Scientific Computing

Jonathan Goodman, Fall, 2022

1 Linear Algebra, perturbations, conditioning

This Section describes tools for deciding how accurate a linear algebra calculation might be. Sometimes accuracy depends on the type of problem or the type of matrix involved, but more often it depends on the specific problem or the specific matrix. For example, the eigenvalues of a symmetric matrix generally can be computed reliably, except possibly the ones closest to zero. The eigenvalues of a non-symmetric matrix, the problem of computing them, might be too ill conditioned to be done in double precision floating point. But this depends on the matrix.

Perturbation theory for matrices and linear algebra means estimating the change in the solution to a linear algebra problem caused by a small change in the input. For example, suppose $Ax = b$ and $(A+\Delta A)(x+\Delta x) = b$. The change in the answer is Δx and the change in the problem is ΔA. Perturbation theory for linear systems of equations describes the relation between ΔA and Δx when ΔA is small. There is perturbation theory for eigenvalues and eigenvectors, and for matrix factorizations.

Perturbation theory has many uses, one of them being to determine the conditioning of linear algebra problems. It allows you to predict how rounding the data of the problem changes the mathematical answer.

The concept of *condition number* or *conditioning* in linear algebra is more subtle in linear algebra than for simple functions with one input. In linear algebra, the input is a vector or matrix with many components. The effect of a perturbation depends on the size and also on its direction. This direction may be hard to know in advance, for example, if the perturbation is caused by rounding the components of the input matrix or vector. A *worst case* analysis provides a single number, usually called a *condition number* that gives the largest change in the output that can be caused by a small change in the input in any direction. The worst case change is caused by a perturbation in the worst case direction.

A worst case analysis calls for a way to measure the size of a perturbation that has many components. We use vector and matrix *norms* for this. Unfortunately there are different vector and matrix norms. Sometimes different norms give similar answers, but sometimes not. Different norms give different results, for example, when a norm is added as a *regularization* of an ill conditioned problem. We saw that regularizing a linear least squares problem using the $2-$norm (Tikhonov regularization) can be solved using the SVD. Regularizing using the $1-$norm is more likely to give a *sparse* answer, which is an answer with many components equal to zero.

2 Norms

This is review, for most readers. If it is not review for you, look in a linear algebra book for fuller explanations.

A *vector norm* is a number associated to a vector that "measures" the size of the vector. The norm of a vector x is written $\|x\|$. If we need to distinguish between different norms (there are many) we use a subscript such as $\|x\|_2$ for the 2−norm. A function $x \rightsquigarrow \|x\|$ is a norm if it has these properties:

positivity: $\|x\| \geq 0$ for all x, and $\|x\| = 0$ only if $x = 0$.

homogeniety: $\|ax\| = |a| \cdot \|x\|$ (a is a real or possibly complex number)

triangle inequality: $\|x + y\| \leq \|x\| + \|y\|$.

Examples:

2−norm: $\|x\|_2 = \left(|x_1|^2 + \cdots + |x_n|^2\right)^{\frac{1}{2}}$. (also called the *euclidean* norm)

1−norm: $\|x\|_1 = |x_1| + \cdots + |x_n|$

max norm: $\|x\|_\infty = \max\{|x_1|, \cdots, |x_n|\}$ (also called the *infinity* norm)

p−norm: $\|x\| = (|x_1|^p + \cdots + |x_n|^p)^{\frac{1}{p}}$. Must have $p \geq 1$.

H−norm: $\|x\|_H = \left(x^T H x\right)^{\frac{1}{2}}$. H is a positive definite symmetric matrix.

The 1−norm and 2−norm are the p−norm with $p = 1$ and $p = 2$ respectively. The max norm is the limit of the p−norm as $p \to \infty$, which explains the notation $\|x\|_\infty$. The 2−norm formula is simpler when the components x_k are real, because $|x_k|^2 = x_k^2$. The H−norm is the 2− norm when x is real and $H = I$. The fact that the p−norm satisfies the triangle inequality "might not be obvious". It is the *Minkowski inequality*.

