Regularized Partial Least Squares with an Application to NMR Spectroscopy

Genevera I. Allen[1,2,3,*], **Christine Peterson**[1], **Marina Vannucci**[1], and **Mirjana Maletić-Savatić**[2,3]

[1]Department of Statistics, Rice University, Houston, TX, USA

[2]Department of Pediatrics-Neurology, Baylor College of Medicine, Houston, TX, USA

[3]Jan and Dan Duncan Neurological Research Institute, Texas Children's Hospital, Houston, TX, USA

Abstract

High-dimensional data common in genomics, proteomics, and chemometrics often contains complicated correlation structures. Recently, partial least squares (PLS) and Sparse PLS methods have gained attention in these areas as dimension reduction techniques in the context of supervised data analysis. We introduce a framework for Regularized PLS by solving a relaxation of the SIMPLS optimization problem with penalties on the PLS loadings vectors. Our approach enjoys many advantages including flexibility, general penalties, easy interpretation of results, and fast computation in high-dimensional settings. We also outline extensions of our methods leading to novel methods for non-negative PLS and generalized PLS, an adoption of PLS for structured data. We demonstrate the utility of our methods through simulations and a case study on proton Nuclear Magnetic Resonance (NMR) spectroscopy data.

Keywords

1. INTRODUCTION

Technologies to measure high-throughput biomedical data in proteomics, chemometrics, and genomics have led to a proliferation of high-dimensional data that pose many statistical challenges. As genes, proteins, and metabolites, are biologically interconnected, the variables in these data sets are often highly correlated. In this context, several have recently advocated using partial least squares (PLS) for dimension reduction of supervised data, or data with a response or labels [1–4]. First introduced by Wold [5] as a regression method that uses least squares on a set of derived inputs accounting for multicolinearities, others have since proposed alternative methods for PLS with multiple responses [6] and for classification [7,8]. More generally, PLS can be interpreted as a dimension reduction technique that finds projections of the data that maximize the covariance between the data and the response. Recently, several have proposed to encourage sparsity in these projections, or loadings vectors, to select relevant features in high-dimensional data [3,4,9]. In this paper,

we seek a more general and flexible framework for regularizing the PLS loadings that is computationally efficient for high-dimensional data.

There are several motivations for regularizing the PLS loadings vectors. Partial least squares is closely related to principal components analysis (PCA); namely, the PLS loadings can be computed by solving a generalized eigenvalue problem [6]. Several have shown that the PCA projection vectors are asymptotically inconsistent in high-dimensional settings [10,11]. This is also the case for the PLS loadings, recently shown in Refs 4 and 12. For PCA, encouraging sparsity in the loadings has been shown to yield consistent projections [10,13,14]. While an analogous result has not yet been shown in the context of PLS, one could surmise that such a result could be attained. In fact, this is the motivation for Chun and Kele 's [4] recent Sparse PLS method. In addition to consistency motivations, sparsity has many other qualities to recommend it. The PLS loadings vectors can be used as a data compression technique when making future predictions; sparsity further compresses the data. As many variables in high-dimensional data are noisy and irrelevant, sparsity gives a method for automatic feature selection. This leads to results that are easier to interpret and visualize.

While sparsity in PLS is important for high-dimensional data, there is also a need for more general and flexible regularized methods. Consider NMR spectroscopy as a motivating example. This high-throughput data measures the spectrum of chemical resonances of all the latent metabolites, or small molecules, present in a biological sample [15]. Typical experimental data consists of discretized, functional, and non-negative spectra with variables measuring in the thousands for only a small number of samples. Additionally, variables in the spectra have complex dependencies arising from correlation at adjacent chemical shifts, metabolites resonating at more than one chemical shift, and overlapping resonances of latent metabolites [16]. Because of these complex dependencies, there is a long history of using PLS to reduce the NMR spectrum for supervised data [17,18]. Classical PLS or Sparse PLS, however, are not optimal for this data as they do not account for the non-negativity or functional nature of the spectra. In this paper, we seek a more flexible approach to regularizing PLS loadings that will permit (i) general penalties such as to encourage sparsity, group sparsity, or smoothness, (ii) constraints such as non-negativity, and (iii) that directly account for known data structures such as ordered chemical shifts for NMR spectroscopy. Our framework, based on a penalized relaxation of the SIMPLS optimization problem [6], also leads to a more computationally efficient numerical algorithm.

As we have mentioned, there has been previous work on penalizing the PLS loadings. For functional data, Goutis and Fearn [19] and Reiss and Ogden [20] have extended PLS to encourage smoothness by adding smoothing penalties. Our approach is more closely related to the Sparse PLS methods of Lê Cao *et al*. [3] and Chun and Kele [4]. In the latter, a generalized eigenvalue problem related to PLS objectives is penalized to achieve sparsity, although they solve an approximation to this problem via the elastic net Sparse PCA approach of Zou *et al*. [21]. Noting that PLS can be interpreted as performing PCA on the deflated cross-products matrix, Lê Cao *et al*. [3] replace PCA with Sparse PCA using the approach of Shen and Huang [22]. The core of our algorithm with an ℓ_1 penalty is similar to this approach. We show that this algorithm directly solves a penalized generalization of the SVD problem that is a concave relaxation of the SIMPLS criterion. The major novelty of our work is from this optimization approach, developing a flexible regularization scheme that includes general norm penalties, non-negativity constraints, and generalizations for structured data. Our flexible algorithms lead to interpretable results and fast computational approaches for high-dimensional data.

The paper is organized as follows. Our framework for regularized partial least squares (RPLS) is introduced in Section 2. In Section 3, we introduce two novel extensions of PLS and RPLS: non-negative PLS and generalized PLS (GPLS) for structured data. We illustrate the comparative strengths of our approach in Sections 4 and 5 through simulation studies and a case study on NMR spectroscopy data, respectively, and conclude with a discussion in Section 6.

2. REGULARIZED PARTIAL LEAST SQUARES

In this section, we introduce our framework for regularized partial least squares. While most think of PLS as a regression technique, here we separate the steps of the PLS approach into the dimension reduction stage where the PLS loadings and factors are computed and a prediction stage where regression or classification using the PLS factors as predictors is performed. As our contributions lie in our framework for regularizing the PLS loadings in the dimension reduction stage, we predominately focus on this aspect.

2.1. RPLS Optimization Problem

Introducing notation, we observe data (predictors), $\mathbf{X} \in \Re^{n \times p}$, with p variables measured on n samples and a response $\mathbf{Y} \in \Re^{n \times q}$. We will assume that the columns of $\mathbf{X}$ have been previously standardized. The possibly multivariate response ($q > 1$) could be continuous as in regression or encoded by dummy variables to indicate classes as in Ref. 8, a consideration which we ignore while developing our methodology. The $p \times q$ sample cross-product matrix is denoted as $\mathbf{M} = \mathbf{X}^T \mathbf{Y}$.

Both of the two major algorithms for computing the multivariate PLS factors, NIPALS [5] and SIMPLS [6], can be written as solving a single-factor eigenvalue problem of the following form at each step: maximize$_{\mathbf{v}}$ $\mathbf{v}^T \mathbf{M} \mathbf{M}^T \mathbf{v}$ subject to $\mathbf{v}^T \mathbf{v} = 1$, where $\mathbf{v} \in \Re^p$ are the PLS loadings. Chun and Kele [4] extend this problem by adding an ℓ_1-norm constraint, $\|\mathbf{v}\|$ t, to induce sparsity and solve an approximation to this problem using the Sparse PCA method of Zou *et al.* [21]. Lê Cao *et al.* [3] replace this optimization problem with that of the Sparse PCA approach of Shen and Huang [22].

We take a similar algorithmic approach, but seek regularized PLS factors that directly optimize a criterion related to classical PLS. Notice that the single factor PLS problem can be re-written as the following: maximize$_{\mathbf{v},\mathbf{u}}$ $\mathbf{v}^T \mathbf{M} \mathbf{u}$ subject to $\mathbf{v}^T \mathbf{v} = 1$ & $\mathbf{u}^T \mathbf{u} = 1$, where $\mathbf{u} \in \Re^q$ is a nuisance parameter. The equivalence of these problems was pointed out by de Jong [6] and is a well understood matrix analysis fact. Our single-factor RPLS problem penalizes a direct concave relaxation of this problem:

$$\begin{aligned} \underset{\mathbf{v},\mathbf{u}}{\text{maximize}} \quad & \mathbf{v}^T \mathbf{M} \mathbf{u} - \lambda P(\mathbf{v}) \\ \text{subject to} \quad & \mathbf{v}^T \mathbf{v} \le 1 \,\&\, \mathbf{u}^T \mathbf{u} = 1. \end{aligned} \quad (1)$$

Here, we assume that $P()$ is a convex penalty function that is a norm or semi-norm; these assumptions are discussed further in the subsequent section. To induce sparsity, for example, we can take $P(\mathbf{v}) = \|\mathbf{v}\|_1$. Notice that we have relaxed the equality constraint for $\mathbf{v}$ to an inequality constraint. In doing so, we arrive at an optimization problem that is simple to maximize via an alternating strategy. Fixing $\mathbf{u}$, the problem in $\mathbf{v}$ is concave, and fixing $\mathbf{v}$ the problem is a quadratically constrained linear program in $\mathbf{u}$ with a global solution. Our optimization problem is most closely related to some recent direct biconcave relaxations for two-way penalized matrix factorizations [23,24] and yields similar results to regression based sparse approaches [3,22,25]. Studying the solution to this problem and its properties in the subsequent section will reveal some of the advantages of this optimization approach.

