1. INTRODUCTION

Principal components analysis (PCA), and the singular value decomposition (SVD) upon which it is based, form the foundation of much of classical multivariate analysis. It is well known, however, that with both high-dimensional data and functional data, traditional PCA can perform poorly (Silverman 1996; Jolliffe, Trendafilov, and Uddin 2003; Johnstone and Lu 2009). Methods enforcing sparsity or smoothness on the matrix factors, such as sparse or functional PCA (SPCA or FPCA), have been shown to consistently recover the factors in these settings (Silverman 1996; Johnstone and Lu 2009). Recently, these techniques have been extended to regularize both the row and column factors of a matrix (Huang, Shen, and Buja 2009; Witten, Tibshirani, and Hastie 2009; Lee et al. 2010). All of these methods, however, can fail to capture relevant aspects of structured high-dimensional data, or data in which variables are associated with a location. In this article, we propose a general and flexible framework for PCA that can directly account for structural dependencies and incorporate two-way regularization for exploratory analysis of massive structured datasets.

Examples of high-dimensional structured data in which classical PCA performs poorly in terms of recovering the true subspace or low-rank signal abound in areas of imaging, environmental studies, time series and longitudinal studies, and network data. Consider, for example, functional MRI (fMRI) that measures three-dimensional images of the brain over time with high spatial resolution. Each three-dimensional pixel, or voxel, in the image corresponds to a measure of the bold oxygenation level-dependent (BOLD) response (hereafter referred to as “activation”), an indirect measure of neuronal activity at a particular location in the brain (Lazar 2008). In many fMRI studies, and especially in resting-state studies, a primary analysis goal is to find major brain activation patterns, or spatially contiguous groups of voxels exhibiting strong co-activation, and their associated temporal hemodynamic response patterns; this is often referred to as functional connectivity (Friston et al. 1993). To achieve this, the voxels are typically vectorized at each time point to form a high-dimensional matrix of voxels ($\approx 10,000–100,000$) by time points ($\approx 100–10,000$) to which unsupervised multivariate techniques are applied to find the major spatial and temporal patterns. Perhaps surprisingly to statisticians, PCA is typically not used for functional connectivity analysis in fMRI data. Instead, neuroimagers commonly employ independent component analysis (ICA; McKeown et al. 1998) and many extensions and variations of ICA (Calhoun et al. 2001; Beckmann and Smith 2005; Calhoun, Liu, and Adalı 2009); some have suggested PCA variations, such as FPCA (Viviani, Gron, and Spitzer 2005) or nonlinear PCA (Friston et al. 1999; Thirion and Faugeras 2003), yet these methods are still rarely used by neuroscientists.

The question then remains, why is PCA problematic for finding major patterns in fMRI data? Furthermore, and according to Rogers et al. (2007), “Two disadvantages of the approach [PCA] are the difficulty of interpreting the many eigenimage/time course pairs that can be produced from fMRI data ... and the propensity of a single feature of interest to be split across multiple components, complicating interpretation further.” Along these lines, many have noted that the first several PCs of fMRI data appear to capture spatial and temporal dependencies in the noise rather than patterns of brain activation; subsequent PCs can be a mix of noise patterns and signal of interest (Andersen, Gash, and Avison 1999; Friston et al. 1999; Thirion and Faugeras 2003; Viviani, Gron, and Spitzer 2005).
Perhaps, then, the strong structured dependencies in fMRI data are responsible for PCA’s poor performance in this context. The noise in fMRI data is characterized by strong correlation between neighboring voxels in the three-dimensional image, often made worse by a common preprocessing technique that smoothes the data via a Gaussian kernel smoother (Lazar 2008), and strong temporal dependencies over the time course measured for each voxel (Friston et al. 1995). Spatially, many have modeled the noise using random field models (Worsley et al. 1996), whereas temporally, many have employed autoregressive and other time series models (Friston et al. 1995). Given these well-understood noise dependencies directly related to the structure of fMRI data, we propose an extension of the PCA model and algorithm that takes these two-way dependencies directly into account in the matrix decomposition.

To understand the poor performance of PCA in the context of fMRI data and our proposed methods, we briefly examine them mathematically. Suppose we observe data \( \mathbf{Y} \in \mathbb{R}^{n \times p} \), for example, \( n \) voxels by \( p \) times points for fMRI data; PCA considers the following model: \( \mathbf{Y} = \mathbf{M} + \mathbf{UDV}^T + \mathbf{E} \), where \( \mathbf{M} \) denotes a mean matrix, \( \mathbf{D} \) the singular values, \( \mathbf{U} \) and \( \mathbf{V} \) the left and right factors, respectively, and \( \mathbf{E} \) the noise matrix where the elements are assumed to be independent and identically distributed (Mardia, Bibby, and Kent 1979; Anderson 2003). This PCA model is estimated by minimizing a Frobenius norm loss: \( \| \mathbf{X} - \mathbf{UDV}^T \|^2_F = \sum_{i=1}^n \sum_{j=1}^p (X_{ij} - u_i v_j)^2 \), where \( u_i \) is the \( i \)th column of \( \mathbf{U} \) and \( v_j \) is analogous and \( \mathbf{X} \) denotes the centered data, \( \mathbf{X} = \mathbf{Y} - \mathbf{M} \). This sums of squares loss function weighs errors associated with each matrix element equally and cross-product errors, between elements \( X_{ij} \) and \( X_{i'j'} \), for example, are ignored. It comes as no surprise then that the Frobenius norm loss and thus PCA perform poorly with data exhibiting strong structural dependencies among the matrix elements.

To permit unequal weighting of the matrix errors according to the data structure, we assume that the noise is structured: \( \text{vec}(\mathbf{E}) \sim (\mathbf{0}, \mathbf{R}^{-1} \otimes \mathbf{Q}^{-1}) \), or the covariance of the noise is separable and has a Kronecker product covariance (Gupta and Nagar 1999). Our loss function then changes from the Frobenius norm to a transposable quadratic norm that permits unequal weighting of the matrix error terms based on \( \mathbf{Q} \) and \( \mathbf{R} \): \( \| \mathbf{X} - \mathbf{UDV}^T \|^2_{Q,R} = \sum_{i=1}^n \sum_{j=1}^p \sum_{j'=1}^p Q_{ii'} \mathbf{R}_{jj'} (X_{ij} - u_i v_j)^2 \). By finding the best low-rank approximation of the data with respect to this transposable quadratic norm, we develop a decomposition that directly accounts for two-way structural dependencies in the data. This gives us a method, generalized PCA (GPCA) that extends PCA for applications to structured data.

After initial development of our methods, previous work on unequal weighting of matrix elements in PCA via a row and column generalizing operators came to our attention (Escoufier 1977). This so-called, “duality approach to PCA” is known in the French multivariate community, but has perhaps not gained the popularity in the statistics literature that it deserves (Escoufier 2006; Dray and Dufour 2007; Pardou 2007; Holmes 2008). Much more methodological development is needed to apply this concept of unequal matrix weights for dimension reduction of high-dimensional structured data. Our work is also related to the matrix factorization of Feng and He (2009) for repeated samples of matrix-variate data.

In this article, we aim to develop a flexible mathematical framework for PCA that accounts for known structure and incorporates regularization in a manner that is computationally feasible for analysis of massive structured datasets. Our specific contributions include (a) a matrix decomposition that generalizes the SVD to account for known two-way structure, (b) a framework for two-way regularization in PCA and in GPCA, which reveals the class of penalty functions that may be used to regularize the matrix factors, and (c) results and algorithms allowing one to compute (a) and (b) for ultra high-dimensional data. Specifically, beyond the previous work in this area reviewed by Holmes (2008), we provide results allowing for (i) general weighting matrices, (ii) computational approaches for high-dimensional data, and (iii) a framework for two-way regularization in the context of GPCA. As our methods are flexible and general, they will enjoy wide applicability to a variety of structured datasets common in medical imaging, remote sensing, engineering, environmetrics, and networking.

The article is organized as follows. In Sections 2 and 3, we develop the mathematical framework for our generalized least-square matrix decomposition (GMD) and resulting GPCA as well as two-way regularized GPCA. As the objective of this work is to build a theory and method foundation for GPCA and regularized GPCA, we assume that the weighting matrices, or quadratic operators, are fixed and known in these sections. To apply our methods in practice, however, one needs to specify these weighting matrices, a topic discussed in Section 2.5. Results on simulated and real fMRI data are given in Section 4 and we conclude with a discussion of our work in Section 5.

2. A GENERALIZED LEAST-SQUARE MATRIX DECOMPOSITION

2.1 GMD Model

Here and for the remainder of the article, we will assume that the observed data, \( \mathbf{X} \in \mathbb{R}^{n \times p} \) have previously been centered. Two PCA models are typically employed: the low-rank mean model (Anderson 2003) and the spiked-covariance model:

Low-rank mean model:
\[
\mathbf{X} = \sum_{k=1}^K d_k \mathbf{u}_k \mathbf{v}_k^T + \mathbf{E}; \quad \mathbf{E}_{ij} \overset{\text{iid}}{\sim} (0, 1).
\]

Spiked covariance model:
\[
\mathbf{X} = \sum_{k=1}^K d_k \mathbf{u}_k \mathbf{v}_k^T; \quad d_k \sim P, \quad \text{cov} (\mathbf{X}) = \Delta = \sum_{k=1}^K d_k \mathbf{v}_k \mathbf{v}_k^T.
\]

such that \( \mathbf{U}^T \mathbf{U} = \mathbf{I} \) and \( \mathbf{V}^T \mathbf{V} = \mathbf{I} \).

