performance loss is potentially meaningful.

4.3 Computational cost

Importantly, this matrix manifold solver does not incur massive computational cost. The only additional computation beyond standard unconstrained first-order optimization of dr variables is the projection onto or along the manifold to ensure a feasible $M \in \mathcal{O}^{d \times r}$, which in any scheme requires a matrix decomposition (see Appendix A). Thus each algorithmic step carries an additional cost of $O(dr^2)$. This cost is in many cases dwarfed by the larger cost of calculating matrix-matrix products with a data matrix $X \in \mathbb{R}^{d \times n}$ (which often appear in the gradient calculations $\nabla_M f$). Second order methods approximate or evalu-

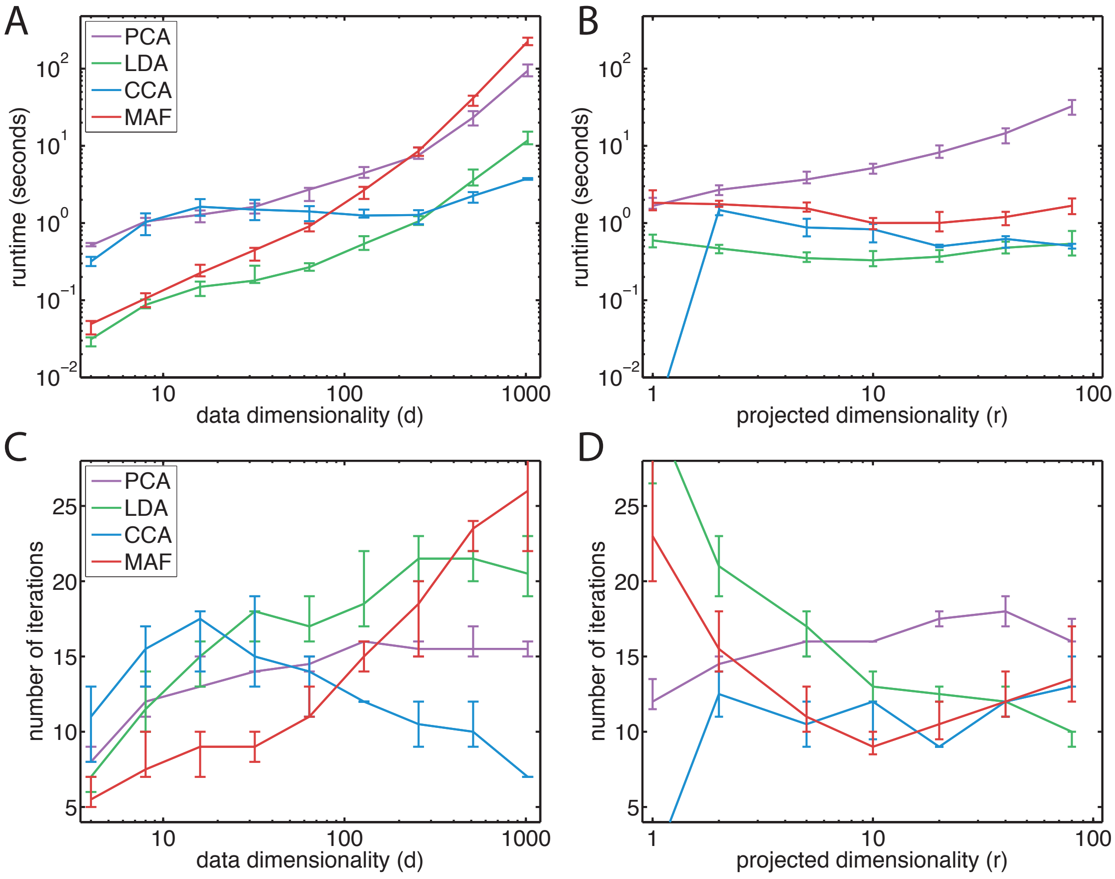

Figure 3: Computational cost of direct optimization of linear dimensionality reduction objectives. Datasets are the same as those in Figure 2. The vertical axis in panels A and B denotes runtime in seconds. Panels C and D show the same data by the number of solver iterations.

ate a Hessian, which incurs more complexity per iteration, but as usual at the tradeoff of drastically fewer iterations. Accordingly, the runtime of manifold optimization is at worst moderately degraded compared to an unconstrained first or second order method. Compared to eigenvector heuristics, which if implemented as a compact SVD cost only $O(dr^2)$, direct optimization is an order of magnitude or more slower due to the iterative nature of the algorithm.

Figure 3 shows the computational cost of these methods, using the same data as in the previous section. In Figure 3A, at each of $d \in \{4, 8, 16, ..., 1024\}$ and for $r = 3$, we ran PCA, LDA, CCA, and MAF 20 times, and we show here the median and central 50% of the runtime distribution (in seconds). This panel demonstrates that runtime increases approximately linearly as expected in d: runtime increases by approximately three orders of magnitude over three orders of magnitude increase in d. We do a similar simulation in Figure 3B at each of $r \in \{1, 2, 5, 10, 20, 40, 80\}$ for a fixed $d = 100$, and again runtime is increasing.

