Journal of Statistical Software

MMMMMM YYYY, Volume VV, Issue II. *http://www.jstatsoft.org/*

envlp: A MATLAB Toolbox for Computing Envelope Estimators in Multivariate Analysis

Dennis Cook
University of Minnesota

Zhihua Su
University of Florida

Yi Yang
University of Minnesota

Abstract

Envelope models and methods represent new constructions that can lead to substantial increases in estimation efficiency in multivariate analyses. The **envlp** toolbox implements a variety of envelope estimators under the framework of multivariate linear regression, including the envelope model, partial envelope model, heteroscedastic envelope model, inner envelope model, scaled envelope model, and envelope model in the predictor space. The toolbox also implements the envelope model for estimating a multivariate mean. The capabilities of this toolbox include estimation of the model parameters, as well as performing standard multivariate inference in the context of envelope models; for example, prediction and prediction errors, F test for two nested models, the standard errors for contrasts or linear combinations of coefficients, and more. Examples and datasets are contained in the toolbox to illustrate the use of each model. All functions and datasets are documented.

Keywords: multivariate linear regression, envelope models, dimension reduction, Grassmann manifold, MATLAB.

1. Introduction

The envelope model is a new construction originally introduced by Cook, Li, and Chiaromonte (2010) in the context of multivariate linear regression

$$\mathbf{Y} = \boldsymbol{\alpha} + \boldsymbol{\beta}\mathbf{X} + \boldsymbol{\varepsilon}, \tag{1}$$

where $\mathbf{Y} \in \mathbb{R}^r$ is the multivariate response vector, $\mathbf{X} \in \mathbb{R}^p$ is the non-stochastic predictor vector centered at 0 in the sample, the error vectors $\boldsymbol{\varepsilon} \in \mathbb{R}^r$ are identically and independently distributed across observations with mean 0 and positive definite covariance matrix $\boldsymbol{\Sigma} \in \mathbb{R}^{r\times r}$, and $\boldsymbol{\alpha} \in \mathbb{R}^r$ is the unknown intercept. The key parameters are the elements of the coefficient

matrix $\boldsymbol{\beta} \in \mathbb{R}^{r\times p}$. Compared to the standard ordinary least squares estimator $\widehat{\boldsymbol{\beta}}_{ols}$, the envelope estimator $\widehat{\boldsymbol{\beta}}_{em}$ is potentially less variable and thus more efficient. This is achieved by allowing for the possibility that the distribution of some linear combinations of $\mathbf{Y}$ is invariant to changes in $\mathbf{X}$, and we call this the immaterial part of $\mathbf{Y}$. The immaterial part of $\mathbf{Y}$ provides no worthwhile information on $\boldsymbol{\beta}$, and yet it increases the variation in $\widehat{\boldsymbol{\beta}}_{ols}$. The envelope model identifies and accounts for the immaterial information and therefore reduces the variation in estimation. This reduction can be substantial, especially when the immaterial part of $\mathbf{Y}$ introduces large variation.

Several extensions have been developed following Cook *et al.* (2010). The partial envelope model (Su and Cook 2011) focuses on the estimation of the coefficients for a selected subset of the predictors, and is therefore more efficient in estimating those coefficients. The inner envelope model (Su and Cook 2012) applies the enveloping idea in a novel way, which results in new methodology that is able to gain efficiency even when there is no immaterial information in the data. The heteroscedastic envelope model (Su and Cook 2013) removes the constant variance assumption in the envelope model, making it more flexible and more widely applicable. The scaled envelope model (Cook and Su 2013) is a scale invariant version of the envelope model, which can offer efficiency gains beyond those from the envelope model itself. The envelope model in the predictor space (Cook, Helland, and Su 2013) focuses on dimension reduction for the predictors. It is equivalent to the partial least squares (PLS) in the population and yet performs better than PLS with finite samples. The envelope model that estimates a multivariate mean can be viewed as an alternative to Stein estimation. Like the other methods it is particularly effective and can perform better than Stein estimation when there is immaterial information present in the data.

The only software that now performs envelope estimation is MATLAB (The MathWorks, Inc. 2012b) package **LDR** (Cook, Forzani, and Tomassi 2009). This package is mainly focused on likelihood-based sufficient dimension reduction, not envelope estimation. It implements the basic envelope model in Cook *et al.* (2010), but not any of its extension or any inference methods in the envelope model context. This article describes the toolbox **envlp**, which implements all the existing envelope methods. It also contains functions for dimension selection, bootstrap estimation, prediction and hypothesis testing. Examples are provided to illustrate the use of the toolbox. All the documentation, as well as updates can be checked at the website `http://code.google.com/p/envlp/`.

The rest of this paper is organized as follows. The envelope models are discussed in Section 2. Section 3 is an overview of the toolbox. Section 4 provides some examples on using the package. Discussion on future developments is in Section 5.

2. Envelope models

2.1. The basic envelope model

Let $(\boldsymbol{\Gamma}, \boldsymbol{\Gamma}_0) \in \mathbb{R}^{r\times r}$ be an orthogonal matrix. If

$$\boldsymbol{\Gamma}_0^\top \mathbf{Y} \mid \mathbf{X} \sim \boldsymbol{\Gamma}_0^\top \mathbf{Y}, \text{ and } \boldsymbol{\Gamma}^\top \mathbf{Y} \perp\!\!\!\perp \boldsymbol{\Gamma}_0^\top \mathbf{Y} \mid \mathbf{X},$$

then $\boldsymbol{\Gamma}_0^\top \mathbf{Y}$ carries no information on $\boldsymbol{\beta}$ and it represents the immaterial part of $\mathbf{Y}$, while $\boldsymbol{\Gamma}^\top \mathbf{Y}$ is the material part. Let $\mathcal{B} = \text{span}(\boldsymbol{\beta})$. Cook *et al.* (2010) showed that the previous two

conditions are equivalent to the following two conditions

$$\text{(2a) } \mathcal{B} \subseteq \text{span}(\boldsymbol{\Gamma}), \quad \text{(2b) } \boldsymbol{\Sigma} = \mathbf{P}_{\boldsymbol{\Gamma}}\boldsymbol{\Sigma}\mathbf{P}_{\boldsymbol{\Gamma}} + \mathbf{Q}_{\boldsymbol{\Gamma}}\boldsymbol{\Sigma}\mathbf{Q}_{\boldsymbol{\Gamma}}, \tag{2}$$

where $\mathbf{P}_{(\cdot)}$ is a projection matrix onto the subspace indicated by its argument and $\mathbf{Q}_{(\cdot)} = \mathbf{I} - \mathbf{P}_{(\cdot)}$. If we have (2b), span($\boldsymbol{\Gamma}$) is called a reducing subspace of $\boldsymbol{\Sigma}$ (Conway 1990). An envelope subspace is defined as the smallest reducing subspace of $\boldsymbol{\Sigma}$ containing $\mathcal{B}$ (Cook *et al.* 2010), and is denoted by $\mathcal{E}_{\boldsymbol{\Sigma}}(\mathcal{B})$. In the context of (1), let $\boldsymbol{\Gamma} \in \mathbb{R}^{r\times u}$ span the envelope subspace $\mathcal{E}_{\boldsymbol{\Sigma}}(\mathcal{B})$. The envelope model is then written as follows

$$\mathbf{Y} = \boldsymbol{\alpha} + \boldsymbol{\Gamma}\boldsymbol{\eta}\mathbf{X} + \boldsymbol{\varepsilon}, \quad \boldsymbol{\Sigma} = \boldsymbol{\Sigma}_1 + \boldsymbol{\Sigma}_2 = \boldsymbol{\Gamma}\boldsymbol{\Omega}\boldsymbol{\Gamma}^\top + \boldsymbol{\Gamma}_0\boldsymbol{\Omega}_0\boldsymbol{\Gamma}_0^\top,$$

where $\boldsymbol{\beta} = \boldsymbol{\Gamma}\boldsymbol{\eta}$, $\boldsymbol{\eta} \in \mathbb{R}^{u\times p}$, $\boldsymbol{\Omega} \in \mathbb{R}^{u\times u}$ and $\boldsymbol{\Omega}_0 \in \mathbb{R}^{(r-u)\times(r-u)}$ are unknown positive definite matrices, u is the dimension of the envelope subspace. From this model, the two conditions in (2) are satisfied: $\mathcal{B}$ is contained in span($\boldsymbol{\Gamma}$), and $\boldsymbol{\Sigma}$ is the sum of $\boldsymbol{\Sigma}_1 = \mathsf{VAR}(\mathbf{P}_{\boldsymbol{\Gamma}}\mathbf{Y})$, the variance related to the material part of $\mathbf{Y}$, and $\boldsymbol{\Sigma}_2 = \mathsf{VAR}(\mathbf{Q}_{\boldsymbol{\Gamma}}\mathbf{Y})$, the variance related to the immaterial part. It is seen from the envelope model that $\boldsymbol{\beta}$ and $\boldsymbol{\Sigma}$ are linked by $\boldsymbol{\Gamma}$ and it is this link that results in more efficient estimation of $\boldsymbol{\beta}$. In effect, the estimation process accounts for the variation in the immaterial information $\boldsymbol{\Gamma}_0^\top\mathbf{Y}$. Let $\|\cdot\|$ denote the spectral norm of a matrix. When $\|\boldsymbol{\Sigma}_1\| \ll \|\boldsymbol{\Sigma}_2\|$, the immaterial part has relatively large variation and the envelope model will offer substantial efficiency gains over the standard model (1).

