HIGH-DIMENSIONAL METRICS IN R

VICTOR CHERNOZHUKOV, CHRISTIAN HANSEN, MARTIN SPINDLER

Abstract. The package High-dimensional Metrics (`hdm`) is an evolving collection of statistical methods for estimating and drawing inferences in high-dimensional approximately sparse models. This vignette offers a brief introduction and a tutorial to 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 22, 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, and we create many technical variables, a dictionary, from the raw characteristics. 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 data science fields, new methods for analyzing those data have been developed. The R package `hdm` contains implementations of recently developed methods for high-dimensional approximately sparse models, mainly relying on forms of Lasso and post-Lasso as well as related estimation and inference methods. The methods are illustrated with econometric applications, but 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 other packages in mainly the following three major ways:

1) First, we provide efficient estimators and uniformaly valid confidence intervals for various low-dimensional causal/structural parameters appearing in high-dimensional approximately sparse models. For example, we provide efficient estimators and uniformly valid confidence intervals for a regression coefficient on a target variable (e.g., a treatment or policy variable) in a high-dimensional sparse regression model. We also provide estimates and confidence intervals for average treatment effect (ATE) and average treatment effect for the treated (ATET), as well extensions of these parameters to the endogenous setting.

2) Second, we provide theoretically grounded, data-driven choice of the penalty level λ in the Lasso regressions is both theoretical grounded and data-driven. 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 it lacks a theoretical justification and some theoretical proposals for the choice of λ are often not feasible.

3) Third, we provide a version of Lasso regressions that expressly handle and allow for non-Gaussian and heteroscedastic errors.

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 econometric 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 we simply type,

```
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,

```
library(hdm)
```

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(rlasso)
```

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(rlasso)
```

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 "sub data sets"which have the following structure:

- 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

The four data sets differ mainly in the dependent variable wich is some kind of house price index or the GDP, exactly speaking the logarithm of those variales: repeat-sales FHFA/OFHEO house price index for metro (FHFA) and non-metro (NM), the Case-Shiller home price index (CS), and state-level GDP from the Bureau of Economic Analysis.

4. Prediction using Approximate Sparsity

4.1. **Prediction in Linear Models using Approximate Sparsity.** We consider linear high dimensional approximate sparse regression models. These models have a large number of regressors p, possibly much larger than the sample size n, but only a relatively small number $s = o(n)$ 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.

The model reads:

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

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 errors ε_i are *not* restricted to be Gaussian or homoscedastic. This means that we allow for both non-Gaussian and heteroscedastic errors.

The model can be exactly sparse, namely

$$\|\beta_0\|_0 \leq s = o(n),$$

or approximately sparse, so that the values of coefficients, sorted in decreasing order, $(|\beta_0|_{(j)})_{j=1}^p$ obey,

$$|\beta_0|_{(j)} \leq \mathsf{A} j^{-\mathsf{a}}, \quad a > 1/2, \quad j = 1, ..., p.$$

An approximately sparse model can be well-approximated by an exactly sparse model with sparsity index

$$s \propto n^{1/(2\mathsf{a})}.$$

