Lesson 3
Average Case Analysis

Assessing Performance Through Analysis And Synthesis

Algorithms

Corazza

Wholeness of the Lesson

Average-case analysis of performance of an algorithm provides a measure of the typical running time of an algorithm. Although computation of average-case performance generally requires deeper mathematics than computation of worst-case performance, it often provides more useful information about the algorithm, especially when worst case analysis yields exaggerated estimates. Likewise, as discussed in SCI, more successful action results from a deeper dive into silence, into pure intelligence, just as, in archery, the arrow flies truer and hits its mark more consistently if it is pulled back farther on the bow.

- Example of average case analysis: What is the average case running time to find an element in an array of length n, if it is known that the element belongs to the array?

  Solution: Intuitively, need only run through half the array, on average. Therefore, average-case running time is $n/2$. Notice that under worst-case analysis, running time is n (since desired element may occur for the first time in the last slot).

- Implicit assumption is that all possible elements in the array are equally likely to occur and all elements are distinct. Now we change the problem so that the only elements that can occur are "A" and "B", and assume that "A" and "B" are equally likely to occur, with no guarantee that both actually occur in the array. What is the average case running time to find "B"?

  Note: When we say "A", "B" are *equally likely to occur*, we mean that the array has been populated so that each slot is filled by making a random selection from the set {"A","B"}. It does not mean, for example, that the array has been deliberately filled in such a way that half the slots are filled with "A" and half with "B."

Solution: Our previous analysis is no longer accurate. Worst-case analysis needs to change to ask the question How long does it take to find the element or discover it is not there, in the worst case? The answer is the same as before: n steps. For average case analysis, notice there is a 50% chance that any given slot will contain "A" and a 50% chance it will contain "B". Intuitively, we expect that "on average," no more than two slots need to be scanned to find an occurrence of "B," even though it is *possible* that "B" does not occur at all. It can be shown (see below) that the *expected number of slots* (defined precisely below) that need to be searched is < 2. So average-case running time in this case is $\mathrm{O}(2) = \mathrm{O}(1)$, i.e. constant time.

- Previous example illustrates the need for
  A. A precise way of talking about the "average" of several quantities, when the probabilities of their occurrence may vary
  B. A precise way of describing "distribution of data" and the concept of "equally likely outcomes".

- The area of mathematics that is concerned with such question is the *theory of probability*. We give a review of the basic concepts as the basis for performing average case analysis in the course.

Review of Basic Probability Theory

A. Probability is a way of measuring the likelihood of an *event* occurring on a *sample space*, which consists of all possible *outcomes* of a particular *experiment*.

B. *Example*: 5 coins are flipped in succession. We might be interested in the probability that all 5 coins come up heads.
   - The experiment is flipping the 5 coins in succession.
   - The outcomes are the different sequences of heads and tails that can come up.
   - The sample space is the set of all such sequences.
   - The event in this case is the set of all outcomes in which all 5 coins are tails (in this case, the event consists of just one possible outcome).
   - The probability in this case is computed by noticing that there are 2^5 possible outcomes, and only one of these yields "all heads". So probability $= 1/32$.

C. Formally, an *event* is a subset of the sample space S; that is, a set consisting of *some* of the outcomes in the space. For example: The event in the sample space S above in which there are exactly 4 heads and just 1 tail is the following subset of S:

$$\{\text{THHHH, HTHHH, HHTHH, HHHTH, HHHHT}\}$$

D. Formally, *probability* is a function Pr that assigns to each event (subset) E of S a real number in the interval $[0, 1]$. We may write $\Pr : \mathcal{P}(S) \to [0, 1]$. Pr must also satisfy the following conditions:

- $\Pr(\emptyset) = 0$ (The probability that there is no outcome at all is 0, i.e., "impossible".)
- $\Pr(S) = 1$ (The probability that the outcome lies in the sample space is 1, i.e., "certain".)
- If A and B are disjoint subsets of S, then $\Pr(A \cup B) = \Pr(A) + \Pr(B)$. (The probability that either A or B occurs is the sum of the probabilities that A occurs and B occurs.)