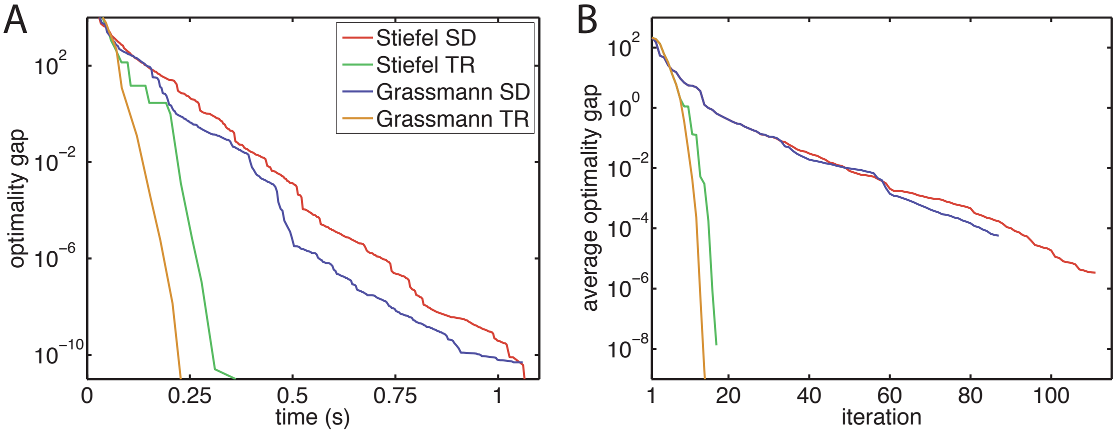

Figure 4: Comparison of different optimization techniques. PCA was run on 100 independent datasets of size $d = 100$, projecting to $r = 10$ dimensions. Panel A shows the median runtime performance (across datasets), with optimality gap as a function of runtime in seconds. Panel B shows the average optimality gap by iteration. PCA was run on each of these datasets independently with the Stiefel steepest descent (red), Stiefel trust region (green), Grassmann steepest descent (blue), and Grassmann trust region (brown) solvers.

Figures 3C and 3D show the same data as in Figures 3A and 3B, but by number of solver iterations. In this figure we used a second-order solver over the Grassmann manifold in PCA, LDA, and MAF (critically, the same solver for all three), and the second-order solver over the product of two Stiefel manifolds in the case of CCA. These two panels again underscore the overall point of Figure 3: runtime complexity is not particularly burdensome across a range of reasonable choices for d and r, even with a generic solver.

4.4 Choice of solver, and identifiability

We have claimed that the choice of optimization over the Stiefel or Grassmann manifold is a question of identifiability, and further that empirically it seems to matter little to algorithmic performance. Figure 4 gives evidence to that claim. We created 100 independent datasets with $d = 100$ and $r = 10$ for PCA. Here the choice of algorithm is less important, and PCA is a sensible choice because we know the global optimum. We ran PCA using four solvers: first-order steepest descent over the Stiefel manifold, first-order steepest descent over the Grassmann manifold, second-order trust region optimization over the Stiefel manifold, and second-order trust region optimization over the Grassmann manifold. Figure 4 shows the optimality gap by solver choice for each of these four solvers. Figure 4A shows the optimality gap as a function of time for the median performing solver (median across the 100 independent datasets), and Figure 4B shows the optimality gap (mean across all the 100 independent datasets) as a function of algorithmic iteration. From these figures it is clear that second-order methods outperform first order methods, though perhaps less than one might typically expect. More importantly, the difference between the choice of optimization over the Stiefel or Grassmann manifold is minor at best. This figure, along with previous results, suggest the feasibility of a generic solver for orthogonal linear dimensionality reduction.

5. Discussion

Dimensionality reduction is a cornerstone of data analysis. Among many methods, perhaps none are more often used than the linear class of methods. By considering these methods as optimization programs of user-specified objectives over orthogonal or unconstrained matrix manifolds, we have surveyed a surprisingly fragmented literature, offered insights into the shortcomings of traditional eigenvector heuristics, and have pointed to straightforward generalizations with an objective-agnostic linear dimensionality reduction solver. The results of Section 4 suggest that linear dimensionality reduction can be abstracted away in the same way that unconstrained optimization has been, as a numerical technology that can sometimes be treated as a black-box solver. This survey also suggests that future linear dimensionality reduction algorithms can be derived in a simpler and more principled fashion. Of course, even with such a method one must be careful to design a linear dimensionality reduction sensibly to avoid the many unintuitive pitfalls of high-dimensional data (e.g., Diaconis and Freedman (1984)).

Other authors have surveyed dimensionality reduction algorithms. Some relevant examples include Burges (2010); De la Torre (2012); Sun et al. (2009); Borga et al. (1997). These works all focus on particular subsets of the dimensionality reduction field, and our work here is no different, insomuch as we focus exclusively on linear dimensionality reduction and the connecting concept of optimization over matrix manifolds. Burges (2010) gives an excellent tutorial review of popular methods, including both linear and nonlinear methods, dividing those methods into projective and manifold approaches. De la Torre (2012) surveys five linear and nonlinear methods with their kernelized counterparts using methods from kernel regression. Borga et al. (1997) and Sun et al. (2009) focus on those methods that can be cast as generalized eigenvalue problems, and derive scalable algorithms for those methods, connecting to the broad literature on optimizing Rayleigh quotients.

The simple optimization framework discussed herein offers a direct approach to linear dimensionality reduction: many linear dimensionality reduction methods seek a meaningful, low-dimensional orthogonal subspace of the data, so it is natural to create a program that directly optimizes some objective on the data over these subspaces. This claim is supported by the number of linear dimensionality reduction methods that fit naturally into this framework, by the ease with which new methods can be created, and by the significant performance gains achieved with direct optimization. Thus we believe this survey offers a valuable simplifying principle for linear dimensionality reduction.

This optimization framework is conceptually most similar to the projection index from important literature in projection pursuit (Huber, 1985; Friedman, 1987): both that literature and the present work focus on optimizing objective functions on projections to a lower dimensional coordinate space. Since the time of the fundamental work in projection pursuit, massive developments in computational power and advances in optimization over matrix manifolds suggest the merit of the present approach. First, the projection pursuit literature is inherently greedy: univariate projections are optimized over the projection index, that structure is removed from the high dimensional data, and the process is repeated. This approach leads to (potentially significant) suboptimality of the results and requires costly computation on the space of the high-dimensional data for structure removal. The present matrix manifold framework circumvents both of these issues. Thus, while the spirit of this framework is very much in line with the idea of a projection index, this framework, both in concept and in implementation, is critically enabled by tools that were unavailable to the original development of projection pursuit.

