Supervised PCA: A Multiobjective Approach

Alexander Ritchie, Laura Balzano, and Clayton Scott

Abstract—Methods for supervised principal component analysis (SPCA) aim to incorporate label information into principal component analysis (PCA), so that the extracted features are more useful for a prediction task of interest. Prior work on SPCA has focused primarily on optimizing prediction error, and has neglected the value of maximizing variance explained by the extracted features. We propose a new method for SPCA that addresses both of these objectives jointly, and demonstrate empirically that our approach dominates existing approaches, i.e., outperforms them with respect to both prediction error and variation explained. Our approach accommodates arbitrary supervised learning losses and, through a statistical reformulation, provides a novel low-rank extension of generalized linear models.

1 Introduction

Supervised principal component analysis (SPCA) is, as its name suggests, the problem of learning a low dimensional data representation in the spirit of PCA, while ensuring that the learned representation is also useful for supervised learning tasks. SPCA has received considerable interest outside of machine learning [1], [2], [3], owing to the broad appeal of PCA and a desire to perform supervised dimensionality reduction. We introduce a straightforward yet novel approach to SPCA from the perspective of multiobjective optimization. In particular, we propose to solve SPCA by optimizing a criterion that explicitly balances the empirical risk associated to a supervised learning problem with the variance explained by the learned representation.

Compared to prior work on SPCA [4], [5], [6], [7], [8], our approach has several advantages. First, many prior works are specific to regression or classification, while our approach accommodates arbitrary loss functions. Second, several existing approaches operate in two stages, first learning the representation by one criterion, and subsequently inferring a prediction model by another. These approaches typically use correlation of the learned representation with the response variables as a proxy for the criterion of ultimate interest, e.g., classification accuracy. Third, many existing approaches do not have a means of specifying a trade-off between prediction and variation explained, which can lead to poor performance. In our approach, this trade-off is achieved by simply tuning a hyperparameter.

Most importantly, prior research on SPCA has only measured performance in terms of prediction error, and has not been concerned with whether the learned representation explains a lot of variation in the data. Our primary conclusion is that jointly optimizing prediction error and variation explained leads to improved generalization. In particular, our approach dominates existing SPCA methods in that it outperforms them in terms of both prediction error and variation explained. Optimizing variation explained thus serves as a form of regularization for the supervised learning problem, and can yield more interpretable features.

• *All Authors are with the Department of Electrical and Engineering and Computer Science, University of Michigan, Ann Arbor, MI, 48109. E-mail: aritch@umich.edu*

This paper makes the following contributions. First, we provide a formulation of SPCA based on multiobjective optimization. Second, we generalize the formulation via a statistical framework, providing a family of SPCA methods similar in spirit to generalized linear models (GLMs). Third, we provide an intuitive maximum likelihood estimation procedure based on manifold optimization. Fourth, we extend the proposed approach to the kernel setting. Finally, we evaluate the proposed approach on real and simulated data, supporting our findings mentioned above.

2 Background and Related Work

Let $X \in \mathbb{R}^{n\times p}$ be a data matrix whose rows are p-dimensional patterns or inputs, and let $Y \in \mathbb{R}^{n\times q}$ be an associated matrix of q-dimensional outputs. The goal of dimensionality reduction (DR) is to find an r-dimensional representation of the input data, $r < p$. If Y is used to find this representation, the problem is referred to as supervised dimensionality reduction (SDR). In this section we review PCA, the most common form of DR, as it relates to our contribution. We also review prior work on supervised PCA and other forms of SDR.

2.1 PCA

PCA was first formulated by Karl Pearson in 1901 [9] and later reinvented by Harold Hotelling [10]. Geoemtrically, it can be thought of as the problem of finding an affine subspace of best fit to a collection of points in the squared error sense. As an optimization problem, PCA can be written

$$\min_{L\in\mathbb{R}^{p\times r}} \ \|X - XLL'\|_F^2 \ \ s.t. \ \ L'L = I_r, \tag{1}$$

where X is assumed to be centered, I_r is the $r \times r$ identity matrix, and $\|A\|_F$ is the Frobenius norm of a matrix A. Projection of X to the subspace spanned by columns of the optimal L gives the best rank-r approximation of X in terms of squared reconstruction error. Equivalently, this projection has the statistical interpretation of capturing the largest possible variance in the data among all rank-r projections. That is, we maximize variation explained, which is given by

$$\text{v. e.} = \frac{\|XL\|_F^2}{\|X\|_F^2} \ \in [0,1]. \tag{2}$$

Note that this formulation of variation explained makes sense for any L with orthonormal columns.

The process of performing PCA prior to a regression task is referred to as principal component regression (PCR), a nice discussion of which is given by Jolliffe [11]. To the authors' knowledge, no such name exists for the analogous approach for classification. This work will refer to that method as principal component classification (PCC).

2.2 Supervised Dimension Reduction

PCA has enjoyed immense popularity in statistical analysis for the past century or so. It remains a useful tool for dimension reduction (DR) due to its effectiveness, ease of computation and interpretability. However, PCA does not make use of any supervisory information, and therefore DR via PCA may not be useful for subsequent classification or regression tasks. This stems from the fact that in most problems of interest, there is a tradeoff between directions that explain variation in X, and those that are predictive of Y. To overcome this limitation, several approaches to SDR have been proposed. We first describe some fundamental SDR methods and highlight their connections to PCA, and then proceed to review existing approaches to SPCA.

Fisher's linear discriminant, or Fisher discriminant analysis (FDA) is arguably the canonical example of supervised dimension reduction in the classification setting. FDA finds a dimension reduced representation of X such that interclass variation is maximized while intraclass variation is minimized. Though it may seem that FDA is generally preferable to PCA for classification, this has been shown not always to be the case, especially when the number of training samples is small [12]. A number of extensions of FDA have been proposed that relax these assumptions. For example, local Fisher discriminant analysis (LFDA) [13] modifies FDA by approximately preserving local distances between same-class points.

Partial least squares (PLS) regression finds projections of the input data that account for a high amount of variation, but are also highly correlated with projections of the dependent variables. It is somewhat different from other methods presented here, in that both X and Y are projected to a new space to determine their relationship. Without means of specifying the trade-off between correlation and variation, PLS tends to put preference on directions that account for high variation rather than high correlation, causing it to behave similarly to PCR [14].

Reduced rank regression (RRR) [15], [16] attempts to minimize regression error under the constraint that the coefficient matrix be low rank. Such models arise in econometrics and other settings where the underlying relationship between predictor and response is believed to be low rank. This model is intimately related to PCA [16], [17]. Yee and Hastie [18] extend RRR to encompass categorical response variables through what they call reduced rank vector generalized linear models (RRVGLMs). Their work primarily explores the case of reduced rank logistic regression.

The earliest of the SPCA approaches [4], which we call Bair's method, is a simple two stage procedure. First, feature selection based on univariate regression coefficients is performed. Second, PCA is performed on the data matrix consisting only of the selected features. This approach may not be optimal, especially in the case where features are jointly predictive but not individually predictive. Furthermore, the method is only applicable to univariate regression and binary classification. On the other hand, this approach has some rigorous theory including a consistency result under an assumption of perfect variable selection with high probability. Recently, the method of iterative supervised principal components (ISPCA) [8] has extended Bair's method to multiclass classification and reduced computational complexity via an iterative deflationary scheme.

A method herein referred to as Barshan's method [6] approaches SPCA by means of the Hilbert-Schmidt Independence Criterion (HSIC). In a universal reproducing kernel Hilbert space (RKHS), two random variables are independent if and only if their HSIC is zero. Barshan's method maximizes an empirical measure of the HSIC, which has the form of a trace maximization problem similar to PCA. This method has also been extended to sparse SPCA [19].

A more recent SPCA method, supervised singular value decomposition [7] (SSVD), takes a somewhat different approach. They propose an inverse regression model in which y is a factor in a low rank generative process for x. Specifically, the SSVD model has the form

$$X = UL' + E, \;\; U = YB + F,$$

where E and F are error matrices, U is a low-rank score matrix, and B is a coefficient matrix. This method has only been developed for regression.

The approach most similar to our work, and the only SPCA method to model prediction error directly, is supervised probabilistic principal component analysis [5] (SPPCA). SPPCA extends the probabilistic principal component analysis (PPCA) [20] framework. As with PPCA, the likelihood model of SPPCA allows for statistical testing and Bayesian inference. The method uses an EM algorithm, which can be slow to converge. In addition, this approach places the same amount of emphasis on the dependent and independent variables, and is sensitive to the relative dimensions of $\boldsymbol{x}$ and $\boldsymbol{y}$. However, SPPCA provides a convenient and straightforward extension to the semi-supervised setting. The relationship of SPPCA to the proposed work is further discussed in § 3.3.3.

Finally, we mention a related line of work [21], [22], [23], that takes a regularization approach for adding supervision to the sparse PCA problem [24].

The present work extends our preliminary work [25] in several ways. First, we motivate our approach from the perspective of multiobjective optimization, which is novel in the SPCA literature, and highlight the interperation of our criterion as a form of regularized empirical risk minimization. Second, we formulate a statistical model to generalize the optimization formulation. This allows us to develop a maximum likelihood approach for hyperparameter selection, eliminating the need for a computationally expensive cross-validation (CV) approach, and to draw connections to generalized linear models. Third, we extend our approach to the kernel setting, allowing for nonlinear SPCA. We also include several new experiments to highlight the role of Pareto optimality in SPCA and to show interpretability of the proposed method.

3 Approach

We propose an approach to SPCA based on multiobjective optimization that leads to a natural nonstatistical formulation. We then describe a generalization via a statistical model which connects SPCA to generalized linear models. Finally, we extend the method to the kernel setting.

3.1 Notation

Column vectors will be written as bold lowercase letters. The i^{th} standard basis column vector is written $\boldsymbol{e}_i$ and the vector of all ones is written $\mathbf{1}$. The i^{th} column of a matrix A is denoted A_i and the entry in the i^{th} row and j^{th} column A_{ij}. The transpose of a real valued matrix A is denoted A', while the pseudoinverse is written A^+. Bold lowercase letters with positive integer subscripts will refer to realized data samples viewed as column vectors, and will comprise the corresponding data matrices such that $X = [\boldsymbol{x}_1\ \boldsymbol{x}_2\ \dots\ \boldsymbol{x}_n]'$ and $Y = [\boldsymbol{y}_1\ \boldsymbol{y}_2\ \dots\ \boldsymbol{y}_n]'$. To simplify notation, all data matrices are assumed to have been centered, meaning the columns have zero mean. Random variables are written as regular font lowercase letters regardless of dimension. The set of positive integers $\{1, 2, \dots, k\}$, is written $[k]$.

3.2 Optimization Formulation

The goal of SPCA is to solve the supervised learning problem while simultaneously performing dimension reduction according to PCA. In other words, any approach to SPCA should learn a feature representation that gives good prediction while explaining as much variation as possible in the data. In general, these two goals are not aligned. Therefore, it is natural to treat SPCA as a multiobjective optimization problem. In multiobjective optimization, Pareto optimal solutions are those for which one objective cannot be improved without sacrificing performance with respect to another. The set of Pareto optimal solutions, called the Pareto frontier, defines a function in the space of performance measures, the epigraph (or hypograph) of which contains all achievable performances on the problem at hand. This concept is illustrated for SPCA in Figure 1.

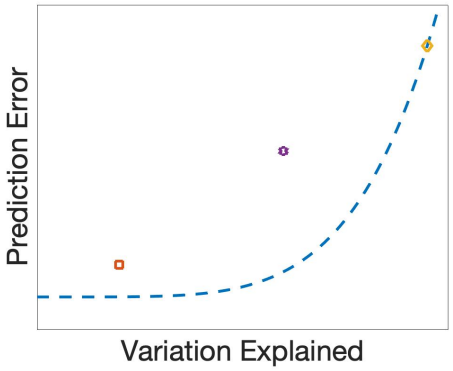


Fig. 1: Illustration of Pareto optimality in SPCA. The dashed blue curve represents the Pareto frontier, with the point on this curve representing a single Pareto optimal solution. Solutions above and to the left of the Pareto frontier are suboptimal in both performance measures.

