HIGH-DIMENSIONAL MICROECONOMETRICS IN R: A VIGNETTE

VICTOR CHERNOZHUKOV, CHRISTIAN HANSEN, MARTIN SPINDLER

Abstract. High-dimensional Microeconometrics (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 introduction to the package. R and the package `HDM` are open-source software projects and can be freely downloaded from CRAN: `http://cran.r-project.org`.

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, 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.

Section 2 shows how to get started with `HDM`. Section 3 describes how to estimate a (Post-)LASSO regression with data-dependent loadings. This function is key as all other functions are based on fitting some kind of LASSO regression. Section 4–6 describe functions to estimate treatment effects in a setting with many controls, many instruments and a combination of both.

2. Getting 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 dowloaded from `cran.r-project.org`. To install the HDM package from R one simply types,

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

Version: April 21, 2015.

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 datails, 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. LASSO and Post-LASSO Estimation under Heteroscedastic and Non-Gaussian Errors

The function lasso estimates a (Post-)LASSO regression under heteroscedastic and non-Gaussian errors. For technical details we refer to (Belloni, Chen, Chernozhukov, and Hansen 2012) who analyse LASSO under heteroscedastic and non-Gaussian errors and (Belloni and Chernozhukov 2013) who analyse the properties of the Post-LASSO estimator. The idea of Post-LASSO is to refit the model by OLS regression with the variables which were selected by a preliminary LASSO regression. This procedure reduces under certain circumstances bias of the estimated coefficients which is typical for LASSO.

In order to demonstrate how the function lasso works, we generate a data set of sample size $n = 250$ which contains 100 variables, but only 10 of them have an influence:

```
> n <- 250
> p <- 100
> px <- 10
> X <- matrix(rnorm(n*p), ncol=p)
> beta <- c(rep(2,px), rep(0,p-px))
> y <- X %*% beta + rnorm(n)
```

We estimate a LASSO regression by the following commands:

```
> lasso.reg <- lasso(x=X, y=y, post=TRUE, intercept=TRUE, normalize=TRUE,
+       control=list(c = 1.1, gamma = 0.1, numIter = 15, tol = 10^-5, lambda = "standard", nu
> lasso.reg <- lasso(y~X)
```

The function lasso can be called in two different modes: either with arguments x and y for the designmatrix and the response or with a formula object. (The function `lasso` is a generic function with function dispatch according to the object type of the arguments.)

The function `lasso` returns an object of S3 class, called `lasso`. For the S3 class `lasso` the following standard methods are available: `summary`, `print`, `predict`, and by default `coefficients` and `residuals`. The first two methods have the `option` `all` with default value `TRUE`. Setting the value to `FALSE` shows only the results for the non-zero coefficients. In a setting with many variables this might increase clarity.

```
> print(lasso.reg, all=FALSE)

Call:
lasso.formula(formula = y ~ X)

   X1     X2     X3     X4     X5     X6     X7     X8
2.064  2.097  2.118  2.056  2.058  1.917  2.047  1.996
   X9    X10
1.984  2.080

> summary(lasso.reg, all=FALSE)

Call:
lasso.formula(formula = y ~ X)

Post-Lasso Estimation:  TRUE

Total number of variables: 100
Number of selected variables: 10

Residuals:
     Min       1Q   Median       3Q      Max
-2.33747 -0.61771 -0.01416  0.60797  2.24521

    Estimate
X1     2.064
X2     2.097
X3     2.118
X4     2.056
X5     2.058
X6     1.917
X7     2.047
X8     1.996
X9     1.984
X10    2.080

Residual standard error: 0.8956

> yhat <- predict(lasso.reg)
> Xnew <- matrix(rnorm(n*p), ncol=p)
> yhat.new <- predict(lasso.reg, newdata=Xnew)
```

The function `lasso` has many options which will be explained now in more detail. The option `post` indicates if Post-LASSO is estimated. The default value is `TRUE`.

