HIGH-DIMENSIONAL METRICS IN R: A TUTORIAL PRELIMINARY – PLEASE DO NOT DISTRIBUTE

VICTOR CHERNOZHUKOV, CHRISTIAN HANSEN, MARTIN SPINDLER

Abstract. High-dimensional Metrics (hdm) is an evolving collection of statistical methods for estimating and drawing inferences in settings with very many variables. An implementation of some of these methods in the R language is available in the package `hdm`. This vignette offers a brief tutorial to the package and the implemented methods. R and the package `hdm` are open-source software projects and can be freely downloaded from CRAN: `http://cran.r-project.org`.

Contents

Version: January 10, 2016.

1. Introduction

Analysis of high-dimensional models, models in which the number of parameters to be estimated is large relative to the sample size, is becoming increasingly important. Such models arise naturally in readily available high-dimensional data which have many measured characteristics available per individual observation as in, for example, large survey data sets, scanner data, and text data. Such models also arise naturally even in data with a small number of measured characteristics in situations where the exact functional form with which the observed variables enter the model is unknown. Examples of this scenario include semiparametric models with nonparametric nuisance functions. More generally, models with many parameters relative to the sample size often arise when attempting to model complex phenomena.

With increasing availability of such data sets in Economics and other fields, new methods for analyzing those data have been developed. The R package `hdm` contains implementations of recently developed methods with a focus on microeconometric applications and this vignette serves as tutorial. The methods are also useful in other disciplines like Medicine, Biology, Sociology or Psychology to mention a few. The methods which are implemented in this package are distinctive from already available methods in mainly the following: First, the choice of the penalization parameter λ in the Lasso regressions is theoretical grounded and feasible. Because of this we call it "rigorous"LASSO (=`rlasso`). The prefix **r** in function names should underscore this. In high-dimensions setting cross-validation is very popular, but lacking a theoretical justification and other proposals are often not feasible. Second, the Lasso regressions allow for non-Gaussian and heteroscedastic errors. Third, valid inference on low-dimensional (structural) parameters of interest is possible. Examples are inference for selected variables in a high-dimensional regression, a situation which arises in estimation of a treatment effect with very many control variables, or the estimation of a treatment effect in a high-dimensional Instrumental Variable (IV) setting.

In this vignette, we first show how to get started with package. Then we introduce briefly the data sets which are contained in the package and used later for illustration. Next the functions for LASSO and Post-LASSO estimation under heteroscedastic and non-Gaussian errors are presented. They are the core for the further applications. Next, different micoreconometric models in a high-dimensional setting are introduced and illustrated by an empirical application.

2. How to get started

R is an open source software project and can be freely downloaded from the CRAN website along with its associated documentation. The R package `hdm` can be downloaded from `cran.r-project.org`. To install the hdm package from R one simply types,

```
>install.packages("hdm")
```

The most current version of the package (development version) is maintained at R-Forge and can installed by

```
>install.packages("hdm", repos="http://R-Forge.R-project.org")
```

Provided that your machine has a proper internet connection and you have write permission in the appropriate system directories, the installation of the package should proceed automatically. Once the `hdm` package is installed, it needs to be made accessible to the current R session by the command,

Online help is available in two ways. If you know precisely the command you are looking for, e.g. in order to check the details, try:

```
help(package = "hdm")
help(lasso)
```

The former command gives an overview over the available commands in the package, and the latter gives detailed information about a specific command.

More generally one can initiate a web-browser help session with the command,

```
> help.start()
```

and navigate as desired. The browser approach is better adapted to exploratory inquiries, while the command line approach is better suited to confirmatory ones.

A valuable feature of R help files is that the examples used to illustrate commands are executable, so they can be pasted into an R session, or run as a group with a command like,

```
>example(lasso)
```

3. Data Sets

In this section we describe briefly the data sets which are contained in the package and used afterwards. They might also be of general interest for other researchers either for illustrating methods or for class-room presentation.

3.1. **Pension data.**

In the United States 401(k) plans were introduced to increase individual saving for retirement. They allow the individual to deduct contributions from taxable income and allow tax- free accrual of interest on assets held within the plan (within an account). Employers provide 401(k) plans, and employers may also match a certain percentage of an employee's contribution. Because 401(k) plans are provided by employers, only workers in firms offering plans are eligible for participation. This data set contains information about 401(k) participation and socio-economic characteristics of the individuals.

The data set can be loaded with

```
data(pension)
```

A description of the variables and further references are given at the help page

```
help(pension)
## starting httpd help server ...  done
```

The sample is drawn from the 1991 Survey of Income and Program Participation (SIPP) and consists of 9,915 observations. The observational units are household reference persons aged 25-64 and spouse if present. Households are included in the sample if at least one person is employed and no one is self-employed. All dollar amounts are in 1991 dollars. The 1991 SIPP reports household financial data across a range of asset categories. These data include a variable for whether a person works for a firm that offers a 401(k) plan. Households in which a member works for such a firm are classified as eligible for a 401(k). In addition, the survey also records the amount of 401(k) assets. Households with a positive 401(k) balance are classified as participants, and eligible households with a zero balance are considered nonparticipants. Available measures of wealth in the 1991 SIPP are total wealth, net financial assets, and net non-401(k) financial assets. Net non-401(k) assets are defined as the sum of checking accounts, U.S. saving bonds, other interest-earning accounts in banks and other financial institutions, other interest-earning assets (such as bonds held personally), stocks and mutual funds less non-mortgage debt, and IRA balances. Net financial assets are net non-401(k) financial assets plus 401(k) balances, and total wealth is net financial assets plus housing equity and the value of business, property, and motor vehicles.