Note: For this course, a probability function Pr assigns a value to *every possible* subset of the sample space. In more general contexts, this may not be true; Pr may assign values only to the "reasonable" subsets of the space.

E. If S is finite sample space and all outcomes are equally likely, then probability can be computed by counting: For any event $A \subseteq S$:

$$\Pr(A) = \frac{|A|}{|S|}.$$

Example: In the experiment of tossing 5 coins, let A be the event "getting 5 heads"; we may write

$$A = \{(\mathrm{H}, \mathrm{H}, \mathrm{H}, \mathrm{H}, \mathrm{H})\}$$

A is a set consisting of just one outcome. Then, as before, the sample space S is all possible 5-tuples of Hs and Ts, and $|S| = 32$. So

$$\Pr(A) = \frac{|A|}{|S|} = \frac{1}{32}.$$

F. Events A and B are said to be *independent* if $\Pr(A \cap B) = \Pr(A)\Pr(B)$. Intuitively, A and B are independent if the occurrence of A does not affect whether B will occur (and conversely). It's usually assumed that in a coin-flipping experiment, "getting heads" and "getting tails" are independent events.

Example of Independent Events: Consider the experiment in which two coins are tossed in succession. Then $S = \{\text{HH, HT, TH, TT}\}$ is the sample space. Let A be the event that first coin comes up heads; let B be the event that the second coin comes up heads. Then

$$\begin{aligned}
A &= \{\text{HH}, \text{HT}\} \\
B &= \{\text{HH}, \text{TH}\} \\
A \cap B &= \{\text{HH}\} \\
\Pr(A) &= \frac{|A|}{|S|} = \frac{2}{4} = \frac{1}{2} \\
\Pr(B) &= \frac{|B|}{|S|} = \frac{2}{4} = \frac{1}{2} \\
\Pr(A \cap B) &= \frac{|A \cap B|}{|S|} = \frac{1}{4} \\
\Pr(A)\Pr(B) &= \frac{1}{2} \cdot \frac{1}{2} = \frac{1}{4}
\end{aligned}$$

G. A *random variable* is a function from the sample space to the reals; we write $X : S \to \mathbf{R}$.

- A random variable is a way of representing sample space elements as real numbers.

- *Example*:

  Experiment: roll two dice and record the values that come up.

  Sample space: S consists of all pairs (i, j) of possible die rolls, with $1 \leq i \leq 6, 1 \leq j \leq 6$

  Random variable: We can define X by $X = i+j$ (formally, $X(i, j) = i + j$).

  Probability question: What is the probability that $X = 7$? (What is $\Pr(X = 7)$?)

Answer: $\Pr(X = 7)$ is formally defined to be the probability of the event consisting of all pairs (i, j) that are mapped by X to 7. There are 6 such pairs out of a sample space of 36 pairs. Therefore

$$\Pr(X = 7) = \frac{|\{(i,j)|X(i,j) = 7\}|}{|S|} = \frac{6}{36} = \frac{1}{6}.$$

- The *space* of a random variable X is its range. Example: The space of X in previous example is $\{2, 3, \ldots, 12\}$.
- When the space of X is finite or can be written down as a (possibly infinite) sequence, X is a *discrete random variable*.

- *Examples*

  (1) Letting X = the sum of the two dice that come up (as above), X is discrete.

  (2) Experiment: flip a coin till "heads" comes up. Let Y be the number of trials needed. What is the space of Y? Is Y a discrete random variable?

Solution to (2): The space of Y is $\{1, 2, 3, 4, \ldots\}$. Therefore, Y is a discrete random variable.

- A discrete random variable X is *uniformly distributed* if, for any two values x, y in the space of X, $\Pr(X = x) = \Pr(X = y)$.

- Example: In the experiment in which 3 coins are tossed, the sample space is $S = \{(H, H, H), (H, H, T), (H, T, H), \ldots, (TTT)\}$. Outcomes are all equally likely. Define i so that $i(H) = 1$ and $i(T) = 0$. Define X on S by $X(u, v, w) = i(u) \cdot 100 + i(v) \cdot 10 + i(w)$. For instance

$$X(H, T, H) = 1 \cdot 100 + 0 \cdot 10 + 1 = 101.$$

  The space of X is $\{0, 1, 10, 11, 100, 110, 101, 111\}$. Clearly, X is uniformly distributed: $\Pr(X = i) = 1/8$, for any i in the space of X.

