Spatially Weighted Principal Component Regression for High-dimensional Prediction

Dan Shen[1] and Hongtu Zhu[2]

[1] Interdisciplinary Data Sciences Consortium, Department of Mathematics and Statistics, University of South Florida, Tampa, FL, USA
danshen@usf.edu

[2] Department of Biostatistics and Biomedical Research Imaging Center, University of North Carolina at Chapel Hill, Chapel Hill, NC, USA
htzhu@email.unc.edu

Abstract. We consider the problem of using high dimensional data residing on graphs to predict a low-dimensional outcome variable, such as disease status. Examples of data include time series and genetic data measured on linear graphs and imaging data measured on triangulated graphs (or lattices), among many others. Many of these data have two key features including spatial smoothness and intrinsically low dimensional structure. We propose a simple solution based on a general statistical framework, called spatially weighted principal component regression (SWPCR). In SWPCR, we introduce two sets of weights including importance score weights for the selection of individual features at each node and spatial weights for the incorporation of the neighboring pattern on the graph. We integrate the importance score weights with the spatial weights in order to recover the low dimensional structure of high dimensional data. We demonstrate the utility of our methods through extensive simulations and a real data analysis based on Alzheimer's disease neuroimaging initiative data.

Keywords: Graph; Principal component analysis; Regression; Spatial; Supervise; Weight.

1 Introduction

Our problem of interest is to predict a set of response variables $\mathbf{Y}$ by using high-dimensional data $\mathbf{x} = \{\mathbf{x}_g : g \in \mathcal{G}\}$ measured on a graph $\zeta = (\mathcal{G}, \mathcal{E})$, where $\mathcal{E}$ is the edge set of ζ and $\mathcal{G} = \{g_1, \ldots, g_m\}$ is a set of vertexes, in which m is the total number of vertexes in $\mathcal{G}$. The response $\mathbf{Y}$ may include cognitive outcome, disease status, and the early onset of disease, among others. Standard graphs including both directed and undirected graphs have been widely used to build complex patterns [10]. Examples of graphs are linear graphs, tree graphs, triangulated graphs, and 2-dimensional (2D) (or 3-dimensional (3D)) lattices, among many others (Figure 1). Examples of $\mathbf{x}$ on the graph $\zeta = (\mathcal{G}, \mathcal{E})$ include time series and genetic data measured on linear graphs and imaging data measured on

triangulated graphs (or lattices). Particularly, various structural and functional neuroimaging data are frequently measured in a 3D lattice for the understanding of brain structure and function and their association with neuropsychiatric and neurodegenerative disorders [9].

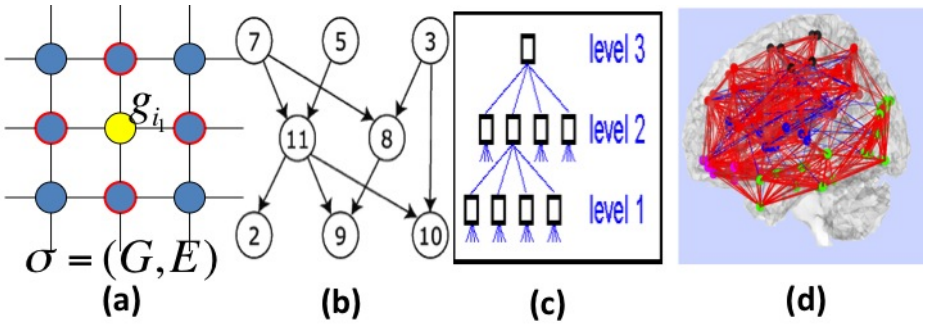

Fig. 1. Illustration of graph data structure $\zeta = (\mathcal{G}, \mathcal{E})$: (a) two-dimensional lattice; (b) acyclic directed graph; (c) tree; (d) undirected graph.

The aim of this paper is to develop a new framework of spatially weighted principal component regression (SWPCR) to use $\mathbf{x}$ on graph $\zeta = \{\mathcal{G}, \mathcal{E}\}$ to predict $\mathbf{Y}$. Four major challenges arising from such development include *ultra-high dimensionality*, *low sample size*, *spatially correlation*, and *spatial smoothness*. SWPCR is developed to address these four challenges when high-dimensional data on graphs ζ share two important features including spatial smoothness and intrinsically low dimensional structure. Compared with the existing literature, we make several major contributions as follows:

- (i) SWPCR is designed to efficiently capture the two important features by using some recent advances in smoothing methods, dimensional reduction methods, and sparse methods.
- (ii) SWPCR provides a powerful dimension reduction framework for integrating feature selection, smoothing, and feature extraction.
- (iii) SWPCR significantly outperforms the competing methods by simulation studies and the real data analysis.

2 Spatially Weighted Principal Component Regression

In this section, we first describe the graph data that are considered in this paper. We formally describe the general framework of SWPCR.

2.1 Graph Data

Consider data from n independent subjects. For each subject, we observe a $q \times 1$ vector of discrete or continuous responses, denoted by $\mathbf{y}_i = (\mathbf{y}_{i,1}, \ldots, \mathbf{y}_{i,q})^T$, and

a $m \times 1$ vector of high dimensional data $\mathbf{x}_i = \{\mathbf{x}_{i,g} : g \in \mathcal{G}\}$ for $i = 1, \ldots, n$. In many cases, q is relatively small compared with n, whereas m is much larger than n. For instance, in many neuroimaging studies, it is common to use ultra-high dimensional imaging data to classify a binary class variable. In this case, $q = 1$, whereas m can be several million number of features. In many applications, $\mathcal{G} = \{g_1, \ldots, g_m\}$ is a set of prefixed vertexes, such as voxels in 2D or 3D lattices, whereas the edge set $\mathcal{E}$ may be either prefixed or determined by $\mathbf{x}_i$ (or other data).