Our approach to SPCA (proposed first in [25]) is to minimize a weighted sum of the PCA objective as given in § 2.1 and an empirical risk associated with the prediction problem. This is a direct way to explicitly trade off these two objectives at the expense of adding a tuning parameter, and can be expected to yield a Pareto optimal point that depends on the tuning parameter. The problem is formulated

$$\begin{aligned} \min_{L,\beta}\ & \frac{1}{n}\sum_{i=1}^{n} g(\boldsymbol{y}_i, \boldsymbol{x}_i, L, \beta) + \lambda \|X - XLL'\|_F^2 \\ s.t.\ & L'L = I_r, \end{aligned} \tag{3}$$

where $g(\cdot)$ is a loss function relating the dimension reduced data to its label, n is the number of observations, r a hyperparameter for subspace dimension, $\lambda > 0$ is a tuning parameter, and the remaining quantities are described in Table 1. The two loss functions suggested in [25] are the squared error and logistic losses given by

$$g_{\text{LS}}(\boldsymbol{y}_i, \boldsymbol{x}_i, L, \beta) = \|\boldsymbol{y}_i - \beta' L' \boldsymbol{x}_i\|_2^2 \quad \text{and}$$
$$g_{\text{LR}}(\boldsymbol{y}_i, \boldsymbol{x}_i, L, \beta) = \log \frac{\exp(\boldsymbol{x}_i' L \beta_{\boldsymbol{y}_i})}{\sum_{j'=1}^{q} \exp(\boldsymbol{x}_i' L \beta_{j'})},$$

respectively, where $\beta_{\boldsymbol{y}_i}$ is the column of β corresponding to the class given by $\boldsymbol{y}_i$. These two methods will be referred to as least squares PCA (LSPCA) and logistic regression PCA (LRPCA). Note that L is constrained to the Stiefel manifold,

TABLE 1: Description of Key Variables

VARIABLE	DESCRIPTION
$\underset{n\times p}{X}$	Data matrix
$\underset{n\times q}{Y}$	Response variables matrix
$\underset{p\times r}{L}$	Basis for the learned subspace
$\underset{n\times r}{XL}$	Dimension reduced form of X
$\underset{r\times q}{\beta}$	Learned coefficient matrix

i.e., the set of all matrices with orthonormal columns.

As the solution to (3) will not in general be given by the SVD, it is necessary to enforce the orthogonality of the columns of L if we hope to recover orthogonal components as in PCA. The primary means of solving such an optimization problem are manifold gradient algorithms which have been thoroughly developed in the literature [26], [27]. Our algorithms will be presented in § 4.

3.3 Statistical Formulation

In this section we propose a statistical reframing of the approach in (3). The benefit of the statistical approach is the incorporation of hyperparameters in a maximum likelihood estimation procedure, which obviates the need for parameter tuning by CV. This will be discussed in further detail in § C. We propose the following model

$$x \sim N(0, \sigma_x^2 I_p + \alpha LL'), \quad y|x \sim P_{y|x}, \tag{4}$$

where $\alpha > 0$ and $P_{y|x}$ is an exponential family distribution. Using the language of GLMs, let h be an invertible link function such that the first moment of $P_{y|x}$ satisfies $h(\mathbb{E}(Y|X)) = XL\beta$. Let $\ell_x(L, \sigma_x^2, \alpha; \boldsymbol{x}_i)$ be the log likelihood function for x evaluated at $\boldsymbol{x}_i$ and likewise define $\ell_{y|x}(\mathbb{E}(y|\boldsymbol{x}_i), \theta; \boldsymbol{y}_i) = \ell_{y|x}(h^{-1}(\beta' L' \boldsymbol{x}_i), \theta; \boldsymbol{y}_i)$ where θ represents any additional parameters needed to specify the

conditional distribution. Ignoring additive constants, the negative log likelihood (NLL) can be written

$$
\begin{aligned}
&2G(L,\beta,\alpha,\sigma_x^2,\theta;X,Y) \\
&\quad\triangleq -2\sum_{i=1}^{n}\ell_{y|x}(h^{-1}(\beta' L'\boldsymbol{x}_i),\theta;\boldsymbol{y}_i) \qquad (5)\\
&\qquad -2\sum_{i=1}^{n}\ell_x(L,\sigma_x^2,\alpha;\boldsymbol{x}_i) \\
&\quad= -2\sum_{i=1}^{n}\ell_{y|x}(h^{-1}(\beta' L'\boldsymbol{x}_i),\theta;\boldsymbol{y}_i) \qquad (6)\\
&\qquad +\frac{1}{\sigma_x^2}\|X-\frac{\sqrt{\sigma_x^2+\alpha}-\sigma_x}{\sqrt{\sigma_x^2+\alpha}}XLL'\|_F^2 \\
&\qquad +n(p-k)\log(\sigma_x^2)+nk\log(\sigma_x^2+\alpha),
\end{aligned}
$$

The derivation is shown in the appendix.

We are interested in the maximum likelihood estimates (MLEs) of L and β. The optimization problem is written

$$\min_{L,\beta,\alpha,\sigma_x^2,\theta} \; G(L,\beta,\alpha,\sigma_x^2,\theta;X,Y) \;\; s.t. \;\; L'L=I_r. \qquad (7)$$

We consider α, σ_x^2, and θ to be nuisance parameters, i.e., they are ultimately not of interest but must be accounted for to estimate L and β. Setting these parameters in practice will be discussed further in § C.

Examining the limiting behavior of G with respect to the nuisance parameters reveals several dimension reduction problems to be special cases of the suggested model. As $\alpha\to\infty$, minimizing G with respect to L and β is equivalent to (3) where $g=-\ell_{y|x}$. Similarly we obtain PCA as $\sigma_x^2\to 0$, the RRVGLM corresponding to $\ell_{y|x}$ as $\sigma_x^2\to\infty$, and maximum likelihood estimation of $\ell_{y|x}$ if we take $r=p$ (in which case L just represents a change of basis).

3.3.1 Reinterpreting LSPCA and LRPCA

The proposed model accommodates a variety of response models and link functions, drawing a parallel to generalized linear models (GLMs) [28]. The connection to GLMs is explored further in § 3.3.2. For brevity, we explore in detail only the cases where $P_{y|x}$ is Gaussian or categorical and h is the corresponding canonical link function.

For LSPCA we take the response variable to be Gaussian with identity link function. As such, $y|x\sim N(\beta' L'x,\sigma_y^2 I_q)$ and the NLL, ignoring additive constants, is

$$
\begin{aligned}
2G_{\text{LS}} &= \frac{1}{\sigma_y^2}\|Y-XL\beta\|_F^2+nq\log(\sigma_y^2) \\
&\quad+\frac{1}{\sigma_x^2}\|X-\frac{\sqrt{\sigma_x^2+\alpha}-\sigma_x}{\sqrt{\sigma_x^2+\alpha}}XLL'\|_F^2 \\
&\quad+n(p-k)\log(\sigma_x^2)+nk\log(\sigma_x^2+\alpha).
\end{aligned}
$$

Note that with regard to (6), we have $\theta=\sigma_y^2$ in this case.

For LRPCA we we take the response variable to be categorical with the logistic link function. Taking $\boldsymbol{y}_i$ to be one-hot vectors encoding class membership, the full NLL, again ignoring additive constants, is

$$
\begin{aligned}
2G_{\text{LR}} &= -2\sum_{i=1}^{n}\Psi(L,\beta;\boldsymbol{x}_i,\boldsymbol{y}_i) \\
&\quad+\frac{1}{\sigma_x^2}\|X-\frac{\sqrt{\sigma_x^2+\alpha}-\sigma_x}{\sqrt{\sigma_x^2+\alpha}}XLL'\|_F^2 \\
&\quad+n(p-k)\log(\sigma_x^2)+nk\log(\sigma_x^2+\alpha),
\end{aligned}
$$

where $\Psi(L,\beta;\boldsymbol{x}_i,\boldsymbol{y}_i)$ is the log softmax function

$$
\begin{aligned}
\Psi(L,\beta;\boldsymbol{x}_i,\boldsymbol{y}_i) &\triangleq \log\left(\Pr(y=\boldsymbol{y}_i|x=\boldsymbol{x}_i;L,\beta)\right) \\
&= \sum_{j=1}^{q}\boldsymbol{e}_j'\boldsymbol{y}_i\log\frac{\exp(\boldsymbol{x}_i'L\beta_j)}{\sum_{j'=1}^{q}\exp(\boldsymbol{x}_i'L\beta_{j'})}.
\end{aligned}
$$

Note that with regard to (6), there is no θ in this case as the categorical distribution is completely specified by the class probabilities.

Viewing G as a function of L and β with the nuisance parameters held fixed, we have

$$G=\sum_{i=1}^{n}-\ell_{y|x}(h^{-1}(\beta' L'\boldsymbol{x}_i),\theta;\boldsymbol{y}_i)+\lambda\|X-\gamma XLL'\|_F^2+c, \qquad (8)$$

where c is a constant term, $\lambda=\frac{1}{2\sigma_x^2}$ for LSPCA, $\lambda=\frac{\sigma_y^2}{\sigma_x^2}$ for LRPCA, and $\gamma=1-(\frac{\sigma_x^2}{\sigma_x^2+\alpha})^{\frac{1}{2}}$ for both.

The form of G in (8) shows the connection between the optimization and statistical formulations. For concision, moving forward we will use λ and γ, as previously defined, wherever possible. As such, we will define the NLL function arguments to be $G(L,\beta,\lambda,\gamma;X,Y)$. For clarity, we state the general optimization problem

$$\min_{L,\beta,\lambda,\gamma} \; G(L,\beta,\lambda,\gamma;X,Y) \;\; s.t. \;\; L'L=I_r. \qquad (9)$$

3.3.2 Connection to Generalized Linear Models

The proposed methods can be summarized as GLMs with two key modifications:

1) The link function $\widetilde{h}(\mathbb{E}(Y|X))=X\beta$ is replaced $h(\mathbb{E}(Y|X))=XL\beta$, which we call the *reduced rank link function*.
2) The parameters are estimated by optimizing the *joint* log likelihood $\ell_{x,y}$, while parameters for GLMs are estimated by optimizing the *conditional* log likelihood $\ell_{y|x}$.

When learning GLMs in high dimensions, regularization is used to avoid overfitting [29], [30]. In general, regularization can be thought of as the incorporation of additional information or assumptions for solving an ill-posed problem. For instance, ridge regression biases the regression coefficients toward the origin, expressing a degree of belief that the best solution should not have large norm. Alternatively, ridge regression can be viewed as shrinking the effects of the low-variance principal components of X on the regression estimate without ever completely removing them [14]. We can also think of our proposed methods as a form of regularization. For example, the *joint* NLL for LSPCA, restated here for convenience, is

$$\|Y-XL\beta\|_F^2+\lambda\|X-\gamma XLL'\|_F^2+c.$$

The *conditional* NLL consists only of the first term, and its optimum yields the RRR solution. It is clear minimizing the above with respect to L and β will not yield the RRR solution in general. Therefore, optimizing the joint likelihood rather than the conditional likelihood of the proposed models may be thought of as a form of regularization that shrinks the optimal L for RRR toward the PCA solution.

Viewed in the above context, the proposed methods are conceptually similar to RRVGLMs [18], of which RRR is a special case. However, the second point highlighted above

still distinguishes the two works. The RRVGLM model does not incorporate the low rank structure of the problem into a generative model for x. As such, the goal of RRVGLMs is to improve out of sample prediction while the focus of this work is SDR. It is possible to extend the proposed methods to the VGLM setting, but this is left to future work.

3.3.3 Connection to SPPCA

SPPCA takes a latent variable approach similar to PPCA, extending PPCA to the supervised setting by modeling the conditional distribution of y given the latent variable z. Furthermore, SPPCA assumes conditional independence of $y|z$ and $x|z$. The resulting model is

$$y|z \sim N(W_y z, \sigma_y^2 I_q),\ x|z \sim N(W_x z, \sigma_x^2 I_p),\ z \sim N(0, \sigma_z^2 I_r),$$

where W_x and W_y are learned parameters modeling x and y as linear functions of z, respectively.

The conditional independence assumption may be overly strong, especially when the subspace dimension is misspecified, e.g., the subspace dimension is set too small to capture the full relationship between y and x.