Here, the orthonormal PC row and column factors, \( \mathbf{u}_k \in \mathbb{R}^n \) and \( \mathbf{v}_k \in \mathbb{R}^p \), are assumed to be fixed signals of interest, and \( d_k \) is random according to some distribution \( P \). In the low-rank mean model, the noise is assumed to be additive (\( d_k \) is fixed), while the noise is multiplicative in the spiked-covariance model (\( d_k \) is random). We will predominately work with the additive noise model in this article.

As we alluded to in the introduction, the PCA model assumes that the elements of the additive noise are iid, an assumption that is unrealistic for data with strong structural dependencies such as fMRI data. We seek to broaden this model to permit
two-way dependencies in the noise:

\[ X = \sum_{k=1}^{K} d_k u_k v_k^T + E; \quad E \sim (0, \Delta \otimes \Sigma). \]

such that \( U^T \Sigma U = I \) and \( V^T \Delta V = I. \)  \hspace{1cm} (1)

Here, we assume that the additive noise matrix, \( E, \) arises from a matrix-variate model with separable row covariance \( \Sigma \in \mathbb{R}^{n \times n} \) and column covariance \( \Delta \in \mathbb{R}^{p \times p} \) (Gupta and Nagar 1999). In other words, if we vectorize the noise, then the noise covariance is given by the Kronecker product between the row and column covariances: \( \text{cov}(\text{vec}(E)) = \Delta \otimes \Sigma. \) Note that in practice, this separability assumption of the row and column covariances can be checked via several statistical tests (Mitchell, Genton, and Gumpertz 2006; Li, Genton, and Sherman 2008). So that the GMD model is identifiable, we assume that the factors are orthogonal with respect to the covariance of the noise.

While we motivate the GMD problem in the next section via the GMD mean model, another model of interest that permits two-way dependencies in the noise is a combination of the additive and multiplicative noise models of the mean model and spiked-covariance model:

GMD hybrid model:

\[ X = \sum_{k=1}^{K} d_k u_k v_k^T + E; \quad E \sim N_{n,p}(0, \Delta \otimes \Sigma) \& d_k \sim P. \]

such that \( U^T \Sigma U = I \) and \( V^T \Delta V = I. \)

\[ \iff \text{cov}(\text{vec}(X)) = \sum_{k=1}^{K} d_k v_k v_k^T \otimes \sum_{k=1}^{K} d_k u_k u_k^T + \Delta \otimes \Sigma. \]  \hspace{1cm} (2)

Thus, this hybrid model can be interpreted as a covariance decomposition model in which the separable row and column covariances are decomposed into a signal component and a noise component such that the eigenvectors of the signal are orthogonal with respect to those of the noise. We will revisit this model in Section 2.6 after we first introduce our GMD methodology.

### 2.2 GMD Problem

It is well known that the PCA model is solved by the SVD that gives the best rank-\( K \) approximation with respect to the Frobenius norm (Eckart and Young 1936):

\[
\text{minimize}_{U, V} ||X - UD^TV||_F^2 \quad \text{subject to} \quad U^TU = I_K, \\
V^TV = I_K \text{ and } \text{diag}(D) \geq 0. 
\]  \hspace{1cm} (3)

The solution is given by the first \( K \) singular vectors and singular values of \( X. \) This Frobenius norm loss naturally arises according to the probabilistic interpretation of PCA: if we assume that \( X \) is from the PCA low-rank mean model with multivariate normal noise, \( X \sim N(UD^TV, I), \) then the Frobenius norm loss is proportional to the spherical Gaussian log-likelihood.

We use this probabilistic interpretation of PCA to motivate our GMD optimization problem. Specifically, assume our GMD mean model (1), and let the noise be matrix-variate normal, \( X \sim N_{n,p}(UD^TV, \Delta \otimes \Sigma). \) Then the matrix-variate log-likelihood can be written as

\[ \ell(X|\Sigma, \Delta) \propto \text{tr}(\Sigma^{-1}(X - UD^TV)(X - UD^TV)^T) = ||X - UD^TV||_{\Sigma^{-1}}^2. \]

Notice that unlike the Frobenius norm loss of PCA, this loss uniquely weights the matrix errors according to the inverse row and column covariances. Formally, we define our loss and GMD problem in terms of general matrices \( Q \) and \( R, \) noting that the population GMD model is given by letting \( Q = \Sigma^{-1} \) and \( R = \Delta^{-1}; \) our loss function is then given by a transposable quadratic norm, the \( Q, R \)-norm, defined as follows:

**Definition 1.** Let \( Q \in \mathbb{R}^{n \times n} \) and \( R \in \mathbb{R}^{p \times p} \) be positive semidefinite matrices, \( Q, R \succeq 0. \) Then the \( Q, R \)-norm (or seminorm) is defined as \( ||X||_{Q,R} = \sqrt{\text{tr}(QXRXX^T)}. \)

We note that if \( Q \) and \( R \) are both positive definite, \( Q, R \succ 0, \) then the \( Q, R \)-norm is a proper matrix norm. If \( Q \) or \( R \) are positive semidefinite, then it is a seminorm, meaning that for \( X \neq 0, ||X||_{Q,R} = 0 \) if \( X \in \text{null}(Q) \) or if \( X^T \in \text{null}(R). \) We call the \( Q, R \)-norm a transposable quadratic norm, as it is right and left multiplies \( X \) and is thus an extension of the quadratic norm, or “norm induced by an inner product space” (Horn and Johnson 1985). Note that \( ||X||_{Q,R} \triangleq ||X||_{Q,1} \) and \( ||X^T||_{R} \triangleq ||X^T||_{R,1} \) are then quadratic norms.

Our GMD optimization problem is the best rank-\( K \) approximation to the data with respect to the \( Q, R \)-norm:

\[
\text{minimize}_{U, V} ||X - UD^TV||_{Q,R}^2 \quad \text{subject to} \quad U^TU = I_K, \\
V^TV = I_K \text{ and } \text{diag}(D) \geq 0. 
\]  \hspace{1cm} (4)

So we do not confuse the elements of the GMD with that of the SVD, we call \( U \) and \( V \) the left and right GMD factors, respectively, and the diagonal elements of \( D, \) the GMD values. The matrices \( Q \) and \( R \) are called the left and right quadratic operators of the GMD, respectively. Notice that the left and right GMD factors are constrained to be orthogonal in the inner product space induced by the \( Q \)-norm and \( R \)-norm, respectively, as in (1). Since the Frobenius norm is a special case of the \( Q, R \)-norm taking \( Q = I \) and \( R = I, \) the GMD is in fact a generalization of the SVD that permits unequal weighting of the matrix errors. In this article, we develop the methodology of a matrix decomposition and two-way matrix decomposition assuming that the left and right quadratic operators, \( Q \) and \( R, \) are known. While this may seem restrictive, for structured data, there are many well-studied choices for the quadratic operators that can be borrowed from the functional data analysis literature. These considerations are discussed in Section 2.6.

### 2.3 GMD Solution

The GMD solution, \( \hat{X} = U^*D^*(V^*)^T, \) is comprised of \( U^*, \ D^*, \) and \( V^*, \) the optimal points minimizing the GMD problem (4). The following result states that the GMD solution is an SVD of an altered data matrix.

**Theorem 1.** Suppose \( \text{rank}(Q) = l \) and \( \text{rank}(R) = m. \) Decompose \( Q \) and \( R \) by letting \( Q = Q_0Q_0^T \) and \( R = R_0R_0^T, \) then \( Q_0 \in \mathbb{R}^{n \times l} \) and \( R_0 \in \mathbb{R}^{p \times m} \) are of full column rank. Define \( \hat{X} = \hat{Q}_0^TXR \) and let \( \hat{X} = \hat{U}\hat{D}\hat{V}^T \) be the SVD of \( \hat{X}. \) Then, the GMD solution, \( \hat{X} = U^*D^*(V^*)^T, \) is given by the GMD factors \( U^* = \hat{Q}_0^{-1}\hat{U} \) and \( V^* = R_0^{-1}\hat{V} \) and the GMD values, \( D^* = \hat{D}. \)
Here, \( \tilde{Q}^{-1} \) and \( \tilde{R}^{-1} \) are any left matrix inverse: \( (\tilde{Q}^{-1})^T \tilde{Q} = I(1) \) and \( (\tilde{R}^{-1})^T \tilde{R} = I(m) \).

