Sparse Multivariate Factor Regression
Milad Kharratzadeh, Mark Coates

Abstract—We consider the problem of multivariate regression in a setting where the relevant predictors could be shared among different responses. We propose an algorithm which decomposes the coefficient matrix into the product of a long matrix and a wide matrix, with a separate $\ell_1$ penalty on each. The first matrix linearly transforms the predictors to a set of latent factors, and the second one regresses the responses on these factors. Our algorithm simultaneously performs dimension reduction and coefficient estimation and automatically estimates the number of latent factors from the data. Our formulation results in a non-convex optimization problem, which despite its flexibility to impose effective low-dimensional structure, is difficult, or even impossible, to solve exactly in a reasonable time. We specify an optimization algorithm based on alternating minimization to solve this non-convex problem and provide theoretical results on its convergence and local optimality. Finally, we demonstrate the effectiveness of our algorithm via experiments on simulated and real data.

I. INTRODUCTION

Multivariate regression analysis, also known as multiple-output regression, is concerned with modelling the relationships among a set of real-valued output vectors, known as responses, and a set of real-valued input vectors, known as predictors or features. The multivariate responses are measured over the same set of predictors and are often correlated. Hence, the goal of multivariate regression is to exploit these dependencies to learn a predictive model of responses based on an observed set of input vectors paired with corresponding outputs. Multiple-output regression can also be seen as an instance of the problem of multi-task learning, where each task is defined as predicting individual responses based on the same set of predictors. The multivariate regression problem is encountered in numerous fields including finance [1], computational biology [2], geostatistics [3], chemometrics [4], and neuroscience [5].

Given $p$-dimensional predictors $x_i = (x_{i1}, \ldots, x_{ip})^T \in \mathbb{R}^p$ and $q$-dimensional responses $y_i = (y_{i1}, \ldots, y_{iq})^T \in \mathbb{R}^q$ for the $i$-th sample, we assume there is a linear relationship between the inputs and outputs as follows:

$$y_i = D^T x_i + \epsilon_i, \quad i = 1, \ldots, N,$$  \hspace{1cm} (1)

where $D_{p \times q}$ is the regression coefficient matrix and $\epsilon_i = (\epsilon_{i1}, \ldots, \epsilon_{iq})$ is the vector of errors for the $i$-th sample.

We can combine these $N$ equations into a single matrix formula:

$$Y = XD + E,$$  \hspace{1cm} (2)

where $X$ denotes the $n \times p$ matrix of predictors with $x_i^T$ as its $i$-th row, $Y$ denotes the $n \times q$ matrix of responses with $y_i^T$ as its $i$-th row, and $E$ denotes the $n \times q$ matrix of errors with $\epsilon_i^T$ as its $i$-th row. For $q = 1$, this multivariate linear regression model reduces to the well-known, univariate linear regression model.

We assume that the columns of $X$ and $Y$ are centred and hence the intercept terms are omitted. We also assume that the error vectors for $N$ samples are iid Gaussian random vectors with zero mean and covariance $\Sigma$, i.e. $\epsilon_i \sim \mathcal{N}(0, \Sigma), i = 1, \ldots, N$.

In the absence of additional structure, many standard procedures for solving (2), such as linear regression and principal component analysis, are not consistent unless $p/n \to 0$. Thus, in a high-dimensional setting where $p$ is comparable to or greater than $n$, we need to impose some low-dimensional structure on the coefficient matrix. For instance, element-wise sparsity can be imposed by constraining the $\ell_1$ norm of the coefficient matrix, $\|D\|_{1,1}$ [6], [7]. This regularization is equivalent to solving $q$ separate univariate lasso regressions for every response; thus we consider tasks separately. Another way to introduce sparsity is to consider the mixed $\ell_1/\ell_\gamma$ norms ($\gamma > 1$). In this approach (sometimes called group lasso), the mixed norms impose a block-sparse structure where each row is either all zero or mostly zeros. Particular examples, among many other works, include results using the $\ell_1/\ell_\infty$ norm [8], [9], and the $\ell_1/\ell_2$ norm [10], [11]. Also, there are the so-called “dirty” models which are superpositions of simpler low-dimensional structures such as element-wise and row-wise sparsity [12], [13] or sparsity and low rank [14], [15].

Another approach is to impose a constraint on the rank of the coefficient matrix. In this approach, instead of constraining the regression coefficients directly, we can apply penalty functions on the rank of $D$, its singular values and/or its singular vectors [16]–[20]. These algorithms belong to a broad family of dimension-reduction methods known as linear factor regression, where the responses are regressed on a set of factors achieved by a linear transformation of the predictors. The coefficient matrix is decomposed into two matrices: $D = A_{p \times m} B_{m \times q}$. Matrix $A$ transforms the predictors into $m$ latent factors, and matrix $B$ determines the factor loadings.
Our contributions

Here, we propose a novel algorithm which performs sparse multivariate factor regression (SMFR). We jointly estimate matrices \( A \) and \( B \) by minimizing the mean-squared error, \( \| Y - XAB \|_F^2 \), with \( \ell_1 \) penalties on both matrices. We provide a formulation to estimate the number of effective latent factors, \( m \). To the best of our knowledge, our work is the first to strive for low-dimensional structure by imposing sparsity on both factoring and loading matrices. This can result in a set of interpretable factors and loadings with high predictive power; however, these benefits come at the cost of a non-convex objective function. Most current approaches for multivariate regression solve a convex problem (either through direct formulation or by relaxation of a non-convex problem) to impose low-dimensional structures on the coefficient matrix. Although non-convex formulations, such as the one introduced here, can be employed to achieve very effective representations in the context of multivariate regression, there are few theoretical performance guarantees for optimization schemes solving such problems. We formulate our problem in Section \[II\]. In Section \[III\] we propose an optimization procedure based on alternating minimization and provide theoretical guarantees for its convergence and local optimality. We show that under mild conditions on the predictor matrix, every limit point of the minimization algorithm is a local minimum of the objective function. Through analysis of simulations on synthetic datasets in Section \[V\] and two real-world datasets in Section \[VI\] we show that compared to other multivariate regression algorithms, our proposed algorithm can provide a more effective representation of the data, resulting in a higher predictive power.

Related Methods

Many multivariate regression techniques impose a low-dimensional structure on the coefficient matrix. Element-wise sparsity, here noted as LASSO, is the most common approach where the cost function is defined as \( \| D \|_{1,1} \) \([6, 7]\). An extension of LASSO to the multivariate case is the row-wise sparsity with the \( \ell_1/\ell_2 \) norm as the cost function: \( \| D \|_{1,2} \) \([10, 11]\). Peng et al. (2010) proposed a method, called RemMap, which imposes both element-wise and row-wise sparsity and solves the following problem:

\[
\min_{D} \| Y - XD \|_F^2 + \lambda_1 \| D \|_{1,1} + \lambda_2 \| D \|_{1,2}.
\]

These algorithms impose some form of structured sparsity on the coefficient matrix. In another approach, \([18]\) extend the partial least squares (PLS) framework by imposing an additional sparsity constraint and proposed Sparse PLS (SPLS).

In a similar algorithm to ours, called Sparse Reduced Rank Regression (SRRR), \([17]\) proposed solving the following problem:

\[
\min_{A, B} \| Y - XAB \|_F^2 + \lambda \| A \|_{1,2}, \quad \text{such that } BB^T = I.
\]

Similar to our algorithm, the coefficient matrix is decomposed into two matrices. However, our problem formulation differs in two important ways: (i) sparsity is imposed on matrix \( B \); and (ii) the number of factors is determined directly, without the need for cross-validation. The former difference has substantial consequences; when decomposing the coefficient matrix into two matrices, the first matrix has the role of aggregating the input signals to form the latent factors and the second matrix performs a multivariate regression on these factors. Thus, and as originally motivated by LASSO, we would like to impose the sparsity constraint on the second matrix in order to enhance the interpretability and prediction performance.