Now consider a latent variable model corresponding to LSPCA, where all variables retain their previous definitions:

$$y|x \sim N(\beta' L' x, \sigma_y^2 I_q),\ x|z \sim N(Lz, \sigma_x^2 I_p),\ z \sim N(0, \sigma_z^2 I_r).$$

Empirically we have observed that LSPCA significantly outperforms SPPCA (see § 5). It is clear that the latent variable models differ, though they possess many similarities. We now explore how the differences can explain the proposed method's improved performance. Consider the expectation step in the expectation maximization procedure for SPPCA [5],

$$z_i = (\frac{1}{\sigma_x^2} W_x' W_x + \frac{1}{\sigma_y^2} W_y' W_y + I_r)^{-1} (\frac{1}{\sigma_x^2} W_x' x_i + \frac{1}{\sigma_y^2} W_y' y_i),$$

where $z_i \in \mathbb{R}^k$ is the latent representation of the i^{th} data point (x_i, y_i). The above is the MLE of $z|x, y$. At test time, we do not have access to y and so z is taken to be the MLE of $z|x$, which does not depend on W_y. This is problematic since y does not depend on z through W_x. Therefore, it cannot be assumed that W_x will capture the relationship between y and z. Furthermore, if the end goal is to estimate y from $z|x$, it makes sense to directly encode this in the model. This is what LSPCA does. According to the LSPCA model, the MLE of $z|x$ is $z_i = L' x_i$ and y_i only depends on x_i through this quantity. Additionally, this change explicitly shares the parameter L between P_x and the $P_{y|x}$.

To make a direct comparison with SPPCA, we rewrite the LSPCA latent variable model such that x and y are conditioned on the same (reparameterized) latent variable:

$$\begin{aligned} &y|\widetilde{z} \sim N(\beta' \widetilde{z}, \sigma_y^2 I_q),\ \ x|\widetilde{z} \sim N(L\widetilde{z}, \sigma_x^2 (I_p - LL')), \\ &\quad \widetilde{z} \sim N(0, (1+\alpha)\sigma_x^2 I_r), \\ \implies &\ x \sim N(0, \sigma_x^2 (I_p + \alpha LL')). \end{aligned}$$

Details of the derivation are given in the appendix. The above yields some valuable insight: y and x are conditionally independent given the reparameterization $\widetilde{z}$ that explicitly assumes x is noiseless in the subspace corresponding to $\widetilde{z}$. This is a direct result of incorporating the MLE of $z|x$ in the model for $y|x$. While this causes the loss of the conditional independence assumption made by SPPCA, it also causes the MLE of $z|x$ to have a stronger relationship with y. Reparameterizing such that $\alpha \leftarrow \sigma_x^2 \alpha$ and integrating out the reparameterized latent variable $\widetilde{z}$ yields the LSPCA model.

3.4 Kernel Supervised Dimension Reduction

In this section we extend all proposed methods to perform kernel SDR. Further details are provided in the suplementary material.

Kernel PCA (kPCA) [31] is a means of performing nonlinear unsupervised dimension reduction by performing PCA in a high-dimensional feature space associated to a symmetric positive definite kernel. Let $k : \mathbb{R}^p \times \mathbb{R}^p \to \mathbb{R}$ be a symmetric positive definite kernel function. Associated to k is a high dimensional feature space $\mathcal{F}$ and mapping Φ such that $\Phi : \mathbb{R}^p \to \mathcal{F}$. The kernel matrix associated to k is $K_{ij} = k(\boldsymbol{x}_i, \boldsymbol{x}_j)$, and $k(\boldsymbol{y}, \boldsymbol{z}) = \langle \Phi(\boldsymbol{y}), \Phi(\boldsymbol{z}) \rangle_{\mathcal{F}}\ \forall \boldsymbol{y}, \boldsymbol{z} \in \mathbb{R}^p$. Let X_Φ be the matrix with n rows where each row is the representation in $\mathcal{F}$ of the corresponding row of X. Note that kPCA finds the projection of X_Φ onto its top r principal components, rather than the principal components themselves. Computing the principal components themselves is usually impractical or intractable as Φ may be unknown and/or $\mathcal{F}$ may be of arbitrarily high dimension. Computationally, all that is required is to find the eigenvectors corresponding to the r largest eigenvalues of the centered kernel matrix $\widetilde{K} = K - \frac{1}{n}\mathbf{1}\mathbf{1}'K - \frac{1}{n}K\mathbf{1}\mathbf{1}' + \frac{1}{n^2}\mathbf{1}\mathbf{1}'K\mathbf{1}\mathbf{1}'$. This amounts to solving

$$\hat{L} = \min_L \|\widetilde{K} - \widetilde{K}LL'\|_F^2 \ \ s.t.\ \ L'L = I_r, \tag{10}$$

where now $p = n$, and so L is $n \times r$. Noting (10) has the same form as (1), and that the projection of X_Φ onto its top r principal components is given by $\widetilde{K}\hat{L}$, we can kerneliize LSPCA and LRPCA (which we call kLSPCA and kLRPCA, respectively) by simply substituting $\widetilde{K}$ for X in (9), i.e.,

$$\min_{L,\beta,\lambda,\gamma} \ G(L, \beta, \lambda, \gamma; \widetilde{K}, Y) \ \ s.t.\ \ L'L = I_r. \tag{11}$$

4 Algorithms

In this section we present algorithms for solving the proposed optimization problems.

4.1 Grassmannian Constraints for Linear Prediction

All proposed methods have been presented with the Stiefel manifold constraint $L'L = I_r$. Considering the form of the objectives for LSPCA and LRPCA, the optimal value of the objective functions for a given L only depends on the subspace spanned by the columns of L. This can be seen by applying the same rotation to L and β. In settings such as this, the Grassmann manifold, the set of r dimensional subspaces in $\mathbb{R}^p$, is often used for ease of computation. We will only consider Grassmannian optimization in this work, since this allows projection to the tangent space and geodesic steps can be performed more efficiently. To be clear, even though points on the Grassmannian are subspaces, numerical algorithms require a representation of the subspace to be stored. These representations are taken to be matrices with orthogonal columns that span the subspace.

4.2 Proposed Algorithms

We propose an alternating optimization approach because the L and β subproblems can be solved relatively easily. Furthermore, the alternating approach allows the algorithm to be extended for response variables modeled by any invertible link function, as long as the inverse link function is differentiable with respect to L. Let the objective corresponding to the desired method be represented by G. For the squared error and logistic losses, in the linear and kernel settings the β subproblem is convex.

Though it is not convex, it is easily shown that the PCA problem on the Grassmannian admits no spurious local optima, i.e., a single critical point is a local minimum and all others are strict saddles or local maxima. Several recent works have studied this setting and shown that gradient descent and several other first order methods almost always avoid strict saddle points [32], [33]. This implies PCA can be solved via Grassmannian gradient descent.

The squared error and logistic losses are convex in L, and as a result the Hessian of the L subproblem is the Hessian of the PCA problem on the Grassmannian plus a positive (semi-)definite matrix. We suspect this makes the L subproblem well structured in a way that that makes optimization easy. Empirically, we observe that the proposed optimization scheme always converges to a good solution when initialized via PCA.

Since kLSPCA and kLRPCA have the same form as LSPCA and LRPCA, respectively, only the linear setting will be described. In the kernel setting, the only difference is the use of the kernel matrix K directly in place of X.

Because the β subproblems are unconstrained convex optimization problems, a wide variety of approaches can be utilized. For LSPCA the β subproblem is ordinary least squares (OLS) with data matrix XL and response matrix Y. Since, $XL \in \mathbb{R}^{n\times r}$, where r is the reduced dimension and likely small, the Cholesky decomposition can be used to efficiently solve the problem with complexity $\mathcal{O}(nr^2+r^3)$. For LRPCA, the subproblem is logistic regression with data matrix XL and responses Y. Common implementations of logistic regression use stochastic gradient or quasi-Newton methods. For our Matlab implementation, the backslash operator for LSPCA and built in logistic regression function for LRPCA were used.

The L subproblem for LRPCA and LSPCA is solved using manifold conjugate gradient descent (MCGD) on the Grassmannian [26]. We restate the algorithm using our notation in the appendix. In all algorithms we specify a call to $\mathrm{MCGD}(G(L), L_0)$, where G is a cost function to be minimized over the Grassmannian and L_0 is an initial iterate. We note that, while manifold gradient descent with Armijo line search is guaranteed to converge to a stationary point, no such guarantee exists for manifold conjugate gradient descent. However, it is known that if the algorithm converges to a local minimum, it does so superlinearly [26]. In any case, we observe excellent performance for the problems considered. We found the implementation of Grassmannian conjugate gradient in Manopt [34] to be more efficient than a Matlab only custom implementation. For this reason, we utilize Manopt to solve the L subproblem.

The necessary (Riemannian) partial derivatives with respect to L are

$$\begin{aligned}\operatorname{grad} G_{\mathrm{LS}} &= -(I_p - LL')X'(Y - XL\beta)\beta' \\ &\quad + \lambda\left(\gamma^2 - \frac{\gamma}{2}\right)(I_p - LL')X'XL \qquad (12)\end{aligned}$$

$$\begin{aligned}&\operatorname{grad} G_{\mathrm{LR}} = \\ &-(I_p - LL')\sum_{\substack{j\in[q]\\ i\in w_j}}\left(\frac{e^{\boldsymbol{x}_i'L\beta_j}\boldsymbol{x}_i\sum_{j'=1}^{q}e^{\boldsymbol{x}_i'L\beta_j}(\beta_{j'}' - \beta_j')}{(\sum_{j'=1}^{q}e^{\boldsymbol{x}_i'L\beta_j})^2}\right) \\ &+\lambda\left(\gamma^2 - \frac{\gamma}{2}\right)(I_p - LL')X'XL. \qquad (13)\end{aligned}$$

With all the pieces in place, a general alternating algorithm for LSPCA and LRPCA, which extends naturally to the corresponding kernel problems, is given in Algorithm 1. Before moving to experiments, we mention an alternative algorithm for LSPCA.

Algorithm 1 LSPCA/LRPCA Alternating Algorithm

Input: An $n \times p$ data matrix X, an $n \times q$ response matrix Y, a $p \times r$ orthogonal matrix L_0 with columns given by the first r principal components of X, the reduced dimension r, a hyperparameter $\lambda > 0$ (if doing CV)
Output: The $n \times r$ reduced data matrix Z^*, the coefficients β^*, a $p \times r$ orthogonal matrix L^* such that $Z^* = XL^*$

procedure $\mathrm{SPCA}_{\mathrm{ALT}}(X, Y, L_0, r, \lambda)$
  ▷ Initialize γ and β
  $\gamma \leftarrow 1$
  if LSPCA **then**
    $\beta_0 \leftarrow (XL_0)^+Y$
  else if LRPCA **then**
    $\beta_0 \leftarrow \mathrm{solveLR}(XL_0, Y)$
  $k \leftarrow 0$
  repeat
  ▷ Optionally, perform hyperparameter updates
    if MLE **then**
      $\gamma, \lambda \leftarrow \mathrm{UpdateParams}(X, Y, L_{k-1}, \beta_{k-1}, \gamma)$
  ▷ With β fixed, solve for L
    if LSPCA **then**
      $L_k \leftarrow \mathrm{MCGD}(G_{\mathrm{LS}}(L, \beta_{k-1}, \lambda, \gamma; X, Y), L_{k-1})$
    else if LRPCA **then**
      $L_k \leftarrow \mathrm{MCGD}(G_{\mathrm{LR}}(L, \beta_{k-1}, \lambda, \gamma; X, Y), L_{k-1})$
  ▷ With L fixed, solve for β
    if LSPCA **then**
      $\beta_k \leftarrow (XL_k)^+Y$
    else if LRPCA **then**
      $\beta_k \leftarrow \mathrm{solveLR}(XL_k, Y)$
    $k \leftarrow k + 1$
  until Convergence
  $Z = XL_k$
return Z, β_k, L_k

4.3 A Faster Algorithm for LSPCA

In the case of LSPCA, the optimal β given L is the OLS solution. Denote the OLS solution $\beta^*(L) = (XL)^+Y$ as a function of L. We define the objective function

$$G_{\mathrm{LS}}^{\mathrm{sub}}(L, \lambda, \gamma; X, Y) \triangleq G_{\mathrm{LS}}(L, \beta^*(L), \lambda, \gamma; X, Y)$$

It is easily observed from the chain rule

$$\nabla G_{\text{LS}}^{\text{sub}}(L) = \frac{\partial G_{LS}(L, \beta^*(L), \lambda, \gamma; X, Y)}{\partial L} + \frac{\partial \beta^*(L)}{\partial L} \underbrace{\frac{\partial G_{LS}(L, \beta^*(L), \lambda, \gamma; X, Y)}{\partial \beta}}_{=0}.$$

In words, calculating $\nabla G_{\text{LS}}^{\text{sub}}$ is the same as calculating the partial derivative of G_{LS} with respect to L and plugging in $\beta^*(L)$. The same argument applies to the Riemannian gradient. This allows us to eliminate β from the optimization problem by simple substitution in the objective and the gradient. Empirically, we observe this approach to be faster than the alternating optimization approach. It is applicable in both the MLE and non-MLE settings. The detailed procedure is given in Algorithm 2.