When $u = r$, there is no immaterial information in $\mathbf{Y}$, and the envelope model is equivalent to the standard model (1). This will happen when the rank of $\boldsymbol{\beta}$ is equal to r.

2.2. Partial envelope model

The partial envelope model (Su and Cook 2011) is appropriate when part of the predictors are of special interest. It is often more efficient than the envelope model for the purpose of estimating the regression coefficients for those predictors.

Suppose we can partition $\mathbf{X}$ to $(\mathbf{X}_1^\top, \mathbf{X}_2^\top)^\top$, where $\mathbf{X}_1 \in \mathbb{R}^{p_1}$ are the predictors of special interest and $\mathbf{X}_2 \in \mathbb{R}^{p_2}$ are covariates, $p_1 + p_2 = p$. Then $\boldsymbol{\beta}$ can be partitioned accordingly into $(\boldsymbol{\beta}_1, \boldsymbol{\beta}_2)$, and (1) can be written as

$$\mathbf{Y} = \boldsymbol{\alpha} + \boldsymbol{\beta}_1\mathbf{X}_1 + \boldsymbol{\beta}_2\mathbf{X}_2 + \boldsymbol{\varepsilon},$$

where $\boldsymbol{\beta}_1 \in \mathbb{R}^{r\times p_1}$ is the key parameter.

The partial envelope model applies the enveloping idea on $\boldsymbol{\beta}_1$: Let $\mathcal{B}_1 = \text{span}(\boldsymbol{\beta}_1)$. A partial $\boldsymbol{\Sigma}$-envelope of $\mathcal{B}_1$, denoted by $\mathcal{E}_{\boldsymbol{\Sigma}}(\mathcal{B}_1)$, is the smallest reducing subspace of $\boldsymbol{\Sigma}$ containing $\mathcal{B}_1$. The coordinate form of the partial envelope model is

$$\mathbf{Y} = \boldsymbol{\alpha} + \boldsymbol{\Gamma}\boldsymbol{\eta}\mathbf{X}_1 + \boldsymbol{\beta}_2\mathbf{X}_2 + \boldsymbol{\varepsilon}, \quad \boldsymbol{\Sigma} = \boldsymbol{\Sigma}_1 + \boldsymbol{\Sigma}_2 = \boldsymbol{\Gamma}\boldsymbol{\Omega}\boldsymbol{\Gamma}^\top + \boldsymbol{\Gamma}_0\boldsymbol{\Omega}_0\boldsymbol{\Gamma}_0^\top,$$

where $\boldsymbol{\Gamma} \in \mathbb{R}^{r\times u_1}$ spans $\mathcal{E}_{\boldsymbol{\Sigma}}(\mathcal{B}_1)$, $\boldsymbol{\Gamma}_0$ spans $\mathcal{E}_{\boldsymbol{\Sigma}}^{\perp}(\mathcal{B}_1)$, the subspace orthogonal to $\mathcal{E}_{\boldsymbol{\Sigma}}(\mathcal{B}_1)$, u_1 is the dimension of $\mathcal{E}_{\boldsymbol{\Sigma}}(\mathcal{B}_1)$, $\boldsymbol{\eta} = \boldsymbol{\Gamma}^\top\boldsymbol{\beta}_1 \in \mathbb{R}^{u_1\times p}$, $\boldsymbol{\Omega} \in \mathbb{R}^{u_1\times u_1}$ and $\boldsymbol{\Omega}_0 \in \mathbb{R}^{(r-u_1)\times(r-u_1)}$ are both positive definite matrices. Compared to the envelope model, $\mathcal{E}_{\boldsymbol{\Sigma}}(\mathcal{B}_1) \subseteq \mathcal{E}_{\boldsymbol{\Sigma}}(\mathcal{B})$ and $u_1 \leq u$. Intuitively, more information is immaterial relative to $\boldsymbol{\beta}_1$, so the partial envelope model is typically more efficient than the envelope model for the purpose of estimating $\boldsymbol{\beta}_1$.

The partial envelope model degenerates to the standard model when $u_1 = r$, which means no information is immaterial to $\boldsymbol{\beta}_1$. This happens when the rank of $\boldsymbol{\beta}_1$ is equal to r. So

in a regression problem where $\text{rank}(\boldsymbol{\beta}) = r$, the envelope model degenerates to the standard model; while as long as $p_1 < r$, the partial envelope model is still applicable. In this sense, the partial envelope model is more flexible than the envelope model.

2.3. Heteroscedastic envelope model

The envelope model in Section 2.1 assumes homogeneity of the error variance. The heteroscedastic envelope model (Su and Cook 2013) removes this assumption and allows for non-constant covariance structure. The heteroscedastic envelope model was developed in the context of estimating multivariate means for different populations. This problem can be formulated as

$$\mathbf{Y}_{(i)j} = \boldsymbol{\mu} + \boldsymbol{\beta}_{(i)} + \boldsymbol{\varepsilon}_{(i)j},\ i = 1, \cdots, p,\ j = 1, \cdots, n_{(i)}, \tag{3}$$

where the subscripts with parentheses denote groups and subscripts without parentheses denote observations within a group, $\mathbf{Y}_{(i)j} \in \mathbb{R}^r$ is the jth observation in the ith group, $\boldsymbol{\mu} \in \mathbb{R}^r$ is the grand mean over all the observations, $\boldsymbol{\beta}_{(i)} \in \mathbb{R}^r$ is the main effect of the ith group and we assume that $\sum_{i=1}^{p} n_{(i)}\boldsymbol{\beta}_{(i)} = 0$, $n_{(i)}$ is the sample size for the ith group, $\boldsymbol{\varepsilon}_{(i)j} \in \mathbb{R}^r$ follows a distribution with mean 0 and covariance matrix $\boldsymbol{\Sigma}_{(i)} \in \mathbb{R}^{r\times r}$. From this formulation, the errors have heteroscedastic covariance structure.

The heteroscedastic envelope model applies the enveloping idea on all the $\boldsymbol{\beta}_{(i)}$'s, and at the same time accommodates the heteroscedastic covariance structure. Let $\mathcal{M} = \{\boldsymbol{\Sigma}_{(i)} : i = 1, \cdots, p\}$ be the collection of covariance matrices and let $\mathcal{B} = \text{span}(\boldsymbol{\beta}_{(1)}, \cdots, \boldsymbol{\beta}_{(p)})$. The $\mathcal{M}$-envelope of $\mathcal{B}$, denoted by $\mathcal{E}_{\mathcal{M}}(\mathcal{B})$, is the intersection of all subspaces that contain $\mathcal{B}$ and reduce each member of $\mathcal{M}$. The coordinate form of this model is

$$\mathbf{Y}_{(i)j} = \boldsymbol{\mu} + \boldsymbol{\Gamma}\boldsymbol{\eta}_{(i)} + \boldsymbol{\varepsilon}_{(i)j}, \quad \boldsymbol{\Sigma}_{(i)} = \boldsymbol{\Sigma}_{1(i)} + \boldsymbol{\Sigma}_2 = \boldsymbol{\Gamma}\boldsymbol{\Omega}_{1(i)}\boldsymbol{\Gamma}^\top + \boldsymbol{\Gamma}_0\boldsymbol{\Omega}_0\boldsymbol{\Gamma}_0^\top,$$

where $\boldsymbol{\beta}_{(i)} = \boldsymbol{\Gamma}\boldsymbol{\eta}_{(i)}$, $\boldsymbol{\Gamma} \in \mathbb{R}^{r\times u}$ is a semi-orthogonal matrix that spans $\mathcal{E}_{\mathcal{M}}(\mathcal{B})$, $\boldsymbol{\Gamma}_0 \in \mathbb{R}^{r\times(r-u)}$ spans its orthogonal complement, $\boldsymbol{\eta}_{(i)} = \boldsymbol{\Gamma}^\top\boldsymbol{\beta}_{(i)} \in \mathbb{R}^u$, $\boldsymbol{\Omega}_{1(i)} \in \mathbb{R}^{u\times u}$ and $\boldsymbol{\Omega}_0 \in \mathbb{R}^{(r-u)\times(r-u)}$ are both positive definite matrices, and u is the dimension of $\mathcal{E}_{\mathcal{M}}(\mathcal{B})$. When $u = r$, the heteroscedastic envelope model degenerates to the standard model (3).