II. Problem Setup

In this work, we introduce a novel low-dimensional structure where we decompose \( D \) into the product of two sparse matrices \( A_{p \times m} \) and \( B_{m \times q} \) where \( m < \min(p, q) \). This decomposition can be interpreted as first identifying a set of \( m \) factors which are derived by some linear transformation of the predictors (through matrix \( A \)) and then identifying the transformed regression coefficient matrix \( B \) to estimate the responses from these \( m \) factors. We provide a framework to find \( m \), the number of effective latent factors, as well as the transforming and regression matrices, \( A \) and \( B \). For a fixed \( m \), define:

\[
\hat{A}_m, \hat{B}_m = \arg\min_{A_{p \times m}, B_{m \times q}} f(A, B),
\]

where

\[
f(A, B) = \frac{1}{2} \|Y - XAB\|_F^2 + \lambda_1 \| A \|_{1,1} + \lambda_2 \| B \|_{1,1}.
\]

Then, we solve the following optimization problem:

\[
\hat{m} = \max(m) \text{ such that rank}(\hat{A}_m) = \text{rank}(\hat{B}_m) = m,
\]

and choose \( \hat{A}_{\hat{m}} \) and \( \hat{B}_{\hat{m}} \) as solutions. Thus, we find the maximum number of factors such that the solution of \( (3) \) has full row rank factor and loading matrices. In other words, we find the maximum \( m \) such that the best possible regularized reconstruction of responses, i.e., the solution of \( (3) \), results in a model where the factors (columns of \( \hat{A} \)) and their contributions to the responses (rows of \( \hat{B} \)) are linearly independent.

We always require \( m < \min(p, q) \); however, in practical settings where \( p \) and \( q \) are very large, we impose an upper bound on \( m \) to have a reasonable number of factors and avoid overfitting. In choosing the optimal value of \( m \), we differ from the common approach in the literature in two important ways. First, we focus on the ranks of individual matrices \( A \) and \( B \) and make sure they are full rank,
while most algorithms focus on the rank of the coefficient matrix, \( D = AB \). In general, we have \( \text{rank}(AB) \leq \min\{\text{rank}(A), \text{rank}(B)\} \). The linear independence of the factors and loadings, and consequently the ranks of \( A \) and \( B \), are important for determining the number of effective latent factors, and thus focusing on the rank of their product might be misleading. The second difference is related to the way we formulate our problem. Consider the following problem that solves the same optimization problem as in (3), but finds \( m \) to minimize the objective function (as opposed to our formulation where we find the largest \( m \) that results in full row rank solutions):

\[
\hat{A}, \hat{B}, \hat{m} = \underset{A_{p \times m}, B_{m \times q}, m}{\text{argmin}} \frac{1}{2} \| Y - XAB \|F^2 \\
+ \lambda_1 \|A\|_{1,1} + \lambda_2 \|B\|_{1,1}, \tag{6}
\]

such that \( \text{rank}(A) = \text{rank}(B) = m \). \( \tag{7} \)

Solving this problem (or similar problems) directly is difficult and the common approach in the literature is to solve it for a fixed \( m \) and use cross-validation to find the optimal \( \hat{m} \). \( \{3, 13, 17, 19, 20\} \). The two formulations in (3) and (6) do not result in the same solutions in general.

We always have \( \hat{m} \geq \hat{m} \) since for \( m < \hat{m} \), and for all \( A \in \mathbb{R}^{p \times m} \) and \( B \in \mathbb{R}^{m \times q} \), we have \( f(A, B) \geq f(\hat{A}, \hat{B}) \); otherwise we can insert \( m - \hat{m} \) columns and rows of zeros at the end of \( A \) and \( B \), respectively, and get matrices \( A'_{p \times \hat{m}} \) and \( B'_{\hat{m} \times q} \) for which we have \( f(A', B') < f(\hat{A}, \hat{B}) \) which is in contradiction with the optimality of \((\hat{A}, \hat{B})\).

If \( \hat{m} \neq \hat{m} \), we have two matrices \( A'_{p \times \hat{m}} \in \mathbb{R}^{p \times \hat{m}} \) and \( B'_{\hat{m} \times q} \in \mathbb{R}^{\hat{m} \times q} \) such that \( f(A', B') \leq f(\hat{A}, \hat{B}) \) and \( \text{rank}(B') < \hat{m} \); if there are not two such alternative matrices, then \( \hat{A} \) and \( \hat{B} \) minimize \( f \) for dimension \( \hat{m} \) (even ignoring the rank constraint) and this would therefore be a viable solution for the problem in (3). \( \{3, 5\} \). So, in the subspace \( \mathbb{R}^{p \times \hat{m}} \times \mathbb{R}^{\hat{m} \times q} \), the best possible regularized reconstruction of responses does not result in a set of factors with linearly independent contributions to the responses and \( \hat{m} \) is not a good estimate of the effective number of factors. With the formulation presented in (3), we find a more meaningful estimate of the number of latent factors compared to the common approach of identifying the number of latent factors via cross-validation.

### III. OPTIMIZATION TECHNIQUE AND THEORETICAL RESULTS

The optimization problem defined in (3) is not a convex problem and it is difficult, if not impossible, to solve exactly (i.e., to find the global optimum) in polynomial time. Therefore, we have to employ heuristic algorithms, which may or may not converge to a locally or globally optimal solution. In this section, we propose an algorithm based on alternate minimization, and show that it is locally well-posed: under mild conditions, it converges to a local minimum. We present our results through a series of theorems with detailed proofs in the Appendix.

For a fixed \( m \), the objective function in (3), \( f(A, B) \), is biconvex if either \( A \) or \( B \) is fixed. Let us define \( C = (A, B) \). To solve (3) for a fixed \( m \), we perform Algorithm 1, with an arbitrary, non-zero starting value \( A_0 \).

#### Algorithm 1 Solving problem (3) for fixed \( m \)

1: \( A \leftarrow A_0, i \leftarrow 0 \)
2: \textbf{while} value of \( f(A, B) \) not converged \textbf{do}
3: \hspace{1em} \( B_{i+1} \leftarrow \text{argmin}_B \frac{1}{2} \| Y - XAB \|_F^2 + \lambda_2 \| B \|_{1,1} \)
4: \hspace{1em} \( A_{i+1} \leftarrow \text{argmin}_A \frac{1}{2} \| Y - XAB \|_F^2 + \lambda_1 \| A \|_{1,1} \)
5: \hspace{1em} \( i \leftarrow i + 1 \)
6: \textbf{end while}
7: \( \hat{A}, \hat{B} \leftarrow \text{values of } A \text{ and } B \text{ at convergence} \)

**Definition 1.** \( \hat{C} = (\hat{A}^*, \hat{B}^*) \) is called a partial optimum of \( f \) if

\[
f(\hat{A}^*, \hat{B}^*) \leq f(A^*, B), \forall B \in \mathbb{R}^{m \times q} \tag{8}
\]

and

\[
f(\hat{A}^*, \hat{B}^*) \leq f(A, B^*), \forall A \in \mathbb{R}^{n \times m}. \tag{9}
\]

**Definition 2.** A point \( C^* \) is an accumulation point or a limit point of a sequence \( \{C_i\}_{i \in \mathbb{N}} \), if for any neighbourhood \( V \) of \( C^* \), there are infinitely many \( j \in \mathbb{N} \) such that \( C_j \in V \). Equivalently, \( C^* \) is the limit of a subsequence of \( \{C_i\}_{i \in \mathbb{N}} \).

**Main Results:** We show that the sequence \( f(C_i) \) monotonically converges. We also show that the sequence of solutions, \( \{C_i\}_{i \in \mathbb{N}} \), has at least one accumulation point and moreover, under a mild assumption on the predictor matrix, for any given starting point, all its accumulation points are partial optima and have the same function value. We prove that all the partial optima of \( f \) are indeed local optima (i.e., not saddle points). Finally, we show that if all the accumulation points have full rank matrices, the solutions, \( \{C_i\}_{i \in \mathbb{N}} \)

will converge in a practical sense.

**Proposition 1.** The sequence \( f(A_i, B_i) \) generated by Algorithm 1 converges monotonically.

The value of \( f \) is always positive and is reduced in each of the two main steps of Algorithm 1. Thus, it is guaranteed that the stopping criterion of Algorithm 1 will be reached. Next, we study the properties of \( C = (\hat{A}, \hat{B}) \) at convergence.

