II.7 Condition numbers

We have seen that floating point arithmetic induces errors in computations, and that we can typically bound the absolute errors to be proportional to $C\epsilon_{\mathrm{m}}$. We want a way to bound the effect of more complicated calculations like computing $A\mathbf{x}$ or $A^{-1}\mathbf{y}$ without having to deal with the exact nature of floating point arithmetic, as it will depend on the *data* A and $\mathbf{x}$. That is, we want to reduce floating point stability to a more fundamental property: *mathematical stability*: how does a mathematical operation like $A\mathbf{x}$ change in the precense of small perturbations (random noise or structured floating point errors)?

1. Backward error analysis: We introduce the concept of *backward error* analysis, which is a more practical

way of understanding and bounding floating point errors. 2. Condition numbers: We introduce a *condition numbers*, which can capture the effect of perturbations in A for linear algebra oerations. More precisely: matrix operations are mathematically *stable* when the condition number is small. 3. Bounding floating point errors for linear algebra: we see how simple operations like $A\mathbf{x}$ can be put into a backward error analysis framework, leading to bounds on the errors in terms of the condition number.

1. Backward error analysis

So far we have done forward error analysis, e.g., to understand $f(x) \approx f^{\mathrm{FP}}(x)$ we consider either the absolute

$$f^{\mathrm{FP}}(x) = f(x) + \delta_{\mathrm{a}}$$

or relative

$$f^{\mathrm{FP}}(x) = f(x)(1 + \delta_{\mathrm{r}})$$

errors of the *output*. More generally, for two vector spaces V and W (e.g. $V = \mathbb{R}^n$ and $W = \mathbb{R}^m$) consider functions $\mathbf{f} = V \to W$. We write

$$\mathbf{f}^{\mathrm{FP}}(\mathbf{x}) = \mathbf{f}(\mathbf{x}) + \delta_{\mathrm{f}}$$

where we bound a norm of $\delta_{\mathrm{f}} \in W$ either *absolutely*:

$$\|\delta_{\mathrm{f}}\|_W \leq C\varepsilon$$

or *relative* to the true result:

$$\|\delta_{\mathrm{f}}\|_W \leq C\|\mathbf{f}(\mathbf{x})\|_W\varepsilon$$

(which is similar to PS4, Q1.3).

On the other hand, *backward error analysis* considers computations as errors in the *input*. That is, one writes the approximate function as the true function with a pertubed input: e.g. find $\tilde{\mathbf{x}} \in V$ such that

$$\mathbf{f}^{\mathrm{FP}}(\mathbf{x}) = \mathbf{f}(\tilde{\mathbf{x}}).$$

We study the *backward error*, the error in the input

$$\tilde{\mathbf{x}} = \mathbf{x} + \delta_{\mathrm{b}}$$

where $\delta_{\mathrm{b}} \in \mathbb{R}^n$ by bounding either the absolute error

$$\|\delta_{\mathrm{b}}\|_V \leq C\varepsilon$$

or relative error:

$$\delta_{\mathrm{b}}\|_V \leq C\|\mathbf{x}\|_V\varepsilon$$

We shall see that *some* algorithms (like `mul_rows`) lead naturally to backward error results.

2. Condition numbers

So now we get to a mathematical question independent of floating point: can we bound the *relative error* in approximating

$$A\mathbf{x} \approx (A + \delta A)\mathbf{x}$$

if we know a bound on the relative backward error $\|\delta A\|$? It turns out we can in turns of the *condition number* of the matrix:

Definition 2 (condition number) For a square matrix A, the *condition number* (in p-norm) is

$$\kappa_p(A) := \|A\|_p\|A^{-1}\|_p$$

with the default being the 2-norm condition number, writable in terms of the singular values as:

$$\kappa(A) := \kappa_2(A) = \|A\|_2\|A^{-1}\|_2 = \frac{\sigma_1}{\sigma_n}.$$

Theorem 1 (relative forward error for matrix-vector) Assume we have the relative backward error bound $\|\delta A\| \leq \|A\|\varepsilon$. Then for

$$(A + \delta A)\mathbf{x} = A\mathbf{x} + \delta_{\mathrm{f}}$$

the forward error has the relative error bound

$$\|\delta_{\mathrm{f}}\| \leq \|A\mathbf{x}\|\kappa(A)\varepsilon$$

Proof We can assume A is invertible (as otherwise $\kappa(A) = \infty$). Denote $\mathbf{y} = A\mathbf{x}$ and we have

$$\frac{\|\mathbf{x}\|}{\|A\mathbf{x}\|} = \frac{\|A^{-1}\mathbf{y}\|}{\|\mathbf{y}\|} \leq \|A^{-1}\|$$

Thus we have:

$$\frac{\|\delta_{\mathrm{f}}\|}{\|A\mathbf{x}\|} \leq \frac{\|\delta A\|\|\mathbf{x}\|}{\|A\mathbf{x}\|} \leq \underbrace{\|A\|\|A^{-1}\|}_{\kappa(A)}\varepsilon.$$

∎

3. Bounding floating point errors for linear algebra

We now observe that errors in implementing matrix-vector multiplication using floating points can be captured by considering the multiplication to be exact on the wrong matrix: that is, `A*x` (implemented with floating point as `mul_rows`) is precisely $A + \delta A$ where δA has small norm, relative to A. That is, we have a bound on the *backward relative error*.