Acknowledgments

JPC and ZG received funding from the UK Engineering and Physical Sciences Research Council (EPSRC EP/H019472/1). JPC received funding from a Sloan Research Fellowship, the Simons Foundation (SCGB#325171 and SCGB#325233), the Grossman Center at Columbia University, and the Gatsby Charitable Trust.

Appendix A. Optimization over the Stiefel maifold

Here we offer a basic introduction to optimization over matrix manifolds, restricting our focus to a first-order, projected gradient optimization over the Stiefel manifold $\mathcal{O}^{d\times r}$. Intuitively, manifold projected gradient methods are iterative optimization routines that require firstly an understanding of search directions along the constraint set, called the tangent space (§A.1). With an objective f, gradients $\nabla_M f$ are then calculated in the full space, in this case $I\!R^{d\times r}$. These gradients are projected onto that tangent space (§A.2). Any nonzero step in a linear tangent space will depart from the nonlinear constraint set, so finally a *retraction* is needed to map a step onto the constraint set (§A.3). With these three components, a standard first-order iterative solver can be carried out, with typical convergence guarantees. We conclude this tutorial appendix with pseudocode in §A.4 and a figure summarizing these steps (Figure 5).

We have previously introduced the Stiefel manifold $\mathcal{O}^{d\times r}$ as the set of all matrices with orthonormal columns, namely: $\mathcal{O}^{d\times r} = \left\{M \in I\!R^{d\times r} : M^\top M = I\right\}$, where I is the $r \times r$ identity matrix. $\mathcal{O}^{d\times r}$ is a manifold, an embedded submanifold of $I\!R^{d\times r}$, and bounded and closed (and thus compact). From these facts we can carry over all intuitions of an explicit (though nonlinear and nonconvex) constraint set within $I\!R^{d\times r}$.

A.1 Tangent space $T_M\mathcal{O}^{d\times r}$

Critical to understanding the geometry of any manifold (in particular to exploit that geometry for optimization) is the *tangent space*, the linear (vector space) approximation to the manifold at a particular point. To define this space, we first define a *curve* on the manifold $\mathcal{O}^{d\times r}$ as a smooth map $\gamma(\cdot) : I\!R \rightarrow \mathcal{O}^{d\times r}$. Then, the tangent space is:

$$T_M\mathcal{O}^{d\times r} = \left\{\dot{\gamma}(0) : \gamma(\cdot) \text{ is a curve on } \mathcal{O}^{d\times r} \text{ with } \gamma(0) = M\right\}, \tag{29}$$

where $\dot{\gamma}$ is the derivative $\frac{d}{dt}\gamma(t)$. Loosely, $T_M\mathcal{O}^{d\times r}$ is the space of directions along the manifold at a point M. While Equation 29 is fairly general for embedded submanifolds, it is abstract and leaves little insight into numerical implementation. Conveniently, the tangent space of the Stiefel manifold has a particularly nice equivalent form.

Claim 1 (Tangent space of the Stiefel Manifold) *The following sets are equivalent:*

$$\begin{aligned} T_M\mathcal{O}^{d\times r} &= \Big\{\dot\gamma(0) : \gamma(\cdot) \text{ is a curve on } \mathcal{O}^{d\times r} \text{ with } \gamma(0) = M\Big\}, && (30)\\ T_1 &= \Big\{X \in \mathbb{R}^{d\times r} : M^\top X + X^\top M = 0\Big\}, && (31)\\ T_2 &= \Big\{MA + (I - MM^\top)B : A = -A^\top, B \in \mathbb{R}^{d\times r}\Big\}. && (32) \end{aligned}$$

Proof The proof proceeds in four steps:

1. $X \in T_M\mathcal{O}^{d\times r} \Rightarrow X \in T_1$

   Considering a curve $\gamma(t)$ from Equation 30, we know $\gamma(t)^\top\gamma(t) = I$ (every point of the curve is on the manifold). We differentiate in t to see $\gamma(t)^\top\dot\gamma(t) + \dot\gamma(t)^\top\gamma(t) = 0$. At $t = 0$, we have $\gamma(0) = M$, and we define the tangent space element $\dot\gamma(0) = X$. Then X is such that $M^\top X + X^\top M = 0$.

2. $X \in T_1 \Rightarrow X \in T_M\mathcal{O}^{d\times r}$

   We must construct a curve such that any $X \in T_1$ is a point in the tangent space; consider $\gamma(t) = (M + tX)(I + t^2X^\top X)^{-1/2}$ (a choice that we will see again below in §A.3). First, this curve satisfies $\gamma(0) = M$. Second, $\gamma(\cdot)$ is a curve on the Stiefel manifold, since every point $\gamma(t)$ satisfies:

   $$\begin{aligned} \gamma(t)^\top\gamma(t) &= (I + t^2X^\top X)^{-1/2}(M + tX)^\top(M + tX)(I + t^2X^\top X)^{-1/2}\\ &= (I + t^2X^\top X)^{-1/2}(M^\top M + tM^\top X + tX^\top M + t^2X^\top X)(I + t^2X^\top X)^{-1/2}\\ &= (I + t^2X^\top X)^{-1/2}(I + t^2X^\top X)(I + t^2X^\top X)^{-1/2}\\ &= I, \end{aligned}$$

   where the third line uses $M \in \mathcal{O}^{d\times r}$ and $X \in T_1$. It remains to show only that $\dot\gamma(0) = X$. We differentiate $\gamma(t)$ as

   $$\dot\gamma(t) = X(I + t^2X^\top X)^{-1/2} + (M + tX)\frac{d}{dt}(I + t^2X^\top X)^{-1/2}. \qquad (33)$$