In order to get theoretically valid results in these settings, we consider the LASSO estimator with data-driven penalty loadings:

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

where $||\beta||_1 = \sum_{j=1}^p |\beta_j|$ and $\hat{\Psi} = \text{diag}(\hat{\psi}_1, \ldots, \hat{\psi}_p)$ is a diagonal matrix consisting of regressor dependent penalty loadings, and $\mathbb{E}_n$ abbreviates the empirical average. 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 in model errors. We discuss the choice of λ and $\hat{\Psi}$ below.

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

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

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 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 the resulting procedure the "rigorous"Lasso and hence add as prefix **r** to the function names.

In the case of homoscedasticity, we set the penalty loadings $\hat{\psi}_j = \sqrt{\mathbb{E}_n x_{i,j}^2}$, which insures basic equivariance properties. The are two choices for penalty level λ: the X-independent choice and X-dependent choice. In the X-independent choice we set the penalty level to

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

where Φ denotes the cumulative standard normal distribution, $\hat{\sigma}$ is a preliminary estimate of $\sigma = \sqrt{\mathbb{E}\varepsilon^2}$, and $c > 1$ is a theoretical constant , which is set to $c = 1.1$ by default, and γ is the probability level, which is set to $\gamma = .01$ by default. [1] In the X-dependent case the penalty level is calculated as

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

where

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

where $X = [x_1, \ldots, x_n]'$ and e_i are iid $N(0,1)$, generated independently from X; this quantity is approximated by simulation. The X-independent penalty is more conservative, while the X-dependent penalty level is least conservative. In particular the X-dependent penalty automatically adapts to highly correlated designs, using less aggressive penalization in this case.

In the case of heteroscedasticity, the loadings are given by $\hat{\psi}_j = \sqrt{\mathbb{E}_n[x_{ij}^2\hat{\varepsilon}_i^2]}$, where $\hat{\varepsilon}_i$ are preliminary estimates of the errors. The penalty level can be X-independent:

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

or it can be X-dependent and estimated by a multiplier bootstrap procedure.

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

where $c_W(1-\gamma)$ is the $1-\gamma$-quantile of the random variable W, conditional on the data, where

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

where $e_i$are iid standard normal distributed, indepently from the data, and $\hat{\varepsilon}_i$ denotes an estimate of the residuals.

The estimation proceeds by iteration. The estimates of residuals $\hat{\varepsilon}_i$ are initialized by running least squares of y_i on five most regressors that are most correlated to y_i. This implies conservative starting values for λ and the penalty loadings, and leads to the initial Lasso and post-Lasso estimates, which are then further updated by iteration.

[1] The probability is probability of mistakenly not removing some of the X's, when all of them have zero coefficients.

R implementation. The function `rlasso` implements Lasso and post-Lasso, where the prefix "r" signifies that these are theoretically rigorous versions of Lasso and post-Lasso. The default option is post-Lasso, `post=TRUE`. The user can also decide if an unpenalized `intercept` should be included (`TRUE` by default), `LassoShooting.fit` is the computational algoritm that underlies the estimation procedure, which implements a version of the Shooting Lasso Algorithm. 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.dependent.lambda = "TRUE"`. 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 following 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. This function returns an object of S3 class `rlasso` for which methods like `predict`, `print`, `summary` are provided.

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`.

Example. Consider generated data from a sparse linear model:

```
set.seed(1)
n = 100 #sample size
p = 100 # number of variables
s = 3 # nubmer of variables with non-zero coefficients
X = matrix(rnorm(n*p), ncol=p)
beta = c(rep(5,s), rep(0,p-s))
Y = X%*%beta + rnorm(n)
```

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

```
# use Lasso for fitting and prediction
lasso.reg = rlasso(Y~X,post=FALSE)  # use Lasso, not-Post-Lasso
print(lasso.reg, all=FALSE)
##
## Call:
## rlasso(formula = Y ~ X, post = FALSE)
##
##       X1        X2        X3       X83       X85
##  4.69459   4.83269   4.79304  -0.04094   0.01293
# summary(lasso.reg, all=FALSE) # use this option to summarize results
yhat.lasso = predict(lasso.reg)   #in-sample prediction
```

```
Xnew = matrix(rnorm(n*p), ncol=p)  # new X
Ynew =  Xnew%*%beta + rnorm(n)  #new Y
yhat.lasso.new = predict(lasso.reg, newdata=Xnew)  #out-of-sample prediction

# now use Post-Lasso for fitting and prediction
post.lasso.reg = rlasso(Y~X,post=TRUE)
print(post.lasso.reg, all=FALSE)    #use this option to print results
##
## Call:
## rlasso(formula = Y ~ X, post = TRUE)
##
##    X1     X2     X3
## 4.945  5.049  4.984
#summary(post.lasso.reg, all=FALSE)  #use this option to summarize results
yhat.postlasso = predict(post.lasso.reg)  #in-sample prediction
yhat.postlasso.new = predict(post.lasso.reg, newdata=Xnew)  #out-of-sample prediction
# compute Mean Absolute Error for Lasso and Post-Lasso predictions:
MAE<- apply(cbind(abs(Ynew-yhat.lasso.new), abs(Ynew - yhat.postlasso.new)),2, mean)
names(MAE)<- c("Lasso MAE", "Post-Lasso MAE")
print(MAE, digits=2)
##      Lasso MAE Post-Lasso MAE
##           0.84           0.77
```

The methods `print` and `summarize` have the option `all`, which by setting to `FALSE` shows only the non-zero coefficients.

5. Inference on Target Regression Coefficients in Regression, Using Approximate Sparsity

Here we consider inference on the target coefficient α in the model:

$$y_i = d_i\alpha_0 + x_i'\beta_0 + \epsilon_i, \quad \mathbb{E}\epsilon_i(x_i', d_i)' = 0.$$

We assume approximate sparsity for the part of the regression function $x_i'\beta$, with the sufficient speed of decay of the sorted components of β_0, namely $a_\beta > 1$. This translated into the effective sparsity index $s = o(n)$. In general d_i is correlated to x_i, so α_0 can not be consistently estimated by the regression of y_i on d_i. Write

$$d_i = x_i'\gamma_0 + v_i, \quad \mathbb{E}v_i x_i = 0.$$

To estimate α_0, we also impose approximate sparsity on the regression function $x_i'\gamma_0$, with the sufficent speed of decay of sorted coefficients of γ, namely $a_\gamma > 1$.

Note that we can not use naive estimates of α based simply on applying Lasso and Post-Lasso to the first equations. Such strategy in general does not produce root-n consistent and asymptotically normal estimators of α, because there is too much omitted variable bias resulting from estimating the nuisance function $x_i'\beta_0$ in high-dimensional setting. In order to overcome the ommitted variable bias we need to use orthogonalized estimating equations for α_0. These equations in turn rely on the classical ideas of partialling out.

5.1. **Intuition to Partialling Out.** One way to think about estimation of α is to think of the equation

$$u_i = \alpha v_i + \epsilon_i,$$

where u_i is the residual that is left after partialling out the linear effect of x_i from y_i and v_i is the residual that is left after partialling out the linear effect of x_i from d_i, both done in the population. Note that we have $\mathbb{E}u_i x_i = 0$, i.e. $u_i = y_i - x_i'\delta_0$ where $x_i'\delta_0$ is the linar projection of y_i on x_i. After partialling out, α is the regression coefficient in the univariate regression of u_i on v_i. This is the Frisch-Waugh-Lovell theorem.

In low-dimensional settings the empirical version of the partialling out approach is simply another way to do the least squares. Let's verify this in an example. First, we generate some data

```
set.seed(1)
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)
```

One way is to conduct a full OLS fit and then report the parameter of interest

```
# full fit
fmla = as.formula(paste("y ~ ", paste(colnames(X), collapse= "+")))
full.fit= lm(fmla, data=dat)
summary(full.fit)$coef["d",1:2]
```

```
##   Estimate Std. Error
## 0.97807455 0.01371225
```

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)
summary(partial.fit.ls)$coef["rD",1:2]
##   Estimate Std. Error
## 0.97807455 0.01368616
```

One can see that the estimates are identical, while standard errors are nearly identical, in fact they are asymptotically equivalent due to asymptotically negligible effect from estimating projection parameters on the first-order asymptotics of the partial least squares estimator α.

We can also try Lasso and Post-Lasso for partialling out. Let's see how this strategy would work in our low-dimensional example:

```
# partial fit via post-lasso
rY = rlasso(fmla.y, data =dat)$res
rD = rlasso(fmla.d, data =dat)$res
partial.fit.postlasso= lm(rY~rD)
summary(partial.fit.postlasso)$coef["rD",1:2]
##   Estimate Std. Error
## 0.97273870 0.01368677
```

We see that this estimate and standard errors are nearly identical to those given above. In fact they are asymptotically equivalent to those reported above in the low-dimensional settings.

In low-dimensional settings this is entirely expected, as any sensible way of approximately partialling out will work to produce first-order equivalent estimators of α_0. In the high-dimensional settings, we can no longer rely on the least-squares and have to rely on lasso and post-lasso for estimating the projection parameters. This results in estimators of α_0 that are root-n consistent and asymptotically normal, in fact achieving the semi-parametric efficiency bounds.

