Problem Set 6: Function Approximation

Jacob Adenbaum and Albert Rodriguez-Sala
Programming and Numerical Methods for Economics – ECNM10115
The University of Edinburgh

Spring 2024

Due Monday, March 11, 2024

Instructions: Attempt all the exercises. Partial credit will be awarded for incomplete solutions which demonstrate some understanding. You should always try to explain your results, and provide as much intuition as possible. Pay careful attention to how you present your results. Be sure to take some time to make them easy to read and to understand. Avoid printing out irrelevant output, and pay attention to how you construct your figures and tables. You will be assessed both on the correctness of your results, and on how easy you make it to understand them.

In particular, this problem set asks you to generate a lot of graphs. You should be careful in how you present them, and choose a visual presentation that makes it easy to understand the point you're trying to make. It is *not* sufficient simply to print everything out and expect your reader to sort through it all.

Problem 1. Consider the following five functions

$$\begin{aligned}
f_1(x) &= \exp(2x+2) && x \in [-1,1] \\
f_2(x) &= \min\left(\max\left[-1, 4\left(x - \frac{1}{5}\right)\right], 1\right) && x \in [-1,1] \\
f_3(x) &= -2x^{-\frac{1}{2}} && x \in [0,1] \\
f_4(x) &= \frac{1}{1+x^2} && x \in [-5,5]
\end{aligned} \tag{1}$$

(a) Approximate each of these functions with Chebyshev polynomials, Lagrange polynomials, linear interpolation, and cubic splines[1]. Try a variety of grid sizes ($n = 5, 10, 20, 50, 100$) and plot the resulting approximations against the true function. Plot the approximation error $f - \hat{f}$ over the relevant domains. How well do the methods seem to be working?

Note: you will need to write your own implementation of Chebyshev polynomials and Lagrange interpolation to do this, however you should feel free to use a standard implementation of linear interpolation/cubic splines.

(b) Let's try to study the approximation errors more rigorously. Choose your favorite way to calculate the approximation error[2] $||f - \hat{f}||^2$. For each function f_i, calculate the approximation ϵ_n with each of the four methods you used in part (a) for $n = 5, 6, 7, 8, \ldots, 1000$. Save the results in an array, and plot them

[1] Be careful to choose an appropriate boundary condition, like we discussed in class

[2] Remember that if $f : [a,b] \to \mathbb{R}$, then

$$||f - \hat{f}||^2 = \int_a^b \left(f(x) - \hat{f}(x)\right)^2 dx$$

You will need to approximate this, either using your preferred numerical integration technique, or by taking an average. For instance, try generating a really fine grid over $[a,b]$ (say, with 1000 points). Evaluate f and $\hat{f}$ at these test points, and take the average of the squared difference between the true function and your approximation:

$$||f - \hat{f}||^2 \approx \frac{1}{1000} \sum_{i=1}^{1000} (f(x_i) - \hat{f}(x_i))^2$$

against n. You will want to use a log-scale (i.e., $\log(n)$ on the x-axis, and $\log(\epsilon_n)$ on the y-axis). This will give you four plots, with four lines each. Interpret your results. What do the slopes tell you?

Hint: Think about the rates of convergence for these methods.

Problem 2. Consider the function $v(k) = a + b\log(k)$ for some parameters a and b, and the corresponding maximization problem

$$\begin{aligned} &\max_{c,k'} \ \log(c) + \beta v(k') \\ &\text{s.t. } c + k' = k^\alpha \\ &\quad\ 0 \le c \le k^\alpha \end{aligned} \tag{2}$$

where α and β are also parameters.

(a) Start with $a = -10$, $b = 0.5$, $\alpha = 1/2$ and $\beta = 0.9$. Write code that solves eq. (2) for any choice of k.[3] Try it out for $k = 0.5, 1, 5$, and 9, and find a sensible way to visualize the results.[4]

(b) Now, suppose you cannot use v directly in solving eq. (2) (for instance, imagine that it is costly to evaluate). Instead, you must use an approximation $\hat{v}$. Approximate v using linear interpolation, cubic splines, and Chebyshev polynomials, for $n = 5, 20$ and 100.[5] Re-solve eq. (2) using $\hat{v}$ instead of v. Compare (and visualize) your results. How similar are the optimized values? How close are the implied optimal choices of c and k'? Which approximation algorithm generates the best results (and why)? What do you take away from this?

(c) With the parameters you used in part (a), check whether or not this problem has the property that

$$\begin{aligned} v(k) = \ &\max_{c,k'} \ \log(c) + \beta v(k') \\ &\text{s.t. } c + k' = k^\alpha \\ &\quad\ 0 \le c \le k^\alpha \end{aligned} \tag{3}$$

You should verify this on a grid of points. Check it for every k on an evenly spaced grid of 100 points from 0.1 to 10. Plot the difference between your max values and the v you started with.

(d) Start from our initial guess $a_0 = -10$ and $b_0 = 0.5$, and try the following procedure:

1. For a given guess of (a_i, b_i), solve eq. (2) for several values of k (call them k_i) and store the maximum value that is attained at each point in a vector (call the values v_i). For this exercise, use an evenly spaced grid of 5 points from 0.1 to 10.
2. Find the values of a and b that solve the problem:[6]

$$\min_{a,b} \sum_{i=1}^{5} (v_i - a - b\log(k_i))^2 \tag{4}$$

3. Set (a_{i+1}, b_{i+1}) equal to the results from Step 2. If (a_{i+1}, b_{i+1}) is close enough (let's say, within a tolerance of 10^{-8}) then stop. Otherwise, go back to Step 1 and repeat.

Report your resulting a and b, and plot the results. Try answering (c) again with the new a and b. What do you find? Interpret your results.

[3] Note that you will want to substitute the budget constraint into the objective so that you have an unconstrained problem.

[4] For instance, you could make a plot of the objective function for each value of k, where you include a vertical line at the minimum you found.

[5] Note: You will need to pick a domain for your approximation. I'd recommend using $[10^{-6}, 10]$. What happens if you use 0 as the lower bound of the domain?

[6] If you look closely, you can see that this is in fact a linear regression problem, as we are choosing parameters that minimize the sum of squared residual errors in the equation

$$v_i = a + b\log(k_i) + \epsilon_i$$

Here, we are approximating our function with a linear regression!