   The rightmost derivative term of Equation 33 does not have a closed form, but is the unique solution to a Sylvester equation. Letting $\alpha(t) = (I + t^2X^\top X)^{-1/2}$, we seek $\dot\alpha(0)$. By implicit differentiation:

   $$\begin{aligned} \left[\frac{d}{dt}\alpha(t)\alpha(t)\right]_{t=0} &= \left[\frac{d}{dt}(I + t^2X^\top X)^{-1}\right]_{t=0}\\ \dot\alpha(0)\alpha(0) + \alpha(0)\dot\alpha(0) &= \Big[(I + t^2X^\top X)^{-1}\Big(2tX^\top X\Big)(I + t^2X^\top X)^{-1}\Big]_{t=0}\\ 2\dot\alpha(0) &= 0, \end{aligned}$$

   since $\alpha(0) = I$. Thus we see $\dot\alpha(0) = \left[\frac{d}{dt}(I + t^2X^\top X)^{-1/2}\right]_{t=0} = 0$. Equation 33 yields $\dot\gamma(0) = X$, which completes the proof of the converse.

3. $X \in T_2 \Rightarrow X \in T_1$

   Let $X = MA + (I - MM^\top)B$ according to Equation 32. Then:

$$\begin{aligned} M^\top X + X^\top M &= M^\top MA + M^\top(I - MM^\top)B + A^\top M^\top M + B^\top(I - MM^\top)M \\ &= A + A^\top \\ &= 0, \end{aligned}$$

   by the skew-symmetry of A and $M \in \mathcal{O}^{d\times r}$.

4. $X \in T_1 \Rightarrow X \in T_2$

   We show the transposition $X \notin T_2 \Rightarrow X \notin T_1$. By the definition of T_2, $X = MA + (I - MM^\top)B$ is not in T_2 if and only if $A \neq -A^\top$. Then, using the previous argument, we see that such an X has $M^\top X + X^\top M \neq 0$, and thus is not a member of T_1.

Thus, the three tangent space definitions Equations 30-32 are equivalent. The definition of Equation 32 is particularly useful as it is constructive, which is essential when considering optimization. ■

A.2 Projection $\pi_M : \mathbb{R}^{d\times r} \to T_M\mathcal{O}^{d\times r}$

Because $\mathcal{O}^{d\times r}$ is an embedded submanifold of $\mathbb{R}^{d\times r}$, it is natural to consider the metric implied by Euclidean space : $\mathbb{R}^{d\times r}$ endowed with the standard inner product $\langle P, N\rangle = \mathrm{tr}(P^\top N)$, and the induced Frobenius norm $||\cdot||_F$. With this metric, the Stiefel manifold is then a Riemannian submanifold of Euclidean space. This immediately allows us to consider the projection of an arbitrary vector $Z \in \mathbb{R}^{d\times r}$ onto the tangent space $T_M\mathcal{O}^{d\times r}$:

$$\begin{aligned} \pi(Z) &= \operatorname*{arg\,min}_{X\in T_M\mathcal{O}^{d\times r}} ||Z - X||_F \\ &= \arg\min ||Z - (MA - (I - MM^\top)B)||_F \\ &= \arg\min ||(MM^\top Z - MA) + (I - MM^\top)(Z - B)||_F \\ &= \arg\min ||M(M^\top Z - A)||_F + ||(I - MM^\top)(Z - B)||_F \\ &= \arg\min ||M^\top Z - A||_F + ||(I - MM^\top)(Z - B)||_F \end{aligned}$$

where the last equality comes from the unitary invariance of the Frobenius norm. This expression is minimized by setting $B = Z$ and setting A to be the skew-symmetric part of $M^\top Z$, namely $A := \mathrm{skew}(M^\top Z) = \frac{1}{2}(M^\top Z - Z^\top M)$ (Fan and Hoffman, 1955). This results in the projection:

$$\pi_M(Z) = M\mathrm{skew}(M^\top Z) + (I - MM^\top)Z. \tag{34}$$

We note that an alternative canonical metric is often considered in this literature, namely $\langle P, N\rangle_M = \mathrm{tr}\left(P^\top(I - MM^\top)N\right)$. The literature is divided on this choice; for simplicity we choose the standard inner product.

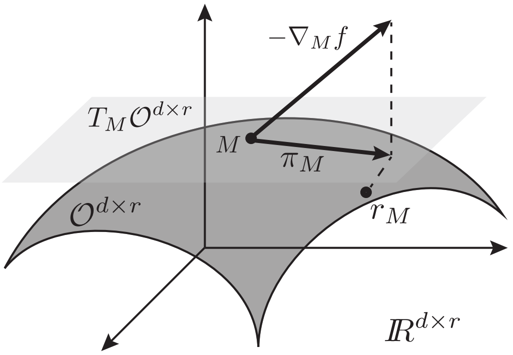

Figure 5: Cartoon of a projected gradient step on the Stiefel manifold. Notation follows Algorithm 1.

A.3 Retraction $r_M : T_M\mathcal{O}^{d\times r} \rightarrow \mathcal{O}^{d\times r}$

Projected gradient methods seek an iterative step in the direction of steepest descent along the manifold, namely $M + \beta\pi_M\left(-\nabla_M f\right)$. For any nonzero step size β, this iterate will leave the Stiefel manifold. Thus, a *retraction* is required to map onto the manifold. A number of projective retractions are available (Kaneko et al., 2013); here we define the retraction of a step Z away from a current manifold point M as

$$r_M(Z) = \underset{N\in\mathcal{O}^{d\times r}}{\arg\min} ||N - (M + Z)||_F, \tag{35}$$

that is, the closest point on the manifold to the desired iterate $M + Z$. For unitarily invariant norms, a classic result is that $r_M(Z) = UV^\top$, where $(M + Z) = USV^\top$ is the singular value decomposition (Fan and Hoffman, 1955), or equivalently, $r_M(Z) = W$ for a polar decomposition $(M + Z) = WP$. Conveniently, when $Z \in T_M\mathcal{O}^{d\times r}$, this retraction has the simple closed form $r_M(Z) = (M + Z)(I + Z^\top Z)^{-1/2}$ (Kaneko et al., 2013), which explains the choice of curve in §A.1.