2.2 SWPCR

We introduce a three-stage algorithm for SWPCR to use high-dimensional data $\mathbf{x}$ to predict a set of response variables $\mathbf{Y}$. The key stages of SWPCR can be described as follows.

- Stage 1. Build an importance score vector (or function) $W_I : \mathcal{G} \to R^+$ and the spatial weight matrix (or function) $W_E : \mathcal{G} \times \mathcal{G} \to R$.
- Stage 2. Build a sequence of scale vectors $\{\mathbf{s}_0 = (\mathbf{s}_{E,0}, \mathbf{s}_{I,0}), \cdots, \mathbf{s}_L = (\mathbf{s}_{E,L}, \mathbf{s}_{I,L})\}$ ranging from the smallest scale vector $\mathbf{s}_0$ to the largest scale vector $\mathbf{s}_L$. At each scale vector $\mathbf{s}_\ell$, use generalized principal component analysis (GPCA) to compute the first few principal components of an $n \times m$ matrix $X = (\mathbf{x_1} \cdots \mathbf{x_n})^\mathbf{T}$, denoted by $A(\mathbf{s}_\ell)$, based on $W_E(\cdot, \cdot)$ and $W_I(\cdot)$ for $\ell = 0, \ldots, L$.
- Stage 3. Select the optimal $0 \le \ell^* \le L$ and build a prediction model (e.g., high-dimensional linear model) based on the extracted principal components $A(\mathbf{s}_{\ell^*})$ and the responses $\mathbf{Y}$.

We slightly elaborate on these stages. In Stage 1, the important scores $w_{I,g}$ play an important feature screening role in SWPCR. Examples of $w_{I,g} = W_I(g)$ in the literature can be generated based on some statistics (e.g., Pearson correlation or distance correlation) between $\mathbf{x}_g$ and $\mathbf{Y}$ at each vertex g. For instance, let $p(g)$ be the Pearson correlation at each vertex g and then define

$$w_{I,g} = -m \log(p(g)) / \left[-\sum_{g \in \mathcal{G}} \log(p(g)) \right]. \tag{1}$$

In Stage 1, without loss of generality, we focus on the symmetric matrix $W_E = (w_{E,gg'}) \in R^{p \times p}$ throughout the paper. The element $w_{E,gg'}$ is usually calculated by using various similarity criteria, such as Gaussian similarity from Euclidean distance, local neighborhood relationship, correlation, and prior information obtained from other data [21]. In Section 2.3, we will discuss how to determine W_E and W_I while explicitly accounting for the complex spatial structure among different vertexes.

In Stage 2, at each scale vector $\mathbf{s}_\ell = (\mathbf{s}_{E,\ell}, \mathbf{s}_{I,\ell})$, we construct two matrices, denoted by $Q_{E,\ell}$ and $Q_{I,\ell}$ based on W_E and W_I as follows:

$$Q_{E,\ell} = F_1(W_E, \mathbf{s}_{E,\ell}) \quad \text{and} \quad Q_{I,\ell} = \text{diag}(F_2(W_I, s_{I,\ell})), \tag{2}$$

where $F_1 : R^{p\times p} \times R^+ \to R^{p\times p}$ and $F_2 : R^p \times R^+ \to R^p$ are two known functions. For instance, let $\mathbf{1}(\cdot)$ be an indicator function, we may set

$$F_2(W_I, s_{I,\ell}) = (\mathbf{1}(w_{I,g_1} \geq s_{I,\ell}), \cdots, \mathbf{1}(w_{I,g_m} \geq s_{I,\ell}))^T, \tag{3}$$

to extract 'significant' vertexes. There are various ways of constructing $Q_{E,\ell}$. For instance, one may set $Q_{E,\ell}$ as

$$Q_{E,\ell} = (|w_{E,gg'}|\mathbf{1}(|w_{E,gg'}| \geq s_{E,\ell;1}, D(g,g') \leq s_{E,\ell;2})),$$

where $\mathbf{s}_{E,\ell} = (s_{E,\ell;1}, s_{E,\ell;2})^T$ and $D(g,g')$ is a graph-based distance between vertexes g and g'. The value of $s_{E,\ell;2}$ controls the number of vertexes in $\{g' \in \mathcal{G} : D(g,g') \leq s_{E,\ell;2}\}$, which is a patch set at vertex g [18], whereas $s_{E,\ell;1}$ is used to shrink small $|w_{E,gg'}|$s into zero.