Algorithm 2 LSPCA Substitution Algorithm

Input: An $n \times p$ data matrix X, an $n \times q$ response matrix Y, a $p \times r$ orthogonal matrix L_0 with columns given by the first r principal components of X, the reduced dimension r, a hyperparameter $\lambda > 0$ (if doing CV)
Output: The $n \times r$ reduced data matrix Z^*, the coefficients β^*, a $p \times r$ orthogonal matrix L^* such that $Z^* = XL^*$

procedure $\text{LSPCA}_{\text{SUB}}(X, Y, L_0, r, \lambda)$
  $\gamma \leftarrow 1$
  $k \leftarrow 0$
  repeat
  ▷ Optionally, perform hyperparameter updates
    if MLE **then**
      $\beta \leftarrow (XL_{k-1})^+ Y$
      $\gamma, \lambda \leftarrow \text{UpdateParams}(X, Y, L_{k-1}, \beta, \gamma)$
  ▷ Solve for L
    $L_k \leftarrow \text{MCGD}(G_{\text{LS}}^{\text{sub}}(L, \lambda, \gamma; X, Y), L_{k-1})$
    $k \leftarrow k + 1$
  until Convergence
  $Z \leftarrow XL_k$
  $\beta \leftarrow Z^+ Y$
return Z, β, L_k

4.3.1 MLEs of the Hyperparameters

For fixed σ_x^2 and α, there is a regularization parameter λ such that the optimization problems (3) and (7) have identical solutions for L and β.[1] Therefore, when σ_x^2, and α (and σ_y^2 in the case of LSPCA) are set via CV, minimizing the NLL of the proposed model is equivalent to solving (3). Therefore, when setting parameters via CV $\gamma = 1$ is fixed and CV is performed over λ only.

Training of the proposed models requires choosing good nuisance parameter values. One can do this via CV or via maximum likelihood estimation, where parameter updates are incorporated in a block coordinate descent approach. In this section, we derive the maximum likelihood estimators of the nuisance parameters. We show the results here for LSPCA, noting that LRPCA is similar. The derivations are given in the appendix. Given L and β, the maximum likelihood estimates of σ_y^2, σ_x^2, and α are

$$\hat{\alpha} = \max(\frac{1}{nr}\|XL\|_F^2 - \hat{\sigma}_x^2, 0) \tag{14}$$

$$\hat{\sigma}_x^2 = \begin{cases} \frac{1}{np}\|X\|_F^2 & \hat{\alpha} = 0 \\ \frac{1}{n(p-r)}\left(\|X\|_F^2 - \|XL\|_F^2\right) & \hat{\alpha} > 0 \end{cases} \tag{15}$$

$$\hat{\sigma}_y^2 = \frac{1}{nq}\|Y - XL\beta\|_F^2. \tag{16}$$

The maximum likelihood estimates of γ and λ can then be calculated by susbstitution.

The biggest benefit of using maximum likelihood updates for the nuisance parameters is the elimination of the computationally burdensome CV procedure. As discussed in § 3.3, the number of tuning parameters can be reduced to one when using CV. However, this still requires solving the full optimization problem for each proposed tuning parameter value. On the other hand, using CV allows for the use of more general criteria for determining the "best" parameter value. For example, one may choose the parameter that yields the best prediction given a certain amount of variation explained.

4.4 Maximum Likelihood Hyperparameter Updates

Since the updates for $\hat{\sigma}_x^2$ and $\hat{\alpha}$ depend on each other, practical considerations must be made for a reasonable update procedure. Since $\hat{\alpha}$ depends on the value of $\hat{\sigma}_x^2$, while $\hat{\sigma}_x^2$ depends only on the positivity of $\hat{\alpha}$, the simplest approach is set $\hat{\sigma}_x^2$ first. Since $\alpha = 0$ implies x is an isotropic Gaussian random variable, a reasonable assumption for real data is $\alpha > 0$. Therefore we suggest initializing $(\hat{\sigma}_x^2)_0$ from the initial subspace estimate L_0 under the assumption that $\hat{\alpha} > 0$. The subsequent hyperparameter update procedure is given in Algorithm 3. To fully understand Algorithm 3, it must be viewed in the context of Algorithms 1 and 2 where the current values of L, β, and α are passed to Algorithm 3 to update α, σ_x^2 and, for LSPCA, σ_y^2. In the case of LRPCA, there is no $\hat{\sigma}_y^2$ to contend with but the updates for σ_x^2 and α are the same as above.

5 Experiments

The datasets used are outlined in Table 2. Most datasets are taken from University of California, Irvine machine learning repository[2] (UCI) or the Arizona State feature selection repository[3] (ASU). Where available, dataset specific links are provided in the appendix. We consider datasets in both the $n < p$ and $n > p$ settings. For the Music dataset, we uniformly subsampled 100 observations for the experiments to obtain a regression dataset in the $n < p$ setting.

In § 5.1, results are presented for comparison on the prediction task, as other SPCA works only consider this metric. As such, in § 5.1 CV is performed to minimize prediction error. In § 5.3 methods are evaluated on the basis of Pareto optimality.

For our method we give results for the MLE hyperparameter updates, and hyperparameter selection via CV. As discussed in § 3.3, $\gamma = 1$ was fixed while CV was performed

1. To see this, consider minimizing $\|X - \gamma XLL'\|_F^2$ over L and note that $\gamma \in (0, 1)$ does not affect the optimal L, just the optimal function value. We can therefore view fixed γ (equivalently, fixed σ_x and α) as a scale factor of the PCA term in (3), set γ to one, and re-scale λ accordingly.

2. https://archive.ics.uci.edu/ml/datasets.php
3. https://jundongl.github.io/scikit-feature/datasets.html

Algorithm 3 Maximum Likelihood Hyperparameter Updates

Input: An $n \times p$ data matrix X, an $n \times q$ response matrix Y, a $p \times r$ orthogonal matrix L, a $r \times q$ coefficient matrix β, a scalar parameter γ

Output: A scalar λ, a scalar γ

procedure UPDATEPARAMS(X, Y, L, β, γ)
  if $\gamma > 0$ **then** ▷ Equivalent to $\alpha > 0$
    $\sigma_x^2 \leftarrow \frac{1}{n(p-r)} \left(\|X\|_F^2 - \|XL\|_F^2 \right)$
  else
    $\sigma_x^2 \leftarrow \frac{1}{np} \|X\|_F^2$
  $\alpha \leftarrow \max(\frac{1}{nr} \|XL\|_F^2 - \sigma_x^2, 0)$
  $\gamma \leftarrow 1 - (\frac{\sigma_x^2}{\sigma_x^2 + \alpha})^{\frac{1}{2}}$
  if LSPCA **then**
    $\sigma_y^2 \leftarrow \frac{1}{nq} \|Y - XL\beta\|_F^2$
    $\lambda \leftarrow \frac{\sigma_y^2}{\sigma_x^2}$
  else if LRPCA **then**
    $\lambda \leftarrow \frac{1}{2\sigma_x^2}$
return γ, λ

for λ as well as the kernel width, where appropriate. We reserve discussion regarding differences between MLE and CV versions of our methods for § 5.4. The SPCA methods we compare against are Barshan's method, SPPCA, SSVD, and ISPCA. We take ISPCA to have subsumed Bair's method. Additionally, we compare against general SDR methods; RRR and PLS are compared against in the regression setting and FDA, LFDA, and kernel LFDA (kLFDA) are compared against in the classification setting. Among SPCA methods, Barshan's method is the only competitor that has proposed a kernelized version. As a baseline, we compare against PCR/PCC and kPCR/kPCC.

For each experiment, the best linear and kernel methods (including general SDR methods) are highlighted, while the best among the SPCA methods in the linear and kernel settings are marked with an asterisk ($*$). For each experiment 20% of the dataset was uniformly selected at random as an independent test set. For methods that require hyperparameter tuning, not including those using maximum likelihood parameter updates, the remaining 80% of data were then used in a 10-fold CV procedure. All methods were then trained on the full 80% with the set of parameters leading to smallest CV error, if applicable, before being evaluated on the independent test set. This process, including test set selection, was then repeated 10 times to produce the results in Table 3. For all kernel methods, a radial basis function (RBF) kernel was used. Note that the variation explained for all kernel methods is calculated with respect to the corresponding kernel matrix.

5.1 Prediction Performance

We first evaluate all the methods for a fixed subspace dimension $r = 2$. This process is repeated 10 times, and results are then averaged to produce the entries in Table 3. We deliberately choose $r = 2$, because this is often the dimension chosen for data visualization. This is meant both to provide some quantitative evaluation of potential visualization and to demonstrate performance in the case where limited memory or other resources make larger representations infeasible. We find our methods achieve better prediction error than existing SPCA methods in nearly every case and never perform worse than second best among all methods considered. In this setting, both SSVD and SPPCA seem to be heavily biased toward PCR/PCC. We further note that our methods are the only SPCA methods capable of consistently meeting or exceeding the performance of the non-SPCA methods considered. However, in the fixed dimension classification setting it appears LFDA and kLFDA are able to outperform the SPCA methods in several cases, albeit at the expense of substantial variation explained.

TABLE 2: Description of the datasets used herein. The type field denotes whether the dataset is for regression or classification. In the classification case q is the number of classes, while in the regression case it is the dimension of the response variable.

Name	Type	q	n	p	Source
Ionosphere	class.	2	354	34	UCI
Sonar	class.	2	208	60	UCI
Colon	class.	2	62	2000	ASU
Arcene	class.	2	200	10000	ASU
Residential	regr.	2	372	103	UCI
Music	regr.	2	100 (1059)	116	UCI
Barshan A	regr.	1	100	4	[6]
HCP	regr.	-	863	34716	[29], [35]

Next, we repeat the above experiments with the subspace dimension $r \geq 2$ chosen via 10-fold CV. Otherwise, the procedure is identical to that described for the $r = 2$ case. The proposed methods preform best among linear SPCA methods in five of six experiments. Additionally, we perform best overall in three of six experiments and are among the top three methods in the remainder. The proposed kernel methods are the best performers in four of six experiments.

5.2 Interpretability

In the classification setting LFDA and kLFDA again give the best prediction in several cases, while struggling to represent variation in the data even as higher subspace dimensions are allowed. However, some of our experiments suggest that good prediction without variation can lead to uninterpretable features. Figure 2 shows test set classification results on MNIST handwritten digits and fashion MNIST (FMNIST) clothing items. We compare embeddings learned by LFDA and LRPCA, since LFDA appears to be the closest competitor in terms of classification accuracy.

For the MNIST experiment, embeddings were learned for the task of binary classification of ones and sevens with subspace dimension $r = 2$. The features learned by LSPCA are clearly interpretable. Moving up and to the right along the direction of maximum variation for the ones yields greater clockwise rotation of the vertical section of either digit. Moving up and to the left yields greater length of the horizontal section that distinguishes the digits seven and one. We can also interpret intra-group variation. It appears that ones primarily vary in rotation, tending not to have the horizontal section present in the sevens, while the sevens have substantial variation along both of these features. As expected, points near the boundary between the two classes have small vertical sections, and look like

TABLE 3: Comparison of mean squared error (regression) or error rate (classification) of competing methods, with standard error shown in parentheses. Subspace dimension ($r = 2$) was held fixed for results in the first column of each dataset, with kernel parameters selected via 10-fold CV. For results in the second column, subspace dimension was also chosen by 10-fold CV. For each experiment, The best linear method is shown in ***red***, the best kernel method is shown in ***blue***, and the best SPCA methods in the linear and kernel settings are marked with an asterisk ($*$).