3.2. **Growth Data.** The question what drives Economic Growth, measured in GDP, is a central question of Economics. A famous data set with information about GDP growth for many countries and long period was collected by Barro and Lee. This data set is also provided in the data set and can be loaded by

```
data(GrowthData)
```

This data sets contains the national growth rates in GDP per capita (Outcome) for many countries with additional covariates. A very important covariate is gdpsh465, which is the initial level of per-capita GDP. For further information we refer to the help page and the references herein, in particular the online descriptions of the data set.

3.3. **Institutions and Economic Development – Data on settler mortality.** This data set was introduced by paper Acemoglu, Johnson, and Robinson (2001) to analyse the effect of institutions on economic development. The data is contained in the package and can be accessed with

```
data(AJR)
```

The data set contains GDP, Settler Morality, an index measuring protection against expropriation risk and Geographic Information (Latitude and Continent dummies). In total 11 variables and 64 observations.

3.4. **Data on Eminent Domain.** Eminent domain refers to the government's taking of private property. This data set was collected to analyse the effect of number of pro-plaintiff appellate takings decisions on economic relevant outcome variables like house prices, measured by some index.

The data set is loaded into R by

```
data(EminentDomain)
```

The data set consists of four groups of variables:

- y: outcome variable, a house price index
- d: the treatment variable,represents the number of pro-plaintiff appellate takings decisions in federal circuit court c and year t
- x: exogenous control variables that include a dummy variable for whether there were relevant cases in that circuit-year, the number of takings appellate decisions, and controls for the distribution of characteristics of federal circuit court judges in a given circuit-year

- z: instrumental variables, here characteristics of judges serving on federal appellate panels

4. Partialling Out

An important concept which underlies some of the methods applied later is the so–called "partialling out", an application of the Frisch-Waugh-Lovell Theorem. The idea is to take out the effect of (conditioning) variables which are not the target of analysis. Suppose we are interested in the effect (=parameter estimate) of a single variable in a regression setting. One could either run an ols regression of the dependent variable on all independent variables and the focus on the parameter estimates of interest. An alternative approach leading to the same result is to partial out the effect of all variables which are not of interest from both the independent variable and the variable under consideration. Partialling out means to project on the space spanned by these variables and taking the residuals. Finally, the residuals of the depedent variable are regressed on the residuals of the variable under consideration and this gives the desired parameter estimates. We illsutrate this procedure with an simulation: First, we generate some data

```
set.seed(1234)
n <- 5000
p <- 20
X <- matrix(rnorm(n * p), ncol = p)
colnames(X) <- c("d", paste("x", 1:19, sep = ""))
xnames <- colnames(X)[-1]
beta <- rep(1, 20)
y <- X %*% beta + rnorm(n)
dat <- data.frame(y = y, X)
```

We are interest in the effect of the variable d.

One way is to conduct a full fit and then concentrate on the parameter of interest

```
# full fit
fmla <- as.formula(paste("y ~ ", paste(colnames(X), collapse = "+")))
full.fit <- lm(fmla, data = dat)
```

An alternative is first to partial out the influence of the x-variables and then do a regression of the corresponding residuals

```
fmla.y <- as.formula(paste("y ~ ", paste(xnames, collapse = "+")))
fmla.d <- as.formula(paste("d ~ ", paste(xnames, collapse = "+")))
# partial fit via ols
rY <- lm(fmla.y, data = dat)$res
rD <- lm(fmla.d, data = dat)$res
partial.fit.ls <- lm(rY ~ rD)
```

Instead of OLS the partialling out might also be done with a different estimation method, e.g. Lasso, which is in particular attractive in high-dimensional settings. An alternative is using the function `rlassoEffect` resp. `rlassoEffectone` which implements the partialling out in a slightly different way.

```
# partial fit via lasso
rY <- rlasso(fmla.y, data = dat)$res
rD <- rlasso(fmla.d, data = dat)$res
## No variables selected!
partial.fit.lasso <- lm(rY ~ rD)
# directly via rlassoEffect
lasso.Effect <- rlassoEffectone(x = X[, -1], d = X[, 1], y = y)
## No variables selected!
alpha <- lasso.Effect$alpha
se <- lasso.Effect$se
```

We see in the table which summarizes the results that full regression and the partial regression give the same results. As in the simulation the number of observations is much larger than the parameters to estimate lasso gives – not surprisingly – comparable results.

```
tab
```

	Estimate	Std. Error
full reg	1.01	0.0140890
partial reg	1.01	0.0140918
partial reg via lasso 1	1.00	0.0140627
partial reg via lasso 2	1.01	0.0139240

5. LASSO and Post-LASSO Estimation under Heteroscedastic and Non-Gaussian Errors

5.1. **Estimation.** We consider linear, high dimensional sparse (HDS) regression models. Despite its simplicity it is very powerful and might also capture / approximate nonlinear functions by using basis functions and transformations of the base regressors. The HDS regression model has a large number of regressors p, possibly much larger than the sample size n, but only a relatively small number $s < n$ of these regressors are important for capturing accurately the main features of the regression function. The latter assumption makes it possible to estimate these models effectively by searching for approximately the right set of the regressors, using ℓ_1-based penalization methods.