After determining $Q_{E,\ell}$ and $Q_{I,\ell}$, we set $\Sigma_c = Q_{E,\ell}Q_{I,\ell}Q_{I,\ell}^TQ_{E,\ell}^T$ and $\Sigma_r = I_n$ for independent subjects. Let $\widetilde{\mathbf{X}}$ be the centered matrix of $\mathbf{X}$. Then we can extract K principal components through minimize the following objective function given by

$$||\widetilde{\mathbf{X}} - UDV^T||^2 \text{ subject to } U^T\Sigma_r U = V^T\Sigma_c V = I_K \text{ and } \text{diag}(D) \geq 0. \tag{4}$$

If we consider correlated observations from multiple subjects, we may use Σ_r to explicitly model their correlation structure. The solution (U_ℓ, D_ℓ, V_ℓ) of the objective function (4) at $\mathbf{s}_\ell$ is the SVD of $\widetilde{\mathbf{X}}_{R,\ell} = \widetilde{\mathbf{X}}Q_{E,\ell}Q_{I,\ell}$. The we can use a GPCA algorithm to simultaneously calculate all components of (U_ℓ, D_ℓ, V_ℓ) for a fixed K as follows. In practice, a simple criterion for determining K is to include all components up to some arbitrary proportion of the total variance, say 85%.

For ultra-high dimensional data, we consider a regularized GPCA to generate (U_ℓ, D_ℓ, V_ℓ) by minimizing the following objective function

$$||\widetilde{\mathbf{X}}_{R,\ell} - \sum_{k=1}^{K} d_{k,\ell}\mathbf{u}_{k,\ell}\mathbf{v}_{k,\ell}^T||^2 + \lambda_{\mathbf{u}}\sum_{k=1}^{K} P_1(d_{k,\ell}\mathbf{u}_{k,\ell}) + \lambda_{\mathbf{v}}\sum_{k=1}^{K} P_2(d_{k,\ell}\mathbf{v}_{k,\ell}) \tag{5}$$

subject to $\mathbf{u}_{k,\ell}^T\mathbf{u}_{k,\ell} \leq 1$ and $\mathbf{v}_{k,\ell}^T\mathbf{v}_{k,\ell} \leq 1$ for all k, where $\mathbf{u}_{k,\ell}$ and $\mathbf{v}_{k,\ell}$ are respectively the k-th column of U_ℓ and V_ℓ. We use adaptive Lasso penalties for $P_1(\cdot)$ and $P_2(\cdot)$ and then iteratively solve (5) [1]. For each k_0, we define $\mathbf{E}_{\ell,k_0} = \widetilde{\mathbf{X}}_{R,\ell} - \sum_{k\neq k_0} d_{k,\ell}\mathbf{u}_{k,\ell}\mathbf{v}_{k,\ell}^T$ and minimize

$$||\mathbf{E}_{\ell,k_0} - d_{k_0,\ell}\mathbf{u}_{k_0,\ell}\mathbf{v}_{k_0,\ell}^T||^2 + \lambda_{\mathbf{u}}P_1(d_{k_0,\ell}\mathbf{u}_{k_0,\ell}) + \lambda_{\mathbf{v}}P_2(d_{k_0,\ell}\mathbf{v}_{k_0,\ell}) \tag{6}$$

subject to $\mathbf{u}_{k_0,\ell}^T\mathbf{u}_{k_0,\ell} \leq 1$ and $\mathbf{v}_{k_0,\ell}^T\mathbf{v}_{k_0,\ell} \leq 1$. By using the sparse method in [12], we can calculate the solution of (6), denoted by $(\hat{d}_{k_0,\ell}, \hat{\mathbf{u}}_{k_0,\ell}, \hat{\mathbf{v}}_{k_0,\ell})$. In this way, we can sequentially compute $(\hat{d}_{k,\ell}, \hat{\mathbf{u}}_{k,\ell}, \hat{\mathbf{v}}_{k,\ell})$ for $k = 1, \ldots, K$.

In Stage 3, select ℓ^* as the minimum point of the objective function (5) or (6) . let $Q_{F,\ell^*} = Q_{E,\ell^*}Q_{I,\ell^*}V_{\ell^*}D_{\ell^*}^{-1}$ and then K principal components $A(\mathbf{s}_{\ell^*}) = \mathbf{X}Q_{F,\ell^*}$. Moreover, K is usually much smaller than $\min(n,m)$. Then, we build a

regression model with $\mathbf{y}_i$ as responses and A_i (the i-th row of $A(\mathbf{s}_{\ell^*})$) as covariates, denoted by $R(\mathbf{y}_i, A_i; \boldsymbol{\theta})$, where $\boldsymbol{\theta}$ is a vector of unknown (finite-dimensional or nonparametric) parameters. Specifically, based on $\{(\mathbf{y}_i, A_i)\}_{i\geq 1}$, we use an estimation method to estimate $\boldsymbol{\theta}$ as follows:

$$\widehat{\boldsymbol{\theta}} = \text{argmin}_{\boldsymbol{\theta}}\{\rho\left(R, \boldsymbol{\theta}, \{(\mathbf{y}_i, A_i)\}_{i\geq 1}\right) + \lambda P_3(\boldsymbol{\theta})\},$$

where $\rho(\cdot,\cdot,\cdot)$ is a loss function, which depends on both the regression model and the data, and $P_3(\cdot)$ is a penalty function, such as Lasso. This leads to a prediction model $R(\mathbf{y}_i, A_i; \boldsymbol{\theta})$. For instance, for binary response $\mathbf{y}_i = 1$ or 0, we may consider a sparse logistic model given by $\text{logit}(P(\mathbf{y}_i = 1|A_i)) = A_i^T\boldsymbol{\theta}$ for $R(\mathbf{y}_i, A_i; \boldsymbol{\theta})$.