Computing the multifactor PLS solution via the two traditional multivariate approaches, SIMPLS and NIPALS, require solving optimization problems of the same form as the single-factor PLS problem at each step. The SIMPLS method is more direct and has several benefits within our framework; thus, this is the approach we adopt. The algorithm begins by solving the single-factor PLS problem; subsequent factors solve the single-factor problem for a Gram-Schmidt deflated cross-products matrix. If we let the matrix of projection weights $\mathbf{R}_k \in \Re^{p \times k}$ be defined recursively then, $\mathbf{R}_k = [\ \mathbf{R}_{k-1} \quad \mathbf{X}^T \mathbf{z}_k / \mathbf{z}_k^T \mathbf{z}_k\ ]$ where $\mathbf{z}_k = \mathbf{X} \mathbf{v}_k$ is the kth sample PLS factor. The Gram–Schmidt projection matrix $\mathbf{P}_k \in \Re^{p \times p}$ is given by $\mathbf{P}_k = \mathbf{I} - \mathbf{R}_k (\mathbf{R}_k^T \mathbf{R}_k)^{-1} \mathbf{R}_k$, which ensures that $\mathbf{v}_k^T \mathbf{X}^T \mathbf{X} \mathbf{v}_j = 0$ for $j < k$. Then, the optimization problem to find the kth SIMPLS loadings vector is the same as the single-factor problem with the cross-products matrix, $\mathbf{M}$, replaced by the deflated matrix, $\mathbf{M}^{(k)} = \mathbf{P}_{k-1} \mathbf{M}^{(k-1)}$ [6]. Thus, our multifactor RPLS replaces $\mathbf{M}$ in (1) with $\mathbf{M}^{(k)}$ to obtain the kth RPLS factor. While our rank-one optimization problem is closely related to the sparse CCA approach of Witten *et al.* [23], the solutions differ in subsequent factors due to the SIMPLS deflation scheme.

The deflation approach employed via the NIPALS algorithm is not as direct. One typically defines a deflated matrix of predictors and responses, $\tilde{\mathbf{X}}_k = \mathbf{X}(\mathbf{I} - \mathbf{V}_k \mathbf{R}_k^T)$ and $\tilde{\mathbf{Y}}_k = \mathbf{Y}(\mathbf{I} - \mathbf{V}_k \mathbf{R}_k^T)$, with the matrix of projection weights defined as above, and then solves an eigenvalue problem in this deflated space: $\text{maximize}_{\mathbf{w}_k} \mathbf{w}_k^T \tilde{\mathbf{X}}_k^T \tilde{\mathbf{Y}}_k \tilde{\mathbf{Y}}_k^T \tilde{\mathbf{X}}_k \mathbf{w}_k$ subject to $\mathbf{w}_k^T \mathbf{w}_k = 1$ [5]. The PLS loadings in the original space are then recovered by $\mathbf{V}_k = \mathbf{W}_k (\mathbf{R}_k \mathbf{W}_k)^{-1}$. While one can incorporate regularization into the loadings, $\mathbf{w}_k$ (as suggested by Chun and Kele [4]), this is not as desirable. If one estimates sparse deflated loadings, $\mathbf{w}$, then much of the sparsity will be lost in the transform to obtain $\mathbf{V}$. In fact, the elements of $\mathbf{V}$ will be zero if and only if the corresponding elements of $\mathbf{W}$ are zero for all values of k. Then, each of the K PLS loadings will have the exact same sparsity pattern, loosing the flexibility of each set of loadings having adaptively different levels of sparsity. Given this, the more direct deflation approach of SIMPLS is our preferred framework.

2.2. RPLS Solution

A major motivation for our optimization framework for RPLS is that it leads to a simple and direct solution and algorithm. Recall that the single-factor RPLS problem, Eq. (1), is concave in $\mathbf{v}$ with $\mathbf{u}$ fixed and is a quadratically constrained linear program in $\mathbf{u}$ with $\mathbf{v}$ fixed. Thus, we propose to solve this problem by alternating maximizing with respect to $\mathbf{v}$ and $\mathbf{u}$. Each of these maximizations has a simple analytical solution:

PROPOSITION 1—Assume that $P()$ is convex and homogeneous of order one, that is $P()$ is a norm or semi-norm. Let $\mathbf{u}$ be fixed at $\mathbf{u}'$ such that $\mathbf{M}\mathbf{u}' \ \ 0$ or $\mathbf{v}$ fixed at $\mathbf{v}'$ such that $\mathbf{M}^T \mathbf{v}' \ \ 0$. Then, the coordinate updates, $\mathbf{u}^*$ and $\mathbf{v}^*$, maximizing the single-factor RPLS problem, (1), are given by the following: Let $\hat{\mathbf{v}} = \text{argmin}_{\mathbf{v}} \{\frac{1}{2} \|\mathbf{M} \mathbf{u}' - \mathbf{v}\|^2 - \lambda P(\mathbf{v})\}$. Then, $\mathbf{v}^* = \mathbf{v} / \|\mathbf{v}\|_2$ if $\|\mathbf{v}\|_2 > 0$ and $\mathbf{v}^* = 0$ otherwise, and $\mathbf{u}^* = \mathbf{M}^T \mathbf{v}' / \| \mathbf{M}^T \mathbf{v}'\|_2$. When these factors are updated iteratively, they monotonically increase the objective and converge to a local optimum.

While the proof of this result is given in the appendix, we note that this follows closely the Sparse PCA approach of Shen and Huang [22] and the use general penalties within PCA problems of Allen *et al.* [24]. Our RPLS problem can then be solved by a multiplicative update for $\mathbf{u}$ and by a simple re-scaled penalized regression problem for $\mathbf{v}$. The assumption that $P()$ is a norm or semi-norm encompasses many penalties types including the ℓ_1-norm or lasso [26], the ℓ_1/ℓ_2-norm or group lasso [27], the fused lasso [28], and ℓ_q-balls [29]. For many possible penalty types, there exists a simple solution to the penalized regression

problem. With a lasso penalty, $P(\mathbf{v}) = \|\mathbf{v}\|_1$, for example, the solution is given by soft-thresholding: $\hat{\mathbf{v}} = S(\mathbf{Mu}, \lambda)$, where $S(x, \lambda) = \text{sign}(x)(|x| - \lambda)_+$ is the soft-thresholding operator. Our approach gives a more general framework for incorporating regularization directly in the PLS loadings that yield simple and computationally attractive solutions.

We note that the RPLS solution is guaranteed be at most a local optimum of Eq. (1), a result that is typical of other penalized PCA problems [21–25] and sparse PLS methods [3,4]. For a special case, however, our problem has a global solution:

COROLLARY 1—When $q = 1$, that is when $\mathbf{Y}$ is univariate, then the global solution to the single-factor penalized PLS problem (1) is given by the following: Let $\hat{\mathbf{v}} = \text{argmin}_{\mathbf{v}}\{\frac{1}{2}\|\mathbf{M} - \mathbf{v}\|^2 - \lambda P(\mathbf{v})\}$. Then, $\mathbf{v}^* = \hat{\mathbf{v}}/\|\hat{\mathbf{v}}\|_2$ if $\|\hat{\mathbf{v}}\|_2 > 0$ and $\mathbf{v}^* = 0$ otherwise.

This, then is an important advantage of our framework over competing methods. We also briefly note that for PLS regression, there is an interesting connection between Krylov sequences and the PLS regression coefficients [30]. As we take the RPLS factors to be a direct projection of the RPLS loadings, this connection to Krylov sequences is broken, although perhaps for prediction purposes, this is immaterial.

