A Online technical appendix: Outline of nonparametric estimation and hypothesis testing.

A.1 Estimation of a single nonparametric term.

Consider a reduced specification of (2) that includes only the nonparametric term $s(z_i)$. Once this basic case is introduced, its extension to the full semiparametric model (2) will be trivial. In the reduced specification, the dependent variable y_i is explained by a single explanatory variable z_i (age or cohort) with a nonlinear effect on y_i:

$$y_i = s(z_i) + \epsilon_i \tag{8}$$

where $s(\cdot)$ is an arbitrary smooth function and ϵ_i is the error term with zero mean and variance σ^2.

Let $\kappa_1 < \cdots < \kappa_M$ be a sequence of breakpoints ('knots') that are distinct numbers that span the range of z_i. In the MGCV algorithm, the smooth function $s(z_i)$ is approximated by a sequence of cubic splines. In general, splines are piecewise polynomials that are joined at the 'knots'. Due to special restrictions, the cubic splines are continuous at the knots, and also have continuous first and second derivatives. Let M denote the number of knots. Then a cubic spline can be represented by truncated cubic basis functions:

$$s(z_i) = \delta_0 + \delta_1 z_i + \delta_2 z_i^2 + \delta_3 z_i^3 + \sum_{m=1}^{M} \delta_{m+3}(z_i - \kappa_m)_+^3 \tag{9}$$

where

$$(z_i - \kappa_m)_+ = \begin{cases} 0 & z_i \leq \kappa_m \\ z_i - \kappa_m & z_i > \kappa_m \end{cases}$$

The cubic spline has a simple interpretation of a *global* cubic polynomial $\delta_0 + \delta_1 z_i + \delta_2 z_i^2 + \delta_3 z_i^3$ and M *local* polynomial deviations $\sum_{m=1}^{M} \delta_{m+3}(z_i - \kappa_m)_+^3$. In matrix form, the truncated cubic basis becomes $\boldsymbol{y} = \boldsymbol{Z}\boldsymbol{\delta} + \boldsymbol{\epsilon}$, where $\boldsymbol{Z}$ is design matrix with ith row vector $\boldsymbol{Z}_i = \begin{bmatrix} 1 & z_i & z_i^2 & z_i^3 & (z_i - \kappa_1)_+^3 & \cdots & (z_i - \kappa_M)_+^3 \end{bmatrix}$, $\boldsymbol{\delta}$ is the corresponding vector of regression parameters, and ϵ is the error term. The smooth function $f(\boldsymbol{Z}, \boldsymbol{\delta})$ is linear in $M + 4$ regression parameters, and can be fitted by minimizing the sum of squared residuals: $(\boldsymbol{y} - \boldsymbol{Z}\boldsymbol{\delta})'(\boldsymbol{y} - \boldsymbol{Z}\boldsymbol{\delta}) = \|\boldsymbol{y} - \boldsymbol{Z}\boldsymbol{\delta}\|^2$, where $\| \cdots \|$ stands for the Euclidean norm.

By increasing the number of knots M, the model becomes more flexible in approximating y. But if the number of knots is too large, the estimates $\hat{s}(z)$ may follow y too closely. In the limit, when $M = n$, the cubic spline simply interpolates y. To prevent too much wiggliness in the estimated curve, a special term that penalizes rapid changes in $\hat{s}(z)$ is added to the fitting criteria. A common penalty is $\lambda \int [s_{zz}(z)]^2 \, dx$, which has a smoothing parameter λ and an integrated squared second derivative $s_{zz}(z)$ of $s(z)$. This results in the penalized least-squares criterion as follows:

$$Q(s,\lambda) = \|\boldsymbol{y} - \boldsymbol{Z\delta}\|^2 + \lambda \int [s_{zz}(z)]^2 \, dx.$$

If $\hat{s}(z)$ is too rough, this will increase the penalty term $\int [s_{zz}(z)]^2 \, dx$. The smoothing parameter λ controls the trade-off between the model fit $\|\boldsymbol{y} - \boldsymbol{Z\delta}\|$ and the roughness penalty $R = \int [s_{zz}(z)]^2 \, dx$. When $\lambda = 0$, the roughness penalty R has no effect on the minimization criterion $Q(f,\lambda)$, producing unpenalized estimates $\hat{s}(x)$ that just interpolate data. In contrast, when $\lambda = +\infty$, this results in the perfectly smooth line, *i.e.*, in a linear regression line with a constant slope.