We make some brief comments regarding this result. First, the decomposition, \( Q = \tilde{Q} \tilde{Q}^T \), where \( \tilde{Q} \in \mathbb{R}^{n \times m} \) is of full column rank, exists since \( Q \succeq 0 \) (Horn and Johnson 1985); the decomposition for \( R \) exists similarly. A possible form of this decomposition, the resulting \( \tilde{X} \), and the GMD solution \( \tilde{X} \) can be obtained from the eigenvalue decomposition of \( \tilde{Q} \) and \( \tilde{R} \): \( \tilde{Q} = \Gamma^Q \Lambda^Q \Gamma^Q \), and \( \tilde{R} = \Gamma^R \Lambda^R \Gamma^R \). If \( Q \) is full rank, then we can take \( \tilde{Q} = \Gamma^Q \Lambda^{1/2} \Gamma^Q \), giving the GMD factor \( U^* = \Gamma^Q \Lambda^{1/2} \Gamma^Q \), similarly for \( \tilde{R} \) and \( \tilde{Q} \). On the other hand, if \( Q \succeq 0 \), then a possible value for \( \tilde{Q} \) is \( \Gamma^Q (\cdot, 1:1) \Lambda^{1/2} (1:1, 1:1) \), giving an \( n \times l \) matrix with full column rank. The GMD factor, \( U^* \) is then given by \( \Gamma^Q (\cdot, 1:1) \Lambda^{-1/2} (1:1, 1:1) \).

From Theorem 1, we see that the GMD solution can be obtained from the SVD of \( \tilde{X} \). Let us assume for a moment that \( X \) is matrix-variate Gaussian with row and column covariance \( Q^{-1} \) and \( R^{-1} \), respectively. Then, the GMD is like taking the SVD of the sphered data, as right and left multiplying by \( \tilde{Q} \) and \( \tilde{R} \) yields data with identity covariance. In other words, the GMD decorrelates the data so that the SVD with equally weighted errors is appropriate. While the GMD values are the singular values of this sphered data, the covariance is multiplied back into the GMD factors.

This relationship to the matrix-variate normal also begs the question, if one has data with row and column correlations, why not take the SVD of the two-way sphered or “whitened” data? This is inadvisable for several reasons. First, prewhitening the data and then taking the SVD yields matrix factors that are in the wrong basis and are thus uninterpretable in the original data space. Given this, one may advocate prewhitening, taking the SVD and then rewhitening, or in other words multiplying the correlation back into the SVD factors. This approach, however, is still problematic. In the special case where \( Q \) and \( R \) are of full rank, the GMD solution is exactly the same as this pre- and rewhitening approach. In the general case where \( Q \) and \( R \) are positive semidefinite, however, whitening cannot be directly performed as the covariances are not full rank. In the following section, we will show that the GMD solution can be computed without taking any matrix inverses, square roots or eigendecompositions and is thus computationally more attractive than naive whitening. Finally, as our eventual goal is to develop a framework for both structured data and two-way regularization, naive prewhitening and then rewhitening of the data would destroy any estimated sparsity or smoothness from regularization methods and is thus undesirable. Therefore, the GMD solution given in Theorem 1 is the mathematically correct way to “whiten” the data in the context of the SVD.

Next, we explore some of the mathematical properties of our GMD solution, \( \tilde{X} = U^*D^*(V^*)^T \) in the following corollaries:

**Corollary 1.** The GMD solution is the global minimizer of the GMD problem, (4).

**Corollary 2.** Assume that range(\( Q \)) \( \cap \) null(\( X \)) = \( \emptyset \) and range(\( R \)) \( \cap \) null(\( X \)) = \( \emptyset \). Then, there exists at most \( k = \min \{ \text{rank}(X), \text{rank}(Q), \text{rank}(R) \} \) nonzero GMD values and corresponding left and right GMD factors.

**Corollary 3.** With the assumptions and \( k \) defined as in Corollary 2, the rank-\( k \) GMD solution has zero reconstruction error in the Q, R-norm. If in addition, \( k = \text{rank}(X) \) and \( Q \) and \( R \) are full rank, then \( X \equiv U^*D^*(V^*)^T \), that is, the GMD is an exact matrix decomposition.

**Corollary 4.** The GMD values, \( D^* \), are unique up to multiplicity. If in addition, \( Q \) and \( R \) are full rank and the nonzero GMD values are distinct, then the GMD factors \( U^* \) and \( V^* \) corresponding to the nonzero GMD values are essentially unique (up to a sign change) and the GMD factorization is unique.

Some further comments on these results are warranted. In particular, Theorem 1 is straightforward and less interesting when \( Q \) and \( R \) are full rank, the case discussed in Escoufier (2006), Purdom (2007), and Holmes (2008). When the quadratic operators are positive semidefinite, however, the fact that a global minimizer to the GMD problem, which is nonconvex, that has a closed form can be obtained is not immediately clear. The result stems from the relation of the GMD to the SVD and the fact that the latter is a unique matrix decomposition in a lower dimensional subspace defined in Corollary 2. Also note that when \( Q \) and \( R \) are full rank, the GMD is an exact matrix decomposition; in the alternative scenario, we do not recover \( X \) exactly, but obtain zero reconstruction error in the Q, R-norm.

Finally, we note that when \( Q \) and \( R \) are rank deficient, there are possibly many optimal points for the GMD factors, although the resulting GMD solution is still a global minimizer of the GMD problem. In Section 2.5, we will see that permitting the quadratic operators to be positive semidefinite allows for much greater flexibility when modeling high-dimensional structured data.

### 2.4 GMD Algorithm

While the GMD solution given in Theorem 1 is conceptually simple, its computation, based on computing \( \tilde{Q}, \tilde{R} \), and the SVD of \( \tilde{X} \), is infeasible for massive datasets common in neuroimaging. We seek a method of obtaining the GMD solution that avoids computing and storing \( \tilde{Q} \) and \( \tilde{R} \) and thus permits our methods to be used with massive structured datasets.

**Algorithm 1 GMD Algorithm (Power Method)**

1. Let \( \tilde{X}^{(1)} \equiv X \) and initialize \( u_1 \) and \( v_1 \).
2. For \( k = 1, \ldots, K \):
   a. Repeat until convergence:
      - Set \( u_k = \frac{\tilde{X}^{(k-1)}Rv_k}{\| \tilde{X}^{(k-1)}Rv_k \|} \).
      - Set \( v_k = \frac{\tilde{X}^{(k-1)}Qu_k}{\| \tilde{X}^{(k-1)}Qu_k \|} \).
   b. Set \( d_k = u_k^T \tilde{Q} \tilde{X}^{(k)}Rv_k \).
   c. Set \( \tilde{X}^{(k+1)} = \tilde{X}^{(k)} - u_k \otimes d_k v_k^T \).
3. Return \( U^* = [u_1, \ldots, u_K], V^* = [v_1, \ldots, v_K], \) and \( D^* = \text{diag}(d_1, \ldots, d_K) \).

**Proposition 1.** If \( \hat{u} \) and \( \hat{v} \) are initialized such that \( \hat{u}^T \tilde{Q} \hat{u}^* \neq 0 \) and \( \hat{v}^T \tilde{R} \hat{v}^* \neq 0 \), then Algorithm 1 converges to the GMD solution given in Theorem 1, the global minimizer of the GMD
problem. If in addition, \(Q\) and \(R\) are full rank, then it converges to the unique global solution.

In Algorithm 1, we give a method of computing the components of the GMD solution that is a variation of the familiar power method for calculating the SVD (Golub and Van Loan 1996). Proposition 1 states that the GMD Algorithm based on this power method converges to the GMD solution. Notice that we do not need to calculate \(\hat{Q}\) and \(\hat{R}\) and thus, this algorithm is less computationally intensive for high-dimensional datasets than finding the solution via Theorem 1 or the computational approaches given in Escoufier (1987) for positive definite operators.

At this point, we pause to discuss the name of our matrix decomposition. Recall that the power method, or alternating least-square method, sequentially estimates \(u\) with \(v\) fixed and vice versa by solving least-square problems and then rescaling the estimates. Each step of the GMD algorithm, then, estimates the factors by solving the following generalized least-square problems and rescaling: \(|X - (d' u)' v |_R^2\) and \(|X^T - (d' v)' u |_R^2\). This is then the inspiration for the name of our matrix decomposition.

2.5 Generalized Principal Components

In this section, we show that the GMD can be used to perform GPCA. Note that this result was first shown in Escoufier (1977) for positive definite generalizing operators, but we review it here for completeness. The results in the previous section allow one to compute these GPCs for high-dimensional data when the quadratic operators may not be full rank and we wish to avoid computing SVDs and eigenvalue decompositions.

Recall that the SVD problem can be written as finding the linear combination of variables maximizing the sample variance such that these linear combinations are orthonormal. For GPCA, we seek to project the data onto the linear combination of variables such that the sample variance is maximized in a space that accounts for the structure or dependencies in the data. By transforming all inner product spaces to those induced by the \(Q\) -norm, we arrive at the following GPCA problem:

\[
\begin{align*}
\text{maximize}_v & \quad v^T R X^T Q X R v_k \\
\text{subject to} & \quad v^T R v_k = 1 \text{ and } v^T R v_{k'} = 0 \quad \forall \, k' < k. 
\end{align*}
\]

Notice that the loadings of the GPCs are orthogonal in the \(R\)-norm. The \(k\)th GPC is then given by \(z_k = X R v_k\).

