Generalized Empirical Likelihood with R

Pierre Chaussé

November 4, 2019

Abstract

This an extention of the gmmS4 vignette, to explain how to use the package for generalized empirical likelihood estimation.

Contents

1 A very brief review of the GEL method

We present how to use the package to estimate models by the Generalized Empirical Likelihood method (GEL) (see Newey and Smith (2004) for the iid case and Anatolyev (2005) for weakly dependent processes). We assume that the reader has read the gmmS4 vignette in which many classes and methods needed are defined. We first describe the method without going into too much details. The author can refer to the above papers for a detailed description, or Chaussé (2010) who explains GEL estimation using the gmm (Chaussé (2019))package.

The estimation is based on the following moment conditions

$$\mathrm{E}[g_i(\theta)] = 0,$$

For the iid case, the estimator is defined as the solution to either

$$\hat{\theta} = \arg\min_{\theta, p_i} \sum_{i=1}^{n} h_n(p_i)$$

subject to,

$$\sum_{i=1}^{n} p_i g_i(\theta) = 0$$

and

$$\sum_{i=1}^{n} p_i = 1,$$

where $h_n(p_i)$ belongs to the following Cressie-Read family of discrepancies:

$$h_n(p_i) = \frac{[\gamma(\gamma+1)]^{-1}[(np_i)^{\gamma+1} - 1]}{n},$$

or

$$\hat{\theta} = \arg\min_{\theta} \left[\max_{\lambda} \frac{1}{n} \sum_{i=1}^{n} \rho\left(\lambda' g_i(\theta)\right) \right] \tag{1}$$

The first is the primal and the second is the dual problem, the latter being preferred in general to define GEL estimators. The vector λ is the Lagrange multiplier associated with the first constraint in the primal problem. Its estimator plays an important role in testing the validity of the moment conditions. $\rho(v)$ is a strictly concave function normalized so that $\rho'(0) = \rho''(0) = -1$. It can be shown that $\rho(v) = \ln(1 - v)$ corresponds to Empirical Likelihood (EL) of Owen (2001) , $\rho(v) = -\exp(v)$ to the Exponential Tilting (ET) of Kitamura and Stutzer (1997), and $\rho(v) = (-v - v^2/2)$ to the Continuous Updated GMM estimator (CUE) of Hansen et al. (1996). In the context of GEL, the CUE is also known at the Euclidean Empirical Likelihood (EEL), because it corresponds to $h_n(p_i)$ being the Euclidean distance.

Once the solution is obtained for θ and λ, the implied probabilities can be computed as follows

$$\hat{p}_i = \frac{\rho'(\hat{\lambda}' g_i(\hat{\theta}))}{\sum_{j=1}^{n} \rho'(\hat{\lambda}' g_j(\hat{\theta}))} \tag{2}$$

If we relax the iid assumption, the problem is identical, but the moment function must be smoothed using a kernel method. Smith (2001) proposes to replace $g_i(\theta)$ by:

$$g_i^w(\theta) = \sum_{s=-m}^{m} w(s) g_{i-s}(\theta)$$

where $w(s)$ are kernel based weights that sum to one (see also Kitamura and Stutzer (1997) and Smith (2001).

2 An S4 class object for GEL models

All classes for GEL models inherit directly from "gmmModels" classes. It is just a gmmModel class with two additional slots: "gelType" and "wSpec". The first is a list with the name of the GEL method and a function for $\rho(v)$, if not already available in the package. The second slot is used to store information about the smoothing of $g_i(\theta)$ in the case of non-iid observations. As for gmmModels, "gelModels" is a union class for "linearGel", "nonlinearGel", "formulaGel" and "functionGel". They inherit directly from the associated GMM model class. For example, "linearGel" is a class that contains a "linearGmm" class object and the two additional slots. The constructor is the gelModel() function. We use the same models presented in the gmmS4 vignette:

```
library(gmm4)

## Loading required package:  sandwich

data(simData)
```

