

# Fast envelope algorithms

R. Dennis Cook\* and Xin Zhang†

## Abstract

In this paper, we develop new fast algorithms for envelope estimation that are stable and can be used in contemporary complex envelope estimation problems. Under the sequential 1D envelope algorithm framework of Cook and Zhang (2016), we develop an envelope coordinate descent (ECD) algorithm that is shown to be much faster than the existing 1D algorithm without loss of accuracy. We also propose a novel class of envelope component screening (ECS) algorithms that serve as a screening step that can further significantly speed computation and that shows promise as precursor methodology when  $n \leq p$ . The ECD and ECS algorithms have both shown promising performance in extensive simulation studies and a data analysis.

**Key Words:** Envelope models; Grassmannian; reducing subspace.

## 1 Introduction

The notion of an envelope was introduced by Cook et al. (2010) for response reduction in multivariate linear models, subsequently studied by Cook et al. (2013) for predictor reduction where they connected envelopes with partial least squares regression, and recently combined with reduced-rank regression by Cook et al. (2015a). Envelope methods increase efficiency in estimation and improve prediction by enveloping the information in the data that is material to estimation, while excluding the information that is immaterial. The improvement in estimation

---

\*R. Dennis Cook is Professor, School of Statistics, University of Minnesota, Minneapolis, MN 55455 (E-mail: dennis@stat.umn.edu).

†Xin Zhang is Assistant Professor, Department of Statistics, Florida State University, Tallahassee, FL, 32306 (Email: henry@stat.fsu.edu).

22 and prediction can be quite substantial, as illustrated by many studies in the literature. Envelope  
23 methodology has been adapted to allow simultaneous response and predictor reduction in mul-  
24 tivariate linear regression, Cook and Zhang (2015b), extended beyond linear regression models  
25 to generic multivariate parameter estimation problems, Cook and Zhang (2015a), and to tensor  
26 (multi-dimensional array) regression in neuroimaging applications Li and Zhang (2016); Zhang  
27 and Li (2016).

28 An envelope is a subspace onto which we project the multivariate parameter vector, matrix  
29 or tensor. For a given envelope dimension  $u$ , the construction of an envelope typically involves  
30 a non-convex optimization problem over a  $u$ -dimensional Grassmannian. Such optimization  
31 requires a good starting value, an initial guess of the manifold, and can be very expensive com-  
32 putationally. Cook and Zhang (2016) proposed a relatively fast and stable envelope algorithm  
33 called the 1D algorithm, which breaks down the  $u$ -dimensional Grassmannian optimization to  
34 a sequence of  $u$  one-dimensional optimizations. The 1D algorithm requires no initial guess,  
35 yields  $\sqrt{n}$ -consistent estimators under mild conditions and was demonstrated to be much faster  
36 than a commonly used algorithm based on direct optimization over the appropriate Grassman-  
37 nian, which is the basis for the *envlp* toolbox of Cook et al. (2015b).

38 The recent advances in adapting envelopes to ever more complex settings come with added  
39 computational burdens. While existing algorithms can be applied in these contemporary con-  
40 texts, computational speed is a major obstacle. Our overarching goal is to provide fast envelope  
41 algorithms without sacrificing significantly on accuracy. Here, we propose a screening algo-  
42 rithm, called envelope component screening (ECS), that reduces the original dimension  $p$  to a

43 manageable dimension  $d \leq n$ , without losing notable structural information on the envelope;  
 44 we design an envelope coordinate descent (ECD) algorithm that can be incorporated into the  
 45 1D algorithm framework and that stabilizes and significantly speeds up the existing 1D algo-  
 46 rithm without loss of any accuracy and potentially improves the accuracy. These algorithms  
 47 can be implemented straightforwardly, we have posted our Matlab code at the author's website  
 48 (<http://ani.stat.fsu.edu/~henry/Software.html>), along with a simple tuto-  
 49 rial about how to use and modify the code (e.g. changing the tolerance level and the maximum  
 50 number of iterations).

51 The rest of the paper is organized as follows. In Section 2, we review the basic definition  
 52 and properties of envelopes, envelope regression, and the 1D envelope algorithm. In Section 3,  
 53 we develop the ECS and the ECD algorithms and their variants. Section 4 contains some  
 54 simulation studies and a data analysis from near-infrared spectroscopy. Proofs are included in  
 55 the Online Supplementary Materials.

56 The following notations and definitions are used in our exposition. Let  $\mathbb{R}^{m \times n}$  be the set  
 57 of all real  $m \times n$  matrices and let  $\mathbb{S}^{p \times p}$  be the set of all real  $p \times p$  symmetric matrices. The  
 58 Grassmannian consisting of the set of all  $u$ -dimensional subspaces of  $\mathbb{R}^p$ ,  $u \leq p$ , is denoted as  
 59  $\mathcal{G}_{p,u}$ . If  $\mathbf{M} \in \mathbb{R}^{m \times n}$ , then  $\text{span}(\mathbf{M}) \subseteq \mathbb{R}^m$  is the subspace spanned by columns of  $\mathbf{M}$ . We use  
 60  $\mathbf{P}_{\mathcal{A}} \equiv \mathbf{P}_{\mathbf{A}} = \mathbf{A}(\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T$  to denote the projection onto  $\mathcal{A} \equiv \text{span}(\mathbf{A})$  and let  $\mathbf{Q}_{\mathbf{A}} = \mathbf{I} - \mathbf{P}_{\mathbf{A}}$   
 61 denote the projection onto the orthogonal complement subspace  $\mathcal{A}^\perp$ .

## 2 A brief review of envelop estimation

### 2.1 Definition of an envelope

In this section we briefly review definitions and some properties of reducing subspaces and envelopes.

**Definition 1.** A subspace  $\mathcal{R} \subseteq \mathbb{R}^p$  is said to be a reducing subspace of  $\mathbf{M} \in \mathbb{R}^{p \times p}$  if  $\mathcal{R}$  decomposes  $\mathbf{M}$  as  $\mathbf{M} = \mathbf{P}_{\mathcal{R}}\mathbf{M}\mathbf{P}_{\mathcal{R}} + \mathbf{Q}_{\mathcal{R}}\mathbf{M}\mathbf{Q}_{\mathcal{R}}$ . If  $\mathcal{R}$  is a reducing subspace of  $\mathbf{M}$ , we say that  $\mathcal{R}$  reduces  $\mathbf{M}$ .

This definition of a reducing subspace is equivalent to the usual definition found in functional analysis, Conway (1990), and in the literature on invariant subspaces, but the underlying notion of reduction is incompatible with how it is usually understood in statistics. Nevertheless, it is common terminology in those areas and is the basis for the definition of an envelope, see Cook et al. (2010); Cook and Zhang (2015a) for example, which is central to our developments.

**Definition 2.** Let  $\mathbf{M} \in \mathbb{S}^{p \times p}$  and let  $\mathcal{U} \subseteq \text{span}(\mathbf{M})$ . Then the  $\mathbf{M}$ -envelope of  $\mathcal{U}$ , denoted by  $\mathcal{E}_{\mathbf{M}}(\mathcal{U})$ , is the intersection of all reducing subspaces of  $\mathbf{M}$  that contain  $\mathcal{U}$ .

The intersection of two reducing subspaces of  $\mathbf{M}$  is still a reducing subspace of  $\mathbf{M}$ . This means that  $\mathcal{E}_{\mathbf{M}}(\mathcal{U})$ , which is unique by its definition, is the smallest reducing subspace containing  $\mathcal{U}$ . Also, the  $\mathbf{M}$ -envelope of  $\mathcal{U}$  always exists because of the requirement  $\mathcal{U} \subseteq \text{span}(\mathbf{M})$ . If  $\text{span}(\mathbf{U}) = \mathcal{U}$  for some matrix  $\mathbf{U}$ , then we write  $\mathcal{E}_{\mathbf{M}}(\mathbf{U}) := \mathcal{E}_{\mathbf{M}}(\mathcal{U})$  to avoid notation proliferation. We let  $\mathcal{E}_{\mathbf{M}}^{\perp}(\mathbf{U})$  denote the orthogonal complement of  $\mathcal{E}_{\mathbf{M}}(\mathbf{U})$ .

A result from Cook et al. (2010) gives a characterization of envelopes.

82 **Proposition 1.** *If  $\mathbf{M} \in \mathbb{S}^{p \times p}$  has  $q \leq p$  eigenspaces, then the  $\mathbf{M}$ -envelope of  $\mathcal{U} \subseteq \text{span}(\mathbf{M})$*   
83 *can be constructed as  $\mathcal{E}_{\mathbf{M}}(\mathcal{U}) = \sum_{i=1}^q \mathbf{P}_i \mathcal{U}$ , where  $\mathbf{P}_i$  is the projection onto the  $i$ -th eigenspace*  
84 *of  $\mathbf{M}$ .*

85 If the eigenvalues of  $\mathbf{M}$  are distinct so  $q = p$  then it follows from this proposition that  
86 the  $\mathbf{M}$ -envelope of  $\mathcal{U}$  is the sum of the eigenspaces of  $\mathbf{M}$  that are not orthogonal to  $\mathcal{U}$ . This  
87 implies that when  $q = p$  the envelope is the span of some subset of the eigenspaces of  $\mathbf{M}$ . In  
88 the regression context,  $\mathcal{U}$  is typically the span of a regression coefficient matrix or a matrix of  
89 cross-covariances, and  $\mathbf{M}$  is chosen as a covariance matrix which is usually positive definite.