	Regression					
	Residential		Barshan A		Music	
	$r = 2$	**CV**	$r = 2$	**CV**	$r = 2$	**CV**
PCR	1.115 ± 0.462	0.430 ± 0.185	0.712 ± 0.346	0.401 ± 0.259	1.930 ± 0.170	1.770 ± 0.164
ISPCA	0.380 ± 0.212	0.097 ± 0.050	0.297 ± 0.094	0.288 ± 0.097	1.884 ± 0.204	1.751 ± 0.144
SPPCA	1.117 ± 0.464	1.097 ± 0.455	0.323 ± 0.128	0.308 ± 0.120	1.987 ± 0.167	1.987 ± 0.167
Barshan	0.684 ± 0.245	0.292 ± 0.085	0.298 ± 0.094	$*\mathbf{0.287 \pm 0.091}$	1.769 ± 0.156	1.691 ± 0.160
SSVD	1.115 ± 0.459	0.416 ± 0.171	0.379 ± 0.166	0.398 ± 0.153	1.931 ± 0.169	1.776 ± 0.169
PLS	0.525 ± 0.218	0.109 ± 0.036	$\mathbf{0.287 \pm 0.081}$	0.288 ± 0.081	1.770 ± 0.151	$\mathbf{1.620 \pm 0.131}$
RRR	0.112 ± 0.091	0.112 ± 0.091	0.289 ± 0.081	0.289 ± 0.081	1.633 ± 0.157	1.633 ± 0.157
LSPCA (CV)	$*\mathbf{0.070 \pm 0.043}$	$*\mathbf{0.060 \pm 0.030}$	0.291 ± 0.078	0.294 ± 0.078	$*\mathbf{1.632 \pm 0.156}$	1.667 ± 0.133
LSPCA (MLE)	0.103 ± 0.112	0.069 ± 0.032	$*0.289 \pm 0.081$	0.289 ± 0.081	1.655 ± 0.142	$*1.642 \pm 0.138$
kPCR	1.076 ± 0.195	0.631 ± 0.142	0.675 ± 0.276	0.341 ± 0.127	2.173 ± 1.091	2.090 ± 1.076
kBarshan	0.899 ± 0.212	0.761 ± 0.166	0.276 ± 0.099	0.269 ± 0.099	$*\mathbf{2.054 \pm 1.070}$	2.054 ± 1.077
kLSPCA (CV)	$*\mathbf{0.287 \pm 0.121}$	0.138 ± 0.097	$*\mathbf{0.163 \pm 0.068}$	$*\mathbf{0.162 \pm 0.065}$	2.061 ± 1.067	$*\mathbf{2.042 \pm 1.069}$
kLSPCA (MLE)	0.445 ± 0.502	$*\mathbf{0.131 \pm 0.096}$	0.223 ± 0.139	0.284 ± 0.144	2.114 ± 1.057	2.057 ± 1.055
	Classification					
	Ionosphere		Colon		Arcene	
	$r = 2$	**CV**	$r = 2$	**CV**	$r = 2$	**CV**
PCC	0.400 ± 0.033	0.146 ± 0.036	0.367 ± 0.090	0.217 ± 0.125	0.374 ± 0.093	0.323 ± 0.086
ISPCA	0.163 ± 0.041	0.134 ± 0.030	0.217 ± 0.137	0.258 ± 0.133	0.313 ± 0.058	0.269 ± 0.077
SPPCA	0.370 ± 0.047	0.173 ± 0.042	0.367 ± 0.090	0.208 ± 0.132	0.374 ± 0.093	0.323 ± 0.089
Barshan	0.146 ± 0.031	0.144 ± 0.041	0.258 ± 0.149	0.258 ± 0.114	0.344 ± 0.050	0.349 ± 0.070
FDA	0.147 ± 0.027	-	0.242 ± 0.107	-	0.228 ± 0.084	-
LFDA	0.160 ± 0.057	0.146 ± 0.033	0.225 ± 0.088	0.208 ± 0.119	$\mathbf{0.167 \pm 0.049}$	$\mathbf{0.169 \pm 0.052}$
RRLR	0.161 ± 0.042	0.151 ± 0.055	0.208 ± 0.106	$\mathbf{0.183 \pm 0.117}$	0.200 ± 0.112	0.208 ± 0.078
LRPCA (CV)	0.161 ± 0.042	$*\mathbf{0.127 \pm 0.025}$	$*\mathbf{0.192 \pm 0.104}$	$*0.200 \pm 0.125$	0.200 ± 0.112	0.223 ± 0.083
LRPCA (MLE)	$*\mathbf{0.141 \pm 0.026}$	0.153 ± 0.046	0.192 ± 0.125	0.242 ± 0.144	$*0.190 \pm 0.084$	$*0.195 \pm 0.058$
kPCC	0.429 ± 0.106	0.060 ± 0.029	0.342 ± 0.073	0.225 ± 0.118	0.349 ± 0.069	0.313 ± 0.071
kBarshan	0.300 ± 0.045	0.327 ± 0.124	0.333 ± 0.162	0.333 ± 0.162	0.359 ± 0.048	0.379 ± 0.081
kLFDA	$\mathbf{0.049 \pm 0.03}$	0.057 ± 0.038	$\mathbf{0.200 \pm 0.131}$	$\mathbf{0.183 \pm 0.110}$	$\mathbf{0.162 \pm 0.050}$	$\mathbf{0.162 \pm 0.040}$
kLRPCA (CV)	$*0.071 \pm 0.040$	$*\mathbf{0.056 \pm 0.030}$	$*0.225 \pm 0.111$	0.225 ± 0.111	$*0.231 \pm 0.073$	$*0.215 \pm 0.047$
kLRPCA (MLE)	0.406 ± 0.115	0.059 ± 0.030	0.358 ± 0.088	$*0.208 \pm 0.090$	0.349 ± 0.069	0.233 ± 0.083

they could be ones or sevens. We note the features learned by LRPCA appear to have the same interpretation as the PCA features, but with improved prediction accuracy. The LFDA embedding has the same property that digits near the boundary have short vertical sections, but there is not the same sense of continuous variation in length of this section. There is no apparent attribute of the digits that changes along the vertical embedding direction. Furthermore, it is difficult to interpret the intra-class variation for either digit.

For the FMNIST experiment, the experimental setup was identical to the MNIST experiment with the task being binary classification of shirts and dresses. Again the LRPCA and PCA features have similar clear interpretations, with LRPCA having slightly better prediction. In this case, moving up and to the right the length-to-width ratio of the clothing item, an obvious discriminatory feature between shirts and dresses, appears to decrease. Moving up and to the left the brightness of clothing appears to decrease. While not a discriminatory feature, this appears to be a major source of intra-class variation for both shirts and dresses. LFDA appears to learn the length-to-width ratio feature, albeit with substantially worse prediction accuracy. Again, LFDA does not seem to capture intra-class variation.

We consider the application of LSPCA to connectomic data from the Human Connectome Project (HCP) [29]. The data are constructed from functional magnetic resonance imaging of subjects brains, which are time-series, by a number of processing steps, the full details of which are given in [35]. First, for each subject, voxels are collected into a coarse partition consisting of 264 functional areas, known as the Power parcellation [36]. Next, voxel-wise behavior is spatially averaged within each functional area, resulting in 264 time-series from which correlation matrices are constructed. We then construct our data by vectorizing the upper-triangular portions of the subjects' correlation matrices. As in Sripada et al. [35], our task is to identify patters of correlated brain activity that predict certain response variables, called phenotypes, associated to each of the subjects, e.g., extroversion, processing speed. Figure 3a shows correlation of actual and predicted General Executive (GE) phenotype [35] on HCP as a function of subspace dimension. The experimental procedure used for this experiment is identical to that described in § 5.1. The current state of the art in this task is brain basis set (BBS) modeling [35], which learns a subspace of small dimension using PCA. LSPCA using $r = 5$ is able to achieve equivalent predictive performance to BBS with $r = 100$, providing for a simpler analysis of relevant connectomic structure using LSPCA. Figure 3b shows three of the first five components produced by PCA and LSPCA (PCs and LSPCs, respectively), reorganized according to the intrinsic connectivity network (ICN) assignments of Power [36]. Components $1 - 4$ are substantially similar between LSPCA and PCA, but the respective fifth components bear little similarity. This is confirmed by cosine similarity of corresponding components. Inclusion of the fifth LSPCA component increases average test set correlation of the predicted phenotype from 0.11 to 0.33, while inclusion of the fifth PC increases average correlation from 0.11 to 0.13. The ICN assignments are determined strictly by intra-individual phenomenon, while the PCs and LSPCs are determined by inter-individual variation. It is therefore remarkable that there should be such alignment between PCs and ICN structure, a matter which is discussed further by Sripada et al. (cite basic units). However, the

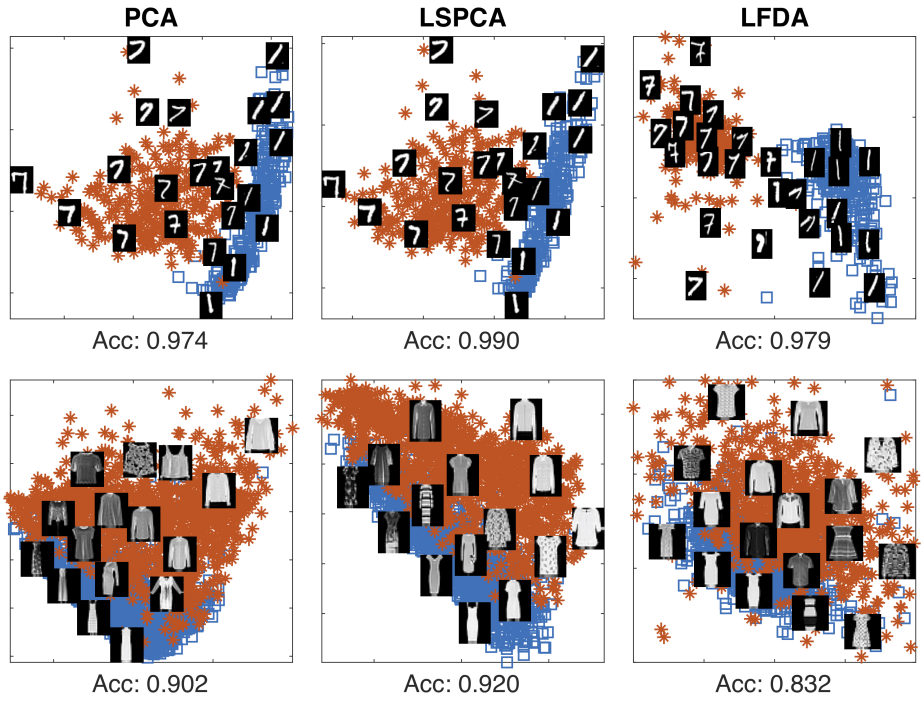


Fig. 2: Comparison of feature interpretability of PCA, LRPCA, and LFDA on the task of classifying (top) ones and sevens from MNIST (top) dresses and shirts from FMNIST.

fifth LSPC, which is the most predictive component, does not demonstrate substantial ICN structure. This suggests that the bulk of the predictive connectivity for GE is not aligned with the Power ICN. This suggests structure of the Power ICN is insufficient to fully understand inter-individual differences in GE.

5.3 Evaluating Pareto Optimality

In this section we compare the proposed approach to competitors through the lens of multiobjective optimization as described in § 3.2. The plots shown in Figure 4 correspond to the tests in Table 3, where the dimension $r = 2$ is fixed so the comparisons between methods can be direct and meaningful. Solutions that don't generalize to unseen data are of little practical use. We therefore show plots corresponding to training and test data. Figure 4i shows plots of variation explained vs. mean squared error of test data for residential and music datasets. Figure 4ii shows plots for training and test sets for ionosphere and colon datasets. The curves shown for the CV versions of LSPCA and LRPCA are parameterized by λ. The maximum likelihood estimates for LSPCA and LRPCA are also shown. All plots were generated according to the same procedure described in § 5.1.