In the cases of the Stiefel and Grassmann manifolds, it is possible to directly calculate a manifold geodesic (shortest path between two points in the manifold). While more aesthetically pleasing, calculating such a geodesic requires a matrix exponential, and thus has similar computational burden as a projective retraction (often the exponential is slightly more expensive). Empirically, we have found very little difference in the convergence or computational burden of this choice, and thus we focus this tutorial on the conceptually simpler retraction. Absil and Malick (2012) discuss projective retractions compared with geodesics/exponential maps.

A.4 Psuedocode for a projected gradient solver

Algorithm 1 gives pseudocode for a projected gradient method over the Stiefel manifold. This generic algorithm requires only a choice of convergence parameters and a linesearch method, choices which are standard for first-order optimization. Chapter 4 of Absil et al. (2008) offers a global convergence proof for such a method using Armijo linesearch. Indeed, the only particular consideration for this algorithmic implementation is the tangent space $T_M\mathcal{O}^{d\times r}$, the projection π_M, and the retraction r_M.

Algorithm 1 Gradient descent over the Stiefel manifold (with linesearch and retraction)

```
initialize $M \in \mathcal{O}^{d \times r}$
while $f(M)$ has not converged do
    calculate $\nabla_M f \in \mathbb{R}^{d \times r}$                      # free gradient of objective
    calculate $\pi_M(-\nabla_M f) \in T_M\mathcal{O}^{d \times r}$          # search direction (Equation 34)
    while $f(r_M(\beta\pi_M(-\nabla_M f)))$ is not sufficiently smaller than $f(M)$ do
        adjust step size $\beta$                  # linesearch (using retraction, Equation 35)
    end while
    $M \leftarrow r_M(\beta\pi_M(-\nabla_M f))$                              # iterate
end while
return (local) minima $M^*$ of $f$.
```

It is worth noting that the above operations imply a two-stage gradient step: Algorithm 1 first projects the free gradient onto the tangent space (π_M), and second the proposed step is retracted onto the manifold (r_M). It is natural to ask why one does not perform this projection in one step, for example by projecting the free gradient directly onto the manifold. Firstly, while there is a rich literature of such 'one-step' projected gradient methods (Bertsekas, 1976), convergence guarantees only exist for convex constraint sets. Indeed, all matrix manifolds we have discussed are nonconvex (except the trivial $\mathbb{R}^{d \times r}$). The theory of convergence for nonconvex manifolds requires this two-step procedure. Secondly, in our empirical experience, while a one-step projection method does often converge, that convergence is typically much slower than Algorithm 1.

This basic algorithm is extended in two ways: first, the constraint manifold $\mathcal{M}$ is taken to be the Grassmann manifold or some other manifold structure (like the product of Stiefel manifolds, as in CCA above); and second, conjugate gradient methods or second-order optimization techniques can be similarly adapted to the setting of matrix manifolds. Beyond these steps, understanding optimization over matrix manifolds in full generality requires topological and differential geometric machinery that is beyond the scope of this work. All of these topics are discussed in the key reference to this appendix (Absil et al., 2008), as well as the literature cited throughout this paper.

References

T. E. Abrudan, J. Eriksson, and V. Koivunen. Steepest descent algorithms for optimization under unitary matrix constraint. *IEEE Trans Signal Processing*, 56:1134–1147, 2008.

P. Absil and J. Malick. Projection-like retractions on matrix manifolds. *SIAM Journal on Optimization*, 22(1):135–158, 2012.

P. A. Absil, R. Mahony, and R. Sepulchre. *Optimization Algorithms on Matrix Manifolds.* Princeton University Press, Princeton, 2008.

K. P. Adragni and D. Cook. Sufficient dimension reduction and prediction in regression. *Phi. Trans. Royal Soc. A*, 367:4385–4405, 2009.

S. Amari. Natural gradient learning for over and under complete bases in ICA. *Neural Computation*, 11:1875–1883, 1999.

A. Baccini, P. Besse, and A. de Faguerolles. A L1-norm PCA and heuristic approach. In *Proc Intl Conf Ordinal and Symbolic Data Analysis*, pages 359–368, 1996.

F. Bach, R. Jenatton, J. Mairal, and G. Obozinski. Convex optimization with sparsity-inducing norms. *Optimization for Machine Learning*, pages 19–53, 2011.

G. H. Bakır, A. Gretton, M. Franz, and B. Schölkopf. Multivariate regression via Stiefel manifold constraints. In *Pattern Recognition*, pages 262–269. Springer, 2004.

A. Bar-Hillel, T. Hertz, N. Shental, and D. Weinshall. Learning distance functions using equivalence relations. In *ICML*, volume 3, pages 11–18, 2003.

M. Belkin and P. Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural computation*, 15(6):1373–1396, 2003.

D. P. Bertsekas. On the goldstein-levitin-polyak gradient projection method. *IEEE Transactions on Automatic Control*, 21(2):174–184, 1976.

C. M. Bishop. *Pattern Recognition and Machine Learning.* Springer, New York, 2006.

I. Borg and P. J. Groenen. *Modern multidimensional scaling: Theory and applications.* Springer Verlag, 2005.

M. Borga, T. Landelius, and H. Knutsson. A unified approach to PCA, PLS, MLR, and CCA. *Technical Report*, 1997.

N. Boumal, B. Mishra, P. Absil, and R. Sepulchre. Manopt, a matlab toolbox for optimization on manifolds. *Journal of Machine Learning Research*, 15:1455–1459, 2014.

S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations & Trends in Machine Learning*, 3(1):1–122, 2011.

A. Bray and D. Martinez. Kernel-based extraction of slow features: Complex cells learn disparity and translation invariance from natural images. In S. Thrun and K. Obermayer, editors, *Advances in Neural Information Processing Systems*, pages 253–260. MIT Press, Cambridge, MA, 2002.