Proposition 2. The solution to the \(k\)th Generalized principal component problem, (5), is given by the \(k\)th right GMD factor.

Corollary 5. The proportion of variance explained by the \(k\)th Generalized principal component is given by \(d_k^2 / ||X||_Q^2\).

Corollary 6. The Generalized PC solution, \(V\), explains the most variance with respect to the \(Q\), \(R\)-norm of any linear projection.

Just as the SVD can be used to find the principal components, the GMD can be used to find the generalized principal components. Also while PCA explains the most variance of any linear projection with respect to the Frobenius norm, so does GPCA with respect to the \(Q\), \(R\)-norm. Thus, GPCA gives an alternative way of explaining the variance-covariance in the data. Specifically, if the hybrid GMD model is assumed (2), then GPCA finds projections of the data that are maximally variant beyond the variance explained by the eigenvectors of \(Q\) and \(R\). This could be especially appealing with fMRI data, for example, where one may wish to find maximally variant projections that are separate from the covariance due to a spatio-temporal noise process.

2.6 Interpretations, Connections, and Quadratic Operators

Thus far, we have assumed that the quadratic operators, \(Q\) and \(R\), were fixed and known. One may ask why not estimate both the GMD components and the quadratic operators from the data? Specifically, one could envision optimizing the GMD problem (4) with respect to \(U\), \(V\), \(Q\), and \(R\). One could do this by iteratively solving the GMD problem and then estimating \(Q\) and \(R\) via methods discussed in Allen and Tibshirani (2010), for example. This, unfortunately, is not as appealing as it may seem. First, optimizing (4) with respect to the GMD factors and the quadratic operator results in a block-concave problem with many local optima that are strongly dependent on initial starting values. There is no guarantee that a desirable solution will be achieved. Second, this approach would be computationally prohibitive for massive datasets, requiring repeated eigenvalue decompositions. Perhaps more importantly, however, is that this approach leads to a blending of the eigenvectors of the signal and the eigenvectors of the noise. Specifically, consider the hybrid GMD model (2), which seeks to decompose the covariance into a noise component and a signal component. If both the eigenvectors of the noise and signal covariance are to be estimated simultaneously, this model is not identifiable, leading to uninterpretable GMD components. Thus, estimation of both the quadratic operators and the GMD components is a challenging problem that deserves thorough investigation. This then is beyond the scope of this initial article. Fortunately, however, for structured data, or data in which the row and/or column variables are associated with a specific location, there are many possible empirical choices for the quadratic operators following from the broader statistical literature of functional data analysis and spatial statistics.

First, from our GMD model (1) and the connection to the matrix-variate normal log-likelihood, we have that \(Q\) and \(R\) behave like inverse row and column covariances, respectively. For structured data, we may assume that the noise covariance is smooth with respect to the data structure. This is common, for example, with fMRI data where random field covariances are often used to model the spatial domain, and stochastic time series processes are used to model the temporal domain (Lazar 2008; Lindquist 2008). Thus, if the covariance is assumed to be smooth for structured data, one could take \(Q\) and \(R\) to be inverse smoothing operators. These assume that the noise covariance is smooth and seek to find empirical eigenvectors that are nonsmooth, being separate or orthogonal to the smooth noise. Some examples of inverse smoothing operators include the squared second and fourth difference matrix often used to penalize departures from smoothness in functional data analysis (Ramsay 2006), the inverse covariance operators from Gaussian Markov Random Fields (Rue and Held 2005), the inverse covariance matrices of time series autoregressive or moving average processes.
(Shaman 1969; Galbraith and Galbraith 1974), or Laplacians of graphs connecting variables based on distances (Merris 1994).

The quadratic operators used to generalize PCA can also be interpreted as weighting matrices that weight each matrix error differently based upon heteroscedastic row and column variabilities. Without assuming a parametric form, if \( \text{cov}(X_{ij}, X_{j'}) = Q_{ij} R_{j'} \), then one may wish to differentially weight the matrix errors according to covariance between the matrix elements. Thus, \( Q \) and \( R \) also behave as covariance matrices and in the context of structured data, as smoothing operators. The interpretation here is straightforward: smoothing operators up-weight variables that are close together on the structural manifold. Thus, GPCA seeks to find empirical eigenvectors that well approximate the data along the structural manifold. Again from the functional data analysis literature, there are many well-studied examples of smoothing operators including random field covariance operators, such as the exponential and Matern class, covariances of time series processes, and time series filters. Up-weighting matrix errors of the GMD according to distances on the structural manifold also have similarities to other nonlinear dimension reduction techniques, such as spectral clustering, local linear embedding, and local multi-dimensional scaling (Chen and Buja 2009).

While there are many connections between GPCA and the functional data analysis and multivariate analysis literature (many more than space permits us to illuminate here), the relationship of GPCA and two-way FPCA of Huang, Shen, and Buja (2009) is worth exploring further. They form row and column smoothing matrices, \( S_r = (I + \lambda_r \Omega_r)^{-1} \) and \( S_c = (I + \lambda_c \Omega_c)^{-1} \), respectively, with \( \Omega \) a penalty matrix related to the structure of the row and column variables. Then, two-way FPCA can be performed by two-way half smoothing: (1) take the SVD of \( X' = S_u^{1/2} S_c^{1/2} X' \), then (2) the FPCA factors are \( U = S_u^{1/2} U' \) and \( V = S_c^{1/2} V' \) (Huang, Shen, and Buja 2009). In other words, a half-smoother is applied to the data, the SVD is taken and another half-smoother is applied to the SVD factors. This is familiar to the GMD solution given in Theorem 1 where (essentially) the matrices \( Q^{1/2} \) and \( R^{1/2} \) are applied to the data and then \( Q^{-1/2} \) and \( R^{-1/2} \) are applied to the GMD factors. If \( Q \) and \( R \) are smoothing operators, then the GMD half-smooths the data, but inverse half-smooths the factors. If \( Q \) and \( R \) are inverse smoothing operators, then the opposite is true. Thus, dissimilar to two-way FPCA that performs two smoothing operations, the GMD essentially applies one smoothing operation and one inverse smoothing operation. This distinction is important as two-way FPCA assumes that the factors are smooth, but makes no assumption about the noise. On the other hand, the GMD assumes that the noise is smooth, and seeks to find factors that are separate from the smooth noise.

Our brief discussion has illustrated that for structured data, there are many possible empirical choices for \( Q \) and \( R \), such as smoothing or inverse smoothing operators. While this article introduces the general methodology for GPCA and two-way regularized GPCA, in practice, we currently advocate employing these as empirical methods where \( Q \) and \( R \) are fixed according to the structure of the data. In doing so, one can borrow the theory and best practices from related functional data analysis methods.

3. GENERALIZED PENALIZED MATRIX FACTORIZATION

With high-dimensional data, many have advocated regularizing principal components by either automatically selecting relevant features as with SPCA or by smoothing the factors as with FPCA (Silverman 1996; Jolliffe, Trendafilov, and Uddin 2003; Zou, Hastie, and Tibshirani 2006; Shen and Huang 2008; Huang, Shen, and Buja 2009; Witten, Tibshirani, and Hastie 2009; Lee et al. 2010). Regularized PCA can be important for massive structured data as well. Consider spatio-temporal fMRI data, for example, where many spatial locations or voxels in the brain are inactive and the time courses are often extremely noisy. Automatic feature selection of relevant voxels and smoothing of the time series in the context of PCA for structured data would thus be an important contribution. In this section, we seek a framework for regularizing the factors of the GMD by placing penalties on each factor. In developing this framework, we reveal an important result demonstrating the general class of penalties that can be placed on matrix factors that are to be estimated via deflation: the penalties must be norms or seminorms.

3.1 GPMF Problem

Recently, many have suggested regularizing the factors of the SVD by forming two-way penalized regression problems (Huang, Shen, and Buja 2009; Witten, Tibshirani, and Hastie 2009; Lee et al. 2010). We briefly review these existing approaches to understand how to frame a problem that allows us to place general sparse or smooth penalties on the GMD factors.

We compare the optimization problems of these three approaches for computing a single-factor two-way regularized matrix factorization:

\[
\begin{align*}
\text{Witten, Tibshirani, and Hastie (2009):} \\
\text{maximize}_{u,v} & \quad u^T X v \text{ subject to } u^T u \leq 1, \quad v^T v \leq 1, \\
& \quad P_1(u) \leq c_1, \quad \text{and } P_2(v) \leq c_2. \\
\text{Huang, Shen, and Buja (2009):} \\
\text{maximize}_{u,v} & \quad u^T X v - \frac{1}{2} P_1(u) P_2(v). \\
\text{Lee et al. (2010):} \\
\text{maximize}_{u,v} & \quad u^T X v - \frac{1}{2} u^T u v^T v - \frac{\lambda_u}{2} P_1(u) - \frac{\lambda_v}{2} P_2(v). 
\end{align*}
\]

Here, \( c_1 \) and \( c_2 \) are constants, \( P_1(u) \) and \( P_2(v) \) are penalty functions, and \( \lambda, \lambda_u, \) and \( \lambda_v \) are penalty parameters. These optimization problems are attractive as they are bi-concave in \( u \) and \( v \), meaning that they are concave in \( u \) with \( v \) fixed and vice versa. Thus, a simple maximization strategy of iteratively maximizing with respect to \( u \) and \( v \) results in a monotonic ascent algorithm converging to a local maximum.