**Theorem 1.** The sequence \( \{C_i\}_{i \in \mathbb{N}} \) generated by Algorithm 1 has at least one accumulation point. Moreover, if the entries of \( X \in \mathbb{R}^{n \times p} \) are drawn from a continuous probability distribution on \( \mathbb{R}^{np} \), then for any starting point, all the associated accumulation points of the solutions...
produced by Algorithm 1 are partial optima and have the same function value.

Both steps of Algorithm 1 correspond to Lagrange dual problems where the constraints are in the form of \( \|B\|_1 \leq t_2 \) and \( \|A\|_1 \leq t_1 \), where \( t_1 \) and \( t_2 \) have a one-to-one correspondence to \( \lambda_1 \) and \( \lambda_2 \), respectively. This means that all the possible solutions produced by Algorithm 1 are contained in a bounded, closed (and hence compact) set. Thus, the sequence \( \{C_i\}_{i \in \mathbb{N}} \) has at least one limit point \( C^* \).

The condition on the entries of \( X \) is sufficient to achieve solution uniqueness for the LASSO problem on line 3 of Algorithm 1 [24]. Although Algorithm 1 converges to a specific value of \( f \), this value can be achieved by different values of \( C \). Thus, the sequence \( C_i \) can have many accumulation points. Provided that the solution of the first step of the alternating minimization is unique, Theorem 1 shows that any accumulation point is a partial optimum. Proposition 1 implies that for any given starting point, all the associated accumulation points have the same value.

**Theorem 2.** If the entries of \( X \in \mathbb{R}^{n \times p} \) are drawn from a continuous probability distribution on \( \mathbb{R}^{n \times p} \), then any partial optimum, and hence any accumulation point, of Algorithm 1, except the one at \( A = 0, B = 0 \), is a local minimum.

Partial optimality of \( (A^*, B^*) \) implies that \( f(A^*, B^*) \leq f(A, B^*) \) and \( f(A^*, B^*) \leq f(A^*, B) \) for any \( A \) and \( B \). We show that if the solution of the first minimization is unique (guaranteed by the condition on the elements of \( X \)), we get a stronger result that \( f(A^*, B^*) \leq f(A, B) \) for any \( A \) and \( B \) in the neighbourhood of \( (A^*, B^*) \) – any accumulation point of Algorithm 1 is a local minimum.

So far, we have discussed results which are based on the solution uniqueness of the first minimization. We can show that if \( B_{i+1} \) has full rank, then \( A_{i+1} \) can be identified uniquely, since \( \|Y - XAB_{i+1}\|_2^2 \) becomes strictly convex (see the Appendix for more details). With this, we can state the following theorem.

**Theorem 3.** If the entries of \( X \in \mathbb{R}^{n \times p} \) are drawn from a continuous probability distribution on \( \mathbb{R}^{n \times p} \) and if for any accumulation point \( C^* = (A^*, B^*) \), matrix \( B^* \) has full rank, in addition to the results of Theorem 2, we have:

\[
\lim_{i \to \infty} \|C_{i+1} - C_i\| = 0, \tag{10}
\]

and the accumulation points form a compact continuum (i.e., the set of accumulation points is connected and compact).

Although the condition in (17) does not guarantee the convergence of the sequence \( \{C_i\}_{i \in \mathbb{N}} \), it is close enough for all practical purposes.

Now, we propose Algorithm 2 to solve the optimization problem described in (5) to find the number of latent factors as well as the factor and loading matrices. Optimal values of \( \lambda_1 \) and \( \lambda_2 \) are found via 5-fold cross-validation.

**Algorithm 2 Sparse Multivariate Factor Regression (SMFR) via Alternating Minimization**

1. **Input:** Training Set \( X_{n \times p}, Y_{n \times q}, \lambda_1, \lambda_2 \)
2. **Output:** Solution of problem (3–5): \( \hat{A}, \hat{B}, \hat{m} \)
3. \( m \leftarrow r \triangleright r: \) upper bound on the number of factors
4. while true do
5. \( A \leftarrow A_0 \in \mathbb{R}^{p \times m}, i \leftarrow 0 \)
6. while value of \( f(A, B) \) not converged do
7. \( B_{i+1} \leftarrow \text{argmin}_{B} \|Y - XA, B\|_F^2 + \lambda_2 \|B\|_1,1 \)
8. \( A_{i+1} \leftarrow \text{argmin}_{A} \|Y - X\hat{A}B_{i+1}\|_F^2 + \lambda_1 \|A\|_1,1 \)
9. \( i \leftarrow i + 1 \)
10. end while
11. \( \hat{A}, \hat{B} \leftarrow \text{values of } A \text{ and } B \text{ at convergence} \)
12. if rank(\( \hat{A} \)) < \( m \) or rank(\( \hat{B} \)) < \( m \) then
13. \( m \leftarrow m - 1 \)
14. else
15. break
16. end if
17. end while

Assume that for a specific value of \( m \), the value of \( f \) converges in \( k \) steps, and the matrices \( A_k \) and \( B_k \) are full rank. Since we are at convergence, the value of \( f \) does not decrease further. Thus, \( f(A_k, B_k) = f(A_k, B_{k+1}) \). We know that minimizing \( f \) for a fixed \( A_k \) has a unique solution (Lemma 3 in the Appendix) and so, \( B_k = B_{k+1} \). Also, we know that for a fixed full rank \( B_k \), minimizing \( f \) with respect to \( A \) has a unique solution (Lemma 7 in the Appendix) and so, \( A_k = A_{k+1} \). Therefore, Algorithm 2 will converge both in terms of the \( f \) value and the factor and loading matrices.

**IV. Fully Sparse PCA**

In [22], Zou, Hastie, and Tibshirani propose a sparse PCA, arguing that in regular PCA “each principal component is a linear combination of all the original variables, thus it is difficult to interpret the results”. Assume that we have a data matrix \( X_{n \times p} \) with the following SV decomposition: \( X = UDV^T \). The principal components are defined as \( Z = UD \) with the corresponding columns of \( V \) as the loadings. In [22], Zou et al. show that solving the following optimization problem leads to exact PCA:

\[
(\hat{A}, \hat{B}) = \arg\min_{A,B} \|X - X\hat{A}B\|_F^2 + \lambda \sum_i \|A_i\|_2 \text{ s.t. } BB^T = I,
\]

where \( A_i \) denotes the \( i \)‘th column of \( A \). Thus, we have \( \hat{A}_i \propto \hat{V}_i \), and \( X\hat{A}_i \) corresponds to the \( i \)‘th principal component. Sparse PCA (SPCA) is introduced by adding
an $\ell_1$ penalty on matrix $A$ to the objective function. Zou et al. propose an alternating minimization scheme to solve this problem.

The ordinary principal components are uncorrelated and their loadings are orthogonal. SPCA imposes sparsity on the construction of the principal components. Here sparsity means that each component is a combination of only a few of the variables. By enforcing sparsity, the principal components become correlated and the loadings are no longer orthogonal. On the other hand, SPCA assumes, like regular PCA, that the contributions of these components are orthonormal ($BB^T = I$). In our algorithm, if we replace $Y$ with $X$, i.e., regressing $X$ on itself, we get a similar algorithm. However, our work differs in two ways. First, we also impose sparsity on the contributions of principal components. This comes at the expense of higher computational costs, but results in more interpretable results. Also, the contributions will not be orthonormal anymore. However, by the full-rank constraint we impose on the two matrices, we make sure that the principal components and their contributions are linearly independent. Moreover, we find the sufficient number of principal components to explain the data from the data itself.

V. SIMULATION STUDY

In this section, we use synthetic data to compare the performance of our algorithm with the five related multivariate regression methods reviewed in the introduction.

A. Simulation Setup

We generate the synthetic data in accordance with the model described in (2), $Y = XD + E$, where $D = AB$. First, we generate an $n \times p$ predictor matrix, $X$, with rows independently drawn from $N(0, \Sigma_X)$, where the $(i,j)$-th element of $\Sigma_X$ is defined as $\sigma_{i,j}^X = 0.7|j-i|$. This is a common model for predictors in the literature [13], [19]. [23]. The rows of the $n \times q$ error matrix are sampled from $N(0, \Sigma_N)$, where the $(i,j)$-th element of $\Sigma_N$ is defined as $\sigma_{i,j}^N = \sigma_n^2 \cdot 0.4|j-i|$. The value of $\sigma_n^2$ is varied to attain different levels of signal to noise ratio (SNR). Each row of the $p \times m$ matrix $A$ is chosen by first randomly selecting $m_0$ of its elements and sampling them from $N(0, 1)$ and then setting the rest of its elements to zero. Finally, we generate the $m \times q$ matrix $B$ by the element-wise product of $B = U \circ W$, where the elements of $U$ are drawn independently from $N(0, 1)$ and elements of $W$ are drawn from Bernoulli distribution with success probability $s$.