W. Brendel, R. Romo, and C. K. Machens. Demixed principal component analysis. In *Advances in Neural Information Processing Systems*, pages 2654–2662, 2011.

W. Buntine. Variational extensions to EM and multinomial PCA. In *In Proc. ECML 2002*, 2002.

C. J. C. Burges. Dimension reduction: a guided tour. *Foundations & Trends in Machine Learning*, 2(4):275–365, 2010.

E. J. Candes, X. Li, Y. Ma, and J. Wright. Robust principal component analysis? *Journal of the ACM*, 58(3):11:1–11:37, 2011.

G. Chechik, U. Shalit, V. Sharma, and S. Bengio. An online algorithm for large scale image similarity learning. In *Advances in Neural Information Processing Systems*, pages 306–314, 2009.

V. Choulakian. L1-norm projection pursuit principal component analysis. *Comput. Stat. Data Anal.*, 50(6):1441–1451, 2006.

M. M. Churchland, J. P. Cunningham, M. T. Kaufman, J. D. Foster, P. Nuyujukian, S. I. Ryu, and K. V. Shenoy. Neural population dynamics during reaching. *Nature*, 487:51–56, 2012.

R. R. Coifman and S. Lafon. Diffusion maps. *Applied and computational harmonic analysis*, 21(1):5–30, 2006.

M. Collins, S. Dasgupta, and R. E. Schapire. A generalization of principal component analysis to the exponential family. In *Advances in Neural Information Processing Systems.* MIT Press, Cambridge, MA, 2002.

T. F. Cox and M. A. Cox. *Multidimensional scaling*, volume 88. CRC Press, 2001.

J. P. Cunningham and B. M. Yu. Dimensionality reduction for large-scale neural recordings. *Nature neuroscience*, 17:1500–1509, 2014.

A d'Aspremont, L. El Ghaoui, M. I. Jordan, and G. R. Lanckriet. A direct formulation for sparse PCA using semidefinite programming. *SIAM Review*, 49:434–448, 2007.

A d'Aspremont, F. R. Bach, and L. El Ghaoui. Optimal solutions for sparse principal component analysis. *Journal of Machine Learning Research*, 9:1269–1294, 2008.

F. De la Torre. A least-squares framework for component analysis. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(6):1041–1055, 2012.

D. De Ridder, R. P. W. Duin, and J. Kittler. Texture description by independent components. In *Structural, Syntactic, and Statistical Pattern Recognition*, pages 587–596. Springer, 2002.

S. C. Deerwester, S. T. Dumais, T. K. Landauer, G. W. Furnas, and R. A. Harshman. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6):391–407, 1990.

A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm (with discussion). *J Royal Statistical Society B*, 39:1–38, 1977.

M. Der and L. K. Saul. Latent coincidence analysis: a hidden variable model for distance metric learning. In *Advances in Neural Information Processing Systems*, pages 3230–3238, 2012.

P. Diaconis and D. Freedman. Asymptotics of graphical projection pursuit. *The Annals of Statistics*, pages 793–815, 1984.

C. Eckart and G. Young. The approximation of one matrix by another of lower rank. *Psychometrika*, 1:211–218, 1936.

A. Edelman, T. A. Arias, and S. T. Smith. The geometry of algorithms with orthogonality constraints. *SIAM J. Matrix Anal. Appl.*, 1998.

K. Fan and A. J. Hoffman. Some metric inequalities in the space of matrices. *Proceedings of the American Mathematical Society*, 6:111–116, 1955.

S. Fiori. Quasi-geodesic neural learning algorithms over the orthogonal group: a tutorial. *J Machine Learning Research*, 6:743–781, 2005.

R. A. Fisher. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7(2):179–188, 1936.

J. H. Friedman. Exploratory projection pursuit. *Journal of the American Statistical Association*, 82(397):249–266, 1987.

K. Fukumizu, F. R. Bach, and M. I. Jordan. Dimensionality reduction for supervised learning with reproducing kernel hilbert spaces. *The Journal of Machine Learning Research*, 5:73–99, 2004.

K. Fukumizu, F. R. Bach, and M. I. Jordan. Kernel dimension reduction in regression. *The Annals of Statistics*, pages 1871–1905, 2009.

K. Fukunaga. *Introduction to statistical pattern recognition.* Academic press, 1990.

D. Gabay. Minimizing a differentiable function over a differentiable manifold. *J Optimization theory and applications*, 37(2):177–219, 1982.

J. S. Galpin and D. M. Hawkins. Methods of L1 estimation of a covariance matrix. *Comput. Stat. Data Anal.*, 5:305–319, 1987.

A. Globerson and S. T. Roweis. Metric learning by collapsing classes. In *Advances in Neural Information Processing Systems*, pages 451–458, 2005.

J. Goldberger, S. T. Roweis, G. Hinton, and R. Salakhutdinov. Neighbourhood components analysis. *Advances in Neural Information Processing Systems*, 2004.

G. H. Golub and C. F. Van Loan. *Matrix Computations, 3rd edition.* Hopkins University Press, Baltimore, 1996.

A. Gretton, O. Bousquet, A. Smola, and B. Schölkopf. Measuring statistical dependence with hilbert-schmidt norms. In *Algorithmic learning theory*, pages 63–77. Springer, 2005.

A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. Smola. A kernel two-sample test. *The Journal of Machine Learning Research*, 13(1):723–773, 2012.

D. R. Hardoon and J. Shawe-Taylor. Convergence analysis of kernel canonical correlation analysis: theory and practice. *Machine learning*, 74(1):23–38, 2009.

D. R. Hardoon, S. Szedmak, and J Shawe-Taylor. Canonical correlation analysis: An overview with application to learning methods. *Neural Computation*, 16(12):2639–2664, 2004.