If an intercept should be included and be estimated this can be adjusted by the option `intercept`. The value `TRUE` effects that the mean is substracted both from the dependent variable and the independent variables. Often the regressors are measured on different scales which influences the results of the LASSO regression, as LASSO is not scale-invariant. With the option `normalize=TRUE` the regressors can be normalized with variance equal to one.

An important choice is the choice of penalization parameter λ and regressor specific loadings. For the choice of the penalization parameter three procedures are possible: "X-dependent" "X-independent" and "standard" . The last two choices give quite similar results. The "X-dependent" method relies on a simulation method for which the number of simulations `numSim` can be specified. c and γ are constants used for the calcuation of λ. c is required to be larger than 1 (1.1 by default) and γ is a constant in $(0, 1)$. For details of the calculation of the penalization parameter we refer to (Belloni, Chernozhukov, and Hansen 2010). The calculation of the loadings follows the Appendix A in (Belloni, Chen, Chernozhukov, and Hansen 2012) to which we refer for a detailed explanation. `numIter` denotes the number of iterations of the the algorithm and `tol` gives a tolerance so that the algorithm breaks when the improvement in absolute parameter values is below the tolerance.

The last option (`lambda.start`) is to handle a vector of initial parameter values to the function.

Fitting of the LASSO regression is done in the function `LassoShooting.fit` which implements the shooting LASSO (Fu 1998) with regressor dependent penalization parameters (the product of the penalization parameter λ and the loadings).

4. Estimation and Inference on Structural Effects in High-dimensional Settings

In many situations a researcher is concerned with estimating and making inference on a structural effect / treatment effect. If the treatment variable is endogenous which arises often in observational studies in Economic applications, there are several ways to deal with the endogeneity of the treatment variable. One is to include additional control variables / observables so that conditional on those variables the treatment will be plausibly exogenous. Another method is to use instrumental variables to estimate the treatment effect. Finally, a combination of both approaches is possible. In any case, the researcher must decide which variables to include as controls and as instruments. Including all available variables might lead to overfitting and bad statistical properties of the resulting estimator, especially in small samples. In the next sections we will present functions for estimation of treatment effects in a setting with many controls, many instrumental variables and both. For an in-depth treatment of those methods we refer the reader to the original articles, namely (Belloni, Chernozhukov, and Hansen 2014), (Belloni, Chen, Chernozhukov, and Hansen 2012), and, (Chernozhukov, Hansen, and Spindler 2015).

4.1. **Estimation and Inference on Structural Effects with High-dimensional Controls.** In this section we show how to do estimation in setting with high-dimensional controls. First, we generate a data set. The function `DGP.HC` generates a data set as used for the simulation study in (Belloni, Chernozhukov, and Hansen 2014). Estimation is done by the function `HC.lasso` as the following example shows:

```
>  data <- DGP.HC()
>   y <- data$y
>   d <- data$d
>   x <- data$X
>   l <- HC.lasso(x,y,d, I3=NULL, post= TRUE, intercept = TRUE, normalize=FALSE,
+          control=list(c=1.1, gamma=.1, numIter=15, tol=10^-5, lambda="standard", lambda.star
>   l$coef
```

```
(Intercept)         xd1         xx1         xx3
 0.12660933  0.56955272  0.43999105  0.02180306
```

```
>   l$se
```

```
[1] 0.07059313
```

With the option `I3` (a logical vector) the researcher can specify variables in `x` which should be included in any case. All other arguments, following `I3`, are passed to the function `lasso`.

To show that the estimator has the desired properties we also conduct a small simulation study:

```
> R <- 1000 # number of repitions
> result <- matrix(NA, ncol=2, nrow=R) #matrix for the results
> colnames(result) <- c("estimate", "s.e.")
> for (i in 1:R) {
+   data <- DGP.HC()
+   l <- HC.lasso(data$X,data$y, data$d, I3=NULL, post= TRUE, intercept = TRUE, normalize=FAL
+   result[i,1] <- l$alpha
+   result[i,2] <- l$se
+ }
```

Finally, we plot the centered and rescaled estimated values which should approximately normally distributed as given in Figure 4.1.