## 90 2.2 The 1D algorithm

91 In this section, we review the 1D algorithm, Cook and Zhang (2016), in terms of estimating  
92 a generic envelope  $\mathcal{E}_{\mathbf{M}}(\mathbf{U})$ , where  $\mathbf{M} > 0$  and  $\mathbf{U} \geq 0$  are both in  $\mathbb{S}^{p \times p}$ . Then  $\text{span}(\mathbf{U}) \subseteq$   
93  $\text{span}(\mathbf{M}) = \mathbb{R}^p$  and the envelope is well-defined. A generic objective function  $F$  was proposed  
94 by Cook and Zhang (2016) for estimating  $\mathcal{E}_{\mathbf{M}}(\mathbf{U})$ :

$$F(\mathbf{G}) = \log |\mathbf{G}^T \mathbf{M} \mathbf{G}| + \log |\mathbf{G}^T (\mathbf{M} + \mathbf{U})^{-1} \mathbf{G}|, \quad (2.1)$$

95 where  $\mathbf{G} \in \mathbb{R}^{p \times u}$  is semi-orthogonal with given envelope dimension  $0 \leq u \leq p$ . Since  
96  $F(\mathbf{G}) = F(\mathbf{G}\mathbf{O})$  for any orthogonal  $u \times u$  matrix  $\mathbf{O}$ , the minimizer of  $F(\mathbf{G})$  is not unique and  
97 the above optimization is essentially over  $\mathcal{G}_{p,u}$ . However, we are interested only in the span of  
98 the minimizer, which is unique as shown in the following proposition from Cook and Zhang  
99 (2016).

---

**Algorithm 1** The 1D algorithm (Cook and Zhang, 2016).

---

Let  $\mathbf{g}_k \in \mathbb{R}^p$ ,  $k = 1, \dots, u$ , be the sequential directions obtained. Let  $\mathbf{G}_k = (\mathbf{g}_1, \dots, \mathbf{g}_k)$ , let  $(\mathbf{G}_k, \mathbf{G}_{0k})$  be an orthogonal basis for  $\mathbb{R}^p$  and set initial value  $\mathbf{g}_0 = \mathbf{G}_0 = 0$ .

For  $k = 0, \dots, u - 1$ , repeat Step 1 and 2 in the following.

1. Let  $\mathbf{G}_k = (\mathbf{g}_1, \dots, \mathbf{g}_k)$ , and let  $(\mathbf{G}_k, \mathbf{G}_{0k})$  be an orthogonal basis for  $\mathbb{R}^p$ . Set  $\mathbf{N}_k = [\mathbf{G}_{0k}^T(\mathbf{M} + \mathbf{U})\mathbf{G}_{0k}]^{-1}$ ,  $\mathbf{M}_k = \mathbf{G}_{0k}^T \mathbf{M} \mathbf{G}_{0k}$  and the unconstrained objective function

$$\phi_k(\mathbf{w}) = \log(\mathbf{w}^T \mathbf{M}_k \mathbf{w}) + \log(\mathbf{w}^T \mathbf{N}_k \mathbf{w}) - 2 \log(\mathbf{w}^T \mathbf{w}). \quad (2.2)$$

2. Solve  $\mathbf{w}_{k+1} = \arg \min \phi_k(\mathbf{w})$ , then the  $(k + 1)$ -th envelope direction is  $\mathbf{g}_k = \mathbf{G}_{0k} \mathbf{w}_{k+1} / \|\mathbf{w}_{k+1}\|$ .
- 

100 **Proposition 2.** Let  $\tilde{\Gamma}$  be any minimizer of  $F(\mathbf{G})$ . Then  $\text{span}(\tilde{\Gamma}) = \mathcal{E}_{\mathbf{M}}(\mathbf{U})$ .

101 When  $u$  is large, the minimization of (2.1) can be computationally expensive and it re-  
 102 quires a good initial value to avoid local minima. Algorithm 1 summarizes the 1D algorithm  
 103 which breaks down the optimization of (2.1) to “one-direction-at-a-time”. We review the  $\sqrt{n}$ -  
 104 consistency of Algorithm 1 that was established by Cook and Zhang (2016) and is the theoretic-  
 105 cal foundation to the  $\sqrt{n}$ -consistency of our ECD algorithm (Corollary 2).

106 **Theorem 1.** Suppose  $\mathbf{M} > 0$ ,  $\mathbf{U} \geq 0$  and  $\widehat{\mathbf{M}}$  and  $\widehat{\mathbf{U}}$  are  $\sqrt{n}$ -consistent estimators for  $\mathbf{M}$  and  
 107  $\mathbf{U}$ . Let  $\widehat{\mathbf{G}}_u$  denote the estimator obtained from Algorithm 1 with  $\widehat{\mathbf{M}}$ ,  $\widehat{\mathbf{U}}$  and the true envelope  
 108 dimension  $u$ . Then  $\mathbf{P}_{\widehat{\mathbf{G}}_u}$  is  $\sqrt{n}$ -consistent for the projection onto  $\mathcal{E}_{\mathbf{M}}(\mathbf{U})$ .

### 109 2.3 Envelope regression and parameter estimation

110 In the multivariate linear regression context of  $\mathbf{Y} = \boldsymbol{\alpha} + \boldsymbol{\beta}\mathbf{X} + \boldsymbol{\epsilon}$ , the envelope  $\mathcal{E}_{\mathbf{M}}(\mathbf{U})$  is  
 111 constructed based on whether we want to reduce the predictors, Cook et al. (2013), or the  
 112 response variables, Cook et al. (2010), or even both sets of variables simultaneously, Cook

113 and Zhang (2015b). Then  $\mathbf{M}$  is chosen to be the covariance matrix of  $\mathbf{X}$ ,  $\Sigma_{\mathbf{X}} \equiv \text{cov}(\mathbf{X})$ , or  
 114 the conditional covariance of  $\mathbf{Y}$  given  $\mathbf{X}$ ,  $\Sigma \equiv \text{cov}(\mathbf{Y} \mid \mathbf{X}) = \text{cov}(\epsilon)$ , or the direct sum of  
 115 the two,  $\Sigma_{\mathbf{X}} \oplus \Sigma$ . Accordingly,  $\mathbf{U}$  may be chosen as  $\beta^T \beta$ ,  $\beta \beta^T$ , or  $\beta^T \beta \oplus \beta \beta^T$ . When  
 116 additional structural information is available, the envelope construction can be adjusted to gain  
 117 more efficiency. For instance, a partial envelope, Su and Cook (2011), is used when only a  
 118 subset of predictors is of special interest. A reduced-rank envelope, Cook et al. (2015a), is  
 119 appropriate when regression coefficient matrix  $\beta$  is rank deficient and multivariate reduced-  
 120 rank regression is preferred over ordinary least squares regression. See Cook and Zhang (2016)  
 121 for an introductory example of the working mechanism of envelope regression and for a more  
 122 detailed discussion of the connections between various envelopes and the choice of  $\mathbf{M}$  and  $\mathbf{U}$ .  
 123 Beyond regression models, envelope estimation is a way to improve estimative efficiency in  
 124 multivariate parameter estimation problems, as described by Cook and Zhang (2015a). In this  
 125 more general context, the envelope can still be estimated from objective function (2.1) with  
 126 different choices for  $\mathbf{M}$  and  $\mathbf{U}$ .

### 127 **3 Two envelope component-wise algorithms**

128 In this section, we introduce two moment-based and model-free envelope algorithms: an en-  
 129 velope component screening (ECS) algorithm and an envelope coordinate descent (ECD) algo-  
 130 rithm. The ECS algorithm allows for screening out eigenvectors of  $\mathbf{M}$  lying in  $\mathcal{E}_{\mathbf{M}}^{\perp}(\mathbf{U})$ . Since  
 131 the ECS algorithm is computationally efficient and robust, it is applicable to situations where  
 132  $n \lesssim p$  or even  $n \ll p$  and it reduces the dimension  $p$  to a lower dimension  $d < n$  such that the

---

**Algorithm 2** The envelope component screening (ECS) algorithm.

---

1. Construct an eigenvalue decomposition of  $\mathbf{M}$  as  $\mathbf{M} = \sum_{i=1}^p \lambda_i \mathbf{v}_i \mathbf{v}_i^T$ , where  $\mathbf{v}_i^T \mathbf{v}_j$  equals 1 if  $i = j$  and 0 otherwise.
  2. Evaluate  $f_i = F(\mathbf{v}_i) = \log(\lambda_i) + \log(\mathbf{v}_i^T (\mathbf{M} + \mathbf{U})^{-1} \mathbf{v}_i)$ , and then order them as  $f_{(p)} \leq \dots \leq f_{(1)} \leq 0$  with corresponding  $\mathbf{v}_{(i)}$ .
  3. Let  $\mathbf{A}_0 = (\mathbf{v}_{(1)}, \dots, \mathbf{v}_{(p-d)})^T$  and  $\mathbf{A} = (\mathbf{v}_{(p-d+1)}, \dots, \mathbf{v}_{(p)}) \in \mathbb{R}^{p \times d}$  with a pre-specified number  $d$ .
  4. Estimate  $\mathcal{E}_{\mathbf{M}}(\mathbf{U})$  as  $\mathbf{A} \mathcal{E}_{\mathbf{A}^T \mathbf{M} \mathbf{A}}(\mathbf{A}^T \mathbf{U} \mathbf{A})$ .
- 

133 1D algorithm is applicable. The ECD algorithm, on the other hand, is a refined algorithm that  
 134 is adapted into the 1D algorithm framework and speeds up each iteration of the 1D algorithm.  
 135 In this section, we assume that  $\mathbf{M} > 0$  and  $\mathbf{U} \geq 0$  in all the algorithmic and theoretical results.