These methods, however, differ in the types of penalties that can be employed and their scaling of the matrix factors. Witten, Tibshirani, and Hastie (2009) explicitly restricted the norms of the factors, while the constraints in the method of Huang, Shen, and Buja (2009) are implicit because of the types of functional data penalties employed. Thus, for these methods, as the constants \( c_1 \) and \( c_2 \) or the penalty parameter, \( \lambda \), approach zero, the SVD is returned. This is not the case, however, for problem in Lee et al. (2010) where the factors are not constrained in the optimization problem, although they are later scaled in.
their algorithm. Also, restricting the scale of the factors avoids possible numerical instabilities when employing the iterative estimation algorithm (see especially the supplementary materials of Huang, Shen, and Buja 2009). Thus, we prefer the former two approaches for these reasons. In Witten, Tibshirani, and Hastie (2009), however, only the lasso and fused lasso penalty functions are employed and it is unclear whether other penalties may be used in their framework. Huang, Shen, and Buja (2009), on the other hand, limited their consideration to quadratic functional data penalties, and their optimization problem does not implicitly scale the factors for other types of penalties.

As we wish to employ general penalties, specifically sparse and smooth penalties, on the matrix factors, we adopt an optimization problem that leads to simple solutions for the factors with a wide class of penalties, as discussed in the subsequent section. We then define the single-factor generalized penalized matrix factorization (GPMF) problem as the following:

maximize $v, u$ $u^T Q X R v - \lambda_u P_1(v) - \lambda_u P_2(u)$
subject to $u^T Q u \leq 1$ and $v^T R v \leq 1, \quad (6)$

where, as before, $\lambda_u$ and $\lambda_u$ are penalty parameters and $P_1()$ and $P_2()$ are penalty functions. Note that if $\lambda_u = \lambda_v = 0$, then the left and right GMD factors can be found from (6), as desired. Strictly speaking, this problem is the Lagrangian of the problem introduced in Witten, Tibshirani, and Hastie (2009), keeping in mind that we should interpret the inner products as those induced by the $Q, R$-norm. As we will see in Theorem 2 in the next section, this problem is tractable for many common choices of penalty functions and avoids the scaling problems of other approaches.

3.2 GPMF Solution

We solve the GPMF criterion, (6), via block coordinate ascent by alternating maximizing with respect to $u$ then $v$. Note that if $\lambda_u = 0$ or $\lambda_v = 0$, then the coordinate update for $u$ or $v$ is given by the GMD updates in Step (b) (i) of the GMD Algorithm. Consider the following result:

Theorem 2. Assume that $P_1()$ and $P_2()$ are convex and homogenous of order one: $P(c x) = c P(x) \forall c > 0$. Let $u$ be fixed at $u'$ or $v$ be fixed at $v'$ such that $u'^T Q X \neq 0$ or $v'^T X^T R \neq 0$. Then, the coordinate updates, $u^*$ and $v^*$, maximizing the single-factor GPMF problem, (6), are given by the following: Let $\tilde{v} = \text{argmin}_v \{ \frac{1}{2} ||X^T Qu - v||_Q^2 + \lambda_u P_1(v) \}$ and $\tilde{u} = \text{argmin}_u \{ \frac{1}{2} ||X R v - u||_Q^2 + \lambda_u P_2(u) \}$. Then

$$v^* = \begin{cases} \tilde{v} / ||\tilde{v}||_R & \text{if } ||\tilde{v}||_R > 0 \\ 0 & \text{otherwise} \end{cases}$$

$$u^* = \begin{cases} \tilde{u} / ||\tilde{u}||_Q & \text{if } ||\tilde{u}||_Q > 0 \\ 0 & \text{otherwise} \end{cases}$$

Theorem 2 states that for penalty functions that are convex and homogenous of order one, the coordinate updates giving the single-factor GPMF solution can be obtained by a generalized penalized regression problem. Note that penalty functions that are norms or seminorms are necessarily convex and homogenous of order one. This class of functions includes common penalties, such as the lasso (Tibshirani 1996), group lasso (Yuan and Lin 2006), fused lasso (Tibshirani et al. 2005), and the generalized lasso (Tibshirani and Taylor 2011). Importantly, these do not include the ridge penalty, elastic net, concave penalties such as SCAD, and quadratic smoothing penalties commonly used in functional data analysis. Many of the penalized regression solutions for these penalties, however, can be approximated by penalties that are norms or seminorms. For instance, to mimic the effect of a given quadratic penalty, one may use the square root of this quadratic penalty. Similarly, the natural majorization–minimization algorithms for SCAD-type penalties (Fan and Li 2001) involve convex piecewise linear majorizations of the penalties that satisfy the conditions of Theorem 2. Thus, our single-factor GPMF problem both avoids the complicated scaling problems of some existing two-way regularized matrix factorizations, and still permits a wide class of penalties to be used within our framework.

Following the structure of the GMD Algorithm, the multifactor GPMF can be computed via the power method framework. That is, the GPMF algorithm replaces Steps (b) (i) of the GMD Algorithm with the single-factor GPMF updates given in Theorem 2. This follows the same approach as that of other two-way matrix factorizations (Huang, Shen, and Buja 2009; Witten, Tibshirani, and Hastie 2009; Lee et al. 2010). Notice that unlike the GMD, subsequent factors computed via this greedy deflation approach will not be orthogonal in the $Q, R$-norm to previously computed components. Many have noted in the SPCA literature, for example, that orthogonality of the sparse components is not necessarily warranted (Jolliffe, Trendafilov, and Uddin 2003; Zou, Hastie, and Tibshirani 2006; Shen and Huang 2008). If only one factor is penalized, however, orthogonality can be enforced in subsequent components via a Gram–Schmidt scheme (Golub and Van Loan 1996).

In the following section, we give methods for obtaining the single-factor GPMF updates for sparse penalty types; the solution for smooth penalties is given in online supplementary materials. Note that many other penalties may be employed with our methods beyond the ones discussed in this work. As the single-factor GPMF problem is symmetric in $u$ and $v$, we solve the $R$-norm penalized regression problem in $v$, noting that the solutions are analogous for the $Q$-norm penalized problem in $u$.

3.2.1 Sparsity: Lasso and Related Penalties. With high-dimensional data, sparsity in the factors or principal components yields greater interpretability and, in some cases have better properties than unpenalized principal components (Jolliffe, Trendafilov, and Uddin 2003; Johnstone and Lu 2009). With neuroimaging data, sparsity in the factors associated with the voxels is particularly warranted as in most cases relatively few brain regions are expected to truly contribute to the signal. Hence, we consider our GPMF problem, (6), with the lasso (Tibshirani 1996), or $\ell_1$-norm penalty, commonly used to encourage sparsity.

The penalized regression problem given in Theorem 2 can be written as a lasso problem: $\frac{1}{2} ||X^T Qu - v||_Q^2 + \lambda_v ||v||_1$. If $R = I$, then the solution for $\tilde{v}$ is obtained by simply applying the soft thresholding operator: $\tilde{v} = S(X^T Qu, \lambda)$, where $S(x, \lambda) = \text{sign}(x)(|x| - \lambda)^+$. (Tibshirani 1996). When $R \neq I$, the solution is not as simple:
Claim 1. If $R$ is diagonal with strictly positive diagonal entries, then $\tilde{v}$ minimizing the $R$-norm lasso problem is given by $\tilde{v} = S(X^T Qu, \lambda, R^{-1} 1(p))$.

Claim 2. The solution, $\tilde{v}$, that minimizes the $R$-norm lasso problem can be obtained by iterative coordinate-descent where the solution for each coordinate $\tilde{v}_j$ is given by $\tilde{v}_j = \frac{1}{p_j} S(R_{1j} X^T Qu - R_{1j}, 1_j, \lambda, R_{1j})$, with the subscript $R_{1j}$ denoting the row elements associated with column $j$ of $R$.

Claim 1 states that when $R$ is diagonal, the solution for $v$ can be obtained by soft thresholding the elements of $y$ by a vector penalty parameter. For general $R$, Claim 2 gives that we can use coordinate-descent to find the solution without having to compute $\tilde{R}$. Thus, the sparse single-factor GPMF solution can be calculated in a computationally efficient manner. One may further improve the speed of coordinate-descent by employing warm starts and iterating over active coordinates as described in Friedman, Hastie, and Tibshirani (2010).

We have discussed the details of computing the GPMF factors for the lasso penalty, and similar techniques can be used to efficiently compute the solution for the group lasso (Yuan and Lin 2006), fused lasso (Tibshirani et al. 2005), generalized lasso (Tibshirani and Taylor 2011), and other sparse convex penalties. As mentioned, we limit our class of penalty functions to those that are norms or seminorms, which does not include popular concave penalties such as the SCAD penalty (Fan and Li 2001). As mentioned, we limit our class of penalty functions to those that are norms or seminorms, which does not include popular concave penalties such as the SCAD penalty (Fan and Li 2001). As mentioned above, these penalties, however, can still be used in our framework as concave penalized regression problems can be solved via iterative weighted lasso problems (Mazumder, Friedman, and Hastie 2011). Thus, our GPMF framework can be used with a wide range of penalties to obtain sparsity in the factors.