2.3. RPLS Algorithm

Given our RPLS optimization framework and solution, we now put these together in the RPLS algorithm, Algorithm 1. Note that this algorithm is a direct extension of the SIMPLS algorithm [6], where the solution to our single-factor RPLS problem, Eq. (1), replaces the typical eigenvalue problem in Step 2(b). Thus, the algorithmic structure is analogous to that of Lê Cao *et al*. [3]. Since our RPLS problem is nonconcave, there are potentially many local solutions and thus the initializations of $\mathbf{u}$ and $\mathbf{v}$ are important. Similar to much of the Sparse PCA literature [21,22], we recommend initializing these factors to the global single-factor SVD solution, Step 2(a). Second, notice that choice of the regularization parameter, λ, is particularly important. If λ is large enough that $\mathbf{v}_k = 0$, then the kth RPLS factor would be zero and the algorithm would cease. Thus, care is needed when selecting the regularization parameters to ensure they remain within the relevant range. For the special case where $q = 1$, computing $\lambda_{\max}^{(k)}$, the value at which $\hat{\mathbf{v}_k} = 0$, is a straightforward calculation following from the Karush–Khun–Tucker conditions. With the LASSO penalty, for example, this gives $\lambda_{\max}^{(k)} = \max_i |\widehat{\mathbf{M}}_i^{(k)}|$ [31]. For general q, however, $\lambda_{\max}$ does not have a closed form. While one could use numerical solvers to find this value, this is a needless computational effort. Instead, we recommend to perform the algorithm over a range of λ values, discarding any values resulting in a degenerate solution from consideration. Finally, unlike deflation-based Sparse PCA methods which can exhibit poor behavior for very sparse solutions, because of orthogonalization with respect to the data, our RPLS loadings and factors are well behaved with large regularization parameters.

Algorithm 1

K-Factor Regularized PLS

---

1   Center the columns of $\mathbf{X}$ and $\mathbf{Y}$. Let $\mathbf{M}^{(1)} = \mathbf{X}^T \mathbf{Y}$.

2   For $k = 1 \dots K$:

       a.   Initialize $\mathbf{u}_k$ and $\mathbf{v}_k$ to the first left and right singular vectors of $\mathbf{M}^{(k)}$.

       b.   Repeat until convergence:

i.

$$\text{Set } \mathbf{u}_k = \frac{\left(\widehat{\mathbf{M}}^{(k)}\right)^{\mathrm{T}} \mathbf{v}_k}{\left\|\left(\widehat{\mathbf{M}}^{(k)}\right)^{\mathrm{T}} \mathbf{v}_k\right\|_2}.$$

ii.

$$\text{Set } \widehat{\mathbf{v}}_k = \operatorname{argmin}_{\mathbf{v}'_k} \left\{ \left\|\widehat{\mathbf{M}}^{(k)} \mathbf{u}_k - \mathbf{v}'_k\right\|_2^2 - \lambda_k P(\mathbf{v}'_k) \right\}.$$

iii. Set $\mathbf{v}_k = \widehat{\mathbf{v}_k}/\|\widehat{\mathbf{v}_k}\|_2$ if $\|\widehat{\mathbf{v}_k}\|_2 > 0$, and set $\mathbf{v}_k = 0$ and exit the algorithm otherwise.

c. RPLS Factor: $\mathbf{z}_k = \mathbf{X} \mathbf{v}_k$.

d. RPLS projection matrix: Set $\mathbf{R}^{(k)} = \begin{bmatrix} \mathbf{R}^{(k-1)} & \mathbf{X}^{\mathrm{T}} \mathbf{z}_k / \mathbf{z}_k^{\mathrm{T}} \mathbf{z}_k \end{bmatrix}$ and $\mathbf{P}_k = \mathbf{I} - \mathbf{R}^{(k)} ((\mathbf{R}^{(k)})^{\mathrm{T}} \mathbf{R}^{(k)})^{-1} (\mathbf{R}^{(k)})^{\mathrm{T}}$.

e. Orthogonalization Step: $\mathbf{M}^{(k+1)} = \mathbf{P}_k \mathbf{M}^{(k)}$

3 Return RPLS Factors $\mathbf{z}_1 \ldots \mathbf{z}_K$ and RPLS Loadings: $\mathbf{v}_1 \ldots \mathbf{v}_K$.

Selecting the appropriate regularization parameter, λ, is an important practical consideration. Existing methods that incorporate regularization in PLS have suggested using cross-validation or other model selection methods in the ultimate regression or classification stage of the full PLS procedure [4,20]. While one could certainly implement these approaches within our RPLS framework, we suggest a simpler and more direct approach. We select λ within the dimension reduction stage of RPLS, specifically in Step 2(b) of our RPLS Algorithm. Doing so, has a number of advantages. First, this increases flexibility as it separates selection of λ from deciding how many factors, K, to use in the prediction stage, permitting a separate regularization parameter, λ_k, to be selected for each RPLS factor. Second, coupling selection of the regularization parameter to the prediction stage requires fixing the supervised modeling method before computing the RPLS factors. With our approach, the RPLS factors can be computed and stored to use as predictors in a variety of modeling procedures. Finally, separating selection of λ_k and K in the prediction stage is computationally advantageous as a grid search over tuning parameters is avoided. Nesting selection of λ within Step 2(b) is also faster as recent developments such as warm starts and active set learning can be used to efficiently fit the entire path of solutions for many penalty types [31]. Practically, selecting λ_k within the dimension reduction stage is analogous to selecting the regularization parameters for Sparse PCA methods on $\mathbf{M}^{(k)}$. Many approaches including cross-validation [22,32] and BIC methods [24,25] have been suggested for this purpose; in results given in this paper, we have implemented the BIC method as described by Allen *et al.* 24. Selection of the number of RPLS factors, K, will largely be dependent on the supervised method used in the prediction stage, although cross-validation can be used with any method. When choosing K for PLS regression, for example, Huang *et al.* [33] suggest an automatic approach via regularization.

Computationally, our algorithm is an efficient approach. As discussed in the previous section, the particular computational requirements for computing the RPLS loadings in Step 2(b) are penalty specific, but are minimal for a wide class of commonly used penalties. Beyond Step 2(b), the major computational requirement is inverting the weight matrix, $\mathbf{R}_k^{\mathrm{T}} \mathbf{R}_k$, to compute the projection matrix. As this matrix is found recursively via the Gram-Schmidt scheme, however, employing properties of the Schur complement can reduce the computational effort to that of matrix multiplication $O(pk)$ [34]. Finally, notice that we take the RPLS factors to be the direct projection of the data by the RPLS loadings. Overall, the advantages of our RPLS framework and algorithm include (i) computational efficiency, (ii) flexible modeling, and (iii) direct estimation of the RPLS loadings and factors.

3. EXTENSIONS

As our framework for regularizing PLS is general, there are many possible extensions of our methodology. Many have suggested, for example, to extend PLS and sparse PLS specifically for discriminant analysis [8] and for generalized linear models [7,35]. Such extensions are also possible within our framework. In this section, we specifically focus on two novel extensions of PLS and RPLS that will be useful for applications to spectroscopy data. These include generalizations for PLS and RPLS with structured data and non-negative PLS and RPLS.

3.1. GPLS for Structured Data

Recently, Allen *et al*. [24] proposed a generalization of PCA (GPCA) that is a appropriate for high-dimensional structured data, or data in which the variables are associated with some known distance metric. As motivation, consider NMR spectroscopy data where variables are ordered on the spectrum and variables at adjacent chemical shifts are known to be highly correlated. Classical multivariate techniques such as PCA and PLS ignore these structures; GPCA encodes structure into a matrix factorization problem through positive semi-definite quadratic operators such as graph Laplacians or smoothing matrices [24,36]. If we assume that the noise in the data follows the data structure and these quadratic operators capture aspects of this structure, then Allen *et al*. [24] showed that GPCA can be interpreted as finding principal modes of variation that are orthogonal to the structured noise. In the context of NMR spectroscopy, we seek PLS factors that are independent from the known noise correlations at adjacent chemical shifts. Similar to GPCA then, we seek to directly account for known structure in PLS and within our RPLS framework.

Our development of GPLS is motivated by NMR spectroscopy; that is, we seek a quadratic operator that encodes the known structural relationships, the correlations between adjacent variables. Let us define then quadratic operator, $\mathbf{Q} \in \Re^{p \times p}$: $\mathbf{Q} \succeq 0$ which we assume is fixed and known. For spectroscopy data, for example, $\mathbf{Q}$ could be taken as a diagonal tapered matrix thus accounting for the correlations between adjacent variables. By transforming all inner-product spaces to those induced by the $\mathbf{Q}$-norm, we can define our single-factor Generalized RPLS optimization problem in the following manner:

$$
\begin{aligned}
\underset{\mathbf{v},\mathbf{u}}{\text{maximize}} \quad & \mathbf{v}^{\mathrm{T}} \mathbf{Q} \mathbf{M} \mathbf{u} - \lambda P(\mathbf{v}) \\
\text{subject to} \quad & \mathbf{v}^{\mathrm{T}} \mathbf{Q} \mathbf{v} \leq 1, \,\& \, \mathbf{u}^{\mathrm{T}} \mathbf{u} = 1.
\end{aligned}
\tag{2}
$$