Let us note that the above strategy, based on partialling out via lasso or post-lasso is implemented in the package by the function `rlassoEffect`.

```
Eff= rlassoEffect(X[,-1],y,X[,1], method="partialling out")
summary(Eff)$coef[,1:2]
##  Estimate. Std. Error
## 0.97273870 0.01368677
```

Another first-order equivalent strategy implemented in the package relies on the double-selection method, which uses the union of $x_{i,j}$'s selected in two projection equations for partialling out. This is equivalent to doing OLS of y_i on d_i and the union of controls selected in the two projection equations:

(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.

We can implement this version by `rlassoEffect` by setting the method option to "double selection":

```
Eff= rlassoEffect(X[,-1],y,X[,1], method="double selection")
summary(Eff)$coef[,1:2]
##  Estimate. Std. Error
## 0.97807455 0.01415624
```

5.2. **Inference.** The function `rlassoEffects` does inference for target 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. `rlassoEffects` is a wrap function for `rlassoEffect` which does inference for a single target regressor.

First we generate data in a sparse linear model:

```
set.seed(1)
n = 100 #sample size
p = 100 # number of variables
s = 3 # nubmer of non-zero variables
X = matrix(rnorm(n*p), ncol=p)
beta = c(rep(3,s), rep(0,p-s))
y = 1 + X%*%beta + rnorm(n)
```

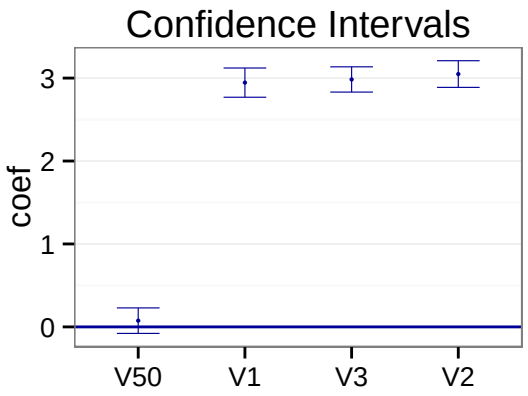
We can do inference on a set of variables of intest, e.g. the first, second, third, and the fiftieth:

```
lasso.effect = rlassoEffects(x=X, y=y, index=c(1,2,3,50))
print(lasso.effect)
##
## Call:
## rlassoEffects(x = X, y = y, index = c(1, 2, 3, 50))
##
## Coefficients:
##      V1       V2       V3      V50
## 2.94547  3.04876  2.98373  0.07519
summary(lasso.effect)
## [1] "Estimation of the effect of selected variables in a high-dimensional regression"
##     Estimate. Std. Error t value Pr(>|t|)
## V1    2.94547    0.08985  32.784   <2e-16 ***
## V2    3.04876    0.08202  37.169   <2e-16 ***
```

```
## V3    2.98373    0.07767  38.416   <2e-16 ***
## V50   0.07519    0.07845   0.958    0.338
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
confint(lasso.effect)
##          2.5 %    97.5 %
## V1   2.7693798 3.1215660
## V2   2.8879933 3.2095182
## V3   2.8315002 3.1359565
## V50 -0.0785745 0.2289456
```

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

```
plot(lasso.effect, main="Confidence Intervals")
```



5.3. **Application: Estimation of the treatment effect in a linear model with many confounding factors.** A part of empirical growth literature has focused on 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 (5.3) 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,drop=F]
d = GrowthData[,3, drop=F]
X = as.matrix(GrowthData)[,-c(1,2,3)]
varnames = colnames(GrowthData)
```

Now we can estimate the influence of initial GDP level to test the convergence hypothesis. First, we estimate by OLS:

```
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= "+")))
ls.effect= lm(fmla, data=GrowthData)
```

Second, we estimate by the partialling out method by (post)-Lasso:

```
dX = as.matrix(cbind(d,X))
lasso.effect = rlassoEffect(x=X, y=y, d=d)
summary(lasso.effect)
## [1] "Estimation of the effect of selected variables in a high-dimensional regression"
##           Estimate. Std. Error t value Pr(>|t|)
## xgdpsh465  -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, 2, 2)
table[1,]= summary(ls.effect)$coef["gdpsh465",1:2]
table[2,]= summary(lasso.effect)$coef[,1:2]
colnames(table)= c("Estimate", "Std. Error") #names(summary(full.fit)£coef)[1:2]
rownames(table)= c("full reg via ols",  "partial reg
via post-lasso ")
tab= xtable(table, digits=c(2, 2,7))
```

