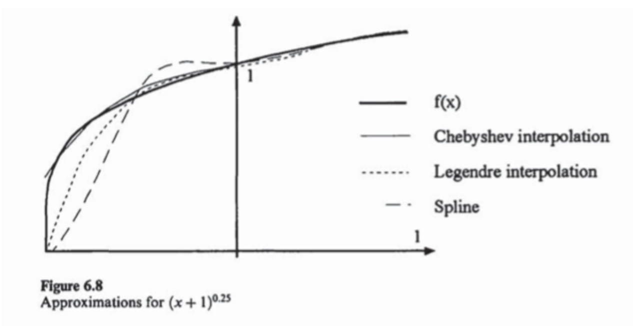
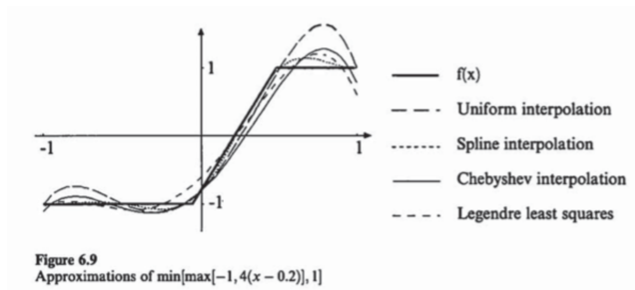
Materials 27 - Approximating functions with the help of Judd, *Numerical Methods*

Laura Gáti

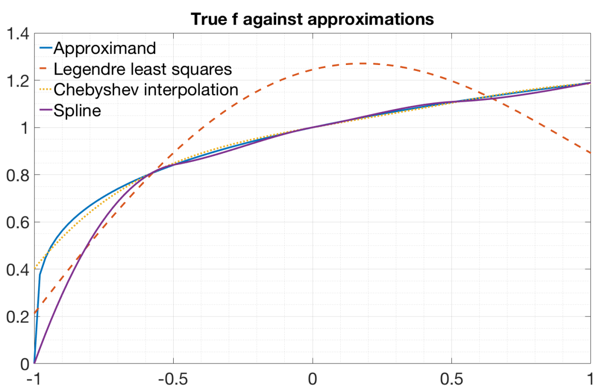
April 28, 2020



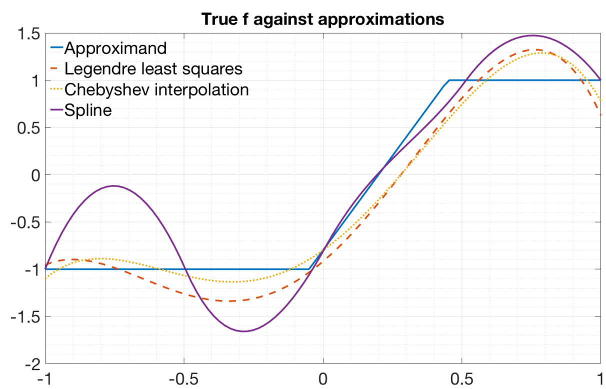
(a)



(b)



(c)



(d)

Details on approximating $f(x)$ on the interval $[a, b]$:

1. Legendre least squares

   (a) The interpolating polynomial is

$$p(x) = \sum_{k=0}^{n} \frac{\langle f, \varphi_k \rangle}{\langle \varphi_k, \varphi_k \rangle} \varphi_k(x) \tag{1}$$

$$\text{where} \quad \langle f, g \rangle \equiv \int_a^b f(x) g(x) \omega(x) \, dx \tag{2}$$

(b) My interpretation of this is that $p(x)$ is like the fitted value of regressing on an orthogonal polynomial of order $1, \ldots, n$, φ_k, with "OLS-coefficients" $\frac{\langle f, \varphi_k \rangle}{\langle \varphi_k, \varphi_k \rangle}$ for each regressor k.

(c) For φ, I use the Legendre polynomial.

(d) The only remaining issue is to compute the inner products. For that, I use Gauss-Chebyshev quadrature:

i. Calculate N quadrature nodes x_j , $j = 1 \ldots n$ as the zeros of the Chebyshev polynomial, and associated quadrature weights w_j as the Chebyshev weights.

ii. Approximate the inner products, e.g. $\langle f, \varphi_k \rangle$ as

$$\int_a^b f(x)\varphi_k(x)\omega^L(x)\, dx \approx \sum_{j=1}^{N} w_j f(x_j)\varphi_k(x_j)\omega^l(x_j) \tag{3}$$

where fortunately the Legendre weights $\omega^l(x) = 1\ \forall x$.

2. Chebyshev interpolation

(a) I'm not entirely sure if Chebyshev interpolation means just choosing the interpolation nodes as the Chebyshev zeros, or if it also means that you use the Chebyshev polynomials as a basis too (I think so...)

(b) What I'm doing is: the interpolating polynomial $\hat{f}$ is a degree n Chebyshev polynomial approximation if

$$\hat{f} = \sum_{j=0}^{n} a_j T_j(z(x)) \tag{4}$$

where T_j is a degree j Chebyshev polynomial, $z(x)$ are the points adapted to a general interval (this is not necessary here of course) and a_j are the Chebyshev coefficients computed on $m = n + 1$ nodes as:

$$a_0 = \frac{1}{m}\sum_{j=1}^{m} f(x_j) \tag{5}$$

$$a_j = \frac{2}{m}\sum_{j=1}^{m} f(x_j)T(z_j) \tag{6}$$

3. Spline

(a) Choose $n + 1$ evenly spaced nodes on $[a, b]$. Divide the interval into n subintervals.