For the case that there is not only one but several treatment variables `d` two functions are provided: `HC.lasso.wrap` and `HC.lasso.mult`. The first one takes `x` and `y` as arguments and additionally an `index` which denotes the column number of variables in `x` which should be handled as treatment variable. For each case then the function `HC.lasso` is applied with `d` selected accordingly. The function `HC.lasso.mult` takes as arguments `x,y` and `d` where the treatment can now be multivariate.

4.2. **Estimation and Inference on Structural Effects with High-dimensional Instruments.** In many empirical applications there are many potential instrumental variables and the researcher has to make a decision which one to include. Selection of instrumental variables in a high-dimensional setting has been analysed in (Belloni, Chen, Chernozhukov, and Hansen 2012). The function `IV.lasso` implements this procedure what we will demonstrate with an example. First, we simulate data with the function `DGP.IV` which implements the data generating process used in (Belloni, Chen, Chernozhukov, and Hansen 2012). Strictly speaking, the function implements the cut-off design with `pz` the number of variables with non-zero coefficients. A detailed description is given in (Belloni, Chen, Chernozhukov, and Hansen 2012) and in the help for the function. An example shows how to estimate the model:

```
> alpha.hat <- (result[,1]-0.5)/result[,2]
> hist(alpha.hat, breaks=25, freq=F, main="")
> curve(dnorm(x), add=TRUE)
```

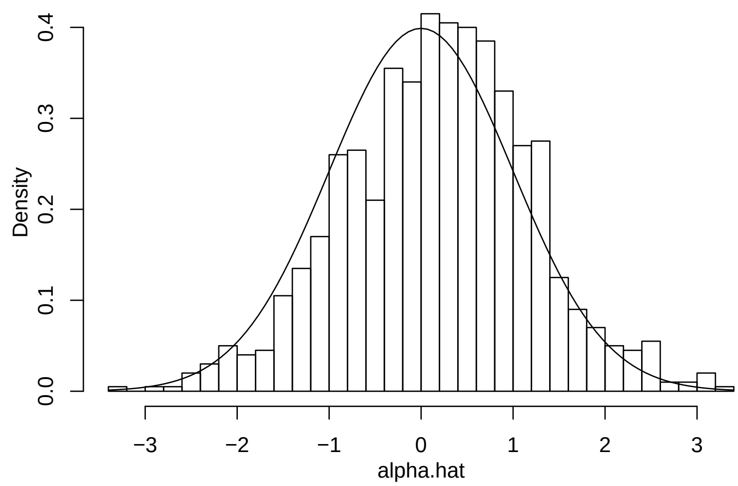

FIGURE 1. Histogram of the centered and rescaled estimated values for the treatment effect α; standard normal distribution is given by the solid line.

```
> data <- DGP.IV(n=250, p=100, pnz=5, Fstat=180,
+                control=list(s2e=1, Cev=.6, s2z=1, szz=.5, pi1=1, alpha=1))
>   y <- data$y
>   d <- data$X
>   z <- data$Z
>   x <- NULL
>   l <- IV.lasso(x,d,y,z, post=TRUE, intercept=TRUE, normalize=FALSE)
>   l$coefficients
       [,1]
d1 1.161421
>   l$vcov
           d1
d1 0.01315478
```

`x` denotes the control variables, `z` the instrumental variables, `d` the endogenous outcome variables, and `y` the outcome or dependent variable. Again, arguments, additionally to the `post`-option can be passed to the function `lasso` via the ...-mechanism.