The "ball" of "radius" $r > 0$, relative to a norm is

$$B_r = \{\ x \text{ with } \|x\| \leq r\}\ .$$

It is a geometric ball in 3D for the 2−norm (hence the term "ball"). A "ball" for the 1−norm in 2D is a square centered at the origin rotated so that its corners are on the axes. A ball in the max norm, in 2D, is a square centered at the origin with horizontal and vertical sides. The triangle inequality is equivalent to a ball being convex. A set S is convex if $x \in S$ and $y \in S$ implies that $\lambda x + (1 - \lambda)y \in S$ if $0 \leq \lambda \leq 1$. The set of such points is the line segment from x to y. Convex means that if points x and y are in S, then the line segment between them is in S. A ball in 2D for an H−norm is an ellipse. In more than two dimensions, an H−norm ball is an *ellipsoid*.

There is more than one notion of the norm of a matrix. One depends directly on the sizes of the entries. Another, often called the *operator norm*, depends

on A as a linear transformation. One commonly used norm that depends on entries is the *Frobenius* norm

$$\|F\|_F = \left(\sum_{j=1}^{m} \sum_{k=1}^{n} a_{jk}^2 \right)^{\frac{1}{2}} .$$

This norm has the property that the norm of the $n \times n$ identity matrix is

$$\|I\|_F = \sqrt{n} \, .$$

Operator norms are defined in terms of the amount by which a matrix "stretches" a vector. The stretch is the ratio of the size of x and the size of Ax. Naturally, those sizes are measured using vector norms. Thus, for any vector norm (more precisely, any pair of vector norms), there is an *induced* matrix norm defined by

$$\|A\| = \max_{x \neq 0} \frac{\|Ax\|}{\|x\|} \, . \tag{1}$$

The norm of A is the largest stretch that A creates, looking at all non-zero vectors x. If we use the $2-$norm for vectors, the corresponding operator matrix norm is the matrix $2-$norm, and similarly for other vector norms:

$$\|A\|_2 = \max_{x \neq 0} \frac{\|Ax\|_2}{\|x\|_2} \, , \text{etc.}$$

The operator matrix norm of the identity matrix, clearly, is equal to 1 for any dimension, because the numerator and denominator in (1) are always the same. This shows, for example, that the Frobenius norm of a matrix is not induced by any vector norm.

All matrix norms have properties they would have if you think of the matrix as a vector. They are positive, homogeneous, and satisfy the triangle inequality $\|A + B\| \leq \|A\| + \|B\|$, assuming A and B have the same shape so they can be added. An operator matrix norm, one induced from vector norms, also has the crucial property that is:

multiplicative: $\|AB\| \leq \|A\| \, \|B\|$, if A and B can be multiplied.

The proof of the multiplicativity property is simple suggests how to think about matrix operator norms. It starts with the basic observation: if s is any positive number and if $\|Ax\| \leq s \, \|x\|$ for all x, then $\|A\| \leq s$. To see this, let x_* be a vector that gives the maximum stretch in the matrix norm (1). Then

$$\|A\| = \frac{\|Ax_*\|}{\|x_*\|} \leq \frac{s \, \|x_*\|}{\|x_*\|} \leq s \, .$$

This might be an unclear way to say something simple. Saying $\|Ax\| \leq s \, \|x\|$ for all x says that A does not stretch any vector more than by a factor of s. That implies that the maximum stretch cannot be more than s.

At the same time, the matrix operator norm definition (1) implies that $\|Ax\| \leq \|A\| \, \|x\|$. With this, we verify the multiplicitivity property. If x is any vector, the stretch of AB applied to x is at most:

$$\begin{aligned} \|ABx\| &\leq \|A\| \, \|Bx\| && \text{(apply the } \|A\| \text{ inequality to the vector } Bx) \\ &\leq \|A\| \, \|B\| \, \|x\| && \text{(apply the } \|B\| \text{ inequality to the vector } x) \\ \|(AB)\, x\| &\leq (\|A\| \, \|B\|) \, \|x\| && \text{(the multiplicative property)} \end{aligned}$$