Given a test feature vector $\mathbf{x}^*$, we can do predictions from our prediction model as follows:

- Center each component of $\mathbf{x}^*$ by calculating $\tilde{\mathbf{x}}^* = \mathbf{x}^* - \hat{\mu}_{\mathbf{x}}$, in which $\hat{\mu}_{\mathbf{x}}$ is the mean and learnt from the training data;
- Optimize an objective function based on $R(\mathbf{y}, \tilde{\mathbf{x}}^{*T}Q_{F,\ell^*}; \widehat{\boldsymbol{\theta}})$ to calculate an estimate of $\mathbf{y}$, denoted by $\hat{\mathbf{y}}^*$.

Our prediction model is applicable to various regression settings for continuous and discrete responses and multivariate and univariate responses, such as survival data and classification problems.

2.3 Importance Score Weights and Spatial Weights

There are two sets of weights in SWPCR including (i) importance score weights enabling a selective treatment for individual features, and (ii) spatial weights accommodating the underlying spatial dependence among features across neighboring vertexes on graph. Below, we propose the strategy of determining both importance score weights and spatial weights.

Importance Score Weights As discussed in Section 2.3, at each vertex g, $w_{I,g}$, such as the Pearson correlation in (1), is calculated based on a statistical model between $\mathbf{x}_g$ and $\mathbf{Y}$ in order to perform feature selection according to each feature's discriminative importance. Statistically, most existing methods use a marginal (or vertex-wise) model by assuming

$$p(\mathbf{x}_i, \mathbf{y}_i) = \prod_{g\in\mathcal{G}} p(\mathbf{x}_{i,g}, \mathbf{y}_i; \boldsymbol{\beta}(g)),$$

where $\boldsymbol{\beta} = (\boldsymbol{\beta}(g) : g \in \mathcal{G})$ and $\boldsymbol{\beta}(g)$ is introduced to quantify the association between $\mathbf{y}_i$ and $\mathbf{x}_{i,g}$ at each vertex $g \in \mathcal{G}$. At the $g-$th vertex, $w_{I,g}$ is a statistic based on the marginal model $\prod_{i=1}^n p(\mathbf{x}_{i,g}, \mathbf{y}_i; \boldsymbol{\beta}(g))$. However, those $w_{I,g}$s largely ignore complex spatial structure, such as homogenous patches defined below, across all vertexes on graph.

For a graph $\zeta = (\mathcal{G}, \mathcal{E})$, it is common to assume that $\boldsymbol{\beta}(g)$ across all vertexes are naturally clustered into P homogeneous patches, denoted by $\{\mathcal{G}_l : l = 1, \ldots, P\}$, such that $P << m$, $\mathcal{G} = \cup_{l=1}^{P} \mathcal{G}_l$, and $\boldsymbol{\beta}(g)$ varies smoothly in each $\mathcal{G}_l$. Note that a patch $\mathcal{G}_l$ consists of a set of vertexes that are completely connected through edges in $\mathcal{E}$. That is, if $g, g' \in \mathcal{G}_l$, then there is a sequence of vertexes $g_0 = g, \cdots, g_M = g'$ in $\mathcal{G}_p$ such that $(g_{j-1}, g_j) \in \mathcal{E}$ for all $j = 1, \ldots, M$. It has been shown that for graph data, algorithms based on patch information have led to state-of-the art techniques for classification and denoising. See for example, [18] for overviews of imaging patches.

We propose the strategy to jointly model $\mathbf{x}_i$ and $\mathbf{y}_i$ and simultaneously calculate $w_{I,g}$ across all vertexes, while learning the homogenous patches $\mathcal{G}_l$. The strategy is to model the conditional distribution of $\mathbf{x}_i$ given $\mathbf{y}_i$, denoted by $p(\mathbf{x}_i|\mathbf{y}_i, \boldsymbol{\beta})$. Then we can learn the patches $\mathcal{G}_l$ in $\mathcal{G}$ from the estimated $\boldsymbol{\beta}$.

Here we consider a set of vertexes $\mathcal{G}$ with unknown edge information $\mathcal{E}$. It is important to learn the homogeneous patches $\mathcal{G}_p$ and then form the edge set $\mathcal{E}$. Let $\mathcal{E}_g(h)$ be an edge set at scale h at each vertex g. We consider a sequence of nested edge sets across multiscales h_s such that $h_0 = 0 \leq h_1 \leq \cdots \leq h_S$ and $\mathcal{E}_g(h_0) = \{g\} \subset \cdots \subset \mathcal{E}_g(h_S)$. To learn the homogeneous patches, a general framework of Multiscale Adaptive Regression Model (MARM) developed in [13] is to maximize a sequence of weighted functions as follows:

$$\hat{\boldsymbol{\beta}}(g; h_s) = \text{argmax}_{\boldsymbol{\beta}(g)} \sum_{i=1}^{n} \sum_{g' \in \mathcal{E}_g(h_s)} \omega(g, g'; h_s) \log p(\mathbf{x}_{i,g'}|\mathbf{y}_i, \boldsymbol{\beta}(g)) \quad \text{for} \quad s = 1, \ldots, S, \tag{7}$$