- Linear models:

```
lin <- gelModel(y~x1+x2, ~x2+z1+z2, data=simData, vcov="iid", gelType="EL")
lin

## GEL Model: Type  EL
## *******************************
## Moment type: linear
## No Smoothing required
##
## Number of regressors: 3
## Number of moment conditions: 4
## Number of Endogenous Variables: 1
## Sample size:  50
```

- Formula-type models:

```
theta0=c(mu=1,sig=1)
x <- simData$x1
dat <- data.frame(x=x, x2=x^2, x3=x^3, x4=x^4)
gform <- list(x~mu,
              x2~mu^2+sig,
              x3~mu^3+3*mu*sig,
              x4~mu^4+6*mu^2*sig+3*sig^2)
form <- gelModel(gform, NULL, gelType="EEL", theta0=theta0, vcov="MDS", data=dat)
form

## GEL Model: Type  EEL
## *******************************
## Moment type: formula
## No Smoothing required
##
## Number of regressors: 2
## Number of moment conditions: 4
## Number of Endogenous Variables: 0
## Sample size:  50
```

- function:

```
fct <- function(theta, x)
    cbind(x-theta[1], (x-theta[1])^2-theta[2],
          (x-theta[1])^3, (x-theta[1])^4-3*theta[2]^2)
```

```
dfct <- function(theta, x)
    {
        m1 <- mean(x-theta[1])
        m2 <- mean((x-theta[1])^2)
        m3 <- mean((x-theta[1])^3)
        matrix(c(-1, -2*m1, -3*m2, -4*m3,
                 0, -1, 0, -6*theta[2]), 4, 2)
    }
theta0=c(mu=1,sig2=1)
func <- gelModel(fct, simData$x3, theta0=theta0, grad=dfct, vcov="iid", gelType="ET")
func
```

```
## GEL Model: Type  ET
## *******************************
## Moment type: function
## No Smoothing required
##
## Number of regressors: 2
## Number of moment conditions: 4
## Sample size:  50
```

- Non-linear models:

```
theta0=c(b0=1, b1=1, b2=1)
gform <- y~exp(b0+b1*x1+b2*x2)
nlin <- gelModel(gform, ~z1+z2+z3+x2, theta0=theta0, vcov="MDS", data=simData,
                 gelType="HD")
nlin
```

```
## GEL Model: Type  HD
## *******************************
## Moment type: nonlinear
## No Smoothing required
##
## Number of regressors: 3
## Number of moment conditions: 5
## Number of Endogenous Variables: 2
## Sample size:  50
```

It is also posssible to convert an existing gmmModels to gelModels:

```
nlin <- gmmModel(gform, ~z1+z2+z3+x2, theta0=theta0, vcov="MDS", data=simData)
nlin <- gmmToGel(nlin, gelType="HD")
```

2.1 The rhoXXX functions

The package provides $\rho(v)$ for ET, EL, EEL and HD. The function names are respectively "rhoET", "rhoEL", "rhoEEL" and "rhoHD". For any other GEL, users can pass user specific $\rho(v)$ function to the rhoFct argument of the gelModel(). The following example shows how the function must be built:

```
rhoEL
```

```
## function (gmat, lambda, derive = 0, k = 1)
## {
##     lambda <- c(lambda) * k
##     gmat <- as.matrix(gmat)
##     gml <- c(gmat %*% lambda)
```

```
##     switch(derive + 1, log(1 - gml), -1/(1 - gml), -1/(1 - gml)^2)
## }
## <bytecode: 0x563f1c1b71b0>
## <environment: namespace:gmm4>
```

Therefore, the function must be in the form $f(X, \lambda, d, k)$, where the first argument is an $n \times q$ matrix, the second argument is a vector of q elements, the third is an integer that indicates the order of derivative to return, and the last is a scalar that depends on the kernel used to smooth the moment function. We will discuss that in more details below. The function must return the vector $\rho(kX\lambda)$, $\rho'(kX\lambda)$ or $\rho''(kX\lambda)$, when derive is 0, 1, or 2, where $\rho'(v)$ and $\rho''(v)$ are the first and second derivative of $\rho(v)$.