Let us first consider an exact or parametric HDS regression model, namely,

$$y_i = x_i'\beta_0 + \varepsilon_i, \mathbb{E}[\varepsilon_i] = 0, \beta_0 \in \mathbb{R}^p, i = 1, \ldots, n$$

where y_i are observations of the response variable, $x_i = (x_{i,j}, \ldots, x_{i,p})$'s are observations of $p-$dimensional fixed regressors, and ε_i's are iid, mean-zero centered disturbances, where possibly $p \gg n$. An important point is that the disturbances are **not** restricted to be Gaussian or homoscedastic. This means that we allow for both non-Gaussian and heteroscedastic errors. In order to get ”valid” results in those settings, a data-driven weighting matrix is introduced.

The LASSO estimator is defined as

$$\hat{\beta} = \arg\min_{\beta \in \mathbb{R}^p} \sum_{i=1}^{n} \left(y_i - \sum_{j=1}^{p} x_{i,j}\beta_j \right)^2 + \lambda \sum_{j=1}^{p} \gamma_j |\beta_j|,$$

where $\lambda > 0$ is the "penalty level"and γ_j are "penalty loadings". The penalty loadings are chosen to insure basic equivariance of coefficient estimates to rescaling of $x_{i,j}$ and can also be chosen to address the heteroskedastiticy, and non-normality in model errors.

Regularization by the ℓ_1-norm naturally helps the LASSO estimator to avoid overfitting the data, but it also shrinks the fitted coefficients towards zero, causing a potentially significant bias. In order to remove some of this bias, let us consider the Post-LASSO estimator that applies ordinary least squares regression to the model $\hat{T}$ selected by LASSO, formally,

$$\hat{T} = \text{support}(\hat{\beta}) = \{j \in \{1, \ldots, p\} : |\hat{\beta}| > 0\}.$$

The Post-LASSO estimate is then defined as

$$\tilde{\beta} \in \arg\min_{\beta \in \mathbb{R}^p} \left(y_i - \sum_{j=1}^{p} x_{i,j}\beta_j \right)^2 : \beta_j = 0 \forall j \in \hat{T}^C,$$

where $\hat{T}^C = \{1, \ldots, p\} \setminus \hat{T}$. In words, the estimator is ordinary least squares applied to the data after removing the regressors that were not selected by LASSO.

A key concept in estimation in high-dimenisonal settings is "sparsity". This states that the number of potential variables might be large (even larger than the sample size), but the number of variables with non-zero coefficients is quite small. Although this assumption is reasonable in many situations, it might look strict in other application and can be weakened to so-called "approximate sparsity". This means that potentially all variables have non-zero coefficients, but the signal can be captured sufficiently well by a small set of variables.

MS: onw section for penalization parameter? More theory?

A crucial point for the performance of LASSO is the choice of the penalization parameter λ. In high-dimensions setting cross-validation is very popular, but lacking a theoretical justification and other proposals are often not feasible. The choice of the penalization parameter λ in the Lasso regressions in this package is theoretical grounded and feasible. Therefore we call it "rigorous"lasso and hence add as prefix **r** to the function names.

The option `penalty` of the function `rlasso` allows different choices for the penalization parameter and loadings. It allows for homoscedastic or heterosceastic errors with default `homoscedastic = FALSE`. Moreover, the dependence structure of the design matrix might be taken into consideration for calculation of the penalization parameter with `X.design = "dependent"`. The default is `"dependent"`. With `lambda.start` initial values for the algorithm can be set to get started. In some cases the user might want to use a predefined, fixed penalization parameter. This can be done by a special option in the follwoing way: set homoscedastic to "none" and supply a values `lambda.start`. Then this value is used as penalty parameter with independent design and heteroscedastic errors to weight the regressors.

We consider again the LASSO optimization problem

$$\min_{\beta\in\mathbb{R}^p} \mathbb{E}_n[(y_i - x_i'\beta)^2] + \frac{\lambda}{n}||\hat{\Psi}\beta||_1$$

where $||\beta||_1 = \sum_{j=1}^p |\beta_j|$ and $\hat{\Psi} = diag(\hat{\psi}_1, \ldots, \hat{\psi}_p)$ is a diagonal matrix consisting of regressor dependent penalty loadings.

In the case of homoscedasticity, $\hat{\Psi}$ is the identity matrix as no weighting in neccessary. In the X-independent case the penalty is set to

$$\lambda = 2c\hat{\sigma}\Phi^{-1}(1-\gamma/(2p)),$$

where Φ denotes the cumulative standard normal distribution. In the X-dependent case the penalty level is calculated as

$$\lambda = 2c\hat{\sigma}\Lambda(1-\gamma|X),$$

where

$$\Lambda(1-\gamma|X) = (1-\gamma) - \text{quantile of} \quad n||\mathbb{E}_n[x_i g_i]||_\infty|X,$$

where $X = [x_1, \ldots, x_n]'$ and g_i are iid $N(0,1)$, which an be easily approximated by simulation.

In the case of heteroscedasticity, the loadings are given by $\hat{\psi}_j = \sqrt{\mathbb{E}_n[x_{ij}^2\varepsilon_i^2]}$. In the X-independet setting, the penalization parameter is given by

$$\lambda = 2c\sqrt{n}\Psi^{-1}(1-\gamma/(2p)).$$

In the X-dependent case, the penalty is estimated by a multiplier bootstrap procedure.

$$\lambda = c \times c_W(1-\gamma),$$

where $c_W(1-\gamma)$ is the $1-\alpha$-quantile of the random W which is given by

$$W := \sqrt{n}\max_{1\le j\le p}|2\mathbb{E}_n[x_{ij}\hat{\varepsilon}_i e_i]|.$$

e_i is iid standard normal distributed and $\hat{\varepsilon}_i$ denotes an estimate of the residuals.

