

**A programmer utility evaluation model based on
the number of operations of the Linux stable
library**

1. Problem analysis

This model belongs to a link in the project. The overall project is based on the programmer's technical strength to analyze the code, and the establishment of this model is used to analyze the strength of the programmer.

2. Model assumption:

- 1、 Assuming that the data in this model is true and effective.
- 2、 For this problem, ignore other irrelevant factors, that is, only use the collected data to analyze the utility of the programmer.

3. Symbol description

U	Utility value
V	Utility value determination
W	Weights
C_i	Consistency index
R_i	Random consistency index
C_R	Test coefficient

4. Model establishment and solution

4.1 Definition of benefits

To judge the contribution and technical strength of a programmer by the number of operations of linux stable, it is necessary to quote the concept of utility value. Judging the skill level of programmers based on the amount of operations often upholds the principle of "Utility" maximization. We define the utility of

programmer i under the contribution of linux stable as U_i , U_i is called the "utility value" of programmer i 's contribution. This programmer has a higher technical definition than another programmer:

$$U_m > U_n, m \neq n, m \in A, n \in A$$

According to the theory of random utility, the utility value is a random variable, which is composed of a fixed term V_i and a random term ε_i

$$U_i = V_i + \varepsilon_i$$

The fixed term V_i usually takes the form of a linear utility function

$$V_i = \sum_{k=1}^n \theta_k X_{ik}, \quad (i \in A)$$

In this formula, X_{ik} is the value of the k -th characteristic variable of programmer i , and θ_k is the unknown parameter corresponding to the k -th variable. According to the weight corresponding to each characteristic variable value, the corresponding parameter θ_k can be obtained. Combining the parameters of each indicator and the internal relationship between the indicators and applying the method of multi-objective decision-making, the total utility U_i of these indicators can be obtained. Here, we select the number of commit operations, the number of fix operations, the number of types of fix operations, the number of merge operations, and the number of merge operations of the programmer to the linux stable library as five characteristic variables to measure the effectiveness of a programmer.

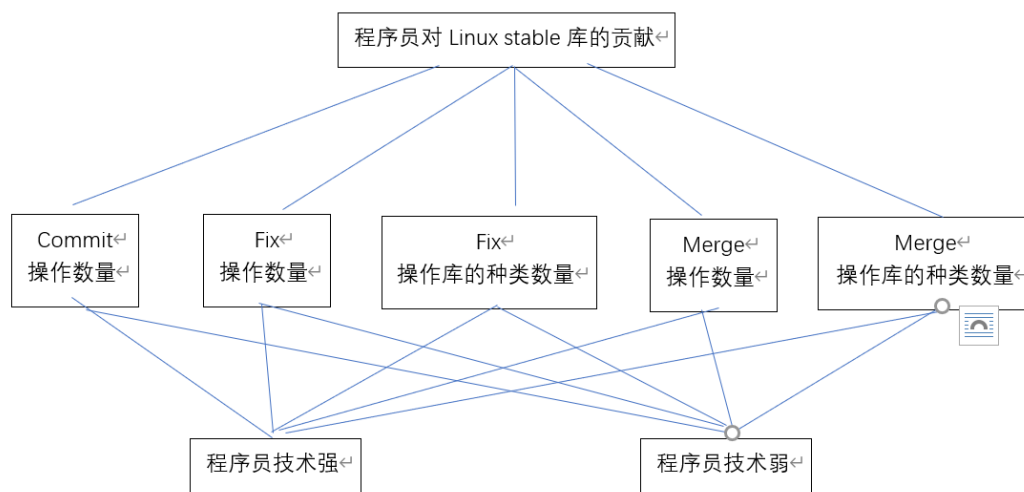
Among them, the number of commit operations, the number of fix operations, the number of types of fix operation libraries, the number of merges,

and the number of types of merge operations take the actual value of the changes made by the programmer as the value of the characteristic variable.

4.2 Solution of utility function

The solution of the utility function is divided into two parts: the solution of the five characteristic variables and the determination of the corresponding coefficient. The data required to calculate the characteristic variable of programmer i can be obtained by data crawling on linux stable, and the value of the characteristic variable is substituted. The coefficients corresponding to the characteristic variables can be obtained by AHP.

Apply AHP to determine the coefficient:



The highest level is the target level, which is evaluating the programmer's contribution to linux stable.

The middle layer is the criterion layer, that is, five indicators to measure the operation of the programmer.

The bottom layer is the program layer, that is, the strength analysis of the programmer.

Since there are mainly four items for programmers, an evaluation matrix is made for the five indicators of the number of commit operations, the number of fix operations, the number of types of fix operation libraries, the number of merges, and the number of types of merge operations. Through the survey, the number of commit operations, the number of fix operations, the number of types of fix operation libraries, the number of merges, and the number of types of merge operations are important in the minds of experts, and the evaluation metrics of the evaluation matrix are given:

Factor i withx factor j	x
Equally important	1
Slightly important	3
Stronger important	5
Strongly important	7
Extremely important	8
The median value of two adjacent judgments	2, 4, 6, 8

After the evaluation matrix of the four indicators is obtained, the weights of the four indicators when determining the transportation mode are determined by pairwise comparison, and the following comparison moments are obtained (from top to bottom, from left to right, the indicators are the number of commit operations, the fix operation Number, number of types of fix operation library,

number of merge, number of types of merge operation)

1,	4,	5,	7,	8
1/4,	1,	2,	5,	8
1/5,	1/2,	1,	7,	4
1/7,	1/5,	1/7,	1,	2
1/8,	1/4	1/8,	1/2,	1

Using the sum product method to calculate the weight W is:

$$W=(0.0297,0.0669,0.1011,0.3423,0.4599)^T$$

Perform consistency check, the largest feature vector:

$$\lambda_{max} = 6.0600$$

Calculated from this C_i, R_i :

$$C_i = 0.1.27 \quad R_i = 1.120$$

Then:

$$C_R = 0.091696 < 0.10$$

Since $CR < 0.1$, the inconsistency of the passenger survey is acceptable, and the obtained weight coefficient can be used.

In summary: $W=(0.0297,0.0669,0.1011,0.3423,0.4599)^T$

4.3 Solution of utility value

It can be obtained through the above two parts of analysis that the five index values are the number of commit operations, the number of fix operations, the number of types of fix operations libraries, the number of merges, and the number of types of merge operations using actual data obtained through the python crawler technology.

The weight coefficient uses the data calculated by the second part using the

analytic hierarchy process, and then is substituted into the formula of the utility function to obtain the final utility value.

The utility value will be given through the CSV table, so far all the utility quantification is completed, that is, each programmer will quantify the contribution of the linux stable library.