where $\omega(g, g'; h)$ characterizes the similarity between the data in vertexes g' and g with $\omega(g, g; h) = 1$. If $\omega(g, g'; h) \approx 0$, then the observations in vertex g' do not provide information on $\boldsymbol{\beta}(g)$. Therefore, $\omega(g, g'; h)$ can prevent incorporation of vertexes whose data do not contain information on $\boldsymbol{\beta}(g)$ and preserve the edges of homogeneous regions. Let $D_1(g, g')$ and $D_2(\hat{\boldsymbol{\beta}}(g; h_{s-1}), \hat{\boldsymbol{\beta}}(g'; h_{s-1}))$ be, respectively, the spatial distance between vertexes g and g' and a similarity measure between $\hat{\boldsymbol{\beta}}(g; h_{s-1})$ and $\hat{\boldsymbol{\beta}}(g'; h_{s-1})$. The $\omega(g, g'; h_s)$ can be defined as

$$\omega(g, g'; h_s) = \mathbb{K}_1(D_1(g, g')/h_s) \cdot \mathbb{K}_2(D_2(\hat{\boldsymbol{\beta}}(g; h_{s-1}), \hat{\boldsymbol{\beta}}(g'; h_{s-1}))/\gamma_n), \tag{8}$$

where $\mathbb{K}_1(\cdot)$ and $\mathbb{K}_2(\cdot)$ are two nonnegative kernel functions and γ_n is a bandwidth parameter that may depend on n. See the detailed algorithm of MARM in [13]. After the iteration h_s, we can obtain $\hat{\boldsymbol{\beta}}(g; h_S)$ and its covariance matrix, denoted by $\text{Cov}(\hat{\boldsymbol{\beta}}(g; h_s))$, across all $g \in \mathcal{G}$ and $\omega(g, g'; h_s)$ for all $g' \in \mathcal{E}_g(h_s)$ and $g \in \mathcal{G}$. Finally, we calculate statistics $w_{I,g}$ based on $\hat{\boldsymbol{\beta}}(g; h_s)$ and $\text{Cov}(\hat{\boldsymbol{\beta}}(g; h_s))$, such as the Wald test, and then we use a clustering algorithm, such as the K-mean algorithm, to group $\{\hat{\boldsymbol{\beta}}(g; h_s) : g \in \mathcal{G}\}$ into several homogeneous clusters, in which $\hat{\boldsymbol{\beta}}(g; h_s)$ varies very smoothly in each cluster. Moreover, each homogenous cluster can be a union of several homogeneous patches.

Spatial Weights As discussed in Section 2.3, $w_{E,gg'}$ often characterizes the degree of certain 'similarity' between vertexes g and g'. The locally spatial weight-

ing matrix consists of non-negative weights assigned to the spatial neighboring vertexes of each vertex. It is assumed that

$$w_{E,gg'} = \frac{\omega(g,g';h_s)\mathbf{1}(g' \in \mathcal{E}_g(h_s))}{\sum_{g' \in \mathcal{E}_g(h_s)} \omega(g,g';h_s)\mathbf{1}(g' \in \mathcal{E}_g(h_s))}, \tag{9}$$

in which $\omega(g,g';h_s)$ is defined in (8). Therefore, $w_{E,gg'} = 0$ for all $g' \notin \mathcal{E}_g(h_s)$ and $\sum_{g' \in \mathcal{G}} w_{E,gg'} = 1$. The weights $\mathbb{K}_1(D_1(g,g')/h_s)$ give less weight to vertex $g' \in \mathcal{E}_g(h_s)$, whose location is far from the vertex g. The weights $\mathbb{K}_2(u)$ down-weight the vertex g' with large $D_2(\hat{\boldsymbol{\beta}}(g;h_s), \hat{\boldsymbol{\beta}}(g';h_s))$, which indicates a large difference between $\hat{\boldsymbol{\beta}}(g';h_s)$ and $\hat{\boldsymbol{\beta}}(g;h_s)$. Moreover, by following [4, 13, 15, 16], we set $\mathbb{K}_1(x) = (1-x)_+$ and $\mathbb{K}_2(x) = \exp(-x)$. Although m is often much larger than n, the computational burden associated with the local spatial weights is very minor when h_s is relatively small.

3 Simulation Study

In this section, we conducted one set of simulation study corresponding to binary responses, in order to examine the finite-sample performance of SWPCR in the high-dimensional classification analysis. We demonstrate that SWPCR outperforms many state-of-the-art methods for at least in the simulated dataset.

We simulated $20 \times 20 \times 10$ ($x \times y \times z$) 3D-images from a linear model given by

$$\mathbf{x}_{i,g} = B_0(g) + B_1(g)\mathbf{y}_i + \epsilon_i(g) \quad \text{for} \quad i = 1, \cdots, n, \tag{10}$$

where $\mathbf{y}_i$ is the class label coded as either 0 or 1 and $\epsilon_i(g)$ are random variables with zero mean. The true mean images of class $\mathbf{y}_i = 0$ and class $\mathbf{y}_i = 1$ are