The constants c and γ from above can be set in the option `penalty`. The quantities $\hat{\varepsilon}$, $\hat{\Psi}$, $\hat{\sigma}$ are calculated in a iterative manner. The maximum number of iterations and the tolerance when the algorithms should stop can be set with `control`.

5.2. **Inference.** Besides estimation inference is highly relevant for empirical applications. Using LASSO as a method for penalized estimation of the coefficients of a sparse linear model is useful for obtaining forecasting rules and for estimating which variables have a strong association to an outcome in a sparse framework. However, naively using the results obtained from such a procedure to draw inferences about model parameters may be invalid. The reason is that only under very strong conditions LASSO is consistent concerning model selection which hardly apply in empirical applications and omission of relevant variables cannot be excluded in general distorting the validity of traditional approaches. This caveat applies to both low- **and** high-dimensional settings when model selection is considered. Moreover, even in the case of consistent model selection non-uniformity of the results might lead to poor finite sample approximations. Pötscher and Leeb have analyzed these issues in a series of papers.

Valid inference on low-dimensional parameters in a high-dimensional setting as described in the previous section is possible by the so-called double selection method.

We are interested in doing inference on a low dimensional parameter α. For ease of exposition, we consider the case that the parameter of interest is a scalar. This gives us the following model

$$y_i = \alpha_0 d_i + x_i'\beta_0 + \xi_i, i = 1, \ldots, n,$$

with x_i p-dimensional vector of controls, d_i the low-dimensional object we are concerned about, and ξ_i iid errors with $\mathbb{E}[\xi_i|d_i, x_i] = 0$. To perform valid inference, we introduce an auxiliary regression:

$$d_i = x_i'\theta + v_i,$$

where $\mathbb{E}[v_i|x_i] = 0$ and captures the relation between variable of interest and the control variables. The double selection method is designed to guard against moderately sized model selection mistakes and consists of the following steps:

(1) Select controls x_{ij}'s that predict y_i by LASSO.
(2) Select controls x_{ij}'s that predict d_i by LASSO.
(3) Run OLS of y_i on d_i and the union of controls selected in steps 1 and 2.

The additional selection step controls the omitted variable bias. In essence the procedure is a selection version of Frisch-Waugh procedure for estimating linear regression.

5.3. **Functions for Lasso and Inference.** The package contains several functions for estimation and inference by LASSO. In this section we introduce the core functions which are then illustrated in an example.

The core functions for estimation are `LassoShooting.fit` and `rlasso`. `LassoShooting.fit` is the working horse for the estimation and implements a version of the Shooting Lasso Algorithm. It allows variable dependent penalties or weights. The function `rlasso` implements LASSO and Post-LASSO estimation with non-Gaussian errors and under heteroskedasticity. The default option is `post=TRUE`. The user can also decide if an unpenalized `intercept` should be included (`TRUE` by default) and if the regressors should be `normalized` (`TRUE` by default). The choice of `penalty` is passed by a list. It can be chosen between homoscedastic errors (`FALSE` by default) and if the penalty should take into consideration the design of the `X` matrix which gives tighter bounds on the penalization parameter. The two options for `X.design` are `dependent` and the default `independent`. This function returns an object of S3 class `rlasso` for which methods like `predict`, `print`, `summary` are provided. The function `rlassoEffect` does inference for selected variables. Those can be specified either by the variable names, an integer valued vector giving their position in `x` or by logical indicating the variables for which inference should be conducted. It returns an object of S3 class `rlassoEffect` for which the methods `summary`, `print`, `confint`, and `plot` are provided. `rlassoEffect` is a wrap function for `rlassoEffectone` which does inference for a single, specified variable.

5.4. **Example.** In this section we demonstrate the functionality for LASSO estimation and inference at a simulated data set. In the next sections the methods will be illustrated at hand of empirical applications using the data sets introduced earlier.

First we generate data in a sparse linear model:

```
set.seed(1234)
n <- 250  #sample size
p <- 100  # number of variables
s <- 10  # nubmer of non-zero variables
X <- matrix(rnorm(n * p), ncol = p)
beta <- c(rep(2, s), rep(0, p - s))
y <- 1 + X %*% beta + rnorm(n)
```

Next we estimate it, show the results and make predictions for the old and for new x-variables

```
lasso.reg <- rlasso(x = X, y = y, post = TRUE, intercept = TRUE,
    normalize = TRUE)
# Methods for Lasso
print(lasso.reg, all = FALSE)
## 
## Call:
## rlasso.default(x = X, y = y, post = TRUE, intercept = TRUE, normalize = TRUE)
## 
##    V1     V2     V3     V4     V5     V6     V7     V8 
## 1.980  2.093  1.976  1.989  1.930  1.925  2.081  1.976 
##    V9    V10 
## 1.974  1.989
summary(lasso.reg, all = FALSE)
## 
## Call:
## rlasso.default(x = X, y = y, post = TRUE, intercept = TRUE, normalize = TRUE)
## 
## Post-Lasso Estimation:  TRUE 
## 
## Total number of variables: 100
## Number of selected variables: 10 
## 
## Residuals: 
##      Min       1Q   Median       3Q      Max 
## -2.50111 -0.66737  0.06159  0.66584  2.49105 
## 
##     Estimate
## V1     1.980
## V2     2.093
## V3     1.976
## V4     1.989
## V5     1.930
## V6     1.925
## V7     2.081
## V8     1.976
## V9     1.974
## V10    1.989
```

```
##
## Residual standard error: 0.9467
yhat <- predict(lasso.reg)
Xnew <- matrix(rnorm(n * p), ncol = p)
yhat.new <- predict(lasso.reg, newdata = Xnew)
```

