Smooth Varying Coefficient Models in Stata

By

Fernando Rios-Avila

Levy Economics Institute, Bard College

Abstract

Non-parametric regressions are powerful statistical tools that can be used to model relationships between dependent and independent variables with minimal assumptions on the underlying functional forms. Despite its potential benefits, these types of models have two weaknesses: The added flexibility creates a curse of dimensionality and procedures available for model selection, like cross-validation, have a high computational cost in samples with even moderate sizes. An alternative to fully-nonparametric models are semiparametric models that combine the flexibility of non-parametric regressions with the structure of standard models. This paper describes the estimation of a particular type of semiparametric modes known as smooth varying coefficient models (Hastie and Tibshirani 1993), based on kernel regression methods, using a new set of commands within vc\_pack. These commands aim to facilitate bandwidth selection, model estimation and create visualizations of the results.

Keywords: smooth varying coefficient models, kernel regression, cross-validation. Semiparametric estimations.

JEL:C14, C21, C52

1. **Introduction**

Non-parametric regressions are powerful statistical tools that can be used to model relationships between dependent and independent variables with minimal assumptions on the underlying functional forms. This flexibility makes non-parametric regressions robust to functional form misspecification, which is one of the main advantages over standard regression analysis.

The added flexibility of non-parametric regressions comes at a cost. On the one hand, the added flexibility creates what is known as the curse of dimensionality. In essence, because non-parametric regressions imply the estimation of a large number of parameters, accounting for interactions and non-linearities, more data is required to obtain results with a similar level of precision as their parametric counterparts. On the other hand, while larger datasets can be used to reduce the curse of dimensionality, procedures used for model selection and estimations are often too computationally intensive, making the estimation of this type of model less practical in samples of moderate to large sizes. Perhaps due to these limitations, and until recent versions, Stata had a very limited set of native commands for the estimation of non-parametric models. Even with the recent development of computing power, the estimation of full non-parametric models, using the currently available commands, remains a challenge when using large samples.[[1]](#footnote-1)

One response to the main weakness of non-parametric methods has been the development of semiparametric methods. These methods combine the flexibility of non-parametric regressions with the structure of standard parametric models, reducing the curse of dimensionality and reducing the computational cost of model selection and estimation.[[2]](#footnote-2) In fact, many community-contributed commands have been proposed for the analysis of a large class of semiparametric models in Stata.[[3]](#footnote-3)

A particular type of semiparametric models, the estimation for which has not been explored within the Stata environment, is known as Smooth Varying Coefficient Models (SVCM) (Hastie and Tibshirani 1993). These models assume that the outcome is a function of two sets of characteristics, and, where the effect of on follows some unspecified smooth function of Z. As described by Henderson and Parmeter (2015) this method is particularly popular in applied settings because they are easy to estimate and interpret, because the is a linear function of conditional on .

For example, as described in Hainmueller, Mummolo, and Xu (2018), SVCM can be thought of as multiplicative interactive models where the variable behaves as a moderator for the treatment variables of interest, relaxing the linear assumption of the interaction. Alternatively, as described in Rios-Avila (2019), SVCM can be used to extend standard Oaxaca-Blinder decomposition analysis to scenarios with continuous group variables, decomposing, for example, wage differences of individuals with different body mass index. Furthermore, under assumptions of an exogenous treatment and unconfoundedness, SVCM can be used to estimate heterogeneous dose treatment effects (see for example Hirano and Imbens (2004) for a discussion on continuous treatment effects).

This paper introduces a new set of commands that aim to facilitate the model selection, estimation, and visualization of SVCM with a single smoothing variable . vc\_bw and vc\_bwalt are commands used for model selection which implement a leave-one-out cross-validation procedure to select the optimal bandwidth. vc\_reg, vc\_bsreg, and vc\_preg are commands used for the estimation of SVCM across a selected set of points of interest, providing different alternatives for the estimation of standard errors. vc\_predict and vc\_test are commands that can be used to obtain model predictions and residuals, provide some statistics of the estimated model, as well as provide some specification tests. vc\_graph can be used to plot the smooth coefficients.

The rest of the paper is structured as follows. Section 2 reviews the estimation of SVCM. Section 3 provides a detailed review of the implementation procedures and commands used for model selection, estimation, and post-estimation. Section 4 illustrates the commands, and section 5 concludes.

1. **Non-parametric regression and Smooth varying coefficient models**
   1. **Nonparametric Regressions**

Consider a model where is the dependent variable and is a set of exogenous independent variables of dimension . Without any assumption on the relationships between these variables, and assuming no omitted variable problem exists, the non-parametric regression model of given the dimensional vector of variables is given by:

|  |  |
| --- | --- |
|  | (1a) |
|  | (1b) |

Essentially, this model specification implies that is related to following some unknown nonlinear functional form. The literature on non-parametric regressions suggests that these types of models can be estimated in at least two ways. On the one hand, the function can be estimated by modeling the conditional mean function as a locally weighted average estimator:

|  |  |
| --- | --- |
|  | (2) |

Where is a vector of bandwidths, is a joint kernel function:

|  |  |
| --- | --- |
|  | (3) |

and is a kernel function defined by the point of reference and the bandwidth

|  |  |
| --- | --- |
|  | (4) |

This function puts more weight to observations that are close to the point , and uses the vector of bandwidths to determine how much information is used for the estimation of the conditional mean. This procedure can be implemented in Stata using the command npregress kernel**.**

An alternative procedure is to estimate using a set of predefined transformations and interactions of the original variables as explanatory variables. The most common practice is to use polynomial or splines basis of the original variables , and its interactions , and estimate the following model:

|  |  |
| --- | --- |
|  | (5) |

Where are all the coefficients associated with each one of the terms of the transformations and interactions . In this setting, the dimension of , or more specifically of and , represent the tuning parameter that determines the roughness of . This procedure can be implemented in Stata using the command npregress series, using polynomials, splines, and B-splines basis**.**

As described in Li and Racine (2007) and Stinchcombe and Drukker (2013), in the case of kernel methods, the effective number of observations for the estimation of the conditional mean decreases rapidly as increase and goes to zero. In the case of transformations and interactions, the number of parameters that need estimation increase exponentially with the number of explanatory variables and the dimension of & , rapidly reducing the degrees of freedom of the model.[[4]](#footnote-5)

* 1. **Smooth Varying Coefficient Model (****SVCM)**

SVCM, as introduced by Hastie and Tibshirani (1993), assume that there is some structure in the model. Instead of estimating a function like equation (1), the authors suggest distinguishing two types of independent variables . are variables that have a linear effect of , but those effects are some unspecified nonlinear functions of . This model is defined by:

|  |  |
| --- | --- |
|  | (6a) |
|  | (6b) |

This specification reduces the problem of curse of dimensionality of the estimated model, compared to (1), by assuming have a parametric effect on , conditional on , allowing the coefficients to be unknown smooth nonlinear functions of . For simplicity, I will refer to as the set of smoothing variables. The existence of two types of variables raises the question of deciding which variables should be included in or . The empirical literature suggests that deciding which variables should be considered as part of the smoothing variables Z will depend on the research question of interest.

For example, Li et al. (2002) analyze the production function of the nonmetal mineral manufacturing industry in China, analyzing the marginal productivity of capital and labor (X), analyzing the heterogeneity based on expenditure on intermediate production and expenditure on management (Z). Liu and Egan (2019) analyze recreation demand, focusing on the effect of travel cost and household income on households willingness to pay (), allowing for heterogeneity across demographic characteristics, possession of a hunting/fishing license, and environmental organization membership (). Centorrino and Racine (2017) revisit the role of experience, race, and geographical location () as determinants of wages, analyzing the heterogeneity across education attainment (). Polemis and Stengos (2015) analyze labor productivity as a function of labor share ratio, market size, capital, intermediate inputs, and energy cost (), analyzing the heterogeneity across a measure of market concentration ().