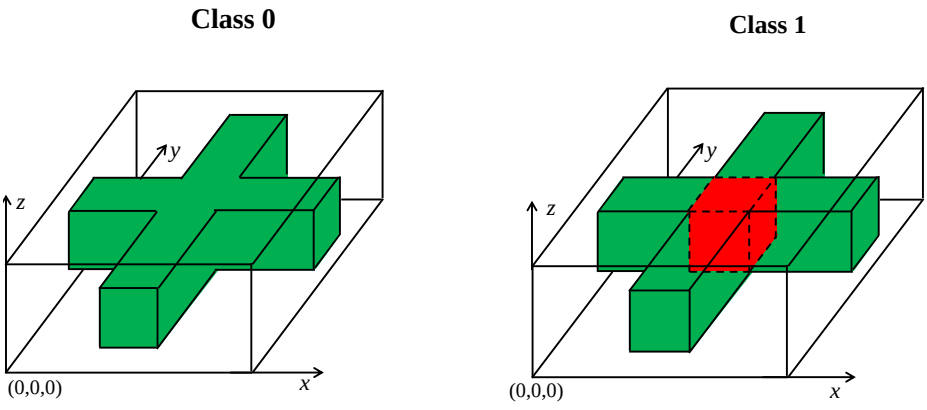

Fig. 2. True mean images for the simulation study: Class 0 in the left panel and Class 1 in the right panel. The white, green, and red colors, respectively, correspond to 0, 1, and 2.

shown in Figure 2. Voxels in the red cuboid region have the maximum difference 1 between classes 0 and 1. The dimension of red cuboid is $3 \times 3 \times 4$ and contains 36 voxels. In this case, $m = 4,000$ and we set $n = 100$ with 60 images from Class 0 and the rest from Class 1. We consider three types of noise $\epsilon_i(g)$ in (10). First, $\epsilon_i^{(1)}(g)$ were independently generated from a $N(0, 2^2)$ generator across all voxels g. Second, $\epsilon_i^{(2)}(g) = \sum_{\|g'-g\| \le 1} \epsilon_i^1(g')/m_g$ were generated from $\epsilon_i^{(1)}(g)$ in order to introduce the short range spatial correlation, where m_g is the number of voxels in the set $\{\| g' - g \| \le 1\}$. Third, to introduce long range spatial correlation, $\epsilon_i^{(3)}(g)$ were generated according to $\epsilon_i^{(3)}(g) = 2\sin(\pi g_1/10)\xi_{i,1} + 2\cos(\pi g_2/10)\xi_{i,2} + 2\sin(\pi g_3/5)\xi_{i,3} + \epsilon_i^{(1)}(g)$, where $\xi_{i,k}$ for $k = 1, 2, 3$ were independenly generated from a $N(0, 1)$ generator. Moreover, the noise variances in all voxels of the red cuboid region equal 4, 4/6, and $4\{\sin(\pi g_1/10)^2 + \cos(\pi g_2/10)^2 + \sin(\pi g_3/5)^2\} + 4$ for Type I, II, and III noises, respectively. Therefore, among the three types of noise, Type III noise has the smallest signal-to-noise ratio and Type II noise has the largest one.

Table 1. Classification results for the first set of simulations: comparison between SWPCR and other Classification Methods. sLDA denotes sparse linear discriminant analysis; SPLS denotes sparse partial least squares; SLR denotes sparse logistic regression; SVM denotes support vector machine; ROAD denotes regularized optimal affine discriminant; and PCA denotes principal component analysis.

Noise	sLDA	SPLS	SLR	SVM	ROAD	PCA	SWPCR
Type I	0.28	0.43	0.45	0.38	0.36	0.36	0.10
Type II	0.27	0.08	0.18	0.26	0.08	0.45	0.03
Type III	0.52	0.30	0.61	0.60	0.50	0.35	0.09

We ran the three stages of SWPCR as follows. In Stage 1, let $\{h_\ell = 1.2^\ell,\ \ell = 0, 1, \ldots, S = 5\}$, and for each $g \in \mathcal{G}$, $w_{I,g} = -m \log(p(g))/\left[-\sum_{g\in\mathcal{G}} \log(p(g))\right]$, where $p(g)$ is the p-value of Wald test $B_1(g) = 0$ in (7) ($\beta(g) = (B_0(g), B_1(g))^T$) for each voxel g. The spatial weight W_E is given by (9). Here we haven't used the simple Pearson correlation (1) for computing weights because it neglects the spatial correlation of the data set. In Stage 2, for each h_ℓ, we define $Q_{E,\ell} = W_E$ and generate $Q_{I,\ell}$ through (2) and (3), where $s_{I,\ell}$ thresholds out the $w_{I,g}$ with $p(g) < 0.01$. Then we extract different K principal components of GPCA to reconstruct the low dimensional representations of simulated images and then do classification analysis. The results are very stable for different number of principal components and here we let $K = 5$. In Stage 3, we tried different classification methods, including linear regression, k-Nearest Neighbor (k-NN) [11] and support vector machine (SVM) [14], on these low dimensional spaces. Based on the misclassification error for the leave-one-out cross validation, the linear regression is slight better than others. The linear regression uses class label

$\mathbf{y}_i$ as dependent variable and principal components as explanatory variables. If the prediction value is less than 0, the image is classified as 0. Otherwise, the image is classified as 1.

We compared SWPCR with other state-of-the-art classification methods. The leave-one-out cross validation is used here to calculate the misclassification rates of the different methods. Other classification methods considered here include sparse linear discriminant analysis (sLDA) [6], sparse partial least squares (SPLS) analysis [5], sparse logistic regression (SLR) [20], SVM, and regularized optimal affine discriminant (ROAD) [8]. These methods are well known for their excellent performance in various simulated and real data sets. Inspecting Table 1 reveal that except SWPCR, all classification methods perform pretty poor, when the signal-to-noise ratio is low in those simulated datasets with Type I and II noises. Except SPLS, PCA, and SWPCR, all other methods are seem to be sensitive to the presence of the long-range correlation structure in Type III noise.