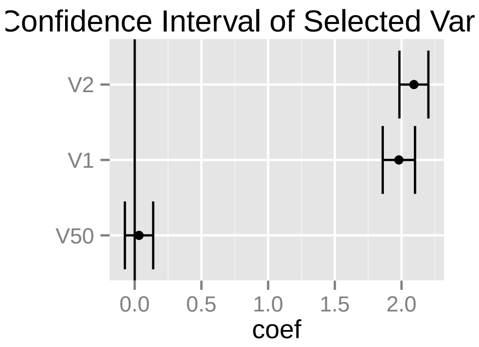
The methods `print` and `summarize` have the option `all` which by setting to `FALSE` shows only the non-zero coefficients which gives clarity in very high-dimensional settings.

We can also do inference on selected variables, e.g. variables number one, two and fifty:

```
lasso.effect <- rlassoEffect(x = X, y = y, index = c(1, 2, 50))
## No variables selected!
## No variables selected!
## No variables selected!
print(lasso.effect)
##
## Call:
## rlassoEffect.default(x = X, y = y, index = c(1, 2, 50))
##
## Coefficients:
##      V1       V2      V50
## 1.98009  2.09274  0.03237
summary(lasso.effect)
## [1] "Estimation of the effect of selected variables in a high-dimensional regression"
##      coeff.     se. t-value p-value
## V1  1.98009 0.06174  32.071  <2e-16 ***
## V2  2.09274 0.05540  37.777  <2e-16 ***
## V50 0.03237 0.05411   0.598    0.55
## ---
## Signif. codes:
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
confint(lasso.effect)
##           2.5 %    97.5 %
## V1   1.85908403 2.1011058
## V2   1.98416197 2.2013161
## V50 -0.07367725 0.1384185
```

Finally, we can also plot the estimated effects with their confidence intervals:

```
plot(lasso.effect, main = "Confidence Interval of Selected Variables")
## [1] "Estimation of the effect of selected variables in a high-dimensional regression"
##      coeff.     se. t-value p-value
## V1  1.98009 0.06174  32.071  <2e-16 ***
## V2  2.09274 0.05540  37.777  <2e-16 ***
## V50 0.03237 0.05411   0.598    0.55
## ---
## Signif. codes:
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## Scale for 'x' is already present.  Adding another scale for 'x', which
will replace the existing scale.
```



6. Instrumental Variable Estimation in a High-Dimensional Setting

In many applied settings the researcher is interested in estimating the (structural) effect of a variable (treatment variable), but this variable is endogenous, i.e. correlated with the error term which might occur because of several reasons. Classical examples are omitted variables, measurement error or what often arises in economic and medical applications that the assignment was not random but result of some optimization procedure with respect to the potential outcomes. While ordinary least squares estimation of the parameter of interest are inconsistent in general, instrumental variables (IV) estimation might be a solution.

We consider the linear instrumental variables model

$$y_i = \alpha_0 d_i + \beta_0 x_i' + \varepsilon_i, \quad (1)$$

$$d_i = z_i'\Pi + \gamma_0 x_i' + v_i, \quad (2)$$

where $\mathbb{E}[\varepsilon_i|x_i, z_i] = \mathbb{E}[v_i|x_i, z_i] = 0$, but $\mathbb{E}[\varepsilon_i v_i] \neq 0$ leading to endogeneity. In this setting d_i is a scalar endogenous variable of interest, z_i is a p_z-dimensional vector of instruments and x_i is a p_x-dimensional vector of control variables.

In this section we present methods to estimate the effect α_0 in a setting, where either p_x is high-dimensional, either p_z is high-dimensional or both p_x and p_z are high-dimensional. The wrap function `rlassoIV` handels all these cases, but allows additionally low-dimensional p_x and p_z which leads to classical two-stage least squares (tsls) estimation. The function `rlassoIV` hast the options `select.X` and `select.Z` which determine if selection shall take place, both with default values `TRUE`. The class of the return object depends on the chosen options, but the methods `summary`, `print` and `confint` are available.

6.1. **Selection on the IVs.**

6.1.1. *Estimation.*

6.1.2. *Application – Eminent Domain.* Federal court rulings that a government seizure was unlawful (pro-plaintiff rulings) thus uphold individual property rights and make future exercise of eminent domain more difficult due to the structure of the US legal system. A more detailed discussion of the economics of takings law (or eminent domain) and other institutional and econometric considerations can be found in Belloni, Chen, Chernozhukov, and Hansen (2012) and Chen and Yeh (2012). The analysis of the effects of takings law is complicated by the possible endogeneity between takings law decisions and economic variables: for example, a taking may be less likely if real estate prices are low and sellers are eager to unload property. To address the potential endogeneity of takings law, we employ an instrumental variables strategy based on the identification argument of Chen and Sethi (2010) and Chen and Yeh (2012) that relies on the random assignment of judges to federal appellate panels. Because judges are randomly assigned to three-judge panels to decide appellate cases, the exact identity of the judges and their demographics are randomly assigned conditional on the distribution of characteristics of federal circuit court judges in a given circuit-year. Under this random assignment, the characteristics of judges serving on federal appellate panels can only be related to property prices through the judges' decisions; thus the judge's characteristics will plausibly satisfy the instrumental variable exclusion restriction.