To discuss floating point errors we need to be precise which order the operations happened. We will use the definition `mul_rows(A,x)` (which is equivalent to `mul_cols(A,x)`). Note that each entry of the result is in fact a dot-product of the corresponding rows so we first consider the error in the dot product `dot(x,y)` as implemented in floating-point:

$$\mathrm{dot}(\mathbf{x}, \mathbf{y}) = \bigoplus_{k=1}^{n} (x_k \otimes y_k).$$

We first need a helper proposition:

Proposition 1 [PS2 Q2.1] If $|\epsilon_i| \leq \epsilon$ and $n\epsilon < 1$, then

$$\prod_{k=1}^{n} (1 + \epsilon_i) = 1 + \theta_n$$

for some constant θ_n satisfying

$$|\theta_n| \leq \underbrace{\frac{n\epsilon}{1 - n\epsilon}}_{E_{n,\epsilon}}.$$

Lemma 1 (dot product backward error) For $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$,

$$\mathrm{dot}(\mathbf{x}, \mathbf{y}) = (\mathbf{x} + \delta\mathbf{x})^\top \mathbf{y}$$

where, assuming $n\epsilon_{\mathrm{m}} < 2$, the entries satisfy

$$|\delta x_k| \leq E_{n,\epsilon_m/2}|x_k|.$$

Proof

This is related to PS2 Q2.3 but asks for the *backward error* instead of the *forward error*. Note

$$\mathrm{dot}(\mathbf{x}, \mathbf{y}) = \bigoplus_{j=1}^{n}(x_j \otimes y_j) = \bigoplus_{j=1}^{n}(x_j y_j)(1+\delta_j) = x_1 y_1(1+\theta_n) + \sum_{j=2}^{n} x_j y_j(1+\theta_{n-j+2})$$

where $|\theta_n|, |\theta_k| \leq E_{n,\epsilon_m/2}$ (the subscript denotes the number of terms bounded by $\varepsilon_m/2$). Thus we can define

$$\delta\mathbf{x} := \begin{bmatrix} x_1\theta_n \\ x_2\theta_n \\ \vdots \\ x_n\theta_2 \end{bmatrix}$$

where

$$|\delta x_k| \leq E_{n,\epsilon_m/2}|x_k|.$$

∎

We can use this to get a relative backward error bound on `mul_rows`:

Theorem 2 (matrix-vector backward error) For $A \in \mathbb{R}^{m\times n}$ and $\mathbf{x} \in \mathbb{R}^n$ (both with normal float entries) we have

$$\mathrm{mul\_rows}(A, \mathbf{x}) = (A + \delta A)\mathbf{x}$$

where, assuming $n\epsilon_m < 2$ and all operations are in the normalised range, the entries (denoting $\delta a_{kj} = \delta A[k, j] = \mathbf{e}_k^\top \delta A \mathbf{e}_j$) satisfy

$$|\delta a_{kj}| \leq E_{n,\epsilon_m/2}|a_{kj}|.$$

Proof The bound on the entries of δA is implied by the previous lemma since each row is equivalent to a dot product. ∎

Corollary 1 (Norms)

$$\begin{aligned} \|\delta A\|_1 &\leq E_{n,\epsilon_m/2}\|A\|_1 \\ \|\delta A\|_2 &\leq \sqrt{\min(m,n)}E_{n,\epsilon_m/2}\|A\|_2 \\ \|\delta A\|_\infty &\leq E_{n,\epsilon_m/2}\|A\|_\infty \end{aligned}$$

In particular,

$$\mathrm{mul\_rows}(A, \mathbf{x}) = A\mathbf{x} + \delta_{\mathrm{f}}$$

where

$$\|\delta_{\mathrm{f}}\| \leq \|A\mathbf{x}\| \kappa(A) E_{n,\epsilon_{\mathrm{m}}/2}$$

Proof

The 1-norm follow since

$$\|\delta A\|_1 = \max_j \sum_{k=1}^m |\delta a_{kj}| \leq E_{n,\epsilon_{\mathrm{m}}/2} \max_j \sum_{k=1}^m |a_{kj}| = E_{n,\epsilon_{\mathrm{m}}/2} \|A\|_1$$

and the proof for the ∞-norm is similar.

This leaves the 2-norm, which is a bit more challenging. We will prove the result by going through the Fröbenius norm and using

$$\|A\|_2 \leq \|A\|_F \leq \sqrt{r} \|A\|_2$$

where r is rank of A (see PS6 Q5.2). So we deduce

$$\begin{aligned} \|\delta A\|_2^2 \leq \|\delta A\|_F^2 &= \sum_{k=1}^m \sum_{j=1}^n |\delta a_{kj}|^2 \leq E_{n,\epsilon_{\mathrm{m}}/2}^2 \sum_{k=1}^m \sum_{j=1}^n |a_{kj}|^2 \\ &= E_{n,\epsilon_{\mathrm{m}}/2}^2 \|A\|_F^2 \leq E_{n,\epsilon_{\mathrm{m}}/2}^2 r \|A\|_2^2. \end{aligned}$$

and the rank of A is bounded by $\min(m, n)$. The bound on the forward error then follows from Theorem 1.

∎

We can also bound the error of back-substitution in terms of the condition number (see PS7). If one uses QR to solve $A\mathbf{x} = \mathbf{y}$ the condition number also gives a meaningful bound on the error. As we have already noted, there are some matrices where PLU decompositions introduce large errors, so in that case well-conditioning is not a guarantee of accuracy (but it still usually works).