```
tab
```

	Estimate	Std. Error
full reg via ols	-0.01	0.0298877
partial reg via post-lasso	-0.04	0.0168463

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

$$y_i = \alpha_0 d_i + \beta_0 x_i' + \varepsilon_i, \tag{1}$$

$$d_i = z_i'\Pi + \gamma_0 x_i' + v_i, \tag{2}$$

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.

6.1. **Inference.** To get efficient estimators and uniformly valid confidence intervals for the strucutural parameters there are different strategies which are asymptotically equivalent where again partialling out is a key concept.

For inference in a setting with many instruments (and low-dimensional exogenous variables) in a first stage predicted values $\hat{d}$ are calculated based on a (Post-)LASSO regression of d on the instruments (z). In a second step, these predicted values are plugged into the first equation and then the coefficient of the treatment effect α_0 is estimated by orindary least squares. An alternative, asympotically equivalent approach would be in a preliminary step to partial out the effect of the exogenoues variables and then proceed as before.

In the case of many exogenous variables (and "few"instruments) we employ the following estimation strategy: First, the effect of the exogenous variables is partialled out from the dependent variable, the treatment variable and the instruments by (Post-)LASSO. Second, the resulting residuals are used to conduct a two-stage least squares regression with no exogenous variables included.

Finally, in the case of many instruments and many x-variables the treatment effect is estimated with the following algorithm: Algorithm 1. *(1) Do Lasso or Post-Lasso Regression of d_i on x_i, z_i to obtain $\hat{\gamma}$ and $\hat{\delta}$. (2) Do Lasso or Post-Lasso Regression of y_i on x_i to get $\hat{\theta}$. (3) Do Lasso or Post-Lasso Regression of $\hat{d}_i = x_i'\hat{\gamma} + z_i'\hat{\delta}$ on x_i to get $\hat{\vartheta}$. (4) Let $\hat{r}_i^y := y_i - x_i'\hat{\theta}$, $\hat{r}_i^d := d_i - x_i'\hat{\vartheta}$, and $\hat{v}_i := x_i'\hat{\gamma} + z_i'\hat{\delta} - x_i'\hat{\vartheta}$. Get estimator $\hat{\alpha}$ by using standard IV regression of $\hat{r}_i^y$ on $\hat{r}_i^d$ with $\hat{v}_i$ as the instrument. Perform inference on α using $\hat{\alpha}$ and conventional heteroscedasticity robust standard errors.*

R Implementation. 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. The fuctions `rlassoSelectX` and `rlassoSelectZ` do selection on either the X-variables resp. on the Z-variables. Selection on both is done in `rlassoIV`. All functions all the option `post` with default `TRUE` which enables post-LASSO at the corresponding steps. Further arguments, like `intercept` can be passed through to the underlying `rlasso` function with the `...`-mechanism. Finally, a function `tsls` is contained which allows "classical"two-stage least squares estimation where an `intercept` can be specified and it can be chosen between homoscedastic and heteroscedastic errors where the default `homoscedastic=TRUE`.

6.2. **Applications.**

6.2.1. *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(~ -1 + (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)
## Error:  ncol(X) == length(object$coefficients) ist nicht TRUE
AJR.Xselect = rlassoIVselectX(x=X, d=d, y=y, z=z, intercept=TRUE, post=TRUE)
## Error:  ncol(X) == length(object$coefficients) ist nicht TRUE
print(AJR.Xselect)
## Error in print(AJR.Xselect):  Objekt 'AJR.Xselect' nicht gefunden
summary(AJR.Xselect)
## Error in summary(AJR.Xselect):  Objekt 'AJR.Xselect' nicht gefunden
confint(AJR.Xselect)
## Error in confint(AJR.Xselect):  Objekt 'AJR.Xselect' nicht gefunden
```

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

```
# parialling out by lasso
rY = rlasso(fmla.y, data = AJR)$res
rD = rlasso(fmla.d, data = AJR)$res
rZ = rlasso(fmla.z, data = AJR)$res
ivfit.lasso = tsls(y=rY,d=rD, x=NULL, z=rZ, intercept=FALSE)
```

Partialling out with LASSO is implemented in the function `rlassoSelectX` so that both procedures give the same results. We see that by doing selection on the exogenous variables, the effect of institutions is decreased but the selection enables a much more precise estimation leading to small standard errors.