Just like with non-parametric regressions, various methods have been proposed for the estimation of this type of model. Hastie and Tibshirani (1993) suggest estimating using spline basis or penalized splines with respect to Z. Hoover et al. (1998) and Li et al. (2002), suggest instead using local polynomial kernel regressions as a feasible strategy to estimate . More recently, Li and Racine (2010) expanded on the use of kernel methods for the estimation and inference of these types of models when is a mixture of continuous and discrete data.[[5]](#footnote-6) In the next section, I describe the estimation of SVCM using kernel methods when there is a single smoothing variable in .

* 1. **SVCM: Local Kernel estimator**

Consider a simplified version of the SVCM (equation 6), as described in Li et al. (2002), where is the dependent variable, is a single continuous variable, and is a set of variables including a constant. Because contains a single variable, the bandwidth will be a single scalar, dropping the subscript from equation (4).

Following Li and Racine (2007), the coefficients in equation (6) can be derived as follows. Starting from equation (6), premultiply both sides by take expectations conditional on , and solve for , which yields:

|  |  |
| --- | --- |
|  |  |
|  | (7) |

Using sample data, equation (7) can be an unfeasible estimator of because there may be few or no observations for which , making impossible to estimate.[[6]](#footnote-7) As an alternative, a feasible estimation for equation (5) can be obtained using kernel methods, for any point :

|  |  |
| --- | --- |
|  | (8a) |

Or the equivalent in matrix form:

|  |  |
| --- | --- |
|  | (8b) |

Where is the kernel function, as defined in (4), that puts more weight to observations where is closer to , given the bandwidth . is a diagonal matrix with the element equal to . Equation (8b) constitutes the local constant estimator of SVCM.

A drawback of the local constant estimator is that it is well known for its potentially large bias when estimating functions near boundaries of the support of . A simple solution to reduce this bias is to use a local linear estimator, based on a first-order approximation of the coefficients . This implies that instead of estimating equation (6), one can instead estimate the following model:

|  |  |
| --- | --- |
|  |  |
|  |  |
|  | (9) |

This implies that an approximation for can be obtained using a linear expansion with respect to , and that the closer is to z, the more accurate will be the approximation.

Define to be the row of , and to be the Kronecker product, such that indicates that each variable in is multiplied by the auxiliary variable (. Based on equation (8b) the coefficients and can be estimated as:

|  |  |
| --- | --- |
|  | (10) |

Where constitute the local linear estimator of and is the first derivative of that coefficient with respect any point

* 1. **Example: SVCM and Weighted Least Squares**

While it may not seem evident, equations (6) and (9) show that the estimation of SVCM using kernel methods can be easily obtained using weighted ordinary least squares, where weights are defined by the kernel functions. To show this, consider the dataset “Fictional data on monthly drunk driving citations” (dui.dta) and a simple model that assumes that citations are a linear function of college, taxes, citizen, and fines. This model can be estimated using the following command:

regress citations i.college i.taxes i.csize fines

Say that you are interested in analyzing how the effect of college, taxes, and csize varies as a function of fines.[[7]](#footnote-8) Assume for simplicity that you are interested in one point of the distribution: Fines at the 10th percentile (=9). In this example, there are enough observations with values exactly equal to 9, so it is possible to estimate the model using this constraint. Because we are estimating regressions for specific values of fines, this variable is taken out of the specification:

regress citations i.college i.taxes i.csize if fines==9

In general, it may be more convenient to estimate the model using kernel functions as weights. As discussed in the literature, the choice of kernel function is not as important as the choice of bandwidth. For simplicity, I will use a gaussian kernel with a bandwidth . This is directly implemented using the normalden() function, with the smoothing variable fines as the first argument, the point of interest (9) as the second argument, and the bandwidth () as the third argument:

regress citations i.college i.taxes i.csize [aw=normalden(fines,9,0.5)]

This example implements the local constant estimators following equation (7). For the implementation of the local linear estimator, an auxiliary variable needs to be constructed () df=fines-9. This variable is created and added to the model specification creating interactions with all other explanatory variables. Using factor notation, this is straight forward:

regress citations i.(college taxes csize)##c.df [aw=normalden(fines,9,0.5)]

To see how these models compare to each other, figures 1a and 1b provide a simple plot of the coefficients associated with college and taxes, using the three specifications sketched above, using every distinct value of fines comparing them to the standard regression estimations.

**Figure 1. VCM across fines: College and Taxes**



“VCM-Exact” corresponds to the models that constrain data to , whereas “SVCM-LC” and “SVCM-LL” indicate the estimates come from the local constant and local linear estimators of the SVCM model, respectively. You will notice that there are no estimations for the “VCM-Exact” model at the boundaries of the distribution of fines because there are simply not enough observations to obtain those estimates. Also, notice that the “VCM-Exact” produces coefficients that are very volatile. Both “SVCM-LC” and “SVCM-LL” produce smooth plots. The local constant estimators are somewhat flat at the boundaries of the distribution, which is to be expected. In contrast, the local linear estimator seems to be less affected by the boundary bias, following more closely “VCM-Exact” coefficients. At this point, however, nothing can be said in terms of statistical inference regarding the merits of either model.

While this simple illustration shows the simplicity of estimating SVCM, there are many details regarding model choice and statistical inference that requires further examination. In the next section, I discuss some of the details regarding these problems, introducing the commands in vc\_pack that can be used to estimate SVCM with a single smoothing variable.

1. **Smooth Varying coefficient models: vc\_pack**
   1. **Model selection: vc\_bw and vc\_bwalt**
      1. **Leave-one-out cross-validation**

The most important aspect of the estimation of SVCM is the choice of the bandwidth parameter . While larger bandwidths may help to reduce the variance of the estimations, by allowing more information to be used on the local estimation process, it will increase the bias of the estimators, by restricting the model flexibility. In contrast, smaller bandwidths can reduce bias, allowing for greater flexibility in the estimation, but at the cost of higher variability.[[8]](#footnote-9)

The illustration presented in the previous section is an example of this phenomenon. The standard OLS coefficients can be considered as an extreme scenario where the bandwidth is so large that all observations receive equal weight regardless of the point of interest. This is guaranteed to obtain the minimum variance for the estimated parameters, but with a potentially large cost in terms of model bias. On the opposite side of the spectrum, the results where the regressions are estimated using samples restricted to observations with a specific value of fines (VCM-Exact) are based on a bandwidth that is essentially. While this is the most flexible model possible given the data, figure 1 also shows that the results are highly volatile, and estimations were not feasible for some areas.

While there are many suggestions in the literature regarding bandwidth selection (see for example Zhang and Lee (2000)), the methodology used here is based on a leave-one-out cross-validation procedure. Consider the model described in equation (6), and a sample of size . The optimal bandwidth is such that it minimizes the cross-validation criteria (CV) defined as:

|  |  |
| --- | --- |
|  | (11) |

Where is the leave-one-out estimator of conditional on a bandwidth , that excludes the observation, and is the leave-one-out prediction of SVCM. is a weighting function that is used to reduce the influence of areas where the distribution of is scant. While this seems a very computationally intensive process that requires the estimation of different sets of parameters, the actual estimation of the criteria requires the estimation of fewer equations based on characteristics of the data and properties of linear regressions.

On the one hand, even if is a continuous variable in nature, it is often recorded as partially discrete data. A person’s age, for example, is a variable that is continuous in nature but is often measured in terms of years. This implies that the number of distinct coefficients are likely to be fewer than the number of observations in the sample .