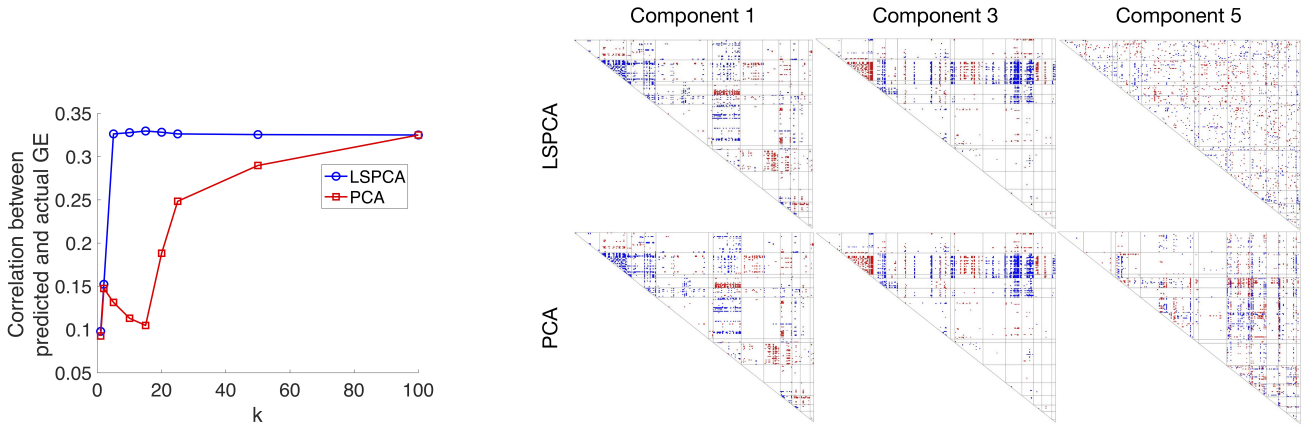
With regard to the regression experiments the proposed methods dominate all SPCA competitors in the Pareto sense. We remark that the maximum likelihood solution for kLSPCA appears to overfit on the residential dataset, performing worse than CV but still outperforming the kernel version of Barshan's method. In cases where one performance criterion is close, our method always appears to perform significantly better in the other criterion. Moreover, the proposed methods appear able to decrease prediction error substantially while losing little variation explained until a point of diminishing returns is reached. After this point, prediction error can be decreased only marginally for the price of substantial variation explained. The classification experiments show a similar pattern. The prominence of this behavior in the training plots suggests that our methods are finding points on or close to the Pareto frontier. Again we see that the maximum likelihood solution for kLRPCA appears to overfit. This supports evidence from § 5.1 that these methods are able to perform well when $r > 2$ is allowed, but seems to struggle in the restricted subspace dimension setting.

5.4 Maximum Likelihood vs. CV

In the supervised setting we find that maximum likelihood hyperparameter updates often yield prediction performance on par with or better than CV, particularly in higher dimension. However, the $r = 2$ experiments show that CV can produce substantially better results for kLRPCA and kLSPCA in the very low dimensional subspace setting. We therefore recommend using CV when r is set very low, as in visualization experiments. Given the computational cost of CV, the maximum likelihood approach is likely superior for practical use when higher dimensions are considered.

6 Conclusion

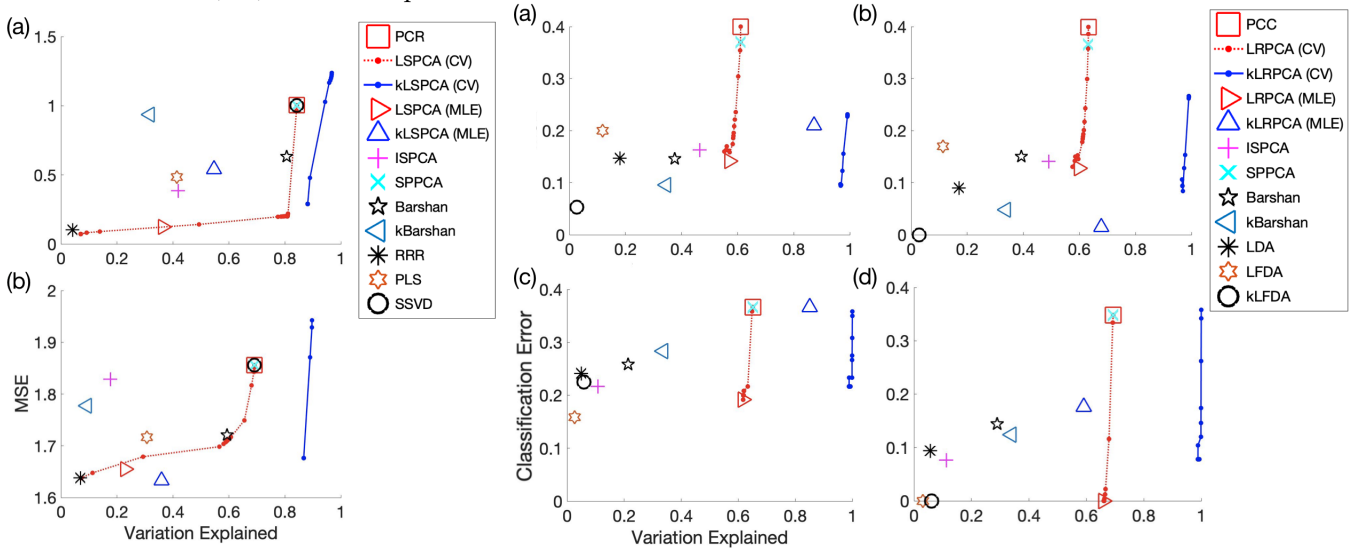
We proposed an intuitive, statistically motivated framework for SPCA in various prediction settings. The method generalizes PCA, RRR, and other reduced rank prediction problems, extending them to the kernel setting. We demonstrated that the proposed approach dominates existing SPCA methods and is competitive with other SDR methods in terms of prediction while outperforming them in the variation explained task. The proposed maximum likelihood hyperparameter updates alleviate the need to perform CV, often



(a) Correlation vs. subpsace dimension.

(b) Visualization of components one, three, and five.

Fig. 3: (3a) Correlation between predicted and actual GE as a function of subspace dimension. (3b) Comparison of components produced by PCA and LSPCA on HCP data, with the components reshaped to reflect ICN assignments of Power [36]. Blue (red) denote component entries that are two standard deviations above (below) the component mean.



(i) Regression Experiments

(ii) Classification Experiments

Fig. 4: Comparison of Pareto optimality of competing methods in terms of prediction error and variation explained, with subspace dimension $r = 2$. Figure 4i shows results for regression datasets (a) residential and (b) music. Figure 4ii shows results for (a) ionosphere and (c) colon. Additionally Figure 4ii shows training error for (b) ionosphere and (d) colon.

yielding better prediction, though occasionally sacrificing variation explained.

Though performance of the proposed methods is encouraging, there are still issues that need further study. Good convergence behavior is observed for the proposed algorithms, but a complete theory is lacking. Future work should address the connection between convergence of the proposed algorithms, and global optimization guarantees of the L and β subproblems.

The statistical formulation of our approach naturally suggests some directions of future work. For example, a Bayesian approach with sparsifying priors on L and β would be of considerable interest. The consistency and sample complexity properties of the maximum likelihood solution should also be explored.

Finally, the use of PCA is ubiquitous in experimental research of the hard sciences, as well as the social sciences. Applying our approach with the proper reduced rank link functions could yield meaningful insight into important problems. For example, given the prevalence of PCA and ordinal regression tasks in neuroscience [35], biology [37], and other fields, applying our method with the ordered logit response models could be a promising new approach.

Acknowledgments

This work was partially supported by AFOSR FA9550-19-1-0026, ARO W911NF1910027, NSF CCF-1845076, NSF BIGDATA IIS-1838179, and DARPA 16-43-D3M-FP-037.

References

[1] S. Roberts and M. A. Martin, "Using supervised principal components analysis to assess multiple pollutant effects," *Environmental health perspectives*, vol. 114, no. 12, pp. 1877–1882, 2006.

[2] X. Chen, L. Wang, J. D. Smith, and B. Zhang, "Supervised principal component analysis for gene set enrichment of microarray data with continuous or survival outcomes," *Bioinformatics*, vol. 24, no. 21, pp. 2474–2481, 2008.

[3] X. Chen, L. Wang, B. Hu, M. Guo, J. Barnard, and X. Zhu, "Pathway-based analysis for genome-wide association studies using supervised principal components," *Genetic epidemiology*, vol. 34, no. 7, pp. 716–724, 2010.

[4] E. Bair, T. Hastie, D. Paul, and R. Tibshirani, "Prediction by supervised principal components," *Journal of the American Statistical Association*, vol. 101, no. 473, pp. 119–137, 2006.

[5] 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*. ACM, 2006, pp. 464–473.

[6] E. Barshan, A. Ghodsi, Z. Azimifar, and M. Z. Jahromi, "Supervised principal component analysis: Visualization, classification and regression on subspaces and submanifolds," *Pattern Recognition*, vol. 44, no. 7, pp. 1357–1371, 2011.

[7] G. Li, D. Yang, A. B. Nobel, and H. Shen, "Supervised singular value decomposition and its asymptotic properties," *Journal of Multivariate Analysis*, vol. 146, pp. 7–17, 2016.

[8] J. Piironen and A. Vehtari, "Iterative supervised principal components," in *International Conference on Artificial Intelligence and Statistics*, 2018, pp. 106–114.

[9] K. Pearson, "Liii. on lines and planes of closest fit to systems of points in space," *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, vol. 2, no. 11, pp. 559–572, 1901.

[10] H. Hotelling, "Analysis of a complex of statistical variables into principal components." *Journal of educational psychology*, vol. 24, no. 6, p. 417, 1933.

[11] I. T. Jolliffe, "A note on the use of principal components in regression," *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, vol. 31, no. 3, pp. 300–303, 1982.

[12] A. M. Martínez and A. C. Kak, "PCA versus LDA," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, no. 2, pp. 228–233, 2001.

[13] M. Sugiyama, "Dimensionality reduction of multimodal labeled data by local Fisher discriminant analysis," *Journal of machine learning research*, vol. 8, no. May, pp. 1027–1061, 2007.

[14] J. Friedman, T. Hastie, and R. Tibshirani, *The Elements of Statistical Learning*. Springer series in statistics New York, 2001, vol. 1.

[15] T. W. Anderson *et al.*, "Estimating linear restrictions on regression coefficients for multivariate normal distributions," *The Annals of Mathematical Statistics*, vol. 22, no. 3, pp. 327–351, 1951.

[16] A. J. Izenman, "Reduced-rank regression for the multivariate linear model," *Journal of multivariate analysis*, vol. 5, no. 2, pp. 248–264, 1975.

[17] R. Velu and G. C. Reinsel, "Multivariate reduced-rank regression: theory and applications." Springer Science & Business Media, 2013, vol. 136, ch. 2, pp. 35–38.

[18] T. W. Yee and T. J. Hastie, "Reduced-rank vector generalized linear models," *Statistical modelling*, vol. 3, no. 1, pp. 15–41, 2003.

[19] S. Sharifzadeh, A. Ghodsi, L. H. Clemmensen, and B. K. Ersbøll, "Sparse supervised principal component analysis (sspca) for dimension reduction and variable selection," *Engineering Applications of Artificial Intelligence*, vol. 65, pp. 168–177, 2017.

[20] M. E. Tipping and C. M. Bishop, "Probabilistic principal component analysis," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 61, no. 3, pp. 611–622, 1999.

[21] S. Kawano, H. Fujisawa, T. Takada, and T. Shiroishi, "Sparse principal component regression with adaptive loading," *Computational Statistics & Data Analysis*, vol. 89, pp. 192–203, 2015.

[22] ——, "Sparse principal component regression for generalized linear models," *Computational Statistics & Data Analysis*, vol. 124, pp. 180–196, 2018.

[23] S. Kawano, "Sparse principal component regression via singular value decomposition approach," arXiv preprint, February 2020, https://arxiv.org/abs/2002.09188.

[24] H. Zou, T. Hastie, and R. Tibshirani, "Sparse principal component analysis," *Journal of computational and graphical statistics*, vol. 15, no. 2, pp. 265–286, 2006.

[25] A. Ritchie, C. Scott, L. Balzano, D. Kessler, and C. S. Sripada, "Supervised principal component analysis via manifold optimization," in *Proceedings of 2019 IEEE Data Science Workshop (DSW)*, 2019.

[26] A. Edelman, T. A. Arias, and S. T. Smith, "The geometry of algorithms with orthogonality constraints," *SIAM journal on Matrix Analysis and Applications*, vol. 20, no. 2, pp. 303–353, 1998.