### 136 3.1 The ECS algorithm

137 Here and in later statements we use the objective function  $F(\cdot)$  defined at (2.1), but we no longer  
 138 require the column dimension of its argument to be a given envelope dimension.

139 **Proposition 3.** *Let  $\mathbf{A}$  be a semi-orthogonal basis matrix for any reducing subspace of  $\mathbf{M}$  and  
 140 let  $(\mathbf{A}, \mathbf{A}_0) \in \mathbb{R}^{p \times p}$  be an orthogonal matrix. Then  $F(\mathbf{A}_0) \leq 0$ , and  $F(\mathbf{A}_0) = 0$  if and only if  
 141  $\text{span}(\mathbf{U}) \subseteq \text{span}(\mathbf{A})$ . In addition, if  $F(\mathbf{A}_0) = 0$  then  $\mathcal{E}_{\mathbf{M}}(\mathbf{U}) = \mathbf{A} \mathcal{E}_{\mathbf{A}^T \mathbf{M} \mathbf{A}}(\mathbf{A}^T \mathbf{U} \mathbf{A})$ .*

142 Proposition 3 provides support for the moment-based objective function (2.1), and it in-  
 143 spired a way of detecting and eliminating components in  $\mathcal{E}_{\mathbf{M}}^{\perp}(\mathbf{U})$ : if we can find an  $\mathbf{A}_0$  such  
 144 that  $F(\mathbf{A}_0) = 0$  then Proposition 3 implies that  $\mathcal{E}_{\mathbf{M}}(\mathbf{U}) \subseteq \text{span}(\mathbf{A})$  and that we can find  
 145  $\mathcal{E}_{\mathbf{M}}(\mathbf{U})$  by pursuing the lower dimension envelope  $\mathcal{E}_{\mathbf{A}^T \mathbf{M} \mathbf{A}}(\mathbf{A}^T \mathbf{U} \mathbf{A})$ . Thus, Proposition 3 pro-



146 vides a foundation for eliminating parts of  $\mathcal{E}_M^\perp(\mathbf{U})$  by maximizing  $F(\mathbf{A}_0)$  over the reducing  
 147 subspaces of  $\mathbf{M}$ . In the extreme, if we can find  $\mathbf{A}_0 \in \mathbb{R}^{p \times (p-u)}$  satisfying  $F(\mathbf{A}_0) = 0$ , then  
 148  $\mathcal{E}_M(\mathbf{U}) = \text{span}(\mathbf{A})$  because  $u$  is the dimension of the envelope.

149 Proposition 3 inspired the ECS algorithm to facilitate envelope estimation by enabling us to  
 150 estimate a  $u$ -dimensional envelope within a smaller space  $\mathbb{R}^d$  instead of  $\mathbb{R}^p$ , where  $u \leq d < p$ .  
 151 We state the population version of the ECS algorithm in Algorithm 2, while the sample version  
 152 uses estimators  $\widehat{\mathbf{M}}$  and  $\widehat{\mathbf{U}}$  instead of  $\mathbf{M}$  and  $\mathbf{U}$ . Step 1 of the ECS algorithm constructs an  
 153 eigen-decomposition of  $\mathbf{M}$ . Step 2 of the algorithm orders the eigenvectors of  $\mathbf{M}$  by their value  
 154 of  $F(\mathbf{v}_i)$ , where  $F$  is as defined in (2.1). The value  $f_i \equiv F(\mathbf{v}_i)$  can be viewed as a negative  
 155 pseudo-log-likelihood, which achieves its maximum of zero if and only if  $\mathbf{v}_i \in \mathcal{E}_M^\perp(\mathbf{U})$ . Hence  
 156 the ordered series  $f_{(p)} \leq \dots \leq f_{(1)} \leq 0$  in Step 2 ranks  $\mathbf{v}_{(i)}$  in terms of their ‘‘closeness’’ to  
 157  $\mathcal{E}_M^\perp(\mathbf{U})$ . Steps 3 and 4 of Algorithm 2 then determine a partition of  $(\mathbf{A}, \mathbf{A}_0)$ , where  $\text{span}(\mathbf{A})$   
 158 contains the envelope and  $\text{span}(\mathbf{A}_0)$  lies within the orthogonal complement of the envelope.  
 159 Then  $\mathbf{A}_0$  is discarded and we pursue envelope estimation via  $\mathbf{A}\mathcal{E}_{\mathbf{A}^T\mathbf{M}\mathbf{A}}(\mathbf{A}^T\mathbf{U}\mathbf{A})$ .

**Proposition 4.** *In the population ECS algorithm,*

$$f_{(p)} \leq \dots \leq f_{(p-\tilde{u}+1)} < f_{(p-\tilde{u})} = \dots = f_{(1)} = 0,$$

160 where  $\tilde{u}$  satisfies  $u \leq \tilde{u} \leq p$  and is the number of eigenvectors from the eigen-decomposition  
 161  $\mathbf{M} = \sum_{i=1}^p \lambda_i \mathbf{v}_i \mathbf{v}_i^T$  (Step 1; Algorithm 2) that are not orthogonal to  $\text{span}(\mathbf{U})$ . Moreover, if  
 162  $d \geq \tilde{u}$  is used in the algorithm then  $\mathbf{A}\mathcal{E}_{\mathbf{A}^T\mathbf{M}\mathbf{A}}(\mathbf{A}^T\mathbf{U}\mathbf{A}) = \mathcal{E}_M(\mathbf{U})$ .

163 Proposition 4 has two implications. First, the  $u$ -dimensional envelope is contained within

164 the span of  $\tilde{u}$  eigenvectors of  $\mathbf{M}$  that satisfies  $f_i = F(\mathbf{v}_i) < 0$ , whereas the other eigenvectors  
165 have  $f_i = 0$ . Secondly, for  $d \geq \tilde{u}$ , the ECS estimate of the envelope is indeed the original  
166 envelope in the population,  $\mathbf{A}\mathcal{E}_{\mathbf{A}^T\mathbf{M}\mathbf{A}}(\mathbf{A}^T\mathbf{U}\mathbf{A}) = \mathcal{E}_{\mathbf{M}}(\mathbf{U})$ . Thus, the ECS envelope estimator  
167 is Fisher consistent as long as the dimension  $d$  in the ECS algorithm is specified no less than  
168 the number  $\tilde{u}$ . Since  $\tilde{u} \geq u$ , we need to specify  $d$  such that  $d \geq \tilde{u} \geq u$ .

169 We have introduced  $\tilde{u}$  because of an identification issue related to the eigenvectors of  $\mathbf{M}$ .  
170 To gain intuition about this issue, let  $(\mathbf{\Gamma}, \mathbf{\Gamma}_0) \in \mathbb{R}^{p \times p}$  be an orthogonal matrix, where  $\mathbf{\Gamma} \in$   
171  $\mathbb{R}^{p \times u}$  is a basis matrix for  $\mathcal{E}_{\mathbf{M}}(\mathbf{U})$ . Then we can write  $\mathbf{M} = \mathbf{\Gamma}\mathbf{\Omega}\mathbf{\Gamma}^T + \mathbf{\Gamma}_0\mathbf{\Omega}_0\mathbf{\Gamma}_0^T$  and  $\mathbf{U} =$   
172  $\mathbf{\Gamma}\mathbf{\Phi}\mathbf{\Gamma}^T$ , where  $\mathbf{\Omega}, \mathbf{\Omega}_0 > 0$  and  $\mathbf{\Phi} \geq 0$ . If there is an eigenvalue of  $\mathbf{M}$  corresponding to a  
173 two-dimensional eigenspace spanned by eigenvectors  $\mathbf{u} \in \text{span}(\mathbf{\Gamma})$  and  $\mathbf{w} \in \text{span}(\mathbf{\Gamma}_0)$ , then  
174  $F(\mathbf{u}) > 0$  and  $F(\mathbf{w}) = 0$ . However, because the eigen-decomposition is not unique, for this  
175 particular eigenvalue we may also get eigenvectors  $\mathbf{v}_1 = \mathbf{u} + \mathbf{w}$  and  $\mathbf{v}_2 = \mathbf{u} - \mathbf{w}$  that lie  
176 in neither  $\text{span}(\mathbf{\Gamma})$  nor  $\text{span}(\mathbf{\Gamma}_0)$ , and thus  $F(\mathbf{v}_1) > 0$  and  $F(\mathbf{v}_2) > 0$ . An extreme case is  
177  $\mathbf{M} = \mathbf{I}_p$ , if we form eigenvectors of  $\mathbf{M}$  as columns of  $(\mathbf{\Gamma}, \mathbf{\Gamma}_0) \in \mathbb{R}^{p \times p}$ ,  $(\mathbf{v}_1, \dots, \mathbf{v}_p) = (\mathbf{\Gamma}, \mathbf{\Gamma}_0)$ ,  
178 then  $F(\mathbf{v}_i) > 0$  for  $i = 1, \dots, u$  and  $F(\mathbf{v}_i) = 0$  for  $i = u + 1, \dots, p$ . On the other hand, any  
179 orthogonal matrix  $\mathbf{O} = (\mathbf{o}_1, \dots, \mathbf{o}_p) \in \mathbb{R}^{p \times p}$  forms a set of eigenvectors for  $\mathbf{M} = \mathbf{I}_p$  but it is  
180 possible that  $F(\mathbf{o}_i) > 0$  for all  $i = 1, \dots, p$ .