On the other hand, the estimation of the does not require the explicit estimation of , but the estimation of the leave-one-out error . With linear regressions, it is possible to obtain by re-scaling the SVCM error using the leverage statistic () [[9]](#footnote-10):

|  |  |
| --- | --- |
|  | (12) |

Where is the local leverage statistic, defined as the diagonal element of the local projection matrix :

|  |  |
| --- | --- |
|  | (13) |

Using this shortcut, can be rewritten to reflect only the number of necessary regressions that need to be estimated. Consider the vector of all unique values of , with . Using this, the can then be written as:

|  |  |
| --- | --- |
|  | (14) |

While (14) shows the number of estimated equations () is potentially smaller than the total number of observations in the sample (), in some applications maystill be too large to allow for a fast estimation of . A feasible alternative in such cases is to use what Hoti and Holmström (2003) and Ichimura and Todd (2007) denominates as block or binned local linear regressions, to obtain an approximation of the criteria.

Consider the vector of all unique values of which are organized in non-overlapping bins of width , and a center equal to , such that:

|  |  |
| --- | --- |
|  | (15a) |
|  | (15b) |

Instead of estimating set of parameters, for each distinct value of , one estimates sets of parameters using the points of reference . These parameters are used to obtain linear approximations around for the predicted values (), predicted errors () and leverage statistics , for all observations within their corresponding bins:

|  |  |
| --- | --- |
|  | (16a) |
|  | (16b) |
|  | (16c) |

Using these expressions, an approximation for the leave-one-out error () for observation with , and can be approximated as follows:

|  |  |
| --- | --- |
|  | (17) |

This can be used to obtain an alternative expression for the criteria:

|  |  |
| --- | --- |
|  | (18) |

Which reduces the number of estimated equations from to It is straight forward to see that as the number of groups P increase, and smaller is the bin width is, the better will be the approximation of to As shown in Hoti and Holmström (2003), binned local linear kernel regressions can provide good approximations for of the overall model predictions, as long as the ratio between the implicit bandwidth used for the construction of the bins and optimal bandwidth is relatively small.[[10]](#footnote-11) In addition, even if one considers the bandwidth based on the approximation to be a poor approximation of the full information bandwidth, it can still be used for exploratory analysis and as a starting point for the estimation of , reducing the computational cost of bandwidth selection.

* + 1. **Automatic model selection**

vc\_pack offers two commands for the automatic model selection based on the modified Cross-validation procedure previously described, minimizing the objective function. vc\_bwimplements a Newton-Raphson type algorithm that works well when the objective function is smooth and differentiable, with a local minima. This is an iterative algorithm that searches for optimal bandwidth using:

|  |  |
| --- | --- |
|  | (19) |

stopping when and are sufficiently close and selecting . The first and second-order derivatives are estimated using numerical methods with three points of reference. The scalar is equal to 1 as long as there is an improvement in the maximization process (i.e. , otherwise, is reduced by half until an improvement is found.

vc\_bwaltimplements a bisection type algorithm that works well in a larger set of scenarios, especially when is not smooth nor differentiable function of , but it may be slower in finding the optimal bandwidth. The algorithm starts with three points of reference: . If the optimal bandwidth is between and (i.e. and ) the algorithm will evaluate the cross-validation criteria using midpoints between and , and update the reference points so , with corresponding to the bandwidth with the lowest , and and corresponding to the two closest, previously evaluated, points of reference that are above and below . If the is potentially smaller than (i.e. ), a fourth point will be evaluated until finding a point such that , which suggests is between and . A similar process is implemented if is potentially larger than . The algorithm stops when and are sufficiently close, selecting .

Both commands use the following syntax:

vc\_bw[alt] depvar indepvar [if], vcoeff(svar) [kernel(kernel) bwi(#) knots(#k) km(#km) trimsample(trimvar) plot]

Where depvaris the dependent variable , indepvaris the list of all independent variables that we assume to have a conditional linear effect on the dependent variable , and svar is the smoothing variable Z.

kernel(.) indicates the kernel function (see equation 4) that will be used to create the local weights and estimate the local regressions. The default is the Gaussian kernel, but other kernels are allowed.[[11]](#footnote-12)

bwi(#) provides the command with an initial value for searching for the optimal bandwidth. The default option uses the bandwidth from the command lpolyusing the same kernel function declared in kernel()**.**

knots(#k) and km(#km) are options that can be used to request the minimization of the approximate criteria as described in equation (18). Using knots(#k), with #k, requests the creation of a new variable that groups the smoothing variable svar into groups of equal width. Using knots(0) indicates to create groups, where is the nearest integer of . When knots(0) is used, one can also use the option km(#km), so that is the nearest integer of .[[12]](#footnote-13) Whenever knots(#k) is used, the command reports the number of knots employed, and the implicit bin width (see equation 15a) .

The default is to use all distinct values in the smoothing variable, up to 500 distinct values. When more than 500 distinct values are detected, the command uses the options knots(0) km(2). While there is nothing to indicate that this rule of thumb provides the most appropriate number of knots and implicit bin width (), simulations presented in Hoti and Holmström (2003) suggest that the approximate criteria is reasonable if when using Gaussian kernels, and , when using epanechnikov, biweight and triangular kernels.

Using the option knots(-2) requests the estimation of the CV criteria for all distinct values in the conditioning variable.

trimsample(trimvar) provides the name of a binary variable (trimvar) that indicates the subsample of the data that will be used for the estimation of the criteria. Observations with trimvar equal to zero are not used for the calculation. This plays the role of the weighing function .

The option plot requests the command to plot all the bandwidths and estimated internally. This can be used for visual inspection to verify the bandwidth is indeed minimizing the objective function.

After finishing the minimization process, the program stores the optimal bandwidth, the kernel function, and the smoothing variable name as globals: $opbw\_, $kernel\_, and $vcoeff\_. This is done so other programs in the package can reuse this information.

* 1. **Model estimation and inference: vc\_reg, vc\_preg and vc\_bsreg**
     1. **Estimation of the variance-covariance matrix**

As shown in section 2, once the bandwidth is selected, the estimation of the SVCM is a simple process that requires three steps:

S1. Select the point or points of interest for which the model will be estimated. (typically a subset of all possible values of the smoothing variable ),

S2. Construct the appropriate kernel weights, based on the points of interest, selected kernel function and the optimal bandwidth , and

S3. Construct the auxiliary variable , which will be interacted with all independent variables in the model.

Once the auxiliary variables have been created, one can obtain the model coefficients, as well as their gradients, conditional on all the selected points of interest, by estimating equation (9) using kernel weighted least squares as in equation (10). The next step is the estimation of standard errors of the estimated parameters to obtain statistical inferences from the SVCM.

Following Li et al. (2002) and Li and Racine (2007, 2010), a feasible estimator for the asymptotic variance-covariance matrix of the SVCM, given a point of interest and bandwidth , can be obtained as follows: [[13]](#footnote-14)

|  |  |
| --- | --- |
|  | (20) |

Where is a diagonal matrix where the element is equal to , and and are defined as in equation (10). There is little guidance in the literature regarding in the context of kernel regressions. Li et al. (2002) and Li and Racine (2007, 2010) assume , which is valid asymptotically. Notice, however, that the expression given by equation (17) is the same as the robust standard errors for weighted least squares. Standard practice in those cases is to use , where indicates the total number of coefficients that need to be estimated in the model, and is the sample size. In semiparametric and nonparametric models, however, one must differentiate between sample size N, and effective sample size (see section 3.3 expected kernel observations).

Following the literature on the estimation of robust standard errors under heteroscedasticity (Long and Ervin 2000), it is also possible to estimate the variance-covariance matrix substituting with or in the diagonal matrix D, where is the leverage statistic as defined in equation (13). In this case . This is equivalent to the estimation of HC2 and HC3 standard errors. According to Long and Ervin (2000), for the standard linear model, HC2 and HC3 outperform robust standard errors when the model is heteroskedastic and samples are relatively small (N<250). While there is no formal study on the use of HC2 and HC3 standard errors when combined with SVCM, it is my conjecture that these standard errors may also be better than robust standard errors when the expected/effective sample size is small.

