Floating-point Arithmetic and Stability

Yu Sheng
Center for Statistical Science

Fall 2018

Representation of real numbers in the computer

- Real numbers differ from integers in that they have a decimal part that potentially can be infinitely long.

- You probably have learned from you programming class that real numbers are declared as “float”, which means floating-point numbers.

- In contrast to floating-point, there was once a mechanism called the “fixed-point” system.

Fixed-point system

- A position in the string is specified for the radix point.
  - E.g., a fixed-point scheme might be to use a string of 8 decimal digits with the decimal point in the middle, whereby "00012345" would represent 0001.2345.
- Disadvantage
  - The scale is fixed, and it is very easy to overflow or underflow.

Floating-point system

- Very similar to *scientific notation*

$$significand \times base^{exponent}$$

  - Significand – integer
  - Exponent – integer
  - Base can be 2, 10, or 16.

- E.g.

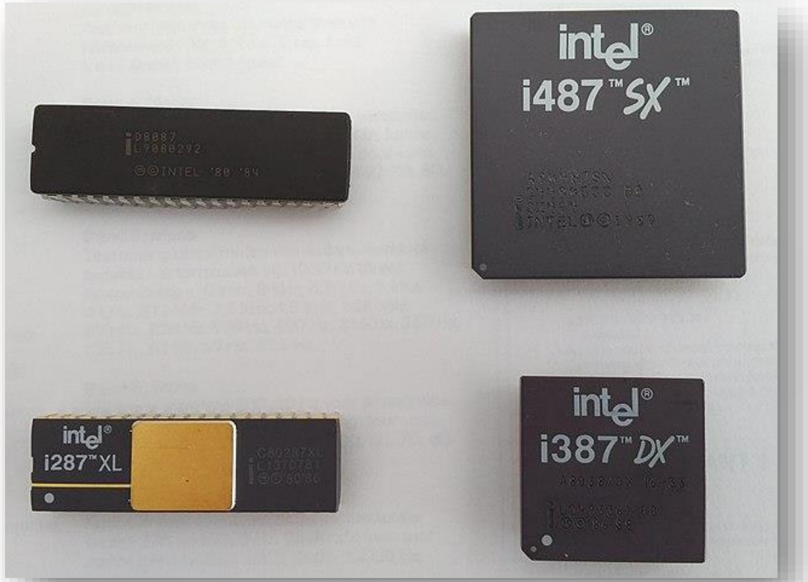
$$1.2345 = 12345 \times 10^{-4}$$

- The term floating point refers to the fact that a number's radix point can "float".

Did you know…

- Early CPUs (such as Intel 8088) could not do floating-point arithmetic, and a separate Floating-point units (FPU, such as Intel 8087) were sold as a add-on coprocessor.

- On the contrary, fixed-point systems were much easier to implement and could use integer arithmetic hardware.



清华大学统计学研究中心

Representable numbers

- Apparently, floating-point system can only represent rational numbers.

- However, not all rational number can be represented by floating-point system.
  - E.g., try to represent 1/5 in binary.
  - Extended question: how to convert decimal to binary?

Base conversion

- For whole numbers, repeatedly divide by the base and record the remainders.
  - E.g., convert decimal 500 to hexadecimal.

- For the fractional part, repeatedly multiply the part after the radix by the base, and record the part before the radix.
  - E.g., convert decimal 0.2 to binary.

Range and precision

- As it turns out, even a number as simple as 0.2 may not be accurately represented as a floating-point number. There are gaps between representable numbers, and the gap scales with the exponent. Naturally, the next question would be how accurate are floating-point numbers.

- By the IEEE 754 Standard, 有效数字

Type	Sign	Exponent	Significand	Number of decimal digits
Half precision	1	5	10	~3.3
Single precision	1	8	23	~7.2
Double precision	1	11	52	~16.0

- At a product release event, Nvidia boasted about half precision computing speed of its GPU, claiming that 3 digits of precision is enough. Is it really so?

Floating-point arithmetic

- **Addition and subtraction**: shift and represent the smaller number with the same exponent with the larger one, then proceed with usual addition or subtraction.