Compared with the envelope model in Section 2.1, recognizing the heteroscedastic error structure leads to more reliable estimators and greater efficiency gains. To test homogeneity of the covariance matrices, Box's M test (Johnson and Wichern 2007) can be used.

2.4. Inner envelope model

The inner envelope model (Su and Cook 2012) provides an envelope method that can achieve efficiency gains even when all of $\mathbf{Y}$ is material. It has a different mechanism in utilizing the tool of reducing subspaces. Under the standard model (1), an inner $\boldsymbol{\Sigma}$-envelope of $\mathcal{B}$, denoted by $\mathcal{IE}_{\boldsymbol{\Sigma}}(\mathcal{B})$, is the reducing subspace of $\boldsymbol{\Sigma}$ with maximal dimension that is contained in $\mathcal{B}$. Let $\boldsymbol{\Gamma}_1 \in \mathbb{R}^{r\times u}$ be an orthogonal basis that spans $\mathcal{IE}_{\boldsymbol{\Sigma}}(\mathcal{B})$. The coordinate form of the inner envelope model is then

$$\mathbf{Y} = \boldsymbol{\alpha} + (\boldsymbol{\Gamma}_1\boldsymbol{\eta}_1^\top + \boldsymbol{\Gamma}_0\mathbf{B}\boldsymbol{\eta}_2^\top)\mathbf{X} + \boldsymbol{\varepsilon}, \quad \boldsymbol{\Sigma} = \boldsymbol{\Gamma}_1\boldsymbol{\Omega}_1\boldsymbol{\Gamma}_1^\top + \boldsymbol{\Gamma}_0\boldsymbol{\Omega}_0\boldsymbol{\Gamma}_0^\top,$$

where $\boldsymbol{\beta} = \boldsymbol{\Gamma}_1\boldsymbol{\eta}_1^\top + \boldsymbol{\Gamma}_0\mathbf{B}\boldsymbol{\eta}_2^\top$, $\boldsymbol{\Gamma}_0 \in \mathbb{R}^{r\times(r-u)}$ spans $\mathcal{IE}_{\boldsymbol{\Sigma}}(\mathcal{B})^\perp$, $\mathbf{B} \in \mathbb{R}^{(r-u)\times(p-u)}$ is an orthogonal matrix such that $\text{span}(\boldsymbol{\Gamma}_1, \boldsymbol{\Gamma}_0\mathbf{B}) = \mathcal{B}$, $\boldsymbol{\eta}_1 \in \mathbb{R}^{p\times u}$, $\boldsymbol{\eta}_2 \in \mathbb{R}^{p\times(p-u)}$, $\boldsymbol{\Omega}_1 \in \mathbb{R}^{u\times u}$ and $\boldsymbol{\Omega}_0 \in \mathbb{R}^{(r-u)\times(r-u)}$ are positive definite matrices, u is the dimension of $\mathcal{IE}_{\boldsymbol{\Sigma}}(\mathcal{B})$.

The inner envelope model divides $\boldsymbol{\beta}$ into two parts by $\mathcal{IE}_{\boldsymbol{\Sigma}}(\mathcal{B})$. If the part $\boldsymbol{\Gamma}_1\boldsymbol{\eta}_1^\top$ is estimated with greater precision than the standard model (particularly when $\|\boldsymbol{\Sigma}_1\| \ll \|\boldsymbol{\Sigma}_2\|$), and the part $\boldsymbol{\Gamma}_0\mathbf{B}\boldsymbol{\eta}_2^\top$ is estimated with about the same precision, then overall we get better efficiency in estimating $\boldsymbol{\beta}$. The possible values of u are from 0 to p. When $u = 0$, the inner envelope model reduces to the standard model and when $u = p$, the inner envelope model is equivalent to the envelope model in Section 2.1.

2.5. Scaled envelope model

The scaled envelope model (Cook and Su 2013) is a scale invariant version of the envelope model in Section 2.1. It is invariant under scale transformation of the responses and can achieve efficiency gains beyond those offered by the envelope model. It is an alternative choice to the envelope model especially when $u = r$ is inferred via the envelope model.

Let $\boldsymbol{\Lambda} \in \mathbb{R}^{r\times r}$ be a diagonal matrix to represent the scale transformation of the responses. Its diagonal elements are $\lambda_i > 0$, $i = 1, \cdots, r$, with $\lambda_1 = 1$ and the rest to be estimated. Under the framework of (1), the coordinate form of a scaled envelope model is

$$\mathbf{Y} = \boldsymbol{\alpha} + \boldsymbol{\Lambda}\boldsymbol{\Gamma}\boldsymbol{\eta}\mathbf{X} + \boldsymbol{\varepsilon}, \quad \boldsymbol{\Sigma} = \boldsymbol{\Lambda}\boldsymbol{\Gamma}\boldsymbol{\Omega}\boldsymbol{\Gamma}^\top\boldsymbol{\Lambda} + \boldsymbol{\Lambda}\boldsymbol{\Gamma}_0\boldsymbol{\Omega}_0\boldsymbol{\Gamma}_0^\top\boldsymbol{\Lambda},$$

where $\boldsymbol{\Gamma} \in \mathbb{R}^{r\times u}$ is a semi-orthogonal matrix that spans the $\boldsymbol{\Lambda}^{-1}\boldsymbol{\Sigma}\boldsymbol{\Lambda}^{-1}$-envelope of $\boldsymbol{\Lambda}^{-1}\mathcal{B}$, $\boldsymbol{\Gamma}_0 \in \mathbb{R}^{r\times(r-u)}$ is the completion of $\boldsymbol{\Gamma}$, $\boldsymbol{\eta} \in \mathbb{R}^{u\times p}$, $\boldsymbol{\Omega} \in \mathbb{R}^{u\times u}$ and $\boldsymbol{\Omega}_0 \in \mathbb{R}^{(r-u)\times(r-u)}$ are positive definite matrices, and u is the dimension of the $\boldsymbol{\Lambda}^{-1}\boldsymbol{\Sigma}\boldsymbol{\Lambda}^{-1}$-envelope of $\boldsymbol{\Lambda}^{-1}\mathcal{B}$. The scaled envelope model reduces to the standard model (1) when $u = r$. Like the other envelope methods, the goal of the scaled envelope is to improve estimation efficiency in the estimation of $\boldsymbol{\beta} = \boldsymbol{\Lambda}\boldsymbol{\Gamma}\boldsymbol{\eta}$. When $u < r$, the scaled envelope model is not nested within the standard model or any scaled envelope model with a large dimension, so likelihood ratio testing cannot be applied for selection of u.

2.6. Envelope model in the predictor space

The envelope model in the predictor space (Cook *et al.* 2013) is based on the possibility that the distribution of the full response vector $\mathbf{Y}$ is invariant to changes in some linear combinations of the predictors $\mathbf{X}$. It can be applied under the context of (1) with the response being univariate or multivariate. In terms of prediction, the performance of this estimator is asymptotically as good as or better than the least squares estimator. In population, it is equivalent to the partial least squares estimator obtained from the SIMPLS algorithm (de Jong 1993), but it typically has better performance with finite samples.

