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

LDR a Package for Likelihood-Based Sufficient Dimension Reduction

Dennis Cook
University of Minnesota

Liliana Forzani
IMAL

Diego Tomassi
Universidad Nacional del Litoral

Abstract

We introduce a software package running under Matlab that implements several recently proposed likelihood-based methods for sufficient dimension reduction. Current capabilities include estimation of reduced subspaces with a fixed dimension d, as well as estimation of d by use of likelihood-ratio testing, permutation testing and information criteria. The methods are suitable for preprocessing data for both regression and classification. Implementations of related estimators are also available. Although the software is more oriented to command-line operation, a graphical user interface is also provided for prototype computations.

Keywords: Key words and phases: Dimension Reduction, Inverse Regression, Principal Components .

1. Introduction

Since the introduction of sliced inverse regression (SIR; Li 1991) and sliced average variance estimation (SAVE; Cook and Weisberg 1991) there has been considerable interest in dimension reduction methods for the regression of a real response Y on a random vector $\mathbf{X} \in \mathbb{R}^p$ of predictors. A common goal of SIR, SAVE and many other dimension reduction methods is to estimate the central subspace $\mathcal{S}_{Y|\mathbf{X}}$ (Cook 1994, 1998), which is defined as the intersection of all subspaces $\mathcal{S} \subseteq \mathbb{R}^p$ with the property that $Y \perp\!\!\!\perp \mathbf{X}|P_{\mathcal{S}}\mathbf{X}$. Informally, these methods pursue the estimation of the fewest linear combinations of the predictors that contain all the regression information on the response. SIR uses a sample version of the first conditional moment $\mathsf{E}(\mathbf{X}|Y)$ to construct such an estimator, while SAVE uses sample first and second $\mathsf{E}(\mathbf{X}\mathbf{X}^T|Y)$ conditional moments. Although SIR and SAVE have found wide-spread use in applications, both have well known limitations. In particular, the subspace $\mathcal{S}_{\text{sir}}$ estimated by SIR is typically a proper subset of $\mathcal{S}_{Y|\mathbf{X}}$ when the response surface is symmetric about the

origin. SAVE was developed in response to this limitation and provides exhaustive estimation of $\mathcal{S}_{Y|\mathbf{X}}$ under mild conditions (Li and Wang 2007; Shao, Cook and Weisberg 2007), but its ability to detect linear trends is generally inferior to SIR's. For these reasons, SIR and SAVE have been used as complementary methods, with satisfactory results often being obtained by informally combining their estimated directions (see, for example, Bura and Pfeiffer 2003; Cook 2003; Ye and Weiss 2003; Li and Li 2004; Zhu, Othaki and Li, 2005; Pardoe, Yin and Cook 2007). Another method called directional regression (DR) was recently proposed by Li and Wang (2007) and it was shown to provide exhaustive estimation of $\mathcal{S}_{Y|\mathbf{X}}$ under mild conditions. In a series of recent articles, Cook (2007) and Cook and Forzani (2008, 2009a-b) took a substantial step forward in the development of dimension reduction methods based on the first two conditional moments. The idea behind this new methodology is to estimate $\mathcal{S}_{Y|\mathbf{X}}$ via maximum likelihood. Likelihood-based methods provide exhaustive estimation of $\mathcal{S}_{Y|\mathbf{X}}$ under the same conditions as SIR, DR and SAVE. Unlike those methods, however, a likelihood-based objective function is employed to acquire the reduced dimensions, giving these methods $\sqrt{n}$ consistency and asymptotical efficiency, which is not a claimed attribute for any previous method. The dimension d of $\mathcal{S}_{Y|\mathbf{X}}$ can be estimated using likelihood ratio testing or an information criterion like AIC or BIC, and conditional independence hypotheses involving the predictors can be tested straightforwardly. Furthermore, it was demonstrated both theoretically and with simulations that these methods have good robustness properties, even under substantial deviations of the error distribution from its nominal specification.

The goal of this article is to introduce software tools for these recently proposed likelihood-based methods for sufficient dimension reduction. We have called the package LDR (standing for Likelihood-based Dimension Reduction) and it runs under Matlab, which allows use of computational tools developed outside of the statistics community. In particular, likelihood maximization under some of the models discussed below requires optimization on a Grassman manifold $\mathcal{G}_{(d,p)}$, which is the set of all d dimensional subspaces of $\mathbb{R}^p$ (Edelman, Arias and Smith 1999). Optimization on Grassmann manifolds is fairly common in some areas such as computer science and signal processing, but is relatively rare in statistics. The software is oriented to command-line operation and it is intended to provide an easy-to-use interface similar to Weisberg's DR package for R (`http://cran.us.r-project.org/web/packages/dr`). Fixed-dimension as well as dimension selection tasks are supported, both for continuous and for discrete responses. Functions to compute other estimators such as SIR, SAVE and DR are also provided, although methods for dimension estimation are not yet available for these methods.

The paper is organized as follows. In Section 2 we briefly review some theory concerning likelihood-based sufficient dimension reduction. In Section 3, we describe the main features of the software and then give some examples of how to use it in Section 4. Finally, we briefly describe how to use the graphical user interface in Section 4.3. The package and the complete software documentation is available at `http://liliana.forzani.googlepages.com/ldr-package`.