- Use decimal and 7 digits of precision as example:

```
123456.7 + 101.7654 = (1.234567 × 10^5) + (1.017654 × 10^2)
                    = (1.234567 × 10^5) + (0.001017654 × 10^5)
                    = (1.234567 + 0.001017654) × 10^5
                    =  1.235584654 × 10^5
final sum: e=5;  s=1.235585   (123558.5)
```

Floating-point arithmetic

- **Cancellation**: a phenomenon called cancellation can occur when subtracting nearly equal numbers, and it is a common source of loss of significance.

- In this example, the result has in fact only remaining 1 significant digit.

```
  e=5;  s=1.234571
- e=5;  s=1.234567
------------------
  e=5;  s=0.000004
  e=-1; s=4.000000 (after rounding and normalization)
```

Floating-point arithmetic

相加事至多两倍相对误差

- Cancellation is *benign* and controlled within 2 machine epsilon when using an additional *guard digit*.

截断估计（截断近似）

- However, cancellation can be *catastrophic* when the operands involve rounding errors. E.g. the two operands were in fact 123457.1467 and 123456.659. The true result should be e = −1; s = 4.877000, which differs more than 20% from e = −1; s = 4.000000.

```
  e=5;  s=1.234571
- e=5;  s=1.234567
------------------
  e=5;  s=0.000004
  e=-1; s=4.000000 (after rounding and normalization)
```

当各自做了四舍五入，并且相近相减的时候做差可能会有严重误差

Floating-point arithmetic

- **Multiplication and division**: to multiply, the significands are multiplied while the exponents are added, and the result is rounded and normalized. Division is similar.

```
  e=3;   s=4.734612
x e=5;   s=5.417242
-----------------------
  e=8;   s=25.648538980104  (true product)
  e=8;   s=25.64854         (after rounding)
  e=9;   s=2.564854         (after normalization)
```

- There are no cancellation or absorption problems with multiplication or division.

乘法和除法相对较为安全一些

Accuracy problem

- Cancellation is a major source of inaccuracy, and a lot of care needs to be taken.

- Consider the quadratic formula:

$$r_1 = \frac{-b + \sqrt{b^2 - 4ac}}{2a}$$

$$r_2 = \frac{-b - \sqrt{b^2 - 4ac}}{2a}$$

此时在电脑里做运算的时候，根号的运算已经是做过截断近似了，因此会发生像上面的 rounding error

when $b^2 \gg 4ac$, $\sqrt{b^2 - 4ac} \approx |b|$. This implies $-b + \sqrt{b^2 - 4ac}$ or $-b - \sqrt{b^2 - 4ac}$ will involve catastrophic cancellation.

- How can you avoid this problem?

清华大学统计学研究中心

Accuracy problem

- The expression of $x^2 - y^2$ is another formula that exhibits catastrophic cancellation.

- How can you avoid this problem? **使用平方差公式**

Accuracy problem

- Evaluating the derivative numerically:

$$f'(x) = \lim_{d \to 0} \frac{f(x+d) - f(x)}{d}$$

- Intuitively one would want a d very close to zero, however the smallest number of d possible will give a more erroneous approximation of a derivative than a somewhat larger number.
  - Because evaluating $f(x)$ involves rounding error.
  - The computation of $f(x)$ may already be unstable.

导致有时候取大一点的d**比小一点的**d更接近真实值

- To make things worse, $f(x)$ itself may be *ill-conditioned*.

类似于 不稳定的

Condition of a problem

- Consider a problem as a function $f: X \to Y$ from a normed vector space X of data to a normed vector space Y of solutions.

- A *well-conditioned* problem is one with the property that all small perturbations of x lead to only small changes in $f(x)$.

稳定性与否

- An *ill-conditioned* problem is one with the property that some small perturbations of x leads to a large change in $f(x)$.

Absolute condition number

- Let δx denote a small perturbation of x, and write $\delta f = f(x+\delta x) - f(x)$. The *absolute condition number* $\hat{\kappa} = \hat{\kappa}(x)$ of the problem f at x is defined as

$$\hat{\kappa} = \lim_{\delta \to 0} \sup_{\|\delta x\| \le \delta} \frac{\|\delta f\|}{\|\delta x\|} \triangleq \sup_{\delta x} \frac{\|\delta f\|}{\|\delta x\|},$$ 绝对误差的比值

  with the understanding that δx is infinitesimal.