181 **Proposition 5.** *If  $\mathbf{M}$  has  $p$  distinct eigenvalues, or, if all eigenspaces of  $\mathbf{M}$  are contained*  
182 *completely in either  $\mathcal{E}_{\mathbf{M}}(\mathbf{U})$  or  $\mathcal{E}_{\mathbf{M}}^\perp(\mathbf{U})$ , then  $u = \tilde{u}$  for any eigen-decomposition in the ECS*  
183 *algorithm. Depending on the particular eigen-decomposition in the ECS algorithm,  $\tilde{u}$  can be*

184 any integer from  $\{u, u + 1, \dots, u + K\}$ , where  $K$  is the sum of the dimensions of eigenspaces  
 185 of  $\mathbf{M}$  that intersect both  $\mathcal{E}_{\mathbf{M}}(\mathbf{U})$  and  $\mathcal{E}_{\mathbf{M}}^{\perp}(\mathbf{U})$ .

186 The number  $\tilde{u}$  of nonzero  $f_i$ 's in the ECS algorithm is unique and equal to  $u$  for all possible  
 187 eigen-decompositions of  $\mathbf{M}$  when all eigenspaces of  $\mathbf{M}$  are contained completely in either  
 188  $\mathcal{E}_{\mathbf{M}}(\mathbf{U})$  or  $\mathcal{E}_{\mathbf{M}}^{\perp}(\mathbf{U})$ . However,  $\tilde{u}$  is no longer unique if some eigenspace of  $\mathbf{M}$  intersects non-  
 189 trivially with both  $\mathcal{E}_{\mathbf{M}}(\mathbf{U})$  and  $\mathcal{E}_{\mathbf{M}}^{\perp}(\mathbf{U})$ : some eigen-decomposition yields  $\tilde{u} = u$  and others  
 190 may get  $\tilde{u} > u$ . Since  $d \geq \tilde{u}$  is needed for the Fisher consistency of the ECS algorithm, the  
 191 dimension reduction achieved by the ECS algorithm can be somewhere between  $(p - u)$  and  
 192  $(p - u - K)$  subject to the particular eigen-decompositions.

193 In the sample version of the algorithm, estimators  $\widehat{\mathbf{M}}$  and  $\widehat{\mathbf{U}}$  are substituted into Algorithm  
 194 2. Let  $\widehat{\mathbf{A}}$  and  $\widehat{\mathbf{A}}_0$  be the estimators from the sample ECS algorithm. Based on Proposition 3,  
 195 we want  $F(\widehat{\mathbf{A}}_0) \rightarrow 0$  as  $n \rightarrow \infty$  so that the components to be discarded,  $\widehat{\mathbf{A}}_0$ , are orthogonal  
 196 to the envelope, and the remaining components of  $\text{span}(\widehat{\mathbf{A}})$  converge to a reducing subspace of  
 197  $\mathbf{M}$  that contains  $\text{span}(\mathbf{U})$ . We have the sample objective function

$$F_n(\widehat{\mathbf{A}}_0) = \log |\widehat{\mathbf{A}}_0^T \widehat{\mathbf{M}} \widehat{\mathbf{A}}_0| + \log |\widehat{\mathbf{A}}_0^T (\widehat{\mathbf{M}} + \widehat{\mathbf{U}})^{-1} \widehat{\mathbf{A}}_0|$$

198 available instead of the population objective function  $F(\widehat{\mathbf{A}}_0)$ , so we need to show  $F_n(\widehat{\mathbf{A}}_0) \rightarrow 0$   
 199 as  $n \rightarrow \infty$  similar to the convergence of  $F(\widehat{\mathbf{A}}_0)$ .

200 **Proposition 6.** Suppose  $\widehat{\mathbf{M}}$  and  $\widehat{\mathbf{U}}$  are  $\sqrt{n}$ -consistent estimators for  $\mathbf{M} > 0$  and  $\mathbf{U} \geq 0$ .  
 201 If  $d \geq \tilde{u}$  is used in the sample ECS algorithm, then  $F(\widehat{\mathbf{A}}_0) = O_p(n^{-1/2})$  and  $F_n(\widehat{\mathbf{A}}_0) =$   
 202  $F(\widehat{\mathbf{A}}_0) + O_p(n^{-1/2})$ .

203 The number  $d$  serves as an upper bound for the envelope dimension and does not have to  
 204 be accurately specified. For instance, if we are estimating a 10-dimensional envelope in  $\mathbb{R}^{100}$ ,  
 205 it is usually reasonable to choose  $d = 50$ . In practice, we may adopt a data-driven modification  
 206 to Step 3 in the sample ECS algorithm, where the tuning parameter  $d$  is selected from the data  
 207 rather than pre-specified. Unlike selecting the envelope dimension  $u$  using information criteria  
 208 or cross-validation, the selection for  $d$  is less crucial and is performed with negligible computa-  
 209 tional cost. Since  $F_n(\widehat{\mathbf{A}}_0) \leq 0$  is monotonically increasing in the number of components  $d$ , we  
 210 can select  $d$  as the largest number such that  $F_n(\widehat{\mathbf{A}}_0) > C_0$  for some pre-specified cutoff value  
 211  $C_0 < 0$ . Because  $F_n(\widehat{\mathbf{A}}_0)$  goes to zero at rate  $\sqrt{n}$ , we could choose  $C_0$  to have a smaller order  
 212 so that no important components is missed with high probability. Based on our experience, the  
 213 cutoff value  $C_0 = -n^{-1}$  in Step 3 performs well. We conjecture that the ECS algorithm is  $\sqrt{n}$ -  
 214 consistent if  $\widehat{\mathbf{M}}$  and  $\widehat{\mathbf{U}}$  are  $\sqrt{n}$ -consistent estimators and the estimation of  $\mathcal{E}_{\widehat{\mathbf{A}}^T \widehat{\mathbf{M}} \widehat{\mathbf{A}}}(\widehat{\mathbf{A}}^T \widehat{\mathbf{U}} \widehat{\mathbf{A}})$   
 215 at the final step is from any  $\sqrt{n}$ -consistent envelope algorithm, 1D algorithm or the ECD al-  
 216 gorithm in Section 3.3. To further speed computation,  $F_n(\widehat{\mathbf{A}}_0)$  can be well approximated by  
 217  $\sum_{i=1}^{p-d} f_{(i)}$ . We illustrate this data-driven approach for selecting  $d$  in the numerical analysis in  
 218 Section 4, where  $C_0$  is chosen as  $-n^{-1}$  and  $F_n(\widehat{\mathbf{A}}_0)$  is approximated by  $\sum_{i=1}^{p-d} f_{(i)}$ . We note  
 219 that  $C_0 = -n^{-1}$  is quite conservative in most cases where  $d$  is much bigger than  $u$ . We also  
 220 varied  $C_0 = -n^{-0.5}$  to  $C_0 = -n^{-1.5}$  and the results were not sensitive to the choice of  $C_0$ .

221 The ECS algorithm is rather general and can be easily modified for specific problems of  
 222 interests, as we discuss in the next section.

## 223 3.2 Variations on the ECS algorithm

224 The following result is a useful implication of Proposition 3.

225 **Corollary 1.** *Let  $\mathbf{A}$  be a semi-orthogonal basis matrix for any reducing subspace of  $\mathbf{M} + \mathbf{U}$*   
 226 *and let  $(\mathbf{A}, \mathbf{A}_0) \in \mathbb{R}^{p \times p}$  be an orthogonal matrix. Then  $F(\mathbf{A}_0) \leq 0$ , and  $F(\mathbf{A}_0) = 0$  if and only*  
 227 *if  $\text{span}(\mathbf{U}) \subseteq \text{span}(\mathbf{A})$ . In addition, if  $F(\mathbf{A}_0) = 0$  then  $\mathcal{E}_{\mathbf{M}}(\mathbf{U}) = \mathbf{A}\mathcal{E}_{\mathbf{A}^T\mathbf{M}\mathbf{A}}(\mathbf{A}^T\mathbf{U}\mathbf{A})$ .*

228 Corollary 1 is derived straightforwardly from Proposition 3 by noticing that if  $\text{span}(\mathbf{A})$   
 229 contains  $\text{span}(\mathbf{U})$  then it reduces  $\mathbf{M}$ , which is equivalent to reducing  $\mathbf{M} + \mathbf{U}$ . It has two key  
 230 implications. First, we can replace  $\mathbf{M}$  with  $\mathbf{M} + \mathbf{U}$  in Step 1 of the ECS algorithm (Algorithm  
 231 2), leading to these alternative Steps 1 and 2 of the ECS algorithm.

- 232 1. Construct the eigenvalue decomposition of  $\mathbf{M} + \mathbf{U}$  as  $\mathbf{M} + \mathbf{U} = \sum_{i=1}^p \lambda_i \mathbf{v}_i \mathbf{v}_i^T$ , where  
 233  $\mathbf{v}_i^T \mathbf{v}_j$  equals 1 if  $i = j$  and 0 otherwise.
- 234 2. Evaluate  $f_i = F(\mathbf{v}_i) = \log(\mathbf{v}_i^T \mathbf{M} \mathbf{v}_i) - \log(\lambda_i)$ , and then order them as  $f_{(p)} \leq \dots \leq$   
 235  $f_{(1)} \leq 0$  with corresponding  $\mathbf{v}_{(i)}$ .

