A Tutorial on the GFL Package

1 Introduction

The Matlab package GFL implements the group-fused-Lasso (GFL) procedure proposed in Qian and Su (2016) for the estimation of linear regression with an unknown number of breaks. Here we consider a straightforward extension of Qian and Su (2016), allowing some coefficients to be constant. Specifically, we consider the following model,

$$y = x_t'\beta_t + z_t'\gamma + u_t,$$

where x_t is a p-by-1 vector of variables that may have time-varying effects on y, z_t is a q-by-1 vector of variables that have constant effects on y. Suppose we observe $(y_t, x_t, z_t, t = 1, \ldots, n)$ and want to estimate the model without knowing the number or the dates of structural breaks in β_t. Let m be the number of common breaks in β_t. We assume that although m is unknown, it is much smaller than the sample size n.

2 Usage

The main Matlab procedure is gfl:

[regime,alpha,Sigma,R2,ssr,resid] = gfl(y,x,z,option);

2.1 Inputs:

- y: explained variable (n-by-1)
- x: explanatory variables with time-varying effect (n-by-p)
- z: (optional) explanatory variables without time-varying effect (n-by-q)

- option: (optional) a construct for settings.
  - option.date: (optional) a n-by-1 cell array of dates.
  - option.lambda: (optional) the tuning parameter on the GFL penalty, if option.lambda='ic' (default), then lambda will be chosen by information criterion. If option.lambda='cv', then lambda will be chosen by cross-validtion.
  - option.L: (optional) the set of lambda's to be considered in the automatic selection of option.lambda. The default setting is option.L=[], in which case the procedure will generate one.
  - option.XTol: (optional) The error tolerance level, a small positive number. The default choice is 1e-6;
  - option.maxIter: (optional) The maximum number of iterations in the the block-coordinate-descent (BCD) algorithm. The default choice is 1000.
  - option.minseg: (optional) The minimum length of segments. The default choice is $p + q + 1$;
  - option.mex: (optional) An integer indicating whether to use mex implementation. The default choice is 1, using mex implementation. If the code does not work (e.g., causing Matlab to crash), then set option.mex=0, in which case the program will call a slower but platform-independent version of the BCD code.

2.2 Outputs:

- regime: a vector of break dates in the form of $[1\ T_1\ T_2\ \cdots\ T_m\ n+1]$, where $(T_1\ T_2\ \cdots T_m)$ are m break dates. (We may understand $T_0 = 1$ and $T_{m+1} = n + 1$.) For example, if $n = 120$ and we obtain regime = $\{1,\ 41,\ 81,\ 121\}$, then it implies that there are two breaks at 41 and 81, respectively. In other words, there are three regimes: 1:40, 41:80, and 81:120.

- alpha: The estimated coefficients ($[(m+1)*p+q]$-by-1). The first $(m+1)*p$ elements correspond to β_t and the last q elements correspond to γ
- Sigma: The estimated covariance matrix ($[(m+1)*p+q]$-by-$[(m+1)*p+q]$).
- R2: R^2 for each regime and for the whole sample.
- ssr: The sum of squared residuals (a scalar).

The Matlab file test_gfl.m illustrates how this procedure may be applied to simulated data. An empirical example is offered below.

3 An Example

To analyze structural changes in the US monetary policy, we consider the following policy rule:

$$r_t = c_t + \beta_t \cdot gap_t + \gamma_t \cdot \pi_t + u_t,$$

where r_t is the US federal funds rate, gap_t is the GDP gap (the gap between GDP and its potential, in percentage terms), π_t is the inflation rate, (c_t, β_t, γ_t) are coefficients that may have a few "common breaks". We obtain quarterly data on (r_t, gap_t, π_t) and estimate the model.

Note that we are dealing with common breaks in this exercise. As a result, we only need to supply to the gfl procedure x and y, and set $z = [\,]$, an empty set. The first column of x is a vector of ones, corresponding to the intercept. The second column of x stores gap_t in the order of time, and the third column of x stores π_t. The vector of y stores r_t. We may use the following Matlab command,

```
>> option.lambda='ic';
>> [regime,alpha,Sigma,R2] = gfl(y,x,[ ],option);
```

The first command is actually not necessary, since the default value for option.lambda is exactly 'ic'. In this case, gfl automatically choose a tuning parameter by minimizing by the information criterion proposed in Qian and Su (2016). If let option.lambda='ic', then cross-validation would be used to select lambda. The user can also directly supply a lambda, for example,

```
>> option.lambda = 100;
>> [regime,alpha,Sigma,R2] = gfl(y,x,[ ],option);
```

If the user does not specify outputs and simply run:

```
>> option.lambda='ic';
>> gfl(y,x,[ ],option);
```

Then the program would report formatted results on the screen. In this empirical exercise, we obtain:

```
Group-Fused-Lasso estimation of time series regression with breaks

Lambda is chosen by information criterion.

Estimated break dates and time-varying parameters in each regime:

Regimes && Parameter Estimates && Standard Errors && P-values

1966Q1 to 1980Q3: && 0.7006 & 1.2396 & 0.5543 && 0.4792 & 0.0885 & 0.0591 && 0.1494 & 0.0000 & 0.0000 &

1980Q4 to 1989Q1: && 2.8561 & 1.5459 & 0.1281 && 0.5186 & 0.1166 & 0.0831 && 0.0000 & 0.0000 & 0.1332 &

1989Q2 to 2000Q4: && 1.3859 & 1.2219 & 0.6054 && 0.5415 & 0.1803 & 0.0903 && 0.0140 & 0.0000 & 0.0000 &

2001Q1 to 2015Q2: && -0.3144 & 1.2583 & 0.2704 && 0.5850 & 0.2846 & 0.0504 && 0.5932 & 0.0000 & 0.0000 &
```

The output should be self-evident.

References

- Qian, J., L. Su, 2016, Shrinkage estimation of regression models with multiple structural Changes. Econometric Theory, 32 (6), 1376-1433.