universidade de aveiro

2020/2021

Advanced Algorithms

Approximate Hit Count

Hypothesis B - Random Strings

Diogo Andrade 89265

10 January 2020

Index

Problem

Using a large number of counters to track the number of occurrences of many different events causes large amounts of data. Which is a problem, as there is a need for fast and efficient memory processing.

To solve this problem, I developed a class Counter who allows us to count the number of occurrences of characters in random strings. This class has three methods: an exact counter, fixed probability counter and logarithmic counter.

The goal of this work is to answer the next question: “Is it possible to use a small counter to keep approximate counts of large numbers?”.

Execution explanation

Usage:

```
main.py [-h] [-a ALPHABET] [-s CHAIN_SIZE] [-t TRIALS] [-p PROB]
        [-b BASE_LOG] [-w]

optional arguments:
  -a ALPHABET      Chain characters, default='diogoandrélopesandrade'
  -s CHAIN_SIZE    Chain size, default=100
  -t TRIALS        Number of test repetitions, default=1000
  -p PROB          Probability of counter, default=1/16
  -b BASE_LOG      Base of logarithm of counter, default=sqrt(2)
  -w               Show Results, default: Write on file
```

Execution:

```
python3 main.py
```

Result: (complete results on file report_100_1000.txt)

```
Chain size 100 - 1000 trials
Chain:
slndodnrdgioddresrgrlleneaoadaddiddrdaonroddeldosadgdganpdaaloodorpdéo
aisnsaadanoésddoesldnnpdodrdno

Letter: s
   Exact value: 7

   Prob 1/16 Counter:
       Expected Value: 0
       Expected Variance: 0.41
       Expected Standard Deviation: 0.64
       Max Absolute Error: 4
       Min Absolute Error: 0
       Mean Absolute Error: 1.2

       Max Relative Error: 200.0%
       Min Relative Error: 200.0%
       Mean Relative Error: 200.0%

       Largest Counter Value: 4
       Smaller Counter Value: 0

       Mean Counter Value: 1.2
       Mean Absolute Deviation: 0.33
       Standard Deviation: 0.44
       Maximum Deviation: 2.8
       Variance: 0.2

       counter value:          1 -        298 times
       counter value:          2 -         64 times
       counter value:          3 -          3 times
       counter value:          4 -          1 times

   Log with base sqtr(2) Counter:
       Expected Value: 5
       Expected Variance: 1.45
       Expected Standard Deviation: 1.2

       Max Absolute Error: 5
       Min Absolute Error: 0
       Mean Absolute Error: 1.28

       Max Relative Error: 200.0%
       Min Relative Error: 0.0%
       Mean Relative Error: 31.19%

       Largest Counter Value: 7
       Smaller Counter Value: 0

       Mean Counter Value: 3.77
       Mean Absolute Deviation: 0.7
       Standard Deviation: 0.87
       Maximum Deviation: 3.77
       Variance: 0.76

       counter value:          2 -         55 times
       counter value:          3 -        323 times
       counter value:          4 -        439 times
       counter value:          5 -        162 times
       counter value:          6 -         20 times
       counter value:          7 -          1 times

Letter: l
   …
```

Probability Counter

The probability counter runs through each character in the string and increments if the random number generated is less than the probability. This probability is fixed.

Probability: 1/16

Test Results

Chain size 100 – 1000 trials

```
'Expected Value': '5'
'Expected Standard Deviation': '0.67'     -     'Standard Deviation': '0.49'
'Expected Variance': '0.53'               -     'Variance': '0.31'
'Mean Absolute Deviation': '0.35'
'Mean Absolute Error': '0.9'
'Mean Relative Error': '138.35%'
```

Chain size 215 – 1000 trials

```
'Expected Value': '13'
'Expected Standard Deviation': '0.98'     -     'Standard Deviation': '0.82'
'Expected Variance': '1.05'               -     'Variance': '0.78'
'Mean Absolute Deviation': '0.66'
'Mean Absolute Error': '0.74'
'Mean Relative Error': '61.09%'
```