By finding the RPLS factors in the $\mathbf{Q}$-norm, we find factors separate from the noise structure of the data. For the multifactor Generalized RPLS problem, the factors and projection matrices are also changed. The kth factor is given by $\mathbf{z}_k = \mathbf{X} \mathbf{Q} \mathbf{v}_k$, the weighting matrix, $\mathbf{R}_k = [\ \mathbf{R}_{k-1} \quad \mathbf{X}^{\mathrm{T}} \mathbf{z}_k / \mathbf{z}_k^{\mathrm{T}} \mathbf{z}_k \ ]$ as before, and the projection matrix is $\mathbf{P}_k = \mathbf{I} - \mathbf{R}_k^{\mathrm{T}} (\mathbf{R}_k^{\mathrm{T}} \mathbf{Q} \mathbf{R}_k)^{-1} \mathbf{R}_k^{\mathrm{T}} \mathbf{Q}$ with deflated cross-products matrix $\mathbf{M}^{(k)} = \mathbf{P}_{k-1} \mathbf{M}^{(k-1)}$. Note that if $\lambda = 0$ and if the inequality constraint is forced to be an equality constraint, then we have the optimization problem for GPLS. Notice also that instead of enforcing orthogonality of the PLS loadings with respect to the data, $\mathbf{v}_k^{\mathrm{T}} \mathbf{X}^{\mathrm{T}} \mathbf{X} \mathbf{v}_j$, the GPLS problem enforces orthogonality in a projected data space, $\mathbf{v}_k^{\mathrm{T}} \mathbf{Q} \mathbf{X}^{\mathrm{T}} \mathbf{X} \mathbf{Q} \mathbf{v}_j$. If we let $\tilde{\mathbf{Q}}$ be a matrix square root of $\mathbf{Q}$ as defined by Allen *et al*. [24], then (2) is equivalent to the multifactor RPLS problem for $\tilde{\mathbf{X}} = \mathbf{X} \tilde{\mathbf{Q}}$ and $\tilde{\mathbf{v}} = \tilde{\mathbf{Q}} \mathbf{v}$. This equivalence is shown in the proof of the solution to Eq. (2).

As with PLS and our RPLS framework, GPLS and Generalized RPLS can be solved by coordinate-wise updates that converge to the global and local optimum respectively:

PROPOSITION 2

1. GPLS: The GPLS problem, Eq. (2) when $\lambda = 0$, is solved by the first set of GPCA factors of $\mathbf{M}$. The global solution to the GPLS problem can be found by iteratively updating the following until convergence: $\mathbf{v} = \mathbf{Mu}/\|\mathbf{Mu}\|_{\mathbf{Q}}$ and $\mathbf{u} = \mathbf{M}^{\mathrm{T}}\mathbf{Q}\,\mathbf{v}/\|\,\mathbf{M}^{\mathrm{T}}\mathbf{Q}\,\mathbf{v}\|_2$, where $\|x\|_{\mathbf{Q}}$ is defined as $\sqrt{x^{\mathrm{T}}\mathbf{Q}\,x}$.

2. Generalized RPLS: Under the assumptions of Proposition 1, let $\widehat{\mathbf{v}} = \operatorname{argmin}_{\mathbf{v}'}\{\|\mathbf{Mu} - \mathbf{v}'\|_{\mathbf{Q}}^2 + \lambda P(\mathbf{v}')\}$, then the coordinate-wise updates to Eq. (2) are given by: $\mathbf{v}^* = \mathbf{v}/\|\mathbf{v}\|_{\mathbf{Q}}$ if $\|\mathbf{v}\|_{\mathbf{Q}} > 0$ and $\mathbf{v}^* = 0$ otherwise, and with $\mathbf{u}^*$ defined as above. When updated iteratively, these converge to a local optimum of (2).

Thus, the solution to our Generalized RPLS problem can be solved by a generalized penalized least squares problem. Algorithmically, solving the multi-factor GPLS and Generalized RPLS problems follow the same structure as that of Algorithm 2.2. The solutions outlined above replace Step 2(b), with the altered Generalized RPLS factors and projections matrices replacing Steps 2 (c), (d), and (e). In other words, GPLS or Generalized RPLS is performed by finding the GPCA or Regularized GPCA factors of a deflated cross-products matrix, where the deflation is performed to rotate the cross-products matrix so that it is orthogonal to the data in the $\mathbf{Q}$-norm. Computationally, these algorithms can be performed efficiently using the techniques described by Allen *et al.* [24] that do not require inversion or taking eigenvalue decompositions of $\mathbf{Q}$. Thus, the GPLS and Generalized RPLS methods are computationally feasible for high-dimensional data sets.

We have shown the most basic extension of GPCA technology to PLS and our RPLS framework, but there are other possible formulations. For two-way data, projections in the 'sample' space may be appropriate in addition to projecting variables in the $\mathbf{Q}$-norm. With neuroimaging data, for example, the data matrix may be oriented as brain locations, voxels, by time points. As the time series is most certainly not independent, one may wish to transform these inner product spaces using another quadratic operator, $\mathbf{W} \in \Re^{n \times n}$, changing $\mathbf{M}$ to $\mathbf{X}^{\mathrm{T}}\mathbf{W}\mathbf{Y}$ and $\mathbf{R}_k$ to $\mathbf{R}_k = [\ \mathbf{R}_{k-1} \quad \mathbf{X}^{\mathrm{T}}\mathbf{W}\mathbf{z}_k/\mathbf{z}_k^{\mathrm{T}}\mathbf{W}\mathbf{z}_k\ ]$, analogous to Allen *et al.* [24]. Overall, we have outlined a novel extension of PLS and our RPLS methodologies to work with high-dimensional structured data.

3.2. Non-Negative PLS

Many have advocated estimating non-negative matrix factors [37] and non-negative principal component loadings [38] as a way to increase interpret-ability of multivariate methods. For scientific data sets such as NMR spectroscopy in which variables are naturally non-negative, enforcing non-negativity of the loadings vectors can greatly improve interpretability results and the performance of methods [36]. Here, we illustrate how to incorporate non-negative loadings into our RPLS framework. Consider the optimization problem for single-factor non-negative RPLS:

$$\begin{aligned} \underset{\mathbf{v},\mathbf{u}}{\text{maximize}} \quad & \mathbf{v}^{\mathrm{T}}\mathbf{Mu} - \lambda P(\mathbf{v}) \\ \text{subject to} \quad & \mathbf{v}^{\mathrm{T}}\mathbf{v} \le 1, \ \mathbf{u}^{\mathrm{T}}\mathbf{u} = 1 \,\&\, \mathbf{v} \ge 0. \end{aligned} \tag{3}$$

Solving this optimization problem is a simple adaption of Proposition 1; the penalized regression problem is replaced by a penalized non-negative regression problem. For many

penalty types, these problems have a simple solution. With the ℓ_1-norm penalty, for example, the soft-thresholding operator in the update for **v** is replaced by the positive soft-thresholding operator: $\mathbf{v} = P(\mathbf{Mu}, \lambda) = (\mathbf{Mu} - \lambda)_+$ [36]. Our RPLS framework, then, gives a simple and computationally efficient method for enforcing non-negativity in the PLS loadings. Also, as in Ref. 36, non-negativity and quadratic operators can be used in combination for PLS to create flexible approaches for high-dimensional data sets.

4. SIMULATION STUDIES

We explore the performance of our RPLS methods for regression in a univariate and a multivariate simulation study.

4.1. Univariate Simulation

In this simulation setting, we compare the mean squared prediction error and variable selection performance of RPLS against competing methods in the univariate regression response setting with correlated predictors. Following the approach in Section 5.3 of Chun and Kele [4], we include scenarios where n is greater than p and where n is less than p with differing levels of noise. For the $n > p$ setting, we use $n = 400$ and $p = 40$. For the first $n < p$ setting, we use $n = 40$ and $p = 80$. For these cases, 75% of the p predictors are true predictors, while the remaining 25% are spurious predictors that are not used in the generation of the response. We also include an additional $n < p$ setting where the underlying signal is more sparse. In this case, we use $n = 40$ and $p = 200$ with 25% of the p predictors as true predictors. For the low and high noise scenarios, we use signal-to-noise ratios (SNR) of 10 and 5.

To create correlated predictors as in Ref. 4, we construct hidden variables $H_1, \ldots, H_3$, where $H_i \sim \mathcal{N}(0, 25\mathbf{I}_n)$. The columns of the predictor matrix X_i are generated as the sum of a hidden variable and independent random noise as follows. For the cases where 75% of the p predictors are used in the generation of the response, $X_i = H_1 + \varepsilon_i$ for $1 \leq i \leq 3p/8$, $X_i = H_2 + \varepsilon_i$ for $3p/8 < i \leq 3p/4$, and $X_i = H_3 + \varepsilon_i$ for $3p/4 < i \leq p$, where $\varepsilon_i \sim \mathcal{N}(0, \mathbf{I}_n)$. When 25% of the p predictors are used in the generation of the response, $X_i = H_1 + \varepsilon_i$ for $1 \leq i \leq p/8$, $X_i = H_2 + \varepsilon_i$ for $p/8 < i \leq p/4$, and $X_i = H_3 + \varepsilon_i$ for $p/4 < i \leq p$, where $\varepsilon_i \sim \mathcal{N}(0, \mathbf{I}_n)$. For all cases, the response vector $Y = 3H_1 - 4H_2 + f$, where $f \sim \mathcal{N}(0, 25\mathbf{I}_n/\text{SNR})$. Training and test sets for all settings of n, p and SNR are created using this approach.