- Example: Consider the experiment in which two die are tossed and the sum of the values is recorded. The sample space S consists of all pairs (i, j), with $1 \leq i \leq 6$ and $1 \leq j \leq 6$ and all outcomes are equally likely. Define $X = i + j$ (that is, $X(i, j) = i + j$). Recall that

  - $\Pr(X = 7) = 1/6$, and
  - $\Pr(X = 2) = 1/36$.

  Therefore, X is *not* uniformly distributed in this case.

H. The *expected value* $E(X)$ of a random variable X is the "weighted average" of its values. If X is discrete with space R, $E(X)$ is given by

$$E(X) = \sum_{x \in R} x \Pr(X = x)$$

Example: Experiment is roll of die. Let X be value that comes up. $\Pr(X = i) = 1/6$ for $i = 1, 2, \ldots, 6$. Then

$$E(X) = \sum_{x \in R} x \Pr(X = x) = 1 \cdot \frac{1}{6} + 2 \cdot \frac{1}{6} + \ldots + 6 \cdot \frac{1}{6} = 3.5.$$

Example: A set S contains n distinct numbers. A process scans S and inserts each element one at a time into an array (removing the element from S), where the array index in each case is chosen at random (and whenever an array index occurs a second time, it is discarded). One of the elements of S is the number 0. The usual search algorithm for an array is used to determine the position of the number 0 in the array. What is the average-case running time for the algorithm? More precisely, what is the expected number of slots to check in order to find the slot that contains 0?

Solution: In this problem, we have asserted from the beginning that values in the array are distinct and randomly generated. Our intuition, that only half the array, on average, needs to be searched, is valid. Mathematically, let X be a random variable denoting the number of slots that are checked to finally locate the number 0. The space of X is therefore $\{1, 2, \ldots, n\}$. Notice that the event $X = i$ occurs if 0 is found in slot $i - 1$—since each of the possible slots in the array are equally likely to contain 0, the events $X = i, 1 \leq i \leq n$, are also all equally likely to occur. Therefore $\Pr(X = i) = \frac{1}{n}$, for each i. We have therefore:

$$\begin{aligned} E[X] &= \sum_{i=1}^{n} i \cdot \Pr(X = i) \\ &= \sum_{i=1}^{n} i \cdot \frac{1}{n} \\ &= \frac{1}{n} \cdot \left(\frac{n(n+1)}{2}\right) \\ &= \frac{n+1}{2} \end{aligned}$$

And so

$$\begin{aligned} &\text{Average-case running time to find 0} \\ &\quad = \text{Expected number of slots checked} \\ &\quad = \frac{n+1}{2} \end{aligned}$$

- The sum of two random variables X, Y on S, denoted $X + Y$, is the random variable Z on S where $Z(s) = X(s) + Y(s)$. Also, if a is a number, the random variable $U = aX$ is defined by $U(s) = a \cdot (X(s))$.

- *Linearity of Expectation*: If a, b are real numbers and X, Y are random variables, then

$$E(aX + bY) = aE(X) + bE(Y)$$

- *Example*. In experiment where two dice are rolled and sum recorded, let Z be the sum. Then $E(Z) = 7$. See this by letting X denote value of first die, Y the value of the second die. Then $Z = X + Y$. So

$$E(Z) = E(X + Y) = E(X) + E(Y) = 3.5 + 3.5 = 7.$$

Bernoulli Trials

- Consider the following experiment: Repeatedly flip a coin until you get heads. What is the expected number of trials for a given experiment?

- An experiment in which there are just two possible outcomes (which we label "success" and "failure"), and in which the outcome of one such experiment is independent of the outcome of another, is called a *Bernoulli trial*. The coin flipping experiment above is an example of repeated Bernoulli trials.