In contrast to the previous envelope models, we now assume that the predictors are random so $(\mathbf{Y}, \mathbf{X})$ has a joint distribution. Let $\boldsymbol{\Sigma}_{\mathbf{X}} = \mathsf{VAR}(\mathbf{X})$ and let $\mathcal{B}^* = \mathrm{span}(\boldsymbol{\beta}^\top)$. Then the $\boldsymbol{\Sigma}_{\mathbf{X}}$-envelope of $\mathcal{B}^*$, denoted by $\mathcal{E}_{\boldsymbol{\Sigma}_{\mathbf{X}}}(\mathcal{B}^*)$, is the smallest reducing subspace of $\boldsymbol{\Sigma}_{\mathbf{X}}$ that contains $\mathcal{B}^*$. Letting $\boldsymbol{\Gamma} \in \mathbb{R}^{p\times u}$ be an orthogonal basis of $\mathcal{E}_{\boldsymbol{\Sigma}_{\mathbf{X}}}(\mathcal{B}^*)$, the coordinate form of the envelope model in the predictor space is

$$\mathbf{Y} = \boldsymbol{\mu} + \boldsymbol{\eta}^\top\boldsymbol{\Omega}^{-1}\boldsymbol{\Gamma}^\top\mathbf{X} + \boldsymbol{\varepsilon}, \quad \boldsymbol{\Sigma}_{\mathbf{X}} = \boldsymbol{\Gamma}\boldsymbol{\Omega}\boldsymbol{\Gamma}^\top + \boldsymbol{\Gamma}_0\boldsymbol{\Omega}_0\boldsymbol{\Gamma}_0^\top, \tag{4}$$

where $\boldsymbol{\beta} = \boldsymbol{\eta}^\top\boldsymbol{\Omega}^{-1}\boldsymbol{\Gamma}^\top$, $\boldsymbol{\Gamma}_0 \in \mathbb{R}^{p\times(p-u)}$ is an orthogonal basis for $\mathcal{E}^{\perp}_{\boldsymbol{\Sigma}_{\mathbf{X}}}(\mathcal{B}^*)$, $\boldsymbol{\eta} \in \mathbb{R}^{u\times r}$, $\boldsymbol{\Omega} \in \mathbb{R}^{u\times u}$, $\boldsymbol{\Omega}_0 \in \mathbb{R}^{(p-u)\times(p-u)}$, and u is the dimension of the envelope $\mathcal{E}_{\boldsymbol{\Sigma}_{\mathbf{X}}}(\mathcal{B}^*)$. When $u = p$, this envelope model reduces to the standard model (1).

Under model (4), let $\mathcal{E}$ denote $\mathcal{E}_{\mathbf{\Sigma_X}}(\mathcal{B}^*)$ for subscripts, then $\mathbf{Y}$ is conditionally uncorrelated with $\mathbf{Q}_\mathcal{E}\mathbf{X}$ given $\mathbf{P}_\mathcal{E}\mathbf{X}$, and $\mathbf{Q}_\mathcal{E}\mathbf{X}$ is uncorrelated with $\mathbf{P}_\mathcal{E}\mathbf{X}$. Then $\mathbf{Q}_\mathcal{E}\mathbf{X}$ is immaterial to the regression. By recognizing and accounting for the immaterial part, the envelope model (4) has a better prediction performance than the standard model or even the partial least squares estimator.

2.7. Envelope model with small sample size

When the sample size is smaller than r in the envelope model (Section 2.1) or p in the envelope model in the predictor space (Section 2.6), the usual envelope estimators cannot be computed. In these cases, a sequential algorithm (Cook 2012) can be used to obtain estimators that are (i) equivalent to the usual envelope estimators in the population, (ii) not generally as efficient when $n > r$ or $n > p$, but (iii) can still provide useful results in small samples.

The usual estimators of an envelope subspace are obtained by optimizing an objective function over a Grassmann manifold. For example, to estimate $\mathcal{E}_{\mathbf{\Sigma}}(\mathcal{B})$ (cf. Section 2.1), we minimize the following objective function over a Grassmann manifold $\mathcal{G}(r, u)$:

$$\widehat{\mathbf{\Gamma}} = \arg\min_{\mathbf{\Gamma}\in\mathcal{G}(r,u)} \log|\mathbf{\Gamma}^\top\widehat{\mathbf{\Sigma}}_{\text{res}}\mathbf{\Gamma}| + \log|\mathbf{\Gamma}^\top\widehat{\mathbf{\Sigma}}_{\mathbf{Y}}^{-1}\mathbf{\Gamma}|,$$

where $\widehat{\mathbf{\Sigma}}_{\mathbf{Y}} \in \mathbb{R}^{r\times r}$ is the sample covariance matrix of $\mathbf{Y}$, $\widehat{\mathbf{\Sigma}}_{\text{res}} \in \mathbb{R}^{r\times r}$ is the sample covariance matrix of the residuals from the least squares regression of $\mathbf{Y}$ given $\mathbf{X}$, and $|\cdot|$ is the determinant. The matrix $\widehat{\mathbf{\Sigma}}_{\mathbf{Y}}$ is singular when the sample size is smaller than r and consequently the objective fuction is not well-defined. However, a sequential algorithm can be used to obtain an alternative estimator of $\mathbf{\Gamma}$. This estimator then allows straightforward computation of the other parameters in the envelope model, including $\boldsymbol{\beta}$.

Let $\mathbf{u} \in \mathbb{R}^{a\times b}$ have $\text{rank}(\mathbf{u}) \le b$, let $\mathcal{S} = \text{span}(\mathbf{u}) \subseteq \text{span}(\mathbf{M})$, where $\mathbf{M} \in \mathbb{R}^{a\times a}$ is a semi positive-definite matrix. Suppose that the $\mathbf{M}$-envelope of $\mathcal{S}$, $\mathcal{E}_\mathbf{M}(\mathcal{S})$, has dimension d. Set $\mathbf{w}_0 = 0$, $\mathbf{W} = \mathbf{w}_0$, and $\mathbf{U} = \mathbf{u}\mathbf{u}^\top$. Then, for $k = 0, 1, \cdots, d-1$, construct

$$\begin{aligned}
\mathcal{E}_k &= \text{span}(\mathbf{M}\mathbf{W}_k) \\
\mathbf{w}_{k+1} &= l_1(\mathbf{Q}_{\mathcal{E}_k}\mathbf{U}\mathbf{Q}_{\mathcal{E}_k}) \\
\mathbf{W}_{k+1} &= \text{span}(\mathbf{w}_0, \cdots, \mathbf{w}_k, \mathbf{w}_{k+1}),
\end{aligned}$$

where $l_1(\mathbf{A})$ means the eigenvector corresponding to the largest eigenvalue of $\mathbf{A}$. At termination, $\mathcal{E}_\mathbf{M}(\mathcal{S}) = \text{span}(\mathbf{W}_d)$. The sample version of this algorithm is obtained by simply substituting sample versions of $\mathbf{U}$ and $\mathbf{M}$.

This sequential algorithm can be used for estimating a general envelope subspace. In this toolbox, it is implemented for the envelope model and the envelope model in the predictor space. With envelope model in the predictor space, Cook *et al.* (2013) showed that in the population the envelope subspace provided by this algorithm is the same as that provided by the SIMPLS algorithm.

The sequential algorithm described above can also be used for large sample size cases, and it is much faster than performing the Grassmann manifold optimization. It also provides a $\sqrt{n}$ consistent estimator of the envelope subspace, although with large sample size, this estimator's performance may not be as good as that of the estimator based on Grassmann optimization.

2.8. Envelope estimator for multivariate mean

The context for the envelope methodology in this section is a bit different from that in previous sections, as now we consider estimating the multivariate mean, not fitting multivariate linear regression. Assuming that the sample $\mathbf{Y}_1, \cdots, \mathbf{Y}_n$ is independent and identically distributed with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma} \in \mathbb{R}^{p\times p}$, the sample mean $\bar{\mathbf{Y}} = \sum_{i=1}^{n} \mathbf{Y}_i$ is a natural estimator of $\boldsymbol{\mu}$. James and Stein proved that this estimator is not admissible and is dominated by the James-Stein estimator for $p \geq 3$. Preliminary investigations have indicated that the envelope estimator for multivariate mean has a smaller mean square error than $\bar{\mathbf{Y}}$, and it often has a smaller mean square error than the James-Stein estimator.

The envelope estimator for the multivariate mean is based on the assumption that $\boldsymbol{\mu}$ is orthogonal to some eigenvectors of $\boldsymbol{\Sigma}$. Diaconis and Freedman (1984) showed that as the dimension tends to infinity, two random vectors are orthogonal to each other with probability 1. In the envelope model for estimating the multivariate mean, it is assumed that $\boldsymbol{\mu}$ lies within the space spanned by a subset of the eigenvectors of $\boldsymbol{\Sigma}$, and we call the space $\mathcal{S}$. By a result in Cook *et al.* (2010), $\mathcal{S}$ is the $\boldsymbol{\Sigma}$-envelope of $\mathcal{M}$, where $\mathcal{M} = \text{span}(\boldsymbol{\mu})$.