Finally, we note that as our GMD Algorithm can be used to perform GPCA, the sparse GPMF framework can be used to find sparse generalized principal components by setting $\lambda_u = 0$ in (6). This is an approach well studied in the SPCA literature (Shen and Huang 2008; Witten, Tibshirani, and Hastie 2009; Lee et al. 2010).

3.3 Selecting Penalty Parameters and Variance Explained

When applying the GPMF and Sparse or Functional GPCA to real structured data, there are two practical considerations that must be addressed: (a) the number of factors, $K$, to extract and (b) the choice of penalty parameters, $\lambda_u$ and $\lambda_v$, controlling the amount of regularization. Careful examination of the former is beyond the scope of this article. For classical PCA, however, several methods exist for selecting the number of factors (Buja and Eyuboglu 1992; Troyanskaya et al. 2001; Owen and Perry 2009). Extending the imputation-based methods of Troyanskaya et al. (2001) for GPCA methods is straightforward; we believe that extensions of Buja and Eyuboglu (1992) and Owen and Perry (2009) are also possible in our framework. A related issue for selecting the number of factors is the amount of variance explained. While this is a simple calculation for GPCA (see Corollary 5), this is not as straightforward for two-way regularized GPCA as the factors are no longer orthonormal in the $Q, R$-norm. Thus one must project out the effect of the previous factors to compute the cumulative variance explained by the first several factors.

**Proposition 3.** Let $U_k = [u_1 \ldots u_k]$ and $V_k = [v_1 \ldots v_k]$ and define $P^{(U)}_k = U_k (U_k^T Qu_k)^{-1} U_k^T$ and $P^{(V)}_k = V_k (V_k^T Rv_k)^{-1} V_k^T$ yielding $X_k = P^{(U)}_k QXRP^{(V)}_k$.

Then, the cumulative proportion of variance explained by the first $k$ regularized GPCs is given by $tr(QX_kRX_k^T)/tr(QXRX^T)$.

Note that since the deflation-based GPMF algorithm is greedy, the components are not necessarily ordered in terms of variance explained as those of the GMD. Also, note that Proposition 3 gives the cumulative proportion of variance explained in the $Q, R$-norm, which is different from the typically reported variance explained for PCA methods based on the Frobenius norm. If one wishes to calculate the variance explained by GPCs or regularized GPCs in the Frobenius norm, one could simply replace $Q$ and $R$ with $I$ in Proposition 3. (Conversely, one could calculate the variance explained by PCs or regularized PCs in the $Q, R$-norm by Proposition 3 as well.)

When applying the GPMF, the penalty parameters $\lambda_u$ and $\lambda_v$ control the amount of sparsity or smoothness in the estimated factors. We seek a data-driven way to estimate these penalty parameters. Many have proposed cross-validation approaches for the SVD (Troyanskaya et al. 2001; Owen and Perry 2009) or nested generalized cross-validation or Bayesian Information Criterion (BIC) approaches (Huang, Shen, and Buja 2009; Lee et al. 2010). While all of these methods are appropriate for our models as well, we propose an extension of the BIC selection method of Lee et al. (2010) appropriate for the $Q, R$-norm.

**Claim 3.** The BIC selection criterion for the GPMF factor $v$ with $u$ and $d$ fixed at $u^*$ and $d^*$, respectively, is given by the following: $\text{BIC}(\lambda_v) = \log(||X - d^* u^*||_F^2) + \frac{\log(np)}{np} \dim(v)$. The BIC criterion for the other factor $u$ is analogous. Here, $\hat{d}(\lambda_v)$ is an estimate of the degrees of freedom for the particular penalty employed. For the lasso penalty, for example, $\hat{d}(\lambda_v) = ||\tilde{v}||_1$ (Zou et al. 2007). Expressions for the degrees of freedom of other penalty functions are given in Kato (2009) and Tibshirani and Taylor (2011).

Given this model selection criterion, one can select the optimal penalty parameter at each iteration of the factors for the GPMF algorithm as in Lee et al. (2010), or use the criterion to select parameters after the iterative algorithm has converged. In our experiments, we found that both of these approaches perform similarly, but the latter is more numerically stable. The performance of this method is investigated through simulation studies in Section 4.1. Finally, we note that since the GPMF only converges to a local optimum, the solution achieved is highly dependent on the initial starting point. Thus, we use the GMD solution to initialize all our algorithms, an approach similar to related methods that only achieve a local optimum (Witten, Tibshirani, and Hastie 2009; Lee et al. 2010).

4. RESULTS

4.1 Simulations

We simulate data from the GMD model given in (1): $X = X^* + E$ for low-rank signal $X^* = \sum_{k=1}^K \phi_k v_k v_k^T$ with $\phi_k$ giving
the signal magnitude and correlated noise $E \sim N(0, \Sigma \otimes \Delta)$ for row covariance $\Sigma$ and column covariance $\Delta$. Our simulation is inspired by neuroimaging data, with the true spatial factors $V$ and temporal factors $U$ as shown in the top panel of Figure 1. Specifically, $V \in \mathbb{R}^{25 \times 25}$ is structured as a $25 \times 25$ image with discrete (sparse) regions of interest, and $U \in \mathbb{R}^{200 \times 2}$ is taken as a sinusoidal and Gaussian-modulated sinusoidal temporal activation pattern. The signal magnitude is given by $\phi_1 = 2n$ and $\phi_2 = n$. The spatial and temporal covariances are taken as Toeplitz (or autoregressive) matrices directly related to the data structure: $\Sigma_{ij} = \theta^{|i-j|}$ and $\Delta_{ij} = \theta^{|d_{ij}|}$, where $d_{ij}$ is the Chebychev (taxi-cab) distance between variables on the two-dimensional grid. Here, $0 \leq \theta < 1$ controls the amount of spatio-temporal correlation in the data. In our simulations, we would like to evaluate the effect of noise correlations on PCA and GPCA; thus, we change the amount of correlation but keep the signal-to-noise ratio (SNR) fixed so that all results are comparable. Specifically, the SNR for this model is given by $E[\text{tr}(X^T X)]/E[\text{tr}(\Sigma)] = E[\sum_{k=1}^K \phi_k^2 \mathbf{u}_k^T \mathbf{u}_k \mathbf{v}_k^T \mathbf{v}_k]/E[\text{tr}(\Sigma)]$ (Gupta and Nagar 1999).

In Table 1, we investigate the effect of noise correlations on the performance of PCA and GPCA measured in terms of subspace recovery and low-rank matrix recovery. Subspace recovery, denoted in the table as $\angle U$ is given by $1 - \cos \angle (\hat{U}^*, \hat{U})$, where $\angle$ is the principal angle between the subspaces; thus, smaller values of $\angle U$ indicate better subspace recovery. Low-rank matrix recovery, denoted in the table as $\text{RMSE}(X)$, is given by $\|X - \hat{X}\|_F/\sqrt{n}$.

Results in Table 1 demonstrate that while PCA is able to recover the true signal when the noise is independent or exhibits minor correlations, the method begins to perform poorly with moderate and especially large amounts of correlated noise. On the other hand, GPCA where the quadratic operators are set to those of the generating model, $Q = \Sigma^{-1}$ and $R = \Delta^{-1}$, does not show a decline in performance with increasing amounts of correlated noise.

While Table 1 investigates GPCA with the quadratic operators determined by the true population covariances, these may not be known in practice. Thus, we test our methods for quadratic operators taken as standard smoothing or inverse smoothing operators used throughout functional data analysis (Ramsay 2006). Specifically, we employ a standard exponential smoothing matrix, $S_{ij} = \exp^{-d_{ij}/\sigma}$ for $\sigma = 2$ and $d_{ij}$ the Chebychev distance between variables on a grid, or an inverse smoothing matrix $Q = \Sigma^{-1}$ and $R = \Delta^{-1}$.

Table 1. Spatio-temporal simulation results averaged over 50 replicates for varying amounts of correlation given by $\theta$. Here, $Q = \Sigma^{-1}$ and $R = \Delta^{-1}$, the true generating inverse covariances for GPCA. Best performing methods are boldfaced, indicating that PCA fails to recover the true signal with moderate to large amounts of correlation in the noise.