We evaluate the performance of a given algorithm with three different metrics. We evaluate the predictive performance over a test set $(X_{test}, Y_{test})$, separate from the training set, in terms of the mean-squared error:

$$\text{MSE} = \frac{\|X_{test} \hat{D} - Y_{test}\|^2_F}{nq}, \quad (11)$$

where $\hat{D}$ is the estimated coefficient matrix. In our case, we have $\hat{D} = \hat{A} \hat{B}$. We also compare different algorithms based on their signed sensitivity and specificity of the support recognition:

$$\text{Signed Sensitivity} = \frac{\sum_{i,j} 1[d_{i,j} \cdot \hat{d}_{i,j} > 0]}{\sum_{i,j} 1[d_{i,j} \neq 0]},$$

$$\text{Specificity} = \frac{\sum_{i,j} 1[d_{i,j} = 0] \cdot 1[\hat{d}_{i,j} = 0]}{\sum_{i,j} 1[d_{i,j} = 0]},$$

where 1 represents the indicator function.

We compare the performance of our algorithm, SMFR, with the other algorithms reviewed in Section I. We consider three different regimes: (i) high-dimensional problems with few instances ($50$) compared to the number of predictors or responses ($50$, $100$, or $150$); (ii) problems with increased number of instances ($p, q < n$); and (iii) problems where the structural assumption of our technique is violated. In the first regime, which is of most interest to us due to high-dimensionality, we explore different parameter settings. The values of $\sigma_n$ and $s$ affect the SNR—lower values of $\sigma_n$ and higher values of $s$ correspond to higher values of SNR (e.g., $\sigma_n = 5, s = 0.1$ corresponds to a very low SNR). In regime (ii), we violate the assumption about the structure of the coefficient matrix, i.e., $D = AB$, in two ways. In the first case, $D$ has an element-wise sparsity with $20\%$ non-zero elements, and in the second, it has row-wise sparsity where $60\%$ of the rows are all zeros and the rest have $30\%$ non-zero elements. We consider these cases to compare our algorithm with others in an unfavourable setting.

B. Results

a) Predictive performance: The means and standard deviations of different algorithms are presented in Table I. We use five-fold cross-validation to find the tuning parameters of all algorithms. For the starting point of our algorithm, we sample the elements of $A_0$ from $N(0, 1)$. We set $r$, the maximum number of factors, to $20$. For the first two simulation regimes, our algorithm outperforms the other algorithms and results in lower MSE means and standard deviation. The improvements are more significant in the high-dimensional settings with high SNR. However, in settings with low SNR ($\sigma_n = 5, s = 0.1$) or high number of instances ($n = 500$), we still observe lower errors for SMFR. On average, our algorithm reduces the test error by $8.3\%$ compared with LASSO, $16.7\%$ compared with $\ell_1/\ell_2$, $7.4\%$ compared with SRRR, $10.2\%$ compared with RemMap, and $14.8\%$ compared with SPLS.

In the last two simulations, where the assumed factor structure is abandoned, our algorithm has no advantage over simpler methods with no factor structure (such as LASSO) and gives a higher error.
### TABLE I

Comparison of six algorithms for different setups. We report mean and standard deviations of the MSE over the test sets, based on 20 simulation runs.

| Parameters | SMFR | LASSO | MSE over test set | SRRR | RemMap | SPLS |
|------------|------|-------|-------------------|------|--------|------|
| n | p | q | m | m₀ | σₙ | s | MSE over test set | ℓ₁/ℓ₂ |
| 50 | 150 | 50 | 10 | 1 | 3 | 0.2 | 0.074 | 0.083 | 0.090 | 0.084 | 0.083 | 0.091 |
| | | | | | | | (0.003) | (0.005) | (0.005) | (0.005) | (0.005) | (0.007) |
| 10 | 1 | 3 | | | | | 0.083 | 0.104 | 0.105 | 0.099 | 0.104 | 0.110 |
| | | | | | | | (0.006) | (0.008) | (0.007) | (0.007) | (0.008) | (0.008) |
| 10 | 1 | 5 | | | | | 0.110 | 0.118 | 0.133 | 0.117 | 0.123 | 0.122 |
| | | | | | | | (0.004) | (0.005) | (0.004) | (0.005) | (0.004) | (0.007) |
| 15 | 2 | 3 | | | | | 0.096 | 0.108 | 0.112 | 0.109 | 0.107 | 0.114 |
| | | | | | | | (0.005) | (0.006) | (0.007) | (0.008) | (0.006) | (0.008) |
| 50 | 100 | 100 | 10 | 1 | 5 | 0.1 | 0.069 | 0.070 | 0.092 | 0.071 | 0.075 | 0.073 |
| | | | | | | | (0.002) | (0.002) | (0.002) | (0.002) | (0.002) | (0.002) |
| 500 | 150 | 50 | 10 | 1 | 3 | 0.2 | 0.0173 | 0.0180 | 0.0198 | 0.0176 | 0.0184 | 0.0216 |
| | | | | | | | (0.0001) | (0.0001) | (0.0001) | (0.0001) | (0.0001) | (0.0007) |
| 500 | 100 | 100 | 10 | 1 | 5 | 0.3 | 0.0202 | 0.0209 | 0.0222 | 0.0204 | 0.0214 | 0.0222 |
| | | | | | | | (0.0001) | (0.0002) | (0.0001) | (0.0001) | (0.0001) | (0.0003) |
| 50 | 100 | 100 | | | | | 0.084 | 0.078 | 0.096 | 0.080 | 0.085 | 0.081 |
| | | | | | | | (0.001) | (0.001) | (0.002) | (0.001) | (0.001) | (0.001) |
| 50 | 150 | 50 | | | | | 0.083 | 0.076 | 0.080 | 0.079 | 0.075 | 0.083 |
| | | | | | | | (0.004) | (0.003) | (0.003) | (0.003) | (0.003) | (0.003) |

b) **Variable selection:** In Figure 1, we compare the average signed sensitivity and specificity of different algorithms (based on 20 simulation runs) as the number of instances increase (other parameters kept fixed). We observe that our algorithm has higher sensitivity and specificity. This effect for specificity is reduced as the number of instances increases. This shows that our algorithm is more advantageous in high-dimensional settings where the number of instances is comparable to or less than the number of predictors/responses. Although we only show the plots for a specific parameter setting, the results are similar for other parameters.

c) **Number of latent factors:** In Table II, we compare the number of estimated factors for the three algorithms that perform dimensionality reduction. For SRRR and SPLS, we find the number of factors by 5-fold cross-validation. The true number of factors is 10 in all parameter settings. We observe that except for the first set of parameters, our algorithm provides as good or better estimates of the number of factors.

d) **Computation time:** We also compare the computation time of different algorithms for the parameter settings.

Fig. 1. Sensitivity and specificity comparison of different algorithms as the number of instances increases. (p = 150, q = 50, m = 10, m₀ = 1, σₙ = 3, s = 0.3)
corresponding to the first row of Table I. We report the median computation time (on a PC with 16GB RAM and quad-core CPU at 3.4GHz) of each algorithm (excluding the cross-validation part) over 20 runs; SMFR: 4.0s, SRRR: 5.6s, RemMap: 0.3s, SPLS: 0.3s, LASSO: 0.02s, and $\ell_1/\ell_2$: 0.33s. As expected, SMFR and SRRR have higher computation times, since they provide extra information (factoring of predictors)—which can result in better predictive performance and provide more insight about the structure of data—compared to the more basic algorithms.

| $n$  | $p$  | $q$  | $m$  | $m_0$ | $s$  | SMFR | SRRR | SPLS | LASSO |
|------|------|------|------|-------|------|------|------|------|-------|
| 50   | 150  | 50   | 10   | 1     | 3    | 0.3  | 14   | 8    | 15    |
| 50   | 150  | 50   | 10   | 1     | 5    | 0.3  | 11   | 8    | 9     |
| 50   | 150  | 50   | 10   | 1     | 5    | 0.2  | 10   | 7    | 7     |
| 50   | 100  | 100  | 10   | 1     | 5    | 0.1  | 7    | 6    | 4     |
| 500  | 150  | 50   | 10   | 1     | 3    | 0.2  | 10   | 10   | 15    |
| 500  | 100  | 100  | 10   | 1     | 5    | 0.3  | 10   | 10   | 15    |

**TABLE II**

Median number of estimated factors (20 runs).

