Notes on Error Propagation in Linear Systems

CS3220 Summer 2008 - Jonathan Kaldor

Up to this point, we have talked about solving $n \times n$ linear systems $\mathbf{Ax} = \mathbf{b}$ where $\mathbf{A}$ is of full rank. We have talked about, in passing, "difficult" systems to solve and made oblique mentions of matrices that are "nearly singular", but to this point we have assumed that as long as our matrix is not strictly singular, we can compute the answer by applying the LU factorization. Although this is true in a mathematical sense, what we will look at today are linear systems that, although technically nonsingular and thus invertible, have solutions that are highly dependent on small changes in the values of the known components (i.e. the entries of $\mathbf{A}$ and $\mathbf{b}$). The technical term used to describe this is called "conditioning" - we say that a problem is *well-conditioned* when the answer does not change much for small perturbations in the inputs, and correspondingly a problem is *ill-conditioned* when small changes in the inputs produce large changes in the answer.

Before we get into the details, why do we need to consider error? To begin with, oftentimes our inputs come from experimental measurements, and there may be sources of error or inaccuracies; for instance, we may only be confident in our inputs to 3 or 4 significant digits. Beyond that, the introduction of floating point numbers introduces additional, albeit relatively small, errors. Although the propagation of error due to floating point arithmetic is an interesting topic in its own right, for the most part we will be considering errors in the input numbers only.

Lets look at an example to make this concrete. Take the linear system

$$\begin{bmatrix} 1 & 1 \\ 1 & 1.0001 \end{bmatrix} \mathbf{x} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

We can see that the answer to this system is $\mathbf{x} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$. Suppose that our right hand side is perturbed slightly, so instead we have $\mathbf{b}' = \begin{bmatrix} 1.0001 \\ 0.9999 \end{bmatrix}$. Our solution then becomes $\mathbf{x}' = \begin{bmatrix} 3.0001 \\ -2 \end{bmatrix}$. Introducing an error of magnitude approximately 0.00014 in our inputs has resulted in a change of magnitude 2 to 3 in each of the components of the computed answer. Even if we compute the introduced error in a relative sense - $\frac{\|\mathbf{x}' - \mathbf{x}\|_2}{\|\mathbf{x}\|_2}$ compared to $\frac{\|\mathbf{b}' - \mathbf{b}\|_2}{\|\mathbf{b}\|_2}$ - we see that we have introduced a small relative error in our right hand side that results in a large relative error in our answer: our relative error in our inputs is appx. $1\text{E} - 4$, while the relative error in our solution is 2.8285, resulting in a relative change of $2.8285\text{E}4$.

Why is this the case? Observe that although our matrix is technically nonsingular, it is pretty close to a singular matrix, namely $\begin{bmatrix}1 & 1\\ 1 & 1\end{bmatrix}$. We can confirm this by looking at the SVD of our matrix. The singular values are $\sigma_1 = 2.00005$, $\sigma_2 = 0.00005$. Although our matrix is not singular, we can see that one of the singular values is very close to 0, and so we are very close to a rank-deficient matrix.

Another way of looking at this is to look at the value we are computing: $\mathbf{x} = \mathbf{A}^{-1}\mathbf{b}$. Again using the SVD, we get $\mathbf{x} = \mathbf{V}\Sigma^{-1}\mathbf{U}^T\mathbf{b}$, which can also be expressed as $\mathbf{x} = \sum_{i=1}^{n} \frac{\mathbf{u}_i^T\mathbf{b}}{\sigma_i}\mathbf{v}_i$. When we have a small singular value σ_i, we are dividing by a small number, and so a small change in $\mathbf{u}_i^T\mathbf{b}$ is magnified.

One note: we also learned in linear algebra that for singular matrices $\mathbf{A}$, $\det(\mathbf{A}) = 0$. Unfortunately, this is not a reliable predictor of ill-conditioned matrices; there are numerous examples of matrices that are nearly singular with perfectly reasonable determinants, and similarly matrices that have very small determinants but are perfectly well behaved. For an instance of the latter, consider $a\mathbf{I}$ for any small choice of a: $\det(a\mathbf{I}) = a$, but the matrix $a\mathbf{I}$ does not magnify the relative error in our solution.

