SPARSE VARIABLE REDUCED RANK REGRESSION VIA STIEFEL OPTIMIZATION

M.O. Ulfarsson† and V. Solo‡

†University of Iceland, Dept. Electrical Eng., Reykjavik, ICELAND
‡University of New South Wales, School of Electrical Eng., Sydney, AUSTRALIA

Abstract
Reduced rank regression (RRR) has found application in various fields of signal processing. In this paper we propose a novel extension of the RRR model which we call sparse variable reduced rank regression (svRRR). By using a vector $l_1$ penalty we remove variables completely from the RRR. The proposed estimation algorithm involves optimization on the Stiefel manifold and we illustrate it both on a simulated and a real functional magnetic resonance imaging (fMRI) data set.

Index Terms— Reduced rank regression, sparsity, optimization, Stiefel manifold.

1. INTRODUCTION
Multivariate reduced rank regression (RRR) [1] (or equivalently canonical correlation analysis (CCA)) has found practical application in signal processing [2], for instance, in frequency estimation [3], and array processing [4]. The RRR model can be written as

$$y_t = F D U^T x_t + e_t, \quad t = 1, ..., T, \quad (1)$$

where $F = [f_1, ..., f_M]^T$ is an $M_f \times r$ orthogonal loading matrix, $D$ is an $r \times r$ diagonal matrix, $U = [u_1, ..., u_M]^T$ is an $M_u \times r$ orthogonal loading matrix, $\min(M_f, M_u) \geq r$, and $e_t$ is an $M_f \times 1$ error vector. A signal processing interpretation of this model was given by [5, 6]: $x_t$ is an information signal used to send a message $y_t$, but such a message can only be transmitted through $r$ channels. Thus $U^T x_t$ acts as code and on receipt of the code, form $F D U^T x_t$ which is hoped to be as close as possible to $y_t$.

Although sparse signal processing methods such as various kinds of sparse regression [7, 8] and various kinds of sparse PCA [9, 10, 11, 12] have attracted much attention recently, there seem to be only a handful of papers devoted to sparse RRR problems [13, 14].

In this paper we combine automatic variable selection with the RRR model and call the resulting method sparse RRR (svRRR). svRRR is based on penalized optimization with vector $l_1$ penalties that penalize the rows of the $F$ and $U$ loading matrices. The vector $l_1$ penalty is able to zero out variables by zeroing out some of the rows completely.

The proposed estimation algorithm involves optimization on the Stiefel manifold. Recently, computationally efficient computer algorithms have been proposed for manifold optimization [15, 16]. We develop a cyclic descent algorithm that involves a steepest descent on the Stiefel manifold. Since the degree of sparseness depends on the selection of tuning parameters a tuning parameter criterion is developed. Finally, the algorithm is applied on real and simulated data sets. The papers [14, 13] solve different problem to ours; [14] penalizes elements using the scalar $l_1$ penalty, and [13] penalizes elements using the scalar $l_0$ penalty. Because they use scalar penalties these methods retain all variables but zero out some loadings; we can call them sparse loading RRR (siRRR). Whereas our method removes variables completely; hence the name sparse variable RRR (svRRR). This makes it clear that siRRR and svRRR are solving different problems and are not comparable.

2. OPTIMIZATION ON THE STIEFEL MANIFOLD
The constrained optimization problem

$$\min_{x \in St(M,r)} J(X), \quad (2)$$

where $St(M,r) = \{ X \in \mathbb{R}^{M \times r} : X^T X = I_r \}$ and $J(X)$ a criterion of interest occurs frequently in signal processing. The set $St(M,r)$ endowed with its submanifold structure is called the Stiefel manifold [16, 15, 17]. In general an iterative optimization method is needed to solve (2).

The steepest descent method on the Stiefel manifold generates a sequence of iterates $\{X_k\}$ such that each $X_k$ is on the Stiefel manifold. Before we describe how to update an iterate we need few definitions. The tangent space to $St(M,r)$ at $X$ is the set

$$T_X St(M,r) = \{ Z \in \mathbb{R}^{M \times r} : X^T Z + Z^T X = 0 \}.$$ 

The Stiefel gradient at $X$ lies in the tangent space and is given by [15]

$$\nabla J = \frac{\partial J}{\partial X} - X \frac{\partial I}{\partial X^T} X,$$

and the projection operator onto the Stiefel manifold is given by

$$\Pi(X) = P I_{M,r} Q^T,$$

where $X = PLQ^T$ is a singular value decomposition (SVD).