6.2.2. *Eminent Domain Decisions.* 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). Here we consider the regional GDP as outcome variable.

```
data(EminentDomain)
Z <- EminentDomain$logGDP$z
```

```
X <- EminentDomain$logGDP$x
y <- EminentDomain$logGDP$y
d <- EminentDomain$logGDP$d
```

As mentioned above, y is the Economic outcome, the logarithm of the GDP, 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:

```
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. In both cases post-LASSO and an intercept are included as they are default options.

```
lasso.IV.Z = rlassoIV(x=X, d=d, y=y, z=Z, select.X=FALSE, select.Z=TRUE) # wrap function
#lasso.IV.Z = rlassoIVselectZ(x=X, d=d, y=y, z=Z) #direct call
print(lasso.IV.Z)
##
## Call:
## rlassoIVselectZ(x = x, d = d, y = y, z = z, post = post)
##
## Coefficients:
##      d1
## 0.01659
summary(lasso.IV.Z)
## [1] "Estimation of the effect of selected variables in a high-dimensional regression"
##     coeff.     se. t-value p-value
## d1 0.01659 0.01775   0.934    0.35
confint(lasso.IV.Z)
##          2.5 %     97.5 %
## d1 -0.01821032 0.05138162
```

Finally, we do selection on both the x and z variables.

```
lasso.IV.XZ = rlassoIV(x=X, d=d, y=y, z=Z, select.X=TRUE, select.Z=TRUE) # wrap function
print(lasso.IV.XZ)
##
## Call:
## rlassoIV(x = X, d = d, y = y, z = Z, select.Z = TRUE, select.X = TRUE)
```

```
##
## Coefficients:
##        d1
## -0.005441
summary(lasso.IV.XZ)
## Estimation of the effect of selected variables in a high-dimensional IV regression
##       coeff.       se. t-value p-value
## d1 -0.005441  0.188125  -0.029   0.977
confint(lasso.IV.XZ)
##         2.5 %    97.5 %
## d1 -0.3741584 0.3632765
```

Comparing the results we see, that the OLS estimates indicate that the influence of pro plaintiff appellate takings decisions in federal circuit court is siginifcantly positive. But the 2SLS estimates which account for the potential endogeneity render the results insignificant. Employing selection on the instruments confirms this result. Selection on both the x- and z-variables drives the estimate below zero, but yielding similar economic conclusions.

Finally, we compare all results

