Università degli studi di Genova

DIBRIS

Informatics, Bioengineering
Systems Engineering, Robotics and System Engineering

Research Track

Research track II assignment 1
Statistic

Student:
Fabio Guelfi,
Samuele Viola

Abstract

This paper concerns a statistical analysis to compare the performance of two different algorithms used in the same execution environment.
Different measurements of execution times and successes were collected for each algorithm by keeping the same token position or changing the position of the targets, so Student's t test and Z test statistics were used to evaluate the data. The results indicate that the two algorithms have significant differences in execution times and success rates, suggesting that one of the algorithms is more efficient and reliable than the other. These results provide a solid basis for further optimisation and improvement of the analysed algorithms.

Introduction

This report aims to compare and contrast two different algorithms used in the first assignment, evaluating their efficacy in two different environments: one where tokens are placed in a circle with constant ray, and then one where ray varies randomly at each test.

To ensure a robust analysis, we will consider two primary performance evaluators:

Average Time to Completion: This metric measures the speed of each implementation by recording the time required to finish the task across multiple trials. A lower average time indicates a faster solution.

Success Rate: This evaluator tracks the number of successful completions versus failures. A higher success rate reflects the reliability and robustness of the implementation in handling the given task.

We will do a statistical analysis of both the algorithms in order to evaluate these two characteristics. In the following sections, we will detail the statistical methodology, present the results of our analysis, and discuss the implications of these findings.

Algorithm implementation

In this statistical analysis, we are comparing the results of the application of two different algorithms to the same problem, here there is a short introduction to both algorithms:

Algorithm A

This algorithm has been developed by Fabio Guelfi.
Initially, the algorithm performs a 360° rotation to detect all tokens in the environment, and then chooses the location of the first one detected as the collection

point for all others.
When it detects no obstacles in the vicinity, on the way to the targets or back to the base point, it has the ability to accelerate to complete the task faster.
When all initially detected tokens have been collected, the robot performs another rotation on itself to ensure that no tokens have been forgotten. If so, it terminates its execution; if not, it takes the forgotten tokens back to the base and re-runs the check until all tokens are in the collection point.

Algorithm B

This algorithm has been developed by Samuele Viola.
It places all the tokens in the middle of the arena, and starts with the assumption that tokens are placed in circle: this thing will imply, as we will see later, a limitation when environment changes. First of all, robot goes to take the first token, then goes through the most distant token, stops in the middle and release the token. Then, for each remaining token, it goes through it and, when grabbed, rotate of 180° and goes to the stack of "already grabbed tokens", and release it. When it leaves the last token, it goes back and make a rotation of 90°.

Methods

For evaluate the two algorithms, we had to do a statistical analysis, first evaluating speed in the environment in which the algorithms has been developed, i.e. with the six tokens placed in circle, then in a different configuration, where tokens are, always place in circle, but with the ray that assumes random values each time. We measured the time to succeed for both the configurations: in the second configuration we observed that algorithm B failed sometimes in doing its stuff; because of this we decided to evaluate also the success/failure rate.

Hypothesis

In order to evaluate well both algorithms, we made some hypothesis and, by analyzing some data we decided if they're true or not. The hypothesis we made are the following:

- H_0^1: The two algorithms have the same speed (they ends their task with a similar time), so they are efficient in the same way, when the ray of the circle is the same for every attempt;
- H_0^2: The two algorithms have, in a probabilistic way, the same speed, so they are efficient in the same way, when the ray of the circle varies randomly at every attempt;
- H_0^3: Both the algorithms are good and never fail even if the ray is random.

Data Collection

The first step of our test was to collect a sufficient number of data in order to have a good statistic, so by creating a .sh file that runs automatically the program when it ends, and by modifying the script in order to save to a logfile the time of each execution (and the number of failures), we took for each algorithm 30 time measures when the ray was constant and other 30 when it varies to guarantee a Gaussian distribution of the population. Obviously all the data collected are independent and are executed in the same computer in order to not make the different computer performances impact on the statistic.

Test 1 - Speed Test

To analyze the time measurements for each algorithm under both configurations (constant and varying radius), we used the paired t-test method: it compares the means of two related groups to determine whether there is a statistically significant difference between these means.

The step for analyze data using paired t-test are:

- calculate the difference between paired observations for each configuration;
- compute the mean of the differences:

$$\bar{d} = \frac{1}{n}\sum_{i=1}^{n} d_i$$

where n is the number of paired observations (in this case, 30);

- calculate the standard deviation of the differences:

$$s_d = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(d_i - \bar{d})^2}$$

- determine the t-value:

$$t = \frac{\bar{d}}{s_d/\sqrt{n}}$$

- compare the calculated t-statistic to the critical t-value from the t-distribution table at the desired significance level (in our case $\alpha = 0.001$) with $n-1$ degrees of freedom to determine whether to reject the null hypothesis.