There is a debate regarding the use of analytical variance-covariance matrices in the framework of nonparametric kernel regressions. Cattaneo and Jansson (2018) advocate for the use of resampling methods, in specific paired bootstrapped samples, to obtain correctly estimate the variance-covariance matrix of the estimated coefficients when estimating kernel-based semiparametric models. In fact, they indicate percentile-based confidence intervals provide better coverage because paired bootstrap automatically corrects for non-negligible estimation bias.[[14]](#footnote-15) Broadly speaking the paired bootstrap procedure, adapted to the estimation of SVCM, is as follows:

S1. Using the original sample , Estimate the coefficients and , using the bandwidth , and all points of interest.

S2. Obtain a paired bootstrapped sample with replacement from the original sample, and estimate and using the same points of interest as in S1, and bandwidth .

S3. Repeat S2 times. The bootstrapped standard errors for the coefficients are defined as:

|  |  |
| --- | --- |
|  | (21) |

Where and are vectors containing all the coefficients and that were estimated for each bootstrap sample . The percentile confidence interval is defined as the lower and upper quantiles of the empirical distribution of and , where is the significance level.

* + 1. **Implementation: vc\_reg, vc\_preg and vc\_bsreg**

vc\_pack offers three commands for the estimation of SVCM, offering various alternatives for the estimation of the variance-covariance matrix (). vc\_regandvc\_pregestimate SVCM using equation (20) for the estimation of , using different definitions for the model error . Bootstrapped standard errors and percentile-based confidence intervals can be obtained using the command vc\_bsreg.

vc\_preguses the SVCM error defined as , for the estimation of the asymptotic standard errors. vc\_reg, instead, uses , which is the local linear approximation of for the point of reference .

While vc\_pregproduces the correct asymptotic standard errors, as suggested by Li and Racine (2007, 2010), it may be slow because the command estimates the SVCM for all points of the smoothing variable in order to obtain the . vc\_reg is faster by default because it only uses the local linear approximation and does not require additional steps for the estimation of the standard errors. These standard errors, however, contain approximation errors that increase the farther is from the point of reference , but can be used as a first quick approximation to analyze the data and draw statistical inferences. Empirically, vc\_reg produces results that are comparable to the ones produced by vc\_pregbecauseobservations where and differ greatly will have a small influence in the estimation of standard errors (equation 20) because is also likely to be far from the point of reference .

The three commands share the same basic syntax:

vc\_[bs|p]reg depvar indepvar [if] [in], [vcoeff(svar) kernel(kernel) bw(#) k(#) klist(numlist)]

Similar to vc\_bw[alt], depvar is the dependent variable , indepvar are the set of explanatory variables () that will have a linear effect on depvar, conditional on the smoothing variable svar . kernel(.) and bw(#) are used to provide specific information regarding the model estimation. The default option is to use information stored in $vcoeff\_, $kernel\_ and $opbw\_.

Because the richness of SVCM comes from the estimation of the linear effects as a function of the smoothing variable , these commands offer two alternatives to select the points of interest over which the local regressions will be estimated. The option k(#), which must be equal or larger than 2, requests to estimate regressions using equidistant points between the 1st and 99th percentile of svar. klist(numlist) request to estimate local linear regression using each number of the list numlist as a reference point. When klist() contains a single number, the standard regression output is reported. Otherwise, when k(#) or klist(numlist) are used to estimate 2 or more models, vc\_[bs|p]reg produce no output but stores the betas and variance-covariance matrices for each regression as a separate matrix in e(). This information can be used to create plots of the coefficients across svar. Both vc\_reg and vc\_preg produce robust standard errors by default (equation 20), but can also report HC2 and HC3 standard errors by using hc2 or hc3 as options. Clustered standard errors are also possible using the option cluster(cluster varname), but cannot be combined with HC2 or HC3 options.

Because vc\_preg requires full information errors for the estimation of the variance-covariance matrix, by default, the command will obtain predictions for the errors and leverage statistics , using all distinct values of the smoothing variable (svar). Because this can be computationally costly, similar to our discussion on the calculation of the cross-validation criteria, it is possible to use the options knots() and km() to reduce the number of internally estimated regressions. This command uses the same default options as vc\_[alt]bw. When binning options are used, the errors and leverage approximations defined in (16a-16c) are implemented. Alternatively, it is also possible to provide the command with previously estimated sample errors ( and leverage statistics (using options err(err varname) and lev(lev varname).

vc\_bsreg estimates bootstrap standard errors using a paired bootstrapped strategy. Following the syntax for the command bootstrap, one can specify information for strata() and cluster(), as well as define a seed() for the reproducible generation of the random samples. The default number of bootstrap samples is 50, but that can be changed using the option reps(#). In addition to the bootstrapped standard errors, vs\_bsreg also stores the 95% percentile confidence interval, but it can be changed to other levels using the option pci(#), using any number between 0 and 100.

When estimating a single equation, vc\_reg, vc\_preg, and vc\_bsreg stores the 2 variables in the dataset: \_delta\_, containing , and \_kwgt\_ containing the standardized kernel weights (see the next section).

* 1. **Model post-estimation: vc\_predict and vc\_test**

vc\_pack provides two commands that can be used to obtain summary statistics of the model, as well as report some tests for model specification against parametric alternatives. The first command, vc\_predict, has a similar syntax as vc\_[alt]bw and vc\_[p|bs]reg:

vc\_predict indepvar depvar, [vcoeff(svar) kernel() bw() knots() km() stest] [yhat(newvar) res(newvar) lvrg(newvar) looe(newvar) stest]

In addition to the options previously described, vc\_predict can be used to obtain predictions of the model -yhat(newvar)-, the model residual -res(newvar)-, the leverage statistic -lvrg(newvar)-, or the leave-one-out error -looe(newvar) (equation 12 and 13). Each one of these options requires to specify a new variable name (newvar) to store the specified information. One can also use the options knots()and km() to speed up the computing the process, in which case the approximations described in (16a-16c) are used.

Residuals and leverage from this command can be used, for example, for the estimation of SVCM using vc\_preg. This command also provides some basic information about the model, as well as perform some specification tests when option stest is used. The next section describes the methods and formulas reported by this command.

* + 1. ***Log Mean Squared Leave-one-out Errors***

Consider the SVCM described in equation (6). Given the smoothing variable (svar), kernel function (kernel()), and bandwidth (bw()), vc\_predict reports the log of the mean squared leave-one-out error:

|  |  |
| --- | --- |
|  | (22) |

when no binning option is used, or its approximation:

|  |  |
| --- | --- |
|  | (23) |

when binning options (knots() km()) are used. This is the same statistic used for the model selection except that it does not use the weighting factor for its calculation.

* + 1. ***Goodness of Fit***

vc\_predict produces two measures of goodness-of-fit statistic that direct analogs to standard linear models. The first one is based on the standard decomposition of the sum of squares:

|  |  |
| --- | --- |
|  | (24) |

Which is the same one used by npregress kernel. Because this statistic is known to produce undesirable results, such as negative values for , vc\_predict also reports the goodness-of-fit statistic suggested in Henderson and Parmeter (2015):

|  |  |
| --- | --- |
|  | (25) |

When binning options are used, is substituted by in equations (24) and (25).

* + 1. ***Model and Residual Degrees of Freedom***

The effective number of degrees of freedom is a statistic that has proven useful in the literature of non-parametric econometrics for the comparison of models with different types of smoothers. Following the terminology from Hastie and Tibshirani (1990), consider any parametric and non-parametric model with a projection matrix of dimension such that , where is a vector of the predicted values corresponding to any particular model. Hastie and Tibshirani (1990)emphasizes two estimators for the estimation of the number of degrees of freedom:

|  |  |
| --- | --- |
|  | (26a) |
|  | (26b) |

In the context of linear regression models, where the projection matrix , these definitions are equivalent to each other. However, in the case of kernel regressions and penalized smooth spline regressions, the matrix is not symmetrical and the above definitions of degrees of freedom will differ from each other. is commonly used as an approximation of the number of degrees of freedom of the model, whereas is used as the number of residual degrees of freedom.

