Generalized Methods of Moments Estimation

Markus G. Schmidt
Universität Innsbruck

Abstract

The **GmmEst** package (https://R-Forge.R-project.org/projects/uibk-rprog-2017/) estimates the parameter vector of user defined models using the generlized methods of moments estimation framework. A brief overview of the package is provided, along with some illustrations.

Keywords: GMM, regression, R.

1. Introduction

The Generalized Method of Momentds (GMM) framework is one of the main tools to analyze economic and financial data, introduced by Hansen (1982) in the econometrics literature. In contrast to Maximum Likelihood estimation, GMM works without the need to specify the likelihood function. The main idea of the estimation framework is the usage of population moment conditions that are deduced from econometric models. The population moment condition is defined by

$$E[f(x_t, \theta_0)] = 0 \tag{1}$$

where θ_0 is a vector of unknown parameters which are to be estimated, x_t a vector of random variables and $f()$ a vector of functions.

The GMM estimator uses the sample counterpart of the population moment condition and chooses the estimated θ in such a way that the different moment conditions are as close to zero as possible. This is achieved by minimizing $Q_T(\theta)$

$$Q_T(\theta) = T^{-1} \sum_{t=1}^{T} f(x_t, \theta)\ W_T\ T^{-1} \sum_{t=1}^{T} f(x_t, \theta) \tag{2}$$

where W_T is a positive semi-definite matrix which may depend on the data but converges in probability to a positive definite matrix of constants.

See Greene (2011) for a short introduction into GMM estimation and Hall (2005) for a comprehensive treatment of GMM estimation. Cochrane (2005) discussess the application of GMM with a finance focus. Besides them, the seminal article by Hansen (1982) should not be missing in the list of references.

2. Implementation

In difference to many other packages for R (R Core Team 2017), the main model fitting function `GmmEst()` does not use a formula-based interface, but instead requires a user defined

function of the sample moment conditions.[1]. The function returns an (S3) object of class `GmmEst`:

```
GmmEst(func, theta0, data,
    est_type=c("2step","1step","iter"),
    func_jac=NULL, initial_W=NULL,
    crit=10e-7, itermax=100,
    optim_method=c("BFGS","Nelder-Mead","L-BFGS-B"),
    control = GmmEst_control(\dots), \dots)
```

A number of standard S3 methods are provided, see Table 1.

The user supplied function of the sample moment conditions `func` should have the form `func(param,data)` and should return a number of observations times number of moments matrix - corresponding to the sample counterpart of $f(x_t, \theta)$. See the next section for two examples.

3. Illustration

3.1. An easy example: GMM can do OLS

Suppose we want to estimate a simple linear model.

$$y_i = \alpha + \beta \cdot x_i + \varepsilon_i \tag{3}$$

The typical OLS assumptions are that $E[\varepsilon_i] = 0$ and $E[\varepsilon_i x_i] = 0$. We can use that assumption to form the population moment conditions:

$$0 = \quad E[y_i - \alpha_0 - \beta \cdot x_i] \tag{4}$$
$$0 = \quad E[(y_i - \alpha_0 - \beta \cdot x_i) \cdot x_i] \tag{5}$$

[1] A formula-based interface will be implemented for linear models in the near future.

Method	Description
`print()`	Simple printed display with coefficients
`summary()`	Standard summary; returns `summary.GmmEst` object (with `print()` method)
`coef()`	Extract coefficients
`vcov()`	Associated covariance matrix
`nobs()`	Extract number of observations
`bread()`	Extract bread for **sandwich** covariance
`estfun()`	Extract estimating functions (= gradient contributions) for **sandwich** covariances

Table 1: S3 methods provided in **GmmEst**.

as well as the the sample counterparts:

$$0 = \quad E_N[y_i - a - b \cdot x_i] \tag{6}$$
$$0 = \quad E_N[(y_i - a - b \cdot x_i) \cdot x_i] \tag{7}$$

where E_N is the short form for $N^{-1} \sum_{i=1}^{N}()$ and $\theta_0 = (\alpha, \beta)$ and $\theta = (a, b)$. Defined in such a way, we have 2 unknown parameters and 2 moment conditions such that the model is just identified.

Let's use the standard `mtcars` dataset to estimate a simple linear regression of miles per gallons on horse power by the efficient two-step feasible GMM estimator made popular by Hansen (1982). I start by loading the data and by defining the sample moment condition function $f(x_i, \theta)$.

```
data = mtcars
mom_cond = function(theta, data){
  a = theta[1]
  b = theta[2]
  y = data$mpg
  x = data$hp

  u = y - a - b*x
  return(cbind(u,u*x))
}
```

While not necessary to estimate the model, especially in that case, the so called d-matrix, which is the gradient/jacobian of the sample moment conditions w.r.t. the parameters ($E_N[\frac{\partial f(x_i,\theta_0)}{\partial \theta_0}]$) is genereally important. If not supplied by the user, it will be numerically approximated. If supplied, it is used in the minimization as well. Although the jacobian in that case does not depend on the parameter itself, the function is defined with the same arguments as the moment condition function.

```
mom_cond_grad = function(theta, data){
  x = data$hp

  d1 = -1
  d2 = -mean(x)
  d3 = -mean(x)
  d4 = -mean(x^2)

  d = matrix(c(d1,d2,d3,d4),nrow=2,ncol=2)
  return(d)
}
```