First, we load the data an construct the matrices with the controls (X) and instruments (Z) and the output (y) and treatment variables (d)

```r
data(EminentDomain)
Z <- as.matrix(EminentDomain[, grep("z", colnames(EminentDomain))])
X <- as.matrix(EminentDomain[, grep("x", colnames(EminentDomain))])
y <- as.matrix(EminentDomain[, "y"])
d <- as.matrix(EminentDomain[, "d"])
```

As mentioned above, y is the Economic outcome, the Case-Shiller house price index, d the number of pro plaintiff appellate takings decisions in federal circuit court c and year t, X a matrix with control variables and Z the matrix with instruments, here socio-economic and demographic characteristics of the judges.

First, we estimate the effect of the treatment variable by simple OLS and 2SLS using two instruments:

```r
ED.ols <- lm(y ~ cbind(d, X))
ED.2sls <- tsls(y = y, d = d, x = X, z = Z[, 1:2], intercept = FALSE)
```

Next we estimate the model with selection on the instruments

```r
# lasso.IV <- rlassoIV(x=X, d=d, y=y, z=Z, select.X=FALSE,
# select.Z=TRUE, post=TRUE, normalize=FALSE, intercept=TRUE)
# # wrap function
lasso.IV <- rlassoIVselectZ(x = X, d = d, y = y, z = Z, post = TRUE,
    normalize = TRUE, intercept = FALSE)  #direct call
print(lasso.IV)
##
## Call:
## rlassoIVselectZ(x = X, d = d, y = y, z = Z, post = TRUE, normalize = TRUE,
##     intercept = FALSE)
```

```
##
## Coefficients:
##      d1
## 0.05598
summary(lasso.IV)
## [1] "Estimation of the effect of selected variables in a high-dimensional regression"
##     coeff.     se. t-value p-value
## d1 0.05598 0.03538   1.582   0.114
confint(lasso.IV)
##          2.5 %    97.5 %
## d1 -0.01337748 0.1253287
###############################################################
dZlasso <- rlasso(x = Z, y = d, post = TRUE, normalize = TRUE,
    intercept = FALSE)
Zsel <- Z[, dZlasso$index]
ivift.lasso <- tsls(y = y, d = d, x = X, z = Zsel, intercept = FALSE)
```

MS: not exactly the estimate of BCCH 2012, in particular se!!

6.2. **Selection on the exogenous variables.**

6.2.1. *Estimation.*

6.2.2. *Economic Development and Institutions.*

Estimating the effect of institutions on output is complicated by the clear potential for simultaneity between institutions and output: specifically, better institutions may lead to higher incomes, but higher incomes may also lead to the development of better institutions. To help overcome this simultaneity, Acemoglu, Johnson, and Robinson (2001) use mortality rates for early European settlers as an instrument for institution quality. The validity of this instrument hinges on the argument that settlers set up better institutions in places where they are more likely to establish long-term settlements; that where they are likely to settle for the long term is related to settler mortality at the time of initial colonization; and that institutions are highly persistent. The exclusion restriction for the instrumental variable is then motivated by the argument that GDP, while persistent, is unlikely to be strongly influenced by mortality in the previous century, or earlier, except through institutions. In this application, a crucial point is on which variables to condition on. Economic / scientific reasoning suggests that Latitude is an important control variable, but somehow it is not obvious if this variable should enter linear, quadratic or an even higher order polynomial. In such a situation a data-driven selection rule might be beneficial, especially in the case when the ratio of variables to number of observations is quite high as in this application. Here we include Latitude additional as quadratic term and include all first-order interactions of all exogenous variables.

First, we process the data

```
data(AJR)
y <- AJR$GDP
d <- AJR$Exprop
z <- AJR$logMort
```

```
X <- model.matrix(~(Latitude + Latitude2 + Africa + Asia + Namer +
    Samer)^2, data = AJR)
```

Then we estimate an IV model with selection on the X

```
# AJR.Xselect <- rlassoIV(x=X, d=d, y=y, z=z, select.X=TRUE,
# select.Z=FALSE)
AJR.Xselect <- rlassoIVselectX(x = X, d = d, y = y, z = z, intercept = TRUE,
    normalize = TRUE, post = TRUE)
## No variables selected!
print(AJR.Xselect)
##
## Call:
## rlassoIVselectX(x = X, d = d, y = y, z = z, post = TRUE, intercept = TRUE,
##     normalize = TRUE)
##
## Coefficients:
##    d1
## 0.771
summary(AJR.Xselect)
## [1] "Estimation of the effect of selected variables in a high-dimensional regression"
##    coeff.   se. t-value  p-value
## d1  0.771 0.203   3.798 0.000146 ***
## ---
## Signif. codes:
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
confint(AJR.Xselect)
##        2.5 %   97.5 %
## d1 0.3731934 1.168885
```

Here we demonstrate also the concept of partialling out which is quite useful and underlies much of the theory. First, we partial out the effect of the exogenous variables with ols and then apply 2SLS

```
# parialling out by linear model
fmla.y <- GDP ~ (Latitude + Latitude2 + Africa + Asia + Namer +
    Samer)^2
fmla.d <- Exprop ~ (Latitude + Latitude2 + Africa + Asia + Namer +
    Samer)^2
fmla.z <- logMort ~ (Latitude + Latitude2 + Africa + Asia + Namer +
    Samer)^2
rY <- lm(fmla.y, data = AJR)$res
rD <- lm(fmla.d, data = AJR)$res
rZ <- lm(fmla.z, data = AJR)$res
ivfit.lm <- tsls(y = rY, d = rD, x = NULL, z = rZ, intercept = FALSE)
```