2. Likelihood-based sufficient dimension reduction

The likelihood-based approach to dimension reduction uses model-based inverse regression of $\mathbf{X}$ on Y to gain reductive information for the forward regression of Y on $\mathbf{X}$. We assume that the data consist of n independent observations on $(\mathbf{X}, Y)$, with $\mathbf{X} \in \mathbb{R}^p$, $Y \in \mathbb{R}$, $n > p$ and

that $\mathbf{X}|Y \sim N(\boldsymbol{\mu}_y, \boldsymbol{\Delta}_y)$. Then we can write

$$\mathbf{X} = \boldsymbol{\mu}_y + \boldsymbol{\varepsilon}, \tag{1}$$

where $\boldsymbol{\varepsilon} \sim N(0, \boldsymbol{\Delta}_y)$, $\boldsymbol{\Delta}_y > 0$. The goal is to find the maximum likelihood estimator for the central subspace, $\mathcal{S}_{Y|\mathbf{X}}$. Let $d = \dim(\mathcal{S}_{Y|\mathbf{X}})$ and let $\boldsymbol{\alpha}$ denote a $p \times d$ semi-orthogonal matrix whose columns form a basis for $\mathcal{S}_{Y|\mathbf{X}}$, so $Y|\mathbf{X} \sim Y|\boldsymbol{\alpha}^T\mathbf{X}$. Likelihood-based methods are designed to estimate $\mathcal{S}_{Y|\mathbf{X}} = \text{span}(\boldsymbol{\alpha})$ under different structures for the mean $\boldsymbol{\mu}_y$ and the covariance $\boldsymbol{\Delta}_y$ functions in model (1). These methods are summarized in the sections that follow. In preparation, let $\mathcal{M} = \text{span}\{\boldsymbol{\mu}_y - \boldsymbol{\mu} | y \in S_Y\} \subseteq \mathbb{R}^p$, where S_Y denotes the sample space of Y, and let $\hat{a}$ stand for the maximum likelihood estimator of a parameter a.

2.1. Principal Component, PC.

In this model, which was introduced by Cook (2007), it is assumed that the errors are isotonic, $\boldsymbol{\Delta}_y = \sigma^2\mathbf{I}_p$. As a consequence of this structure, $\mathcal{S}_{Y|\mathbf{X}} = \mathcal{M}$ and therefore we have the representation $\boldsymbol{\mu}_y = \boldsymbol{\mu} + \boldsymbol{\alpha}\boldsymbol{\nu}_y$, where $\boldsymbol{\nu}_y \in \mathbb{R}^d$ is unknown for all $y \in S_y$. Under this setting, the maximum likelihood estimator of $\mathcal{S}_{Y|\mathbf{X}}$ is the span of the first d eigenvectors of the sample covariance matrix $\tilde{\boldsymbol{\Sigma}}$ of $\mathbf{X}$.

2.2. Isotonic Principal Fitted Components, IPFC.

As in the PC model, it is assumed that the errors are isotonic. Consequently $\mathcal{S}_{Y|\mathbf{X}} = \mathcal{M}$ and we can again represent the conditional means as $\boldsymbol{\mu}_y = \boldsymbol{\mu} + \boldsymbol{\alpha}\boldsymbol{\nu}_y$. However, following Cook (2007), the coordinate vectors $\boldsymbol{\nu}_y$ are now modeled as

$$\boldsymbol{\nu}_y = \boldsymbol{\beta}\{\mathbf{f}_y - \mathsf{E}(\mathbf{f}_Y)\}, \tag{2}$$

where $\mathbf{f}_y \in \mathbb{R}^r$ is a known vector-valued function of y with linearly independent elements and $\boldsymbol{\beta} \in \mathbb{R}^{d\times r}$, $d \leq \min(r, p)$, is an unrestricted rank d matrix. For this model the maximum likelihood estimator of $\mathcal{S}_{Y|\mathbf{X}}$ is the first d eigenvectors of $\tilde{\boldsymbol{\Sigma}}_{\text{fit}}$, where $\tilde{\boldsymbol{\Sigma}}_{\text{fit}}$ is the sample covariance matrix of the fitted vectors from the multivariate linear regression of $\mathbf{X}_y$ on $\mathbf{f}_y$, including an intercept. When Y is discrete or slicing is used to approximate the coordinates of $\boldsymbol{\nu}_y$ with step functions,

$$\tilde{\boldsymbol{\Sigma}}_{\text{fit}} = \sum_{y=1}^{h} \frac{n_y}{n}(\bar{\mathbf{X}}_y - \bar{\mathbf{X}})(\bar{\mathbf{X}}_y - \bar{\mathbf{X}})^T,$$

where $\bar{\mathbf{X}}_y$ the average predictor vector in slice y, $\bar{\mathbf{X}}$ the overall average, and h is the number of slices.

2.3. Structured Principal Fitted Components, SPFC.

This model was introduced by Cook and Forzani (2009-a). The coordinate vectors $\boldsymbol{\nu}_y$ are again modeled as in (2), but now a linear structure is used to model $\boldsymbol{\Delta} = \sum_{i=1}^m \delta_i\mathbf{G}_i$, where $m \leq p(p+1)/2$, $\mathbf{G}_1, \ldots, \mathbf{G}_m$ are known real symmetric $p \times p$ linearly independent matrices and the elements of $\boldsymbol{\delta} = (\delta_1, \ldots, \delta_m)^T$ are functionally independent. It is required also that $\boldsymbol{\Delta}^{-1}$ have the same linear structure as $\boldsymbol{\Delta}$: $\boldsymbol{\Delta}^{-1} = \sum_{i=1}^m s_i\mathbf{G}_i$. To model a diagonal $\boldsymbol{\Delta}$ we set $\mathbf{G}_i = \mathbf{e}_i\mathbf{e}_i^T$, where $\mathbf{e}_i \in \mathbb{R}^p$ contains a 1 in the i-th position and zeros elsewhere. This

basic structure can be modified straightforwardly to allow for a diagonal $\mathbf{\Delta}$ with sets of equal diagonal elements, and for a non-diagonal $\mathbf{\Delta}$ with equal off-diagonal entries and equal diagonal entries. In the latter case, there are only two matrices $\mathbf{G}_1 = \mathbf{I}_p$ and $\mathbf{G}_2 = \mathbf{e}\mathbf{e}^T$, where $\mathbf{e} \in \mathbb{R}^p$ has all elements equal to 1.