2.2 The Lagrange multiplier solver

The function getLambda() function solve the maximization problem

$$\max_{\lambda} \frac{1}{n}\sum_{i=1}^{n} \rho(\lambda' X_i)$$

It is used to solve problem (1). The arguments are:

```
args(getLambda)

## function (gmat, lambda0 = NULL, gelType = NULL, rhoFct = NULL,
##     tol = 1e-07, maxiter = 100, k = 1, method = "BFGS", algo = c("nlminb",
##         "optim", "Wu"), control = list())
## NULL
```

The first argument is the $n \times q$ matrix X. The second argument is the starting value for λ. If set to NULL, a vector of zeroes is used. The third argument is the GEL type, which is either "ET", "EL", "EEL", "REEL" or "HD". If the rhoFct is provided, getType is ignored. The argument control is used to pass a list of arguments to optim(), nlminb() or constrOptim(). The argument "k" is the scalar described in the previous subsection. To describe all other arguments, we are presenting the options by type of GEL:

- EL: There are three possible options for EL. The default is "nlminb", in which case, the arguments tol, maxiter and method are ignored. If algo is set to "optim", the solution is obtained using constrOptim(), with the restriction $(1 - \lambda' X_i) > 0$ for all i. If algo is set to "Wu", the Wu (2005) algorithm is used. The argument tol and maxit are used to control the stopping rule.
- HD: Identical to EL, except that the "Wu" algorithm is not available for that type of GEL.
- EEL: There is an analytical solution, so all other arguments are ignored.
- REEL: This is EEL with a non-negativity constraint on all implied probabilities. In that case, the maxiter can be used to control the number of iterations.
- : Others: When rhoFct is provided or the type is ET, the solution is obtained by either "nlminb" or "optim". In that case, the algorithms are controlled through the control argument.

Here are a few example using the simulated dataset (convergence code of 0 means that the algorithm converged with no detected problem):

```
X <- simData[c("x3","x4","z5")]
(res <- getLambda(X, gelType="EL"))$lambda

## [1] -0.04178415 -0.06745195 -0.02075063

res$convergence$convergence

## [1] 0
```

```
(res <- getLambda(X, gelType="EL", algo="Wu"))$lambda

## [1] -0.04178415 -0.06745195 -0.02075063

res$convergence$convergence

## [1] 0
```

```
(res <- getLambda(X, gelType="ET",control=list(maxit=2000)))$lambda

## [1] -0.04171694 -0.06804520 -0.02120255

res$convergence$convergence

## [1] 0
```

```
(res <- getLambda(X, rhoFct=rhoEEL))$lambda

## [1] -0.04119158 -0.06769691 -0.02148683

res$convergence$convergence

## [1] 0
```

Although we rarely call the function directly, it is good to understand how it works, because it is an important part of the estimation procedure.

3 Methods for gelModels Classes

References

S. Anatolyev. Gmm, gel, serial correlation, and asymptotic bias. *Econometrica*, 73:983–1002, 2005.

Pierre Chaussé. Computing generalized method of moments and generalized empirical likelihood with r. *Journal of Statistical Software*, 34(11):1–35, 2010. URL http://www.jstatsoft.org/v34/i11/.

Pierre Chaussé. *gmm: Generalized Method of Moments and Generalized Empirical Likelihood*, 2019. R package version 1.6-3.

L. P. Hansen, J. Heaton, and A. Yaron. Finit-sample properties of some alternative gmm estimators. *Journal of Business and Economic Statistics*, 14:262–280, 1996.

Y. Kitamura and M. Stutzer. An information-theoretic alternative to generalized method of moments estimation. *Econometrica*, 65(5):861–874, 1997.

W. K. Newey and R. J. Smith. Higher order properties of gmm and generalized empirical likelihood estimators. *Econometrica*, 72:219–255, 2004.

A. B. Owen. *Empirical Likelihood*. Chapman and Hall, 2001.

R. J. Smith. Gel criteria for moment condition models. *Working Paper, University of Bristol*, 2001.

C. Wu. Algorithms and R codes for the pseudo empirical likelihood method in survey sampling. *Survey Methodology*, 31(2):239, 2005.

Appendix

A Some extra codes