236 Apparently, we no longer need to compute the inverse of  $\mathbf{M} + \mathbf{U}$  in Step 2 of the ECS algorithm,  
 237 which can be helpful in high-dimensional settings. Second, in some applications the eigenvec-  
 238 tors of  $\mathbf{M} + \mathbf{U}$  might be more interpretable than those of  $\mathbf{M}$ . For example, in multivariate linear  
 239 regression  $\mathbf{Y} = \boldsymbol{\alpha} + \boldsymbol{\beta}\mathbf{X} + \boldsymbol{\epsilon}$ , the matrix  $\mathbf{M}$  is taken as  $\boldsymbol{\Sigma}_{\mathbf{X}}$  for a predictor envelope, Cook et al.  
 240 (2013). Then the original ECS algorithm, which selects principal components of  $\mathbf{X}$  according  
 241 to its closeness to  $\text{span}(\boldsymbol{\beta}^T)$ , is essentially a type of supervised principal component analysis,

242 see Bair et al. (2006); Li et al. (2015a,b) for example. If we are interested in the response en-  
 243 velopes of Cook et al. (2010) then  $\mathbf{M} = \boldsymbol{\Sigma} = \text{cov}(\boldsymbol{\epsilon})$  and  $\mathbf{M} + \mathbf{U} = \boldsymbol{\Sigma}_{\mathbf{Y}}$ , and this modified  
 244 ECS algorithm may be more interpretable because it selects among principal components of  $\mathbf{Y}$ .

245 Another important variation on the ECS algorithm is for its sample version when  $n < p$  or  
 246 even  $n \ll p$ . Sample estimators for  $\mathbf{M}$  and  $\mathbf{M} + \mathbf{U}$ , which are typically the sample covariance  
 247 matrices, are substituted in the objective function (2.1) and in the envelope algorithms. For  
 248 small sample problems where  $n < p$ , the sample matrices  $\widehat{\mathbf{M}}$  and  $\widehat{\mathbf{M}} + \widehat{\mathbf{U}}$  are typically rank  
 249 deficient with rank  $n$  or  $n - 1$  and existing envelope algorithms fail. One easy way to get  
 250 around the problem is to follow Proposition 3 and first downsize the envelope estimation of  
 251  $\mathcal{E}_{\mathbf{M}}(\mathbf{U})$  to  $\mathbf{A}\mathcal{E}_{\mathbf{A}^T\mathbf{M}\mathbf{A}}(\mathbf{A}^T\mathbf{U}\mathbf{A})$ , with the columns of  $\mathbf{A}$  as nontrivial  $n$  or  $n - 1$  eigenvectors  
 252 of  $\widehat{\mathbf{M}}$  or  $\widehat{\mathbf{M}} + \widehat{\mathbf{U}}$ . Then the ECS algorithm and other envelope estimation algorithms can be  
 253 applied. We demonstrate this in the simulations.

### 254 3.3 The ECD algorithm

255 For each direction  $\mathbf{w}_{k+1}$  in the 1D algorithm, we need to minimize  $\phi_k(\mathbf{w})$  iteratively. One  
 256 way to do this is by a nonlinear conjugate gradient method, for example the Polak-Ribiere type  
 257 conjugate gradient (PRCG) and the Fletcher-Reeves type conjugate gradient (FRCG) meth-  
 258 ods. Other optimization methods such as gradient descent, Newton-Raphson and quasi-Newton  
 259 methods can be applied as well. PRCG and FRCG methods have better performance from our  
 260 experience. If the dimension  $p$  is large, these standard methods can be expensive and ineffi-  
 261 cient, and, since the objective function  $\phi_k(\mathbf{w})$  is non-convex and has local minima, it may be

---

**Algorithm 3** The envelope coordinate descent (ECD) algorithm for solving  $\phi_k(\mathbf{w})$ .

---

1. Eigenvalue decomposition of  $\mathbf{M}_k$  as  $\mathbf{M}_k = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^T$  where  $\mathbf{V}$  is an orthogonal matrix and  $\mathbf{\Lambda} = \text{diag}(\lambda_1, \dots, \lambda_{p-k})$  is a diagonal matrix.
2. Transform the original objective function into canonical coordinates:  $\mathbf{v} \leftarrow \mathbf{V}^T \mathbf{w}$ ,  $\tilde{\mathbf{N}} \leftarrow \mathbf{V}^T \mathbf{N}_k \mathbf{V}$  and

$$\phi_k(\mathbf{w}) = \varphi_k(\mathbf{v}) = \log(\mathbf{v}^T \mathbf{\Lambda} \mathbf{v}) + \log(\mathbf{v}^T \tilde{\mathbf{N}} \mathbf{v}) - 2 \log(\mathbf{v}^T \mathbf{v}). \quad (3.1)$$

3. For  $t = 1, \dots, T_{\max}$ , where  $T_{\max}$  is the maximum number of iterations, update  $\mathbf{v}^{(t)}$  following Step 4-7 and terminate iteration if  $\varphi_k(\mathbf{v}^{(t)}) - \varphi_k(\mathbf{v}^{(t-1)}) \leq \epsilon$ , for some tolerance value  $\epsilon > 0$ . At the termination, transform back to  $\mathbf{w}_{k+1} = \arg \min \phi_k(\mathbf{w}) = \mathbf{V} \mathbf{v}$ .
4. Update  $a^{(t)} \leftarrow (\mathbf{v}^T \mathbf{\Lambda} \mathbf{v})^{-1}$ ,  $b^{(t)} \leftarrow (\mathbf{v}^T \tilde{\mathbf{N}} \mathbf{v})^{-1}$  and  $c^{(t)} \leftarrow (\mathbf{v} \mathbf{v}^T)^{-1}$  according to current stage  $\mathbf{v}^{(t)}$ .
5. For  $j = 1, \dots, p-k$ , if  $a^{(t)} \lambda_j + b^{(t)} \tilde{N}_{jj} - 2c^{(t)} \neq 0$  then consider moving each coordinate of  $\mathbf{v}$  as

$$v_j^{(t+1)} \leftarrow \left( \sum_{i \neq j}^{p-k} b^{(t)} \tilde{N}_{ij} v_i^{(t)} \right) / \left( a^{(t)} \lambda_j + b^{(t)} \tilde{N}_{jj} - 2c^{(t)} \right). \quad (3.2)$$

6. If the objective function is not decreased by moving  $v_j^{(t)}$  to  $v_j^{(t+1)}$  then back up  $v_j^{(t+1)}$  to  $v_j^{(t)}$ .
  7. If none of the coordinates is updated, then run one iteration of any standard nonlinear optimization method to update  $\mathbf{v}$ .
- 

262 hard to find an algorithm that stably minimizes it at each iteration. Here we propose a fast and  
 263 stable *envelope coordinate descent* (ECD) algorithm for solving  $\phi_k(\mathbf{w})$ . It is much faster than  
 264 any standard nonlinear optimization method and is guaranteed to not increase the value of the  
 265 objective function at each iteration. Since the ECD algorithm is built within the 1D algorithm  
 266 framework, we outline only the part of it for solving  $\phi_k(\mathbf{w})$  in (2.2) of Algorithm 1.

267 The coordinate descent algorithm can be more efficient when the objective function is

268 separable in coordinates. We transform the basis to canonical coordinate  $\mathbf{w} \mapsto \mathbf{v}$  so that  
 269 the first term in the objective function is more separable:  $\log(\mathbf{w}^T \mathbf{M}_k \mathbf{w}) \mapsto \log(\mathbf{v}^T \mathbf{\Lambda} \mathbf{v}) =$   
 270  $\log(\sum_i \lambda_i v_i^2)$ . This speeds up the algorithm and makes the optimization more accurate.

271 Step 5 in Algorithm 3 approximates the solution to  $\partial \varphi_k(\mathbf{v}) / \partial v_j = 0$ , which can be written  
 272 as

$$\frac{2\lambda_j v_j}{\mathbf{v}^T \mathbf{\Lambda} \mathbf{v}} + \frac{2 \sum_{i=1}^{p-k} \tilde{N}_{ij} v_i}{\mathbf{v}^T \tilde{\mathbf{N}} \mathbf{v}} - \frac{4v_j}{\mathbf{v}^T \mathbf{v}} = 0.$$

273 The approximate solution is obtained by treating the denominators  $\mathbf{v}^T \mathbf{\Lambda} \mathbf{v}$ ,  $\mathbf{v}^T \tilde{\mathbf{N}} \mathbf{v}$  and  $\mathbf{v}^T \mathbf{v}$  as  
 274 constants at the current step, and solving the resulting linear equation in  $v_j$  from the numerators.  
 275 Step 6 is then a back-tracking step to make sure that the objective function is monotonically  
 276 non-increasing. Step 7 guarantees that the algorithm will converge because of basic properties  
 277 of the standard nonlinear optimization method chosen in Step 7. Thus, this ECD algorithm has  
 278 a convergence rate bounded below by the convergence rate of the standard nonlinear optimiza-  
 279 tion method chosen in Step 7. Our experience suggests that the approximated solution in Step  
 280 5 is usually very close to the true minimizer for the coordinate.

281 The  $\sqrt{n}$ -consistency of the ECD algorithm follows as a result of the 1D algorithm consis-  
 282 tency (Theorem 1) and also because that the ECD algorithm is guaranteed to solve  $\phi_k(\mathbf{w})$  from  
 283 steps 6–7 of Algorithm 3.

284 **Corollary 2.** *Suppose  $\mathbf{M} > 0$ ,  $\mathbf{U} \geq 0$  and  $\widehat{\mathbf{M}}$  and  $\widehat{\mathbf{U}}$  are  $\sqrt{n}$ -consistent sample estimators  
 285 for  $\mathbf{M}$  and  $\mathbf{U}$ . Let  $\widehat{\mathbf{G}}_u$  denote the estimator obtained from the ECD algorithm using  $\widehat{\mathbf{M}}$  and  $\widehat{\mathbf{U}}$   
 286 where  $u$  is the dimension of the envelope. Then  $\mathbf{P}_{\widehat{\mathbf{G}}_u}$  is  $\sqrt{n}$ -consistent for the projection onto*



287  $\mathcal{E}_M(\mathbf{U})$ .