In this case the central subspace $\mathcal{S}_{Y|\mathbf{X}} = \mathbf{\Delta}^{-1}\mathcal{M}$ and then $\boldsymbol{\mu}_y = \boldsymbol{\mu} + \mathbf{\Delta}\boldsymbol{\alpha}\boldsymbol{\beta}\mathbf{f}_y$. The maximum likelihood estimator for $\mathcal{S}_{Y|\mathbf{X}}$ is the span of the first d eigenvectors of $\widehat{\mathbf{\Delta}}^{-1}\tilde{\mathbf{\Sigma}}$ where $\widehat{\mathbf{\Delta}}$ indicates the MLE of $\mathbf{\Delta}$. The MLE $\widehat{\mathbf{\Delta}}$ is relatively complicated and an iterative algorithm is evidently required for its computation

2.4. Extended Principal Fitted Components, EPFC.

In this version the coordinate vectors $\boldsymbol{\nu}_y$ are again modeled like (2) so that $\boldsymbol{\mu}_y = \boldsymbol{\mu} + \boldsymbol{\alpha}\boldsymbol{\beta}\mathbf{f}_y$, but now $\mathbf{\Delta}$ is modeled as $\mathbf{\Delta} = \boldsymbol{\alpha}\mathbf{\Omega}\boldsymbol{\alpha}^T + \boldsymbol{\alpha}_0\mathbf{\Omega}_0\boldsymbol{\alpha}_0^T$, where $\mathbf{\Omega} \in \mathbb{R}^{d\times d}$ and $\mathbf{\Omega}_0 \in \mathbb{R}^{p-d,p-d}$ are positive definite matrices and still $\mathcal{S}_{Y|\mathbf{X}} = \text{span}(\boldsymbol{\alpha})$. Then the maximum likelihood estimator for $\mathcal{S}_{Y|\mathbf{X}}$ maximizes over $\mathcal{S}(\boldsymbol{\alpha}) \in \mathcal{G}_{(d,p)}$ the log likelihood function

$$F(\boldsymbol{\alpha}) = -\frac{np}{2}(1 + \log(2\pi)) - \frac{n}{2}\log|\tilde{\mathbf{\Sigma}}| - \frac{n}{2}\log|\boldsymbol{\alpha}^T\tilde{\mathbf{\Sigma}}^{-1}\boldsymbol{\alpha}| - \frac{n}{2}\log|\boldsymbol{\alpha}^T\tilde{\mathbf{\Sigma}}_{\text{res}}\boldsymbol{\alpha}|,$$

where $\tilde{\mathbf{\Sigma}}_{\text{res}} = \tilde{\mathbf{\Sigma}} - \tilde{\mathbf{\Sigma}}_{\text{fit}}$ with $\tilde{\mathbf{\Sigma}}_{\text{fit}}$ defined in the IPFC model. See Cook (2007) for further discussion.

2.5. Principal Fitted Components, PFC.

In this model, which was studied by Cook and Forzani (2009-a) the coordinate vectors $\boldsymbol{\nu}_y$ are again modeled as in (2), but $\mathbf{\Delta}$ is now an unstructured positive definite matrix that is functionally independent of y. The central subspace is again $\mathcal{S}_{Y|\mathbf{X}} = \mathbf{\Delta}^{-1}\mathcal{M}$ and the MLE of $\mathcal{S}_{Y|\mathbf{X}}$ is the span of $\tilde{\mathbf{\Sigma}}_{\text{res}}^{-12}\mathbf{v}$, with $\mathbf{v}$ the first d eigenvector of $\tilde{\mathbf{\Sigma}}_{\text{res}}^{-1/2}\tilde{\mathbf{\Sigma}}\tilde{\mathbf{\Sigma}}_{\text{res}}^{-1/2}$, with $\tilde{\mathbf{\Sigma}}_{\text{res}}$ defined as in the EPFC model.

2.6. Covariance Reduction, CORE.

Covariance reduction addresses a rather different dimension reduction issue. Consider the problem of characterizing the covariance matrices $\mathbf{\Delta}_y$, $y = 1, \ldots, h$, of a random vector $\mathbf{X}$ observed in each of h normal populations. There is no interest in the population means $\boldsymbol{\mu}_y$. The methodology reviewed here, which was proposed by Cook and Forzani (2008), is based on reducing the sample covariance matrices to an informational core that is sufficient to characterize the variance heterogeneity across the h populations. It may be a useful alternative to the spectral modeling in many applications.

For this problem the central subspace, $\mathcal{S}_{Y|\mathbf{X}} = \text{span}(\boldsymbol{\alpha})$ is redefined to be the smallest subspace satisfying $\mathbf{\Delta}_y = \mathbf{\Delta} + P^T_{\boldsymbol{\alpha}(\mathbf{\Delta}_y)}(\mathbf{\Delta}_y - \mathbf{\Delta})P_{\boldsymbol{\alpha}(\mathbf{\Delta}_y)}$, where $P_{\boldsymbol{\alpha}(\mathbf{\Delta}_y)} = \boldsymbol{\alpha}(\boldsymbol{\alpha}^T\mathbf{\Delta}_y\boldsymbol{\alpha})^{-1}\boldsymbol{\alpha}^T\mathbf{\Delta}_y$ the projection onto $\mathcal{S}_{Y|\mathbf{X}}$ using the inner product defined by $\mathbf{\Delta}_y$. We assume that the data consist of n_y independent observations of $\mathbf{X}_y$, $y = 1, \ldots, h$. Then the maximum likelihood estimator of $\mathcal{S}_{Y|\mathbf{X}}$ is the subspace span$(\boldsymbol{\alpha})$ that maximizes over $\mathcal{G}_{(d,p)}$ the log likelihood function

$$L_d(\boldsymbol{\alpha}) = c - \frac{n}{2}\log|\tilde{\mathbf{\Delta}}| + \frac{n}{2}\log|\boldsymbol{\alpha}^T\tilde{\mathbf{\Delta}}\boldsymbol{\alpha}| - \sum_{y=1}^{h}\frac{n_y}{2}\log|\boldsymbol{\alpha}^T\tilde{\mathbf{\Delta}}_y\boldsymbol{\alpha}|, \tag{3}$$

where c is a constant depending only on p, n_y and $\tilde{\boldsymbol{\Delta}}_y$, $y = 1, \ldots, h$ denote the sample covariance matrix from population y computed with divisor n_y, and $\tilde{\boldsymbol{\Delta}} = \frac{1}{n}\sum_{y=1}^{h} n_y \tilde{\boldsymbol{\Delta}}_y$.