What we would like to do is bound the maximum error we will see in our outputs relative to the error in the inputs. For the moment, only consider errors in our right hand side $\mathbf{b}$. Later on we will discuss errors in our matrix $\mathbf{A}$ as well.

Suppose we have a RHS vector $\mathbf{b}$, and a perturbed RHS $\mathbf{b} + \delta\mathbf{b}$. Let $\mathbf{x} = \mathbf{A}^{-1}\mathbf{b}$; that is, $\mathbf{x}$ is the exact solution to our original system. We can then write

$$\mathbf{A}(\mathbf{x} + \delta\mathbf{x}) = \mathbf{b} + \delta\mathbf{b}$$

that is, $\mathbf{x} + \delta\mathbf{x}$ is the solution to the perturbed system. Expanding the left hand side leads to

$$\begin{aligned}\mathbf{A}\mathbf{x} + \mathbf{A}\delta\mathbf{x} &= \mathbf{b} + \delta\mathbf{b}\\ \mathbf{A}\delta\mathbf{x} &= \delta\mathbf{b}\\ \delta\mathbf{x} &= \mathbf{A}^{-1}\delta\mathbf{b}\end{aligned}$$

since $\mathbf{A}\mathbf{x} = \mathbf{b}$. Since we are interested in bounding the magnitude of the error, we now take the norms of both sides. As we saw before, we have many choices for what norm to use, and each norm will give different answers. In this case, though, it doesn't particularly matter which norm we choose, since we are interested in the magnitude of the changes, and

anyways, vector norm a can be bounded by vector norm b and constants c_1, c_2 such that $c_1\|\mathbf{v}\|_b \leq \|\mathbf{v}\|_a \leq c_2\|\mathbf{v}\|_b$. So, in short, it doesn't matter which norm we use as long as we use the same norm everywhere (for both vectors and matrices)

Taking the norms of both sides gives us

$$\begin{aligned}\|\delta\mathbf{x}\| &= \|\mathbf{A}^{-1}\delta\mathbf{b}\| \\ &\leq \|\mathbf{A}^{-1}\|\|\delta\mathbf{b}\|\end{aligned}$$

where the second statement is true by the definition of the matrix norm induced by the vector norm. This tells us the absolute value of the change, but we would like to bound the relative error. To do that, we can also use the following inequality:

$$\begin{aligned}\|\mathbf{b}\| &= \|\mathbf{A}\mathbf{x}\| \\ &\leq \|\mathbf{A}\|\|\mathbf{x}\|\end{aligned}$$

We can combine these two inequalities as follows:

$$\begin{aligned}\|\delta\mathbf{x}\| &\leq \|\mathbf{A}^{-1}\|\|\delta\mathbf{b}\| \\ \frac{\|\delta\mathbf{x}\|}{\|\mathbf{b}\|} &\leq \|\mathbf{A}^{-1}\|\frac{\|\delta\mathbf{b}\|}{\|\mathbf{b}\|} \\ \frac{\|\delta\mathbf{x}\|}{\|\mathbf{A}\|\|\mathbf{x}\|} &\leq \|\mathbf{A}^{-1}\|\frac{\|\delta\mathbf{b}\|}{\|\mathbf{b}\|} \\ \frac{\|\delta\mathbf{x}\|}{\|\mathbf{x}\|} &\leq \|\mathbf{A}\|\|\mathbf{A}^{-1}\|\frac{\|\delta\mathbf{b}\|}{\|\mathbf{b}\|} \\ \frac{\|\delta\mathbf{x}\|}{\|\mathbf{x}\|} &\leq \kappa(\mathbf{A})\|\frac{\|\delta\mathbf{b}\|}{\|\mathbf{b}\|}\end{aligned}$$

where $\kappa(\mathbf{A}) = \|\mathbf{A}\|\|\mathbf{A}^{-1}\|$ is called the *condition number* of the matrix $\mathbf{A}$. If $\mathbf{A}$ is singular, then we define $\kappa(\mathbf{A}) = \infty$. Note the following properties of the condition number:

1. $\kappa(\mathbf{A}) = \kappa(\mathbf{A}^{-1})$

2. $\kappa(\mathbf{A}) \geq 1$
3. $\kappa(c\mathbf{A}) = \kappa(\mathbf{A})$ for any $c \neq 0$
4. If we are using the 2-norm for our analysis, then $\kappa_2(\mathbf{A}) = \frac{\sigma_1}{\sigma_n}$.
5. $\kappa_2(\mathbf{Q}) = 1$ if $\mathbf{Q}$ is orthogonal

