

HomeTask

Jingying Weng

2022/6/30

Main questions:

Question 1: Given a length 4 vector V , and two numbers $N < M$, what is the probability to find two components a and b from V that $a \geq N$ and $b \geq M$ (task 1).

- Randomly select two out of 4 abilities, check if I can fulfill the 2 requirement. randomly select one of the two to be higher and another to be lower. Is it same to select on higher and select one lower from remaining?

Possible combinations:

2-1: $\text{Choose}(4,2) = 6$ multiply 2 = 12

4-3: $\text{Choose}(4,1)$ multiply $\text{Choose}(3,1) = 12$

So we have 12 possible combinations of tasks.

- Then the probability to fulfill the tasks depends on the maximum value we have for each abilities.

For example, if we only had one $a \geq N$ and another $b \geq M$, the probability p is $1/12$. If we had k refresh chances, the probability increased to $p_k = 1 - (11/12)^k = 0.2297454$. If we had one $a \geq N$, and $b_1, b_2 \geq M$, then $p = 2/12$.

To calculate the probability, first find the count A of $a_i \geq N$. Then find the count B of $b_i \geq M$.

If $A < 2$, $p = 0$;

If $A \geq B \geq 2$, $p = (2! \text{choose}(B,2) + (A-B)*B)/12$;

If $A \geq B \geq 1$, $A \geq 2, B < 2$, $p = (A-B)*B)/12$;

Note1: $\text{choose}(0,2)$ and $\text{choose}(1,2)$ are both 0 Note2: If $A < 2$, B must < 2

So we can use $P = (2! \text{choose}(B,2) + (A-B)*B)/12$ for all conditions

Question 2 Given another length 4 vector E , only one component of E can be used to be added to the corresponding position of V to fulfill task 1. What is the probability now (task 2).

Given E , the final number of $a \geq N$ and $b \geq M$ combinations is the union of each new combinations after adding one e_i to the corresponding component v_i in V .

We can separate the final possibility calculation into following steps:

Step 1: Obtain the numbers of combinations of the set $VE_i = V + e_i$, $i=1:4$.

Step 2: Calculate the probability p_0 for the union of combinations in Step 1.

Step 3: Calculate the final p_1 as $1-(1-p_0)^{(1+\text{refresh times})}$

For step 1 and 2:

Since we only add one e_i a time, it affects one component v_i only. Thus any new combinations after adding e_i will be different from the new combinations after adding another e_i . We only need to add the number of new combinations ΔC_{ve_i} with the number of combinations C_v for the original vector V .