VI. APPLICATION TO REAL DATA

We apply the proposed algorithm to real-world datasets and show that it exhibits better or similar predictive performance compared to state-of-the-art algorithms. We show that the factoring identified by our algorithm provides valuable insight into the underlying structure of the datasets.

A. Montreal’s bicycle sharing system (BIXI)

The first dataset we consider provides information about Montreal’s bicycle sharing system called BIXI. The data contains the number of available bikes in each of the 400 installed stations for every minute. We use the data collected for the first four weeks of June 2012. From this dataset we form the set of predictors and responses as follows. We allocate two features to each station corresponding to the number of arrivals and departures of bikes to or from that station for every hour. The learning task is to predict the number of arrivals and departures for all the stations from the number of arrivals and departures in the last hour (i.e., a vector autoregressive model). The choice of this model is a compromise between accuracy and complexity.

We perform the prediction task on each of the four weeks. For each week, we take the data for the first 5 days (120 data points; Friday to Tuesday) as the training set (with the fifth day data as the validation set), and the last two days as the test set (48 data points; Wednesday and Thursday). We compare the algorithms performing dimensionality reduction in terms of their predictive performance on the test sets and the number of chosen factors in Table III. We also include LASSO and $\ell_1/\ell_2$ as baseline algorithms. To avoid showing very small numbers, we present the value of MSE, defined in (11), times $nq$.

In terms of the prediction performance, we observe that our algorithm outperforms the others in all 4 weeks. We can also compare the algorithms in terms of the number of chosen factors. SMFR always chooses more factors compared to SRRR resulting in a better representation of data according to MSE. So, SRRR seems to underestimate the number of factors. SPLS chooses almost a constant number of factors, showing that it is not adaptive to the changes in the data over the four weeks.

### TABLE III

Total squared error (MSE $\times nq$) and the number of factors for BIXI dataset.

| Week | SMFR | SRRR | SPLS | LASSO | $\ell_1/\ell_2$ |
|------|------|------|------|-------|----------------|
| 1    | 560.1| 570.0| 1661 | 580.4 | 591.0 |
|      | factors: 7 | 3 | 7 | — | — |
| 2    | 589.1| 602.2| 1888 | 610.9 | 623.7 |
|      | factors: 9 | 3 | 6 | — | — |
| 3    | 639.7| 641.9| 2159 | 643.4 | 657.8 |
|      | factors: 14 | 7 | 5 | — | — |
| 4    | 556.5| 594.6| 1621 | 594.9 | 588.0 |
|      | factors: 4 | 3 | 6 | — | — |

To gain more insight into the quality of predictions, we randomly choose a feature in week 4 and plot the predictions made by different algorithms over the test set in Figure 2; the results for other features are similar. The $y$-axis represents the cumulative number of bikes. We observe that SMFR provides a better fit to the data.

To further investigate the variable selection of our algorithm, we run it on the whole data and examine the resulting factors. Three of these are shown in Figure 3. In these figures, all the bike stations are shown with green plus signs. For each factor, we show its constituent features; red crosses and blue circles correspond respectively to the departure and arrival features of each station that are present in that factor. Examining these factors provide useful insight into the data. For instance, the factor in Figure 3(a) shows that the departures from populated residential areas (The Plateau, Mile End, Outremont) and arrivals at downtown (Ville Marie) are combined together to form a factor. This agrees with the intuition that many people are taking bikes to go from their homes to downtown where universities and businesses are located. The factor in Figure 3(b) shows another strong effect which corresponds to the flow from the peripheries of downtown to more central locations (Place des Arts, Old Port). Many hotels and several universities are situated at the edge of downtown; numerous restaurants, cafés and tourist sites are located in the centre, and several festivals occurred there during June. The third factor represents the flow within this central part.
Fig. 2. Comparing fit of different algorithms for three randomly-chosen features. The proposed algorithm performs better than others for most stations.

Fig. 3. Three of the factors identified in the BIXI dataset by our algorithm. Green plus signs show the stations, red crosses show the departure features and blue circles show the arrival features of a station chosen in the factor.

B. S&P 500 stocks

We consider 294 companies from the S&P 500 [24] and collect their daily returns (percentage change in value from one day to the next) between March 1992 and December 2013 for a total of 5500 days. These returns are volatility adjusted using a GARCH model [25] and market adjusted by subtracting the market’s average return for each day. We use the global industry classification standard [26] which categorizes all major public companies into 10 sectors. Since there are very few companies in the Telecom sector, we ignore this sector altogether. We divide the companies into two equal groups such that the number of companies from a specific sector is the same in each of the two groups. Our learning task is to predict the daily returns of the second group of companies from the first group. Although this is not a prediction task that would be of most interest in practice (where we want to predict future returns), it is a good test to examine the ability of an algorithm to extract the underlying factors and existing structure in the data.

We divide the 5500 day period into 10 intervals of 550 days. For each interval, we choose the first 400 days as the training set (with the last 100 days as validation set) and the last 150 days as the test set. The average and standard deviation of MSE for SMFR is 129.3 (16.2), for SRRR is 129.2 (15.8), and for LASSO is 129.4 (17.1). There is minimal difference in the predictive performance of these algorithms on this dataset; however, we gain significant insight into the nature of data by looking at the factors created by our algorithm. Figure 4 compares the resulting factors of SRRR and SMFR run over a period of 3000 days (this length is chosen to have clearer factors). We place companies from the same sectors next to each other in predictor and response matrices, and separate them with green lines. The sectors, from top to bottom, are Energy, Materials, Industrials, Consumer Discretionary, Consumer Staples, Health Care, Financials, IT, and Utilities. The proposed algorithm, SMFR, captures the sector factors to a good extent, whereas the structure of factors identified by SRRR is less clear.

Fig. 4. Factor matrices for SMFR and SRRR.
C. Sparse PCA

We compare our proposed fully sparse PCA with the well-known SPCA of Zou et al. introduced in [22]. We use our BIXI dataset again. Remember that there are 800 features in this dataset. We use the first 200 data points of the dataset to simulate a high-dimensional setting. Thus, our data matrix, \( X \), is \( 200 \times 800 \). We compute the first 6 principal components and compare them using two metrics. First, we compute the adjusted explained variance; since the sparse principal components are not uncorrelated as in regular PCA, computing their explained variance separately is not correct. Before computing the explained variance of the \( k \)'th principal component, regression projection is used to remove its linear dependence to components 1 to \( k - 1 \). See [22] for more details on how to compute the adjusted explained variance.

We also compare the loading sparsity. To have sparse components, we set the regularization parameters such that each component receives contributions from at most 10\% of the features. As a benchmark, we also consider the simple thresholding where the values of the regular principal components with absolute value smaller than a threshold are set to zero (here, we keep the top 80).

The results are summarized in Table IV. Compared to SPCA our algorithm explains more variance in the data (a total of 63.9\% over the first 6 components compared to 18.6\%) with sparser components. Also, we achieve a higher adjusted total variance compared to the simple thresholding (63.9\% compared to 28.3\%).

|   | Adjusted Var (%) | loading sparsity (\( \| \cdot \|_{1,1} \)) |
|---|----------------|----------------------------------|
| PC | SMFR | SPCA | Thresholding | SMFR | SPCA |
| 1 | 14.5 | 8.7 | 15.2 | 78 | 80 |
| 2 | 12.4 | 2.9 | 4.2 | 35 | 79 |
| 3 | 12.3 | 2.0 | 2.7 | 66 | 80 |
| 4 | 11.5 | 1.9 | 2.3 | 43 | 78 |
| 5 | 8.0 | 1.7 | 2.1 | 80 | 77 |
| 6 | 5.2 | 1.4 | 1.8 | 71 | 79 |

TABLE IV
ADJUSTED EXPLAINED VARIANCE AND LOADING SPARSITY.