Again, we demonstrate the asymptotic properties of the estimator in a small simulation study:

```
> R <- 1000
> result <- matrix(NA, ncol=2, nrow=R)
> for (i in 1:R) {
+   data <- DGP.IV(n=250, p=100, pnz=5, Fstat=180, control=list(s2e=1, Cev=.6, s2z=1, szz=.5,
```

```
+   l <- try(IV.lasso(NULL,data$X,data$y,data$Z, post=TRUE, intercept=TRUE, normalize=FALSE))
+   if (class(l)=="error") {
+     next
+   } else {
+     result[i,1] <- l$coef[1]
+     result[i,2] <- l$vcov[1,1]
+   }
+ }
```

Finally, we plot the centered and rescaled estimated values which should approximately normally distributed as shown in Figure 4.2.

```
> alpha.hat <- (result[,1]-1)/sqrt(result[,2])
> hist(alpha.hat, breaks=25, freq=F, main="")
> curve(dnorm(x), add=TRUE)
```

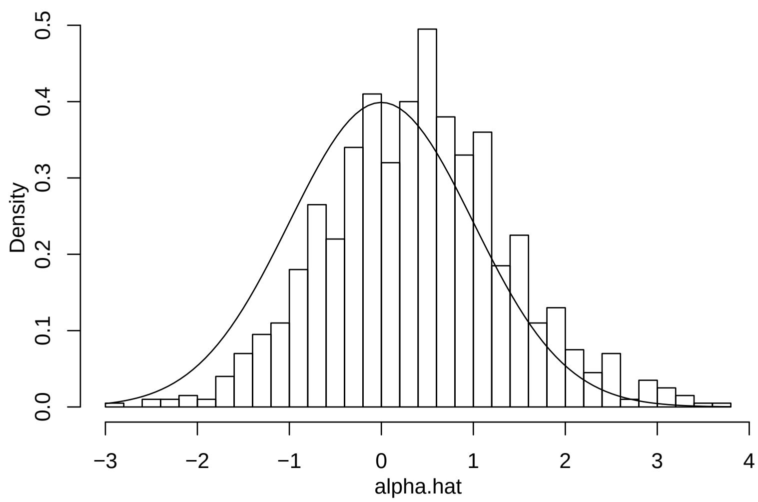

FIGURE 2. Histogram of the centered and rescaled estimated values for the treatment effect α; standard normal distribution is given by the solid line.

4.3. **Estimation and Inference on Structural Effects with High-dimensional Controls and Instruments.** Finally, in this section we combine the settings in the two previous sections to estimation in a setting with many controls and many instruments. A formal analysis is provided in (Chernozhukov, Hansen, and Spindler 2015). The method relying on double selection is implemented in the function `HCHIV.lasso`. To demonstrate how to apply this function we first simulate a data set with a dgp as described in (Chernozhukov, Hansen, and Spindler 2015) and then estimate it:

```
>  data <- DGP.HCHIV()
>   y <- data$y
>   d <- data$d
>   x <- data$X
>   z <- data$Z
```

```
>   res <- HCHIV.lasso(x,z,y,d)
>   res

$coefficient
         [,1]
[1,] 1.096856

$se
          [,1]
[1,] 0.1455782
```

A simulation example is as following

```
> R <- 1000
> result <- matrix(NA, ncol=2, nrow=R)
> for (i in 1:R) {
+   data <- DGP.HCHIV()
+   res <- try(HCHIV.lasso(data$X,data$Z,data$y,data$d))
+   if (class(res)=="error") {
+     next
+   } else {
+   result[i,1] <- res[[1]]
+   result[i,2] <- res[[2]]
+   }
+ }
```

with the empirical distribution of the centered and normalized estimates given in Figure 4.3.

5. Related R packages

R already contains many very useful packages for high-dimensional problems. In this section we review some packages which are related to our package but without any claim to be complete as currently the field is evolving very fast. A good starting point for the interested reader is the CRAN Task View `Machine Learning Statistical Learning` which points to many methods available as R packages.