## 288 **4 Numerical Studies**

289 In this section, we compare the 1D algorithm to our proposed algorithms. In the simulated  
290 data studies of Section 4.1, because the true envelope structure is known, we find that there  
291 is no significant difference among methods in terms of accuracy in estimating envelopes and  
292 thus we compare the algorithms in terms of their computation time. The shared estimation  
293 accuracy is summarized in table legends. In the data analysis of Section 4.2, the true envelope  
294 structure is unknown and we compare the methods in terms of cross-validation prediction mean  
295 squared errors (PMSE) and also computation time. The computation was done on a Windows  
296 7 computer with Intel(R) Core(TM) i5-5300U CPU@2.30GHz processor, 8.00 GB installed  
297 memory (RAM), 64-bit Operating System.

298 The coordinate descent algorithm can be more efficient when the objective function is sep-  
299 arable in coordinates. Our ECD algorithm thus takes advantage of the canonical coordinates.  
300 However, transformation of the coordinate system has little effect on the 1D algorithm solved  
301 by any standard nonlinear optimization methods (such as PRCG).

		ECD	1D	ECS ( $d = u$ )
(I)	$p = 20$	$3.8 (0.4) \times 10^{-2}$	$7.2 (0.3)$	$2.6 (0.3) \times 10^{-2}$
	$p = 50$	$2.0 (0.1) \times 10^{-1}$	$2.6 (0.1) \times 10$	$1.5 (0.1) \times 10^{-1}$
	$p = 200$	$9.1 (0.1)$	$1.7 (0.04) \times 10^2$	$1.5 (0.01) \times 10$
(II)	$p = 20$	$3.4 (0.4) \times 10^{-2}$	$4.2 (0.1) \times 10$	$1.0 (0.3) \times 10^{-2}$
	$p = 50$	$1.9 (0.1) \times 10^{-1}$	$1.4 (0.01) \times 10^2$	$6.8 (0.5) \times 10^{-2}$
	$p = 200$	$8.2 (0.06)$	$7.0 (0.01) \times 10^2$	$3.5 (0.02)$
(III)	$p = 20$	$4.4 (0.7) \times 10^{-2}$	$3.4 (0.1) \times 10$	$1.6 (0.6) \times 10^{-2}$
	$p = 50$	$2.4 (0.1) \times 10^{-1}$	$4.9 (0.1) \times 10$	$8.2 (0.7) \times 10^{-2}$
	$p = 200$	$8.1 (0.1)$	$7.2 (0.04) \times 10^2$	$3.8 (0.04)$

Table 1: Computing time in seconds for each methods with simulated matrices  $\mathbf{M}$  and  $\mathbf{U}$ . Each cell of the table was averaged over 20 runs with standard error in parentheses. The estimation accuracy is  $\|\mathbf{P}_{\Gamma} - \mathbf{P}_{\hat{\Gamma}}\|_F < 10^{-6}$  for every methods at each of these settings and is thus not reported the table.

Models	(I)	(II)	(III)
ECD	0.20 (0.01)	0.09 (0.01)	0.145 (0.01)
1D	4.28 (0.07)	2.35 (0.04)	1.65 (0.02)

Table 2: Computing time in seconds using simulated matrices  $\widehat{\mathbf{M}}$  and  $\widehat{\mathbf{U}}$  with  $p = 20$  and  $n = 100$ . Each cell of the table was averaged over 100 runs with standard error in parentheses. The estimation accuracies  $\|\mathbf{P}_{\Gamma} - \mathbf{P}_{\hat{\Gamma}}\|_F$  for the three models are 0.42, 1.20 and 0.14, respectively, and there was no significant difference between any two methods at any of the three settings. Therefore, estimation accuracy is not reported the table.

Models	Time				ECS selected $d$
	ECD	1D	ECS-ECD	ECS-1D	
(I)	0.56 (0.02)	12.19 (0.08)	0.62 (0.02)	11.94 (0.08)	47.0 (0.1)
(II)	0.45 (0.01)	9.46 (0.10)	0.42 (0.01)	8.78 (0.09)	39.9 (0.2)
(III)	0.74 (0.02)	6.59 (0.05)	0.14 (0.01)	0.14 (0.01)	5 (0)

Table 3: Computing time in seconds using simulated matrices  $\widehat{\mathbf{M}}$  and  $\widehat{\mathbf{U}}$  with  $p = 50$  and  $n = 100$ . Each cell of the table was averaged over 100 runs with standard error in parentheses. The estimation accuracies  $\|\mathbf{P}_{\mathbf{r}} - \mathbf{P}_{\widehat{\mathbf{r}}}\|_F$  for the three models are 0.98, 1.94 and 0.29, respectively, and there was no significant difference between any two methods at any of the three settings. Therefore, estimation accuracy is not reported the table.

Models	Time					ECS selected $d$
	ECD	1D	ECS-ECD	ECS-1D	ECS $_n$	
(I)	0.92 (0.01)	5.64 (0.05)	1.15 (0.01)	5.12 (0.06)	9.20 (0.03)	86.9 (0.2)
(II)	0.86 (0.01)	4.62 (0.07)	0.54 (0.01)	1.39 (0.03)	9.45 (0.03)	40.8 (0.4)
(III)	NA	NA	0.72 (0.01)	62.13 (0.76)	9.24 (0.04)	0 (0)

Table 4: Computing time in seconds using simulated matrices  $\widehat{\mathbf{M}}$  and  $\widehat{\mathbf{U}}$  with  $p = 2000$  and  $n = 100$ . Each cell of the table was averaged over 100 runs with standard error in parentheses. The ECS $_n$  is the pre-process step of applying the ECS algorithm to reduce the dimension from  $p = 2000$  to  $d = n = 100$ . Then we recorded the computing time of the four methods (ECD, 1D, ECS-ECD and ECS-1D) applied on the reduced data. The estimation accuracies  $\|\mathbf{P}_{\mathbf{r}} - \mathbf{P}_{\widehat{\mathbf{r}}}\|_F$  for the three models are 1.31, 1.45, 3.16, respectively, and there was no significant difference between any two methods at any of the three settings. Therefore, estimation accuracy is not reported the table.

## 302 4.1 Simulated data

303 In this section, we consider the problem of estimating a generic envelope  $\mathcal{E}_{\mathbf{M}}(\mathbf{U})$ , where matri-  
 304 ces were generated as

$$\begin{aligned} \mathbf{M} &= \begin{cases} \mathbf{\Gamma}\mathbf{\Omega}\mathbf{\Gamma}^T + \mathbf{\Gamma}_0\mathbf{\Omega}_0\mathbf{\Gamma}_0^T, & \text{Model I,} \\ \mathbf{\Gamma}\mathbf{\Gamma}^T + 0.01\mathbf{\Gamma}_0\mathbf{\Gamma}_0^T, & \text{Model II,} \\ 0.01\mathbf{\Gamma}\mathbf{\Gamma}^T + \mathbf{\Gamma}_0\mathbf{\Gamma}_0^T, & \text{Model III,} \end{cases} \\ \mathbf{U} &= \mathbf{\Gamma}\mathbf{\Phi}\mathbf{\Gamma}^T, \quad \text{for all models,} \end{aligned}$$

305 where  $\mathbf{\Gamma} \in \mathbb{R}^{p \times u}$  was randomly generated by first filling in with random numbers from the  
 306 Uniform  $(0, 1)$  distribution and then transforming so that  $\mathbf{\Gamma}$  is semi-orthogonal,  $\mathbf{\Gamma}_0 \in \mathbb{R}^{p \times (p-u)}$   
 307 was the completion of  $\mathbf{\Gamma}$  such that  $(\mathbf{\Gamma}, \mathbf{\Gamma}_0)$  was orthogonal,  $\mathbf{\Omega}$  was generated as  $\mathbf{A}\mathbf{A}^T \geq 0$ , where  
 308  $\mathbf{A}$  had the same size of  $\mathbf{\Omega}$  and was filled in with random numbers from Unifrom  $(0, 1)$ ,  $\mathbf{\Omega}_0$  and  
 309  $\mathbf{\Phi}$  were both generated in the same way as  $\mathbf{\Omega}$  with  $\mathbf{A}$  matching the dimensions of  $\mathbf{\Omega}_0$  and  $\mathbf{\Phi}$ .  
 310 Finally, to guarantee  $\mathbf{M} > 0$  in Model I, we added  $0.00001\mathbf{I}_p$  to  $\mathbf{M}$  after it was simulated.

311 The first set of simulations compares the methods primarily on the time it takes to recover  
 312 the envelope in the population, using the true values for  $\mathbf{M}$  and  $\mathbf{U}$  in the objective function  $F$ .  
 313 For each of the three models, we fixed  $u = 5$  and generated 20 pairs of  $\mathbf{M}$  and  $\mathbf{U}$  for each  
 314 of the three dimensions,  $p = 20, 50$ , and  $200$ . Three methods are to be compared here: ECD  
 315 algorithm; 1D algorithm; ECS algorithm with  $d = u$  components selected. The ECS method  
 316 worked as a stand-alone method because  $\mathbf{M}$  and  $\mathbf{U}$  were population quantities. We recorded  
 317 the estimation error, the Frobenius norm  $\|\mathbf{P}_{\mathbf{\Gamma}} - \mathbf{P}_{\hat{\mathbf{\Gamma}}}\|_F$ , and also the computing time for each  
 318 run. The results were summarized in Table 1. All three methods had the same accuracy in these

319 settings, since we used appropriate tolerance and maximum iteration numbers, the estimation  
320 errors were simply due to rounding errors in the program. In terms of computation time, ECS  
321 and ECD were equally fast, and about a hundred times faster than the 1D algorithm.