```
tab
```

	Estimate	Std. Error
ols regression	0.01	0.0052848
IV estimation	0.02	0.0183376
selection on Z	0.02	0.0177534
selection on X and Z	-0.01	0.1881246

7. Inference on Treatment Effects in a High-Dimensional Setting

7.1. **Treatment Effects – A short Introduction.** In many situations researchers are asked to evaluate the effect of a policy intervention. Examples are the effectiveness of a job-related training programme or the effect of a newly developed drug. We consider n units or individuals, $i = 1, \ldots, n$. For each individual we observe the treatment status. The treatment variable D_i takes the value 1, if the unit received (active) treatment, and 0, if it received the control treatment. For each individual we observe the outcome for only one of the two potential treatment states. Hence, the observed outcome depends on the treatment status and is denoted by $Y_i(D_i)$.

A quantity of interest is the average treatment effect (ATE) which is defined as

$$\mathbb{E}[Y(1) - Y(0)] = \mathbb{E}[Y(1)] - \mathbb{E}[Y(0)].$$

This quantity can be interpreted as the average effect of the policy intervention.

Researchers might also be interested in the average treatment effect on the treated (ATET) given by

$$\mathbb{E}[Y(1) - Y(0)|D = 1] = \mathbb{E}[Y(1)|D = 1] - \mathbb{E}[Y(0)|D = 1].$$

This is the average treatment effect restricted to the subsample of the treated individuals. Under certain assumptions those quantities can be estimated unbiased.

In observational studies the outcome variable often depends on the treatment variable given the conditioning variables. In such situations the local average treatment effect (LATE) can be of interest which can be estimated when a binary instrumental variable is available and is defined as

$$\mathbb{E}[Y(1) - Y(0)|D(1) > D(0)].$$

The random variables $D(1)$ and $D(0)$ indicate the potential participation decisions under the instrument states 1 It can be interpreted as the average treatment effect for the subgroup of compliers. The compliers are the individuals that take the treatment when assigned and the control treatment when not assigned. In an analog way, the the local average treatment effect can be restricted to the treated, yielding the so-called local average treatment effect of the treated (LATET):

$$\mathbb{E}[Y(1) - Y(0)|D(1) > D(0)|D = 1].$$

7.2. **Estimation and Inference of Treatment effects.** We consider the estimation of the effect of an endogenous binary treatment, D, on an outcome variable, Y, in a setting with very many potential vontrol variables. In the case of endogeneity, the presence of a binary instrumental variable, Z, is required for the estimation of the LATE and LATET.

When trying to estimate treatment effects, the researcher has to decide what conditioning variables to include. In the case of a non-randomly assigned treatment or instrumental variable, the researcher must select the conditioning variables so that the instrument or treatment is plausibly exogenous. Even in the case of random assignment, for a precise estimation of the policy variable selection of control variables is neccessary to absorb residual variation, but overfitting should be avoided. For uniformly valid post-selection inference, "low-bias"moment functions and approximate sparsity of the underlying structural parameters are key in a high-dimensional setting where the number of conditioning variables might surmount the number of observations. The estimation strategy which is implemented in the packages employs LASSO regressions to estimate the structural quantities which replicate the sparse structure.

R Impementation. The package contains the functions `rlassoATE`, `rlassoATET`, `rlassoLATE`, and `rlassoLATE` to estimate the corresponding treatment effects. All functions have as arguments the outcome variable y, the treatment variable d, and the conditioning variables x. The functions `rlassoLATE`, and `rlassoLATE` have additionally the argument z for the binary instrumental variable. For the calculation of the standard errors bootstrap methods are implemented. The choice options are `Bayes`, `normal`, and `wild` bootstrap. The number of repitions can be specified with the argument `nRep` and the default is set to 500. By default no booststrap standared errors are provided (`bootstrap="none"`). For

the functions the logicals `intercept` and `post` can be specified to include an intercept and to do post-LASSO at the selection steps. The family of treatment functions returns an object of class `rlassoTE` for which the methods `print`, `summary`, and `confint` are available.

7.3. **Application: 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)[,-1]
```

Now we can calculate the interesting treatment effects

```
pension.late = rlassoLATE(X,d,y,z,
                          bootstrap="none", post=TRUE, intercept=TRUE)
print(pension.late)
##
## Call:
## rlassoLATE(x = X, d = d, y = y, z = z, bootstrap = "none", post = TRUE,
##     intercept = TRUE)
##
## Treatment Effect
## Type: LATE
## Value:
## [1]  5384
pension.latet = rlassoLATET(X,d,y,z,
                            bootstrap="none", post=TRUE, intercept=TRUE)
print(pension.latet)
##
```

```
## Call:
## rlassoLATET(x = X, d = d, y = y, z = z, bootstrap = "none", post = TRUE,
##     intercept = TRUE)
##
## Treatment Effect
## Type: LATET
## Value:
## [1]  2112
pension.ate =  rlassoATE(X,d,y,
                         bootstrap="none", post=TRUE, intercept=TRUE)
print(pension.ate)
##
## Call:
## rlassoLATE(x = x, d = d, y = y, z = z, bootstrap = bootstrap,
##     nRep = nRep, post = TRUE, intercept = TRUE)
##
## Treatment Effect
## Type: ATE
## Value:
## [1]  6787
pension.atet =  rlassoATET(X,d,y,
                           bootstrap="none", post=TRUE, intercept=TRUE)
print(pension.atet)
##
## Call:
## rlassoLATET(x = x, d = d, y = y, z = z, bootstrap = bootstrap,
##     nRep = nRep, post = TRUE, intercept = TRUE)
##
## Treatment Effect
## Type: ATET
## Value:
## [1]  2675
```

The results are summarized in the following

```
tab
```

	Estimate	Std. Error
ATE	6786.65	2956.4811601
ATET	2675.24	8187.0595142
LATE	5383.90	5023.5001893
LATET	2112.31	12206.9027803

8. 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 .