Smooth varying coefficient models in Stata

Yet another semiparametric approach

Rios-Avila, Fernando[1]

[1]friosavi@levy.org
Levy Economics Institute

Stata Conference, July 2020
At home edition

Table of Contents

Table of Contents

Introduction

- Nonparametric regressions are powerful tools to capture relationships between dependent and independent variables with minimal functional forms assumptions. (very flexible)
- The added flexibility comes at a cost:
  - Curse of dimensionality. Larger sample sizes are needed to achieve same power as parametric models.
  - Computational burden. Procedures for model selection and estimation demand a lot of time.
- Perhaps because of this, Stata had a limited set of native commands for the estimation of nonparametric models.
- This changed with `npregress series/kernel`. (still they kind be slow and too flexible)

Introduction

- A response to the main weakness of NP methods has been the development of semiparametric (SP) methods.
- SP combine the flexibility of NP regressions with the structure of standard parametric models.
- The added structure reduces the curse of dimensionality and the computational cost of model selection and estimation.
- Many community-contributed commands have been proposed for the analysis of a large class of semiparametric models in Stata.
  See: Verardi(2013) ▸ Semipar-Stata

Introduction

- In this presentation, I'll describe the estimation of a particular type of SP model known as Smooth varying coefficient models (SVCM).

Introduction

- In this presentation, I'll describe the estimation of a particular type of SP model known as Smooth varying coefficient models (SVCM).
- I'll show how they could be estimated "manually"

Introduction

- In this presentation, I'll describe the estimation of a particular type of SP model known as Smooth varying coefficient models (SVCM).
- I'll show how they could be estimated "manually"
- and introduce the package `vc_pack`, that can be used for the model selection, estimation, and visualization of this type of model.

Table of Contents

What do they do?

- Consider a model with 3 set of variables such that:

$$y = f(X, Z, e)$$

- Where X and Z are observed and W=[X;Z], $E(e|x, z) = 0$

What do they do?:Parametric Regression

- a Standard OLS (parametric model under linearity assumption), will estimate their relationship with respect to Y such that :

$$E(y|x,z) = x * b_x + z * b_z$$

- where its well known that:

$$b_w = (W'W)^{-1}(W'Y)$$

$$W = [X;Z]\&b'_w = [b'_x;b'_w]$$

What do they do?:NonParametric Regression

- NP regression assumes the conditional expected value of the Y is a smooth function.

$$E(y|x, z) = g(x, z)$$

- In this model, often, there are not parameters to be estimated, but conditional means

$$g(x, z) = \frac{\sum y_i * K(w_i, w, h)}{\sum K(w_i, w, h)}$$

- where $K()$ is a product of Kernel functions. (thus this is a kernel-based NP regression)
- So the NP regression is simply the estimation of weighted means.
- One can also use Splines, series, or penalized splines.

What do they do?:SVCM Regression

- SVCM regression assumes the model is linear conditional on z:

$$E(y|x,z) = xb_x(z)$$

- This model combines the linear structure of OLS, assuming the coefficients are nonlinear with respect to Z.
- If we have enough observations for Z=z, the estimator is simply:

$$b_x(z) = E(X'X|Z=z)^{-1}E(X'y|Z=z)$$

$$b_x(z) = (X'\mathcal{K}(z)X)^{-1}(X'\mathcal{K}(z)y)$$

- where $\mathcal{K}(z)$ is a matrix with the diagonal equal to the K(Z,z,h).

What do they do?:SVCM Regression

- However, local constant tends to be bias at the boundaries of Z. So as an alternative, Local Linear (LL) estimator can be used:

$$b_x(Z_i) \approx b_x(z) + \frac{\partial b_x(z)}{\partial z}(Z_i - z)$$

- But we are still interested in $b_x(z)$.
- The estimator above remains the same, but X is substituted by $\mathcal{X} = (X; (Z_i - z)X)$