Using the $a = 20$ and $b = 0$ as starting values, we can simply estimate the model by

```
theta0 = c(20,0)
mdl_gmm = GmmEst(mom_cond, theta0, mtcars, est_type = "2step",
                 optim_method = 'BFGS', func_jac = mom_cond_grad)
summary(mdl_gmm)

## model did converge
##
## Coefficients:
##      Estimate Std. Error z value Pr(>|z|)
## [1,] 30.09886    2.07661  14.494  < 2e-16 ***
## [2,] -0.06823    0.01356  -5.031 4.87e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Model is just-identified
## J-statistic: 1.547e-16 , p-value: NA
```

and compare it to a an OLS regression:

```
mdl_lm = lm(mpg~hp,data)
coeftest(mdl_lm, df = Inf, vcov = vcovHC, type = "HC1")

##
## z test of coefficients:
##
##              Estimate Std. Error z value  Pr(>|z|)
## (Intercept) 30.098861   2.076615 14.4942 < 2.2e-16 ***
## hp          -0.068228   0.013560 -5.0314 4.868e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The estimated coefficients are identical as well as the standard errors, which are by default robust standard errors w.r.t. heteroskedasticity in `GmmEst`.

3.2. The standard example: Estimation of the risk-aversion parameter in the Consumption based Asset Pricing Model

Assuming power utility for an representative investor, the consumption based asset pricing model predicts that

$$E_t[\beta \frac{C_{t+1}}{C_t}^{-\gamma} R^e_{t+1}] = 0 \tag{8}$$

where C is consumption in period t and $t+1$, γ is the risk-aversion parameter, β is the time-preference parameter and R^e_{t+1} is an excess return of an (any) asset. Furthermore, E_t denote the expectation conditional on time t information. In finance, one is often interested in estimating this Euler equation as it is important w.r.t. the equity premium puzzle.

The package `GmmEst` contains a dataset that can be used to estimate the model.[2] In the dataset with 63 yearly observations from 1952 to 2014, the gross consumption growth in the US is included as well as the (excess net) returns of some widely known asset portfolios $(rmrf, smb, hml)$ and the (net) risk free rate (rf). Using the market portfolio $rmrf$ and the high-minus-low portfolio hml, we can estimate γ, fixing $\beta = 1$ and hence estimating the model assuming no time-preference by the representative investor. In the following, I load the data and define the moment conditions as well as the gradient of the moment conditions and estimate the parameter using a one-step GMM estimator.

```
data("data_consumption",package = "GmmEst")
mom_cond = function(params, data){
  gamma = params[1]

  beta = 1
  dc = data$dc
  rmrf = data$rmrf
  hml = data$hml
  rf = data$rf

  gt1 = beta*dc^(-gamma)*rmrf
  gt2 = beta*dc^(-gamma)*hml
  gt = cbind(gt1, gt2)
  return(gt)
}

mom_cond_grad = function(params,data){
  gamma = params[1]

  beta = 1
  dc = data$dc
  rmrf = data$rmrf
  hml = data$hml

  d1 = -dc^(-gamma)*log(dc)*rmrf*beta
  d2 = -dc^(-gamma)*log(dc)*hml*beta
  d = c(mean(d1),mean(d2))
  return(d)
}

mod1 = GmmEst(mom_cond, 100, data, est_type = "1step",
              optim_method = 'BFGS', func_jac = mom_cond_grad)
summary(mod1)

## model did converge
##
```

[2] See the descrition of the dataset for further information about the variables.

```
## Coefficients:
##      Estimate Std. Error z value Pr(>|z|)
## [1,]    80.96      52.20   1.551    0.121
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Model is over-identified
## J-statistic: 0.02543 , p-value: 0.873
```

The estimated value of 80.96 is huge and the heart of the equity premium puzzle as it needs a huge value of γ to make sense of the high equity premium. Nevertheless, it is not statically significant if we put the same weight initial weight on the both portfolios. The J-test of overidentification does not reject the model and tells us that the pricing errors are not jointly significant different from zero.

If we apply the efficient two-step GMM procedure, more weight is given to the moments that are better measured, i.e. the *hml* portfolio is less volatile than $rmrf$ to get a more efficient estimate of γ in a statistical sense.

```
mod2 = GmmEst(mom_cond, 100, data, est_type = "2step",
              optim_method = 'BFGS', func_jac = mom_cond_grad)
summary(mod2)

## model did converge
##
## Coefficients:
##      Estimate Std. Error z value Pr(>|z|)
## [1,]    88.34      33.41   2.644   0.0082 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Model is over-identified
## J-statistic: 0.0263 , p-value: 0.871
```

While the estimate is now a higher with 88.34, the standard error nearly cut in half. The J-test changed only slightly and still doesn't reject the model.[3]

References

Cochrane JH (2005). *Asset Pricing*. Revised edition. Princeton University Press.

Greene WH (2011). *Econometric Analysis*. 7 edition. Prentice Hall.

Hall AR (2005). *Generalized Method of Moments*. Advanced texts in econometrics, 1 edition. Oxford University Press.

[3] HAC standard errors will be implemented in the near future.

Hansen LP (1982). "Large sample properties of generalized method of moments estimators." *Econometrica*, **50**(4), 1029–1054.

R Core Team (2017). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/.

Affiliation:

Markus G. Schmidt
Department of Banking and Finance
School of Management
Universität Innsbruck
Universitätsstr. 15
6020 Innsbruck, Austria
E-mail: markus.g.schmidt@uibk.ac.at