The minimization of the penalized criterion $Q(s,\lambda)$ is simplified by noting that derivatives and integrals of $s(z)$ are linear transformations of parameters $d^m(z)$ in the cubic spline basis, with $s_{zz}(z) = \sum_{m=1}^{M} \delta_m d_{zz}^m(z)$ and $\int s(z)dz = \sum_{m=1}^{M} \delta_m \int d^m(z)dz$, where $d^m(z)$ denotes a particular form of basis function (such as the truncated cubic basis function in (9)). Thus, $s_{zz}(z) = \boldsymbol{d}_{zz}(z)'\boldsymbol{\delta}$, from which it follows that $[s_{zz}(z)]^2 = \boldsymbol{\delta}'\boldsymbol{d}_{zz}(z)'\boldsymbol{d}_{zz}(z)\boldsymbol{\delta} = \boldsymbol{\delta}'\boldsymbol{\Delta}(z)\boldsymbol{\delta}$. Finally,

$$R = \int [s_{zz}(z)]^2 dz = \boldsymbol{\delta}' \left(\int S(z)dz \right) \boldsymbol{\delta} = \boldsymbol{\delta}'\boldsymbol{\Delta}\boldsymbol{\delta}.$$

Thus, the roughness penalty R can be represented as a quadratic form in the parameter vector δ and matrix $\boldsymbol{\Delta}$ of known coefficients that are derived from the basis function $d^m(z)$.

Substituting the roughness penalty R with $\boldsymbol{\delta}'\boldsymbol{\Delta}\boldsymbol{\delta}$, the penalized least-squares criterion becomes

$$Q(s,\lambda) = \|\boldsymbol{y} - \boldsymbol{Z\delta}\|^2 + \lambda\boldsymbol{\delta}'\boldsymbol{\Delta}\boldsymbol{\delta}. \tag{10}$$

Differentiating $Q(f,\lambda)$ with respect to $\boldsymbol{\delta}$ and setting the derivative to zero produces an estimate of $\boldsymbol{\delta}$:

$$\hat{\boldsymbol{\delta}} = (\boldsymbol{Z}'\boldsymbol{Z} + \lambda\boldsymbol{\Delta})^{-1} \boldsymbol{Z}'\boldsymbol{y}. \tag{11}$$

The estimate of δ depends on the value of unknown smoothing parameter λ. The MGCV algorithm selects an appropriate value of λ by using the concept of hat matrix from the ordinary least-squares model. In the model, the hat matrix $\boldsymbol{H}$ projects the vector of dependent variable $\boldsymbol{y}$ into the vector of predicted values $\hat{\boldsymbol{y}} = \boldsymbol{Hy}$, with $\boldsymbol{H} = \boldsymbol{Z}(\boldsymbol{Z}'\boldsymbol{Z})^{-1}\boldsymbol{Z}'$. Using the estimate of $\hat{\delta}$ from (11), the hat matrix of the penalized spline model can be similarly defined as $\boldsymbol{H}_S = \boldsymbol{Z}(\boldsymbol{Z}'\boldsymbol{Z} + \lambda\boldsymbol{\Delta})^{-1}\boldsymbol{Z}'$. Since the matrix $\boldsymbol{H}_S$ transforms the vector of $\boldsymbol{y}$ into the vector of its smoothed values, the matrix $\boldsymbol{H}_S$ is often called a smoother matrix. In the MGCV algorithm, the optimal value of λ is found by minimizing the GCV criteria $V_g(\lambda)$ that depends on the sum of squared residuals $\|\boldsymbol{y} - \boldsymbol{Z}\hat{\boldsymbol{\delta}}\|^2$ and the trace of smoother matrix $\boldsymbol{H}_S$:

$$V_g(\lambda) = \frac{n\|\boldsymbol{y} - \boldsymbol{Z}\hat{\boldsymbol{\delta}}\|^2}{[n - \mathrm{tr}(\boldsymbol{H}_S)]^2} \tag{12}$$

where n is the number of observations, and $\mathrm{tr}(\boldsymbol{H}_S)$ is the trace of $\boldsymbol{H}_S$.

Though the MGCV algorithm selects an appropriate degree of smoothness with respect to parameter λ, this parameter is not informative in evaluating the estimated degree of smoothness. It is much easier to interpret the trace of the smoother matrix tr $(\boldsymbol{H}_S)$, since it is equal to the number of degrees of freedom, needed to approximate the smoothed function $f(z)$ (Ruppert *et al.*, 2003). Let $\nu = \operatorname{tr}(\boldsymbol{H}_S)$. Since the smoothing parameter λ is a part of $\boldsymbol{H}_S$, λ and ν are correlated. In particular, a small degree of smoothing is indicated by $\lambda \to 0$ and $\nu \to \infty$. Conversely, a high degree of smoothing corresponds to $\lambda \to \infty$ and $\nu \to 0$. An important special case is when $\nu \leq 1$. This range of ν indicates a parametric effect, when a single variable is sufficient to approximate the smoothed function $s(z)$.