Let $\boldsymbol{\Gamma} \in \mathbb{R}^{p\times u}$ be a semi-orthogonal matrix that spans $\mathcal{E}_{\boldsymbol{\Sigma}}(\mathcal{M})$, then the envelope model is

$$\boldsymbol{\mu} = \boldsymbol{\Gamma}\boldsymbol{\eta}, \quad \boldsymbol{\Sigma} = \boldsymbol{\Gamma}\boldsymbol{\Omega}\boldsymbol{\Gamma}^\top + \boldsymbol{\Gamma}_0\boldsymbol{\Omega}_0\boldsymbol{\Gamma}_0^\top,$$

where u is the dimension of $\mathcal{E}_{\boldsymbol{\Sigma}}(\mathcal{M})$, $\boldsymbol{\Gamma}_0 \in \mathbb{R}^{p\times(p-u)}$ is the completion of $\boldsymbol{\Gamma}$, $\boldsymbol{\eta} \in \mathbb{R}^u$, $\boldsymbol{\Omega} \in \mathbb{R}^{u\times u}$ and $\boldsymbol{\Omega}_0 \in \mathbb{R}^{(p-u)\times(p-u)}$ carry the coordinates. The envelope estimator has the form $\hat{\boldsymbol{\mu}}_{em} = \mathbf{P}_{\widehat{\boldsymbol{\Gamma}}}\bar{\mathbf{Y}}$.

The difference between the James-Stein estimator and the envelope estimator can be visualized in Figure 1. In the figure, the ellipse represents the distribution of $\bar{\mathbf{Y}}$. The James-Stein estimator of $\boldsymbol{\mu}$ is denoted as $\hat{\boldsymbol{\mu}}_{JS}$, and it shrinks $\bar{\mathbf{Y}}$ towards the origin. In contrast to $\hat{\boldsymbol{\mu}}_{JS}$, the envelope estimator $\hat{\boldsymbol{\mu}}_{em}$ is the projection of $\bar{\mathbf{Y}}$ onto the estimated envelope $\hat{\mathcal{E}}_{\boldsymbol{\Sigma}}(\mathcal{M})$. In this figure, $\mathcal{E}_{\boldsymbol{\Sigma}}(\mathcal{M})$ aligns with the eigenvector corresponding to the smaller eigenvalue of $\boldsymbol{\Sigma}$. Then the envelope estimator $\hat{\boldsymbol{\mu}}_{em}$ is much less variant than $\bar{\mathbf{Y}}$, and it is expected to have a smaller mean squared error than $\bar{\mathbf{Y}}$, or even $\hat{\boldsymbol{\mu}}_{JS}$.

2.9. Role of normality

None of the envelope models discussed in Sections 2.1–2.8 require constraints on the distribution of the errors $\boldsymbol{\varepsilon}$ beyond those listed previously. Adding the assumption that the errors are normally distributed facilitates an analysis by providing a well-defined likelihood and asymptotic standard errors. Excluding the sequential methods, all fitting in the **envlp** toolbox is based on normal likelihoods, along with their corresponding inference methods. Those likelihoods also provide $\sqrt{n}$-consistent estimators without normality and experience has shown that they perform well in non-normal settings. However, inference methods may be impacted by clear deviations from normality and then it is recommended that the bootstrap methods available in the **envlp** toolbox be used for standard errors and inference. The bootstrap is the only method provided for computing standard errors for the sequential estimators, as listed in Table 1.

3. Toolbox overview

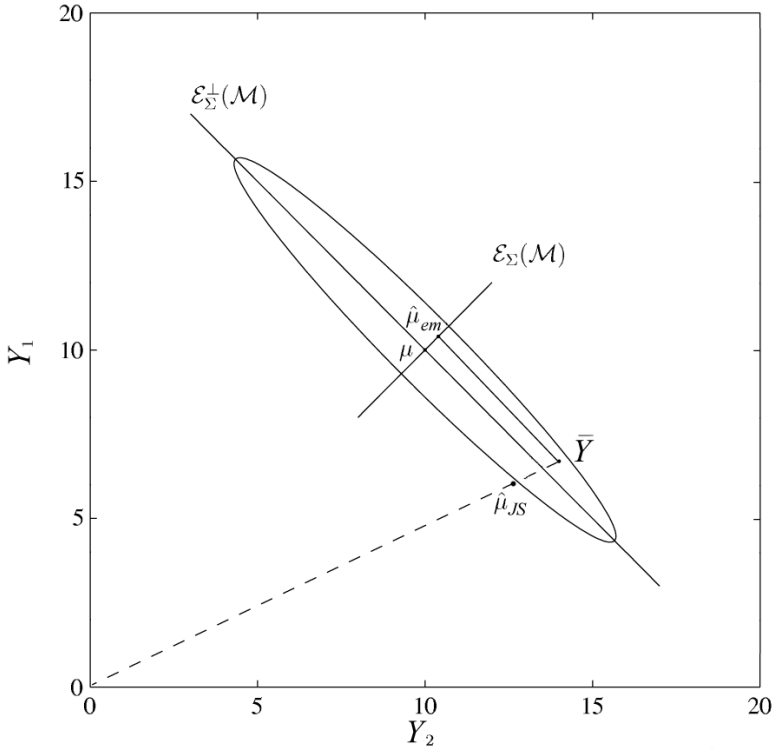

Figure 1: Graphical illustration of James-Stein estimator $\hat{\boldsymbol{\mu}}_{JS}$ and envelope estimator $\hat{\boldsymbol{\mu}}_{em}$.

The toolbox **envlp** implements all the envelope methods discussed in Section 2. It is modularized, with nine modules, each written for a model: `env` for the basic envelope model, `henv` for the heteroscedastic envelope model, `ienv` for the inner envelope model, `penv` for the partial envelope model, `senv` for the scaled envelope model, `xenv` for the envelope model in the predictor space, `envmean` for the envelope estimator of the multivariate mean, `envseq` and `xenvpls` are counterparts of `env` and `xenv` in small sample size cases. Each module has three parts: the core function that fits the model, dimension selection functions, and inference tools. In this toolbox, the core function always has the same name as the module. The dimension selection functions and inference tools available are different from module to module, as the nature of the models is different. All modules will be described in details later in this section. The structure of this toolbox is summarized in Table 1.

This toolbox relies on MATLAB toolbox **sg_min 2.4.3** (Lippert 2004) for Grassmann manifold optimization. **sg_min 2.4.3** uses the analytical first derivative and numerical second derivative of the objective function to perform the optimization, and we find it is stable. Some modifications are made to it for the envelope model context. A few auxiliary functions in the toolbox **envlp** rely on MATLAB **Statistics** toolbox (The MathWorks, Inc. 2012a), **LDR** toolbox (Cook *et al.* 2009), **Tcodes** toolbox (Strang 2000) and function `MBoxtest` (Trujillo-Ortiz and Hernandez-Walls 2002).

To install the toolbox, direct the MATLAB working directory to the folder "envlp", and type the command `install_envlp`. If a previous version is present, simply replace the folder by that of the latest version and type `install_envlp`. The installation will be completed if you agree with the license agreement. You do not need to load the auxiliary functions or

Module	Dimension selection	Inference tools	Section
`env`	AIC BIC LRT m-fold CV	Estimation and prediction Bootstrap for estimating standard errors Hypothesis test on coefficients	2.1
`envseq`	m-fold CV	Bootstrap for estimating standard errors	2.7
`henv`	AIC BIC LRT m-fold CV	Estimation and prediction Bootstrap for estimating standard errors Hypothesis test on coefficients	2.3
`ienv`	AIC BIC LRT m-fold CV	Estimation and prediction Bootstrap for estimating standard errors Hypothesis test on coefficients	2.4
`penv`	AIC BIC LRT m-fold CV	Estimation and prediction Bootstrap for estimating standard errors Hypothesis test on coefficients	2.2
`senv`	AIC BIC m-fold CV	Estimation and prediction Bootstrap for estimating standard errors Hypothesis test on coefficients	2.5
`xenv`	AIC BIC LRT m-fold CV	Estimation and prediction Bootstrap for estimating standard errors Hypothesis test on coefficients	2.6
`xenvpls`	m-fold CV	Bootstrap for estimating standard errors	2.7
`envmean`	AIC BIC LRT m-fold CV	Estimation and prediction Bootstrap for Estimating Standard Errors Hypothesis test	2.8

Table 1: Structure of toolbox **envlp**.

toolboxes mentioned before. Once the toolbox is installed, you can call functions or datasets in the toolbox from any current working directory.