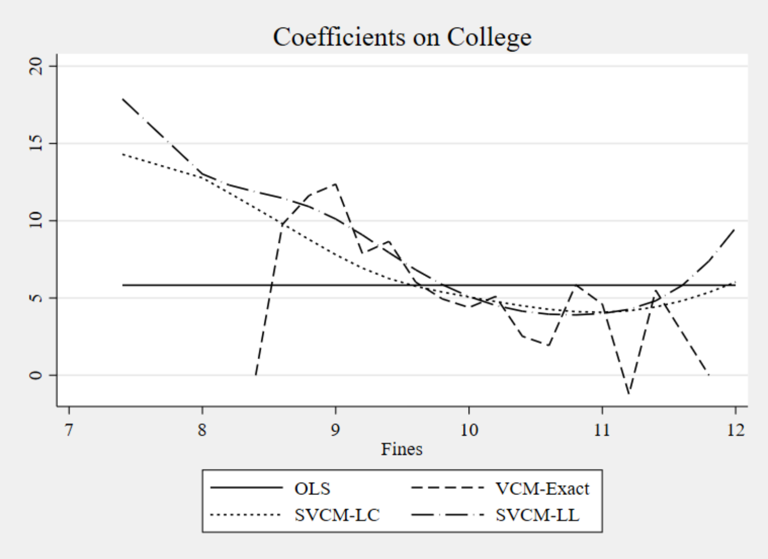
Table of Contents

SVCM-Kernel Local Linear Estimation

- The estimation of SVCM is relatively straight forward, specially if Z is a single variable.
  - Choose point(s) of reference Z (probably many points)
  - Choose appropriate bandwidth h
  - Choose between local constant or local linear (or local polynomial)
  - Estimate coefficients, and done
  - Or, use splines instead of kernel (see f_able)

```
* Local constant
. webuse dui, clear
. regress citations college taxes i.csize ///
  if fines==9 (as if h=0)
. regress citations college taxes i.csize ///
  [iw=normalden(fines,9,.5)]
* Local Linear
. gen dz=fines-9
. regress citations c.dz##c.(college taxes i.csize) ///
  [iw=normalden(fines,9,.5)]
```

Example


Coefficients on College
0
5
10
15
20
7
8
9
10
11
12
Fines
OLS
VCM-Exact
SVCM-LC
SVCM-LL

Example: Remarks

- While the estimation is "easy", important aspects need to be address:
- Model selection and choice of bandwidth
- Systematic model estimation and standard errors.
- Post estimation and evaluation of the model.
- and plots of conditional effects.

Table of Contents

SVCM in Stata: vc_pack

- To address these points, I propose and present a set of commands that aim to facilitate the estimation of SVMC.
- In specific, the commands can be used for the estimation of SVCM using a local linear estimator and assuming a single conditioning variable z.

Model selection: vc_bw and vc_bwalt

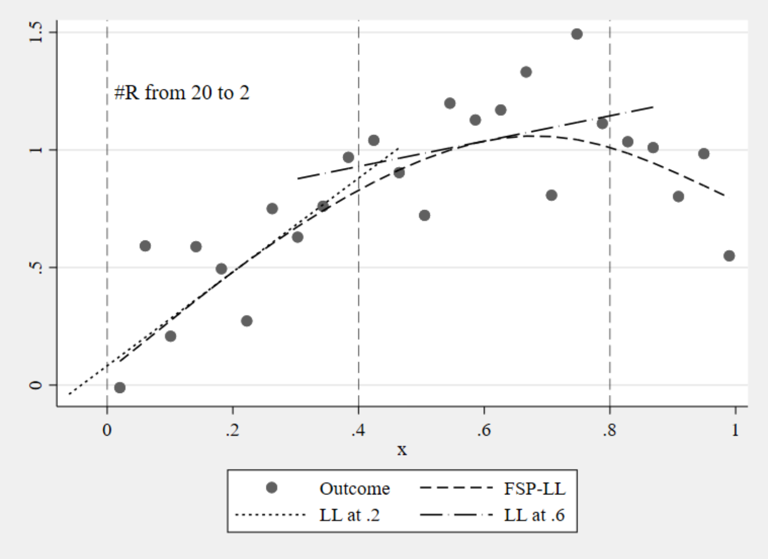
- The first (most important) step is the selection of the bandwidth h. This reflects the trade off between variance and Bias in the model estimation.
- vc_bw and vc_bwalt provide two options (different algorithms) that can be used to select an optimal bandwidth using a leave-one-out Cross validation procedure:

$$h^* = min_h \sum_{i=1}^{N} \omega(z)(y_i - \hat{y}_{-i})^2$$

- For a faster estimation of the CV criteria and h^*, both commands use binned Local Linear regressions.

```
vc_bw[alt] y x1 x2 x3, vcoeff(z) ///
[kernel(kfun) trimsample(varname) otheroptions]
```