The GCV criterion $V_g(\lambda)$ has one problem in selecting an optimal smoothness. Monte Carlo studies by Kim and Gu (2004) and Bacchini *et al.* (2007) demonstrated that $V_g(\lambda)$ may choose too small values of λ, which results in undersmoothing. The problem can be solved by multiplying tr $(\boldsymbol{H}_S)$ in (12) by a parameter $\eta > 1$ that increases the cost per trace of $\boldsymbol{H}_S$:

$$\bar{V}_g(\lambda) = \frac{n\|\boldsymbol{y} - \boldsymbol{Z}\hat{\boldsymbol{\delta}}\|^2}{[n - \eta \cdot \operatorname{tr}(\boldsymbol{H}_S)]^2} \, . \tag{13}$$

In estimating the smoothing cohort model, we followed the recommendation in Wood (2006) that a good value for η is 1.4. In practice, the modification had little effect on our estimates of age or cohort effects.

After specifying how the smooth function $s(x)$ is estimated by spline basis functions, the basic model (8) can be easily extended to the full semiparametric model (2) that adds the parametric part with cohort and year effects. For the smoothing age model, the parametric part $\boldsymbol{W}$ includes matrices of dummy variables D_t^Y, D_ℓ^C. After the extension, the truncated cubic basis (9) still has the form $\boldsymbol{y} = \tilde{\boldsymbol{Z}}\tilde{\boldsymbol{\delta}} + \epsilon$, but the basis $\tilde{\boldsymbol{Z}}$ now includes an expanded design matrix $\tilde{\boldsymbol{Z}} = [\boldsymbol{Z}, \boldsymbol{W}]$. The estimate of $\tilde{\boldsymbol{\delta}}$ is obtained from (11), where the smoothing parameter λ is found by minimizing either $V_g(\lambda)$ or $\bar{V}_g(\lambda)$.

A.2 Estimation of a joint effect of two smooth functions.

In this subsection, we describe how we estimated the joint effects of age and cohort of housing in Model 4 (specification (5)). While the effect of single nonparametric term z_i on y_i in 8 produces a smooth line that account a possible nonlinear relationship, the joint effect of two variables a_i (age) and c_i (cohort) on y_i is given by $y_i = s(a_i, c_i) + \epsilon_i$. The joint effect of a_i and c_i on y_i produces a smooth surface, in which the effect of a_i on y_i may be not only nonlinear, but also different at various levels of c_i.

In estimating the smooth effect of two covariates a_i and c_i on y_i, we used a tensor product smoother that was introduced in Wood (2006). The smoother is closely related to the univariate smoother that we described in subsection A.1. Essentially, the joint smoother of a_i and c_i is constructed from marginal bases and penalties of each of the covariates. Consider the construction of the joint basis function of $s(a, c)$. Let marginal smoothing terms for $s_a(a)$ and $s_c(c)$ be denoted by $s_a(a) = \sum_{q=1}^{M_q} \theta_q^a d^q(a)$ and $s_c(c) = \sum_{r=1}^{M_r} \theta_r^c d^r(c)$, where θ_q^a and θ_r^c are regression parameters (similar to the parameter δ in the univariate specification equation (9)), and $d^q(a)$ and $d^r(c)$ are basis functions for a and c. To proceed from $s_a(a)$ and $s_c(c)$ to $s(a, c)$, we first assume that θ_q^a in the basis function of $s_a(a)$ is a smooth function of c, with

$\theta^a_q(c) = \sum_{r=1}^{M_r} \delta_{qr} d^r(c)$. Then the joint basis for a and c becomes

$$s(a,c) = \sum_{q=1}^{M_q} \theta^a_q(c) d^q(a) = \sum_{q=1}^{M_q} \sum_{r=1}^{M_r} \delta_{qr} d^r(c) d^q(a) \tag{14}$$

In matrix form, the joint basis regression model is written by $\boldsymbol{y} = \boldsymbol{Z}(a,c)\boldsymbol{\delta} + \boldsymbol{\epsilon}$. Essentially, the joint basis function $\boldsymbol{Z}(a,c)$ is constructed as the Kronecker product of individual marginal smoothing bases of a and c, denoted $\boldsymbol{Z}_a$ and $\boldsymbol{Z}_c$. For example, for the univariate smooth term a, the individual smoothing base was defined by $\boldsymbol{Z}$, which we already discussed in subsection A.1.

The roughness penalty for the joint smoother is constructed similarly to the joint smoothing basis function $\boldsymbol{Z}$, by using marginal roughness penalties for a and c. For the univariate smooth of a, such a penalty was already defined by (10). To construct the composite penalty term, let $s_{a|c}(a)$ be a joint smooth of a and c with some fixed c. Then the roughness of $s_{a|c}$ is given by $R_a(s_{a|c})$. By integrating $R_a(s_{a|c})$ across different c, we obtain $R_a(s_a) = \int R_a(s_{a|c})dc$, which measures the total roughness of $s(a,c)$ in the direction of a.