The toolbox contains three types of functions: the core functions, functions for dimension selection and functions for inference tools. Section 3.1 to 3.3 are devoted to the description of these three types.

3.1. Core functions

The functions that fit the envelope models are the core functions of this package. There are nine of them, one in each module, and they share the same names as the module names. For example, the function `env` fits the envelope model, and the function `envmean` finds the envelope estimator for the multivariate mean. The envelope models in the regression context are `env`, `henv`, `ienv`, `penv`, `senv`, `xenv`, `envseq` and `xenvpls`. The inputs for these models are `X`, `Y` and `u`, where `X` and `Y` store the data matrices for the predictors and the responses, and `u` is the dimension of the envelope, which can be obtained by the functions discussed in Section 3.2. The inputs for `envmean` are the data matrix `Y` and the dimension of the envelope `u`, as this context does not involve any predictors. The output of these nine functions is a list containing the envelope estimators of model parameters, and important statistics calculated from the models like the value of the maximized log-likelihood function, asymptotic covariance matrix of the estimators, number of parameters in the model and many others.

We present an example by applying the envelope model to the wheat protein data in Cook *et al.* (2010). The wheat protein data contains seven variables, the logarithms of near infrared reflectance measured at six wavelengths and a group indictor taking value 0 or 1 for wheat with low or high protein content. In multivariate linear regression (1), we take the group indicator as the predictor and the spectral measurements as responses. The regression coefficients are then the mean differences between the two groups. For demonstration purpose, we take only the third and fourth measurements as responses, so that we can visualize the data. First we load the data and assign the predictor and responses.

```
load wheatprotein.txt
X = wheatprotein(:, 8);
Y = wheatprotein(:, 3 : 4);
```

Figure 2 displays the data with two axis assigned to the two responses. For better visualization, we centered the data to have mean 0. Under the standard model, the estimated coefficients in $\boldsymbol{\beta}$ are 7.52 and -2.06, with the associated standard errors for these two elements being 8.64 and 9.49. The standard errors returned by `Out.asySE` are asymptotic, for actual standard errors, we need to multiply by $1/\sqrt{n}$, where n is the sample size.

```
Out1 = fit_OLS(X, Y);
Out1.betaOLS


ans =

    7.5224
   -2.0609
```

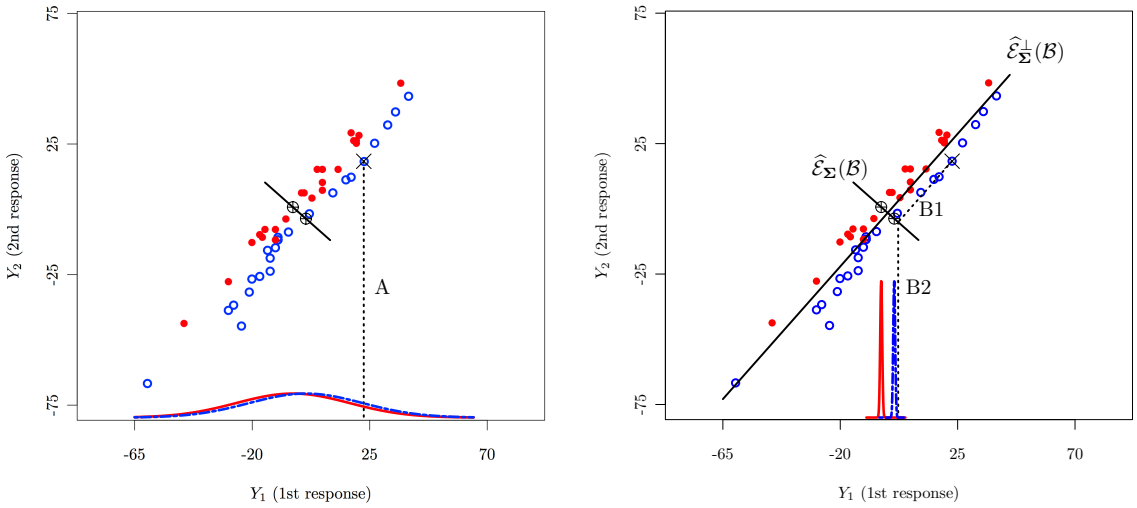

Figure 2: Graphical illustration of the working mechanism of the envelope model. The solid dots mark the wheat with high protein content and the open circles mark the wheat with low protein content.

```
n = 50;
Out1.asySE / sqrt(n)

ans =

    8.6372
    9.4852
```

The standard errors are large relative to the absolute value of elements in $\widehat{\boldsymbol{\beta}}$, so it is hard to tell the difference between the two groups. The two curves in the left panel of Figure 2 present the projection distribution of the two groups onto the Y_1 axis, with the solid line for the high protein group and the dashed line for the low protein group. The projection path for a sample point 'x' is marked as A in the plot. We notice that the two curves almost overlap with each other, so it is hard to distinguish between the two groups. This is consistent with the comparison of the absolute values of elements in $\widehat{\boldsymbol{\beta}}$ and their associated standard errors.

To fit the envelope model to this data, we need the dimension of the envelope. Dimension selection will be discussed in Section 3.2, for now we just fixed the dimension of the envelope at 1.

```
ModelOutput = env(X, Y, 1);
ModelOutput

ModelOutput =
```

```
         beta: [2x1 double]
        Sigma: [2x2 double]
        Gamma: [2x1 double]
       Gamma0: [2x1 double]
          eta: -6.9506
        Omega: 6.0042
       Omega0: 2.0510e+03
        alpha: [2x1 double]
            l: -377.3568
    covMatrix: [2x2 double]
       asyEnv: [2x1 double]
        ratio: [2x1 double]
     paramNum: 6
            n: 50
```

After fitting the envelope model, the output is a list containing the estimates of regression coefficients $\boldsymbol{\beta}$, error covariance matrix $\boldsymbol{\Sigma}$, parameters in the envelope model including $\boldsymbol{\Gamma}$, $\boldsymbol{\eta}$, $\boldsymbol{\Omega}$, and $\boldsymbol{\Omega}_0$, as well as important statistics like the value of the maximized log-likelihood l, the asymptotic covariance matrix of vectorized $\widehat{\boldsymbol{\beta}}$, the asymptotic standard error for each element in $\widehat{\boldsymbol{\beta}}$, the number of parameters in the model and the sample size. To get the estimated group difference, we call the respective component in the list `ModelOutput.beta`. Similar to the standard model, we can get the standard errors for elements in $\boldsymbol{\beta}$ by dividing their asymptotic standard errors by $\sqrt{n}$.

```
ModelOutput.beta

ans =

    5.1405
   -4.6782

ModelOutput.asySE / sqrt(n)

ans =

    0.5142
    0.4685
```

The envelope estimators of the two elements in $\widehat{\boldsymbol{\beta}}$ are 5.14 and -4.68, with standard errors 0.51 and 0.47. Compared to the size of the elements in $\widehat{\boldsymbol{\beta}}$, the standard errors are small and it is easy to tell the difference between the two groups. The right panel of Figure 2 illustrates the envelope analysis: The envelope model identifies the variation in the direction of $\widehat{\mathcal{E}}^{\perp}_{\boldsymbol{\Sigma}}(\mathcal{B})$ as carrying no information on $\boldsymbol{\beta}$, so a sample data point 'x' is projected first onto the envelope subspace $\widehat{\mathcal{E}}_{\boldsymbol{\Sigma}}(\mathcal{B})$, and then onto the Y_1 axis. The projection route is marked as B. The uniqueness of the envelope model is reflected on the first segment of B, which accounts for the immaterial information in the data. The two curves on the Y_1 axis are projection distributions of the two groups, with each data point following route similar to B. The two

curves are well separated, indicating that we have obtained substantial efficiency gains. To quantify the gains, we can compare the standard errors of the standard estimator and the envelope estimator by taking their ratios. In this example, the ratios are 16.80 and 20.25 for the two elements in $\boldsymbol{\beta}$.