Matrix operator norms have simple mathematical properties, but there usually is no formula for the norm in terms of the matrix elements. The exceptions are the matrix $1-$norm and max norm. The matrix $1-$ norm is the maximum *column sum* of the matrix:

$$\|A\|_1 = \max_k \sum_{j=1}^{m} |a_{jk}| \ . \tag{2}$$

The sum on the right is the sum of the absolute values of the entries in column k of the matrix. The matrix max norm is the maximum *row sum*

$$\|A\|_\infty = \max_j \sum_{k=1}^{n} |a_{jk}| \ . \tag{3}$$

The $2-$norm of A is the largest singular value of A. But this is a tautology, as the definition of σ_1 involved the maximization problem (1). The proofs of (2) and (3) have two steps. First you find a vector a vector x_* with $\|Ax_*\| \leq (\max) \, \|x_*\|$. The max being the column or row sum, depending on which case. Then you show that $\|Ax\| \leq (\max) \, \|x\|$ for any other x.

Clearly there are many vector norms and matrix norms. The choice of norms may or may not matter, depending on the situation. If the norm does not matter, we just write $\|x\|$ without saying which norm. All that matters is that the same norm is used each time. The matrix operator norm definition (1) is an example. Another example is in error bounds with unspecified constants. As an example of that, let $ff(x) = f(x_1, \cdots, x_n)$ be a function of n variables. The first order Taylor approximation for small x is

$$f(x) = f(x) + \nabla f(x)^T x + O\left(\|x\|^2\right) \ .$$

The "big oh" on the right means 'on the order of", without saying exactly where. Technically, it means that there are positive numbers, C and r, so that if $\|x\| < r$ then

$$\left| \, f(x) - \left[f(x) + \nabla f(x)^T x \right] \, \right| \leq C \, \|x\|^2 \ . \tag{4}$$

The $\|x\| < r$ is the "where", and $|\cdots| \leq C \, \|x\|^2$ is says the quantity on the left is "on the order of" $\|x\|^2$. (There is more on big oh coming in a future section.)

The choice of norm in the error bound (4) is irrelevant because all vector norms in any given dimension are *equivalent*. "Equivalent" does not mean "the

same" in the sense of being equal or having the same value (the literal meaning of "equi-valent"). It means that each one is bounded in terms of the other. If $\|\cdot\|_a$ and $\|\cdot\|_b$ are any two vector norms, then there are positive numbers $r_{ab} \leq R_{ab}$ so that, for all x,

$$r_{ab} \|x\|_a \leq \|x\|_b \leq R_{ab} \|x\|_a \ . \tag{5}$$

If the error bound (4) is true in the $b-$norm, $\|\cdot\|_b$, then it is also true in the $a-$norm, but with the constant $C' = R_{ab}C$. That is

$$C \|x\|_b \leq CR_{ab} \|x\|_a \ .$$

The "equivalence" inequalities (5) represent the fact that different ways to measure the "size" of a collection of n number yield different measures. For example, let $\|x\|_a$ be the max norm and $\|x\|_b$ the $1-$norm. If x is the vector of all ones, then $\|x\|_\infty = 1$ and $\|x\|_1 = 1 + \cdots + 1 = n$. This suggests that the best R in (5) is $R_{\infty,1} = n$ and

$$\|x\|_1 \leq n \|x\|_\infty \ .$$

The factor of n may be harmless. But if n is a million (not large vector in modern computing), it could be serious. The bigger the computer, and the bigger the vector, the more it might matter.

The sums that define norms are problematic when the components of a vector or a matrix have different unite. For example, suppose x is a two component vector with x_1 representing time measured in hours and x_2 representing distance, measured in meters. The expression $\|x\|_1 = |x_1| + |x_2|$ does not make sense because it is some number of hours plus some number of meters. You have to be thoughtful when you assign a simple size number to a complicated object.

3 Condition number

The concept of condition number is critical for designing computing strategies. But it is impossible to define precisely for problems with many inputs and outputs. The best of the not-good approaches to conditioning for multi-variate problems involves norms and worst-cast analysis.