For the specific case of SVCM, the projection matrix S is defined as follows:

|  |  |
| --- | --- |
|  | (27) |

Where is a matrix with the diagonal element is equal to 1 if , and zero elsewhere. This implies that the first measure of degrees of freedom is equivalent to:

|  |  |
| --- | --- |
|  | (28) |

The second measure of degrees of freedom is computationally more difficult to estimate, as it requires operations. As an alternative, Hastie and Tibshirani (1990)suggest using the following approximation:

|  |  |
| --- | --- |
|  | (29) |

vc\_predict reports and as measures of model and residual degrees of freedom, respectively. When binning options are used, is substituted by in equation (28).

* + 1. ***Expected Kernel Observations***

One of the drawbacks of nonparametric regression analysis is the rapid decline of the effective number of observations used for the estimation of the parameters of interest, the larger is the number of explanatory variables used in the model (the curse of dimensionality), and smaller are the bandwidths. To provide the user with a statistic summarizing the amount of information used through the estimation process, its common practice to report as the expected number of Kernel observations , where is the product of all bandwidths of the explanatory variables.[[15]](#footnote-16) This statistic, however, can be misleading.

Consider the estimation of a model with a single independent variable, for which an optimal bandwidth is selected. If the scale of the independent variable doubles, the optimal bandwidth of the rescaled variable will double but should remain the same. The statistic , however, suggests that the has doubled as well.[[16]](#footnote-17)

As an alternative measure to , I propose a statistic based on what I denominate standardized kernel weights , which are defined as:[[17]](#footnote-18)

|  |  |
| --- | --- |
|  | (30) |

This kernel weights are guaranteed to fall between 0 and 1. While this change in the scale of local weights have no impact on the estimation of the point estimates of the models, it provides a more intuitive understanding of the role of weights in the estimation process. Observations where is equal to will receive a weight of 1, and one can consider that the information of that observation is fully used when estimating the local linear regression. If an observation has a of say 0.5, one can consider that the information contributed by that observation to the local kernel regression is half of an observation where . Finally, observations with a do not contribute to the local estimation at all. These kernel weights can be used to estimate the effective number of observations () used for the estimation of the parameters of interest for a given point of reference :[[18]](#footnote-19)

|  |  |
| --- | --- |
|  | (31) |

Because areas with higher density use more observations than areas where z is sparsely distributed, the expected number of kernel observations can be defined as the simple weighted average of using all observations in the sample. This leads to the following:

|  |  |
| --- | --- |
|  | (32) |

Where is the number of observations I with When binning options are used, the estimator is:

|  |  |
| --- | --- |
|  | (33a) |
|  | (33b) |

Where is the number of observations that fall within the bin.

If is continuous, this statistic has two convenient properties respect to the bandwidth :

|  |  |
| --- | --- |
|  | (34) |

This provides a more intuitive understanding of the effect of the bandwidth has on the average amount of information used for the estimation of local regressions compared to the standard parametric model. At the very least, there will be one observation for the estimation of the local estimation, and at the most, all the data will be used for each local estimation. This statistic is also reported after vc\_predict.

* + 1. ***Specification tests***

In addition to reporting the basic summary statistics described above, vc\_predict can also produce basic specification tests when the option stestis specified. The specification tests follow Hastie and Tibshirani (1990) and provide what the authors call *approximate* F-test, comparing the SVCM to 4 parametric alternatives:

|  |  |
| --- | --- |
| with | (35a) |
| with | (35b) |
| with | (35c) |
| with | (35d) |

Where is the number of explanatory variables defined in plus the constant. Define as the residual degrees of freedom of the SVMC (see equation (26)), and to be the predicted residual for the parametric model (0, 1, 2 or 3). The approximate F statistic is defined as:

|  |  |
| --- | --- |
|  | (36) |

The null hypothesis is that the parametric model (0, 1, 2 or 3) is correctly specified, whereas the alternative hypothesis is that states the SVCM is correct. While the exact distribution of this statistic is not known, Hastie and Tibshirani (1990):p65 suggests using critical values for a -statistic with degrees of freedom in the numerator and degrees of freedom in the denominator, a rough test for a quick inspection of the model specification. When binning options are used is substituted by (equations 16a-16c).

* + 1. ***Cai, Fan, and Yao (2000) specification test:* vc\_test**

Because the exact distribution of the approximate F statistic is not known, vc\_pack also offers the implementation of the specification test proposed by Cai, Fan, and Yao (2000), based on a wild bootstrapped approach, as described in Henderson and Parmeter (2015). The test statistic is constructed in a way similar to the approximate F-statistic, but without adjusting for the differences in degrees of freedom:

|  |  |
| --- | --- |
|  | (37) |

Where corresponds to the residuals for the parametric model, see equations (35a-35d), and corresponds to the residuals of the SVCM. The null hypothesis (), which states that the parametric model is correctly specified, is rejected in favor of the SVCM if the statistic is above some critical value.

Because the distribution of the statistic is not known, a wild bootstrapped procedure can be used to obtain its empirical distribution using the following procedure:

S1. Define to be predicted residual based on the parametric model (35a-35d).

S2. Construct a new dependent variable , using a two-point wild bootstrapped error as follows:

Where follows a Bernoullidistribution with

S3. Using the new dependent variable , re-estimate the parametric model and SVCM, using the optimal bandwidth , and calculate the statistic

S4. Repeat S2 and S3 a sufficient number of times, to obtain the empirical distribution of the statistic .

If is larger than the upper -percentile of the empirical distribution obtained through the wild bootstrapped procedure, one can reject the null hypothesis.

The command vc\_test implements this specification test using the following syntax.

vc\_test indepvar depvar, [vcoeff(svar) kernel() bw() knots() km() degree(#d) wbsrep(#wb)]

Similar to previous commands, one has to specify the dependent and independent variables in the model but specifying vcoeff(svar), kernel and bandwidth are optional. The program uses the information stored by vc\_[alt]bw by default. Because the test requires the estimation of the whole model multiple times, one can specify the options knots() and km() to implement the binned version of the statistic, and increase the speed of the computations. This substitutes with in equation (37).

degree(#d) is used to define the model under the null hypothesis. #d can take the values 0, 1, 2 or 3, which corresponds to the models described in equations (35a)-(35d). The default is degree(0)

wbsrep(#wb) is used to indicate the number of wild bootstrap repetitions used for the estimation of the empirical distribution of the statistic . The default number of repetitions is 50. The command reports the 90th, 95th and 97.5th percentiles of the empirical distribution of to be used as critical values.

* 1. **Model visualization: vc\_graph**

One attractive feature of semiparametric models in general, and SVCM in particular, is the potential to visualize effects across the range of the explanatory variables that enter the model non-parametrically. These plots can be used for a richer interpretation of marginal effects. As described in section 3.2, when vc\_[bs|p]reg is used to estimate models for more than 1 point of reference, the command produces no report, but stores the coefficients, variances and confidence intervals in e().

vc\_graph is a command that can be used as a post-estimation tool to produce plots of coefficients of the independent variables, or their gradients, using the information estimated via vc\_[bs|p]reg. The command uses the following syntax:

vc\_graph [indevpar], [constant delta ci(#) ci\_off pci graph(stub) rarea addgraph() xvar(varname)]

indevpar can contain a subset of all independent variables used in the estimation of the SVCM. If factor variables and interactions were used, the same format must be used when using vc\_graph.

constant is used to plot the varying coefficients associated with the constant.

delta request vc\_graph to plot the gradients of the variables listed in indevpar. The default is to plot the coefficients . If the options delta and constant are used, vc\_graph will plot the coefficient of the auxiliary variable .

ci(#) sets the level of the confidence intervals, using any number between 0-100. The default is 95%. Confidence intervals can be omitted from the plot using the option ci\_off.