For the comparison of methods, **X** and Y are standardized, and parameter selection is carried out using 10-fold cross validation on the training data. For the sparse partial least squares (SPLS) method described by Chun and Kele [4], the spls R package [39] is used with η chosen from the sequence $(0.1, 0.2, \ldots, 0.9)$ and K from 5 to 10. Note that for our methods, we choose to select K automatically via the lasso penalized PLS regression problem as in Ref. 33. Thus for RPLS, lasso penalties were used with λ and γ chosen from 25 equally spaced values between 10^{-5} and $\log(\max(|\mathbf{X}'Y|))$ on the log scale. For the lasso and elastic net, the glmnet R package [31] is used with the same choices for λ.

The average mean squared prediction error (MSPE), true positive rate (TPR), and false positive rate (FPR) across 30 simulation runs are given in Table 1. The penalized regression methods clearly outperform traditional PLS in terms of the mean squared prediction error, with RPLS having the best prediction accuracy among all methods. SPLS and RPLS are nearly perfect in correctly identifying the true variables, but SPLS tends to have higher rates of false positives. In contrast, the lasso and elastic net have high specificity, but fail to identify many true predictors.

4.2. Multivariate Simulation

In this simulation setting, we compare the mean squared prediction error of regularized PLS against competing methods for multivariate regression. As in the univariate simulation, we include scenarios where $n > p$ and $n < p$ with varying levels of noise, but now our response $\mathbf{Y}$ is a matrix of dimension $n \times q$ with $q = 10$. For the $n > p$ scenario, we use $n = 400$ and $p = 40$ with 5 true predictors. For the moderate $n < p$ scenario, we use $n = 40$ and $p = 80$ with 10 true predictors. We also include a more extreme scenario where we use $n = 40$ and $p = 200$ with 25 true predictors. In each case, we test the methods using SNR of 2 and 1.

The simulated data is generated using eight binary hidden variables $H_1, \ldots, H_8$ with entries drawn from the Bernoulli (0.5) distribution. The coefficient matrix $\mathbf{A}$ contains standard normal random entries for the first p_{true} columns, with the remaining columns set to 0. The predictor matrix $\mathbf{X} = \mathbf{H} \cdot \mathbf{A} + \mathbf{E}$, where the entries of $\mathbf{E}$ are drawn from the $\mathcal{N}(0, 0.1^2)$ distribution. The coefficient matrix $\mathbf{B}$ contains entries drawn from the $\mathcal{N}(0, \text{SNR} \cdot n \cdot q/\text{tr}(\mathbf{HH}'))$ distribution. The response matrix $\mathbf{Y} = \mathbf{H} \cdot \mathbf{B} + \mathbf{F}$, where the entries of $\mathbf{F}$ are drawn from the standard normal distribution. Both training and test sets are generated using this procedure, and both $\mathbf{X}$ and $\mathbf{Y}$ are standardized.

For the penalized methods including sparse PCA (SPCA) and regularized PLS (RPLS) the penalty parameter λ is chosen from 25 equally spaced values between -5 and $\log(\max(|\mathbf{X}'\mathbf{Y}|))$ on the log scale using the BIC criterion. For RPLS, γ, the PLS regression penalty parameter for selecting K, is chosen from the same set of options as λ using the BIC criterion. To obtain the coefficient $\beta = \mathbf{VZ}'\mathbf{Y}_{\text{training}}$, the columns of $\mathbf{V}$ and $\mathbf{Z}$ were normalized. The results shown in Table 2 demonstrate that regularized PLS outperforms both sparse PCA and standard PLS.

5. CASE STUDY: NMR SPECTROSCOPY

We evaluate the utility of our methods through a case study on NMR spectroscopy data, a classic application of PLS methods from the chemometrics literature. We apply our RPLS methods to an *in vitro* one-dimensional NMR data set consisting of 27 samples from five classes of neural cell types: neurons, neural stem cells, microglia, astrocytes, and ogliodendrocytes [40], analyzed by some of the same authors using PCA methods in Ref. 36. Data are pre-processed in the manner described by Dunn *et al*. [18]: functional spectra is discretized into bins of size 0.04 ppms yielding a total of 2394 variables, spectra for each sample are baseline corrected and normalized to their integral, and variables are standardized. For all PLS methods, the response, $\mathbf{Y}$ is 27×5 and coded with indicators inversely proportional to the sample size in each class as described by Barker and Rayens [8]. For each method, five PLS or PCA factors were taken and used as predictors in linear discriminant analysis to classify the NMR samples. To be consistent, the BIC method was used to select any penalty parameters except for the Sparse PLS method of Chun and Kele [4] where the default in the R package spls was employed [39]. The Sparse GPCA and Sparse GPLS methods were applied with non-negativity constraints as described by Allen & Maleti -Savati [36] and in Section 3. Finally, for the generalized methods, the quadratic operator was selected by maximizing the variance explained by the first component; a weighted Laplacian matrix with weights inversely proportional to the Epanechnikov kernel with a bandwidth of 0.2 ppms was employed [24].

In Table 3, we give the training and leave-one out cross-validation misclassification errors for our methods and competing methods on the neural cell NMR data. Notice that our Sparse GPLS method yields the best error rates followed by the Sparse PLS [4] and our GPLS methods. Additionally, our Sparse GPLS methods are significantly faster than competing approaches. In Table 4, the time in seconds to compute the entire solution path

(51 values of λ) is reported. Timing comparisons were done on a Intel Xeon X5680 3.33 Ghz processor with 16 GB RAM using single-threaded scripts coded in Matlab or C as indicated. (These comparisons should be interpreted keeping in mind that there are possibly minor speed differences between Matlab and R.)

In addition to faster computation and better classification rates, our method's flexibility leads to easily interpretable results. We present the Sparse GPLS loadings superimposed on the scaled spectra from each neural cell type and sample heatmaps in Fig. 1. For comparison, we give the first two PLS loadings in Fig. 2 for PLS and Sparse PLS [4]. The PLS loadings are noisy, and the sample PLS components for PLS and Sparse PLS are difficult to interpret as the loadings are both positive and negative. By constraining the PLS loadings to be non-negative, the chemical shifts the metabolites indicative of each neural cell type are readily apparent with the Sparse non-negative GPLS loadings. Additionally as shown in the sample PLS heatmaps, the neural cell types are well differentiated. For example, chemical resonances at 1.30 and 3.23 ppms characterize Glia (Astrocytes and Ogliodendrocytes) and Neurons (PLS 1), resonances at 1.19 and 3.66 ppms characterize Microglia (PLS 2), resonances at 3.23 and 2.65 ppms characterize Astrocytes (PLS 3), resonances at 1.30, 3.02, and 3.55 ppms characterize Ogliodendrocytes (PLS 4), and resonances at 1.28 and 3.23 ppms characterize Neural stem cells. Note that some of these metabolites were identified by some of the same authors using PCA methods in Refs 36,40. Using our flexible PLS approach for supervised dimension reduction, however, gives a much clear metabolic signature of each neural cell type. Overall, this case study on NMR spectroscopy data has revealed the many strengths of our method as well as identified possible metabolite biomarkers for further biological investigation.

6. DISCUSSION

We have presented a framework for regularizing partial least squares with convex and order one penalties. Additionally, we have shown how this approach can be extend for structured data via GPLS and Generalized RPLS and extended to incorporate non-negative PLS or RPLS loadings. Our approaches directly solve penalized relaxations of the SIMPLS optimization criterion. These in turn, have many advantages including computational efficiency, flexible modeling, easy interpretation and visualization, better feature selection, and improved predictive accuracy as demonstrated in our simulations and case study on NMR spectroscopy.

There are many future areas of research related to our methodology. As many have advocated using PCA, or even supervised PCA [41], as a dimension reduction technique prior supervised modeling, RPLS may be a powerful alternative in this context. While we have briefly discussed the use of our methods for general regression or classification procedures, specific investigation of the RPLS factors as predictors in the generalized linear model framework [7,35], the survival analysis framework [42], and others are needed. Additionally, following the close connection of PLS for classification with the classes coded as dummy variables to Fisher's discriminant analysis [8], our RPLS approach may give an alternative strategy for regularized linear discriminant analysis. Further development of our novel extensions for GPLS and Nonnegative PLS is also needed. Finally, Nadler and Coifman [12] and Chun and Kele [4] have shown asymptotic inconsistency of PLS regression methods when the number of variables is permitted to grow faster than the sample size. For related PCA methods, a few have shown consistency of Sparse PCA in these settings [10,13]. Proving consistent recovery of the RPLS loadings and the corresponding regression or classification coefficients is an open area of future research.