Binned Regression



Estimation and Inference: vc_reg; vc_bsreg & vc_preg

- The next step is the model estimation. While the estimation itself is simple, the estimation of standard errors require special care.
- Three options are provided. vc_[p|bs]reg
- These commands estimate LL-SVCM for a selected "ref. points".
- vc_[p]reg Estimate VcoV matrix a Sandwich formula:

$$\Sigma(B(z)) = q_c(\mathcal{X}'\mathcal{K}(z)\mathcal{X})^{-1}(\mathcal{X}'\mathcal{K}(z)D(e_i)\mathcal{K}(z)\mathcal{X})(\mathcal{X}'\mathcal{K}(z)\mathcal{X})^{-1}$$

  The difference between them is how e_i is estimated.
  Either using F-LL or Binn-LL
- vc_bsreg instead uses a Bootstrap procedure to estimate Σ.

```
vc_[p|bs]reg y x1 x2 x3, [vcoeff(z) bw(#) kernel(kfun)] ///
[klist(numlist) or k(#) ] ///
[robust cluster(varname) hc2 hc3 or reps(#)]
```

Post estimation: vc_predict & vc_test

- The third step would be summarize and evaluate the estimated model.
- This can be done with vc_predict & vc_test
- The first command has the following syntax:

```
vc_predict y x1 x2 x3, [ vcoeff(svar) bw(#) kernel(kfun)] ///
[yhat(newvar) res(newvar) looe(newvar) lvrg(newvar)] [stest]
```

- This command provides some information regarding model fitness.
- And can be used to obtain model predictions, residuals, Leave-one-out residuals, or the leverage statistics
- option stest, estimates the approximate F-Statistic for testing against parametric models.

Post estimation: vc_predict

- Log Mean Squared LOO-errors:

$$LogMSLOOE = log\left[\frac{1}{N}\sum(y_i - \hat{y}_{-i})^2\right]$$

- Goodness of Fit (R^2): (Henderson and Parmeter 2014)

$$R_1^2 = 1 - \frac{SSR}{SST} \text{ or } R_2^2 = \frac{Cov(y_i, \hat{y}_i)^2}{\sqrt{Var(y_i)Var(\hat{y}_i)}}$$

Post estimation: vc_predict

- Degrees of Freedom: Hastie and Tibshirani (1990)

$$Model : df1 = Tr(S)$$

$$Resid : N - df2 = N - (1.25 * Tr(S) - .5)$$

Where S is a $N \times N$ matrix. The SVCM projection matrix

- Expected Kernel Observations:

$$Kobs(z) = \sum_{i=1}^{N} k_w \left(\frac{Z_i - z}{h} \right) = \sum_{i=1}^{N} k \left(\frac{Z_i - z}{h} \right) * k^{-1}(0)$$

$$E(Kobs(z_i)) = \frac{1}{N} \sum_{i=1}^{N} Kobs(z_i)$$

Post estimation: vc_predict

- Specification test (Approximate F-test)

$$aF = \frac{\sum \hat{e}_{ols}^2 - \sum \hat{e}_{svcm}^2}{\sum \hat{e}_{svcm}^2} * \frac{n - df2}{df2 - df_{ols}} \sim F_{n-df2, df2-df_{ols}}$$

- where the alternative parametric models are:

$$M0 : y = Xb_x + Zb_z + e_{ols}$$

$$M1 : y = Xb_x + (X * Z)b_{xz1} + Zb_z + e_{ols}$$

$$M2 : y = Xb_x + (X * Z, X * Z^2)b_{xz2} + Zb_z + e_{ols}$$

$$M3 : y = Xb_x + (X * Z, X * Z^2, X * Z^3)b_{xz3} + Zb_z + e_{ols}$$

Post estimation: vc_test

- I also include a command to implement Cai, Fan, and Yao (2000) specification test.

$$\hat{J} = \frac{\sum \hat{e}_{ols}^2 - \sum \hat{e}_{svcm}^2}{\sum \hat{e}_{svcm}^2}$$

Where the Critical values are estimated via Wild Bootstrap Procedure.

```
vc_test y x1 x2 x3, [vcoeff(svar) bw(#) kernel(kernel)] ///
[knots(#) km(#) degree(#d) wbsrep(#wb)]
```

