Implementing a Class of Structural Change Tests: An Econometric Computing Approach

Achim Zeileis

http://www.ci.tuwien.ac.at/~zeileis/
Contents

❖ Why should we want to do
  ❖ tests for structural change,
  ❖ econometric computing?

❖ Generalized M-fluctuation tests
  ❖ Empirical fluctuation processes: gefp
  ❖ Functionals for testing: efpFunctional

❖ Applications
  ❖ Austrian National Guest Survey
Structural change tests

Structural change has been receiving a lot of attention in econometrics and statistics, particularly in time series econometrics.

**Aim:** to learn if, when and how the structure underlying a set of observations changes.

In a parametric model with parameter $\theta_i$ for $n$ totally ordered observations $Y_i$ test the null hypothesis of parameter constancy

$$H_0 : \quad \theta_i = \theta_0 \quad (i = 1, \ldots, n).$$

against changes over “time”. 
Econometric computing

Econometrics & computing:

- Computational econometrics: methods requiring substantial computations (bootstrap or Monte Carlo methods),

- Econometric computing: translating econometric ideas into software.

To transport methodology to the users and apply new methods to data software is needed.
Desirable features of an implementation:

- easy to use,
- numerically reliable,
- computationally efficient,
- flexible and extensible,
- reusable components,
- open source,
- object oriented,
- reflect features of the conceptual method.

Undesirable: single monolithic functions.

Also important: software delivery.
Econometric computing

All methods implemented in the R system for statistical computing and graphics

http://www.R-project.org/

in the contributed package strucchange.

Both are available under the GPL (General Public Licence) from the Comprehensive R Archive Network (CRAN):

http://CRAN.R-project.org/
Data from the Austrian National Guest Survey about the summer seasons 1994 and 1997.

**Here:** use logistic regression model

- response: cycling as a vacation activity (done/not done),
- available regressors: age (in years), household income (in AT$S$/month), gender and year (as a factors/dummies),
- fit model for the subset of male tourists (6256 observations),
- (log-)income is not significant.

```r
R> gsa.fm <- glm(cycle ~ poly(Age, 2) + Year, data = gsa,
                  family = binomial)
```

**But:** Maybe there are instabilities in the model for increasing income?
M-fluctuation tests

- fit model
- compute empirical fluctuation process reflecting fluctuation in
  - residuals
  - coefficient estimates
  - M-scores (including OLS or ML scores etc.)
- theoretical limiting process is known
- choose boundaries which are crossed by the limiting process (or some functional of it) only with a known probability $\alpha$.
- if the empirical fluctuation process crosses the theoretical boundaries the fluctuation is improbably large $\Rightarrow$ reject the null hypothesis.
**Model fitting:** parameters can often be estimated based on a score function or estimating equation $\psi$ with

$$\mathbb{E}[\psi(Y_i, \theta_i)] = 0.$$ 

Under parameter stability estimate $\theta_0$ by:

$$\sum_{i=1}^{n} \psi(Y_i, \hat{\theta}) = 0.$$ 

Includes: OLS, ML, Quasi-ML, robust M-estimation, IV, GMM, GEE.

Available in R: linear models `lm`, GLMs, logit, probit models `glm`, robust regression `rlm`, etc.
**Empirical fluctuation processes**

**Test idea:** if $\theta$ is not constant the scores $\psi$ should fluctuate and systematically deviate from 0.

Capture fluctuations by partial sums:

$$ efp(t) = \hat{J}^{-1/2} n^{-1/2} \sum_{i=1}^{\lfloor nt \rfloor} \psi(Y_i, \hat{\theta}). $$

and scale by covariance matrix estimate $\hat{J}$. 
Test idea: if $\theta$ is not constant the scores $\psi$ should fluctuate and systematically deviate from 0.

Capture fluctuations by partial sums:

$$efp(t) = \hat{J}^{-1/2} \frac{1}{n} \sum_{i=1}^{\lfloor nt \rfloor} \psi(Y_i, \hat{\theta}).$$

and scale by covariance matrix estimate $\hat{J}$.

Functional central limit theorem: empirical fluctuation process converges to a Brownian bridge

$$efp(\cdot) \xrightarrow{d} W^0(\cdot)$$
Empirical fluctuation processes

Implementation idea:

- don’t reinvent the wheel: use existing model fitting functions and just extract the scores or estimating functions,
- also allow plug-in of HC and HAC covariance matrix estimators,
- provide infrastructure for computing processes.
Empirical fluctuation processes

Implementation idea:

* don’t reinvent the wheel: use existing model fitting functions and just extract the scores or estimating functions,
* also allow plug-in of HC and HAC covariance matrix estimators,
* provide infrastructure for computing processes.

gefp(..., fit = glm, scores = estfun,
       vcov = NULL, order.by = NULL)
Empirical fluctuation processes