Next, we replace the “ols operator”by lasso for partialling out

```r
# parialling out by lasso
rY <- rlasso(fmla.y, data = AJR)$res
rD <- rlasso(fmla.d, data = AJR)$res
## No variables selected!
rZ <- rlasso(fmla.z, data = AJR)$res
ivfit.lasso <- tsls(y = rY, d = rD, x = NULL, z = rZ, intercept = FALSE)
```

6.3. Selection on the IVs and exogenous variables.

6.3.1. *Eminent Domain Example continued.* Here again we consider the example from above, but now we do selection on the exogenous variables and on the instruments.

```r
data(EminentDomain)
Z <- as.matrix(EminentDomain[, grep("z", colnames(EminentDomain))])
X <- as.matrix(EminentDomain[, grep("x", colnames(EminentDomain))])
y <- as.matrix(EminentDomain[, "y"])
d <- as.matrix(EminentDomain[, "d"])
```

```r
lasso.IV <- rlassoIV(x = X, d = d, y = y, z = Z, select.X = TRUE,
    select.Z = TRUE)  # wrap function
## No variables selected!
## Error in solve.default(t(Z) %*% Z): Lapackroutine dgesv:  System ist
genau singulär:  U[1,1] = 0
print(lasso.IV)
##
## Call:
## rlassoIVselectZ(x = X, d = d, y = y, z = Z, post = TRUE, normalize = TRUE,
##     intercept = FALSE)
##
## Coefficients:
##      d1
## 0.05598
summary(lasso.IV)
## [1] "Estimation of the effect of selected variables in a high-dimensional regression"
##     coeff.     se. t-value p-value
## d1 0.05598 0.03538   1.582   0.114
confint(lasso.IV)
##          2.5 %    97.5 %
## d1 -0.01337748 0.1253287
```

7. Estimation of Treatment Effects in a High-Dimensional Setting

7.1. Estimation of the treatment effect in the case of many controls.

7.1.1. *Application: Determinants of Growth.* the empirical growth literature is estimating the effect of an initial (lagged) level of GDP (Gross Domestic Product) per capita on the growth rates of GDP per capita. In particular, a key prediction from the classical Solow-Swan-Ramsey growth model is the hypothesis of convergence,which states that poorer countries should typically grow faster and therefore should tend to catch up with the richer countries. Such a hypothesis implies that the effect of the initial level of GDP on the growth rate should be negative. As pointed out in Barro and Sala-i-Martin, this hypothesis is rejected using a simple bivariate regression of growth rates on the initial level of GDP. (In this data set, linear regression yields an insignificant positive coefficient of 0.0013.) In order to reconcile the data and the theory, the literature has focused on estimating the effect conditional on the pertinent characteristics of countries. Covariates that describe such characteristics can include variables measuring education and science policies, strength of market institutions, trade openness, savings rates and others. The theory then predicts that for countries with similar other characteristics the effect of the initial level of GDP on the growth rate should be negative. Thus, we are interested in a specification of the form:

$$y_i = \alpha_0 + \alpha_1 \log G_i + \sum_{j=1}^{p} \beta_j X_{ij} + \varepsilon_i,$$

where y_i is the growth rate of GDP over a specified decade in country i, G_i is the initial level of GDP at the beginning of the specified period, and the X_{ij}'s form a long list of country i's characteristics at the beginning of the specified period. We are interested in testing the hypothesis of convergence, namely that $\alpha_1 < 0$. Given that in standard data-sets, such as Barro and Lee data, the number of covariates p we can condition on is large, at least relative to the sample size n, covariate selection becomes a crucial issue in this analysis. In particular, previous findings came under severe criticism for relying on ad hoc procedures for covariate selection. In fact, in some cases, all of the previous findings have been questioned. Since the number of covariates is high, there is no simple way to resolve the model selection problem using only classical tools. Indeed the number of possible lower-dimensional models is very large, although [16] and [22] attempt to search over several millions of these models. We suggest ℓ_1-penalization and post- ℓ_1-penalization methods to address this important issue. In Section 8, using these methods we estimate the growth model (7.1.1) and indeed find rather strong support for the hypothesis.

First, we load and prepare the data

```
data(GrowthData)
dim(GrowthData)
## [1] 90 63
y <- GrowthData[, 1]
d <- GrowthData[, 3]
X <- as.matrix(GrowthData)[, -c(1, 2, 3)]
varnames <- colnames(GrowthData)
```

