ENGR 695 Advanced Topics in Engineering Mathematics Fall 2024

Lab #4: Curve Fitting Using Singular Value Decomposition

Introduction

In previous lab exercises we applied the normal equation to the problem of finding a set of coefficients to approximate a data set using a weighted sum of Gaussian functions. The approximation can be expressed as

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

where the actual data are represented as $y(x_i)$ and the approximation as $\hat{y}(x_i)$. The basis functions are represented by $\{f_j(x)\}_{j\,=\,1\text{ to }N}$. The coefficients were found by applying the normal equation

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

which implements the basic least squares (LS) optimization method. Later, we applied the constrained LS method that uses the modified normal equation

$$\mathbf{c} = \left(F^T F + \gamma I_N\right)^{-1} F^T \mathbf{y}.$$

The unmodified normal equation sometimes produces coefficients with magnitudes that greatly exceed the data magnitudes and are highly oscillatory. The coefficients can be smoothed using the constrained LS method. However, the Lagrange multiplier γ must be determined via trial and error. Moreover, the method does not provide a useful measure of the conditioning of the problem (i.e., whether the F matrix is ill conditioned).

In this lab exercise, we will see that similar smoothing can be achieved via the singular value decomposition (SVD) method. A type of thresholding can be applied that has an effect much like using the Lagrange multiplier in the constrained LS method. We will also be able to determine the condition number from the SVD results.

To begin, download the *Matlab* script `Lab4start.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 should also locate and keep handy the last page of the Lab #1 handout entitled "Important *Matlab* Commands for Linear Algebra."

Background

The SVD method decomposes a matrix F as

$$F = U \Sigma V^T,$$

where U is an $M \times M$ orthogonal matrix, Σ (sometimes labeled S) is an $M \times N$ diagonal matrix, and V is an $N \times N$ orthogonal matrix. The matrices have the structures depicted below:

$$U = \begin{bmatrix} \uparrow & \uparrow & & \uparrow \\ \mathbf{u}_1 & \mathbf{u}_2 & \cdots & \mathbf{u}_M \\ \downarrow & \downarrow & & \downarrow \end{bmatrix}_{M \times M} \qquad \Sigma = \begin{bmatrix} \sigma_1 & 0 & \cdots & 0 \\ 0 & \sigma_2 & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \cdots & 0 & \sigma_N \\ 0 & 0 & \cdots & 0 \\ 0 & 0 & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \cdots & 0 & 0 \end{bmatrix}_{M \times N} \qquad V = \begin{bmatrix} \uparrow & \uparrow & & \uparrow \\ \mathbf{v}_1 & \mathbf{v}_2 & \cdots & \mathbf{v}_N \\ \downarrow & \downarrow & & \downarrow \end{bmatrix}_{N \times N}$$

The quantities $\mathbf{u}_1$, $\mathbf{u}_2$, etc. are the orthogonal column vectors of length M that make up the matrix U. Thus, $\mathbf{u}_i^T\mathbf{u}_j = \delta_{ij}$, where δ_{ij} is the Kronecker delta (equal to 1 if $i = j$ and 0 if not). The quantities $\mathbf{v}_1$, $\mathbf{v}_2$, etc. are also orthogonal column vectors but of length N that make up the matrix V, so $\mathbf{v}_i^T\mathbf{v}_j = \delta_{ij}$. The diagonal elements in the Σ matrix are called the *singular values* of the matrix F. Their relative sizes give a good indication of the conditioning of F, that is, whether it is a well-conditioned, ill-conditioned, or singular matrix.

Since parts of U and Σ are not actually necessary for matrix calculations in overdetermined systems systems (where $M > N$), the "economy" decomposition is often used to minimize the required computer memory. The matrices in the economy decomposition have the structures depicted below:

$$U = \begin{bmatrix} \uparrow & \uparrow & & \uparrow \\ \mathbf{u}_1 & \mathbf{u}_2 & \cdots & \mathbf{u}_N \\ \downarrow & \downarrow & & \downarrow \end{bmatrix}_{M \times N} \qquad \Sigma = \begin{bmatrix} \sigma_1 & 0 & \cdots & 0 \\ 0 & \sigma_2 & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \cdots & 0 & \sigma_N \end{bmatrix}_{N \times N} \qquad V = \begin{bmatrix} \uparrow & \uparrow & & \uparrow \\ \mathbf{v}_1 & \mathbf{v}_2 & \cdots & \mathbf{v}_N \\ \downarrow & \downarrow & & \downarrow \end{bmatrix}_{N \times N} .$$

If the SVD is applied to the F matrix in an overdetermined curve-fitting problem, then the coefficients can be found via

$$F\mathbf{c} = \mathbf{y} \quad \rightarrow \quad U\Sigma V^T\mathbf{c} = \mathbf{y} \quad \rightarrow \quad \mathbf{c} = V\Sigma^{-1}U^T\mathbf{y},$$

which makes use of the fact that the matrices U and V are orthogonal, so their inverses are equal to their transposes. It can be shown that calculating the coefficients in this way minimizes the approximation error (cost function) $|F\mathbf{c} - \mathbf{y}|^2$ for overdetermined systems in the least squares sense. It therefore yields the same result as the unmodified normal equation. Unfortunately, that means that the coefficient values can exhibit the same issues as those obtained using the normal equation, namely, excessively large magnitudes and severe oscillation. The problem (and a solution) might be made more obvious by expressing the matrix expression for $\mathbf{c}$ above in the equivalent form