2.7. Likelihood Acquired Directions, LAD.

In this model, which was proposed by Cook and Forzani (2009-b), we consider both a general conditional covariance $\boldsymbol{\Delta}_y$ and a general mean $\boldsymbol{\mu}_y$. It requires a discrete response Y. When the response is continuous it is typical to follow Li (1991) and replace it with a discrete version constructed by partitioning its range into h slices. The central subspace $\mathcal{S}_{Y|\mathbf{X}} = \text{span}(\boldsymbol{\alpha})$ is the smallest subspace that satisfies the conditions (i) $\boldsymbol{\Delta}_y = \boldsymbol{\Delta} + P^T_{\boldsymbol{\alpha}(\boldsymbol{\Delta}_y)}(\boldsymbol{\Delta}_y - \boldsymbol{\Delta})P_{\boldsymbol{\alpha}(\boldsymbol{\Delta}_y)}$ and (ii) $\text{span}(\boldsymbol{\alpha}) \subset \boldsymbol{\Delta}^{-1}\text{span}(\boldsymbol{\mu}_y - \boldsymbol{\mu})$. Condition (i) is the same as that encountered in the CORE model, while condition (ii) incorporates the conditional means $\boldsymbol{\mu}_y$. Assume as before that the data consist of n_y independent observations on $\mathbf{X}_y$, $y = 1, \ldots, h$. Then the MLE for $\mathcal{S}_{Y|\mathbf{X}}$ maximizes over $\text{span}(\boldsymbol{\alpha}) \in \mathcal{G}_{(d,p)}$ the log likelihood function

$$L_d(\boldsymbol{\alpha}) = -\frac{np}{2}(1 + \log(2\pi)) - \frac{n}{2}\log|\tilde{\boldsymbol{\Sigma}}| + \frac{n}{2}\log|\boldsymbol{\alpha}\tilde{\boldsymbol{\Sigma}}\boldsymbol{\alpha}| - \frac{1}{2}\sum_{y=1}^{h} n_y \log|\boldsymbol{\alpha}\tilde{\boldsymbol{\Delta}}_y\boldsymbol{\alpha}| \tag{4}$$

where $\tilde{\boldsymbol{\Delta}}_y$ is as defined in the CORE model.

2.8. Envelope Models for Multivariate Linear Regression, MLM

The dimension reduction methodology described in this section is different from the previous methods because it involves a multivariate response $\mathbf{Y} \in \mathbb{R}^r$ and a multivariate predictor $\mathbf{X} \in \mathbb{R}^p$, assumed to be non-stochastic, that are related through the standard normal multivariate linear model:

$$\mathbf{Y} = \boldsymbol{\beta}_0 + \boldsymbol{\beta}\mathbf{X} + \boldsymbol{\varepsilon}, \tag{5}$$

where $\boldsymbol{\beta}_0 \in \mathbb{R}^r$, $\boldsymbol{\beta} \in \mathbb{R}^{r\times p}$ and $\boldsymbol{\varepsilon} \sim N_r(0, \boldsymbol{\Sigma})$. The number of responses r is often large in modern applications of this model. In such situations the response vector may contain irrelevant linear combinations whose conditional distribution does not change with $\mathbf{X}$. It may also contain redundant information characterized by $\boldsymbol{\Sigma}$ having relatively large eigenvectors. Developed recently by Cook, Li and Chiaromonte (2009), envelope models allow for the possibility that such irrelevant or redundant information may be present in $\mathbf{Y}$, and can yield maximum likelihood estimators of $\boldsymbol{\beta}$ with substantially smaller variation that the usual maximum likelihood estimators.

The redundant and irrelevant information in (5) can be characterized by using the reducing subspaces of $\boldsymbol{\Sigma}$. Let $(\boldsymbol{\Gamma}, \boldsymbol{\Gamma}_0) \in \mathbb{R}^{r\times r}$ be an orthogonal matrix, where the columns of $\boldsymbol{\Gamma} \in \mathbb{R}^{r\times d}$ form a basis for the smallest reducing subspace of $\boldsymbol{\Sigma}$ that contains $\text{span}(\boldsymbol{\beta})$. Then the parameters in model (5) have the structure $\boldsymbol{\beta} = \boldsymbol{\Gamma}\boldsymbol{\eta}$, and $\boldsymbol{\Sigma} = \boldsymbol{\Gamma}\boldsymbol{\Omega}\boldsymbol{\Gamma}^T + \boldsymbol{\Gamma}_0\boldsymbol{\Omega}_0\boldsymbol{\Gamma}_0^T$, where $\boldsymbol{\eta} \in \mathbb{R}^{d\times r}$ is an unconstrained coordinate matrix, and $\boldsymbol{\Omega} \in \mathbb{R}^{d\times d}$ and $\boldsymbol{\Omega}_0 \in \mathbb{R}^{r-d\times r-d}$ are positive definite matrices. The theory developed by Cook, Li and Chiaromonte (2009) shows that substantial gains in efficiency are possible if the linear combinations $\boldsymbol{\Gamma}_0^T\mathbf{Y}$ contain both the irrelevant and redundant information in $\mathbf{Y}$.

Fitting with this parameter structure requires estimation of $\boldsymbol{\beta}_0$, $\text{span}(\boldsymbol{\Gamma})$, $\boldsymbol{\eta}$, $\boldsymbol{\Omega}$ and $\boldsymbol{\Omega}_0$. The partially maximized log likelihood for estimating $\text{span}(\boldsymbol{\Gamma})$ is similar to the objective function for extended principal fitted components described in Section 2.4.

3. Software overview

The software is organized to allow users to run different types of processing by calling just a few interface functions. Several auxiliary tools are then internally called to perform computations according to the specified options on those interface functions. As the main example, a single function called `ldr` manages all methods for maximum likelihood estimation of the central subspace. Arguments in this function specify the model to use, the response type – continuous or discrete – and the protocol for the dimension d. This protocol allows d to be specified or estimated using an information criterion, a likelihood-ratio test or in some case a permutation test. Separate functions `mlm` are used for the envelope methodology discussed in Section 2.8. In Section 4 we describe how to use the software in detail.