Suppose the inputs of a problem are $x_1, \cdots, x_n$ and these numbers are arranged into a column vector x. Suppose the outputs are $F_1(x), \cdots, F_m(x)$, also arranged into a column vector $F(x)$. (We call the problem F instead of A because A is a matrix.) A change in the input is another n component column vector $\Delta x = (\Delta x_1 \cdots, \Delta x_n)^T$. The size of the perturbation is $\|\Delta x\|$ and the relative size is the ratio

$$\frac{\|\Delta x\|}{\|x\|} \ .$$

The change in the output is $\Delta F = F(x + \Delta x) - F(x)$. The condition number ratio is

$$\frac{\frac{\|\Delta F\|}{\|F\|}}{\frac{\|\Delta x\|}{\|x\|}} \ .$$

For univariate functions, we just took the limit of this ratio, in the limit $\Delta x \to 0$. Now, even after we constrain $\|\Delta x\|$ to be small, still Δx can be in any direction. The condition number we use takes the worst possible direction.

$$\kappa(x) = \lim_{r \to 0} \max_{\|\Delta x\| = r} \frac{\frac{\|\Delta F\|}{\|F\|}}{\frac{\|\Delta x\|}{\|x\|}} \ . \tag{6}$$

There is a differential formula for κ, which applies when F is differentiable. The *jacobian* matrix of F at a point x may be written $J(x)$ or $F'(x)$, but we prefer DF. The matrix entries of the jacobian matrix are

$$(DF)_{jk} = \frac{\partial F_j}{\partial x_k} \ .$$

The first derivative approximation is the matrix/vector product:

$$\Delta F \approx DF \, \Delta x \ .$$

This is equivalent to the component by component formula

$$\Delta F_j \approx \sum_{k=1}^{n} \frac{\partial F_j}{\partial x_k} \, \Delta x_k \ .$$

In the limit of (6), we may use the first derivative approximation for ΔF. This approximation, and rearranging the fraction give

$$\kappa(x) = \lim_{r \to 0} \left(\max_{\|\Delta x\| = r} \frac{\|DF \, \Delta x\|}{\|\Delta x\|} \right) \frac{\|x\|}{\|F\|} \ .$$

The max in parens is the matrix operator norm of DF, which does not depend on the norm of Δx. The final formula is

$$\kappa(x) = \|DF\| \, \frac{\|x\|}{\|F\|} \ . \tag{7}$$

This formula is consistent with the one we had earlier for one simple functions of one variable, $\kappa(x) = \left| f'(x) \frac{x}{f(x)} \right|$. The formula (7) replaces scalar quantities such as f' with norms of the corresponding matrix or vector quantities. Using norms is a worst-case analyis

4 Condition number of a matrix

There is a formula for "the condition number" of a matrix. You should know this formula because people use it a lot. But you also should understand that it is even more of a worst case analysis than the general formula (7).

Suppose the function F in (7) is linear and defined by a square invertible matrix A, as $F(x) = Ax$. Then A is the jacobian matrix: $DF = A$. The condition number, at a point x, is

$$\|A\| \, \frac{\|x\|}{\|Ax\|} \; .$$

We can ask about the worst case x. The worst case x is the one that maximizes the fraction on the right. You have to rule out $x = 0$, where the condition number does not make sense because the relative change of x from $x = 0$ is infinite (one reason).

$$\max_{x \neq 0} \frac{\|x\|}{\|Ax\|} \; .$$

The maximization can be found using the change of variable $y = Ax$, which may be written in the form $x = A^{-1}y$. This is where it matters that A is invertible. Using the y variable, the maximization problem becomes the definition of the matrix operator norm of A^{-1}:

$$\max_{y \neq 0} \frac{\left\|A^{-1}y\right\|}{\|y\|} = \left\|A^{-1}\right\| \; .$$

Therefore, if we take the worst case x in (7) we get what is called the condition number of the matrix A itself.

$$\kappa(A) = \|A\| \left\|A^{-1}\right\| \; . \tag{8}$$

This formula has the strange property that

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

This implies that the worst case conditioning of computing Ax from x is the same as the worst case conditioning of solving $Ax = b$, which is the same mathematical problem as computing $x = A^{-1}b$.

5 Perturbation theory

The terms "perturbation theory" and "sensitivity" both refer to the small changes in the answer to a problem caused by small changes in the problem. Said more simply, they refer to derivatives.