When SVCM is estimated using vc\_bsreg, one can also request using the percentile-based confidence intervals using the option pci. The level of confidence, in this case, must be set when the model is estimated using vc\_bsreg.

Confidence intervals in figures use range plots with capped spikes by default, but plots with area shading can be requested using the option rarea**.**

All graphs produced by vc\_graph are stored in memory with the name “grph#”, which are numbered consecutively. The names of the store graphs can be changed using graph(stub), where *stub* would be used instead of *grph* to store the graphs in memory. Simple plots can be added to this graph by using the option addgraph() and specifying within quotes the options of the graph.

Finally, vc\_graph offers the option xvar(xvarname) to use a different variable to plot the smooth varying coefficients, as long as this variable xvarname is a monotonic transformation of the original variable svar used of the estimation. For example, say that the SVCM model was estimated using the variable svar as the smoothing variable because it has fewer areas with scant distribution. The researcher, however, is interested in plotting coefficients across svar1 rather than svar. If svar1 is a monotonic transformation of svar, using the option xvar(svar1) request plotting coefficients using svar1 in the horizontal axis. Internally, the mapping between the points of reference from svar to svar1 is done using a local linear approximations, if the exact values are not available.[[19]](#footnote-20)

1. **Illustration: Determinants of Drunk Driving citations: the role of fines**

For this illustration, I use the fictitious dataset dui.dta, presented in section 2.4, to analyze how the number of Drunk driving citations are affected by whether or not a jurisdiction taxes alcohol, if there is a college in the jurisdiction, or whether the jurisdiction is in a small, medium or large city, conditional on fines imposed for drunk driving.

I start the analysis by using vc\_bw to select the optimal bandwidth using the leave-one-out cross-validation strategy, and use the default options.

vc\_bw citations taxes college i.csize, vcoeff(fines)

Kernel: gaussian

Iteration: 0 BW: 0.5539761 CV: 3.129985

Iteration: 1 BW: 0.6870520 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\_

The command suggests a bandwidth of 0.7398 suggesting the bandwidth used in section 2.4 may have been under smoothing the results.

Next, I obtain simple summary statistics of the model using vc\_predict. I also request to report approximate F-test for model specification against the models where *fines* is added as an interaction in the model.

. vc\_predict citations taxes college i.csize, vcoeff(fines) stest

Smooth Varying coefficients model

Dep variable : citations

Indep variables : taxes college i.csize

Smoothing variable : fines

Kernel : gaussian

Bandwidth : 0.73977

Log MSLOOER : 3.11950

Dof residual : 477.145

Dof model : 18.684

SSR : 10323.126

SSE : 37886.249

SST : 47950.838

R2-1 1-SSR/SST : 0.78471

R2-2 : 0.79011

E(Kernel obs) : 277.828

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: 8.24686 with pval 0.00000

Model 1 y=x\*b0+g\*z+(z\*x)b1+e

F-Stat: 5.80948 with pval 0.00000

Model 2 y=x\*b0+gz+(z\*x)\*b1+(z^2\*x)\*b2+e

F-Stat: 0.75986 with pval 0.65167

Model 3 y=x\*b0+gz+(z\*x)\*b1+(z^2\*x)\*b2+(z^3\*x)\*b3+e

F-Stat: -2.07335 with pval 1.00000

The report indicates that the model uses approximately 18.7 degrees of freedom (equation 28), whereas the residuals have 477.15 degrees of freedom (equation 29). The model has an that is larger than the simple regression model (0.718), but is somewhat smaller than the obtained using the full nonparametric model (0.81).[[20]](#footnote-21) The second measure of (see equation (25)) is larger than the standard measure of goodness of fit. Finally, the expected number of kernel observations is 277.8 (equation 32), suggesting that, on average, half of the whole sample is used for each local regression.

The approximate F-test suggests rejecting models 0 and 1, in favor of the SVCM, but one cannot reject the null hypothesis that a model with a quadratic interaction with *fines* is correctly specified. The local fit of the model with a cubic interaction seems to be better than the SVCM, which explains why the F-statistic is negative. I also use vc\_test to implement the alternative specification test, comparing the same parametric models to the SVCM. For this example, I use 200 repetitions, using the option wbsrep(200). Because Model 0 was overwhelmingly rejected, and model 3 seems to have a better fit than the SVCM, only the results comparing against models 1 and 2 are shown:

vc\_test citations taxes college i.csize, degree(1) wbsrep(200) seed(1)

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:0.09382

95th Percentile:0.10351

97.5th Percentile:0.10686

vc\_test citations taxes college i.csize, degree(2) wbsrep(5) seed(1)

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:0.01177

95th Percentile:0.01490

97.5th Percentile:0.01726

The results are consistent with the approximate F-statistic. In the first case, the -statistic of the model is 0.16869, which is larger than the 97.5th percentile of the empirical distribution of the statistic, suggesting the rejection of the null of a linear interaction. The test comparing to the parametric quadratic interaction model is less conclusive. The -statistic is 0.0141, which suggests rejection of the null at 10% of confidence, but the null cannot be rejected at 5%. Despite these results, I’ll go ahead and estimate the SVMC.

To provide an overview of the results that are similar to the output of standard regression models, I provide the conditional effects using the 10th, 50th and 90th percentiles of fines as the points of reference, which correspond to 9, 10 and 11. This is done using the three commands available in vc\_pack, using the basic syntax:

vc\_reg citations taxes college i.csize, klist(9)

vc\_preg citations taxes college i.csize, klist(9)

vc\_bsreg citations taxes college i.csize, klist(9) seed(1)

vc\_reg citations taxes college i.csize, klist(10)

vc\_preg citations taxes college i.csize, klist(10)

vc\_bsreg citations taxes college i.csize, klist(10) seed(1)

vc\_reg citations taxes college i.csize, klist(11)

vc\_preg citations taxes college i.csize, klist(11)

vc\_bsreg citations taxes college i.csize, klist(11) seed(1)

The results from these regressions are shown in table 1, columns 2-4, showing the standard errors obtained with all three commands (Robust, F Robust, and Bootstrap). Column 1 presents the results for the standard regression model. For the SVCM two sub-columns are provided. The one on the left shows the conditional effects , whereas the one on the right shows the gradient of that effect .

Overall, robust standard errors obtained with the local linear approximation (vc\_reg) seems to be a reasonably good approximation to the full information robust standard errors (vc\_preg), with the largest discrepancies observed in areas where the density of the distribution of fines is low (at the top and bottom). These estimates are also consistent with the bootstrapped standard errors (vc\_bsreg).