For Austrian guest survey data:

```r
R> gsa.efp <- gefp(cycle ~ poly(Age, 2) + Year, family = binomial, 
                   data = gsa, order.by = ~ log(HHIncome), parm = 1:3)
```
Empirical fluctuation processes

\[\text{(Intercept)} -0.5 \quad 0.5 \quad 1.5\]
\[\text{poly(Age, 2)}1 -1.0 \quad 0.0 \quad 0.5\]
\[6 \quad 8 \quad 10 \quad 12\]
\[0.0 \quad 1.0 \quad 2.0\]
\[\text{poly(Age, 2)}2\]
\[\log(\text{Household Income})\]
The empirical fluctuation process can be aggregated to a scalar test statistic by a functional $\lambda(\cdot)$

$$\lambda \left( efp_j \left( \frac{i}{n} \right) \right),$$

where $j = 1, \ldots, k$ and $i = 1, \ldots, n$.

$\lambda$ can usually be split into two components: $\lambda_{\text{time}}$ and $\lambda_{\text{comp}}$.

Typical choices for $\lambda_{\text{time}}$: $L_\infty$ (absolute maximum), mean, range.

Typical choice for $\lambda_{\text{comp}}$: $L_\infty$, $L_2$.

$\Rightarrow$ can identify component and/or timing of shift.
Functionals

Double maximum statistic:

$$\max_{i=1,\ldots,n} \max_{j=1,\ldots,k} \left| \frac{efp_j(i/n)}{b(i/n)} \right|,$$

typically with $b(t) = 1$.

Cramér-von Mises statistic:

$$n^{-1} \sum_{i=1}^{n} \left\| efp_j(i/n) \right\|_2^2,$$

Critical values can easily be obtained by simulation of $\lambda(W^0)$. In certain special cases, closed form solutions are known.
**Functionals**

**Implementation idea:**

- specify functional (and boundary function)
- simulate critical values (or use closed form solution)
- combine all information about a functional in a single object: process visualization, computation of test statistic, computation of $p$ values,
- provide infrastructure which can be used by the methods of the generic functions `plot` for visualization and `sctest` for significance testing.

For the double maximum and the Cramér-von Mises functionals such objects are available in `strucchange`: `maxBB`, `meanL2BB`. 
R> plot(gsa.efp, functional = maxBB)
R> plot(gsa.efp, functional = maxBB, aggregate = FALSE)
R> plot(gsa.efp, functional = meanL2BB)
R> sctest(gsa.efp, functional = maxBB)

  M-fluctuation test

  data:  gsa.efp
  f(efp) = 2.0594, p-value = 0.001242

R> sctest(gsa.efp, functional = meanL2BB)

  M-fluctuation test

  data:  gsa.efp
  f(efp) = 2.2119, p-value = 0.005
New functionals can be easily generated with

```r
efpFunctional(
    functional = list(comp = function(x) max(abs(x)), time = max),
    boundary = function(x) rep(1, length(x)),
    computePval = NULL, computeCritval = NULL,
    nobs = 10000, nrep = 50000, nproc = 1:20)
```

An object created by `efpFunctional` has slots with functions

- plotProcess
- computeStatistic
- computePval

that are defined based on lexical scoping.
Use functional similar to double max functional, but with boundary function

\[ b(t) = \sqrt{t \cdot (1 - t)} + 0.05, \]

which is proportional to the standard deviation of the process plus an offset.

```r
myFun1 <- efpFunctional(
  functional = list(comp = function(x) max(abs(x)), time = max),
  boundary = function(x) sqrt(x * (1-x)) + 0.05,
  nobs = 10000, nrep = 50000, nproc = NULL)
```
R> plot(gsa.efp, functional = myFun1)
Functionals

Use standard double max functional but aggregate over “time” first. Leads to the same test statistic and $p$ value, but the aggregated process looks different.

myFun2 <- efpFunctional(
  functional = list(time = function(x) max(abs(x)), comp = max),
  computePval = maxBB$computePval)
R> plot(gsa.efp, functional = myFun2)
R> sctest(gsa.efp, functional = myFun1)

M-fluctuation test

data:  gsa.efp
f(efp) = 4.7947, p-value = < 2.2e-16

R> sctest(gsa.efp, functional = myFun2)

M-fluctuation test

data:  gsa.efp
f(efp) = 2.0594, p-value = 0.001242
Conclusions

The general class of M-fluctuation tests is implemented in strucchange:

- `gefp` — computation of empirical fluctuation processes from (possibly user-defined) estimation functions,

- `efpFunctional` — aggregation of empirical fluctuation processes to test statistics, automatic tabulation of critical values,

- `plot` and `sctest` — methods for visualization and significance testing based on empirical fluctuation processes and corresponding functionals.
See more at ...

useR!
2004

The 1st R user conference
Vienna, May 20–22, 2004

http://www.ci.tuwien.ac.at/Conferences/useR-2004/