The package is designed to run mainly from the command-line, but we have also built a simple graphical user interface to make usage more intuitive and analysis more interactive. Although the graphical interface does not currently provide all the features available from the command-based interface, it is powerfull enough to run prototype computations using standard values for some parameters of the models. A brief tutorial for using this interface is given in Section 4.3.

Our implementation emphasizes easy-reading code over performance. It is mainly based upon built-in functions in Matlab and has been tested for compatibility starting up from version 6.5 Release 13. Nevertheless, some functions require the Statistics Toolbox too. The LDR package relies on Lippert's sg-min toolkit for gradient-based optimization over Stiefel-Grassmann manifolds (Lippert and Edelman 2000). This toolkit is freely available at `http://www-math.mit.edu/~lippert/sgmin.html`. Some minor modifications of sg-min were necessary for this application and this modified version of the sg-min is available with our package. The modifications are described in the documentation that comes with the code.

3.1. Available methods

The LDR package offers an implementation of the models described in Section 2. SIR, SAVE and DR are incorporated as separate Matlab functions with the same names. SIR and SAVE are available in Weisberg's DR package, and DR is available from its authors (Li and Wang 2007). We have included basic implementations here in order to provide a more self-contained tool for dimension reduction. We have also used a separate Matlab function `PC` for the PC model, since this model gives the same solution as the usual principal components and we do not have a novel way to estimate d.

For the rest of the models, which are the primary focus of the LDR package, we have incorporated methods to estimate the central subspace, including estimation of its dimension d. Available procedures for choosing d include information criteria such as AIC and BIC (Burnham and Anderson 2002), likelihood-ratio testing, and permutation testing (Cook and Yin 2001). Table 1 shows the available testing methods for each model. All of these functions run both for continuous and discrete responses and share the common interface function `ldr`.

3.2. Managing data, results and plots

Interface functions like `ldr` have workspace variables as arguments and do not allow managing data files directly. Despite several procedures for reading data being already available in Matlab, they usually do not provide support for data files with headers. We have included

Table 1: Available models and testing criteria for d.

Model	AIC	BIC	LRT	PERM
IPFC	✓	✓	✓	
SPFC	✓	✓	✓	
EPFC	✓	✓	✓	
PFC	✓	✓	✓	
LAD	✓	✓	✓	✓
CORE	✓	✓	✓	✓

functions `loadDATA` and `getDATA` to allow for this. The former returns a data matrix, the text in the header and the labels of the variables if all of them exist. On the other hand, `getDATA` returns the response vector and the predictors directly, but does not return the header. In both of these procedures, data must be numeric and organized with rows as cases and columns as variables, with columns separated by white space. If another delimiter is used to separate columns, you can specify it as an optional input.

The header can be just a line of labels, one for each column and with the same delimiter, or it can have more text describing the data. In either case, the header and variable names should be separated by a line feed. For illustration, assume that `datafile.txt` consists of the following semi-colon delimited data:

```
This is the header for seven variables
var1 var2 var3 var4 var5 var6 var7
1; 2; 3; 4; 5; 6; 7;
8; 9; 10; 11; 12; 13; 14;
```

The command `loadDATA` can be used to read this data file:

```
[data,header,labels] = loadDATA('datafile.txt',';');
```

The semi-colon at the end of the last line is the Matlab command to supress printing. To read the same file with white space used as the delimiter use the command

```
[data,header,labels] = loadDATA('datafile.txt');
```

If the response Y is in the first column of matrix data and all other columns are predictors $\mathbf{X}$, then we can create these data arrays in our workspace with the commands

```
Y = data(:,1);
X = data(:,2:end);
```

We can get the same result by typing:

```
[Y,X] = getDATA('datafile.txt',';');
```

If the response fills the seventh column in the data file we create Y by typing

```
[Y,X] = getDATA('datafile.txt',';',7);
```

Table 2: Additional/optional inputs for available models

Model	Input	Description	Default value
IPFC	`'fy'`	Basis for regression. A matrix must be given after this output.	Polynomial basis of degree $r = \max\{3, d+1\}$.
	`'alpha'`	Confidence level for likelihood ratio test.	0.05
SPFC	`'fy'`	Basis for regression. A matrix must be given after this output.	Polynomial basis of degree $r = \max\{3, d+1\}$.
	`'alpha'`	Confidence level for likelihood ratio test.	0.05
PFC	`'fy'`	Basis for regression. A matrix must be given after this output.	Polynomial basis of degree $r = \max\{3, d+1\}$.
	`'alpha'`	Confidence level for likelihood ratio test.	0.05
EPFC	`'fy'`	Basis for regression. A matrix must be given after this output.	Polynomial basis of degree $r = \max\{3, d+1\}$.
	`'alpha'`	Confidence level for likelihood ratio test.	0.05
	`'initval'`	Starting basis for central subspace estimation.	See text for details.
	`'maxiter'`	Maximum number of iterations.	`10000`
	see **sg-min**	Several optional inputs for optimization.	`'prcg','euclidean'`
LAD	`'nslices'`	Number of slices for continuous response preprocessing.	5 slices.
	`'alpha'`	Confidence level for likelihood ratio test and permutation test.	0.05
	`'npermute'`	Permutations for permutation test.	500 permutations.
	`'initval'`	Starting basis for central subspace estimation.	See text for details.
	`'maxiter'`	Maximum number of iterations.	`10000`
	see **sg-min**	Several optional inputs for optimization.	`'prcg','euclidean'`
CORE	`'nslices'`	Number of slices for continuous response preprocessing.	5 slices.
	`'alpha'`	Confidence level for likelihood ratio test and permutation test.	0.05
	`'npermute'`	Permutations for permutation test.	500 permutations.
	`'initval'`	Starting basis for central subspace estimation.	See text for details.
	`'maxiter'`	Maximum number of iterations.	`10000`
	see **sg-min**	Several optional inputs for optimization.	`'prcg','euclidean'`

If we want just the first five columns as predictors we should type:

```
[Y,X] = getDATA('datafile.txt',';',7,1:5);
```

The graphical user interface also allows for using any column in a data array as the response vector and any number of the remaining columns as the predictors. Nevertheless, for full compatibility with the software, the response should be the first column of the data matrix with the predictors occupying the remaining columns.