Perturbation theory in linear algebra has many uses, past and present. Before computers, there were a few problems with explicit solutions. Perturbation theory allowed one to find approximate solutions to nearby problems. With computers, perturbation theory "updates" can be cheaper than recomputing from the beginning, and accurate enough. Perturbation theory is also a theoretical tool that can help us design a computational strategy. We use it to study the conditioning of various linear algebra problems.

Perturbation theory can be thought of as ordinary differential calculus, However, some common ways of denoting ordinary derivatives seem confusing when you're differentiating, the eigenvectors of a matrix with respect to the matrix entries. Instead, we may use the idea of *directional derivative*. The directional derivative says how a function changes if you change the argument a little, in a given direction. Specifically, the directional derivative of a function $f(x)$ at a point x in the direction y is

$$\frac{d}{ds} f(x + sy)\Big|_{s=0} = \Big(\nabla f(x)\Big)^T y \ .$$

You can write this using a Taylor series in the single variable s

$$\begin{aligned} f(x + sy) &= f(x) + s\Big(\nabla f(x)\Big)^T y + O(s^2) \\ &= f(x) + \Big(\nabla f(x)\Big)^T (sy) + O(s^2) \ . \end{aligned}$$

This is close to (or the same as) the multi-variate first order Taylor series formula

$$f(x + y) = f(x) + \Big(\nabla f(x)\Big)^T y + O(\|y\|^2) \ . \tag{9}$$

Just replace y by sy. We say that the difference between $f(x)$ and $f(x + y)$, to *leading order*, or to *first order*, is $\Big(\nabla f(x)\Big)^T y$.

There are tricks for calculating first derivatives and leading order perturbations. Many of these tricks use the idea that multiplying two first order changes gives a product that can be neglected – to first order. For example, $(x + y)^2 = x^2 + 2xy$, to leading order. Also $(x + y)^3 = x^3 + 3x^2 y$, to leading order. Therefore, to leading order

$$\begin{aligned} (x + y)^2 (x + y)^3 &= \left[x^2 + 2xy + \cdots\right] \left[x^3 + 3x^2 y + \cdots\right] \\ &= x^2 x^3 + x^2 \left(3x^2 y\right) + (2xy)\, x^3 + \cdots \\ &= x^5 + 5x^4 y + \cdots \ . \end{aligned}$$

This is a way to think of the product rule from calculus, at least in the case where x^2 (derivative $2x$) and x^3 (derivative $3x^2$) are multiplied.

An example with matrices shows how easy and clear calculations like this can be. Suppose A and B are $n \times n$ matrices, with A invertible and B small. We want to estimate the difference between $(A + B)^{-1}$ and A^{-1}. We suppose that there is a leading order approximation in which C (another $n \times n$ matrix) is the perturbation. That is:

$$(A + B)^{-1} = A^{-1} + C + \cdots \ .$$

We find a leading order formula for C, starting with the definition of inverse

matrix, calculating, and neglecting products of first order quantities.

$$\begin{aligned} I &= (A+B)(A+B)^{-1} \\ &= [A+B]\left[A^{-1}+C+\cdots\right] \\ &= AA^{-1}+AC+BA^{-1}+\cdots \\ &= I+AC+BA^{-1}+\cdots \\ AC &= -BA^{-1}+\cdots \\ C &= A^{-1}BA^{-1} \text{ to leading order } . \end{aligned}$$

We can write this as

$$(A+B)^{-1} = A^{-1} - A^{-1}BA^{-1} + \cdots . \tag{10}$$

This is first order perturbation theory applied to the matrix inverse.

You can think of the matrix inverse perturbation formula (10) as an instance of the general first order perturbation formula (9). The argument x is the matrix A (with n^2 components). The function if $f(A) = A^{-1}$, which is a differentiable function of A as long as A is non-singular. The perturbation is $y = B$. In the general formula (9), the change in f is linear in the perturbation y. In our matrix formula, the change in the inverse is linear in the perturbation B. The specific formula (10) expresses this linear relationship more easily than creating notation to define the gradient of A^{-1} with respect to A. It can be done, but the formula (10) is easier to use, at least here.

