Making Multiple Copies under possibility of Failure

Jayadev Misra

4/4/97

Problem: It is required to copy a file N times; call each copy a *clone* of the original. The file consists of a sequence of symbols each of which is independently copied. Therefore, we consider the problem when the file consists of a single symbol.

Each copy operation has a probability of success of less than 1. To ensure that all N clones are correct, we could compare each clone against the original, but each comparison is likely to be as expensive as the copy operation itself. A different strategy is as follows. Call the original file the 0^{th} clone; create the $(i+1)^{th}$ clone from the i^{th}. Finally compare the N^{th} clone with the original. If they match, all clones are correct with high probability. If they do not match, apply a binary search to locate a defective clone and recreate all clones from that point. The strategy may be useful in transmitting files around a network, and in broadcasting, where the terminal nodes may send their copies to the broadcaster for checking. Intuitively, once a copy operation is faulty a subsequent copy must be faulty and *recreate the original file* for the original and last clone to match. The probability of the italicized statement is low.

We set up the following abstract version of the problem. We have an alphabet with $d+1$ symbols. A copy operation on symbol x creates x with probability p and any of the other symbols with probability $q = (1-p)/d$. A *run* is a sequence of symbols where the initial file is represented by the first symbol and each clone is a symbol. Thus, in a run two adjacent symbols are identical with probability p. A run is *perfect* if all symbols are identical. A run is *good* if the first and last symbols match. We estimate the probability of a run being perfect given only that the run is good.

For a run of length $n+1$ the probability of its being perfect is p^n since each copy operation – there are n of them – is correct with probability p. Now, we compute the probability of a run being good. Let g_n denote the probability of a run of length $n+1$ being good. We have

$$g_0 = 1$$
$$g_{n+1} = p \times g_n + q \times (1 - g_n)$$

The second equation can be understood as follows. Consider a run of length $(n+2)$. If this run is good, either (1) the first $n+1$ symbols constitute a good run (with probability g_n) and the last symbol matches the previous symbol (with probability p), or (2) the first $n+1$ symbols do not constitute a good run (with probability $1-g_n$) and the last symbol matches the first symbol of the sequence (since the last symbol differs from the previous symbol and it matches the first symbol, the probability of this copy step is q). Simplifying the last equation

$$g_0 = 1$$
$$g_{n+1} = r \times g_n + q, \text{ where } r = p - q$$

Solving, for all n, $n \geq 0$,

$$\begin{aligned} & g_n \\ = \quad & \qquad \{ \text{ expanding the recurrence and collecting terms } \} \\ & r^n + q \times \tfrac{1-r^n}{1-r} \\ = \quad & \qquad \{ \text{ from } p+qd=1 \text{ and } r=p-q, \text{ we have } 1-r=q(d+1) \ \} \\ & r^n + \tfrac{1}{d+1} \times (1-r^n) \\ = \quad & \qquad \{ \text{ simplifying } \} \\ & \tfrac{1+dr^n}{d+1} \end{aligned}$$

Then the probability that a good run of length $n+1$ is perfect is $\frac{(d+1)p^n}{1+dr^n}$. Since $p > r$, we have $p^n > r^n$, and this probability is at least $\frac{(d+1)p^n}{1+dp^n}$. If p and n are are such that p^n is close to 1, then the additional check provides a strong indication of correctness even for modest values of d. For instance, with $p^n = 0.9$ and $d = 10$, this probability exceeds 0.99. For larger values of d, the probability gets closer to 1; with $p^n, d = 0.9, 1000$, the probability is around 0.9999.