ENGR 695 Advanced Topics in Engineering Mathematics Fall 2024

Lab #3: Curve Fitting Using the Constrained Least Squares Method

Introduction

In this lab exercise, you will explore the application of the constrained least squares method with a modified normal equation to a data modeling problem. You will also explore different ways of "smoothing" the coefficients computed by the normal equation.

Before beginning, download the *Matlab* script `Lab3start.m`, which is available at the course Moodle site in the "Lab Materials" section. You should set up a separate folder on your own computer and/or in your Bucknell private Netspace for your ENGR 695 lab activities.

You might also want to locate and keep handy the supplemental reading "Constrained Least-Squares Optimization Using Minimized Coefficient Magnitudes," which is available at the course Moodle site, and the last page of the Lab #1 handout entitled "Important Matlab Commands for Linear Algebra." Both should be very helpful resources for this lab exercise.

Background

Curve fitting problems usually involve an attempt to represent a set of M data points $y(x_i)$, where $i = 1$ to M, by the approximation $\hat{y}(x_i)$, which is defined by

$$y(x_i) \approx \hat{y}(x_i) = \sum_{j=1}^{N} c_j f_j(x_i),$$

where the basis functions $\{f_j(x)\}_{j = 1 \text{ to } N}$ are relatively simple elementary functions and the coefficients $\{c_j\}_{j = 1 \text{ to } N}$ are a set of constant weights. The normal equation

$$\mathbf{c} = \left(F^T F\right)^{-1} F^T \mathbf{y},$$

where the superscript T indicates the matrix transpose operation, can be used to find a set of coefficients that constitute a best fit. For many problems, the coefficients calculated using the basic normal equation are very large in magnitude and oscillate between positive and negative values. This can happen when the normal matrix $F^T F$ is ill-conditioned, that is, nearly singular.

As explained in the supplemental reading "Constrained Least-Squares Optimization Using Minimized Coefficient Magnitudes," the coefficients can be smoothed by applying some type of constraint. In many cases, the result is a modified normal equation of the form

$$\left(F^T F + \gamma H\right)\mathbf{c} = F^T \mathbf{y} \quad \text{or} \quad \mathbf{c} = \left(F^T F + \gamma H\right)^{-1} F^T \mathbf{y},$$

where H is a simple, symmetric, and nearly diagonal matrix. The Lagrange multiplier γ is an empirically determined constant. One of the simplest constraints involves minimizing the squared magnitude of the coefficient vector $\mathbf{c}$. In that case, the matrix H is the identity matrix.

Procedure

Start *Matlab*, and change the current folder to the one in which you saved the file `Lab3start.m`. Then open `Lab3start.m` in the *Matlab* script editor using the "Open" menu item in the ribbon at the top of the main *Matlab* window. You will be able to view and edit the file there.

The first 80 or so lines of `Lab3start.m` define the parameter values for the curve-fitting problem and the data set to be approximated. It also provides extensive comments to guide you through the logical flow of the script.

The next 30 or so lines contain four sections identified by boldface comments where you are to insert new lines of code to find the required coefficients using three possible methods:

1. Unconstrained LS optimization
2. Constrained LS optimization with suppressed coefficient magnitudes
3. Constrained LS optimization with suppressed second finite differences (explained later)
4. Constrained LS optimization with suppressed first finite differences (explained later)

The remaining sections of the script generate three plots to help you visualize the coefficients, the quality of the fit to the data, and the basis functions.

Take some time to familiarize yourself with the script `Lab3start.m` and how it works, and then complete the following steps:

1. Find the text 'Your Name Here' in the code following the line `figure(2)` near the end of the script, and change the text to your name. This will cause your name to appear in one of the plots.
2. Make sure that the first data set (second column of the `data` matrix) is selected. This is determined around line 56 with the `y = y1` command.
3. Add *Matlab* code to the unconstrained LS section and the first constrained LS section (in which $|\mathbf{c}|^2$ is minimized) to calculate the coefficients and the condition number of the normal matrix (F^TF for the unconstrained case). The third and fourth sections (in which the second and first differences are minimized) should remain commented out for now. Run the script initially with $\gamma = 10^{-15}$ for the constrained problem to check your code. This value of γ is so small that the constrained problem is effectively reduced to the unconstrained problem. You should obtain the same set of coefficients and the same condition number for both methods. The condition numbers will appear in the header information above the plot of the coefficients. You should notice that the coefficients have enormous magnitudes and are oscillating between positive and negative values. Besides being displayed in one of the plots, the two sets of coefficients are also listed in the *Matlab* command window.
4. Increase the value of γ until the coefficients no longer oscillate significantly but instead vary smoothly and the curve passes through the data points in a way that seems correct. The coefficient values should be the same order of magnitude as the y data. You will have to change γ by orders of magnitude, at least initially, until you find the "right" value. This is a bit of a judgment call; a wide range of γ values will work. Choose the lowest order of magnitude for γ that seems right to you.