$$\mathbf{c} = \sum_{j=1}^{N} \left(\frac{\mathbf{u}_j^T\mathbf{y}}{\sigma_j} \right) \mathbf{v}_j .$$

where, as explained earlier, $\mathbf{u}_j$ and $\mathbf{v}_j$ are the j^{th} orthogonal column vectors of U and V, respectively. Note that $\mathbf{u}_j^T\mathbf{y}$ is the dot product of $\mathbf{u}_j$ and $\mathbf{y}$. Ill conditioning can be thought of as the case in which one or more of the vectors $\mathbf{u}_j$ is nearly orthogonal to the data vector $\mathbf{y}$. If true, then those particular vectors do not contribute much to fitting the data. This would not be much of a problem if it weren't for the small associated singular value. A small dot product $\mathbf{u}_j^T\mathbf{y}$ by itself would suppress the troublesome term; that is, it would scale the associated $\mathbf{v}_j$ vector by a small value. However, because $\mathbf{u}_j^T\mathbf{y}$ is divided by the tiny singular value σ_j, the quantity in parentheses becomes large and the error is magnified.

This adverse state of affairs can be addressed by modifying the inverse of the singular value matrix. Because Σ is a diagonal matrix, its inverse (in economy form) is given by

$$\Sigma^{-1} = \begin{bmatrix} 1/\sigma_1 & 0 & \cdots & 0 \\ 0 & 1/\sigma_2 & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \cdots & 0 & 1/\sigma_N \end{bmatrix}_{N\times N} .$$

If one or more of the singular values is "too small" (to be qualified later), then the corresponding entry in the inverse matrix is simply set to zero. Doing so effectively sets $\mathbf{u}_j^T\mathbf{y}/\sigma_j = 0$ in the summation above, which eliminates the term that poorly fits the data and contributes to the ballooning of the coefficient values. Finding an appropriate singular value threshold that separates the "useful" terms from the problematic ones is a little involved, but the process is less ambiguous than the one for the Lagrange multiplier in the constrained LS method.

Procedure

The *Matlab* script `Lab4start.m` is very similar to the one used in the previous lab exercises. The first 85 lines set up the curve-fitting problem for the same set of data used before. The next section of code is mostly blank; it is where you will need to provide code to implement the SVD solution. The remaining lines generate helpful plots. Extensive comments guide you through the logical flow of the script.

Take some time to familiarize yourself with the script `Lab4start.m` 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 58 with the `y = y1` command.
3. Add code to the blank section indicated by the comment line "*** SOLUTION USING SINGULAR VALUE DECOMPOSITION (SVD)" to calculate the coefficients using the *Matlab* `svd` command, and use some of the results of the command to determine the condition number of the F matrix. (Do not use the *Matlab* `cond` command.) Store the calculated coefficients in the variable `cSVD` and the condition number of F in the variable `condSVD`.

4. Run the script initially without modifying the matrix Σ^{-1} to check your code. As explained in the "Background" section above, you should obtain the same set of coefficients as for the unconstrained LS method. The condition numbers for the $F^T F$ matrix (the normal matrix) in the LS solution and for F alone in the SVD solution should appear in the header information above the plot of the coefficients. The condition number for $F^T F$ should be much worse than the one for F alone. As we have seen already, the coefficients have enormous magnitudes, and they oscillate between positive and negative values. Each set of coefficients has its own *y*-axis; the one for the SVD coefficients is on the right side of the plot. The two sets of coefficients are listed in the *Matlab* command window in addition to being displayed in one of the plots.
5. Display the singular value matrix (type '`S`' at the command prompt), and examine the relative sizes of the singular values on the main diagonal. The smaller values might be represented as zero even if that is not their actual value. For those singular values, you might have to display them independently using a command such as `S(10,10)`.
6. Now add code to the SVD section that sets the diagonal entries of Σ^{-1} to zero if their corresponding singular values are sufficiently smaller than σ_1, the largest singular value, in a relative sense. The threshold should be defined as a factor that multiplies σ_1, (e.g., $10^{-8}\sigma_1$). Remember that the singular values are arranged from largest to smallest along the diagonal. As the cut-off threshold increases (to a point), the calculated coefficients (the **c** vector) should decrease in magnitude and reduce their tendency to oscillate while still maintaining a good fit to the data. There are guidelines for setting a threshold, but the theory is a little involved. For now, use trial and error to find the threshold that seems to produce "reasonable" coefficient values and a good fit.
7. Save a copy of the plot entitled "Lab #4: 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 ($F^T F$) matrix for the unconstrained LS case and of F for the SVD case. Also add some brief comments explaining why you chose your particular threshold for eliminating problematic singular values and the value (relative to the first singular value σ_1) of the threshold factor that you used. Convert the file to PDF format and name it `LName_Lab4_fa24.pdf`, where `LName` is your last name.

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_Lab4_fa24.m`, where `LName` is your last name (surname).
2. The document (named `LName_Lab4_fa24.pdf`) that contains the saved plot, the associated condition numbers, and your comments explaining why you chose your threshold value for eliminating problematic singular values.

Lab Scoring and Submission Deadline

Your score will be based primarily on the *Matlab* script and the document with figures that you submit according to 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, October 4. 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.