So from above, the condition number bounds the maximum amount of magnification in the relative error of the input that is propagated to the solution. As it turns out, this is a tight bound (within the constraints of our floating point system) – we can choose $\mathbf{b}$ and $\mathbf{e}$ to achieve this maximum error growth. As an example of getting close to this maximal error growth, recall our example from above: the change in relative growth was $2.8285\text{E}4$. Computing the condition number of the matrix gives us $4.0002\text{E}4$, so already we are quite close to the maximal growth, and if we had chosen our RHS and perturbation a bit more carefully (the numbers were chosen more for ease of computation) we could have achieved a growth of $4.0002\text{E}4$ in our relative error.

Using this, we can now see why some of the systems we have discussed are "difficult" to solve. In particular, take the normal equations method for solving the least squares problem. In that, we end up with a system $\mathbf{A}^T\mathbf{A}\mathbf{x} = \mathbf{A}^T\mathbf{b}$. Take the SVD of the matrix $\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$. Then $\mathbf{A}^T\mathbf{A} = \mathbf{V}\mathbf{\Sigma}^2\mathbf{V}^T$. Using the 2-norm for our analysis, then we get $\kappa(\mathbf{A}^T\mathbf{A}) = \frac{\sigma_1^2}{\sigma_n^2} = \left(\frac{\sigma_1}{\sigma_n}\right)^2$. Forming the normal equations squares the condition number of the matrix, and thus squares the maximum amount of magnification in error, meaning that we have less confidence in our solution for a given level of input error.

Similarly, take the Vandermonde matrix of size 12×12, consisting of $x_i = \frac{i}{7}12$ (that is, the 12 points are evenly spaced from 0 to 1). We said that Vandermonde systems are difficult to solve for high degree polynomials; in this case, the condition number is $3.306\text{E}9$. The system magnifies errors greatly, and so any small amount of uncertainty or perturbation in our initial inputs will cause us to potentially lose much of the certainty in our computed answer.

Finally, we have only talked about errors in the right hand side of the equation $\mathbf{A}\mathbf{x} = \mathbf{b}$. Suppose instead that we have some errors in $\mathbf{A}$ – we then have $(\mathbf{A}+\mathbf{E})(\mathbf{x}+\delta\mathbf{x}) = \mathbf{b}+\delta\mathbf{b}$. As it turns out, a careful analysis using calculus on matrices leads to the conclusion that the error is still bounded by the condition number of the matrix $\mathbf{A}$:

$$\frac{\|\delta\mathbf{x}\|}{\|\mathbf{x}\|} = \kappa(\mathbf{A})\left(\frac{\|\delta\mathbf{b}\|}{\|\mathbf{b}\|} + \frac{\|\mathbf{E}\|}{\|\mathbf{A}\|}\right)$$

This means that whether we have uncertainty / errors in our data matrix **or** our right hand side, we can still bound the maximum amount of error we will have in our solution.

Some caveats: this is measuring the relative error of the vector as a whole (using the vector norm), and not at the level of individual components of the vector. We can also end up with poorly conditioned systems that are the result of bad scaling: consider

$$\begin{bmatrix} 1 & 0 \\ 0 & \epsilon \end{bmatrix} \mathbf{x} = \begin{bmatrix} 1 \\ \epsilon \end{bmatrix}$$

The matrix is ill-conditioned for small ϵ, but if we scale the second equation (both sides) by $\frac{1}{\epsilon}$, we end up with a perfectly conditioned system.

Sources

Gene H. Golub and Charles F. Van Loan. *Matrix Computations, Third Edition*. Section 2.7 (pages 80-85). The Johns Hopkins University Press, 1996.

Michael T. Heath. *Scientific Computing: An Introductory Survey, Second Edition*. Section 2.3 (pages 52-62) and Section 2.4.10 (pages 83-84). McGraw-Hill, 2002.

Cleve B. Moler. *Numerical Computing with* MATLAB. Section 2.9 (pages 66-72). Society for Industrial and Applied Mathematics, 2004.

Steve Marschner. *CS322 Notes: Condition Numbers*. CS322, Spring 2007. http://www.cs.cornell.edu/courses/cs322/2007sp/notes/condition.pdf.