[27] P.-A. Absil, R. Mahony, and R. Sepulchre, *Optimization algorithms on matrix manifolds*. Princeton University Press, 2009.

[28] J. A. Nelder and R. W. Wedderburn, "Generalized linear models," *Journal of the Royal Statistical Society: Series A (General)*, vol. 135, no. 3, pp. 370–384, 1972.

[29] S. A. Van de Geer, "High-dimensional generalized linear models and the lasso," *The Annals of Statistics*, vol. 36, no. 2, pp. 614–645, 2008.

[30] X. Wang and M. Wang, "Variable selection for high-dimensional generalized linear models with the weighted elastic-net procedure," *Journal of Applied Statistics*, vol. 43, no. 5, pp. 796–809, 2016.

[31] B. Schölkopf, A. Smola, and K.-R. Müller, "Kernel principal component analysis," in *International conference on artificial neural networks*. Springer, 1997, pp. 583–588.

[32] J. D. Lee, M. Simchowitz, M. I. Jordan, and B. Recht, "Gradient descent only converges to minimizers," in *Conference on learning theory*, 2016, pp. 1246–1257.

[33] J. D. Lee, I. Panageas, G. Piliouras, M. Simchowitz, M. I. Jordan, and B. Recht, "First-order methods almost always avoid strict saddle points," *Mathematical Programming*, pp. 1–27, 2019.

[34] N. Boumal, B. Mishra, P.-A. Absil, and R. Sepulchre, "Manopt, a Matlab toolbox for optimization on manifolds," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1455–1459, 2014.

[35] C. Sripada, M. Angstadt, S. Rutherford, D. Kessler, Y. Kim, M. Yee, and E. Levina, "Basic units of inter-individual variation in resting state connectomes," *Scientific reports*, vol. 9, no. 1, p. 1900, 2019.

[36] J. D. Power, A. L. Cohen, S. M. Nelson, G. S. Wig, K. A. Barnes, J. A. Church, A. C. Vogel, T. O. Laumann, F. M. Miezin, B. L. Schlaggar *et al.*, "Functional network organization of the human brain," *Neuron*, vol. 72, no. 4, pp. 665–678, 2011.

[37] B. W. Dulken, D. S. Leeman, S. C. Boutet, K. Hebestreit, and A. Brunet, "Single-cell transcriptomic analysis defines heterogeneity and transcriptional dynamics in the adult neural stem cell lineage," *Cell reports*, vol. 18, no. 3, pp. 777–790, 2017.

Alexander Ritchie received his B.S.E.E. in 2016 from Georgia Institute of Technology, Atlanta, Georgia. He is currently a Ph.D. candidate in Electrical and Computer Engineering at University of Michigan. His current research interests include nonconvex optimization and fairness in machine learning, as well as ethics and public policy for AI.

Laura Balzano Laura Balzano is an associate professor in Electrical Engineering and Computer Science at the University of Michigan. She is recipient of the NSF Career Award, ARO Young Investigator Award, AFOSR Young Investigator Award, and faculty fellowships from Intel and 3M. Her expertise is in statistical signal processing, matrix factorization, and optimization.

Clayton Scott Clay Scott received his PhD in Electrical Engineering from Rice University in 2004, and is currently Professor of Electrical Engineering and Computer Science at the University of Michigan. He researches statistical machine learning theory and algorithms, with an emphasis on nonparametric methods for supervised and unsupervised learning. He has also worked on a number of applications stemming from various scientific disciplines, including brain imaging, nuclear threat detection, environmental monitoring, and computational biology. In 2010, he received the CAREER Award from the National Science Foundation.

APPENDIX A
DERIVATION OF NLL

In this section we derive, in general terms, the NLL for the proposed model

$$x \sim N(0, \sigma_x^2 I_p + \alpha LL'), \quad y|x \sim P_{y|x},$$

and put it in functional form of the optimization formulation. From the above, we can write the NLL directly as

$$\begin{aligned} G(L, \beta, \alpha, \sigma_x^2, \theta) \triangleq &- \sum_{i=1}^{n} \ell_{y|x}(\mathbb{E}_{y|x}(y|\boldsymbol{x}_i), \theta; \boldsymbol{y}_i) \\ &- \sum_{i=1}^{n} \ell_x(L, \sigma_x^2, \alpha; \boldsymbol{x}_i). \end{aligned}$$

To supplement what is shown in the main paper, w are interested in finding a simplified form for ℓ_x. In order to draw a connection to the optimization formulation, we make a few observations. First we can rewrite the covariance matrix of x as

$$\sigma_x^2 I_p + \alpha LL' = (\sigma_x I_p + \eta LL')^2,$$

where $\eta = \sqrt{\sigma_x^2 + \alpha} - \sigma_x$. Second, we can write the inverse of the covariance matrix

$$\begin{aligned} \left(\sigma_x^2 I_p + \alpha LL'\right)^{-1} &= (\sigma_x I_p + \eta LL')^{-2} \\ &= \frac{1}{\sigma_x^2}\left(I_p - \frac{\eta}{\sigma_x} LL'\right)^{-2} \\ &= \frac{1}{\sigma_x^2}\left(I_p - \frac{\eta}{\sigma_x} L(I_r + \frac{\eta}{\sigma_x} L'L)^{-1} L'\right)^2 \\ &= \frac{1}{\sigma_x^2}\left(I_p - \frac{\eta}{\sigma_x} L(\frac{\sigma_x + \eta}{\sigma_x} I_r)^{-1} L'\right)^2 \\ &= \frac{1}{\sigma_x^2}\left(I_p - \frac{\eta}{\sigma_x + \eta} LL'\right)^2 \end{aligned}$$

where the second step uses the matrix inversion lemma. Third, we simplify the determinant of the covariance matrix

$$\begin{aligned} \left|\sigma_x^2 I_p + \alpha LL'\right| &= \sigma_x^{2p} \left| I_p + \frac{\alpha}{\sigma_x^2} LL' \right| \\ &= \sigma_x^{2p} \left| I_r + \frac{\alpha}{\sigma_x^2} L'L \right| \\ &= \sigma_x^{2p} \left| \frac{\sigma_x^2 + \alpha}{\sigma_x^2} I_r \right| \\ &= \sigma_x^{2p} \left(\frac{\sigma_x^2 + \alpha}{\sigma_x^2} \right)^k, \end{aligned}$$

where the first step makes use of the Weinstein–Aronszajn identity.

We now rewrite the NLL omitting additive constants as

$$\begin{aligned} -2\sum_{i=1}^{n} \ell_x(L, \sigma_x^2, \alpha; \boldsymbol{x}_i) &= \operatorname{Tr}\left(X(\sigma_x^2 I_p + \alpha LL')^{-1} X'\right) \\ &\quad + n \log\left|\sigma_x^2 I_p + \alpha LL'\right| \\ &= \frac{1}{\sigma_x^2} \operatorname{Tr}\left(X \left(I_p - \frac{\eta}{\sigma_x + \eta} LL' \right)^2 X' \right) \\ &\quad + n \log\left(\sigma_x^{2p} \left(\frac{\sigma_x^2 + \alpha}{\sigma_x^2} \right)^k \right) \\ &= \frac{1}{\sigma_x^2} \| X - \frac{\eta}{\sigma_x + \eta} XLL' \|_F^2 \\ &\quad + n(p-k)\log(\sigma_x^2) + nk \log(\sigma_x^2 + \alpha). \end{aligned}$$

A resubstitution for η gives the form shown in the main paper.

APPENDIX B
ON SPPCA

SPPCA takes a latent variable approach similar to PPCA, extending PPCA to the supervised setting by modeling the conditional distribution of y given z. Furthermore, SPPCA assumes conditional independence of $y|z$ and $x|z$. The resulting model is

$$y|x \sim N(W_y z, \sigma_y^2 I_q),\ x|z \sim N(W_x z, \sigma_x^2 I_p),\ z \sim N(0, \sigma_z^2 I_r).$$

The conditional independence assumption may be overly strong, especially when the subspace dimension is misspecified, e.g., the subspace dimension is set too small to capture the full relationship between y and x.

Now consider a latent variable model for LSPCA, where all variables retain their previous definitions (L still has orthonormal columns):

$$y|x \sim N(\beta' L' x, \sigma_y^2 I_q),\ x|z \sim N(Lz, \sigma_x^2 I_p),\ z \sim N(0, \sigma_z^2 I_r).$$

Conditioning y on x alleviates the issue caused by the conditional independence assumption of SPPCA. Forming the joint distribution of x and y, ignoring log terms and additive constants, and integrating out the latent variable yields

$$\begin{aligned} f_{x,y}(x,y) = &f_{y|x}(y|x) \int_{-\infty}^{\infty} f_{x|z}(x) f_z(z) dz \\ &\propto \exp(-\frac{1}{2\sigma_y^2} \|y - \beta' L' x\|_2^2 \\ &\quad - \frac{\sigma_z^2}{2\sigma_x^2(\sigma_x^2 + \sigma_z^2))} x' (\frac{\sigma_x^2 + \sigma_z^2}{\sigma_z^2} I_p - LL') x) \end{aligned}$$

where the $f_{(\cdot)}$ are the corresponding density functions, and logarithmic terms are ignored. If we write $\sigma_z^2 = \alpha \sigma_x^2$ with $\alpha = 2\eta\sigma_x + \eta^2$ (implying $\eta = \sqrt{\sigma_x^2 + \alpha} - \sigma_x$, as before) the negative log likelihood evaluated on data matrices X and Y reduces to

$$-G_{\text{LS}} \propto \|Y - XL\beta\|_F^2 + \frac{\sigma_y^2}{\sigma_x^2} \|X(I_p - \frac{\eta}{\sigma_x + \eta} LL')\|_F^2,$$

which matches the form of LSPCA.

Crucially, how should the conditioning of y on x rather than on z be interpreted? In the suggested model

$$x = Lz + \zeta_x \implies y = \beta'(z + L'\zeta_x) + \zeta_y,$$

where $\zeta_x \sim N(0, \sigma_x^2 I_p)$ and $\zeta_y \sim N(0, \sigma_y^2 I_q)$. First note that $L'\zeta_x \sim N(0, \sigma_x^2 I_r)$, i.e., it is isotropic Gaussian noise in the latent subspace. With the substitution $\sigma_z^2 = \alpha\sigma_x^2$ the model becomes

$$\begin{aligned} x &= L(z + \frac{z'}{\sqrt{\alpha}}) + (I_p - LL')\zeta_x \\ &= L\widetilde{z} + (I_p - LL')\zeta_x, \\ y &= \beta'(z + \frac{z'}{\sqrt{\alpha}}) + \underbrace{\beta'L'(I_p - LL')\zeta_x}_{=0} + \zeta_y \\ &= \beta'\widetilde{z} + \zeta_y \end{aligned}$$

where $z, z' \overset{\text{i.i.d.}}{\sim} N(0, \alpha\sigma_x^2 I_r)$ and $\widetilde{z} \sim N(0, (1+\alpha)\sigma_x^2 I_r)$. We can now write the conditional distributions of y and x on z

$$\begin{aligned} y|\widetilde{z} &\sim N(\beta'\widetilde{z}, \sigma_y^2 I_q), \\ x|\widetilde{z} &\sim N(L\widetilde{z}, \sigma_x^2(I_p - LL')), \\ \widetilde{z} &\sim N(0, (1+\alpha)\sigma_x^2 I_r), \\ \implies x &\sim N(0, \sigma_x^2(I_p + \alpha LL')). \end{aligned}$$

Reparameterizing such that $\alpha \leftarrow \sigma_x^2\alpha$ and integrating out the reparameterized latent variable $\widetilde{z}$ yields the LSPCA model.

APPENDIX C
MLES OF THE NUISANCE PARAMETERS

In this section we derive the MLE updates of the nuisance parameters. For LRPCA, the nuisance parameters are $\sigma_x * 2$ and α, while LSPCA adds an additional nuisance parameter σ_y^2. The partial derivatives w.r.t. $\sigma_x * 2$ and α will be the same for G_{LS} and G_{LR}, implying the MLEs $\sigma_x * 2$ and α will also be the same for both problems. Therefore, we derive the MLEs for LSPCA only.