**Table 1 Determinants of Number of Monthly citations, conditional on Fines. SVCM**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Number of monthly drunk driving citations | OLS | SVCM | | | | | |
|  | (1) | (2) | | (3) | | (4) | |
|  |  | Fines=9 | | Fines=10 | | Fines=11 | |
|  |  |  |  |  |  |  |  |
| 1 if alcoholic beverages are taxed | -4.494 | -6.377 | 3.008 | -3.959 | 1.093 | -3.843 | -0.0505 |
| Robust std err | (0.582) | (1.147) | (1.373) | (0.496) | (0.823) | (0.736) | (0.788) |
| F Robust std err |  | (1.059) | (1.210) | (0.493) | (0.787) | (0.711) | (0.751) |
| Bootstrapped std err | (0.638) | (1.322) | (1.525) | (0.498) | (0.967) | (0.812) | (0.833) |
| 1 if college town; 0 otherwise | 5.828 | 9.871 | -4.578 | 5.305 | -3.191 | 3.797 | -1.024 |
| Robust Std err | (0.588) | (1.113) | (1.318) | (0.516) | (0.896) | (0.888) | (0.963) |
| F Robust std err |  | (1.021) | (1.164) | (0.513) | (0.836) | (0.860) | (0.915) |
| Bootstrapped std err | (0.634) | (1.201) | (1.381) | (0.470) | (0.972) | (0.884) | (0.926) |
| **City Size** |  |  |  |  |  |  |  |
| medium | 5.492 | 6.734 | -1.299 | 5.284 | -2.332 | 3.051 | -2.196 |
| Robust Std err | (0.532) | (0.973) | (1.125) | (0.535) | (0.785) | (0.782) | (0.843) |
| F Robust std err |  | (0.936) | (1.069) | (0.538) | (0.786) | (0.772) | (0.831) |
| Bootstrapped std err | (0.547) | (0.932) | (1.265) | (0.588) | (0.760) | (0.833) | (0.958) |
| large | 11.24 | 14.99 | -4.863 | 10.60 | -3.779 | 7.784 | -2.691 |
| Robust Std err | (0.571) | (1.146) | (1.373) | (0.510) | (0.852) | (0.750) | (0.764) |
| F Robust std err |  | (1.071) | (1.233) | (0.509) | (0.822) | (0.741) | (0.751) |
| Bootstrapped std err | (0.610) | (1.095) | (1.323) | (0.553) | (0.809) | (0.749) | (0.812) |
| Drunk driving fines in thousands of dollars | -7.690 |  | -8.256 |  | -4.906 |  | -3.673 |
| Robust std err | (0.384) |  | (1.327) |  | (0.782) |  | (0.816) |
| F Robust std err |  |  | (1.211) |  | (0.792) |  | (0.804) |
| Bootstrapped std err | (0.405) |  | (1.473) |  | (0.787) |  | (0.810) |
| Constant | 94.22 | 23.96 |  | 16.80 |  | 12.93 |  |
| Robust std err | (3.949) | (1.168) |  | (0.474) |  | (0.746) |  |
| F Robust std err |  | (1.099) |  | (0.478) |  | (0.737) |  |
| Bootstrapped std err | (4.117) | (1.255) |  | (0.501) |  | (0.819) |  |
| N Obs and Kobs | 500 | 243.19 | | 341.64 | | 203.36 | |

Note: Robust std err corresponds to the output with vc\_reg, F Robust standard errors were estimated with vc\_preg, and Bootstrapped standard errors with vc\_bsreg. is defined as the sum of standardized kernel weights based on the point of reference (see equation 31).

To complement the information in this table, and before we provide an interpretation of the results, figure 2 provides a plot with 95% confidence intervals for the conditional effects of all variables in the model, using a predefined set of points of interest. First, I use vc\_preg to estimate the SVCM:[[21]](#footnote-22)

vc\_preg citations taxes college i.csize, klist(7.4(.2)12)

Estimating Varying coefficients models over 24 point(s) of reference

Smoothing variable: fines

Kernel function : gaussian

Bandwidth : 0.73977

vce : robust

Estimating Full model

More than 1 point of reference specified

Results will not be saved in equation form but as matrices

The main difference with the previous examples is that the option klist() contains a list of numbers, or reference points, over which I am requesting the SVCM to be estimated. It indicates that there are 24 points of reference, from 7.4 to 12. Once the estimation is finished, figure 2 can be reproduced with the following commands:

vc\_graph taxes college i.csize,

graph combine grph1 grph2 grph3 grph4

**Figure 2. SVCM: Conditional effects across *fines ***Note: Figures are taken as-is from vc\_graph, and combined with graph combine.

If one is interested in the gradients , they can be plotted using the following commands.

vc\_graph taxes college i.csize, delta

graph combine grph1 grph2 grph3 grph4

**Figure 3. SVCM: Change of the conditional effects across *fines*** **

Note: Figures are taken as-is from vc\_graph, and combined with graph combine.

An interpretation of these results can be given as follows. Overall, when alcoholic beverages are taxed, the number of monthly citations per month decreases in 4.5 units (table 1 col 1). This effect is larger in jurisdictions with low fines, with a point estimate ranging from 15 to just below 4, in jurisdictions with fines levels above 10. No differences can be observed in the conditional effect for fine levels above 10. This is mirrored by the fact that the estimate of in figure 3 are statistically equal to zero.

If there is a college campus in town, the number of citations per month is about 5.8 higher. The conditional impact of college declines as fines increase by almost 10 points between the minimum and maximum levels of fines in the distribution. Based on the estimates from figure 3, when fines are above 11, the change in the effect of college on citations is no longer statistically significant. If the jurisdiction is located in a medium city, the impact on the number of citations is relatively small, statistically significant, but shows no statistically significant change across fines. Finally, If the jurisdiction is located in a large city, the conditional impact is large, ranging from 30 to 10 additional citations per month, declining as fines increase. Something to be noticed from these figures is that most estimates for fines under 9 show large confidence intervals because less then 10% of the data falls below this threshold.

1. **Conclusions**

Smooth varying coefficient models are an alternative to full nonparametric models that can be used to analyze relationships between dependent and independent variables under the assumption that those relationships are linear conditional on a smaller set of explanatory variables. They are less affected by the curse of dimensionality problem because fewer variables enter the estimation non-parametrically. In this article, I provide a review of the model selection, estimation, and testing for these types of models, and introduce a set of commands, vc\_pack, that aim to facilitate the estimation of such models when coefficients are assumed to vary with respect to a single smoothing variable. An empirical application illustrates the usefulness of the procedure.

1. **References**

Cai, Zongwu, Mitali Das, Huaiyu Xiong, and Xizhi Wu. 2006. "Functional coefficient instrumental variables models." *Journal of Econometrics* 133 (1):207-241. doi: https://doi.org/10.1016/j.jeconom.2005.03.014.

Cai, Zongwu, Jianqing Fan, and Qiwei Yao. 2000. "Functional-Coefficient Regression Models for Nonlinear Time Series." *Journal of the American Statistical Association* 95 (451):941-956. doi: 10.1080/01621459.2000.10474284.

Cameron, A. Colin, and Pravin K. Trivedi. 2005. *Microeconometrics: Methods and Applications*. New York: Cambridge University Press.

Cattaneo, Matias D., and Michael Jansson. 2018. "Kernel-Based Semiparametric Estimators: Small Bandwidth Asymptotics and Bootstrap Consistency." *Econometrica* 86 (3):955-995. doi: 10.3982/ecta12701.

Centorrino, Samuele, and Jeffrey Scott Racine. 2017. "Semiparametric Varying Coefficient Models with Endogenous Covariates." *Annals of Economics and Statistics* (128):261-295.

Delgado, Michael S., Deniz Ozabaci, Yiguo Sun, and Subal C. Kumbhakar. 2019. "Smooth coefficient models with endogenous environmental variables." *Econometric Reviews*:1-23. doi: 10.1080/07474938.2018.1552413.

Hainmueller, Jens, Jonathan Mummolo, and Yiqing Xu. 2018. "How Much Should We Trust Estimates from Multiplicative Interaction Models? Simple Tools to Improve Empirical Practice." *Political Analysis* 27 (2):163-192. doi: 10.1017/pan.2018.46.

Hastie, Trevor, and Robert Tibshirani. 1990. *Generalized Additive Models*. New York: Chapman and Hall.

Hastie, Trevor, and Robert Tibshirani. 1993. "Varying-Coefficient Models." *Journal of the Royal Statistical Society. Series B (Methodological)* 55 (4):757-796.

Henderson, Daniel J., and Christopher F. Parmeter. 2015. *Applied Nonparametric Econometrics*. Cambridge, United Kingdom: Cambridge University Press.

Hirano, Keisuke, and Guido W. Imbens. 2004. "The Propensity Score with Continuous Treatments." *Applied Bayesian Modeling and Causal Inference from Incomplete‐Data Perspectives*:73-84. doi: doi:10.1002/0470090456.ch7.