- **Solution to Coin-Flipping Problem** Let Y be a random variable denoting the number of trials required to get "heads". We have

$$\begin{aligned}
\Pr(Y = 1) &= \frac{1}{2} \\
\Pr(Y = 2) &= \frac{1}{4} \\
\dots \quad & \quad \dots \\
\Pr(Y = n) &= \frac{1}{2^n} \\
\dots \quad & \quad \dots
\end{aligned}$$

Therefore,

$$\begin{aligned}
E[Y] &= 1 \cdot \Pr(Y = 1) + 2 \cdot \Pr(Y = 2) \cdot \dots \\
&= \sum_{i=1}^{\infty} \frac{i}{2^i} \\
&= 2.
\end{aligned}$$

Optional: The Computational Trick

- Observation: Define $g(x)$ by

$$g(x) = \sum_{k=0}^{\infty} ax^k$$

Whenever $|x| < 1$, this sum is equal to

$$\frac{a}{1-x}.$$

Since the series is absolutely convergent for $|x| < 1$, the derivative can be computed by

$$g'(x) = \sum_{k=0}^{\infty} kax^{k-1}. \tag{1}$$

The derivative also equals

$$\frac{d}{dx}\frac{a}{1-x} = \frac{a}{(1-x)^2}. \tag{2}$$

The expression

$$\sum_{i=1}^{\infty} i \cdot \left(\frac{1}{2}\right)^i = \left(\frac{1}{2}\right)\sum_{i=1}^{\infty} i \cdot \left(\frac{1}{2}\right)^{i-1}$$

from previous slide has the form of (1) (the $i = 0$ case is irrelevant). Therefore by (2)

$$\sum_{i=1}^{\infty} i \cdot \left(\frac{1}{2}\right)^i = \left(\frac{1}{2}\right)\left(\frac{1}{(1-\frac{1}{2})^2}\right) = 2.$$

Expected Value of the Geometric Random Variable

Using the mathematical technique used in the coin-flipping problem, the following theorem can be proved.

Theorem (*Expected Value of the Geometric Random Variable*). Suppose we perform repeated Bernoulli trials where the probability of success is p. Let Y denote the random variable whose value is the number of trials needed to get a successful outcome. Then

$$E[Y] = \frac{1}{p}.$$

Terminology. In the theorem, if X denotes the number of trials *before the first success*, then X is called the *geometric random variable*. The random variable Y mentioned in the theorem is the trial that occurs just after the last failure, so $Y = X + 1$.

Expected Value of the Negative Binomial Random Variable

The negative binomial random variable is used to compute the number of trials needed to obtain k successes, for some integer $k > 1$, where probability of success is p. Specifically,

$$X = \#\text{failures before the } k\text{th success}$$

Using techniques similar to those for the geometric random variable, we have:

Theorem. Let X denote the negative binomial random variable for k trials, where probability of success is p. Then

$$E(X) = \frac{k(1-p)}{p}.$$

Example. Compute expected number of coin flips required to get exactly k heads.

Here, $p = 1/2$. The expected number of failures – that is, the number of "tails" – is given by

$$\begin{aligned} E(X) &= \frac{k(1-p)}{p} \\ &= k \end{aligned}$$

Since we have also obtained exactly k "heads" in this run of trials,

$$\begin{aligned} &\text{expected \# of coin flips} \\ &\qquad = (\text{expected \# heads}) + (\text{expected \# tails}) \\ &\qquad = k + k \\ &\qquad = 2k \end{aligned}$$

Optional: Chernoff Bound

Chernoff bounds are used to provide a bound to a sum of random variables. Suppose $X_1, \ldots, X_n$ are independent random variables, each with space $\{0, 1\}$. Let p_i denote probability of "success" for X_i; that is, for each i

$$p_i = \Pr(X_i = 1).$$

Notice

$$E(X_i) = 0 \cdot \Pr(X_i = 0) + 1 \cdot \Pr(X_i = 1) = p_i.$$

Let $X = \sum_{i=1}^{n} X_i$. Then $E(X) = \sum_{i=1}^{n} p_i$. Write $\mu = E(X)$. Then for any real number δ, if $0 < \delta \leq 1$, we have

$$\Pr(X < (1 - \delta)\mu) < e^{-\mu\delta^2/2}.$$

Random Permutations

First Algorithm. In the ith slot of the new array, keep trying random numbers in the range $0..n-1$ that have not yet been chosen. Make sure checking if a number is new requires only constant time.

```
Algorithm randomPerm1(n)
    Input The size n of the output array
    Output An array with elements 0, 1, 2, ..., n − 1
    randomly arranged.
output ← new Array(n)
tracking ← new Array(n)
for i ← 0 to n − 1 do
    validNumFound ← false
    while !validNumFound do
       rand ← randomInt(0,n − 1)
       if tracking[rand] = 0 then
         tracking[rand] ← 1
         validNumFound ← true
         output[i] ← rand
return output
```