Finally, we have demonstrated the utility of our methods through a case study on NMR spectroscopy data, but there are many other potential applications of our technology. These include chemometrics, proteomics, metabolomics, high-throughput genomics, imaging, hyperspectral imaging, and neuroimaging. Overall, we have presented a flexible and powerful tool for supervised dimension reduction of high-dimensional data with many advantages and potential areas of future research and application. An R package and a Matlab toolbox named RPLS that implements our methods will be made publicly available.

Acknowledgments

References

1. Nguyen D, Rocke D. Tumor classification by partial least squares using microarray gene expression data. Bioinformatics. 2002; 18(1):39–50. [PubMed: 11836210]

2. Boulesteix A, Strimmer K. Partial least squares: a versatile tool for the analysis of high-dimensional genomic data. Brief Bioinform. 2007; 8(1):32. [PubMed: 16772269]

3. Cao KLê, Rossouw D, Robert-Granié C, Besse P. A sparse PLS for variable selection when integrating omics data. Stat Appl Genet Mol Biol. 2008; 7(1):35.

4. Chun H, Kele S. Sparse partial least squares regression for simultaneous dimension reduction and variable selection. J Roy Stat Soc Ser B. 2010; 72(1):3–25.

5. Wold H. Estimation of principal components and related models by iterative least squares. Multivariate Anal. 1966; 1:391–420.

6. de Jong S. SIMPLS: an alternative approach to partial least squares regression. Chemomet Intell Lab Syst. 1993; 18(3):251–263.

7. Marx B. Iteratively reweighted partial least squares estimation for generalized linear regression. Technometrics. 1996:374–381.

8. Barker M, Rayens W. Partial least squares for discrimination. J Chemomet. 2003; 17(3):166–173.

9. Cao KLê, Boitard S, Besse P. Sparse PLS discriminant analysis: biologically relevant feature selection and graphical displays for multiclass problems. BMC Bioinform. 2011; 12(1):253.

10. Johnstone I, Lu A. On consistency and sparsity for principal components analysis in high dimensions. J Am Stat Assoc. 2009; 104(486):682–693. [PubMed: 20617121]

11. Jung S, Marron J. PCA consistency in high dimension, low sample size context. Ann Stat. 2009; 37(6B):4104–4130.

12. Nadler B, Coifman R. The prediction error in CLS and PLS: the importance of feature selection prior to multivariate calibration. J Chemomet. 2005; 19(2):107–118.

13. Amini A, Wainwright M. High-dimensional analysis of semidefinite relaxations for sparse principal components. Ann Stat. 2009; 37(5B):2877–2921.

14. Shen, D. PhD Thesis. University of North Carolina; 2012. Sparse PCA asymptotics and analysis of tree data.

15. Nicholson J, Lindon J. Systems biology: metabonomics. Nature. 2008; 455(7216):1054–1056. [PubMed: 18948945]

16. De Graaf, R. In Vivo NMR Spectroscopy: Principles and Techniques. Wiley-Interscience; 2007.

17. Goodacre R, Vaidyanathan S, Dunn W, Harrigan G, Kell D. Metabolomics by numbers: acquiring and understanding global metabolite data. Trends Biotechnol. 2004; 22(5):245–252. [PubMed: 15109811]

18. Dunn W, Bailey N, Johnson H. Measuring the metabolome: current analytical technologies. Analyst. 2005; 130(5):606–625. [PubMed: 15852128]

19. Goutis C, Fearn T. Partial least squares regression on smooth factors. J Am Stat Assoc. 1996; 91(434):627–632.

20. Reiss P, Ogden R. Functional principal component regression and functional partial least squares. J Am Stat Assoc. 2007; 102(479):984–996.

21. Zou H, Hastie T, Tibshirani R. Sparse principal component analysis. J Comput Graph Stat. 2006; 15(2):265–286.

22. Shen H, Huang J. Sparse principal component analysis via regularized low rank matrix approximation. J Multivariate Anal. 2008; 99(6):1015–1034.

23. Witten D, Tibshirani R, Hastie T. A penalized matrix decomposition, with applications to sparse principal components and canonical correlation analysis. Biostatistics. 2009; 10(3):515. [PubMed: 19377034]

24. Allen GI, Grosenick L, Taylor J. A generalized least squares matrix decomposition. Rice University Technical Report No TR2011-03. 2011

25. Lee M, Shen H, Huang J, Marron J. Biclustering via sparse singular value decomposition. Biometrics. 2010; 66(4):1087–1095. [PubMed: 20163403]

26. Tibshirani R. Regression shrinkage and selection via the lasso. J Roy Stat Soc Ser B. 1996; 58(1): 267–288.

27. Yuan M, Lin Y. Model selection and estimation in regression with grouped variables. J Roy Stat Soc Ser B. 2006; 68(1):49–67.

28. Tibshirani R, Saunders M, Rosset S, Zhu J, Knight K. Sparsity and smoothness via the fused lasso. J Roy Stat Soc Ser B. 2005; 67(1):91–108.

29. Raskutti, G.; Wainwright, M.; Yu, B. Minimax rates of convergence for high-dimensional regression under LQ-ball sparsity. 47th Annual Allerton Conference on Communication, Control, and Computing; 2009; Allerton. 2009. p. 251-257.

30. Krämer N. An overview on the shrinkage properties of partial least squares regression. Comput Stat. 2007; 22(2):249–273.

31. Friedman J, Hastie T, Tibshirani R. Regularization paths for generalized linear models via coordinate descent. J Stat Softw. 2010; 33(1):1. [PubMed: 20808728]

32. Owen A, Perry P. Bi-cross-validation of the SVD and the nonnegative matrix factorization. Annals. 2009; 3(2):564–594.

33. Huang X, Pan W, Park S, Han X, Miller L, Hall J. Modeling the relationship between LVAD support time and gene expression changes in the human heart by penalized partial least squares. Bioinformatics. 2004; 20(6):888–894. [PubMed: 14751963]

34. Horn, RA.; Johnson, CR. Matrix Analysis. Cambridge University Press; 1985.

35. Chung D, Keles S. Sparse partial least squares classification for high dimensional data. Stat Appl Genet Mol Biol. 2010; 9(1)

36. Allen G, Maleti -Savati M. Sparse non-negative generalized PCA with applications to metabolomics. Bioinformatics. 2011; 27(21):3029–3035. [PubMed: 21930672]

37. Lee D, Seung H. Learning the parts of objects by non-negative matrix factorization. Nature. 1999; 401(6755):788–791. [PubMed: 10548103]

38. Hoyer P. Non-negative matrix factorization with sparseness constraints. J Mach Learn Res. 2004; 5:1457–1469.

39. Chung D, Chun H, Keles S. SPLS: Sparse Partial Least Squares (SPLS) Regression and Classification. R package, version 2.1-1. 2012

40. Manganas L, Zhang X, Li Y, Hazel R, Smith S, Wagshul M, Henn F, Benveniste H, Djuri P, Enikolopov G, Maleti -Savati M. Magnetic resonance spectroscopy identifies neural progenitor cells in the live human brain. Science. 2007; 318(5852):980. [PubMed: 17991865]

41. Bair E, Hastie T, Paul D, Tibshirani R. Prediction by supervised principal components. J Am Stat Assoc. 2006; 101(473):119–137.

42. Nguyen D, Rocke D. Partial least squares proportional hazard regression for application to DNA microarray survival data. Bioinformatics. 2002; 18(12):1625–1632. [PubMed: 12490447]

APPENDIX

PROOFS

Proof of Proposition 1

The proof of this result follows from an argument in Ref. 24, but we outline this here for completion. The updates for $\mathbf{u}$ are straightforward. We show that the subgradient equations of the penalized regression problem, $\frac{1}{2}\|\mathbf{Mu} - \mathbf{v}\| + \lambda P(\mathbf{v})$, for $\mathbf{v}^*$ as defined in the stated result are equivalent to the KKT conditions of Eq. (1). The subgradient equation of the latter is, $\mathbf{Mu} - \lambda \nabla P(\mathbf{v}^*) - 2\gamma^* \mathbf{v}^* = 0$, where $\nabla P()$ is the subgradient of $P()$ and γ^* is the Lagrange multiplier for the inequality constraint with complementary slackness condition, $\gamma^*((\mathbf{v}^*)^{\mathrm{T}} \mathbf{v}^* - 1) = 0$. The subgradient of the penalized regression problem is $\mathbf{Mu} - \hat{\mathbf{v}} - \lambda \nabla P(\hat{\mathbf{v}}) = 0$. Now, since $P()$ is order one, we this subgradient is equivalent to $\mathbf{Mu} - \frac{1}{c}\tilde{\mathbf{v}} - \lambda \nabla P(\tilde{\mathbf{v}})$ for any $c > 0$ and for $\tilde{\mathbf{v}} = c\hat{\mathbf{v}}$. Then, taking $c = 1/\|\mathbf{v}\|_2 = 1/2\gamma^*$ for any $\hat{\mathbf{v}} \neq 0$, we see that both the complimentary slackness condition is satisfied and the subgradients are equivalent. It is easy to verify that the pair (0, 0) also satisfy the KKT conditions of Eq. (1).