322 In the next set of simulations we applied the algorithms to estimates  $\widehat{\mathbf{M}} \sim W_p(\mathbf{M}/n, n)$  and  
323  $\widehat{\mathbf{U}} \sim W_p(\mathbf{U}/n, n)$  instead of their population counterparts  $\mathbf{M}$  and  $\mathbf{U}$ . The Wishart distribution  
324 mimics the linear regression model settings. We chose  $n = 100$  and varied  $p$  as 20, 50, and  
325 2000 to mimic the small ( $p < n$ ), moderate ( $p \lesssim n$ ) and high ( $p \gg n$ ) dimensional situations.

326 For  $p = 20$ , the ECS algorithm was not needed as both the ECD and 1D algorithms are fast  
327 and accurate for relatively small  $p$ . The direct comparison of the ECD algorithm and the 1D  
328 algorithm is summarized in Table 2 where ECD was at least ten times faster.

329 For  $p = 50$ , the ECD and 1D algorithms are still applicable and the ECS algorithm can also  
330 be used as a preparation step for both 1D and ECD algorithms. We chose  $d$  based on the cut-off  
331 value  $C_0 = -n^{-1}$  as discussed in Section 3.1. The results are summarized in Table 3. Again,  
332 the ECD algorithm improved over the 1D algorithm, with and without the preparation step by  
333 ECS algorithm. For Models (I) and (II), the ECS algorithm only eliminated a few components  
334 so that the results did not change much with the ECS algorithm. For Model (III), the ECS  
335 algorithm selected  $d$  equal to the envelope dimension  $u$  every time, implying a clear envelope  
336 structure from the data and thus estimating it as accurate as the 1D or ECD algorithms. The  
337 results were summarized in Table 3.

338 For  $p = 2000$ , the ECD and 1D algorithms are no longer straightforwardly applicable. We  
339 used the ECS algorithm to first reduce the dimension from  $p = 2000$  to  $n = 100$  and then

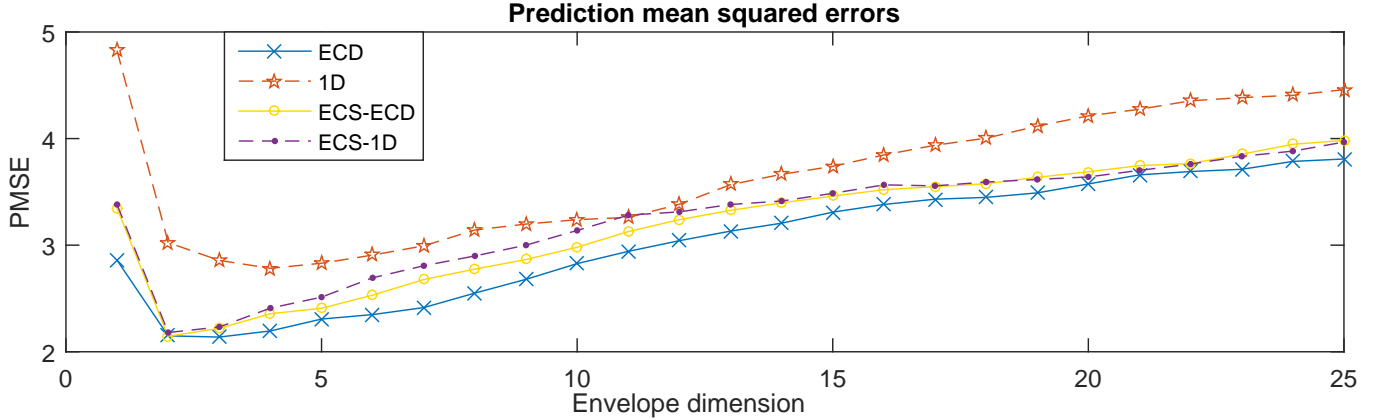


Figure 4.1: Meat Protein Data: prediction mean squares error comparison.

340 applied the ECD and 1D algorithms on the reduced data. We also applied the ECS-ECD and  
 341 ECS-1D on the reduced data with  $d$  selected from the data. Because the ECS step of reducing  
 342 the dimension from 2000 to 100 was the more costly step, we extracted the computing time of  
 343 this step as  $\text{ECS}_n$  in Table 4. The estimation accuracy  $\|\mathbf{P}_\Gamma - \mathbf{P}_{\hat{\Gamma}}\|_F$  for Model (III) was 3.16 for  
 344 all methods because the immaterial part  $\Gamma_0\Gamma_0^T$  dominated the material part  $0.01\Gamma\Gamma^T$  in  $\mathbf{M}$  and  
 345 there was no estimable information from the data – the sample version  $\widehat{\mathbf{M}}$  lay mostly within  
 346  $\text{span}(\Gamma_0)$  as  $n < p$ . Therefore, the ECS algorithm also suggested  $d = 0$  for this situation.

## 347 4.2 Data Analysis

348 We revisited the meat protein data set from Cook et al. (2013) and Cook and Zhang (2016).  
 349 Following these previous studies, we used the protein percentage of  $n = 103$  meat samples as  
 350 the univariate response variable  $Y_i \in \mathbb{R}^1, i = 1, \dots, n$ , and used the corresponding  $p = 50$   
 351 spectral measurements from near-infrared transmittance at every fourth wavelength between  
 352 850nm and 1050nm as the predictor  $\mathbf{X}_i \in \mathbb{R}^p$ . The linear regression model was built as  $Y_i =$

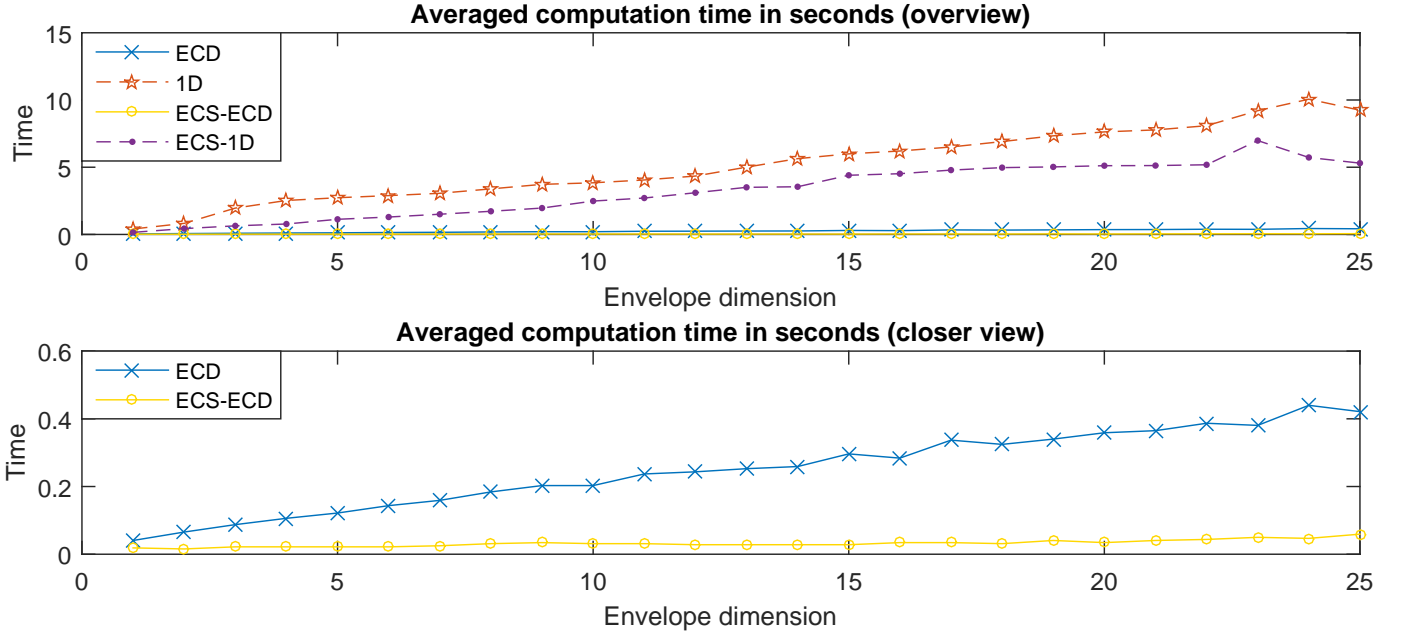


Figure 4.2: Meat Protein Data: computing time comparison.