Analysis of First Algorithm

- Best case: The next random integer is always new. Running time is $O(n)$. Worst case: One slot is never filled – the algorithm never terminates.

- Average case: Let Y_i be a random variable representing the number of tries required to fill the ith slot in the array, $0 \leq i \leq n-1$. Notice each Y_i counts Bernoulli trials, as in the GRV Theorem. We have:
  a. $i = 0$: $p = 1$ so $E[Y_0] = 1$
  b. $i = 1$: $p = \frac{n-1}{n}$ so $E[Y_1] = \frac{n}{n-1}$
  c. $i = 2$: $p = \frac{n-2}{n}$ so $E[Y_2] = \frac{n}{n-2}$
  d. generally: $p = \frac{n-i}{n}$ so $E[Y_i] = \frac{n}{n-i}$

  Let Y be a random variable representing the number of tries required to fill the array. Since checking whether the next random integer has already been used takes only constant time, the average-case running time of `randomPerm1` is proportional to $E[Y]$.

Computing $E[Y]$

$$
\begin{aligned}
E[Y] &= E[Y_0 + Y_1 + \ldots + Y_{n-1}] \\
&= \sum_{i=0}^{n-1} E[Y_i] \\
&= \sum_{i=1}^{n} \frac{n}{i} \\
&= n \cdot \sum_{i=1}^{n} \frac{1}{i}
\end{aligned}
$$

It follows from MathReview.pdf that

$$\textstyle\sum_{i=1}^{n} \frac{1}{i} \text{ is } \Theta(\log n).$$

Therefore,

Running time of `randomPerm1` is $\Theta(n \log n)$.

Second Algorithm. Start with the numbers $0, 1, \ldots, n-1$ in order and rearrange. At position i pick a random number r in $[i, n-1]$ and swap the value at position i with the value at position r.

```
Algorithm randomPerm2(n)
    Input The size n of the output array
    Output An array with elements 0,1,2,...,n−1
    randomly arranged.
output ← new Array(n)
fill(output) //load with 0,1,2,..., n−1.
for i ← 0 to n − 1 do
    rand ← getRandomInt(i,n-1)
    swap(output, i, rand)
return output
```

Main Point

Average case analysis makes essential use of inherently random characteristics of operation of an algorithm to determine an average-case asymptotic bound on its running time. Examples of such random characteristics include uniform distribution of data and random number generation. The laws underlying random behavior are used here to provide a precise estimate of the efficiency of algorithms. The perspective from Maharishi Vedic Science is that all expressions in the universe, however chaotic they may appear, are governed by laws of nature which are grounded in the home of all the laws of nature, the field of pure intelligence. Accessing the home of natural law through expansion of awareness makes it possible to bring order and value into any situation, however disorderly it may appear.

Unity Chart

1. Worst-case analysis of the performance of an algorithm provides an upper bound on the running time of the algorithm.

2. Sometimes, however, worst-case analysis results in a bound that is far worse than would ever actually occur in practice. For such situations, average case analysis provides a better and more useful performance analysis, indicating the typical running time of the algorithm.

3. *Transcendental Consciousness* is the field beyond all bounds, providing the silent foundation for all computation and structure.

4. *Impulses within the Transcendental field.* All the precisely crafted structure of the universe originates from the mistake-free computational dynamics occurring within the transcendent. This unmanifest performance is called by Maharishi *Vedic Mathematics*.

5. *Wholeness moving within itself.* In Unity Consciousness, the boundaries and special characteristics that distinguish objects and individuals from one another are appreciated as lively expressions of unified wholeness.