2.1. The steepest descent algorithm on the Stiefel manifold
The steepest descent algorithm on the Stiefel manifold [17] is given by the following steps:

0. Initialize $X_0$.
1. Compute the negative Stiefel gradient $-\nabla J$.
2. Update the iterate $X_k$

$$X_{k+1} = \Pi(X_k - t \nabla J),$$

where the step size $t$ is found by a line search, i.e., it corresponds to the first local maximum of $J(X_k) - J(\Pi(X_k - t \nabla J))$.
3. Repeat steps 1 and 2 until convergence.
3. SPARSE VARIABLE RRR

svRRR is based on solving the following optimization problem

\[
F, D, U = \arg\min_{F,D,U} J(F, D, U)
\]

subject to \(F^T F = I_r\)

\(U^T U = I_r\)

\(D\) is diagonal with non-negative elements,

where

\[
J(F, D, U) = \frac{1}{2T} \sum_{t=1}^{T} \| y_t - FDU^T x_t \|^2
\]

\[+ h_u \sum_{v=1}^{M_f} \| u_v \| + h_f \sum_{v=1}^{M_f} \| f_v \|, \tag{3}\]

where \(h_u\) and \(h_f\) are tuning parameters that control the sparseness.

If \(a = [a_1, \ldots, a_T]^T\) is a vector then we call \(\| a \| = \sqrt{\sum_j a_j^2}\)
the vector \(l_2\) penalty. It has been shown, for example, in \([8, 11]\)
that the penalty is able to zero out vectors. The justification for
the name svRRR is that if for instance the vector \(u_v = 0\) then
(3) does not depend on variable \(v\) of \(x_t\) and can thus be left out.

Note that the penalties are applied to the rows of \(F\) and \(G\) not columns.

3.1. The estimation algorithm

3.1.1. The unpenalized case

In this case the estimates for \(F, D\) and \(U\) can be obtained via the
SVD. Let \(S_{xy} = \frac{1}{T} Y^T X = S_{xy}^T S_{xx} = \frac{1}{T} X^T X\) where \(Y =
[y_1]^T\) and \(X = [x_1]^T\) and take the SVD

\[S_{yx} S_{xx}^{-1} S_{xy} = VA^2 V^T.\]

Let \(V_r\) be a \(M_f \times r\) matrix containing the first \(r\) columns of \(V\)
and take the compact SVD \((Q\) and \(L\) are \(r \times r\) and \(P\) is \(M_u \times r\)).

\[V_r^T S_{yx} S_{xx}^{-1} = QLP^T;\]

then the estimates for \(F, D, U\) are given by

\[F = V_r Q,\]

\[D = L,\]

\[U = P.\]

3.1.2. The penalized case

We employ the cyclic descent algorithm to minimize the cost function (3)

\[0. \text{ Initialize } F_0, D_0 \text{ and } U_0 \text{ with the solution from 3.1.1.}\]

1. (F-step): Solve

\[F_{k+1} = \arg\min_F J(F, D_k, U_k)\]

subject to \(F^T F = I_r\)

2. (D-step): Calculate \(D_{k+1} = \text{diag}(d_1, \ldots, d_r)\) where

\[d_j = \begin{cases} f_{ij}^T S_{yx} u_{ij} > 0, \quad & \text{if } f_{ij}^T S_{yx} u_{ij} > 0, \\ 0, \quad & \text{else}, \end{cases}\]

where \(u_{ij}\), \(f_{ij}\) are the \(j\)-th column vectors of \(U_k\) and \(F_{k+1}\), respectively. The derivation is given in the appendix.

3. (U-step): Solve

\[U_{k+1} = \arg\min_U J_k(F_{k+1}, D_{k+1}, U)\]

subject to \(U^T U = I_r\).

Iterate steps 1 to 3 until convergence.

3.1.3. The F-step

The F-step is equivalent to minimizing

\[J_f(F) = -\text{tr}(S_{xy} FDU^T) + h_f \sum_{v=1}^{M_f} \| f_v \|,\]

subject to \(F^T F = I_r.\) \tag{4}

with respect to \(F\). This can be viewed as an optimization problem on a Stiefel manifold. Since \(f_v\) is not smoothly differentiable we exchange it for the smooth approximation \(\sqrt{\| f_v \|^2 + \rho^2}\) where \(\rho^2\) is a small constant. We use the steepest descent algorithm on the Stiefel manifold given in section 2.1 to optimize \(J_f\). This requires the derivative of \(J_f\) with respect to \(F\) which is given by

\[\frac{\partial J_f}{\partial F} = -S_{xy} UD + h_f AF,\]

where \(A\) is diagonal with \(A_{uv} = \frac{1}{\sqrt{\| u_v \|^2 + \rho^2}}\).

3.1.4. The U-step

The U-step is equivalent to minimizing

\[J_u(U) = -\text{tr}(S_{xy} FDU^T) + \frac{1}{2} \text{tr}(S_{xx} U D^2 U^T) + h_u \sum_{v=1}^{M_u} \| u_v \|,\]

subject to \(U^T U = I_r.\) \tag{5}

As in the F-step we use the steepest descent on the Stiefel manifold.

The derivative of \(J_u\) with respect to \(U\) is given by

\[\frac{\partial J_u}{\partial U_k} = -S_{xy} FD + S_{xx} UD^2 + h_u BU,\]

where \(B\) is a diagonal matrix with \(B_{uv} = \frac{1}{\sqrt{\| u_v \|^2 + \rho^2}}\).

