December 2007 Ponder This

Chris Shannon, Calgary Canada

December 30, 2007

1 Problem

Suppose we use the following algorithm. Select a subgroup of size .5*n at random and test it. If it passes repeat with different random subgroups of size .5*n until you find a subgroup which fails. Then recursively apply the algorithm to smaller groups until you have isolated the bad part. How many tests on average will this take?

1.1 Solution

Let n_f = the number of failed tests needed to reduce n samples to a single sample.

$n * .5n_f = 1$

so $n_f = \log_2(n)$ failures are needed.

Any given test has a .5 chance of the defective part being present and an independent .5 chance of it failing. Thus the probability of failure is .5 *.5 = .25

Number of failures / Number of tests = Probability of failure

$\frac{n_f}{n_t} = P$

Rearranging

$n_t = \frac{\log_2(n)}{.25} = 4.0000 \log_2(n)$

$C = 4.0000$

2 Problem

Use the algorithm in 1) except you test random subgroups of size $a * n$ instead of $.5 * n$. What is the optimal value of a and how many tests on average will be required?

2.1 Solution

Similar to part 1, but each failure reduces n by a, instead of .5

$na^{n_f} = 1$

take the $\log_2$ of each side

$\log_2(n) + n_f \log_2(a) = 0$

so $n_f = \frac{-\log_2(n)}{\log_2(a)}$ failures are needed.

Any given test has a chance a of the defective part being present and an independent .5 chance of it failing. Thus the probability of failure is

$P = a * .5$

The number of tests, n_t, can be found by the following ratio

$\frac{n_f}{n_t} = P$

$\frac{-\log_2(n)}{.5a\log_2(a)} = n_t$

$n_t = C\log_2(n)$where $C = \frac{-1}{.5a\log_2(a)}$

We are interested in maximizing n_t with respect to a

$\frac{d(n_t)}{da} = 0$

We can remove the constants .5 and log(2)because we are setting the derivative to 0.

$\log(a) + a/a = 0$

$a = e^{-1}$

Plugging this a into the equation above gives $C = 3.7683$ in the equation $n_t = C\log_2(n)$

3 Problem

Divide the parts at random into 2 equal subgroups and test them alternatively until one fails. Then apply the algorithm recursively. How many tests on average will be required to isolate the bad part.

3.1 Solution

Because each failure will reduce n by half, the number of failures needed is the same as part 1.

$n_f = \log_2(n)$

There are two possible cases 2 cases. The defective part will be in the first subgroup, or it will be in the second subgroup.

Case 1: the defective part is in the first subgroup. We can only find it on tests numbers 1, 3, 5, 7.... Since it will fail with probability .5, we can expect that it will take $\frac{1}{.5} = 2$tests to find it. Two tests on subgroup 1 corresponds to 3 tests total.

Case 2: the defective part is in the second subgroup. We can only find it on tests numbers 2, 4, 6, 8 .... We can expect to find it after the second test on this subgroup, which is the fourth test total.

Both cases occur with equal probability so we can expect a total of $\frac{3+4}{2} = 3.5$tests for 1 failure. Thus $n_t = 3.5n_f = 3.5\log_2(n)$, and $C = 3.5000$.

4 Problem

Same as 3) except at each stage the parts are divided into 3 equal subgroups which are tested in turn until one fails. Again how many tests on average will be needed to find the bad part?

4.1 Solution

There will be 3 cases, it will be in the first, second or third subgroup. Like in part 3, we can expect to find it on the second test of that subgroup. For the first case, the tests are 1, 4, 7, 10..., and we can expect to find it on the fourth test. Likewise for the second case, we can expect to find it on the fifth test, and for the third case, we'll find it on the sixth. All cases occur with equal probability so they average to 5 tests.

Each failed test reduces n by 1/3, so we'll have $n_f = \log_3(n)$and $n_t = 5\log_3(n) = \frac{5\log_2(n)}{\log_2 3}$, which gives a numerical value of $C = 3.1546$.

5 Problem

(Not required for credit) The algorithm in 4 is pretty good but not optimal. How many tests does the optimal algorithm require?

5.1 Solution

	Present	Not Present
failed	.5a	0
not failed	.5a	1-a
Total	a	1-a

The probabilities of the four mutual exclusive possible results of testing a subgroup of size a are tabulated. The conditional probability of the part being present given that the test did not fail is $P(p|\bar{f}) = \frac{.5a}{1-.5a}$.

These conditional probability approach doesn't seem to be helpful.

Instead lets look

a and 1-a

Case 1: it is in first subgroup (a)

expect 3 tests.

Case 2: it is in the second subgroup (1-a).

expect 4 tests.

Case 1 occurs with probability a, case 2 occurs with probability 1-a.

Case 1 reduces n by a, case 2 reduces n by 1-a.

$(n * a) * a + n * (1-a) * (1-a) = n(a^2 + (1-a)^2)$

$3 * a + 4 * (1-a) = 4 - a$.

$c = -\frac{4-a}{log_2(a^2+(1-a)^2)}$

Finding the minimum of this function yields on optimum strategy of $a = .5249$and $C = 3.4876$. This is no better than part 4 so we'll continue.

Lets try 3 subgroups, sized a, b, (1-a-b).

Case 1: expect 4 tests, occurs with probability a, reduces by a

Case 2: expect 5 tests, occurs with probability b, reduces by b

Case 3: expect 6 tests, occurs with probability 1-a-b, reduces by 1-a-b.

Expected number of tests for a failure $4a + 5b + 6(1 - a - b) = 6 - 2a - b$

n reduced after each failure$na^2 + nb^2 + n(1-a-b)^2 = n(a^2 + b^2 + (1-a-b)^2)$

$C = -\frac{6-2a-b}{\log_2(a^2+b^2+(1-a-b)^2)}$

$\bigtriangledown C = 0$

$\bigtriangledown C = (\frac{dC}{da}, \frac{dC}{db})$

$\frac{dC}{da} = -(-2 * \log_2(a^2 + b^2 + (1 - a - b)^2) - \frac{(6-2a-b)}{a^2+b^2+(1-a-b)^2}(2a + 2(1 - a - b) * (-1))) = 0$

$\frac{dC}{db} = -(-1 * \log_2(a^2 + b^2 + (1 - a - b)^2) - (6 - 2a - b)/(a^2 + b^2 + (1 - a - b)^2) * 2b + 2(1 - a - b) * (-1)) = 0$

Turns out that Maxima wasn't able to find a minimum for C, so a numerical solution was found using a spreadsheet.

$a = .37054...$

$b = \frac{1}{3}$

giving $C = 3.1313$

$a$looks suspiciously close to e^{-1}but definitely isn't equal to it.

In conclusion, the optimal strategy I found was to divide the group into three subgroups, the first of size .37054, the second of size $\frac{1}{3}$, and the third the remainder. Test each of them in turn until a failure then repeat recursively. This results in the number of tests require to be $n_t = 3.1313 \log_2(n)$.

There are other strategies to consider, such as testing the first subgroup twice, or k times, before testing the second subgroup. Or testing the subgroup in orders such as 1, 2, 1, 3, 1, 2, 1, 3... If my solution is not optimal, I'm interested in seeing optimal solution.