While saving results in text files is a straightforward operation in Matlab, we have included a function `saveastxt` to make it even easier for those who are not familiar with the language. As it is usual in Matlab, help for this or any other function in the LDR package is available with a command of the form `help function-name`. In addition, we have also added some tools for plotting results to supplement those in Matlab. The function `plotDR` allows for an easier labeling of results and automatical selection of the most suitable plot according to the dimension of the reduced subspace and the type of response. A brief overview of these capabilities is given in Section 4. More detailed description is available through the software documentation.

4. Using the software

4.1. Methods for estimating the central subspace

Except for envelope models, a single function called `ldr` provides a unified interface for all dimension reduction methods based on maximum likelihood estimation of the central subspace. Other methods such as SIR, SAVE, DR and PC are called from separate interface functions. Their usage is very similar and is discussed briefly in this section.

To start using the software, set the current directory to the package's location in your disk and type `setpaths`. This command adds all the directories in the package to the Matlab path, so that all functions there become available for computation.

Five arguments are mandatory when calling the `ldr` function and they must be provided in the order that follows: i) the response vector Y; ii) the matrix of predictors X; iii) an identifier for the model to be used, such as `'PFC'` or `'epfc'`; iv) the type of response, which may be `'cont'` or `'disc'` depending on whether Y is continuous or discrete; and v) the dimension d of the central subspace or an identifier such as `'BIC'` for an estimation method along with any necessary argument. Acronyms previously introduced in Section 2 have been used to identify the available models. Identifiers for estimation methods are `'AIC'` for Akaike's information criterion, `'BIC'` for Bayes information criterion, `'LRT'` for likelihood-ratio test and `'PERM'` for permutation test. Matching is case-insensitive, so you can type either `'LRT'` or `'lrt'`. The identifiers `'LRT'` and `'PERM'` should be followed by a test level, typically .01 or .05.

Outputs are the same for all models and they are: i) the projection of the predictors onto the estimated central subspace; ii) a matrix whose columns are the estimated basis vectors for the central subspace; iii) the maximum value of the log likelihood; and the iv) the estimated dimension in case a criterion for d was used.

Depending on the model, additional and optional arguments can be given. As an example, to slice a continuous response into ten slices for processing data under the LAD model, an additional string `'nslices'` and the value `10` should be added to the arguments. In this case, a call to the function `ldr` should look like:

```
[WX,W,L,d] = ldr(Y,X,'LAD','cont','aic','nslices',10);
```

where `WX` is the projection of the predictors onto the estimated central subspace, `W` is a matrix whose columns are the estimated basis vectors for the central subspace, `L` the maximum value of the log likelihood and `d` is the dimension of the central subspace estimated by using Akaike's information criterion.

Methods such as EPFC, LAD and CORE require a starting basis for the central subspace prior to iteration. We determine the starting basis internally by searching over several estimators like SIR, SAVE, DR, PLS and PC and selecting the one that gives the largest value of the likelihood. Nevertheless, in some cases an external starting basis may be useful. A starting basis can be supplied by adding the text string `'initval'` followed by a matrix whose column form a starting basis. Iterative estimation of the central subspace is carried out using the sg-min package. All the available parameters in this toolkit have been retained as optional inputs in LDR. In addition, you can set the maximum number of iterations to be done. By default, computation will stop after 10000 iterations if convergence has not been achieved. Another optional argument available for all methods is `'-v'`, which enables a verbose mode

to get progress information of the running process. This input should always be given as the last one. The previous call to LAD in verbose mode with an initial basis $\mathbf{B} \in \mathbb{R}^{p\times d}$ for the central subspace and with no more than 1000 iterations would look like

```
[WX,W,f,d] = ldr(Y,X,'LAD','cont','aic','nslices',10,'initval',B,'maxiter',1000,'-v').
```

A list of the remaining optional arguments and their identifiers is given in Table 2.

To further illustrate how to use the package, consider a script to study the identification of hand-written digits $\{0, 1, \ldots, 9\}$. The 44 subjects were asked to write 250 random digits. Each digit yields a 16-dimensional feature vector, consisting of 8 pairs of randomly sampled two-dimensional locations on the digit. The 44 subjects were divided into two groups of size 30 and 14, in which the first formed the training set with sample size $7,494$ and the second formed the test set with sample size $3,498$. The data set is available from the UCI machine-learning repository at `ftp://ftp.ics.uci.edu/pub/machine-learning-databases/pendigits`. We focus on dimension reduction of the 16-dimensional feature vectors for the training set, which serves as a preparatory step for developing an efficient classifier. For clarity, we first consider the digits 0, 6 and 9. The reduced data set comprises 2,219 cases. Suppose data are stored in the text file `digits.txt` where the first column takes values 1, 2 or 3 indicating if the observation corresponds to a 0, 6 or 9 respectively. Remembering the file convention discussed in Section 3.2, we first load the response vector and the predictors by typing:

```
setpaths; % adds all directories in the LDR package to Matlabs's path.
data = load('digits.txt'); % reads data from text file.
Y = data(:,1);  % assigns the first column of data to the response.
X = data(:,2:end);  % assigns the remaining columns to the predictors.
```

Now we look for a subspace of dimension $d = 2$ using the LAD model:

```
[WX,W] = ldr(Y,X,'LAD','disc',2);
```

Here we used default options for optimization. To plot results, we can call the provided function `plotDR` as:

```
plotDR(Y,WX,'disc','LAD');
```