Hypothesis Testing For our analysis, we have the following hypotheses:

- H_0^1 : The two algorithms have the same speed (no significant difference in means) when the radius is constant.

- H_0^2 : The two algorithms have, in a probabilistic way, the same speed when the radius varies randomly.

Decision Rule If the calculated t-statistic exceeds the critical t-value for the chosen α and for the number of degrees of freedom, we reject the null hypothesis in favor of the alternative hypothesis (so there is a significant difference in the performance of the two algorithms).

Test 2 - Reliability Analysis

The ratio test, the z-score is a widely used statistical method to evaluate and compare the performance of two algorithms. Through this test, it is possible to determine whether the observed difference in success rates between the two algorithms is statistically significant, providing a solid basis for drawing conclusions on the relative effectiveness of each algorithm.

The ratio test finds application in a variety of contexts, including machine learning, natural language processing, data analysis and scientific experimentation due to its effectiveness for binary categorical data (such as success and failure).

It was for this reason that we devised this test to analyse the success and failure data of the two algorithms when varying the positions of the tokens in the environment, always maintaining their position in a circle of random radius.

Hypothesis

For this analysis we have the following assumption:

- H_0 : The two algorithms have the same reliability in the random radius environment

- H_a : One of the algorithms has a better reliability than the other one

Computation

For each algorithm we compute the successes proportion:

$p_i = x_i \;/\; n_i$

where x_i corresponds to the successes of the i algorithm and
n_i to the number of samples

Then we proceed to compute the statistic of the z-score that represents the distance between the observed proportion for an algorithm and the average expected proportion, measured in standard deviation units, using the following formulas:

standard error: SE = $sqrt(p(1-p)*(1/n_1+1/n_2))$

z-score: z = $(p_1 - p_2)/$ SE

As last step is important to use the z-score value computed to find the correspondent value in the z table to get the percentage of the values with Gaussian distribution that reject the hypothesis H_0

Result

A statistical analysis was conducted to compare the performance of two algorithms, Algorithm A and Algorithm B, in grouping a certain number of tokens at the same point within a simulation environment. The evaluation was performed under two conditions:

- Fixed environment
- Environment with a random radius circle

Additionally, a success and failure analysis was performed for the environment with a random radius circle.

Performance in the Fixed Environment

To evaluate the two performances a paired t-test was used.
As we expected, the result show that the Algorithm A is significantly more efficient than the Algorithm B
In fact we obtained:

- 30 measurements for each algorithm
- same velocity for the two robots (as precondition)
- mean of the difference = 34.84
- deviation standard of the difference = 2.12
- level of significance = 0.01
- t = 89,8363

The t value is very much bigger of the t - table value so the null hypothesis is rejected. We proved that the algorithm A is much more faster and efficient than the algorithm B in the basic conditions of the environment.

Environment with a random radius circle

To evaluate the two performances, as the previous test, a paired t-test was used. The result show that the Algorithm A is significantly more efficient than the Algorithm B also in an environment with the tokens placed always on top of a circle with a random value between 0.9 e 2.4 metres
In fact we obtained:

- 23 measurements for each algorithm
- same velocity for the two robots (as precondition)
- mean of the difference = 32,60
- deviation standard of the difference = 11,78
- level of significance = 0.01
- t = 13.27

The t value is very much bigger of the t - table value so the null hypothesis is rejected. We proved that the algorithm A is much more faster and efficient than the algorithm B also in a continuously changing environment with the condition that all the tokens has to placed each other generating a circle.

Success and failure analysis

The analysis of the number of successes and failures was conducted using the z-score test in the environment with a random size of the radius of the circle in which the tokens are placed.
Algorithm A achieved a 100% success rate, while Algorithm B had a lower success rate.
The statistical details are as follows:

- 30 measurements for each algorithm
- same velocity for the two robots (as precondition)

- 100% success rate for algorithm A

- 76,6% success rate for algorithm B

- combined proportion = 0.88
- z-score = 2.8150

- percentage on the table = 99.74%

This z-score indicates a significant difference between the two algorithms in terms of success rates, with Algorithm A demonstrating significantly better performance.

Conclusions

Through this statistic analysis, we were able to see that Algorithm A had better performance, but also higher reliability than Algorithm B. These statistical results are very useful for understanding which of the two algorithms may be the best solution for a company, and thus which solution to adopt for software development between the two possible choices.