Proof of Corollary 1

The proof of this fact follows in a straightforward manner from that of Proposition 1 as the only feasible solution for $\mathbf{u}$ is $\mathbf{u}^* = 1$. We are then left with a concave optimization problem, $\text{maximize}_{\mathbf{v}} \; \mathbf{v}^{\mathrm{T}} \mathbf{M} - \lambda P(\mathbf{v})$ subject to $\mathbf{v}^{\mathrm{T}} \mathbf{v} \leq 1$. From the proof of Proposition 1, we have that this optimization problem is equivalent to the desired result. As we are left with a concave problem, the global optimum is achieved.

Proof of Proposition 2

First, define $\tilde{\mathbf{Q}}$ to be a matrix square root of $\mathbf{Q}$ as in Ref. 24. In this paper, they showed that Generalized PCA was equivalent to PCA on the matrix $\tilde{\mathbf{X}} = \mathbf{X} \tilde{\mathbf{Q}}$ for projected factors $\mathbf{V} = \tilde{\mathbf{Q}}^{\dagger} \tilde{\mathbf{V}}$. In other words, if $\tilde{\mathbf{X}} = \tilde{\mathbf{U}} \mathbf{D} \tilde{\mathbf{V}}$ is the singular value decomposition, then the GPCA solution, $\mathbf{V}$ can be defined accordingly. Here, we will prove that the multi-factor RPLS problem for $\tilde{\mathbf{X}}$ and $\tilde{\mathbf{v}}_k$ is equivalent to the stated Generalized RPLS problem (2) for $\lambda = 0$. The constraint regions are trivially equivalent so we must show that $\tilde{\mathbf{v}}_k^{\mathrm{T}} \tilde{\mathbf{P}}_{k-1} \tilde{\mathbf{M}} = \mathbf{v}^{\mathrm{T}} \mathbf{Q} \mathbf{P}_{k-1} \mathbf{Q} \mathbf{M}$. The PLS factors, $\tilde{\mathbf{z}}_k = \tilde{\mathbf{X}} \tilde{\mathbf{v}}_k = \mathbf{X} \mathbf{Q} \mathbf{v}_k = \mathbf{z}_k$, are equivalent. Ignoring the normalizing term in the denominator, the columns of the projection weighting matrix are $\tilde{\mathbf{R}}_k = \tilde{\mathbf{X}}^{\mathrm{T}} \tilde{\mathbf{z}}_k = \tilde{\mathbf{Q}}^{\mathrm{T}} \mathbf{X} \mathbf{z}_k = \tilde{\mathbf{Q}}^{\mathrm{T}} \mathbf{R}_k$. Thus, the ij th element of $\tilde{\mathbf{R}}^{\mathrm{T}} \tilde{\mathbf{R}} = \mathbf{z}_i^{\mathrm{T}} \mathbf{X} \tilde{\mathbf{Q}} \tilde{\mathbf{Q}}^{\mathrm{T}} \mathbf{X}^{\mathrm{T}} \mathbf{z}_j = \mathbf{R}_i^{\mathrm{T}} \mathbf{Q} \mathbf{R}_j$ as stated. Putting these together, we have $\tilde{\mathbf{v}}_k^{\mathrm{T}} \tilde{\mathbf{P}}_{k-1} \tilde{\mathbf{M}} = \mathbf{v}_k^{\mathrm{T}} \tilde{\mathbf{Q}} (\mathbf{I} - \tilde{\mathbf{R}}_{k-1}(\mathbf{R}_{k-1}^{\mathrm{T}} \mathbf{Q} \mathbf{R}_{k-1})^{-1} \tilde{\mathbf{R}}_{k-1}^{\mathrm{T}}) \tilde{\mathbf{Q}}^{\mathrm{T}} \mathbf{X}^{\mathrm{T}} Y$ which simplifies to the desired result.

Following this, the proof of the first part is a straightforward extension of Theorem 1 and Proposition 1 in Ref. 24. The proof for the second part follows from combining the arguments in Proposition 1 and those in the proof of Theorem 2 in Ref. 24.
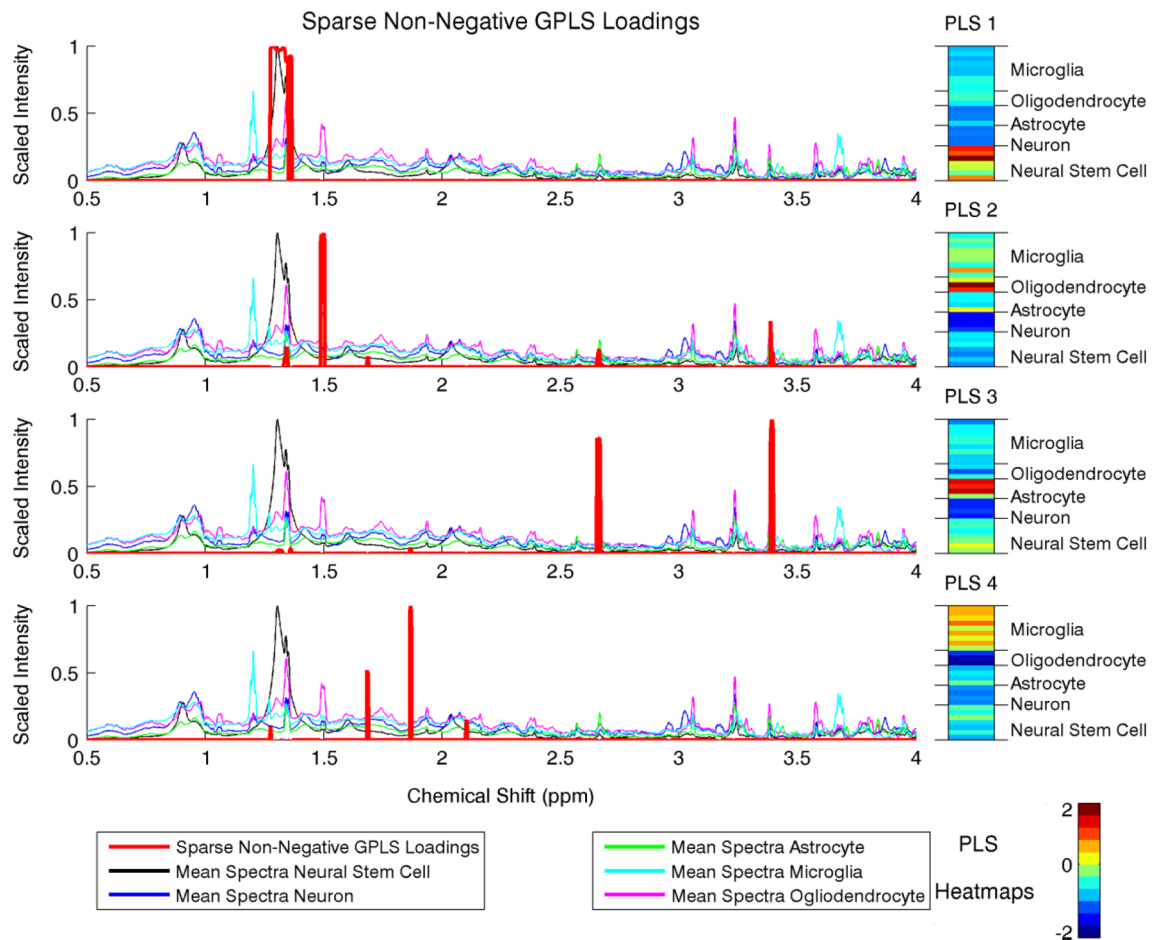
Fig. 1.
Sparse non-negative GPLS loadings and sample PLS heatmaps for the neural cell NMR data. The loadings are superimposed on the mean scaled spectral intensities for each class of neural cells. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