353  $\alpha + \beta \mathbf{X}_i + \epsilon_i$  with the envelope  $\mathcal{E}_{\Sigma_{\mathbf{X}}}(\beta^T)$  in the predictor space, Cook et al. (2013). If  $\mathbf{M} =$   
354  $\Sigma_{\mathbf{X}|\mathbf{Y}} = \Sigma_{\mathbf{X}} - \Sigma_{\mathbf{X}\mathbf{Y}}\Sigma_{\mathbf{Y}}^{-1}\Sigma_{\mathbf{X}\mathbf{Y}}^T > 0$  and  $\mathbf{M} + \mathbf{U} = \Sigma_{\mathbf{X}}$ , then we can obtain the normal likelihood-  
355 based objective function by substituting the corresponding sample covariance matrices  $\widehat{\mathbf{M}}$  and  
356  $\widehat{\mathbf{M}} + \widehat{\mathbf{U}}$  into (2.1). Given the envelope dimension  $u$ , we used  $\widehat{\mathbf{M}}$  and  $\widehat{\mathbf{U}}$  with various algorithms  
357 to get estimators of an envelope basis, denoted as  $\widehat{\Gamma}$ . Then the envelope estimator for the  
358 regression coefficient matrix was written as  $\widehat{\beta}^T = \widehat{\Gamma}(\widehat{\Gamma}^T \widehat{\Sigma}_{\mathbf{X}} \widehat{\Gamma})^{-1} \widehat{\Gamma}^T \widehat{\Sigma}_{\mathbf{X}\mathbf{Y}}$  and the response was  
359 predicted as  $\widehat{Y}^* = \widehat{\mu}_{\mathbf{Y}} + \widehat{\beta}(\mathbf{X}^* - \widehat{\mu}_{\mathbf{X}})$ , where  $\widehat{\mu}_{\mathbf{Y}}$  and  $\widehat{\mu}_{\mathbf{X}}$  are the sample means from observed  
360 data (or from the training data set) and  $\mathbf{X}^*$  denotes new independently observed data. Varying  
361 envelope dimension  $u$  from 1 to 25 and using five-fold cross-validation prediction mean squared  
362 error and computation time as two criteria, Cook and Zhang (2016) compared the 1D envelope  
363 estimator based on Algorithm 1 with OLS and envelope estimator from full Grassmannian

364 (FG) optimization. Their results showed both envelope estimators to be uniformly superior to  
365 OLS and that the 1D envelope estimator was superior to the FG envelope estimator on the two  
366 criteria: the computation time for the 1D estimator was 10 to 100 times faster than the FG  
367 estimator and the prediction error of the 1D estimator was always less than or equal to that of  
368 the FG estimator for all the values of  $u$  from 1 to 25. We next compare the proposed algorithms  
369 only to the “best available” method: the 1D algorithm.

370 We randomly split the data into a testing sample and a training sample in a 1:4 ratio and  
371 recorded the prediction mean squared errors (PMSE) and the computation time for fitting each  
372 envelope basis at each of the 25 envelope dimensions. This procedure was then repeated 100  
373 times and the results averaged. Similar to the simulation studies in Table 3, we compared the  
374 four envelope estimators: ECD, 1D, ECS-ECD, and ECS-1D. For the ECS-ECD and ECS-1D  
375 estimators we used the ECS algorithm to screen the 50 components down to the data-driven  $d$ ,  
376 which was 34.2 on average with 0.2 standard error.

377 Figure 4.1 summarizes the PMSE comparison. The ECD algorithm was again proven to  
378 be the most reliable one. The differences between the 1D and the ECD estimators were due  
379 to the convergence of algorithms on some of the 100 training data sets. The ECD algorithm  
380 is less sensitive to peculiar local optima, while the 1D algorithm seems often trapped in those  
381 local optima. In this data set, there appears to be many local optima mainly due to two reasons:  
382 the number of predictors  $p = 50$  is close to the training set sample size 83; the correlation  
383 among the predictors is very high. From the absolute values of the  $p \times (p - 1)/2 = 1,225$   
384 pairwise sample correlations, we find 53 of them are bigger than 0.99 where the largest one is



385 0.9999. Comparing ECS-ECD to ECD, it is clear that the ECS algorithm sacrificed accuracy  
386 for computational efficiency and fewer components in the model. However, because of fewer  
387 components, the ECS-1D algorithm actually improved over the 1D algorithm. For  $u = 2$ , we  
388 summarize all the PMSE on 100 testing sets using a side-by-side boxplot in the Supplementary  
389 Materials, where the 1D algorithm is clearly outperformed by our proposed estimators using  
390 either means or quantiles of the 100 PMSE as criteria.

391 Figure 4.2 summarizes the computing time comparison. The ECD algorithm was at least  
392 10 times faster than the 1D algorithm across all the envelope dimensions. The ECS algorithm  
393 improved the 1D algorithm by roughly doubling its speed, and it improved the ECD algorithm  
394 speed even more drastically, sometimes more than 10 times faster. This can be explained by  
395 the fact that both the ECD and the ECS algorithms work on the same envelope components or  
396 coordinates, which were the principal components of the 50 predictors in this application, and  
397 that variables in this data set are highly correlated leads to an even faster convergence of the  
398 ECS-ECD algorithm.

399 If we consider choosing the envelope dimension from 1 to 25 using 5-fold cross-validation,  
400 then we need  $25 \times 5 = 125$  individual envelope model fits. The 1D algorithm took a total of  
401 about 11.5 minutes to finish the optimization, while the faster ECD algorithm needs only 0.5  
402 minutes to reach the same conclusion. If we choose the ECS-ECD approach, it is even faster,  
403 with just 0.067 minutes for all the envelope estimations. While these differences might not  
404 seem very large, applied work may often require much more computation. We may wish to  
405 use averages over multiple five-fold cross validations to gain a more reliable picture of relative

406 prediction errors, we might use the bootstrap to estimate standard errors or for estimators based  
407 on bootstrap smoothing, or we might wish to carry out computations for all possible envelope  
408 dimensions. Iterating over alternating envelope fits might be required in some problems, as in  
409 envelopes for simultaneous response and predictor reduction, Cook and Zhang (2015b). For  
410 instance, if we decided for the meat analysis to use averages over 10 five-fold cross validations,  
411 250 bootstrap samples and all envelope dimensions, the computation time could range from  
412 about 80 days for the 1D algorithm to a half day for the ECS-ECD algorithm.

## 413 **5 Discussion**

414 In this paper, we proposed two computational tools to speed up the non-convex Grassmannian  
415 optimization that appears in the estimation of almost all envelope models, for example, Cook  
416 et al. (2010); Su and Cook (2011); Cook et al. (2013); Cook and Zhang (2015a); Li and Zhang  
417 (2016); Zhang and Li (2016). The ECD and the ECS algorithms were developed based on the  
418 idea that the iterative non-convex optimization steps in envelope estimation could be replaced  
419 by crude or approximated solutions after transforming the coordinates. These algorithms can  
420 also be applied to estimate a general envelope provided the objective function  $F$  is reasonable.  
421 The general approach may also be adapted to Grassmannian optimizations that arise in other  
422 multivariate statistical context like likelihood acquired directions, Cook and Forzani (2009).

## 423 **Supplementary Materials**

424 The online Supplementary Materials (PDF) contain technical details and some additional nu-  
425 merical results.

## 426 **Acknowledgments**

427 The authors are grateful to the Editor, an associate editor and three referees for constructive and  
428 insightful comments that led to significant improvements in this article. Xin Zhang’s research  
429 for this article was supported in part by grants DMS-1613154 and CCF-1617691 from the  
430 National Science Foundation.

## 431 **References**

- 432 Bair, E., Hastie, T., Paul, D., and Tibshirani, R. (2006). Prediction by supervised principal  
433 components. *Journal of the American Statistical Association*, 101(473).
- 434 Conway, J. (1990). *A Course in Functional Analysis. Second edition.* Springer, New York.
- 435 Cook, R. D. and Forzani, L. (2009). Likelihood-based sufficient dimension reduction. *Journal*  
436 *of the American Statistical Association*, 104(485):197–208.
- 437 Cook, R. D., Forzani, L., and Zhang, X. (2015a). Envelopes and reduced-rank regression.  
438 *Biometrika*, 102(2):439–456.
- 439 Cook, R. D., Helland, I. S., and Su, Z. (2013). Envelopes and partial least squares regression.  
440 *J. R. Stat. Soc. Ser. B. Stat. Methodol.*, 75(5):851–877.
- 441 Cook, R. D., Li, B., and Chiaromonte, F. (2010). Envelope models for parsimonious and  
442 efficient multivariate linear regression. *Statist. Sinica*, 20(3):927–960.
- 443 Cook, R. D., Su, Z., and Yang, Y. (2015b). envlp: A matlab toolbox for computing envelope  
444 estimators in multivariate analysis. *Journal of Statistical Software*, 62(8).
- 445 Cook, R. D. and Zhang, X. (2015a). Foundations for envelope models and methods. *Journal*  
446 *of the American Statistical Association*, 110(510):599–611.

- 447 Cook, R. D. and Zhang, X. (2015b). Simultaneous envelopes for multivariate linear regression.  
448 *Technometrics*, 57(1):11–25.
- 449 Cook, R. D. and Zhang, X. (2016). Algorithms for envelope estimation. *Journal of Computa-*  
450 *tional and Graphical Statistics*, 25(1):284–300.
- 451 Li, G., Shen, H., and Huang, J. Z. (2015a). Supervised sparse and functional principal compo-  
452 nent analysis. *Journal of Computational and Graphical Statistics*, (just-accepted):00.
- 453 Li, G., Yang, D., Nobel, A. B., and Shen, H. (2015b). Supervised singular value decomposition  
454 and its asymptotic properties. *Journal of Multivariate Analysis*.
- 455 Li, L. and Zhang, X. (2016). Parsimonious tensor response regression. *Journal of the American*  
456 *Statistical Association*, (just-accepted).
- 457 Su, Z. and Cook, R. D. (2011). Partial envelopes for efficient estimation in multivariate linear  
458 regression. *Biometrika*, 98(1):133–146.
- 459 Zhang, X. and Li, L. (2016). Tensor envelope partial least squares regression. *Technometrics*,  
460 (just-accepted).