  类似于某种矩阵范数/极大函数的定义

- If f is differentiable, let $J(x)$ be the Jacobian at x, i.e., $J_{ij} = \partial f_i / \partial x_j$, then $\hat{\kappa} = \|J(x)\|$, where $\|J(x)\|$ is the norm of $J(x)$ induced by the norms on X and Y.

Relative condition number

由这个式子看出相对条件数描述的是自变量的误差与因变量误差的相关程度

- The *relative condition number* $\kappa = \kappa(x)$ is defined by

$$\kappa = \lim_{\delta \to 0} \sup_{\|\delta x\| \le \delta} \left(\frac{\|\delta f\| / \|f(x)\|}{\|\delta x\| / \|x\|} \right) \triangleq \sup_{\delta x} \frac{\|\delta f\| / \|f(x)\|}{\|\delta x\| / \|x\|}$$

相对误差之比

与上一页相比之下，类似于相对误差与绝对误差

- If f is differentiable, then

$$\kappa = \frac{\|J(x)\|}{\|f(x)\| / \|x\|}.$$

- The relative condition number is arguably more useful than the absolute condition number, because floating-point arithmetic introduces relative errors rather than absolute ones.

清华大学统计学研究中心

Question

默认相对条件数

- Is a condition number of 1000 large or small? **大概相当于如果**x丢失一位有效数字那么f(x)会丢3**位有效数字**

- What is its implication on the accuracy of $f(x)$?

Example

- What is the relative condition number of $x/2$?

1

Example

- What is the relative condition number of $\sqrt{x}$?

1/2

Example

- Consider the problem of obtaining the scalar $f(x) = x_1 - x_2$ from the vector $x = (x_1, x_2)^T$, using the ∞-norm.

Example

- Consider the computation of $f(x) = \tan x$ for x near 10^{100}. A minuscule relative perturbation in x can result in arbitrarily large changes in $\tan x$, so $\tan 10^{100}$ is effectively uncomputable on most computers. We don't need to calculate the Jacobian to know the problem is ill-conditioned.