VII. Conclusion

We introduced a new sparse multivariate regression algorithm which imposes a low-dimensional structure on the coefficient matrix by first decomposing it into the product of a long factor matrix and a wide loading matrix, and then imposing \( \ell_1 \) penalties on both these matrices. We also provided a formulation to infer the number of latent factors in a more effective way than current techniques. Although the problem formulation leads to a non-convex optimization problem, we showed that the proposed alternating minimization scheme converges and is locally well-posed. Through experiments on simulated and real datasets, we demonstrated that the proposed algorithm is able to exploit the existing structure in the data to improve predictive performance and model selection.

APPENDIX

A. Proof of Proposition 1

Proposition 1. The sequence \( f(A_i, B_i) \) generated by Algorithm 1 converges monotonically.

Proof: For any given \( A \) and \( B \), we have \( f(A, B) \geq 0 \). In both minimization steps of Algorithm 1 (i.e., problems (13) and (14)), the value of function \( f \) is being decreased. Since \( f \) is bounded from below, the sequence of \( f(A_i, B_i) \) converges to a limit value \( f^* \in \mathbb{R} \).

B. Proof of Theorem 1

Some of the proofs in this subsection exploit the biconvexity of the problem we are addressing and are based on the proofs of similar results in [27]. To prove Theorem 1, we need to introduce the algorithmic map of Algorithm 1 and show that it is closed.

Definition 3. \( A \) is called the algorithmic map of Algorithm 1, if for \( C_1 = (A_1, B_1) \) and \( C_2 = (A_2, B_2) \) we have:

\[
C_2 \in A(C_1) \quad \text{iff} \quad f(A_1, B_2) \leq f(A_1, B), \forall B \in \mathbb{R}^{m \times q} \quad \text{and} \quad f(A_2, B_2) \leq f(A_2, B), \forall A \in \mathbb{R}^{n \times m}.
\]

In other words, \( C_2 \in A(C_1) \) iff we go from \( C_1 \) to \( C_2 \) in one iteration of Algorithm 1.

Lemma 1. The algorithmic map \( A \) is closed, i.e., we have:

\[
\begin{align*}
C_i = (A_i, B_i) \quad \text{and} \quad \lim_{i \to \infty} C_i &= C^* \quad \Rightarrow \quad C' \in A(C^*) \\
C'_i &\in A(C_i) \quad \Rightarrow \quad f(A_i, B'_i) \leq f(A_i, B), \quad \forall B \in \mathbb{R}^{m \times q} \\
&\text{and} \quad f(A'_i, B'_i) \leq f(A, B'_i), \quad \forall A \in \mathbb{R}^{n \times m}.
\end{align*}
\]

Proof:

Since \( f \) is continuous, we have:

\[
\begin{align*}
& f(A^*, B') = \lim_{i \to \infty} f(A_i, B'_i) \leq \lim_{i \to \infty} f(A_i, B) \\
& \quad = f(A^*, B) \quad \forall B \in \mathbb{R}^{m \times q} \\
& f(A', B') = \lim_{i \to \infty} f(A'_i, B'_i) \leq \lim_{i \to \infty} f(A_i, B'_i) \\
& \quad = f(A, B') \quad \forall A \in \mathbb{R}^{n \times m}.
\end{align*}
\]

Thus, \( C' \in A(C^*) \), and \( A \) is closed.
For a fixed \( m \), Algorithm 1 iteratively solves the following two sub-problems until convergence:

\[
\hat{B} = \arg\min_B \frac{1}{2} \| Y - XAB \|_F^2 + \lambda_2 \| B \|_1 \tag{13}
\]

\[
\hat{A} = \arg\min_A \frac{1}{2} \| Y - XAB \|_F^2 + \lambda_1 \| A \|_1 \tag{14}
\]

For problem (13), decomposing \( B \) into its columns, we can rewrite the minimization as follows:

\[
\hat{B} = \arg\min_B \sum_{j=1}^q \left( \frac{1}{2} \| Y(j) - XAB(j) \|_2^2 + \lambda_2 \| B(j) \|_1 \right),
\]

where for any matrix, the superscript of \( j \) denotes its \( j \)th column. Therefore, problem (13) is equivalent to \( q \) separate Lasso problems with the same \( X \) and \( \lambda_2 \) in (13) and (14) are in the penalized form. For any given \( Y \) and \( \lambda_2 \), the Lasso solution is unique with probability one.

In problem (13), if \( X \) is drawn from a continuous probability distribution on \( \mathbb{R}^{n \times p} \), then \( D = XA \) is drawn from a continuous probability distribution on \( \mathbb{R}^{nm} \), since multiplying by a fixed matrix \( A \) is equivalent to a linear transformation. Noting that problem (13) is equivalent to \( q \) separate Lasso problems with the same \( X \) and \( Y \), we can conclude:

**Lemma 2** ([21], Lemma 4). Assume that we have the following Lasso problem:

\[
\min_b \| y - Db \|_2^2 + \lambda \| b \|_1.
\]

Then, if the entries of \( D \in \mathbb{R}^{n \times m} \) are drawn from a continuous probability distribution on \( \mathbb{R}^{nm} \), then for any \( y \) and \( \lambda \), the Lasso solution is unique with probability one.

In problem (13), if \( X \) is drawn from a continuous probability distribution on \( \mathbb{R}^{np} \), then \( D = XA \) is drawn from a continuous probability distribution on \( \mathbb{R}^{nm} \), since multiplying by a fixed matrix \( A \) is equivalent to a linear transformation. Noting that problem (13) is equivalent to \( q \) separate Lasso problems with the same \( X \) and \( Y \), we can conclude:

**Lemma 3.** If the entries of \( X \in \mathbb{R}^{n \times p} \) are drawn from a continuous probability distribution on \( \mathbb{R}^{np} \), then for any \( Y \) and \( \lambda_2 \), the solution of problem (13) is unique with probability one.

**Theorem 1.** The sequence \( C_i \) generated by Algorithm 1 has at least one accumulation point. Moreover, if the entries of \( X \in \mathbb{R}^{n \times p} \) are drawn from a continuous probability distribution on \( \mathbb{R}^{np} \), then for any starting point, all the associated accumulation points of the solutions produced by Algorithm 1 are partial optima and have the same function value.

**Proof:** The two minimization problems described in (13) and (14) are in the penalized form. For any given \( \lambda_1 \) and \( \lambda_2 \), there exist \( t_1 \) and \( t_2 \) such that the following constrained problems are equivalent to those formed in (13) and (14):

\[
B_{i+1} = \arg\min_B \frac{1}{2} \| Y - XAB \|_F^2 \quad \text{s.t.} \quad \| B \|_{1,1} \leq t_1
\]

\[
A_{i+1} = \arg\min_A \frac{1}{2} \| Y - XAB \|_F^2 \quad \text{s.t.} \quad \| A \|_{1,1} \leq t_1
\]

Define \( g(A, B) = \frac{1}{2} \| Y - XAB \|_F^2 \). \( S_A = \{ A : \| A \|_{1,1} \leq t_1 \} \), and \( S_B = \{ B : \| B \|_{1,1} \leq t_2 \} \). Thus the two steps of the algorithm can be written as follows:

\[
B_{i+1} = \arg\min_B g(A_i, B) \quad \text{such that} \quad B \in S_B
\]

\[
A_{i+1} = \arg\min_B g(A, B_{i+1}) \quad \text{such that} \quad A \in S_A
\]

Both \( S_A \) and \( S_B \) are bounded, closed sets. Thus all the possible solutions produced by Algorithm 1 are contained in a bounded, closed (and hence compact) set. Consequently, the sequence \( \{ C_i \}_{i \in \mathbb{N}} \) has at least one limit point \( C^* \).