4. TUNING PARAMETER SELECTION

We propose to select the tuning parameters \(h_u, h_f, r\) by minimizing the following cost-complexity criterion

\[
CC_{h_u, h_f, r} = \sum_{i=1}^{T} \| e_i \|^2 + 2\sigma^2 (M_h r + M_f r - r^2) \tag{6}
\]

where \(e_i = y_i - \hat{F} DU^T x_i\) and \(M_{h, f}\) is the number of rows of \(U\) retained when using penalty \(h_u, h_f\). Similarly for \(M_{f, f}\). The noise variance \(\sigma^2\) is estimated as \(\frac{1}{p} \sum_{i=1}^{T} \| e_i \|^2\). The tuning parameters that corresponding to the minimum (or close to it) of \(CC_{h_u, h_f, r}\) are chosen. Note that (6) resembles Mallows’ Cp statistic [18].
5. SIMULATION

The data was simulated according to (1). The loading matrices $F$ and $U$ are $M \times r$ matrices where $M = 50$, $r = 2$ and $D = I_r$. $F$ and $U$ are shown in Fig. 1. $x_1, ..., x_T$ are $M \times 1$ zero mean i.i.d. Gaussian random vectors with covariance $I_M$, where $T = 100$. $\epsilon_1, ..., \epsilon_T$ are $M \times 1$ zero mean i.i.d. random vectors with covariance $4I_M$. The tuning parameter criterion (using one realization from the random process) for $r = 2$ and various $h_u$ and $h_f$ is depicted on Fig. 2. Although the $CC$ function looks flat near $h_u, h_f = 0.1$ it does take a minimum. Fig. 3. shows the estimates for $U$ and $F$ for the $h_u, h_f$ that correspond to the minimum of the tuning parameter criterion. The estimates correspond to the minimum are zero at the correct places (between $v = 10$ and $v = 40$).

Fig. 1. The $F$ and $U$ loading matrices for the simulation.

Fig. 2. $C$ plotted for $r = 2$ and various $h_u$ and $h_f$ for the simulation.

6. REAL DATA

In this section we analyzed functional resonance magnetic imaging (fMRI) data coming from a visual-motor experiment. We looked at the data from one brain slice known to contain the supplementary motor area (SMA) and the primary motor cortex (PMC). A $7 \times 7$ region of brain voxels was extracted from the SMA. The corresponding signals are contained in a $100 \times 49$ matrix $X$. Similarly a $7 \times 7$ region was extracted from the PMC and contained in a $100 \times 49$ matrix $Y$ (see Fig 4. left). Fig. 4 right shows the stimulus signal for the experiment. When the stimulus signal was equal to 1 the subject performed right hand finger thumb opposition when it was 0 the subject rested. The tuning parameter criterion was computed for a grid of $h_u, h_f$ values and for few values of $r$. A plot of it for $r = 1$ and various $h_u, h_f$ is shown on Fig 5.

Fig. 3. The estimates $F$ and $U$ for the $h_f, h_u$ that correspond to the minimum of the $CC$ criterion.

Fig. 4. Left: The regions of interest. Right: The stimulus signal.

The loadings for the estimates $F$ and $U$ corresponding to the minimum of the $CC$ criterion are shown on Fig. 6. We can see that both $U$ and $F$ are sparse. Finally Fig. 7 displays the scores $F^T y_t$ and $U^T x_t$ for $t = 1, ..., T$. The correlation of the scores is 0.8272. Also note that the scores resemble the stimulus signal.

7. CONCLUSION

In this paper we developed a novel vector $l_1$ penalized method for automatic variable selection for reduced rank regression. The proposed cyclic descent algorithm involves optimization on two Stiefel manifolds. We demonstrated its ability to estimate sparse models both for a simulated data set and for a real fMRI data set.
Appendix: Derivation of the $D$-step.

Minimization of (3) w.r.t. $D$ is equivalent to minimizing

$$J_d(D) = \frac{1}{2} \sum_{j=1}^{r} \alpha_j d_j^2 - \sum_{j=1}^{r} \beta_j d_j$$

subject to $d_j \geq 0$, \hspace{1cm} (7)

where $d_j$ is the j-th diagonal element of $D$, $\alpha_j = u_j^T S_{xx} u_j$, $\beta_j = u_j^T S_{xx} f_{(j)}$ and $u_{(j)}$, $f_{(j)}$ are the j-th column vectors of $U$ and $F$, respectively. The derivative with respect to $d_j$ is $\frac{\partial J_d}{\partial d_j} = \alpha_j d_j - \beta_j$. The value that minimizes (7) is clearly

$$d_j = \begin{cases} \frac{\beta_j}{\alpha_j}, & \beta_j/\alpha_j > 0 \\ 0, & \text{else} \end{cases}$$

8. REFERENCES