Here, arguments `'disc'` and `'LAD'` allows for labeling data according to the response and for setting axes labels. The function `plotDR` behaves like standard Matlab plotting functions. For instance, a second call to `plotDR` will overwrite the first plot, unless the Matlab command for a new figure has been issued. To carry out a similar analysis but using the CORE model, we could just replace `'LAD'` with `'CORE'` in all the statements above. When using SIR, SAVE, DR or PC the dimension d should be given, as no testing method for d is currently available for them in the LDR package. If we choose $d = 2$, corresponding commands should look as:

```
[WX,W]=SIR(Y,X,'disc',2);
[WX,W]=SAVE(Y,X,'disc',2);
[WX,W]=DR(Y,X,'disc',2);
```

Processing data with continuous responses is completely analogous, but it often allows for additional arguments. For illustration, let's review a script to study the data from Figure 5 in Cook and Forzani (2009-b). Suppose you have already loaded your continuous data into workspace variables Y and X (the complete script and data are available in the directory `data`). To process under the LAD model, suppose 5 slices are suitable and $d = 1$. Then, use a statement such as

```
ldr(Y,X,'LAD','cont',1,'nslices',5);
```

If we were looking for a similar analysis but wanted instead to estimate d by using likelihood-ratio testing with level 5% we could type:

```
[WX,W,f,d] = ldr(Y,X,'LAD','cont','lrt','nslices',5,'alpha',0.05);
```

Similarly, if we were considering estimating d with a permutation test using 500 permutations with level of 5 %, we would type:

```
[WX,W,f,d] = ldr(Y,X,'LAD','cont','perm','nslices',5,'alpha',0.05,'npermute',500);
```

These arguments are inappropriate for models like IPFC, SPFC, EPFC and PFC. For them, instead, the $\mathbf{f}_y$'s should be given as an $n \times r$ matrix, say FY, with rows $\mathbf{f}_y^T$. In this package we provide an auxiliary function `get_fy` which returns $\mathbf{f}_y$ as a polynomial of order r, where the order of polynomials r to be used for $\mathbf{f}_y$ should be given. Calling for the PFC model, for example, should look like:

```
FY = get_fy(Y,2);  % builds a matrix of polynomials of order 2.
[WX,W,f,d] = ldr(Y,X,'PFC' ,'cont' ,'bic','fy' ,FY);
```

`[WX,W,f,d] = ldr(Y,X,'PFC' ,'cont' ,'bic','fy' ,FY);` where we have told the software to use a second-order polynomial and we are also estimating d using the Bayes information criterion. To reproduce Figure 5 from Cook and Forzani (2009-b) we would then use

```
plotDR(Y,WX,'cont','PFC');
```

Further examples of how to use the software are provided in the `papers` folder within LDR. They reproduce results and figures published when introducing the methods discussed in Section 2 and they can help in getting used to the main features available in the package while testing the methods.

4.2. Envelope models

The basic command for computing an estimated basis G of span($\mathbf{\Gamma}$) in the envelope model (2.8) is `[GX,G,L,dhat] = mlm_fit(Y,X,dim)`, where `dim` stands for the dimension of the envelope. The essential output consists of the estimate `G` of $\mathbf{\Gamma}$, the value `L` of the log likelihood at the MLE's and the estimated dimension `dhat` of the envelope. `GX` is an exploratory quantity and not used routinely in this application. The following options are avaiable for `dim`: If `dim` is set to an interger d, eg., `mlm_fit(Y,X,2)`, then `dhat` $= d$ and the envelope model is fitted with the

specified dimension. If `dim` is set to `'aic'` or `'bic'` then the indicated information criterion AIC or BIC is used to estmimate d. These commands may take a long time, depending on the size of the regression. If `dim` is set to the pair `'lrt',.05`, eg,. `mlm_fit(Y,X,'lrt',.05)`, then likelihood ratio testing at level .05 is used to estimate d.

The following commands are available following the initial fit that produces `G`.

- `[betaem,eta,Omega,Omega0,S1,S2] = mlm_empars(Y,X,G)` returns the estimated parameters from the fit of the envelope model. `betaem = G x eta` is the estimated coefficient matrix `Omega` and `Omega0` are the estimates of $\mathbf{\Omega}$ and $\mathbf{\Omega}_0$, `S1` is the estimate of $\mathbf{\Gamma\Omega\Gamma}^T$ and `S2` is the estimate of $\mathbf{\Gamma}_0\mathbf{\Omega}_0\mathbf{\Gamma}_0^T$.
- `mlm_emses(Y,X,G)` returns the matrix of standard errors of the elements `betaem` from the fit of the envelope model.
- `mlm_seratios(Y,X,G)` returns the matrix of ratios of standard errors of the full model and envelope model estimates of $\boldsymbol{\beta}$. The ratios are (full model se's)/(envelope model se's).

Other commands are documented in the folder `mlm` that comes with the Matlab code.

4.3. Using the graphical user interface

The graphical user interface (GUI) is expected to provide an easy-to-use tool to perform small prototype tasks using the functions available in the package. The GUI starts by typing `demo` in the command window, provided current directory is the LDR folder. Once it has been loaded, data analysis using the GUI typically follows a sequence of steps:

1. Load data by clicking on the LOAD DATA button and then follow the dialog box until finding the desired data file.
2. Specify the type of the response according to the data by choosing the right option in the popup menu. For continuous responses, you should also give the number of slices for the slicing procedure if you plan to apply methods such as LAD, CORE, SIR, SAVE or DR. Similarly, you can use this editable text box to type specific input values for each method. As another example, if you plan to apply models such us PFC, IPFC, SPFC or EPFC, you should type here the order of the polynomials you want to use in order to estimate $\tilde{\mathbf{\Sigma}}_{\text{fit}}$. Only polynomial bases can be specified when using the GUI.
3. Set the dimension reduction method to apply. Available methods are listed in a popup menu.
4. Select the dimension of the reduced subspace or choose a method to test for it. Currently, the GUI allows only $d = 1, 2, 3$. Testing criteria include all those listed in Table 1. Some testing criteria, in particular permutation tests, take a long computation time and can freeze the GUI for awhile.
5. Start the computation by clicking the RUN button. You will then be asked to select the response vector from the data file. Notice you can select only one vector as the response. By default, data corresponding to the first column in the data file will be highlighted.