Since \( C_{i+1} \in A(C_i) \), we have \( f(C_{i+1}) \leq f(C_i) \). We claim that if \( C_i \) is not a partial optimum, then \( f(C_{i+1}) < f(C_i) \). Otherwise we will have \( f(C_{i+1}) = f(C_i) \) and given that the solution of (13) is unique according to Lemma 3 (i.e., \( B_{i+1} = B_i \)), we have:

\[
C_{i+1} \in A(C_i) \quad \text{and} \quad B_{i+1} = B_i \quad \Rightarrow \quad f(A_i, B_i) \leq f(A_i, B) \quad \forall B \in \mathbb{R}^{m \times q}
\]

and

\[
f(A_i, B_i) \leq f(A_i, B) \quad \forall A \in \mathbb{R}^{n \times m}
\]

which means that \( C_i \) is a partial optimum—a contradiction to our initial assumption. Thus we cannot have \( f(C_{i+1}) = f(C_i) \), and consequently, \( f(C_{i+1}) < f(C_i) \).

We know that the sequence \( \{ C_i \}_{i \in \mathbb{N}} \) has at least one accumulation point, say \( C^* \). Thus we have a convergent subsequence \( \{ C_i \}_{i \in K} \) with \( K \subset \mathbb{N} \) that converges to \( C^* \). Similarly, \( \{ C_{i+1} \}_{i \in K} \) has an accumulation point, say \( C^+ \), to which a subsequence \( \{ C_{i+1} \}_{i \in L} \) with \( L \subset K \) converges. From Lemma 1 we get \( C^+ \in A(C^*) \), and using Proposition 1 we conclude \( f(C^*) = f(C^+) \). However, if \( C^* \) is a partial optimum, we showed in the last paragraph that \( f(C^+) < f(C^*) \), which is a contradiction. Thus \( C^* \) must be a partial optimum. If there are any other accumulation points, their function value must be the same as \( f(C^*) \), according to Proposition 1.

**C. Proof of Theorem 2**

So far, we have shown that our proposed alternative minimization scheme converges to a partial optimum, say \( (A^*, B^*) \). Any partial optimum is a stationary point (with zero gradient) which might or might not be a local minimum. Based on the new notation introduced above, we can redefine the partial optimality as follows:

**Definition 4.** \( C^* = (A^*, B^*) \) is called a partial optimum of \( f \) if

\[
g(A^*, B^*) = \frac{1}{2} \| Y - XAB \|_F^2, \quad S_A = \{ A : \| A \|_{1,1} \leq t_1 \}, \quad \text{and} \quad S_B = \{ B : \| B \|_{1,1} \leq t_2 \}
\]

where \( g(A, B) = \frac{1}{2} \| Y - XAB \|_F^2 \).
Lemma 4. Assume that \((A^*, B^*)\) is a partial optimum. Then, for any \(A \in S_A\), we have:

\[
0 \leq \text{trace}\{B^T \delta_A^* X^T XA B^* - 2(Y - XA B^*)^T X \delta_A B^*\},
\]

where \(\delta_A = A - A^*\). Moreover, the equality holds iff \(g(A, B^*) = g(A^*, B^*)\).

**Proof:** By the definition of partial optimality, we can write:

\[
\|Y - XA B^*\|_F^2 \leq \|Y - XA B\|_F^2 \quad \forall A \in S_A
\]

\[
\Rightarrow \text{trace}\{(Y - XA B^*)^T (Y - XA B^*)\}
\]

\[
\leq \text{trace}\{(Y - XA B)^T (Y - XA B)\}
\]

\[
= \text{trace}\{(Y - X(A^* + \delta_A) B^*)^T (Y - X(A^* + \delta_A) B^*)\}
\]

\[
\forall \delta_A \quad \text{such that} \quad (A^* + \delta_A) \in S_A
\]

Simple algebra and cancelling equal terms from both sides of the inequality yields:

\[
0 \leq \text{trace}\{B^T \delta^*_A X^T XA B^* - 2(Y - XA B^*)^T X \delta_A B^*\}
\]

\[
\forall \delta_A \quad \text{such that} \quad (A^* + \delta_A) \in S_A
\]

The equality holds iff \(g(A, B^*) = g(A^*, B^*)\). \(\blacksquare\)

Lemma 5. Assume that \((A^*, B^*)\) is a partial optimum. Then, for any \(B \in S_B\), we have:

\[
0 \leq \text{trace}\{\delta^*_B A^* T X^T XA^* T \delta_B - 2(Y - XA B^*)^T X A^* T \delta_B\}
\]

where \(\delta_B = B - B^*\). The equality holds iff \(B^* = B\).

**Proof:** The inequality can be proved in exactly the same way as Lemma 4. The equality condition is \(g(A^*, B) = g(A^*, B^*)\). However, from Lemma 3, we know that this equality holds iff \(B = B^*\). \(\blacksquare\)

**Theorem 2.** If the entries of \(X \in \mathbb{R}^{n \times p}\) are drawn from a continuous probability distribution on \(\mathbb{R}^{np}\), then any partial optimum, and hence any accumulation point, of Algorithm 1, except the one at \(A = 0, B = 0\), is a local minimum.

**Proof:** Assume that \(A \in S_A\) and \(B \in S_B\). Also, assume that \(\|\delta_A\|_{\infty, \infty} = \delta_1\) and \(\|\delta_B\|_{\infty, \infty} = \delta_2\), where \(\delta_A = A - A^*\), \(\delta_B = B - B^*\), and for any matrix \(M\), \(\|M\|_{\infty, \infty} = \max_{i,j} |M_{i,j}|\). We prove that \((A^*, B^*)\) is a local minimum by showing that

\[
g(A^*, B^*) \leq g(A, B) \quad \text{as} \quad \delta \to 0^+, \quad (16)
\]

where \(\delta = \max\{\delta_1, \delta_2\}\).

We have:

\[
g(A, B) = \text{trace}\left\{(Y - X(A^* + \delta_A)(B^* + \delta_B))\right\}^T
\]

\[
-Y - X(A^* + \delta_A)(B^* + \delta_B))\]

\[
+ \text{trace}\left\{B^T \delta_A^* X^T XA B^* - 2(Y - XA B^*)^T X \delta_A B^*\right\}
\]

\[
+ \text{trace}\left\{\delta^*_B A^* T X^T XA^* T \delta_B - 2(Y - XA B^*)^T X A^* T \delta_B\right\}
\]

\[
\forall \delta^*_B \quad \text{such that} \quad (A^* + \delta^*_B) \in S_B
\]

By definition, we have \(I = 2g(A^*, B^*)\). According to Lemmas 4 and 5: \(II \geq 0\) and \(III \geq 0\). If \(II + III = 0\), then \(II = 0\) and \(III = 0\). From Lemma 4, \(II = 0\) gives \(g(A^*, B^*) = g(A, B^*)\), and from Lemma 5, \(III = 0\) implies \(B^* = B\). Combining these two results gives \(g(A^*, B^*) = g(A, B^*) = g(A, B)\), and thus, inequality (16) holds (with the two sides being equal).

So, we assume \(II + III > 0\). In this case, \(II + III = \Omega(\delta)\) unless either \(Y = XA B^*\), which automatically gives (16), or \(A^* = 0, B^* = 0\), which cannot happen by assumption. Therefore as \(\delta \to 0^+\), \(II + III + IV \geq 0\) and we get \(g(A, B) > g(A^*, B^*)\). \(\blacksquare\)

**D. Proof of Theorem 3**

So far, we have discussed results which are based on the uniqueness of the solution of problem (13). We can also study the solution uniqueness of problem (14). If we show such uniqueness exists, then we can exploit the following result:

**Lemma 6.** If the solutions to both problems (13) and (14) are unique, as well as the results of Theorem 2, we have:

\[
\lim_{i \to \infty} ||C_{i+1} - C_i|| = 0,
\]

and the accumulation points form a compact continuum (i.e., the set of accumulation points is connected and compact).