Fig. 2.
The first two PLS and Sparse PLS [4] loadings and sample PLS heatmaps for the neural cell NMR data. The loadings are superimposed on the mean scaled spectral intensities for each class of neural cells. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]
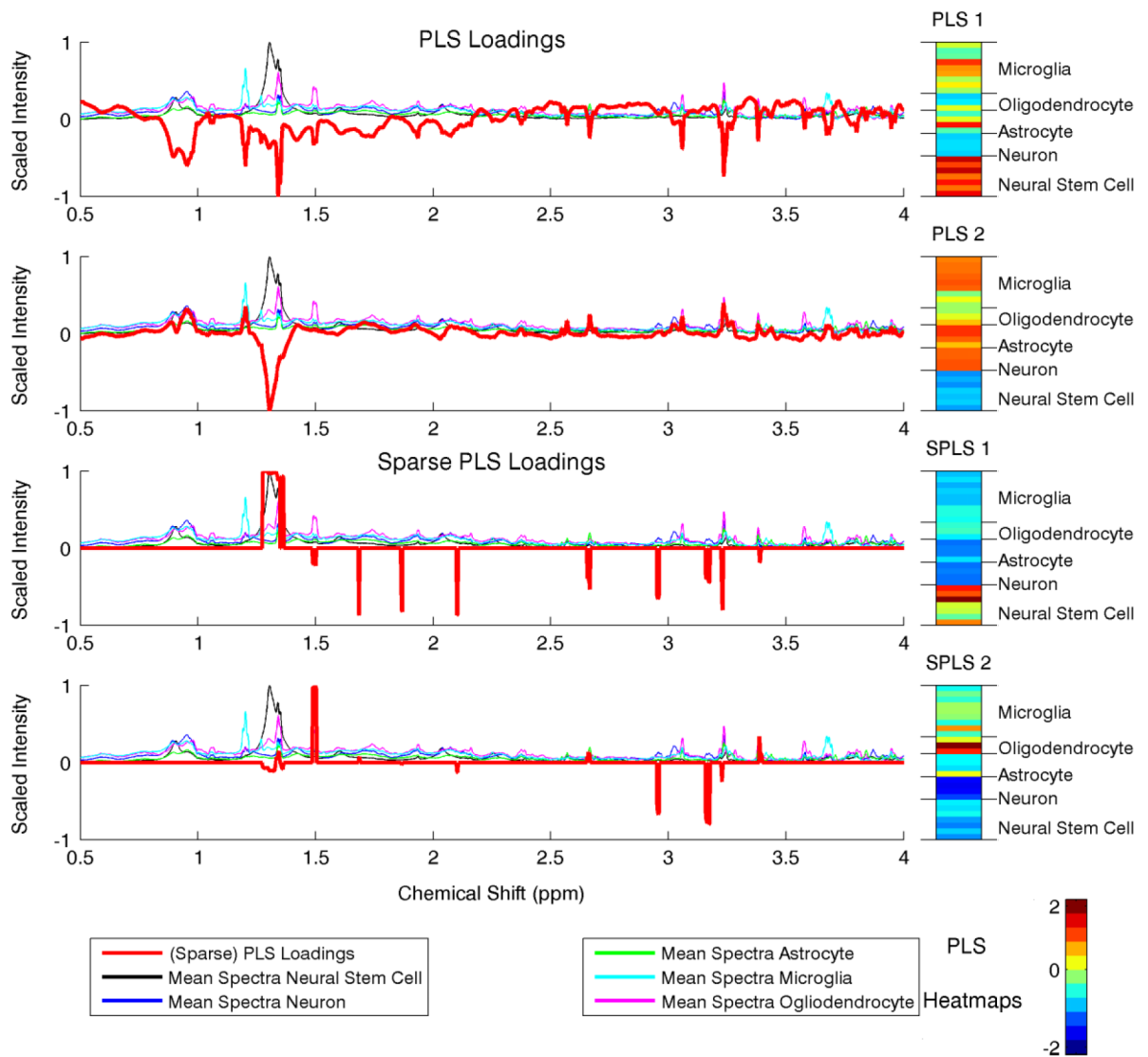
Table 1

Comparison of mean squared prediction error (MSPE), true positive rate (TPR) and false positive rate (FPR) with standard errors (SE).

Simulation 1: $n = 400$, $p = 40$, SNR = 10

Method	MSPE (SE)	TPR (SE)	FPR (SE)
PLS	504.2 (293.8)		
Sparse PLS	72.6 (4.1)	1.00 (0.00)	0.61 (0.27)
RPLS	66.4 (3.8)	1.00 (0.00)	0.22 (0.35)
Lasso	70.9 (4.9)	0.60 (0.07)	0.00 (0.02)
Elastic net	70.5 (4.5)	0.61 (0.07)	0.01 (0.03)

Simulation 2: $n = 400$, $p = 40$, SNR = 5

Method	MSPE (SE)	TPR (SE)	FPR (SE)
PLS	655.2 (212.9)		
Sparse PLS	143.7 (9.8)	1.00 (0.00)	0.66 (0.29)
RPLS	131.4 (9.3)	1.00 (0.00)	0.19 (0.37)
Lasso	139.3 (9.5)	0.49 (0.07)	0.00 (0.00)
Elastic net	139.0 (9.5)	0.50 (0.07)	0.00 (0.00)

Simulation 3: $n = 40$, $p = 80$, SNR = 10

Method	MSPE (SE)	TPR (SE)	FPR (SE)
PLS	624.1 (256.5)		
Sparse PLS	104.9 (26.3)	0.99 (0.05)	0.77 (0.30)
RPLS	76.0 (20.8)	1.00 (0.00)	0.45 (0.43)
Lasso	83.7 (19.7)	0.17 (0.04)	0.02 (0.06)
Elastic net	82.4 (18.6)	0.17 (0.03)	0.02 (0.04)

Simulation 4: $n = 40$, $p = 80$, SNR = 5

Method	MSPE (SE)	TPR (SE)	FPR (SE)
PLS	612.6 (256.8)		
Sparse PLS	206.4 (53.9)	0.98 (0.07)	0.70 (0.31)
RPLS	155.1 (59.0)	1.00 (0.00)	0.52 (0.43)
Lasso	178.3 (49.7)	0.12 (0.04)	0.01 (0.02)
Elastic net	172.7 (46.0)	0.12 (0.04)	0.01 (0.03)

Simulation 5: $n = 40$, $p = 200$, SNR = 10

Method	MSPE (SE)	TPR (SE)	FPR (SE)
PLS	649.7 (175.8)		
Sparse PLS	85.7 (20.4)	1.00 (0.02)	0.76 (0.32)
RPLS	84.8 (52.2)	1.00 (0.00)	0.53 (0.47)
Lasso	83.1 (15.1)	0.19 (0.04)	0.01 (0.01)
Elastic net	82.2 (14.7)	0.19 (0.05)	0.00 (0.01)

Simulation 2: $n = 40$, $p = 200$, SNR = 5

Method	MSPE (SE)	TPR (SE)	FPR (SE)
PLS	691.9 (188.5)		
Sparse PLS	182.0 (42.0)	0.98 (0.08)	0.61 (0.36)
RPLS	153.3 (40.9)	1.00 (0.00)	0.48 (0.49)
Lasso	163.1 (35.7)	0.15 (0.04)	0.00 (0.01)
Elastic net	160.6 (33.7)	0.15 (0.03)	0.01 (0.02)

Table 2

Comparison of mean squared prediction error (MSPE) with standard errors (SE) for multivariate methods.

Simulation 1: $n = 400, p = 40,$ SNR = 2		Simulation 2: $n = 400, p = 40,$ SNR = 1	
Method	**MSPE (SE)**	**Method**	**MSPE (SE)**
SPCA	2376.2 (337)	SPCA	2204.1 (313)
PLS	2567.7 (316)	PLS	2343.3 (281)
RPLS	404.7 (96)	RPLS	339.4 (175)

Simulation 3: $n = 40, p = 80,$ SNR = 2		Simulation 4: $n = 40, p = 80,$ SNR = 1	
Method	**MSPE (SE)**	**Method**	**MSPE (SE)**
SPCA	711.3 (107)	SPCA	721.7 (109)
PLS	647.0 (101)	PLS	659.9 (81)
RPLS	14.2 (3)	RPLS	13.3 (3)

Simulation 5: $n = 40, p = 200,$ SNR = 2		Simulation 6: $n = 40, p = 200,$ SNR = 1	
Method	**MSPE (SE)**	**Method**	**MSPE (SE)**
SPCA	1269.9 (211)	SPCA	1263.1 (208)
PLS	830.0 (102)	PLS	843.7 (87)
RPLS	20.3 (8)	RPLS	19.4 (7)

Table 3

Misclassification errors for methods applied to the neural cell NMR data. Various methods were used to first reduce the dimension with the resulting factors used as predictors in linear discriminant analysis.

	Training error	Leave-one-out CV error
Naive Bayes Classifier	0.0370	0.1481
PCA + LDA	0.1167	0.1852
PLS + LDA [6]	0.0000	0.1481
GPCA + LDA [24]	0.1833	0.1481
GPLS + LDA	0.0000	0.1111
SPCA + LDA [22]	0.1167	0.1481
SPLS + LDA [4]	0.0000	0.1111
SGPCA + LDA [36]	0.1833	0.1481
SGPLS + LDA	0.0000	0.0741

Table 4

Timings comparisons. Time in seconds to compute the entire solution path for the neural cell NMR data.

	Time in seconds
Sparse PLS (via RPLS in C)	1.01
Sparse PLS (via RPLS in Matlab)	57.01
Sparse PLS (R package spls)	1033.86
Sparse Non-negative GPLS (in C)	28.16