4 Real Data Analysis

4.1 ADNI PET Data

The real data set is the baseline fluorodeoxyglucose positron emission tomography (FDG-PET) data downloaded from the Alzheimer's Disease Neuroimaging Initiative (ADNI) web site (www.loni.ucla.edu/ADNI). The ADNI1 PET data set consists of 196 subjects (102 Normal Controls (NC) and 94 AD subjects). There are three subjects, missing the gender and age information. Among the rest of the subjects, there are 117 males whose mean age is 76.20 years with standard deviation 6.06 years and 76 females whose mean age is 75.29 years with standard deviation 6.29 years.

The dimension of the processed PET images is $79 \times 95 \times 69$. Left panel in Figure 3 shows some selected slices of the processed PET images from 2 randomly selected AD subjects and 2 randomly selected NC subjects.

4.2 Binary Classification

Our first goal is to apply SWPCR in classifying subjects from ADNI1 to AD or CN group based on their FDG-PET images. Such goal is associated with the second primary objective of ADNI aiming at developing new diagnostic methods for AD intervention, prevention, and treatment. Similar as in Section 3 , SWPCR contains the three detailed stages that will not be repeated again. The right panel in Figure 3 is the three view slices of the weight matrix $Q_{I,\ell}$ at the coordinate $(40, 57, 26)$ in the stage 2 of SWPCR. The red region in three slices corresponds to the large important score weight and contains the most classification information.

We compared SWPCR with six other classification methods including sLDA, SPLS, SLR, SVM, ROAD, and PCA. We used their leave-one-out cross validation rates. Table 2 shows the classification results of all the seven methods. sLDA

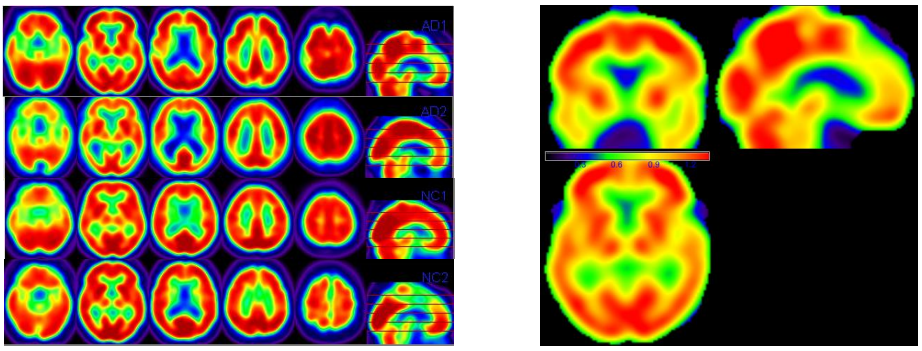

Fig. 3. ADNI1 pet data and the important score weight matrix $Q_{I,\ell}$ in SWPCR. In the left panel, one row sequence of 2-D images belongs to one subject. The first two rows respectively belongs to AD subjects and the rest belongs to NC subjects. In the right panel, the three plots (left -right-bottom) are three view slices of the weight matrix $Q_{I,\ell}$ at the coordinate $(40, 57, 26)$. The red region corresponds to large weight score and contains the most classification information.

performs much worse than all other six methods. ROAD performs slightly better than PCA. SPLS and SVM are comparable with each other, but they outperform SLR and ROAD. SWPCR outperforms all six classification methods. It suggests that the classification performance can be significantly improved by incorporating spatial smoothness and simple dimension reductions methods, such as PCA.

4.3 Age Prediction

Our second goal is to apply SWPCR in predicting subjects' age based on their FDG-PET images. The response variable $\mathbf{y}$ is the age of the subjects and the explanatory variables are the latent scores, extracted from image data. It is very interesting to use memory test scores as the response variable $\mathbf{y}$. However, the data set here contains no such information. The three subjects without the age information are deleted and then we have 193 images left. $\mathbf{y}_i$ in model (10) becomes age of the subjects. Here we will not repeat the detailed stages of SWPCR again, which is similar as in Section 3. The slight difference is stage 3. Here we run regression rather than classification methods between age and the SWPCR latent scores.

Table 2. Misclassification Rates of Different Methods for ADNI 1 Pet Data

sLDA	**SPLS**	**SLR**	**SVM**	**ROAD**	**PCA**	**SWPCR**
0.255	0.163	0.179	0.168	0.189	0.194	0.117

First, we compared SWPCR with three other dimensional reduction methods including PCA, weighted PCA (WPCA) [17], and supervised PCA (SPCA) [2]. We used the leave-one-out cross validation to compute the prediction errors of all the four methods. Let $\hat{\mathbf{y}}_i$ be the fitted response value based on the regression model, and the prediction error is defined as $|\hat{\mathbf{y}}_i - \mathbf{y}_i|/|\mathbf{y}_i|$. Subsequently, we calculated the error difference between SWPCR and all three other methods across different numbers ($K = 5, 7, 10$) of principal components. Panels (a)–(c) in Figure 4 show the boxplots of the error difference between SWPCR and PCA, WPCA, and SPCA, respectively. The error differences are almost always less than 0 (under the dashed line) and these results show the better performance of SWPCR in dimension reduction.