Taking the partial derivative of G_{LS} with respect to σ_y^2 yields

$$\frac{\partial G_{\text{LS}}}{\partial \sigma_y^2} = \frac{1}{\sigma_y^2}\left(-\frac{1}{\sigma_y^2}\|Y - XL\beta\|_F^2 + nq\right)$$

which has a single zero at $\sigma_y^2 = \frac{1}{nq}\|Y - XL\beta\|_F^2$. The second partial derivative is positive at this point, making this the unique minimal σ_y^2.

Looking at the partial derivative with respect to α yields

$$\frac{\partial G_{\text{LS}}}{\partial \alpha} = -\frac{1}{(\sigma_x^2 + \alpha)^2}\|XL\|_F^2 + \frac{nr}{\sigma_x^2 + \alpha}$$

which has a single zero at $\alpha = \frac{1}{nr}\|XL\|_F^2 - \sigma_x^2$. Furthermore, G_{LS} is strictly increasing for $\alpha > \frac{1}{nr}\|XL\|_F^2 - \sigma_x^2$ and strictly decreasing for $\alpha < \frac{1}{nr}\|XL\|_F^2 - \sigma_x^2$, making this critical point a minimizer. Given the nonnegativity constraint on α, note this also implies that if $\frac{1}{nr}\|XL\|_F^2 - \sigma_x^2 < 0$, then the minimizer is $\alpha = 0$.

For the partial derivative with respect to σ_x^2 we have

$$\begin{aligned} \frac{\partial G_{\text{LS}}}{\partial \sigma_x^2} = &-\frac{1}{\sigma_x^4}\|X\|_F^2 + \frac{\alpha(2\sigma_x^2 + \alpha)}{\sigma_x^4(\sigma_x^2 + \alpha^2)}\|XL\|_F^2 \\ &+ \frac{n(p-r)}{\sigma_x^2} + \frac{nr}{\sigma_x^2 + \alpha}. \end{aligned}$$

Evaluating the above at the optimal α, we find that if $\alpha = 0$, $\sigma_x^2 = \frac{1}{np}\|X\|_F^2$, which is exactly what is expected from the model. On the other hand, if $\alpha > 0$ we can note the following. The partial derivative is zero when

$$\begin{aligned} 0 = &-(\sigma_x^2 + \alpha)^2\|X\|_F^2 + \left(\alpha\sigma_x^2 + \alpha(\sigma_x^2 + \alpha)\right)\|XL\|_F^2 \\ &+ (p-r)\sigma_x^2(\sigma_x^2 + \alpha)^2 + nr\sigma_x^4(\sigma_x^2 + \alpha). \end{aligned}$$

Plugging in $\alpha = \frac{1}{nr}\|XL\|_F^2 - \sigma_x^2$ yields the optimality condition

$$0 = \sigma_x^2\frac{p-r}{nr^2}\|XL\|_F^4 + \frac{1}{n^2r^2}\|XL\|_F^4(\|XL\|_F^2 - \|X\|_F^2),$$

which implies $\sigma_x^2 = \frac{1}{n(p-r)}(\|X\|_F^2 - \|XL\|_F^2)$. In either case G_{LS} is strictly decreasing as σ_x^2 approaches the critical point from the left, and strictly increasing as $\sigma_x^2 \to \infty$ from the right of the critical point, making the corresponding critical points minimizers.

In summary, given L and β, the maximum likelihood estimates of σ_y^2, σ_x^2, and α are

$$\hat{\alpha} = \max(\frac{1}{nr}\|XL\|_F^2 - \hat{\sigma}_x^2, 0) \tag{17}$$

$$\hat{\sigma}_x^2 = \begin{cases} \frac{1}{np}\|X\|_F^2 & \hat{\alpha} = 0 \\ \frac{1}{n(p-r)}\left(\|X\|_F^2 - \|XL\|_F^2\right) & \hat{\alpha} > 0 \end{cases} \tag{18}$$

$$\hat{\sigma}_y^2 = \frac{1}{nq}\|Y - XL\beta\|_F^2. \tag{19}$$

C.1 Kernel Supervised Dimension Reduction

In this section we extend all proposed methods to perform kernel SDR.

C.1.1 Kernel PCA

Kernel PCA (kPCA) [31] is a means of performing non-linear unsupervised dimension reduction by performing PCA in a high-dimensional feature space associated to a symmetric positive definite kernel.

Let $k : \mathbb{R}^p \times \mathbb{R}^p \to \mathbb{R}$ be a symmetric positive definite kernel function. Associated to k is a high dimensional feature space $\mathcal{F}$ and mapping Φ such that $\Phi : \mathbb{R}^p \to \mathcal{F}$. The kernel matrix associated to k is $K_{ij} = k(\boldsymbol{x}_i, \boldsymbol{x}_j)$, and $k(\boldsymbol{y}, \boldsymbol{z}) = \langle \Phi(\boldsymbol{y}), \Phi(\boldsymbol{z}) \rangle_{\mathcal{F}}$ $\forall \boldsymbol{y}, \boldsymbol{z} \in \mathbb{R}^p$. Let X_Φ be the matrix with n rows where each row is the representation in $\mathcal{F}$ of the corresponding row of X. Note that kPCA finds the projection of X_Φ onto its top r principal components, rather than the principal components themselves. Computing the principal components themselves is usually impractical or intractable as Φ may be unknown and/or $\mathcal{F}$ may be of arbitrarily high dimension. Computationally, all that is required is to find the eigenvectors corresponding to the r largest eigenvalues of the centered kernel matrix $\widetilde{K} = K - \frac{1}{n}\mathbf{1}\mathbf{1}'K - \frac{1}{n}K\mathbf{1}\mathbf{1}' + \frac{1}{n^2}\mathbf{1}\mathbf{1}'K\mathbf{1}\mathbf{1}'$. This amounts to solving

$$\begin{aligned} \hat{L} = &\min_L \|\widetilde{K} - \widetilde{K}LL'\|_F^2 \\ &s.t.\ \ L'L = I_r, \end{aligned} \tag{20}$$

where now $p = n$, and so L is $n \times r$. Let $\{\boldsymbol{v}_i\}_{i=1}^r$ be the top r principal components in the new feature space $\mathcal{F}$. The columns of $\hat{L}$ are such that

$$\boldsymbol{v}_i = \sum_{j=1}^{n} \hat{L}_{ji} \widetilde{\Phi}(\boldsymbol{x}_j),$$

where $\widetilde{\Phi}(\boldsymbol{x}_j) = \Phi(\boldsymbol{x}_j) - \frac{1}{n}\sum_{j'=1}^{n} \Phi(\boldsymbol{x}_{j'})$ is the centered representation of $\boldsymbol{x}_j$ in $\mathcal{F}$. To ensure the $\boldsymbol{v}_i$ are unit norm, the columns of $\hat{L}$ must be normalized to obtain $\bar{L}$ such that $\bar{L}'K\bar{L} = I_r$. The projection of a data point $\boldsymbol{x}$ onto the i^{th} component is

$$\langle \boldsymbol{v}_i, \widetilde{\Phi}(\boldsymbol{x}) \rangle_F = \sum_{j=1}^{n} \bar{L}_{ji} \widetilde{k}(\boldsymbol{x}, \boldsymbol{x}_j),$$

where

$$\begin{aligned}\widetilde{k}(\boldsymbol{y}, \boldsymbol{z}) &= \langle \widetilde{\Phi}(\boldsymbol{y}), \widetilde{\Phi}(\boldsymbol{z}) \rangle_{\mathcal{F}} \\ &= k(\boldsymbol{y}, \boldsymbol{z}) - \frac{1}{n}\sum_{i=1}^{n} (k(\boldsymbol{y}, \boldsymbol{x}_i) + k(\boldsymbol{x}_i, \boldsymbol{z})) \\ &+ \frac{1}{n^2}\sum_{j=1}^{n}\sum_{j'=1}^{n} k(\boldsymbol{x}_j, \boldsymbol{x}_{j'}).\end{aligned}$$

Most importantly for our purposes, the projection of the training data is

$$\Pi_{\{\boldsymbol{v}_i\}_{i=1}^r}(X) = \widetilde{K}\bar{L}. \tag{21}$$

We will refer to using kPCA in procedures analogous to PCR and PCC as kPCR and kPCC, respectively.

C.1.2 Kernel LSPCA and LRPCA

We highlight the fact that L does not have the same interpretation in the kernel setting as L in the linear setting. As in kPCA, in the problems to follow L provides coefficients for a low dimensional embedding and does not have a direct interpretation in terms of the importance of various features of the original data.

Recall that the projection of the training data is given by $\widetilde{K}\bar{L} \in \mathbb{R}^{n \times k}$. This suggests we could kernelize the proposed methods by substituting $\widetilde{K}$ for X. The problem is that the columns of $\bar{L}$ do not, in general, have unit norm in kPCA. Since the columns of $\bar{L}$ are just scaled versions of the columns of $\hat{L}$, there exists a $\bar{\beta}$ with scaled rows of β such that

$$\widetilde{K}\bar{L}\beta = \widetilde{K}\hat{L}\bar{\beta},$$

where the columns of L have unit norm. Therefore we can just substitute the kernel matrix $\widetilde{K}$ for the data matrix X in LSPCA and LRPCA, allowing the scaling to be absorbed by β. Similar to before, we can write the general kernel SPCA problem

$$\begin{aligned}&\min_{L,\beta,\lambda,\gamma} \; G(L, \beta, \lambda, \gamma; \widetilde{K}, Y) \\ &s.t. \;\; L'L = I_r.\end{aligned}$$

APPENDIX D
MANIFOLD CONJUGATE GRADIENT DESCENT ALGORITHM

Below, we state the manifold conjugate gradient descent algorithm [26], specifically for the Grassmann manifold.

Algorithm 4 Manifold Conjugate Gradient Descent [26]

Input: A cost function $G(L)$, a $p \times r$ orthogonal matrix L_0
Output: A solution L^*

procedure MCGD($G(L), L_0$)
  $\Delta_0 \leftarrow \operatorname{grad} G\big|_{L=L_0}$
  $C_0 \leftarrow -\Delta_0$
  $k \leftarrow 0$
  repeat
  ▷ Form compact SVD
    $U\Sigma V' \leftarrow \operatorname{svd}(C_k)$
  ▷ Perform a line search
    $t_k \leftarrow \min_t G(L_k V \cos(\Sigma t)V' + U\sin(\Sigma t)V')$
  ▷ Update L and search direction
    $L_{k+1} \leftarrow L_k V \cos(\Sigma t_k)V' + U\sin(\Sigma t_k)V'$
    $\Delta_{k+1} \leftarrow -(I_p - L_{k+1}L'_{k+1})(\frac{\partial G}{\partial L}\big|_{L=L_{k+1}})$
    $\widetilde{C}_{k+1} \leftarrow (-L_k V \sin(\Sigma t_k) + U\cos(\Sigma t_k))\Sigma V'$
    $A_k \leftarrow L_k V \sin(\Sigma t_k)$
    $B_k \leftarrow U(I - \cos(\Sigma t_k))$
    $\widetilde{\Delta}_k \leftarrow \Delta_k - (A_k + B_k)U'\Delta_k$
    $d_k \leftarrow \frac{\langle \Delta_{k+1} - \widetilde{\Delta}_k, \Delta_k \rangle}{\langle \Delta_k, \Delta_k \rangle}$
    $C_{k+1} \leftarrow -\Delta_{k+1} + d_k \widetilde{C}_k$
    if $k \equiv 0 \mod r(p-r)$ **then**
      $C_{k+1} \leftarrow -\Delta_{k+1}$
    $k \leftarrow k + 1$
  until Convergence
return L_k

APPENDIX E
LINKS TO DATASETS

The datasets used in this work that are directly available online are Ionosphere[4], Sonar[5], Colon[6], Arcene[7], Residential[8], and Music[9].

4. Ionosphere: https://archive.ics.uci.edu/ml/datasets/Ionosphere
5. Sonar: https://archive.ics.uci.edu/ml/datasets/Connectionist+Bench+%28Sonar%2C+Mines+vs.+Rocks%29
6. Colon: https://jundongl.github.io/scikit-feature/datasets.html
7. Arcene: https://jundongl.github.io/scikit-feature/datasets.html
8. Residential: https://archive.ics.uci.edu/ml/datasets/Residential+Building+Data+Set
9. Music: https://archive.ics.uci.edu/ml/datasets/Geographical+Original+of+Music