对10^100量级的浮点数其相对误差可能比较小，但是绝对误差一定特别大，远大于π，而tan周期为π，因此该问题一定是不稳定的

Example

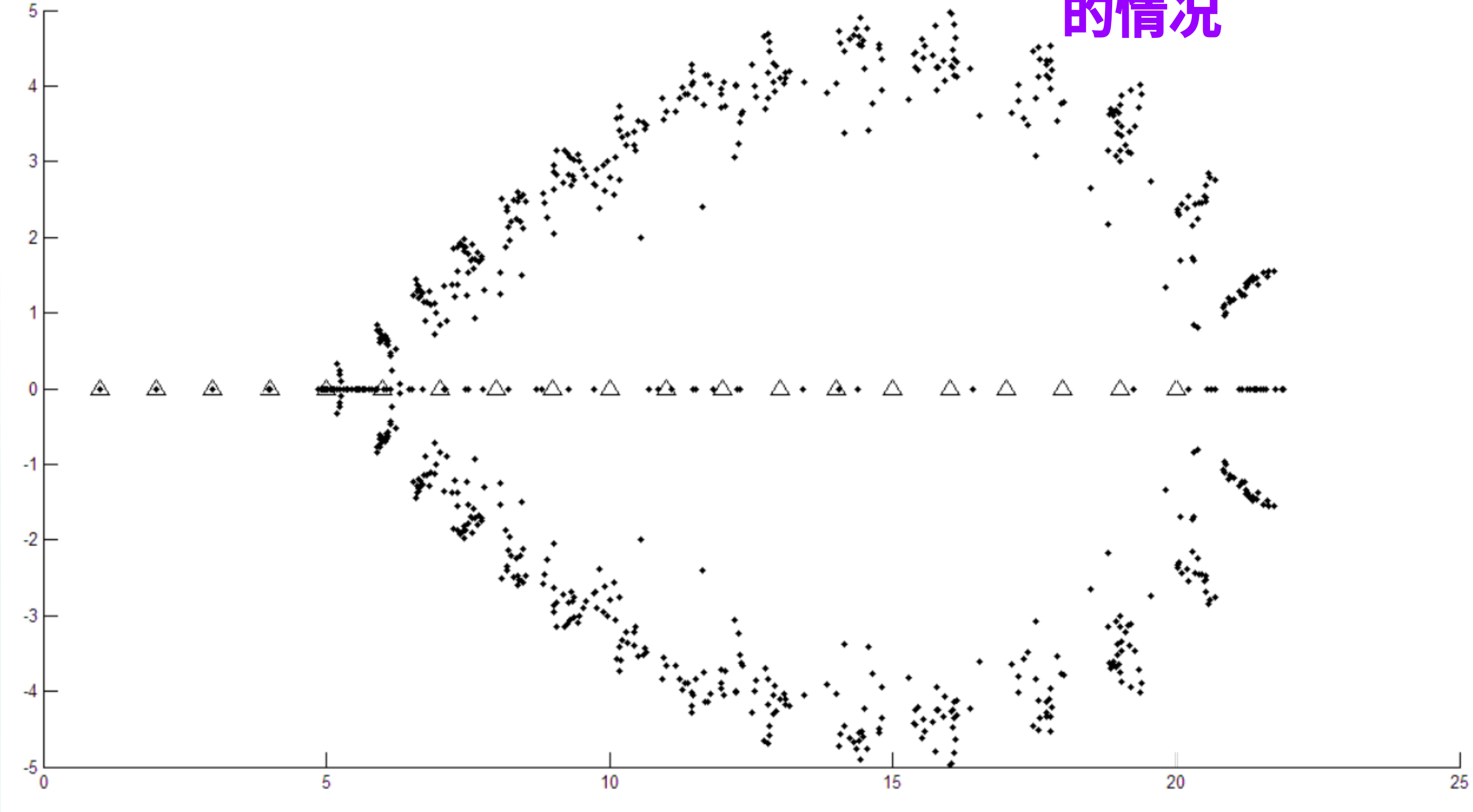
- The determination of the roots of a polynomial, given the coefficients, is a classic example of an ill-conditioned problem. Consider $x^2 - 2x + 1$, with a double root at $x = 1$. A small perturbation in the coefficients can lead to a large change in the roots; for example, $x^2 - 2x + 0.9999 = (x$

此次计算时，方程系数为自变量，而根为因变量

用求根公式易见条件数为∞

Example

散点为原方程的某些系数在极小的随机扰动下的跟**的情况**



Wilkinson's classic example of ill-conditioning: roots of $\prod_{i=1}^{20}(x-i)$

清华大学统计学研究中心

Example

- The problem of computing the eigenvalues of a nonsymmetric matrix is often ill-conditioned. Compare the eigenvalues of the two matrices

$$\begin{pmatrix} 1 & 1000 \\ 0 & 1 \end{pmatrix}, \text{and} \begin{pmatrix} 1 & 1000 \\ 0.001 & 1 \end{pmatrix}$$

- On the other hand, if a matrix is symmetric, then its eigenvalues are well-conditioned.

一般来讲非对称矩阵特征值不稳定

Condition of matrix-vector multiplication

- Fix $A \in R^{m\times m}$ and nonsingular, consider the problem of computing Ax from input x.

$$\kappa = \|A\| \frac{\|x\|}{\|Ax\|} \leq \|A\| \|A^{-1}\|.$$

- Similarly, the problem of solving $Ax = b$ given b has condition number

$$\kappa = \|A^{-1}\| \frac{\|b\|}{\|x\|} \leq \|A\| \|A^{-1}\|.$$

关于线性方程的解/向量的变换的条件数

Condition number of a matrix

- $\|A\|\|A^{-1}\|$ is a commonly used quantity and is thus defined as the condition number of A, denoted by $\kappa(A)$:
$$\kappa(A) = \|A\|\|A^{-1}\|.$$

- For L2 norm, $\|A\| = \sigma_1$, $\|A^{-1}\| = 1/\sigma_m$, where σ_1 and σ_m are the largest and smallest singular values of A. Thus,
$$\kappa(A) = \frac{\sigma_1}{\sigma_m}.$$

Condition of a system of equations

- If we hold b fixed and A is perturbed by infinitesimal δA. Show that the relative condition number is $\kappa(A)$.