Clicking on the RUN button again to select the predictors. A default set is highlighted, but you can change selection if you want to use a subset of them for computations. A third click on the button sets the selected predictors and starts processing data.

By default, processing data with the GUI only returns results through plots. Nevertheless, you can check the SHOW RESULTS option to allow the program to print results on the command window. Furthermore, you can save results as text files by clicking on the SAVE RESULTS button. When you do so, you are first asked to select what results you want to save. All analysis performed so far are listed and you can choose as many as you like. For each selection, you are allowed to save two files. The first one contains the response plus the transformed predictors, while the second one contains only the generating vectors for the reduced subspace. You can skip saving either of them by clicking on the CANCEL button in the respective save dialog.

After some data is loaded, the popups below the figure axes get filled with identifiers for each column in the data file. You can get scatter plots of pairs of variables by selecting one of them in the popup corresponding to the vertical axis and the other one in the popup menu related to the horizontal axis. For a sequential display of scatter plots for some given variable, you can instead set the related identifier in one of these popups and then click on the SCATTER PLOTS button. You can then move through the scatter plots by clicking the UP and DOWN buttons placed below the SCATTER PLOTS button.

5. Conclusion

This paper introduces new software for sufficient dimension reduction based on maximum likelihood estimation of the central subspace. Matlab implementation of these methods allows easy integration of complex optimization tools and provides a flexible environment for graphics capabilities and further model development. Scripts are given to reproduce all published results concerning the methods discussed above, in agreement with the reproducible research philosophy. We plan to continue adding methods to the package and to finish code verification for full compatibility with the OCTAVE programming language in order to provide tools suitable to run under GNU software.

6. Aknowledgements

We are indebted to Eduardo Tabacman for his patient testing of the code and for bringing us some tips about the sg-min package. We also thank Ross Lippert for allowing us to distribute a modified version of his code with our package. This work was supported in part by National Science Foundation grant DMS-0704098 awarded to RDC.

References

Bura, E. and Pfeffer, R. M. (2003). Graphical methods for class prediction using dimension reduction techniques on DNA microarray data. *Bioinformatics* **19**, 1252–1258.

Burnham, K. and Anderson, D. (2002). *Model Selection and Multimodel inference.* New York: Wiley.

Cook, R. D. (1994). Using dimension-reduction subspaces to identify important inputs in models of physical systems. *Proceedings of the Section on Physical and Engineering Sciences*, 18-25. Alexandria, VA: American Statistical Association.

Cook, R. D. (1998). *Regression Graphics.* New York: Wiley.

Cook, R. D. (2003). Dimension reduction and graphical exploration in regression. *Statistics in Medicine* **22**, 1399–1413.

Cook, R. D. (2007). Fisher Lecture: Dimension reduction in regression (with discussion). *Statistical Science* **22**, 1-26.

Cook, R. D. and Forzani, L. (2008). Covariance reducing models: An alternative to spectral modelling of covariance matrices. *Biometrika* **95(4)**, 799-812.

Cook, R. D. and Forzani, L. (2009a). Principal fitted components in regression. *Statistcal Science* **23**, 4, 485-501.

Cook, R. D. and Forzani, L. (2009b). Likelihood-based sufficient dimension reduction. *Journal of the American Statistical Association.* **104**(485): 197-208. doi:10.1198/jasa.2009.0106.

Cook, R. D., Li, B. and Chiaromonte, F. (2009). Envelope models for parsimonious and efficent multivariate linear regression. *Statistica Sinica*, to appear with discussion. A preprint is available at http://www.stat.umn.edu/ dennis/RecentArticles/CLC.pdf.

Cook, R. D. and Yin, X. (2001). Dimension reduction and visualization in discriminant analysis (with discussion), *Australia New Zealand Journal of Statistics* **43** 147-199.

Cook, R. D. and Weisberg, S. (1991). Discussion of "Sliced inverse regression" by K. C. Li. *Journal of the American Statistical Association* **86**, 328–332.

Edelman, A., Arias, T. A. and Smith, S. T. (1999). The geometry of algorithms with orthogonality constraints. *SIAM J. Matrix Anal. Appl.* **20**, 303–353.

Li, K. C. (1991). Sliced inverse regression for dimension reduction (with discussion). *Journal of the American Statistical Association* **86**, 316–342.

Li, L. and Li, H. (2004). Dimension reduction methods for microarrays with application to censored survival data. *Bioinformatics* **20**, 3406–3412.

Li, B. and Wang S. (2007). On directional regression for dimension reduction. *Journal of American Statistical Association* **102**, 997–1008.

Lippert, R. and Edelman, A. (2000). Nonlinear eigenvalue problems with orthogonality constraints. In Bai, Z., Demmel, J., Dongarra, J., Ruhe, A. and van der Vorst, H: *Templates for the Solution of Algebraic Eigenvalue Problems: A Practical Guide.* Philadelphia: SIAM.

Pardoe, I., Yin, X. and Cook, R.D. (2007). Graphical tools for quadratic discriminant analysis. *Technometrics* **49**, 172–183.

Shao, Y., Cook, R. D. and Weisberg, S. (2007) Marginal tests with sliced average variance estimation. *Biometrika* **94**, 285–296.

Ye, Z. and Weiss, R. (2003). Using the Bootstrap to select one of a new class of dimension reduction methods. *Journal of the American Statistical Association* **98**, 968-978.

Zhu, L., Ohtaki, M. and Li, Y. (2005). On hybrid methods of inverse regression-based algorithms. *Computational Statistics and Data Analysis* **51**, 2621-2635.

Affiliation:

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

Liliana M. Forzani
IMAL and Department of Mathematics
Universidad Nacional del Litoral - CONICET
3000 Santa Fe, Argentina
E-mail: liliana.forzani@gmail.com
URL: http://liliana.forzani.googlepages.com/english

Diego R. Tomassi
SINC Lab
Universidad Nacional del Litoral - CONICET
3000 Santa Fe, Argentina
E-mail: diegotomassi@gmail.com

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