3.2. Dimension selection

Likelihood based methods including Akaike information criteria (AIC), Bayesian information criteria (BIC) and likelihood ratio testing (LRT) are implemented for selecting the dimension of an envelope. In small sample size cases where the likelihood is not well-determined, we select the dimension by m-fold cross validation.

The functions `modelselectaic`, `modelselectbic` and `modelselectlrt` choose the dimension for the envelope models in the regression context by AIC, BIC and LRT. The common inputs for these three functions are data matrix `X`, `Y`, and `modelType`, while LRT has an additional input `alpha` indicating the significance level. The choices for `modelType` are `'env'`, `'henv'`, `'ienv'`, `'penv'`, `'senv'` and `'xenv'`.

The function `mfoldcv` chooses the dimension of the envelope models by m-fold cross validation. It divides the data into m folds of about equal size, and then uses one fold in turn as testing samples and the rest as training samples. The function returns the dimension that minimizes the average squared prediction errors using the identity inner product. The inputs for `mfoldcv` are data matrices `X`, `Y`, number of folds `m` and `modelType`. This method can be applied to any model, so the choices for `modelType` are `'env'`, `'envseq'`, `'henv'`, `'ienv'`, `'penv'`, `'senv'`, `'xenv'`, `'xenvpls'` and `'envmean'`.

We write separate dimension selection functions for envelope estimator of multivariate means, as they have different input variables. The input variable of `aic_envmean`, `bic_envmean` and `lrt_envmean` is the data matrix `Y` only.

The output for all the dimension selection functions is an integer `u` for the dimension of the envelope subspace.

Back to the wheat protein data example discussed in Section 3.1, we applied AIC, BIC and LRT with significance level 0.01 to select the dimension.

```
u1 = modelselectaic(X, Y, 'env');
u1

u1 =

     1

u2 = modelselectbic(X, Y, 'env');
u2

u2 =

     1

u3 = modelselectlrt(X, Y, 0.01, 'env');
u3
```

```
u3 =

     1
```

We notice that all three criteria agree that the dimension of the envelope subspace is 1. According to the right panel in Figure 2, $u = 1$ is well agreed by the data, and the estimated envelope subspace $\widehat{\mathcal{E}}_{\mathbf{\Sigma}}(\mathcal{B})$ is marked in the plot.

3.3. Inference tools

The inference tools provided by toolbox **envlp** include bootstrap estimation of standard errors, estimation and prediction at a new observation, and hypothesis testing.

The function `bootstrapse` computes the standard errors for elements in the estimated regression coefficients by bootstrapping the residuals. Its inputs are data matrices `X`, `Y`, the dimension of the envelope `u`, number of bootstrap sample `B`, and `modelType`, which can be `env`, `envseq`, `henv`, `ienv`, `penv`, `senv`, `xenv` or `xenvpls`. The output `bootse` is a matrix having the same dimension as $\boldsymbol{\beta}$ with each element being the standard error of the corresponding element in $\widehat{\boldsymbol{\beta}}$. The function `btrsp_envmean` computes the standard errors for elements in $\hat{\boldsymbol{\mu}}_{em}$. Its inputs and output are similar to `bootstrapse`, except that it does not need `X` and `modelType` for input.

The function `predict` performs estimation or prediction for envelope models in the regression context. It returns a list `PredictOutput` which includes the estimated or predicted value, its standard errors and covariance matrix. The input `ModelOutput` is the output list from the core functions, `Xnew` is a column vector containing the value of $\mathbf{X}$ at which to estimate or predict `Y`, `infType` can be chosen from '`estimation`' or '`prediction`', and `modelType` can be `env`, `henv`, `ienv`, `penv`, `senv` or `xenv`. In the context of estimating a multivariate mean, the prediction function is called `predict_envmean`. It has similar structure as `predict` except that it does not have inputs `Xnew` and `modelType`.

The function `testcoefficient` tests if certain linear combination of the rows or columns of the regression coefficients are equal to some pre-specified values. More specifically, letting $\mathbf{L}$, $\mathbf{R}$ and $\mathbf{A}$ be $a \times r$, $p \times b$ and $a \times b$ matrices of constants, `testcoefficient` tests $H_0 : \mathbf{L}\boldsymbol{\beta}\mathbf{R} = \mathbf{A}$ versus $H_a : \mathbf{L}\boldsymbol{\beta}\mathbf{R} \neq \mathbf{A}$. The inputs are `ModelOutput` which is the output from the core functions, `TestInput` which is a list that specifies $\mathbf{L}$, $\mathbf{R}$ and $\mathbf{A}$ in the hypotheses and `modelType` which can be chosen from `env`, `henv`, `ienv`, `penv`, `senv` and `xenv`. The output `TestOutput` is a list that contains test statistic, degrees of freedom, p value and the covariance matrix of vectorized $\mathbf{L}\widehat{\boldsymbol{\beta}}\mathbf{R}$. At the same time, a table is printed out to display the test results. The function `testcoefficient_envmean` is for testing $H_0 : \mathbf{L}\boldsymbol{\mu} = \mathbf{A}$ versus $H_a : \mathbf{L}\boldsymbol{\mu} \neq \mathbf{A}$, where $\boldsymbol{\mu}$ is the multivariate mean, $\mathbf{L}$ is an $a \times r$ matrix and $\mathbf{A}$ is an $a \times 1$ vector. The output of `testcoefficient_envmean` has the same form as `testcoefficient`, but its input does not include `modelType`.

Continuing with the wheat protein example, the standard error of each element in $\widehat{\boldsymbol{\beta}}$ can also be estimated by residual bootstrap, which be obtained by the command `bootstrapse`. The inputs for `bootstrapse` are the predictors `X`, the responses `Y`, the dimension of the envelope model `u`, the number of bootstrap samples `B`, and a string that represents the model `modelType`. We took $u = 1$ as discussed in Section 3.2, and we put `'env'` for `modelType`.

```
B = 100;
```

```
bootse = bootstrapse(X, Y, 1, B, 'env')

bootse =

    0.5213
    0.4767
```

Recall that the standard errors calculated using asymptotic standard errors are 0.5142 and 0.4685, which are quite close to the bootstrap standard errors. We do not set seeds for the function `bootstrapse`, so the user can get different results each time he runs the function. But when `B` is large, the results should be close to each other.

Now to test if $\boldsymbol{\beta} = 0$, we use the function `testcoefficient`. If we do not input $\mathbf{L}$, $\mathbf{R}$ and $\mathbf{A}$ and leave the input `TestInput` as blank, then by default it is testing if $\boldsymbol{\beta} = 0$.

```
TestOutput = testcoefficient(ModelOutput, 'env');

 Test Hypothesis     Chisq Statistic    DF     P-value
-------------------------------------------------------
L * beta * R = A          100.416        2      0.0000
-------------------------------------------------------
```

The output table shows a highly significant p value, which is strong evidence that the two wheat groups are different.

3.4. Monitoring and controlling the convergence speed

The running time for most examples in the package is in the order of seconds, some are in the order of minutes. It can take longer for larger data sets. Envelope estimation relies on Grassmann manifold optimization, which uses an iterative algorithm. The running time of the functions depends on the nature of the methods, tolerance levels for convergence and the starting value. For example, using AIC or BIC for dimension selection takes longer than using LRT because of different stopping criteria; `senv` runs longer than `env` because of its method of estimation; setting the tolerance level at 10^{-7} can reduce running time than setting the tolerance level at 10^{-9}. For this purpose, we add an optional argument `Opts` to each function so that the user can monitor the iteration process and adjust the tolerance level. `Opts` is a list, and it provides the user the option to display the current number of iteration, specify a starting value, control the maximum number of iteration and set the tolerance levels. If the user does not define any of the components, default values will be used. For more details, please refer to the user's guide.