Hoover, Donald R., John A. Rice, Colin O. Wu, and Li-Ping Yang. 1998. "Nonparametric smoothing estimates of time-varying coefficient models with longitudinal data." *Biometrika* 85 (4):809-822. doi: 10.1093/biomet/85.4.809.

Hoti, Fabian, and Lasse Holmström. 2003. "On the estimation error in binned local linear regression." *Journal of Nonparametric Statistics* 15 (4-5):625-642. doi: 10.1080/10485250310001605469.

Ichimura, Hidehiko, and Petra E. Todd. 2007. "Chapter 74 Implementing Nonparametric and Semiparametric Estimators." In *Handbook of Econometrics*, edited by James J. Heckman and Edward E. Leamer, 5369-5468. Elsevier.

Li, Qi, Cliff J. Huang, Dong Li, and Tsu-Tan Fu. 2002. "Semiparametric Smooth Coefficient Models." *Journal of Business & Economic Statistics* 20 (3):412-422.

Li, Qi, and Jeffrey Scott Racine. 2007. *Nonparametric Econometrics: Theory and Practice*. New Jersey: Princeton University Press.

Li, Qi, and Jeffrey Scott Racine. 2010. "Smooth Varying-Coefficient Estimation and Inference for Qualitative and Quantitative Data." *Econometric Theory* 26 (6):1607-1637.

Liu, Weiwei, and Kevin J. Egan. 2019. "A Semiparametric Smooth Coefficient Estimator for Recreation Demand." *Environmental and Resource Economics* 74 (3):1163-1187. doi: 10.1007/s10640-019-00362-7.

Long, J. Scott, and Laurie H. Ervin. 2000. "Using Heteroscedasticity Consistent Standard Errors in the Linear Regression Model." *The American Statistician* 54 (3):217-224. doi: 10.2307/2685594.

Polemis, Michael L., and Thanasis Stengos. 2015. "Does market structure affect labour productivity and wages? Evidence from a smooth coefficient semiparametric panel model." *Economics Letters* 137:182-186. doi: https://doi.org/10.1016/j.econlet.2015.11.004.

Rios-Avila, Fernando. 2019. "A Semi-Parametric Approach to the Oaxaca–Blinder Decomposition with Continuous Group Variable and Self-Selection." *Econometrics* 7 (2):28.

Seber, George A. F., and Alan J. Lee. 2003. *Linear Regression Analysis*. Second ed. New York: John Wiley & Sons.

Stinchcombe, Maxwell B., and David M. Drukker. 2013. "Regression Efficacy and the Curse of Dimensionality." In *Recent Advances and Future Directions in Causality, Prediction, and Specification Analysis: Essays in Honor of Halbert L. White Jr*, edited by Xiaohong Chen and Norman R. Swanson, 527-549. New York, NY: Springer New York.

Verardi, Vincenzo. 2013. Semiparametric regression in Stata. UK Stata Users Group meeting, London, UK.

Zhang, Wenyang, and Sik-Yum Lee. 2000. "Variable Bandwidth Selection in Varying-Coefficient Models." *Journal of Multivariate Analysis* 74 (1):116-134. doi: https://doi.org/10.1006/jmva.1999.1883.

**Appendix A. Kernel functions and Standardized kernel weights**

For the following definitions where is the evaluated point, is the point of reference and is the bandwidth.

|  |  |  |
| --- | --- | --- |
| **Kernel option** | **Kernel function** | **Standardize kernel weight** |
| gaussian |  |  |
| epan |  |  |
| epan2 |  |  |
| biweight |  |  |
| cosine |  |  |
| parzen |  |  |
| rectan |  |  |
| trian |  |  |

Note: Kernel option kernel() should be used as stated in this table when using all vc\_pack commands.

1. Stata 15 introduced the command npregress kernelwhich estimates fully non-parametric models using kernel methods. More recently, Stata 16 introduced npregress serieswhich estimates fully non-parametric models using series and spline methods. [↑](#footnote-ref-1)
2. A brief review of the semiparametric method is provided in Cameron and Trivedi (2005), Sec. 9.7. For a more in-depth revision of theory on semiparametric models, see Li and Racine (2007) Ch 7-11, whereas (Henderson and Parmeter 2015) offer a more empirical discussion on these types of models. [↑](#footnote-ref-2)
3. See Verardi (2013) for a brief review of commands for the estimation of semiparametric regressions in Stata. [↑](#footnote-ref-3)
4. In both cases, there are different strategies that can be used to select the roughness or smoothness of the estimated models. For a brief review of both strategies, see npregress intro. [↑](#footnote-ref-5)
5. Most methodologies implementing SVCM are based on the assumption that and are exogenous. Discussion on the estimation of SVCM models when is endogenous can be found in Cai et al. (2006), whereas the estimation of models when is endogenous has been discussed and proposed in Centorrino and Racine (2017), Delgado et al. (2019) and Rios-Avila (2019). This, however, is beyond the scope of this paper. [↑](#footnote-ref-6)
6. The estimator in equation 6 only exists if is full rank, but this may not be the case when using sample data. [↑](#footnote-ref-7)
7. One option could be to assume that the effects vary in a linear way with respect to fines. In such case, the following model could be fit using regress citations c.fines##i.(college taxes csize) [↑](#footnote-ref-8)
8. In the context of series, polynomials, and splines, the trade-off between variance and bias is determined by the dimension of the series transformations and the interactions . [↑](#footnote-ref-9)
9. Seber and Lee (2003), ch. 10, provides a simple demonstration of this identity for linear regression models. In addition, Hoover et al. (1998) suggest using a similar expression to speed up the calculation of the CV criteria. [↑](#footnote-ref-10)
10. Simulations provided in Hoti and Holmström (2003) suggests that the binned estimator accuracy, measured by the relative integrated squared error, is similar to the unbinned estimator when <0.3 for Gaussian kernels, and for the Epanechnikov, triangle and biweight kernels. [↑](#footnote-ref-11)
11. See appendix A for the full list of kernels and functions available for the estimation. [↑](#footnote-ref-12)
12. Stata uses this expression to define the number of bins used for a histogram as a default. [↑](#footnote-ref-13)
13. According to Li et al. (2002), the variance-covariance matrix for the SVCM can be consistently estimated by equation 20 if , with conditional heteroscedasticity of unknown form , and if is a standard second-order kernel function. Furthermore, it also assumes that as and . [↑](#footnote-ref-14)
14. It should be noticed that Cattaneo and Jansson (2018) does not explicitly analyze the validity of their findings in the framework of SVCM, but rather provides general conclusions for what he calls semiparametric kernel-based estimators. As a reference, npregress kernel reports the percentile confidence intervals as default, using a paired bootstrapped resampling procedure. [↑](#footnote-ref-15)
15. npregress kernel reports this statistic as “expected kernel observations”. [↑](#footnote-ref-16)
16. To prevent unexpected results, npregress kernel sets the maximum value as the sample size. [↑](#footnote-ref-17)
17. See Appendix A for a list of standardized kernel weight functions. [↑](#footnote-ref-18)
18. This statistic can also be extended to multivariable kernel regression models by simply using the standardize kernels across all independent variables. [↑](#footnote-ref-19)
19. There is no robust theoretical discussion in regards to using transformations of independent variables for the selection and estimation of non-parametric and semiparametric models. However, it is my conjecture that monotonic transformations can be used as an alternative to variable bandwidths, by allowing more information to be used in areas with low density, reducing the variance of the estimator. [↑](#footnote-ref-20)
20. See the do file that accompanies this paper to see where this comes from. [↑](#footnote-ref-21)
21. Results using bootstrapped procedure and percentile confidence intervals are provided in the accompanying dofile or upon request. [↑](#footnote-ref-22)