Visualization: vc_graph

- After model has been estimated, we can produce plots of the Smooth varying coefficients (or the changes across Z)
- vc_graph can be used for this, using all the points of reference estimated via vc_[p|bs]reg

```
vc_graph [varlist] , [ ci(#) constant delta ] ///
[xvar(xvarname) graph(stub) ///
[rarea ci_off pci addgraph(str) ]
```

- varlist should follow the same syntax as in the original model.
- Using delta plots the coefficients for the interactions $x * (Z - z)$, and constant plots the local constant.
- All figures will be stored in memory using sequentially numbers

Table of Contents

Example: Bw selection

```
. ** Stata Conf Example
. qui:webuse dui, clear
. vc_bwalt citations i.college i.taxes i.csize, vcoeff(fines) plot
Kernel: gaussian
Iteration: 0 BW:   0.5539761 CV:   3.129985 Path: \_
Iteration: 1 BW:   0.6093737 CV:  3.1242958 Path: \_/
....
Iteration: 14 BW:   0.7397731 CV:  3.1194971 Path: \_/
Iteration: 15 BW:   0.7397731 CV:  3.1194971
Bandwidth stored in global $opbw_
Kernel function stored in global $kernel_
VC variable name stored in global $vcoeff_
. vc_bw citations i.college i.taxes i.csize, vcoeff(fines) plot
Kernel: gaussian
Iteration: 0 BW:   0.5539761 CV:   3.129985
Iteration: 1 BW:   0.6870521 CV:   3.120199
Iteration: 2 BW:   0.7343729 CV:   3.119504
Iteration: 3 BW:   0.7397456 CV:   3.119497
Iteration: 4 BW:   0.7397999 CV:   3.119497
Bandwidth stored in global $opbw_
Kernel function stored in global $kernel_
VC variable name stored in global $vcoeff_
```

Example:Post-Estimation

```
. vc_predict citations i.college i.taxes i.csize, stest
Smooth Varying coefficients model
Dep variable       : citations
Indep variables    : i.college i.taxes i.csize
Smoothing variable : fines
Kernel             : gaussian
Bandwidth          :    0.73980
Log MSLOOER        :    3.11950
Dof residual       :    477.146
Dof model          :     18.684
SSR                :  10323.152
SSE                :  37886.159
SST                :  47950.838
R2-1 1-SSR/SST     :    0.78471
R2-2               :    0.79010
E(Kernel obs)      :    277.835
```

Example:Post-Estimation

```
Specification Test approximate F-statistic
H0: Parametric Model
H1: SVCM y=x*b(z)+e
Alternative parametric models:
Model 0 y=x*b0+g*z+e
F-Stat: 8.24705 with pval 0.00000
Model 1 y=x*b0+g*z+(z*x)b1+e
F-Stat: 5.80964 with pval 0.00000
Model 2 y=x*b0+g*z+(z*x)*b1+(z^2*x)*b2+e
F-Stat: 0.75977 with pval 0.65174
Model 3 y=x*b0+g*z+(z*x)*b1+(z^2*x)*b2+(z^3*x)*b3+e
F-Stat: -2.07399 with pval 1.00000
```

Example:Post-Estimation

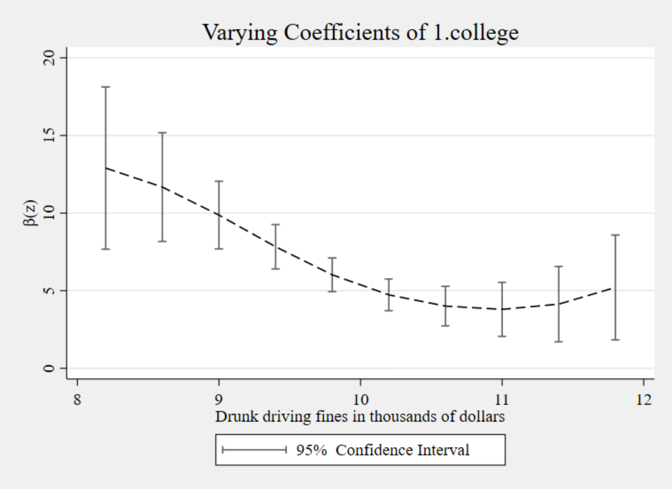
```
. set seed 1
. vc_test citations i.college i.taxes i.csize, wbsrep(100) degree(1)
Estimating J statistic CI using 100 Reps
Specification test.
H0: y=x*b0+g*z+(z*x)*b1+e
H1: y=x*b(z)+e
J-Statistic      :0.16869
Critical Values
90th   Percentile:0.09473
95th   Percentile:0.10543
97.5th Percentile:0.10861

. vc_test citations i.college i.taxes i.csize, wbsrep(100) degree(2)
Estimating J statistic CI using 100 Reps
Specification test.
H0: y=x*b0+g*z+(z*x)*b1+(z^2*x)*b2+e
H1: y=x*b(z)+e
J-Statistic      :0.01410
Critical Values
90th   Percentile:0.01189
95th   Percentile:0.01545
97.5th Percentile:0.01725
```