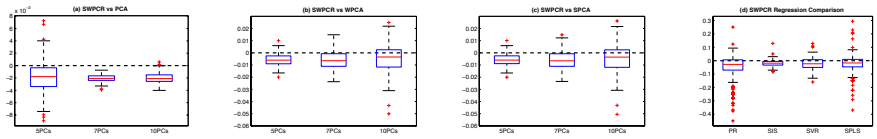

Fig. 4. Performance of SWPCR Regression for ADNI 1 Pet Data. Panels (a)–(c) shows the boxplots of error difference between SWPCR and PCA (WPCA and SPCA). Panel (d) compares SWPCR regression with several other regression methods, including PR, SIS, SVR and SPLS.

Second, we compared SWPCR with several other high-dimensional regression methods including penalized regression (PR) [19], sure independence screening (SIS) regression [7], support vector regression (SVR) [3], and SPLS [5]. Panel (d) in Figure 4 shows the boxplots of the prediction error difference between SWPCR and all the other regression methods. The analysis results further confirm the better performance of SWPCR in regression.

5 Discussion

SWPCR enables a selective treatment of individual features, accommodates the complex dependence among features of graph data, and has the ability of utilizing the underlying spatial pattern possessed by image data. SWPCR integrates feature selection, smoothing, and feature extraction in a single framework. In the simulation studies and real data analysis, SWPCR shows substantial improvement over many state-of-the-art methods for high-dimensional problems.

Acknowledgements. This work was partially supported by the Startup Fund of University of South Florida, NIH grants MH086633, RR025747, and MH092335 and NSF grants SES-1357666 and DMS-1407655.

References

1. Aharon, M., Elad, M., Bruckstein, A.: K-svd: an algorithm for designing overcomplete dictionaries for sparse representation. IEEE Trans. on Signal Processing 54,

4311–4322 (2006)
2. Bair, E., Hastie, T., Paul, D., Tibshirani, R.: Prediction by supervised principal components. Journal of the American Statistical Association 101(473), 119–137 (2006)
3. Basak, D., Pal, S., Patranabis, D.C.: Support vector regression. Neural Information Processing-Letters and Reviews 11(10), 203–224 (2007)
4. Buades, A., Coll, B., Morel, J.M.: A non-local algorithm for image denoising. In: Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on. vol. 2, pp. 60–65. IEEE (2005)
5. Chun, H., Keles, S.: Sparse partial least squares regression for simultaneous dimension reduction and variable selection. J. Roy. Statist. Soc. Ser. B 72,, 3–25 (2010)
6. Clemmensen, L., Hastie, T., Witten, D., Ersbøll, B.: Sparse discriminant analysis. Technometrics 53(4), 406–413 (2011)
7. Fan, J., Lv, J.: Sure independence screening for ultrahigh dimensional feature space. Journal of the Royal Statistical Society: Series B (Statistical Methodology) 70(5), 849–911 (2008)
8. Fan, J., Feng, Y., Tong, X.: A road to classification in high dimensional space: the regularized optimal affine discriminant. Journal of the Royal Statistical Society: Series B (Statistical Methodology) 74(4), 745–771 (2012)
9. Friston, K.J.: Modalities, modes, and models in functional neuroimaging. Science 326, 399–403 (2009)
10. Grenander, U., Miller, M.I.: Pattern Theory From Representation to Inference. Oxford University Press (2007)
11. Hastie, T., Tibshirani, R., Friedman, J.: The Elements of Statistical Learning: Data Mining, Inference, and Prediction (2nd). Springer, Hoboken, New Jersey. (2009)
12. Lee, M., Shen, H., Huang, J.Z., Marron, J.S.: Biclustering via sparse singular value decomposition. Biometrics 66, 1087–1095 (2010)
13. Li, Y., Zhu, H., Shen, D., Lin, W., Gilmore, J.H., Ibrahim, J.G.: Multiscale adaptive regression models for neuroimaging data. Journal of the Royal Statistical Society: Series B 73, 559–578 (2011)
14. Lin, Y.: Support vector machines and the bayes rule in classification. Data Mining and Knowledge Discovery 6, 259–275 (2002)
15. Manjón, J.V., Carbonell-Caballero, J., Lull, J.J., García-Martí, G., Martí-Bonmatí, L., Robles, M.: MRI denoising using non-local means. Medical image analysis 12(4), 514–523 (2008)
16. Polzehl, J., Spokoiny, V.G.: Propagation-separation approach for local likelihood estimation. Probab. Theory Relat. Fields 135, 335–362 (2006)
17. Skočaj, D., Leonardis, A., Bischof, H.: Weighted and robust learning of subspace representations. Pattern recognition 40(5), 1556–1569 (2007)
18. Taylor, K.M., Meyer, F.G.: A random walk on image patches. SIAM J. Imaging Sciences 5, 688–725 (2012)
19. Tibshirani, R.: Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society. Series B (Methodological) 58, 267–288 (1996)
20. Yamashita, O.: Quick manual for sparse logistic regression toolbox ver1.2.1: software at `http://www.cns.atr.jp/~oyamashi/SLR_WEB/` (2011)
21. Yan, S., Xu, D., Zhang, B., Zhang, H.J., Yang, Q., Lin, S.: Graph embedding and extensions: a general framework for dimensionality reduction. IEEE Transactions on Pattern Analysis and Machine Intelligence 29, 40–51 (2007)