Two classical and very established packages to estimate LASSO Regression are `glmnet` ((Friedman, Hastie, and Tibshirani 2010))and `lars` ((Hastie and Efron 2013)). `lars` fit Least Angle Regression, Lasso and Infinitesimal Forward Stagewise regression models, but does not allow data depenent penalties. `glmnet` fits generalized linear models via penalized maximum likelihood. The regularization path is computed for the elastic net penalty (elastic net contains LASSO as a special case) at a grid of values. A separate penalty factors can be applied to each coefficient. The `grplasso` ((**?**)grplasso)) estimates group LASSO models. A very recent R package is `flare` ((**?**)flare)) which implements a family of novel regression methods (Lasso, Dantzig Selector, LAD Lasso, SQRT Lasso, Lq Lasso) and their extensions to sparse precision matrix estimation (TIGER and CLIME using L1) in high dimensions.

```
> alpha.hat <- (result[,1]-1)/result[,2]
> hist(alpha.hat, breaks=25, freq=F, main="")
> curve(dnorm(x), add=TRUE)
```

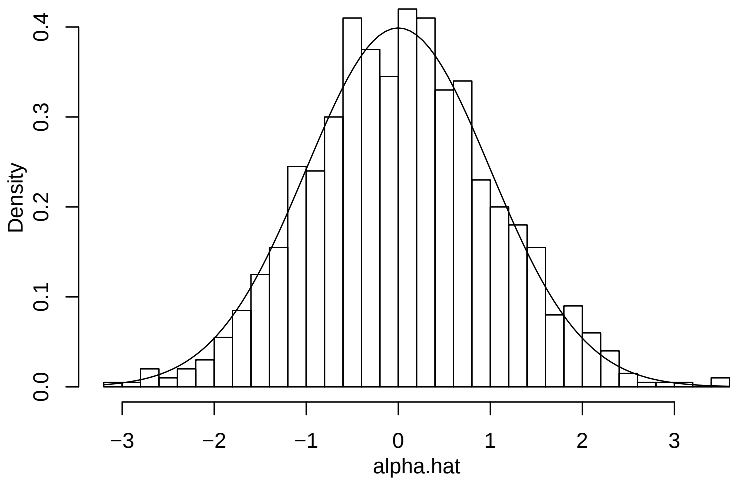

FIGURE 3. Histogram of the centered and rescaled estimated values for the treatment effect α; standard normal distribution is given by the solid line.

6. Some Tips and Tricks

7. 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, unforseen 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 periodcally. 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

Belloni, A., D. Chen, V. Chernozhukov, and C. Hansen (2012): "Sparse Models and Methods for Optimal Instruments with an Application to Eminent Domain," *Econometrica*, 80, 2369–2429, Arxiv, 2010.

Belloni, A., and V. Chernozhukov (2013): "Least Squares After Model Selection in High-dimensional Sparse Models," *Bernoulli*, 19(2), 521–547, ArXiv, 2009.

Belloni, A., V. Chernozhukov, and C. Hansen (2010): "Inference for High-Dimensional Sparse Econometric Models," *Advances in Economics and Econometrics. 10th World Congress of Econometric Society. August 2010*, III, 245–295, ArXiv, 2011.

——— (2014): "Inference on Treatment Effects After Selection Amongst High-Dimensional Controls," *Review of Economic Studies*, 81, 608–650, ArXiv, 2011.

Chernozhukov, V., C. Hansen, and M. Spindler (2015): "Post-Selection and Post-Regularization Inference in Linear Models with Many Controls and Instruments," *American Economic Review: Papers and Proceedings*, 105(5), 1–7.

Friedman, J., T. Hastie, and R. Tibshirani (2010): "Regularization Paths for Generalized Linear Models via Coordinate Descent," *Journal of Statistical Software*, 33(1), 1–22.

Fu, W. J. (1998): "Penalized Regressions: The Bridge Versus the Lasso," *Journal of Computational and Graphical Statistics*, 7, 397–416.

Hastie, T., and B. Efron (2013): *lars: Least Angle Regression, Lasso and Forward Stagewise*R package version 1.2.