Example:Estimation

```
. qui:vc_preg citations i.college i.taxes i.csize, klist(9)
. ereturn display,  cformat(%5.4f) vsquish
------------------------------------------------------------------------------
       citations |     Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-----------------+------------------------------------------------------------
         college |
        college  |   9.8706     1.0206     9.67   0.000      7.5618     12.1794
           taxes |
            tax  |  -6.3768     1.0592    -6.02   0.000     -8.7728     -3.9808
           csize |
         medium  |   6.7344     0.9364     7.19   0.000      4.6162      8.8526
          large  |  14.9946     1.0710    14.00   0.000     12.5719     17.4174
         _delta_ |  -8.2560     1.2105    -6.82   0.000    -10.9944     -5.5175
college#c._delta_ |
        college  |  -4.5777     1.1637    -3.93   0.003     -7.2101     -1.9454
  taxes#c._delta_ |
            tax  |   3.0082     1.2104     2.49   0.035      0.2701      5.7463
  csize#c._delta_ |
         medium  |  -1.2990     1.0685    -1.22   0.255     -3.7163      1.1182
          large  |  -4.8632     1.2333    -3.94   0.003     -7.6531     -2.0734
           _cons |  23.9563     1.0986    21.81   0.000     21.4711     26.4415
------------------------------------------------------------------------------
```

Example:Visualization

```
. qui:vc_preg citations i.college i.taxes i.csize, k(10)
. vc_graph 1.college
```



Example:Visualization

```
. qui:vc_preg citations i.college i.taxes i.csize, k(10)
. vc_graph 1.taxes
```

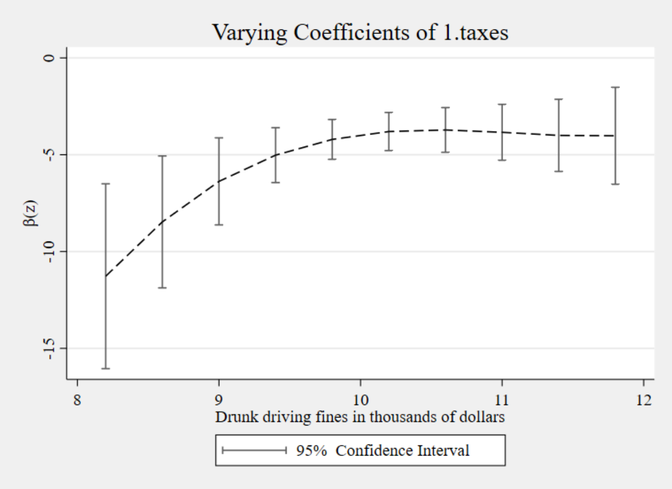


Table of Contents

Conclusions

- SVCMs are an alternative to full nonparametric models for the analysis of data.
- Models are assumed to be linear conditional on a smoothing variable(s) Z.
- In this presentation, I reviewed the implementation of this model using the commands in `vc_pack`
- Thank you!

If interested, current version of programs and paper can be accessed from bit.ly/rios_vcpack

References

Cai, Z., J. Fan, and Q. Yao. 2000. Functional-coefficient regression models for nonlinear time series. Journal of the American Statistical Association 95: 941-956.

Hastie, T. J., and R. J. Tibshirani. 1990. Generalized Additive Models. London: Chapman & Hall-CRC.

—–. 1993. Varying-coefficient models (with discussion). Journal of the Royal Statistical Society, Series B 55: 757-796.

Henderson, D. J., and C. F. Parmeter. 2015. Applied Nonparametric Econometrics. Cambridge: Cambridge University Press.

Rios-Avila, F. (forthcoming) Smooth varying-coefficient models in Stata. Forthcoming in The Stata Journal.