Now we can estimate the influence of initial GDP level to test the Economic hypothesis

```
xnames <- varnames[-c(1, 2, 3)]  # names of X variables
dandxnames <- varnames[-c(1, 2)]  # names of D and X variables
# create formulas by pasting names (this saves typing times)
fmla <- as.formula(paste("Outcome ~ ", paste(dandxnames, collapse = "+")))
full.fit <- lm(fmla, data = GrowthData)
fmla.y <- as.formula(paste("Outcome ~ ", paste(xnames, collapse = "+")))
fmla.d <- as.formula(paste("gdpsh465~ ", paste(xnames, collapse = "+")))
# partial fit via ols
rY <- lm(fmla.y, data = GrowthData)$res
rD <- lm(fmla.d, data = GrowthData)$res
partial.fit.ls <- lm(rY ~ rD)
# partial ls via lasso
rY <- rlasso(fmla.y, data = GrowthData)$res
## No variables selected!
rD <- rlasso(fmla.d, data = GrowthData)$res
partial.fit.lasso <- lm(rY ~ rD)
```

```
dX <- as.matrix(cbind(d, X))
partial.fit.lasso2 <- rlassoEffect(x = dX, y = y, index = 1,
    intercept = TRUE, normalize = TRUE, post = TRUE)
## No variables selected!
# partial.fit.lasso2 <- rlassoEffectone(x=X,y=y,d=d)
summary(partial.fit.lasso2)
## [1] "Estimation of the effect of selected variables in a high-dimensional regression"
##     coeff.      se. t-value p-value
## d -0.04432  0.01685  -2.631 0.00851 **
## ---
## Signif. codes:
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
library(xtable)
table <- matrix(0, 4, 2)
table[1, ] <- summary(full.fit)$coef["gdpsh465", 1:2]
table[2, ] <- summary(partial.fit.ls)$coef["rD", 1:2]
table[3, ] <- summary(partial.fit.lasso)$coef["rD", 1:2]
table[4, ] <- summary(partial.fit.lasso2)["d", 1:2]
colnames(table) <- names(summary(full.fit)$coef["gdpsh465", ])[1:2]
rownames(table) <- c("full reg", "partial reg", "partial reg via lasso 1",
    "partial reg\nvia lasso 2")
# adjust partial ls standard error by sqrt{n/(n-p)}
n <- dim(GrowthData)[1]
p <- dim(GrowthData)[2]
table[2, 2] <- table[2, 2] * sqrt(n/(n - p))
tab <- xtable(table, digits = c(2, 2, 7))
```

```
tab
```

	Estimate	Std. Error
full reg	-0.01	0.0298877
partial reg	-0.01	0.0307801
partial reg via lasso 1	-0.04	0.0153192
partial reg via lasso 2	-0.04	0.0168463

7.2. **Estimation of LATE, ATE, LATET, and , ATET.**

7.2.1. *Treatment effects – A short introduction.* MS: TBD: Short explanation / definition

7.2.2. *401(k) plan participation.* Though it is clear that 401(k) plans are widely used as vehicles for retirement saving, their effect on assets is less clear. The key problem in determin- ing the effect of participation in 401(k) plans on accumulated assets is saver heterogeneity coupled with nonrandom selection into participation states. In particular, it is generally recognized that some people have a higher preference for saving than others. Thus, it seems likely that those individuals with the highest unobserved preference for saving would be most likely to choose to participate in tax-advantaged retirement savings plans and would also have higher savings in other assets than individuals with lower unobserved saving propensity. This implies that con- ventional estimates that do not allow for saver heterogeneity and selection of the participation state will be biased upward, tending to overstate the actual savings effects of 401(k) and IRA participation.

Again, we start first with the data preparation

```
data(pension)
y <- pension$tw
d <- pension$p401
z <- pension$e401
X <- model.matrix(~(age + inc + fsize + educ + marr + twoearn +
    db + pira)^2, data = pension)
```

Now we can calculate the interesting treatment effects

```
pension.late <- rlassoLATE(X, d, y, z, bootstrap = NULL, post = TRUE,
    intercept = FALSE, normalize = FALSE)
print(pension.late)
##
## Call:
## rlassoLATE(x = X, d = d, y = y, z = z, bootstrap = NULL, post = TRUE,
##     intercept = FALSE, normalize = FALSE)
##
## Treatment Effect
## Type: LATE
## Value:
## [1]  7788
pension.latet <- rlassoLATET(X, d, y, z, bootstrap = NULL, post = TRUE,
    intercept = FALSE, normalize = FALSE)
print(pension.latet)
```

```
##
## Call:
## rlassoLATET(x = X, d = d, y = y, z = z, bootstrap = NULL, post = TRUE,
##     intercept = FALSE, normalize = FALSE)
##
## Treatment Effect
## Type: LATET
## Value:
## [1]  9213
```

```
pension.ate <- rlassoATE(X, d, y, z, bootstrap = NULL, post = TRUE,
    intercept = FALSE, normalize = FALSE)
print(pension.ate)
```

```
##
## Call:
## rlassoLATE(x = x, d = d, y = y, z = z, bootstrap = bootstrap,
##     nRep = nRep, post = TRUE, intercept = FALSE, normalize = FALSE)
##
## Treatment Effect
## Type: ATE
## Value:
## [1]  9729
```

```
pension.atet <- rlassoATET(X, d, y, z, bootstrap = NULL, post = TRUE,
    intercept = FALSE, normalize = FALSE)
print(pension.atet)
```

```
##
## Call:
## rlassoLATET(x = x, d = d, y = y, z = z, bootstrap = bootstrap,
##     nRep = nRep, post = TRUE, intercept = FALSE, normalize = FALSE)
##
## Treatment Effect
## Type: ATET
## Value:
## [1]  9579
```

8. Some Tipps and Tricks

9. Conclusion

An introduction to some of the capabilities of the R package `hdm` package has been given with some examples describing its basic functionality. Inevitably, new applications will demand new features and, as the project is in its initial phase, unforeseen bugs will show up. In either case comments and suggestions of users are highly appreciated. It is intended to update the documentation (including this vignette) and the package periodically. The most current version of the R package and its accompanying vignette will be made available at the homepage of the maintainer and `cran.r-project.org`. See the R command `vignette()` for details on how to find and view vignettes from within R .

References