A PRACTICAL WAY OF COMPARISON OF PROGRAMS BASED ON PATTERN RECOGNITION

A. F. GUBKIN

197022 Ленинград ул. проф. Попова, д. 5
ЛЭТИ им. В. И. Ульянова (Ленина)
Кафедра Математического Обеспечения и применения ЭВМ

(Received May 5, 1982)

The comparison of programs from the viewpoint of their efficiency becomes more and more important as software systems are complicated and their scale is increased. The comparison of programs may be used in applied software systems if the systems are developed making use of the integral principles. There is integration on the depth and integration on the width.

Integration on the depth means that every stage of the problem solution is executed by the software system. For example, an aid design system must comprehend all stages of design and must transfere the information stage by stage if this principle is used.

Integration on the width means involving different programs on one stage of the computation. The programs that named alternative programs decide the same tasks but the efficiency of the solution is different for each program. The set of alternative programs allows us to increase the software system efficiency if the best program for every example of input data is applied. The software system must qualify the program that is to be executed on the next example of the input data. In order to make this possible it is necessary to develope the practical methods of comparison of programs.

The essence of the problem comes as follows. Let it be given n alternative programs. The input data and the results of these programs are similar. But these programs can realize different methods and different algorithms. Moreover the difference among the programs may mean different programming of the same algorithm. This is the reason why these programs can have different efficiency for the same examples of the input data.

Therefore it may occur that one program is more efficient for one set of examples of the input data and the other program is more efficient for other set of examples of the input data. In other words the relationship between the sets of input data values and the programs may be established and the problem of the comparison of programs reduces to the definition of the set of input data values for every program. In order to define the set of input data values it is necessary to define the decision rule that separates the input data values.

The set of the variables $X_1, \ldots, X_k$ are introduced to delineate the input data. The values of these variables must characterize the input data complication and further they are called determinant parameters. The determinant parameters are chosen in accordance with the essence of the tasks that are decided by the programs. For example, if sorting algorithms are considered the number of the array elements and the value that characterizes the order of the elements in the array may be chosen as the determinant parameters.

The efficiency of the program execution or the results that the program derives are characterized by the efficiency evaluation T_i for every program i.

The nature of the determinant parameters and efficiency evaluation has a probability character. Indeed, the determinant parameters characterize the input data stream that comes in to the programs. Input data in turn are defined by the application of the programs. Therefore the values of the determinant parameters are diverse for each example of input data. It may be assumed that variables $X_1, \ldots, X_k$ have a random value for the stream of the input data.

Further the set of variables $X_1, \ldots, X_k$ is considered as the system of the random variables and $X = (X_1, \ldots, X_k)$ is the multidimensional random variable.

The efficiency evaluation T_i for every program may be considered as a random variable too. From that follows that the vector of value of determinant parameters $x = (x_1, \ldots, x_k)$ defines not one example of the input data but some set of the examples. The set of examples of input data corresponds to the point in the space of determinant parameters. The values of efficiency evaluation will be different for different examples of input data and these values are distributed in some interval. Also the existence of some distribution function for every random variable T_i concerned with the program i may be allowed.

The consideration above shows that the program i is characterized by the system of the random variables $(T_i, X_1, \ldots, X_k)$.

The application requirements of the programs defines the choice of the efficiency evaluation. The choice of determinant parameters is performed in a more free manner. The variables choosen as the determinant parameters must satisfy the following requirements. First, dependence must exist between every determinant parameter X_j and the efficiency evaluation T_i. Secondly it is desirable that determinant parameters should be independent among themselves. Finally the values of determinant parameters must be calculated simply. It must be emphasized that determinant parameters are choosen unformally and their choice is defined by the essence of the task decided by alternative programs.

The problem of comparison of programs is considered in the next works. In [1] the approach based on the evaluation of the characteristics of the conditional random variable T_i/X are introduced. The estimation of the expectation of the random variable T_i/X as a function of the determinant parameters is found by means of the statistic experiment with the programs. The best program is defined by the comparison of the values of functions for every program.

However this way has some shortcomings. On the one hand this approach demands large capacity calculations and in many practical cases it cannot be performed. On the other hand the separation of the programs may be satisfactory only if the characteristics of the random variables T_i/X, $i = \overline{1, n}$ are sufficiently distinguished. The reliability of the separation of the programs is lower if these characteristics are alike.

In [2] the pattern recognition methods based on the training or learning procedure [4] are suggested to use. These methods suppose that the training pattern set exists and the information about classes may be derived from this set. In order to get the training pattern set in the case of the programs it is necessary to apply every alternative program for every example from some set of examples input data. In this way the values of the determinant parameters and the efficiency evaluation are defined for the examples. This information allows to separate the set of the examples on the subsets. Every subset corresponds to the program which more efficient for these examples than others. In the terms of [4] the subset of examples is named as the class, the pattern vector of the determinant parameters corresponds to the pattern and the set of examples with information about the values of the determinant parameters and about precedence of programs for every example is the training set. Furthermore the methods of the pattern recognition with a teacher may be used.