4. Example

In this section, we provide one more example which uses the module `henv`. We hope the users can get an idea of the similarity and difference in the usage of different modules. The water strider data was analyzed by Su and Cook (2013). It has 30 measurements of eight characteristics for each of the three species of water striders: L. esakii, L. dissortis and L.

rufoscutellatus. In the datafile "waterstrider.mat", `X` contains the two group indicators and `Y` contains the eight characteristics. The coding of the group indicators is a little different from the usual, the first group indicator takes value 1, 0 and -1 for L. esakii, L. dissortis and L. rufoscutellatus, and the second group indicator takes value 1, 0 and -1 for L. dissortis, L. esakii and L. rufoscutellatus. The interest lies in comparison of the three species. First we test if the covariance matrices of the different species are the same. Box's M test (Johnson and Wichern 2007) is implemented in the toolbox for this purpose.

```
load waterstrider.mat
alpha = 0.01;
TestOutput = mtest(X, Y, alpha);
```

```
-------------------------------------------------
     MBox     Chi-sqr.          df          P
-------------------------------------------------
  157.5977   137.3361          72       0.0000
-------------------------------------------------
Covariance matrices are significantly different.
```

The highly significant p-value suggests that the covariance matrices for the three species are different. Therefore, it is not appropriate to use the basic envelope model which assumes constant covariance matrix across the species. Instead we use the heteroscedastic envelope model to fit this data.

```
u1 = modelselectaic(X, Y, 'henv');
u1

u1 =

     6

u2 = modelselectbic(X, Y, 'henv');
u2

u2 =

     4

u3 = modelselectlrt(X, Y, 0.01, 'henv');
u3

u3 =

     6
```

AIC and LRT with significance level 0.01 both yield $u = 6$ while BIC selects $u = 4$. To be more conservative, we fit the heteroscedastic envelope model with $u = 6$.

```
ModelOutput = henv(X, Y, 6);
ModelOutput
```

```
ModelOutput =

          mu: [8x1 double]
         mug: [8x3 double]
        Yfit: [90x8 double]
       Gamma: [8x6 double]
      Gamma0: [8x2 double]
        beta: [8x3 double]
    groupInd: [3x2 double]
       Sigma: [8x8x3 double]
         eta: [6x3 double]
       Omega: [6x6x3 double]
      Omega0: [2x2 double]
    paramNum: 98
           l: 1.0051e+03
   covMatrix: [32x32 double]
       asySE: [8x3 double]
       ratio: [8x3 double]
          ng: [3x1 double]
```

As we are in the context of comparing multivariate mean for different populations, the output list for the heteroscedastic envelope model contains the estimates of the grand mean $\boldsymbol{\mu}$, the group means $\boldsymbol{\mu}_{(i)}$, and the error covariance matrices for each group $\boldsymbol{\Sigma}_{(i)}$. The output list also has the constituent parameters and important statistics just as in the output list of `env`. To get the estimated group mean, we call `ModelOutput.mug`.

```
ModelOutput.mug
```

```
ans =

   -1.1417   -1.1267   -1.0845
   -1.4063   -1.4067   -1.3132
   -1.3314   -1.3336   -1.2152
   -0.3113   -0.1839   -0.1736
    0.4003    0.3847    0.3072
    0.4107    0.3753    0.3735
    0.3467    0.3271    0.3179
   -0.1954   -0.2100   -0.3488
```

If there are p groups, `ModelOutput.mug` will have p columns, each for one group. We can find the corresponding group indicators by calling

```
ModelOutput.groupInd
```

```
ans =

    -1    -1
     0     1
     1     0
```

The ith row in `ModelOutput.groupInd` corresponds to the ith column in `ModelOutput.mug`. For example, the estimated mean vector of the eight characteristics for L. rufoscutellatus is in the first column of `ModelOutput.mug`. To predict a new observation, we input its group indicator. Suppose we want to predict a new observation of L. dissortis.

```
Xnew = [0 1]';
PredictOutput = prediction(ModelOutput, Xnew, 'prediction', 'henv');
[PredictOutput.value, PredictOutput.SE]

ans =

   -1.1267    0.3716
   -1.4067    0.3784
   -1.3336    0.3539
   -0.1839    0.2376
    0.3847    0.4596
    0.3753    0.3519
    0.3271    0.4700
   -0.2100    0.3849
```

The first column gives the predicted value, which is the estimated group mean, and the prediction errors are in the second column.

The usage of other modules is similar, it is just the inputs and outputs of the functions are tailored for different models. For details on the syntax and semantics of the functions, the user can refer to the Reference Manual.

5. Conclusion

The MATLAB toolbox **envlp** implements a variety of envelope models in the context of multivariate linear regression and estimating multivariate means. Complete documentation is provided for each function and a user's guide to the toolbox is also available. Description for all datasets is also included. Scripts are provided to reproduce all published results of these methods. The package is modularized and it is easy for the user to follow the structure of the package if they want to add new methods to the toolbox. Our aim for the future is to extent the package and add more methods to the toolbox as well as providing more inference tools. Updates can be checked on the toolbox website.

Acknowledgement

We are grateful to the editor and two referees for their insightful suggestions and comments that helped us improve the paper. We also thank Guangyu Zhu for his comments on previous

versions of the toolbox. The research in this article is supported in part by National Science Foundation Grants DMS-1007547 and SES-1156026.

References

Conway JB (1990). *A Course in Functional Analysis*. New York: Springer-Verlag.

Cook RD (2012). "Lecture Notes on Dimension Reduction." School of Statistics, University of Minnesota, Minneapolis.

Cook RD, Forzani L, Tomassi D (2009). "**LDR**: A Package for Likelihood-based Sufficient Dimension Reduction." *Journal of Statistical Software*, **39**, 1–20.

Cook RD, Helland I, Su Z (2013). "Envelopes and Partial Least Squares Regression." *Journal of the Royal Statistical Society B*, **75**, 851–877.

Cook RD, Li B, Chiaromonte F (2010). "Envelope Models for Parsimonious and Efficient Multivariate Linear Regression (With Discussion)." *Statistica Sinica*, **20**, 927–1010.

Cook RD, Su Z (2013). "Scaled Envelopes: Scale Invariant and Efficient Estimation in Multivariate Linear Regression." *Biometrika*, **100**(4), 939–954.

de Jong S (1993). "SIMPLS: An Alternative Approach to Partial Least Squares Regression." *Chemometrics and Intelligent Laboratory Systems*, **18**(3), 251–263.

Diaconis P, Freedman D (1984). "Asymptotics of Graphical Projection Pursuit." *The Annals of Statistics*, pp. 793–815.

James W, Stein C (1961). "Estimation with Quadratic Loss." In *Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pp. 361–379.

Johnson RA, Wichern DW (2007). *Applied Multivariate Statistical Analysis*. Upper Saddle River, NJ: Prentice Hall.

Lippert R (2004). ***sg_min**: Stiefel Grassmann Optimization*. MATLAB package version 2.4.3. URL http://web.mit.edu/~ripper/www/sgmin.html.

Strang G (2000). *Tcodes - MATLAB Teaching Codes*. Massachusetts Institute of Technology, Cambridge, Massachusetts. URL http://web.mit.edu/18.06/www/Course-Info/Tcodes.html.

Su Z, Cook RD (2011). "Partial Envelopes for Efficient Estimation in Multivariate Linear Regression." *Biometrika*, **98**(1), 133–146.

Su Z, Cook RD (2012). "Inner Envelopes: Efficient Estimation in Multivariate Linear Regression." *Biometrika*, **99**(3), 687–702.

Su Z, Cook RD (2013). "Estimation of Multivariate Means with Heteroscedastic Errors Using Envelope Models." *Statistica Sinica*, **23**, 213–230.

The MathWorks, Inc (2012a). ***Statistics*** *Toolbox - MATLAB: Perform Statistical Modeling and Analysis*. The MathWorks, Inc., Natick, Massachusetts. URL http://www.mathworks.com/products/statistics/.

The MathWorks, Inc (2012b). *MATLAB - The Language of Technical Computing, Version 8.0*. The MathWorks, Inc., Natick, Massachusetts. URL http://www.mathworks.com/products/matlab/.

Trujillo-Ortiz A, Hernandez-Walls R (2002). ***MBoxtest****: Multivariate Statistical Testing for the Homogeneity of Covariance Matrices by the Box's M*. A MATLAB file. URL http://www.mathworks.com/matlabcentral/fileexchange/2733.

Affiliation:

Dennis Cook
School of Statistics
University of Minnesota
E-mail: dennis@stat.umn.edu
URL: http://users.stat.umn.edu/~rdcook

Zhihua Su
Department of Statistics
University of Florida
E-mail: zhihuasu@stat.ufl.edu
URL: http://www.stat.ufl.edu/~zhihuasu

Yi Yang
School of Statistics
University of Minnesota
E-mail: yiyang@umn.edu
URL: http://users.stat.umn.edu/~yiyang

Journal of Statistical Software http://www.jstatsoft.org/
published by the American Statistical Association http://www.amstat.org/
Volume VV, Issue II *Submitted:* yyyy-mm-dd
MMMMMM YYYY *Accepted:* yyyy-mm-dd