**Proof:** Since \(C_{i+1} \in A(C_i)\) and the solutions to both subproblems are unique, if \(C_{i+1} \neq C_i\), then we have \(f(C_{i+1}) < f(C_i)\). Now, assume that \(||C_{i+1} - C_i|| > \delta\) for infinitely many \(i \in \mathbb{N}\). Also assume that sequences \(\{C_i\}_{i \in \mathbb{N}}\) and \(\{C_{i+1}\}_{i \in \mathbb{N}}\) have limit points \(C^*\) and \(C^+\), respectively. Thus, we have \(||C^* - C^+|| > \delta\) and in particular \(C^* \neq C^+\). From Lemma 1 we get \(C^+ \in A(C^*)\), and using Proposition 1 we conclude \(f(C^+) = f(C^*)\). Since the solutions to both problems (13) and (14) are unique, we get \(C^+ = C^*\), which is a contradiction. \(\blacksquare\)

**Lemma 7.** The solution of (14) is unique if for each accumulation point of the sequence \(\{(A_i, B_i)\}_{i \in \mathbb{N}}\), say \((A^*, B^*)\), matrix \(B^*\) has full row rank.
Proof: As shown in [21], a sufficient condition for the solution uniqueness of (14) is that the function \( g(U) = \|Y - UB\|_F^2 \) be strictly convex (the function argument is defined as \( U = XA \)) and the entries of \( X \in \mathbb{R}^{n \times p} \) are drawn from a continuous probability distribution on \( \mathbb{R}^{np} \). \( g(U) \) is strictly convex in \( U \) iff for any \( 0 < \alpha < 1 \), we have the following:

\[
g(\alpha U + (1 - \alpha) V) < \alpha g(U) + (1 - \alpha) g(V)
\]

Replacing \( g(U) = \|Y - UB\|_F^2 \), we get:

\[
\text{trace} \left\{ (Y - (\alpha U + (1 - \alpha) V)B)'^T(Y - (\alpha U + (1 - \alpha) V)B) \right\} < \alpha \text{trace} \left\{ (Y - UB)'(Y - UB) \right\} + (1 - \alpha) \text{trace} \left\{ (Y - VB)'(Y - VB) \right\}
\]

Performing some algebra yields:

\[
0 < \text{trace} \left\{ B'(U - V)'(U - V)B \right\} = \| (U - V)B \|_F^2
\]

Given that the norm is always greater than or equal to zero, for strict convexity, we need to show that \( (U - V)B \neq 0 \), unless \( U = V \). This holds iff \( B \) has full row rank (i.e., iff \( BB'^T \) is invertible).

\[ \blacksquare \]

**Theorem 3.** If the entries of \( X \in \mathbb{R}^{n \times p} \) are drawn from a continuous probability distribution on \( \mathbb{R}^{np} \) and if for any accumulation point \( C^* = (A^*, B^*) \), matrix \( B^* \) has full row rank, in addition to the results of Theorem 2, we have:

\[
\lim_{i \to \infty} \| C_{i+1} - C_i \| = 0,
\]

and the accumulation points form a compact continuum (i.e., the set of accumulation points is connected and compact).

\[ \blacksquare \]

REFERENCES

[1] C.-F. Lee and J. Lee, *Handbook of quantitative finance and risk management*. Springer Science & Business Media, 2010.

[2] L. R. Monteiro, “Multivariate regression models and geometric morphometrics: the search for causal factors in the analysis of shape,” *Systematic Biology*, pp. 192–199, 1999.

[3] S. Ranhao, Z. Baiping, and T. Jing, “A multivariate regression model for predicting precipitation in the daqing mountains,” *Mountain Research and Development*, vol. 28, no. 3, pp. 318–325, 2008.

[4] S. Wold, M. Sjöström, and L. Eriksson, “PLS-regression: a basic tool of chemometrics,” *Chemometrics and Intelligent Laboratory Systems*, vol. 58, no. 2, pp. 109–130, 2001.

[5] L. Harrison, W. D. Penny, and K. Friston, “Multivariate autoregressive modeling of fmri time series,” *NeuroImage*, vol. 19, no. 4, pp. 1477–1491, 2003.

[6] R. J. Tibshirani, “Regression shrinkage and selection via the lasso: a retrospective,” *Journal of the Royal Statistical Society: Series B*, vol. 73, no. 3, pp. 273–282, 2011.

[7] M. J. Wainwright, “Sharp thresholds for high-dimensional and noisy sparsity recovery using constrained quadratic programming (LASSO),” *IEEE Transactions on Information Theory*, vol. 55, no. 5, pp. 2183–2202, 2009.

[8] B. A. Turlach, W. N. Venables, and S. J. Wright, “Simultaneous variable selection,” *Technometrics*, vol. 47, no. 3, pp. 349–363, 2005.

[9] C.-H. Zhang and J. Huang, “The sparsity and bias of the lasso selection in high-dimensional linear regression,” *The Annals of Statistics*, vol. 36, no. 4, pp. 1567–1594, 2008.

[10] M. Yuan and Y. Lin, “Model selection and estimation in regression with grouped variables,” *Journal of the Royal Statistical Society, Series B*, vol. 68, no. 1, pp. 69–77, 2006.

[11] G. Obozinski, M. J. Wainwright, M. I. Jordan et al., “Support union recovery in high-dimensional multivariate regression,” *The Annals of Statistics*, vol. 39, no. 1, pp. 1–47, 2011.

[12] A. Jalali, S. Sanghavi, C. Ruan, and P. K. Ravikumar, “A dirty model for multi-task learning,” in *Advances in Neural Information Processing Systems*. 2010, pp. 964–972.

[13] J. Peng, J. Zhu, A. Bergamaschi, W. Han, D.-Y. Noh, J. R. Pollack, and P. Wang, “Regularized multivariate regression for identifying master predictors with application to integrative genomics study of breast cancer,” *The Annals of Applied Statistics*, vol. 4, no. 1, pp. 53–77, 2010.

[14] J. Chen, J. Zhou, and J. Ye, “Integrating low-rank and group-sparse structures for robust multi-task learning,” in *Proc. ACM Int. Conf. on Knowledge Discovery and Data Mining*, 2011, pp. 42–50.

[15] V. Chandrasekaran, S. Sanghavi, P. A. Parrilo, and A. S. Willsky, “Rank-sparsity incoherence for matrix decomposition,” *SIAM Journal on Optimization*, vol. 21, no. 2, pp. 572–596, 2011.

[16] M. Pourahmadi, *High-Dimensional Covariance Estimation: With High-Dimensional Data*, ser. Wiley Series in Probability and Statistics. Wiley, 2013.

[17] L. Chen and J. Z. Huang, “Sparse reduced-rank regression for simultaneous dimension reduction and variable selection,” *Journal of the American Statistical Association*, vol. 107, no. 500, pp. 1533–1545, 2012.

[18] H. Chun and S. Keleq, “Sparse partial least squares regression for simultaneous dimension reduction and variable selection,” *Journal of the Royal Statistical Society: Series B*, vol. 72, no. 1, pp. 3–25, 2010.

[19] M. Yuan, A. Ekici, Z. Lu, and R. Monteiro, “Dimension reduction and coefficient estimation in multivariate linear regression,” *Journal of the Royal Statistical Society, Series B*, vol. 69, no. 3, pp. 329–346, 2007.

[20] G. C. Reinsel and R. P. Velu, *Multivariate reduced-rank regression*. Springer, 1998.

[21] R. J. Tibshirani, “The lasso problem and uniqueness,” *Electronic Journal of Statistics*, vol. 7, no. 0, pp. 1456–1490, 2013.

[22] H. Zou, “The Adaptive Lasso and Its Oracle Properties,” *Journal of the American Statistical Association*, vol. 101, no. 476, pp. 1418–1429, 2006.

[23] A. J. Rothman, E. Levina, and J. Zhu, “Sparse multivariate regression with covariance estimation,” *Journal of Computational and Graphical Statistics*, vol. 19, no. 4, pp. 947–962, 2010.

[24] Standard and Poor’s. (2013, January) S&P 500 factsheet. [Online]. Available: http://bit.ly/1QAVSM

[25] C. Franço and J.-M. Zakosan, *GARCH models: structure, statistical inference and financial applications*. John Wiley & Sons, 2011.

[26] GICS. (2013, February) MSCI-Barra GICS Tables. [Online]. Available: http://www.msci.com/products/indexes/sector/gics

[27] J. Gorski, P. Pfeuffer, and K. Klarmuth, “Biconvex sets and optimization with biconvex functions: a survey and extensions,” *Mathematical Methods of Operations Research*, vol. 66, no. 3, pp. 373–407, 2007.