T. Hastie, R. Tibshirani, and J. Friedman. *Elements of Statistical Learning, 2nd edition.* Cambridge University Press, Cambridge, UK, 2008.

X. He and P. Niyogi. Locality preserving projections. In *Advances in Neural Information Processing Systems*, volume 16, page 153, 2004.

X. He, D. Cai, S. Yan, and H. Zhang. Neighborhood preserving embedding. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, volume 2, pages 1208–1213. IEEE, 2005.

N. J. Higham. Matrix nearness problems and applications. In *Applications of Matrix Theory*, pages 1–27. Oxford University Press, 1989.

H. Hotelling. Relations between two sets of variates. *Biometrika*, 28:321–377, 1936.

P. J. Huber. Projection pursuit. *The annals of Statistics*, pages 435–475, 1985.

A. Hyvarinen, J. Karhunen, and E. Oja. *Independent Component Analysis.* John Wiley and Sons, 2001.

M. Joho, H. Mathis, and R. H. Lambert. Overdetermined blind source separation: Using more sensors than source signals in a noisy mixture. In *Proc. ICA*, pages 81–86, 2000.

M. Journee, Y. Nesterov, P. Richtarik, and R. Sepulchre. Generalized power method for sparse principal component analysis. *J Machine Learning Research*, 11:517–553, 2010.

R. E. Kalman. A new approach to linear filtering and prediction problems. *J. Basic Engineering*, 82:35–45, 1960.

T. Kaneko, S. Fiori, and T. Tanaka. Empirical arithmetic averaging over the compact stiefel manifold. *Signal Processing, IEEE Transactions on*, 61(4):883–894, 2013.

Y. H. Kao and B. Van Roy. Learning a factor model via regularized PCA. *Machine Learning*, 91(279-303), 2013.

B. Kulis. Metric learning: A survey. *Foundations & Trends in Machine Learning*, 5(4): 287–364, 2012.

R. Larsen. Decomposition using maximum autocorrelation factors. *Journal of Chemometrics*, 16:427–435, 2002.

N. D. Lawrence. A unifying probabilistic perspective for spectral dimensionality reduction: Insights and new models. *The Journal of Machine Learning Research*, 13(1):1609–1638, 2012.

D. D. Lee and H. S. Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401:788–791, 1999.

J. F. Li and X. Y. Hu. Procrustes problems and associated approximation problems for matrices with k-involutory symmetries. *Linear Algebra Appl.*, 434:820–829, 2011.

K. C. Li. Sliced inverse regression for dimension reduction. *Journal of the American Statistical Association*, 86(414):316–327, 1991.

D. Luenberger. The gradient projection method along geodesics. *Management Science*, 18 (11), 1972.

J. H. Manton. Optimization algorithms exploiting unitary constraints. *IEEE Trans Signal Processing*, 50:635–650, 2002.

J. H. Manton. On the various generalizations of optimization algorithms to manifolds. *Proc of Mathematical Theory of Network and Systems*, 2004.

K. V. Mardia, J. T. Kent, and J. M. Bibby. *Multivariate analysis.* Probability and mathematical statistics. Academic Press, 1979.

S. Mika, G. Rätsch, J. Weston, B. Schölkopf, and K. R. Müller. Fisher discriminant analysis with kernels. In *Proc IEEE Signal Processing Society Workshop.*, pages 41–48. IEEE, 1999.

L. Mirsky. Symmetric gauge functions and unitarily invariant norms. *Quarterly Journal of Mathematics*, 11:80–89, 1960.

A. Mnih and R. Salakhutdinov. Probabilistic matrix factorization. In *Advances in Neural Information Processing Systems*, pages 1257–1264, 2007.

S. Mohamed, K. Heller, and Z. Ghahramani. Bayesian exponential family PCA. In *Advances in Neural Information Processing Systems.* MIT Press, Cambridge, MA, 2008.

R. J. Muirhead. *Aspects of Multivariate Statistical theory, 2nd edition.* Wiley, New Jersey, 2005.

J. Nilsson, F. Sha, and M. I. Jordan. Regression on manifolds using kernel dimension reduction. In *Proceedings of the 24th international conference on Machine learning*, pages 697–704. ACM, 2007.

Y. Nishimori and S. Akaho. Learning algorithms utilizing quasi-geodesic flows on the Stiefel manifold. *Neurocomputing*, 67:106–135, 2005.

K. Pearson. On lines and planes of closest fit to systems of points in space. *Philosophical Magazine*, 2:559–572, 1901.

J. Peltonen, J. Goldberger, and S. Kaski. Fast semi-supervised discriminative component analysis. In *Machine Learning for Signal Processing, 2007 IEEE Workshop on*, pages 312–317. IEEE, 2007.

C. R. Rao. The utilization of multiple measurements in problems of biological classification. *Journal of the Royal Statistical Society. Series B*, 10(2):pp. 159–203, 1948.

C. E. Rasmussen and C.K.I. Williams. *Gaussian Processes for Machine Learning.* MIT Press, Cambridge, 2006.

J. D. Rennie and N. Srebro. Fast maximum margin matrix factorization for collaborative prediction. In *Proc. of the 22nd ICML*, pages 713–719. ACM, 2005.

S. T. Roweis. EM algorithms for PCA and sensible PCA. In *Advances in Neural Information Processing Systems.* MIT Press, Cambridge, MA, 1997.

S. T. Roweis and L. K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, December 2000.

E. Rubinshtein and A. Srivastava. Optimal linear projections for enhancing desired data statistics. *Statistical Computing*, 20:267–282, 2010.

A. Ruhe. Closest normal matrix finally found! *Technical Report, University of Goteberg*, 1987.

B. Schölkopf, A. Smola, and R. K. Muller. Kernel principal component analysis. *Advances in kernel methods: support vector learning*, pages 327–352, 1999.