We can do matrix perturbation calculations using other notation for perturbations. One possibility is writing ΔA for the perturbation in A. The matrix inverse perturbation formula becomes

$$(A+\Delta A)^{-1} = A^{-1} - A^{-1}\,\Delta A\,A^{-1} + O(\|\Delta A\|^2) .$$

Let's apply this kind of notation to computing perturbations of the LU factorization. Suppose $A = LU$ and $A+\Delta A = (L+\Delta L)(U+\Delta U)$. The notation, and the calculations below, assume that ΔL is lower triangular and ΔU is upper triangular. We saw that it is possible to assume that L has ones on its diagonal, so we assume that ΔL has zeros on its diagonal (the perturbations on the diagonal of L are zero because the elements themselves do not change). We ignore the permutation matrix P, but it should be clear how to put it in if you need to. Recall that the inverse of an upper triangular matrix is upper triangular, and similarly for lower triangular matrices. In the last line, we multiply from the left by L^{-1} and from the right by U^{-1}.

$$\begin{aligned} A+\Delta A &= (L+\Delta L)(U+\Delta U) \\ &= LU + \Delta L\,U + L\,\Delta U + \cdots \\ \Delta A &= \Delta L\,U + L\,\Delta U + \cdots \\ L^{-1}\,\Delta A\,U^{-1} &= L^{-1}\Delta L + \Delta U U^{-1} + \cdots . \end{aligned}$$

This gives an algorithm for computing the leading order perturbations ΔL and ΔU. Each entry on the left "belongs" to only one of the matrices on the right. The entries strictly below the diagonal belong to $L^{-1}\Delta L$, because $\Delta U U^{-1}$ is zero below the diagonal. Similarly, $L^{-1}\Delta L$ is zero (it's entries are zeros) on the diagonal and above. This may be said formally using the notation

$$
\begin{aligned}
&L^{-1}\Delta A U^{-1} = M + N \ , \\
&M_{jk} = 0 \text{ for } k \geq j \ (M \text{ is strictly lower triangular}) \\
&N_{jk} = 0 \text{ for } k < j \ (N \text{ is upper triangular, including the diagonal})
\end{aligned}
$$

Matching the non-zeros, we get two equations

$$
\begin{aligned}
M = L^{-1}\Delta L \implies \Delta L = LM \\
N = \Delta U U^{-1} \implies \Delta U = NU \ .
\end{aligned}
$$

These formulas are easy to implement in code. Looking back, you can see that ΔL and ΔU are linear functions of M and N, which are linear functions of ΔA. You can see how clumsy it would be to express these linear relationships directly in the form (9).

6 Eigenvalues, symmetric matrices

The *symmetric eigenvalue problem* is the problem of finding eigenvalues and eigenvectors of a symmetric matrix A. Symmetric matrices occur in "nature" in several ways, including as covariance matrices, as matrices of second partial derivatives (Hessian matrices), and matrices from physical processes. Eigenvalues of symmetric matrices may be the frequencies of "normal modes" of vibration.

The symmetric eigenvalue problem is different from the general one (not necessarily symmetric) in several ways. Eigenvalues are real, eigenvectors corresponding to distinct eigenvalues are orthogonal, no non-trivial Jordan blocks (symmetric matrices are diagonalizable). All of these properties may be seen as consequences of a variational principle for eigenvalues and eigenvectors, a principle that is similar to the one for singular values and singular vectors. The symmetric eigenvalue problem is

- $A^T = A$, A being a real $n \times n$ matrix
- $Av_j = \lambda_j v_j$, $j = 1, \cdots, n$
- λ_j and v_j are real
- $v_j^T v_k = 0$, if $\lambda_j \neq \lambda_k$
- *eigen-space* $= E(\lambda) =$ span $\{v | Av = \lambda v\}$
- $\dim(E(\lambda)) = 0$ if λ is not an eigenvalue of A

The perturbation theory for eigenvalues and eigenvectors of symmetric matrices is sometimes called *Rayleigh Schrödinger* perturbation theory. Lord Rayleigh (a real British Lord) used it in the early 1900's to estimate vibrational frequencies. Erwin Schr'odinger (Austrian physicist, discoverer of the *Schrödinger equation*) used it to estimate the solution of quantum mechanical problems that are close to problems with closed form solutions.

Rayleigh Schrödinger perturbation theory may be derived using the same style of calculation we used to for perturbations of matrix factorizations. The simplest case is