- Function 1: Given refresh times f , your maximum two sums of each ability as vector V , your elf maximum bonus for each ability E , what is the probability to complete task in level X of requirement R

```
P1 <- function(f = 2, R = c(20, 18), V = c(F = 18, D = 18, K = 18,
  S = 18), E = c(F = 6, D = 0, K = 6, S = 4)) {
  N <- min(R)
  M <- max(R)
  A1 <- sum(V >= N)
  B1 <- sum(V >= M)
  P0 = (choose(B1, 2) * factorial(2) + (A1 - B1) * B1)/12
  P = P0
  print(paste("Given V, ", A1, " and ", B1, " fulfill requirement ",
    N, ", ", M, ", P0 = ", round(P, 3), sep = ""))
  if (A1 >= 1) {
    # if none of them are at least N, it is impossible
    # to fulfill the requirement
    for (a in names(V)) {
      VE = V[a] + E[a]
      Vnew = V
      Vnew[a] <- VE
      A2 <- sum(Vnew >= N)
      B2 <- sum(Vnew >= M)
      P1 = (choose(B2, 2) * factorial(2) + (A2 - B2) *
        B2)/12
      deltaP = P1 - P0
      P = P + deltaP
      print(paste("Adding E to position ", a, ", ", A2,
        " and ", B2, " fulfill requirement, ", round(deltaP,
          3), " more likely now", sep = ""))
    }
    P2 <- 1 - (1 - P)^(f + 1)
    print(paste("With refresh times ", f, ", P0 increased from ",
      round(P, 3), " to ", round(P2, 3), sep = ""))
  } else {print("No possible combinations to fulfill the requirement")}
}
## test
P1()
```

```
## [1] "Given V, 4 and 0 fulfill requirement 18, 20, P0 = 0"
## [1] "Adding E to position F, 4 and 1 fulfill requirement, 0.25 more likely now"
## [1] "Adding E to position D, 4 and 0 fulfill requirement, 0 more likely now"
## [1] "Adding E to position K, 4 and 1 fulfill requirement, 0.25 more likely now"
## [1] "Adding E to position S, 4 and 1 fulfill requirement, 0.25 more likely now"
## [1] "With refresh times 2, P0 increased from 0.75 to 0.984"
```

```
P1(f = 0, V = c(F = 18, D = 18, K = 16, S = 16), E = c(F = 0,
  D = 0, K = 0, S = 0))
```

```
## [1] "Given V, 2 and 0 fulfill requirement 18, 20, P0 = 0"
## [1] "Adding E to position F, 2 and 0 fulfill requirement, 0 more likely now"
## [1] "Adding E to position D, 2 and 0 fulfill requirement, 0 more likely now"
## [1] "Adding E to position K, 2 and 0 fulfill requirement, 0 more likely now"
## [1] "Adding E to position S, 2 and 0 fulfill requirement, 0 more likely now"
## [1] "With refresh times 0, P0 increased from 0 to 0"
```

```
P1(f = 0, V = c(F = 18, D = 18, K = 16, S = 16), E = c(F = 6,
  D = 0, K = 0, S = 0))
```

```
## [1] "Given V, 2 and 0 fulfill requirement 18, 20, P0 = 0"
## [1] "Adding E to position F, 2 and 1 fulfill requirement, 0.083 more likely now"
## [1] "Adding E to position D, 2 and 0 fulfill requirement, 0 more likely now"
## [1] "Adding E to position K, 2 and 0 fulfill requirement, 0 more likely now"
## [1] "Adding E to position S, 2 and 0 fulfill requirement, 0 more likely now"
## [1] "With refresh times 0, P0 increased from 0.083 to 0.083"
```

```
P1(f = 0, V = c(F = 18, D = 18, K = 16, S = 16), E = c(F = 6,
  D = 0, K = 6, S = 0))
```

```
## [1] "Given V, 2 and 0 fulfill requirement 18, 20, P0 = 0"
## [1] "Adding E to position F, 2 and 1 fulfill requirement, 0.083 more likely now"
## [1] "Adding E to position D, 2 and 0 fulfill requirement, 0 more likely now"
## [1] "Adding E to position K, 3 and 1 fulfill requirement, 0.167 more likely now"
## [1] "Adding E to position S, 2 and 0 fulfill requirement, 0 more likely now"
## [1] "With refresh times 0, P0 increased from 0.25 to 0.25"
```

Question 3: Given a list of 4 vectors V0, get the max sum for each vector as the vector V for Question 1. Each sum can used up to 2 components.

Question 4: Given V0 and E, get the max sum of each vector as sum of the V0 vector and the corresponding E component, can only use up to 1 component in E and up to total of 4 component in V0.

Question 5: Given V0 and E, what is the minimum value to be add to which component in V0 or E to fulfill a certain probability in task 2.

Background

Daily task list could have several S+ tasks, maximum hometown level requires 20 and 18 points of randomly selected two out of four abilities to complete S+ task. Original S rank Valkyries contribute 8, 10, 12 points in rank S, SS and SSS. Original A rank Valkyries contribute points in rank A to SSS, and original B rank Valkyries contribute points from rank B to SSS. Free elves Jingwei's Wing contributes 1-4 Dexterity points for rank A to SSS and Blood embrace gives 1-4 points in Fitness.

```
## Warning: package 'knitr' was built under R version 4.0.5
```

Same Valkyries usually have same ability except several newly added ones. Free and paid elves also had abilities.