<table>
<thead>
<tr>
<th>$\angle \theta$</th>
<th>PCA</th>
<th>GPCA</th>
<th>$\angle \Sigma$</th>
<th>PCA</th>
<th>GPCA</th>
<th>RMSE($X^*$)</th>
<th>PCA</th>
<th>GPCA</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\theta = 0$</td>
<td>0.019 (0.000)</td>
<td>0.019 (0.000)</td>
<td>0.054 (0.001)</td>
<td>0.054 (0.001)</td>
<td>1.775 (0.000)</td>
<td>1.775 (0.000)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\theta = 0.3$</td>
<td>0.038 (0.001)</td>
<td>0.045 (0.001)</td>
<td>0.089 (0.001)</td>
<td>0.104 (0.001)</td>
<td>1.781 (0.001)</td>
<td>1.791 (0.001)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\theta = 0.6$</td>
<td>0.103 (0.003)</td>
<td>0.051 (0.001)</td>
<td>0.181 (0.004)</td>
<td>0.203 (0.004)</td>
<td>1.800 (0.002)</td>
<td>1.821 (0.002)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\theta = 0.7$</td>
<td>0.189 (0.008)</td>
<td>0.034 (0.001)</td>
<td>0.276 (0.008)</td>
<td>0.240 (0.004)</td>
<td>1.831 (0.003)</td>
<td>1.814 (0.002)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\theta = 0.8$</td>
<td>0.568 (0.027)</td>
<td>0.017 (0.000)</td>
<td>0.664 (0.024)</td>
<td>0.281 (0.009)</td>
<td>1.965 (0.007)</td>
<td>1.802 (0.003)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\theta = 0.9$</td>
<td>0.825 (0.018)</td>
<td>0.003 (0.000)</td>
<td>0.853 (0.014)</td>
<td>0.302 (0.012)</td>
<td>2.135 (0.006)</td>
<td>1.788 (0.003)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Sparse GPCA (SGPCA) were implemented via (6) with sparsity on $\mathbf{V}$; thus for SPCA, this is equivalent to the method of Shen and Huang (2008). All methods used the BIC criterion as described in Section 3.3 to compute the optimal regularization parameter. As the factors recovered by SPCA are invariant to rotations, we can present signal recovery results in terms of the angle for each single-rank factor, $\angle \mathbf{u}_i = 1 - \cos |\hat{\mathbf{u}}_i^T \mathbf{u}_i|$: we also give the true and false positive rates, denoted as $\text{TP}$ and $\text{FP}$, respectively, for the sparse recovery of the spatial factors. Results demonstrate that while SPCA performs well for small amounts of correlation in the noise, its performance deteriorates with larger amounts of correlation to the point where the sparse signal is completely missed. An example of this phenomenon is shown in Figure 1. Sparse GPCA, on the other hand, is still able to recover sparse spatial signals for high correlation in the noise. This is especially true when a temporal smoothing operator and spatial inverse smoothing operator are employed, paralleling their strong performance shown in Table 3. Notice, however, that SGPCA exhibits a higher false positive rate than SPCA for uncorrelated data. This occurs as the $\mathbf{Q}, \mathbf{R}$ norm unequally weights the factor projections by borrowing strength from neighboring variables, resulting in false positives that tend to be spatially contiguous, for example. The slightly higher false positives,

<table>
<thead>
<tr>
<th>$\theta$</th>
<th>SPCA</th>
<th>SGPCA</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\angle \mathbf{u}_1$</td>
<td>0.032 (0.001)</td>
<td>0.041 (0.001)</td>
</tr>
<tr>
<td>$\angle \mathbf{v}_1$</td>
<td>0.014 (0.001)</td>
<td>0.119 (0.011)</td>
</tr>
<tr>
<td>$\angle \mathbf{u}_2$</td>
<td>0.057 (0.001)</td>
<td>0.107 (0.009)</td>
</tr>
<tr>
<td>$\angle \mathbf{v}_2$</td>
<td>0.025 (0.001)</td>
<td>0.260 (0.016)</td>
</tr>
<tr>
<td>$\text{TP} \mathbf{v}_1$</td>
<td>1.000 (0.000)</td>
<td>0.934 (0.011)</td>
</tr>
<tr>
<td>$\text{FP} \mathbf{v}_1$</td>
<td>0.001 (0.000)</td>
<td>0.057 (0.005)</td>
</tr>
<tr>
<td>$\text{TP} \mathbf{v}_2$</td>
<td>1.000 (0.000)</td>
<td>0.771 (0.023)</td>
</tr>
<tr>
<td>$\text{FP} \mathbf{v}_2$</td>
<td>0.001 (0.000)</td>
<td>0.017 (0.002)</td>
</tr>
<tr>
<td>$\text{RMSE}(\mathbf{X}^*)$</td>
<td>1.434 (0.001)</td>
<td>1.549 (0.005)</td>
</tr>
</tbody>
</table>

For higher amounts of correlation, Sparse GPCA offers substantial improvements in terms of both feature selection and signal recovery.
Table 3. Spatio-temporal simulation results averaged over 50 replicates comparing PCA with GPCA when various quadratic operators are employed. Results indicate that standard smoothing and inverse smoothing operators from functional data analysis perform nearly as well for GPCA as the true population inverse covariances.

<table>
<thead>
<tr>
<th></th>
<th>PCA</th>
<th>GPCA, $Q = \mathbf{\Sigma}^{-1}$, $\mathbf{R} = \mathbf{\Delta}^{-1}$</th>
<th>GPCA, $Q = \mathbf{S}$, $\mathbf{R} = \mathbf{S}$</th>
<th>GPCA, $Q = \mathbf{L}$, $\mathbf{R} = \mathbf{L}$</th>
<th>GPCA, $Q = \mathbf{S}$, $\mathbf{R} = \mathbf{L}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\theta = 0.3$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\text{var}_{1,1}$</td>
<td>0.214 (0.001)</td>
<td>0.210 (0.001)</td>
<td>0.212 (0.001)</td>
<td>0.0283 (0.000)</td>
<td>0.204 (0.001)</td>
</tr>
<tr>
<td>$\text{var}_{Q,R}$</td>
<td>0.069 (0.000)</td>
<td>0.071 (0.000)</td>
<td>0.736 (0.003)</td>
<td>0.046 (0.000)</td>
<td>0.232 (0.001)</td>
</tr>
<tr>
<td>$\angle \mathbf{U}$</td>
<td>0.038 (0.000)</td>
<td>0.046 (0.001)</td>
<td>0.055 (0.001)</td>
<td>0.948 (0.005)</td>
<td>0.036 (0.001)</td>
</tr>
<tr>
<td>$\angle \mathbf{V}$</td>
<td>0.089 (0.001)</td>
<td>0.103 (0.001)</td>
<td>0.088 (0.001)</td>
<td>0.968 (0.004)</td>
<td>0.092 (0.001)</td>
</tr>
<tr>
<td>RMSE($\mathbf{X}^*$)</td>
<td>1.779 (0.001)</td>
<td>1.790 (0.001)</td>
<td>1.782 (0.001)</td>
<td>2.124 (0.000)</td>
<td>1.783 (0.001)</td>
</tr>
<tr>
<td>$\theta = 0.7$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\text{var}_{1,1}$</td>
<td>0.290 (0.003)</td>
<td>0.234 (0.002)</td>
<td>0.284 (0.003)</td>
<td>0.162 (0.003)</td>
<td>0.248 (0.003)</td>
</tr>
<tr>
<td>$\text{var}_{Q,R}$</td>
<td>0.031 (0.000)</td>
<td>0.052 (0.000)</td>
<td>0.445 (0.005)</td>
<td>0.064 (0.000)</td>
<td>0.289 (0.001)</td>
</tr>
<tr>
<td>$\angle \mathbf{U}$</td>
<td>0.214 (0.015)</td>
<td>0.033 (0.001)</td>
<td>0.399 (0.025)</td>
<td>0.866 (0.014)</td>
<td>0.040 (0.001)</td>
</tr>
<tr>
<td>$\angle \mathbf{V}$</td>
<td>0.310 (0.015)</td>
<td>0.261 (0.006)</td>
<td>0.405 (0.024)</td>
<td>0.914 (0.011)</td>
<td>0.219 (0.005)</td>
</tr>
<tr>
<td>RMSE($\mathbf{X}^*$)</td>
<td>1.835 (0.004)</td>
<td>1.815 (0.002)</td>
<td>1.883 (0.005)</td>
<td>1.967 (0.003)</td>
<td>1.789 (0.002)</td>
</tr>
<tr>
<td>$\theta = 0.9$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\text{var}_{1,1}$</td>
<td>0.707 (0.012)</td>
<td>0.250 (0.005)</td>
<td>0.705 (0.012)</td>
<td>0.204 (0.004)</td>
<td>0.256 (0.006)</td>
</tr>
<tr>
<td>$\text{var}_{Q,R}$</td>
<td>0.014 (0.001)</td>
<td>0.236 (0.000)</td>
<td>0.848 (0.014)</td>
<td>0.220 (0.001)</td>
<td>0.474 (0.003)</td>
</tr>
<tr>
<td>$\angle \mathbf{U}$</td>
<td>0.843 (0.015)</td>
<td>0.003 (0.000)</td>
<td>0.856 (0.014)</td>
<td>0.536 (0.037)</td>
<td>0.048 (0.002)</td>
</tr>
<tr>
<td>$\angle \mathbf{V}$</td>
<td>0.891 (0.011)</td>
<td>0.313 (0.012)</td>
<td>0.891 (0.012)</td>
<td>0.842 (0.014)</td>
<td>0.343 (0.014)</td>
</tr>
<tr>
<td>RMSE($\mathbf{X}^*$)</td>
<td>2.130 (0.006)</td>
<td>1.790 (0.003)</td>
<td>2.144 (0.006)</td>
<td>1.902 (0.002)</td>
<td>1.802 (0.003)</td>
</tr>
</tbody>
</table>

However, seem a small price to pay when there are strong noise correlations, a setting where PCA and SPCA fail to recover the true signal.