(b) On each interval i, approximate the points in the interval as $s(x) = a_i + b_i x + c_i x^2 + d_i x^3$ (cubic spline).

(c) Set up and solve an equation system for the $4n$ coefficients $(a_i, b_i, c_i, d_i),\ i = 1, \ldots, n$. This consists of the following conditions:

   i. Interpolating conditions for all nodes ($n + 1$ conditions)

   ii. Continuity conditions at inner nodes ($n - 1$ conditions)

   iii. Twice differentiability at inner nodes ($2n - 2$ conditions)

   iv. The missing 2 conditions defining the derivative of $s(x)$ at the first and last node.

A Model summary

$$x_t = -\sigma i_t + \hat{\mathbb{E}}_t \sum_{T=t}^{\infty} \beta^{T-t}\big((1-\beta)x_{T+1} - \sigma(\beta i_{T+1} - \pi_{T+1}) + \sigma r_T^n\big) \tag{A.1}$$

$$\pi_t = \kappa x_t + \hat{\mathbb{E}}_t \sum_{T=t}^{\infty} (\alpha\beta)^{T-t}\big(\kappa\alpha\beta x_{T+1} + (1-\alpha)\beta\pi_{T+1} + u_T\big) \tag{A.2}$$

$$i_t = \psi_\pi \pi_t + \psi_x x_t + \bar{i}_t \qquad \text{(if imposed)} \tag{A.3}$$

$$\text{PLM:} \qquad \hat{\mathbb{E}}_t z_{t+h} = a_{t-1} + b h_x^{h-1} s_t \quad \forall h \geq 1 \qquad b = g_x \, h_x \tag{A.4}$$

$$\text{Updating:} \qquad a_t = a_{t-1} + k_t^{-1}\big(z_t - (a_{t-1} + b s_{t-1})\big) \tag{A.5}$$

$$\text{Anchoring function:} \qquad k_t = k_{t-1} + \mathbf{g}(fe_{t-1}^2) \tag{A.6}$$

$$\text{Forecast error:} \qquad fe_{t-1} = z_t - (a_{t-1} + b s_{t-1}) \tag{A.7}$$

$$\text{LH expectations:} \qquad f_a(t) = \frac{1}{1-\alpha\beta} a_{t-1} + b(\mathbb{I}_{nx} - \alpha\beta h)^{-1} s_t \qquad f_b(t) = \frac{1}{1-\beta} a_{t-1} + b(\mathbb{I}_{nx} - \beta h)^{-1} s_t \tag{A.8}$$

This notation captures vector learning (z learned) for intercept only. For scalar learning, $a_t = \begin{pmatrix} \bar{\pi}_t & 0 & 0 \end{pmatrix}'$ and b_1 designates the first row of b. The observables (π, x) are determined as:

$$x_t = -\sigma i_t + \begin{bmatrix} \sigma & 1-\beta & -\sigma\beta \end{bmatrix} f_b + \sigma \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} (\mathbb{I}_{nx} - \beta h_x)^{-1} s_t \tag{A.9}$$

$$\pi_t = \kappa x_t + \begin{bmatrix} (1-\alpha)\beta & \kappa\alpha\beta & 0 \end{bmatrix} f_a + \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} (\mathbb{I}_{nx} - \alpha\beta h_x)^{-1} s_t \tag{A.10}$$

B Target criterion

The target criterion in the simplified model (scalar learning of inflation intercept only, $k_t^{-1} = \mathbf{g}(fe_{t-1})$):

$$\pi_t = -\frac{\lambda_x}{\kappa}\Bigg\{ x_t - \frac{(1-\alpha)\beta}{1-\alpha\beta}\Big(k_t^{-1} + ((\pi_t - \bar{\pi}_{t-1} - b_1 s_{t-1}))\mathbf{g}_\pi(t)\Big) \\ \Big(\mathbb{E}_t \sum_{i=1}^{\infty} x_{t+i} \prod_{j=0}^{i-1}(1 - k_{t+1+j}^{-1} - (\pi_{t+1+j} - \bar{\pi}_{t+j} - b_1 s_{t+j})\mathbf{g}_{\bar{\pi}}(t+j))\Big)\Bigg\} \tag{B.1}$$

where I'm using the notation that $\prod_{j=0}^{0} \equiv 1$. For interpretation purposes, let me rewrite this as follows:

$$\pi_t = -\frac{\lambda_x}{\kappa} x_t + \frac{\lambda_x}{\kappa}\frac{(1-\alpha)\beta}{1-\alpha\beta}\Big(k_t^{-1} + fe_{t|t-1}^{eve}\mathbf{g}_\pi(t)\Big)\mathbb{E}_t \sum_{i=1}^{\infty} x_{t+i} \\ -\frac{\lambda_x}{\kappa}\frac{(1-\alpha)\beta}{1-\alpha\beta}\Big(k_t^{-1} + fe_{t|t-1}^{eve}\mathbf{g}_\pi(t)\Big)\Big(\mathbb{E}_t \sum_{i=1}^{\infty} x_{t+i} \prod_{j=0}^{i-1}(k_{t+1+j}^{-1} + fe_{t+1+j|t+j}^{eve})\mathbf{g}_{\bar{\pi}}(t+j)\Big) \tag{B.2}$$

Interpretation: tradeoffs from discretion in RE + effect of current level and change of the gain on future tradeoffs + effect of future expected levels and changes of the gain on future tradeoffs