Chain size 1000 – 1000 trials

```
'Expected Value': '63'
'Expected Standard Deviation': '2.15'     -     'Standard Deviation': '2.09'
'Expected Variance': '4.88'               -     'Variance': '4.69'
'Mean Absolute Deviation': '1.66'
'Mean Absolute Error': '1.64'
'Mean Relative Error': '35.44%'
```

Chain size 10000 – 1000 trials

```
'Expected Value': '626'
'Expected Standard Deviation': '6.75'     -     'Standard Deviation': '6.98'
'Expected Variance': '48.83'              -     'Variance': '52.49'
'Mean Absolute Deviation': '5.44'
'Mean Absolute Error': '5.44',
'Mean Relative Error': '12.24%'
```

Chain size 100000 – 1000 trials

```
'Expected Value': '6250'
'Expected Standard Deviation': '21.37'    -     'Standard Deviation': '27.12'
'Expected Variance': '488.28'             -     'Variance': '835.25'
'Mean Absolute Deviation': '17.5'
'Mean Absolute Error': '17.5'
'Mean Relative Error': '3.91%'
```

Logarithm Counter

The logarithm counter works the same way as a probabilistic counter, but the probability is not fixed, the probability decreases as the counter value increases.

The probability value is 1 divided by the base value of the logarithm raised to the character counter value.

Logarithm base: sqrt(2)

Test Results

Chain size 100 – 1000 trials

```
'Expected Value': '66'
'Expected Standard Deviation': '1.27'        -      'Standard Deviation': '0.79'
'Expected Variance': '1.88'                  -      'Variance': '0.72'
'Mean Absolute Deviation': '0.62'
'Mean Absolute Error': '2.38'
'Mean Relative Error': '38.01%'
```

Chain size 215 – 1000 trials

```
'Expected Value': '113'
'Expected Standard Deviation': '1.85'        -      'Standard Deviation': '1.03'
'Expected Variance': '3.71'                  -      'Variance': '1.07'
'Mean Absolute Deviation': '0.81'
'Mean Absolute Error': '5.52'
'Mean Relative Error': '58.17%'
```

Chain size 1000 – 1000 trials

```
'Expected Value': '681'
'Expected Standard Deviation': '4.04'        -      'Standard Deviation': '1.19'
'Expected Variance': '17.26'                 -      'Variance': '1.43'
'Mean Absolute Deviation': '0.94'
'Mean Absolute Error': '47.01'
'Mean Relative Error': '128.08%'
```

Chain size 10000 – 1000 trials

```
'Expected Value': '6455'
'Expected Standard Deviation': '12.69'       -      'Standard Deviation': '1.31'
'Expected Variance': '172.59'                -      'Variance': '1.73'
'Mean Absolute Deviation': '0.97'
'Mean Absolute Error': '521.71'
'Mean Relative Error': '172.03%'
```

Chain size 100000 – 1000 trials

```
'Expected Value': '61097'
'Expected Standard Deviation': '40.17'       -      'Standard Deviation': '1.42'
'Expected Variance': '1725.89'               -      'Variance': '2.01'
'Mean Absolute Deviation': '1.02'
'Mean Absolute Error': '5068.58'
'Mean Relative Error': '182.27%'
```

Conclusion

Comparing the algorithms, for strings up to size 200 the logarithmic counter has better results as we can see by the mean error, but after that value and up to 220 the counters have more or less the same reliability, with a tendency for the fixed probability counter to become the most reliable.

As evidenced by the results above, we can see that with the increase in the size of the string the mean error of the probabilistic counter decreases and that of the logarithmic counter increases.

To answer the question presented in this work, that is, if it is possible to use small counters to count large numbers, taking into account the conclusions presented, the probabilistic counter is the most reliable. This counter with strings of size approximately 100000, has a relative error less than 5%, which is quite reliable.

Along with this report, files with test data were sent for each chain size. Analyzing those files, we can see that the probabilistic counter the greater range of variation of the counter value. What can be a problem, however, despite this greater variation, these values farther from the expected occur very few times.

In conclusion, it is feasible to use fixed probability counters to make approximate counts of large numbers.