Overall, we have first shown that PCA methods perform poorly in terms of subspace recovery, low-rank recovery, and feature selection for data in which there are strong noise dependencies. Our GPCA and Sparse GPCA methods instead perform well in these settings where PCA methods fail.

4.2 Functional MRI Analysis

A central goal of many fMRI studies and especially resting-state studies is to identify activated brain regions exhibiting similar temporal hemodynamic responses, sometimes referred to as functional connectivity (Friston et al. 1993). To this end, unsupervised multivariate techniques, predominately independent components analysis (ICA), are applied to a matrix of vectorized voxels by time points for each subject (McKeown et al. 1998) or concatenated subjects in group analyses (Calhoun, Liu, and Adall 2009). As it is typically applied by neuroimagers, ICA suffers from many weaknesses including (i) factors that are not sub-settable, (ii) unordered factors, (iii) and no automatic methods for feature selection (simple thresholding of the solution is instead employed) (Beckmann 2012). While PCA and SPCA methods correct these deficiencies, many have noted that these methods lack interpretability for fMRI data because the factors tend to be a mix of signal and spatio-temporal noise, and signals of interest are typically blended across multiple components (Andersen, Gash, and Avison 1999; Friston et al. 1999; Thirion and Fuageras 2003; Viviani, Gron, and Spitzer 2005; Rogers et al. 2007). Additionally, some argue that the Gaussian distributional assumption relating to the probabilistic interpretation of PCA is not ideal for neuroimaging data (Calhoun, Liu, and Adall 2009). Through an example on whole brain fMRI data, we investigate the performance of PCA and SPCA methods, state-of-the-art ICA methods, and compare these to our GPCA and Sparse GPCA approaches.

For our functional MRI example, we use a well-studied, publicly available fMRI dataset where subjects were shown images and read audible sentences related to these images, a dataset that refers to as the “StarPlus” data (Mitchell et al. 2004). We first study data for one subject, subject number 04847, which consists of 4698 voxels ($64 \times 64 \times 8$ images with masking) measured for repeated tasks, each lasting for 27 s, yielding 54–55 time points. Equivalent tasks, 20 in total, corresponding to sentence and image agreement were analyzed. The data were preprocessed by standardizing each voxel within each task segment. Rearranging the data yields a $4698 \times 1098$ matrix to which we applied our dimension reduction techniques. (Note that spatial smoothing is a common preprocessing technique for fMRI data (Lazar 2008); as smoothing induces further correlation between neighboring voxels, the precise scenario where PCA methods perform poorly, we omitted this preprocessing step so as to not give an unfair advantage to our GPCA methods.) As the spatio-temporal nature of the noise in fMRI data has been carefully studied (Friston et al. 1995; Lazar 2008; Lindquist 2008), we take $\mathbf{Q}$ and $\mathbf{R}$ to be inverse smoothing and smoothing matrices directly related to the structure of three-dimensional images taken over time. Following from the promising results of our simulation study, we take $\mathbf{R}$ to be a Laplacian matrix of the graph connecting nearest neighbors in the three-dimensional voxel space, and $\mathbf{Q}$ to be an exponential smoothing matrix $\mathbf{Q}_{ij} = \exp(-|i - j|^2/\sigma)$, where $\sigma = 10$ was chosen to maximize the variance explained by the first GPC. We applied PCA, GPCA, ICA via the FastICA algorithm (Hyvarinen and Oja 2000), and SPCA and Sparse GPCA using the formulation in (6) with the BIC to select the optimal sparsity.
Figure 2. Eight axial slices with corresponding time series for the first three PCs of the Starplus fMRI data for PCA (top left), rank three ICA (bottom left), Generalized PCA (top right), and Sparse Generalized PCA (bottom right). Dotted red vertical lines in the time series denote the beginning and end of each task where an image was shown accompanied by an audible sentence that corresponded to the image. The first three spatial PCs of PCA and the first and third of ICA exhibit large patterns of spatial noise where the corresponding time series appear to be artifacts or scanner noise, unrelated to the experiment. The temporal PCs of GPCA and Sparse GPCA, on the other hand, exhibit a clear pattern with respect to the experimental task. The spatial PCs of GPCA, however, are somewhat noisy, whereas when sparsity is encouraged, the spatial PCs of Sparse GPCA illustrate clear regions of activation related to the experimental task.

Results for the first three spatial and temporal factors estimated by PCA, ICA (a rank three estimate ordered according to variance explained), GPCA, and Sparse GPCA are shown in Figure 2. The spatial factors are illustrated via the eight axial slices with gray matter masks and the temporal components are illustrated with vertical lines denoting the onset of each new task segment. Notice that the PCA factors seem to be a blend of spatial noise with the corresponding time series appearing to be artifacts or scanner noise, unrelated to the experimental task; these results are consistent with findings in the fMRI literature. The ICA method separates these noise components and signal components, yielding more interpretable results for the second ICA factor. Our GPCA and Sparse GPCA methods, on the other hand, find factors that are clearly correlated with the experimental task. In particular, the first two SGPCs show bilateral occipital, left-lateralized inferior temporal, and inferior frontal activity characteristic of the well-known “ventral stream” pathway recruited during object identification tasks (Pennick and Kana 2012). Additional fMRI results are given in online supplementary materials. These results further illustrate the benefits of our methods. In particular, only one ICA factor exhibits a clear temporal trend related to the experimental task, with the other top seven factors appearing to be noise. Out of the top eight GPCs, six factors appear correlated with the experimental task, with two appearing to be periodic scanner noise. Our GPCA and Sparse GPCA also explain a much greater proportion of variance than alternative PCA methods, Figure 3. (Note that as PCA is first used by ICA methods to reduce the dimension, ICA factors always explain less than or equal to the amount of variance of PCA factors.) The first eight GPCs and SGPCs explain more variance in the Q, R-norm than all 25 of the first PCs or SPCs.

Next, we use our GPCA and Sparse GPCA methods to perform a group analyses on all six subjects of the Starplus dataset. The first temporal group Sparse GPC is illustrated in Figure 4. This analysis was performed by concatenating the voxels of each of the six subjects resulting in a 29,493 × 1098 matrix, an approach introduced by Calhoun et al. (2001). Our methods are computationally efficient for a dataset of this size, running in 8.49 s and 8.51 min to compute the first five GPCs and Sparse GPCs, respectively. Timing results were performed in Matlab using an Intel Xeon X5680 3.33Ghz processor with single-threaded scripts. Notice that the results of this group analysis even more clearly illustrate temporal activation patterns related to the experimental task segments, as well as the specific brain regions in each subject contributing to this signal.
Overall, our GPCA and Sparse GPCA methods have shown significant advantages for unsupervised analysis of fMRI data, even over the current state-of-the-art ICA methods. In particular, they result in much greater dimension reduction and more scientifically interpretable factors, meaning reduced spatial noise and scanner artifacts, and signals that are well separated between estimated components. Also, PCA methods are currently used in fMRI analyses as a processing step before performing ICA or classification methods (Rogers et al. 2007; Lazar 2008; Beckmann 2012). Given that GPCA results in a more parsimonious model with more interpretable factors, our methods could also be used in place of PCA for processing purposes. More neuroimaging-specific investigations are warranted to determine the full efficacy of our methods in this context.

5. DISCUSSION

In this article, we have presented a generalization of PCA and two-way regularized PCA that permits unequal weighting of the matrix elements related to the structural manifold of the data. We have also developed fast methodological and computational algorithms to apply our methods to massive datasets and provided results that clarify the types of penalty functions that may be employed for two-way regularized matrix factorizations.

There are several statistical areas of further research related to our work. In particular, we have used connections to functional data analysis to advocate and illustrate the use of inverse smoothing and smoothing matrices as quadratic operators. Estimating these weighting matrices from the data in conjunction with GPCA factors is beyond the scope of this initial article, but deserves thorough investigation in future work. Also, our techniques using the \( Q, R \)-norm loss and our formulation of an approach with two-way penalties may be useful in other multivariate analysis methods.

Finally, our work has broad implications in a wide array of applied disciplines. Massive image data are common in areas of neuroimaging, microscopy, hyperspectral imaging, remote sensing, and radiology. Other examples of high-dimensional
structured data can be found in environmental and climate studies, times series and finance, engineering sensor and network data, and astronomy. Our GPCA and regularized GPCA methods can be used with these structured datasets for improved dimension reduction, signal recovery, feature selection, data visualization and exploratory data analysis. In conclusion, we have presented a general framework for PCA and regularized PCA for massive structured data can lead to many future areas of research. An R package sGPCA is available implementing our methods, and further technical details and additional results can be found in online supplementary materials.

SUPPLEMENTARY MATERIALS

Supplementary materials include additional fMRI results, the GPMF with smoothness penalties, and proofs.

[Received October 2012. Revised September 2013]