The total roughness penalty in the direction of c is obtained similarly, by fixing a at some specific points, and integrating the total roughness penalty $R_c(s_c) = \int R_c(s_{c|a})da$ across different fixed values of a. A combined penalty for the joint effect of a and c is specified by

$$\lambda_a \int R_a(s_{a|c})dc + \lambda_c \int R_c(s_{c|a})da.$$

Assuming that $s_{a|c}(a) = \sum \theta^a_q(c) d^q(a)$, we could write $R_a(s_{a|c}) = \boldsymbol{\theta}^a(c)' \boldsymbol{\Delta}_a \boldsymbol{\theta}^a(c)$. A simple reparameterization can be used to provide an approximation to the terms in penalty: $\boldsymbol{\theta}^{a\prime} = \boldsymbol{\Gamma}\boldsymbol{\theta}^a$. Hence the penalty coefficient matrix becomes $\boldsymbol{\Delta}'_a = \boldsymbol{\Gamma}^{-1\prime} \boldsymbol{\Delta}_a \boldsymbol{\Gamma}^{-1}$. Then $R_a(s_a)$ and $R_c(s_c)$ are used to create composite roughness penalties $\bar{\boldsymbol{\Delta}}_a = \boldsymbol{\Delta}'_a \otimes \boldsymbol{I}_{M_r}$ and $\bar{\boldsymbol{\Delta}}_c = \boldsymbol{I}_{M_q} \otimes \boldsymbol{\Delta}'_c$, where $\boldsymbol{I}_{M_r}$ and $\boldsymbol{I}_{M_q}$ denote identity matrices, with M_q and M_r equal to the number of 'knots' in the direction of c and a, respectively.

Using the composite roughness penalties $\bar{\boldsymbol{\Delta}}_a$ and $\bar{\boldsymbol{\Delta}}_c$, the penalized least-squared criterion is constructed similarly to (10), by combining the least-squares term with roughness penalties in the direction of a and c, which are multiplied by the corresponding smoothing parameters λ_a and λ_c:

$$Q(s(a,c), \lambda_a, \lambda_c) = \|\boldsymbol{y} - \boldsymbol{Z}\boldsymbol{\delta}\|^2 + \lambda_a \boldsymbol{\delta}' \bar{\boldsymbol{\Delta}}_a \boldsymbol{\delta} + \lambda_c \boldsymbol{\delta}' \bar{\boldsymbol{\Delta}}_c \boldsymbol{\delta} \tag{15}$$

Specific details about the construction of the joint basis function $\boldsymbol{Z}(a,c)$ and the roughness penalty are provided in Wood (2006). Similarly to the univariate case, individual smoothing parameters λ_a and λ_c are selected by minimizing the GCV criterion, as defined in (13).

A.3 Hypothesis testing with bootstrap.

Since the GAM estimator does not belong to conventional linear regression models, hypothesis testing is complicated because the finite sample distribution of test statistics is not known. The problem can be solved by using a bootstrap testing procedure that resamples residuals from a GAM fit. Consider two models, called Model A and B. Let Model A satisfy the null hypothesis, and Model B satisfy the alternative hypothesis. Denote fitted values and residuals from estimating Model A as $\hat{y}^A$ and $\hat{u}^A$. Let the actual value of test statistic be $\hat{\phi}$. To estimate a p-value for the test statistic $\hat{\phi}$, we used the following bootstrap approach from MacKinnon (2007):

1. Specify the number of bootstrap replications O, and the significance level of the test.

2. For each $o = 1, \cdots, O$, resample regression residuals from $\hat{u}^A$, and denote the bootstrap sample as $\hat{u}^A_o$. Then calculate bootstrap values of y as $y^A_o = \hat{y}^A + \hat{u}^A_o$.

3. Using y^A_o and matrix of independent variables $\boldsymbol{x}$, estimate alternative model B, and calculate a bootstrap test statistic ϕ^*_o .

4. Repeat until the last bootstrap resampling of $\hat{u}^A$ that produces test statistic ϕ^*_O.

5. Estimate a bootstrap p-value for $\hat{\phi}$ by $\hat{p}^*(\hat{\phi}) = \frac{1}{O}\sum_{o=1}^{O} I\left(\phi^*_o > \hat{\phi}\right)$. Suppose that ϕ^*_o was larger than $\hat{\phi}$ at 35 times, and $O = 1000$. Then $\hat{p}^*(\hat{\phi}) = 35/1000 = 0.035$.

6. If $\hat{p}^*(\hat{\phi}) <$ significance level, reject the null hypothesis, and otherwise, accept it.