P. H. Schonemann. A generalized solution of the orthogonal Procrustes problem. *Psychometrika*, 31:1–10, 1966.

C. Shen, H. Li, and M. J. Brooks. A convex programming approach to the trace quotient problem. In *Computer Vision–ACCV 2007*, pages 227–235. Springer, 2007.

C. Spearman. General intelligence, objectively determined and measured. *American Journal of Psychology*, 15:201–293, 1904.

N. Srebro and T. S. Jaakkola. Weighted low-rank approximations. *ICML*, 3:720–727, 2003.

N. Srebro, J. Rennie, and T. S. Jaakkola. Maximum-margin matrix factorization. In *Advances in Neural Information Processing Systems*, pages 1329–1336, 2004.

A. Srivastava and X. Liu. Tools for application-driven linear dimension reduction. *Neurocomputing*, 67:136–160, 2005.

J. V. Stone and J. Porrill. Undercomplete independent component analysis for signal separation and dimension reduction. *Technical Report*, 1998.

M. Stone and R. J. Brooks. Continuum regression: cross-validated sequentially constructed prediction embracing ordinary least squares, partial least squares and principal components regression. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 237–269, 1990.

L. Sun, S. Ji, and J. Ye. A least squares formulation for a class of generalized eigenvalue problems in machine learning. In *Proc of the 26th Intl Conf Machine Learning (ICML)*, pages 977–984. ACM, 2009.

P. Switzer and A. A. Green. Min/max autocorrelation factors for multivariate spatial imagery. *Technical Report, Stanford University*, 1984.

J. B. Tenenbaum, V. deSilva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, December 2000.

C. M. Theobald. An inequality with application to multivariate analysis. *Biometrika*, 62 (2):461–466, 1975.

R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B*, 58:267–288, 1996.

N. H. Timm. *Applied multivariate analysis.* Springer, 2002.

M. E. Tipping and C. M. Bishop. Probabilistic principal component analysis. *J Royal Statistical Society B*, 61(3):611–622, 1999.

W. S. Torgerson. Multidimensional scaling: I. theory and method. *Psychometrika*, 17(4): 401–419, 1952.

L. Torresani and K. Lee. Large margin component analysis. In *Advances in Neural Information Processing Systems*, pages 1385–1392, 2006.

R. Turner and M. Sahani. A maximum-likelihood interpretation for slow feature analysis. *Neural computation*, 19(4):1022–1038, 2007.

M. O. Ulfarsson and V. Solo. Sparse variable PCA using geodesic steepest descent. *IEEE Trans Signal Processing*, 56:5823–5832, 2008.

L. J. P. Van der Maaten, E. O. Postma, and H. J. Van den Herik. Dimensionality reduction: A comparative review. *Tilburg University Technical Report, TiCC-TR 2009-005*, 2009.

K. R. Varshney and A. S. Willsky. Linear dimensionality reduction for margin-based classification: high-dimensional data and sensor networks. *IEEE Trans Signal Processing*, 59: 2496–2512, 2011.

M. Wang, F. Sha, and M. I. Jordan. Unsupervised kernel dimension reduction. In *Advances in Neural Information Processing Systems*, pages 2379–2387, 2010.

K. Q. Weinberger and L. K. Saul. Unsupervised learning of image manifolds by semidefinite programming. *International Journal of Computer Vision*, 70(1):77–90, 2006.

K. Q. Weinberger and L. K. Saul. Distance metric learning for large margin nearest neighbor classification. *The Journal of Machine Learning Research*, 10:207–244, 2009.

K. Q. Weinberger, J. Blitzer, and L. K. Saul. Distance metric learning for large margin nearest neighbor classification. In *Advances in Neural Information Processing Systems*, pages 1473–1480, 2005.

M. Welling, F. Agakov, and C. K. I. Williams. Extreme component analysis. In *Advances in Neural Information Processing Systems*. MIT Press, Cambridge, MA, 2003.

M. Welling, R. S. Zemel, and G. E. Hinton. Probabilistic sequential independent components analysis. In *IEEE-Transactions in Neural Networks , Special Issue on Information Theory*, 2004.

C. K. I. Williams. On a connection between kernel PCA and metric multidimensional scaling. *Machine Learning*, 46(1-3):11–19, 2002.

C. K. I. Williams and F. Agakov. Products of Gaussians and probabilistic minor component analysis. *Neural Computation*, 14(5):1169–1182, 2002.

L. Wiskott. Slow feature analysis: A theoretical analysis of optimal free responses. *Neural Computation*, 15(9):2147–2177, 2003.

L. Wiskott and T. Sejnowski. Slow feature analysis: unsupervised learning of invariances. *Neural Computation*, 14(4):715–770, 2002.

E. P. Xing, M. I. Jordan, S. Russell, and A. Y. Ng. Distance metric learning with application to clustering with side-information. In *Advances in Neural Information Processing Systems*, pages 505–512, 2002.

S. Yan and X. Tang. Trace quotient problems revisited. In *Computer Vision–ECCV 2006*, pages 232–244. Springer, 2006.

L. Yang. An overview of distance metric learning. *Proc. Computer Vision and Pattern recognition, October*, 7, 2007.

L. Yang and R. Jin. Distance metric learning: A comprehensive survey. *Michigan State Universiy Technical Report*, 2006.

S. Yu, K. Yu, V. Tresp, H. P. Kriegel, and M. Wu. Supervised probabilistic principal component analysis. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 464–473. ACM, 2006.

L. Zhang, A. Cichocki, and S. Amari. Natural gradient algorithm for blind separation of overdetermined mixture with additive noise. *IEEE Signal Processing Letters*, 6(11): 293–295, 1999.

D. Zhao, Z. Lin, and X. Tang. Laplacian PCA and its applications. In *ICCV 2007*, pages 1–8, 2007.

H. Zou, T. Hastie, and R. Tibshirani. Sparse principal component analysis. *J Comp Graph Stats*, 15(2):265–286, 2006.