5. Save a copy of the plot entitled "Original Curve and Approximations," which should now have your name on the second line, and import it into your favorite word-processing software. For Microsoft *Word*, the *.tif or *.png formats generally work well. Add your name, the text "ENGR 695," and the lab number to the top of the document. Under the plot, add the condition numbers of the normal matrices for the unconstrained and constrained cases, and briefly explain why you chose your particular value of γ and the significance of the condition numbers.
6. Now increase γ by orders of magnitude above your chosen value and notice what happens to the coefficient values and the quality of the curve fit as γ increases. In your document, describe what happens and try to explain why.
7. Keep the code in the unconstrained LS section of the *Matlab* script active, but comment out the section associated with minimizing $|\mathbf{c}|^2$ and uncomment the following section (in which the second finite differences are minimized). Code has been provided to you to calculate the required matrix H, but you should add code to calculate the coefficients and the condition number of the normal matrix for the constrained case. The second finite difference is defined as

   $$D^2 c_k = \left(c_{k+1} - c_k\right) - \left(c_k - c_{k-1}\right) = c_{k-1} - 2c_k + c_{k+1},$$

   where D^2 is the difference operator (analogous to $\partial^2/\partial x^2$ for continuous functions). It is the discrete math analog of the second derivative and represents the "slope of the slope." If you have ever studied finite differences (and you will in a few weeks!), you should recognize the expression above. As explained in the "Background" section, this constraint is implemented using the modified normal equation

   $$\left(F^T F + \gamma H\right)\mathbf{c} = F^T \mathbf{y} \quad \rightarrow \quad \mathbf{c} = \left(F^T F + \gamma H\right)^{-1} F^T \mathbf{y},$$

   where H has a specific form. We will discuss later how the matrix H is determined, but for now you may use the code for generating H that has been provided to you in the script.
8. Finish modifying the code in this section, and then run the script initially with $\gamma = 10^{-15}$ for the constrained problem to check your code. You should get the same set of coefficients and condition number for both methods (unconstrained and constrained LS). The condition numbers will appear in the header information above the coefficients plot. Once again, both sets of coefficients should oscillate wildly as long as γ is tiny.
9. Increase the value of γ until the constrained LS coefficients vary smoothly and are the same order of magnitude as the y data. This will again involve a judgment call since a wide range of γ values will work. Choose the lowest order of magnitude for γ that seems right to you.
10. Save a copy of the plot entitled "Original Curve and Approximations" (with your name on the second line), and import it into your report document. Under the plot, add the condition numbers of the normal matrices for the unconstrained and constrained cases, and briefly explain why you chose your particular value of γ and the significance of the condition numbers.

11. Again increase γ by orders of magnitude above your chosen value and notice what happens to the coefficient values and the quality of the curve fit as γ increases. In your document, describe what happens and try to explain why. *Hint*: This constraint is trying to minimize the squared magnitude of the second finite difference (analogous to the second derivative) of the coefficients. See the definition of the second difference in Step #7 above.
12. Repeat steps 7 through 11 above, but this time for the first finite difference section of the code. The first finite difference is defined as

$$Dc_k = c_k - c_{k-1},$$

where D is the difference operator (analogous to $\partial/\partial x$ for continuous functions).
13. (optional) Examine what the K and H matrices look like for the first and second finite difference sections.

Assistance will be provided as needed, but try to deduce on your own how to complete as much of the work as possible.

After you have completed the lab activities, e-mail to me the following files:

1. Your modified *Matlab* script (m-file) with the file name `LName_Lab3_fa24.m`, where `LName` is your last name (surname).
2. The document file converted to PDF format (named `LName_Lab4_fa24.pdf`) that contains the saved plots, the condition numbers, and your associated comments.

<u>Lab Scoring and Submission Deadline</u>

Your score will be based primarily on the *Matlab* script and the document with figures that you submit. Scoring will be guided by the rubric posted on the Laboratory page at the course web site.

If you do not complete the exercises during the lab session, you may submit your documentation as late as 11:59 pm on Friday, September 27. If the files are submitted after the deadline, a 10% score deduction will be applied for every 24 hours or portion thereof that the item is late (not including weekend days) unless extenuating circumstances apply. No credit will be given five or more days after the deadline.