The main shortcoming of the two approaches above consists in the need to carry out a large capacity of the calculation. Also it must be emphasized that both the definition of the statistic characteristics and the definition of the training pattern set must be derived from the real examples of input data. In many practical cases the runtime of programs is very large on these examples and it does not allow us to use the approaches practically.

The approach suggested in this paper allows to a certain extent to avoid the difficulties mentioned above and therefore may be used practically. The approach is based on pattern recognition methods known as learning without a teacher.

As in the case of learning with a teacher the patterns are concerned with the vector of the deteminant parameters $X = (X_1, \ldots, X_k)$. The single pattern vector of the determinant parameters corresponds to the example of the input data. But the set of examples of the input data corresponds to the pattern vector. This circumstance is significant when the problem definition is qualified.

There are the deterministic and probable problem definitions in the pattern recognition. The pattern vector of the determinant parameters belongs only to one class in the case of the deterministic problem definition. The probable problem definition allows the pattern vector to belong to the different classes with a certain probability.

The problem definition is probable in the case of the comparison of the programs because the set of examples of the input data is concerned with the point in the space of determinant parameters and the different programs may be efficient for the different examples of the set.

The qualitative consideration above shows that on the one hand the determinant parameters, the efficiency evaluations have a statistic character, the problem definition in the pattern recognition terms must be probable and on the other hand it is desirable to separate the examples of the input data on the programs simultaneously with the useful work that is the creation of the training pattern set must be excluded.

The methods that satisfy these demands and are more appropriate are introduced in [3]. The essence of the approach to the comparison of programs based on learning without a teacher methods comes as follows. There is a stream of examples of input data. Previously the decision rule is intended. The one from the number of alternative programs is choosen accordingly with the rule and applied to the next example of the input data. At the beginning the choice may not be successful but the information about the characteristics of the stream of the input data and the efficiency evaluations are collected and the decision rule is corrected accordingly to this information. The choice is improved as far as more examples will be decided.

The formal problem definition in the terms of [3] follows in the next. The X is considered as k dimensional random variable that characterizes the stream of the input data. It is assumed that the density function $f(x)$ exists. The pattern is concerned with the pattern vector of determinant parameters. The vector components are real numbers.

The number of the classes is known and is equal to the number of the alternative programs. To delineate the classes the next characteristics of the classes are used in [3]. For every class ω_j $f_{\omega_j}(x)$ is the probability density function of x when x comes from ω_j. In the case of the comparison of programs this density function defines the distribution of the values of determinant parameters that belong to the examples of the input data for which the program corresponding to the class ω_j is more efficient than others. P_{ω_j} is a priori probability of class ω_j and means the probability when the pattern comes from the class ω_j. In the case of the programs P_{ω_j} means the probability that the program concerned with the class ω_j is more efficient than the others.

$$\sum_{j=1}^{n} P_{\omega_j} = 1 .$$

Also the next equality holds

$$f(x) = \sum_{j=1}^{n} P \omega_j f \omega_j (x) .$$

$P(\omega_i/x)$ is the conditional density function of the class ω_i and it indicates the probability when the pattern comes from the class ω_i in the point of the determinant parameters space. This function reflects the probability problem definition.

(1)
$$P(\omega_i/x) = \frac{f\omega_i(x)P\omega_i}{\sum_{j=1}^{n} P\omega_j f\omega_j(x)}.$$

The essence of the functions that are used in the pattern recognition methods was considered above in the terms of the comparison of programs. Now it is interesting to express these functions via the characteristics of the variables $T_1, \ldots, T_n$ and X.

In order to avoid the ambiguities we assume that the smaller value of the efficiency evaluation corresponds to the more qualitative solution. Accordingly the event that program i is more efficient than program j will mean that $t_i < t_j$ and event that program i is more efficient than other programs will mean that $t_i < t_j$, $\forall j, j \neq i$.

Furthermore we assume that there are density functions of random variables $T_1, \ldots, T_n f_1(t_1), \ldots, f_n(t_n)$. The variables $T_1, \ldots, T_n$ are independent because they reflect the characteristics of different programs and therefore

$$f(t_1, \ldots, t_n) = \prod_{i=1}^{n} f_i(t_i).$$

The probability $P\omega_i$ is expressed in the form

(2)
$$P\omega_i = \int \underset{\mathcal{D}}{\cdots} \int \prod_{i=1}^{n} f_i(t_i)\, dt_1 \ldots dt_n$$

where $\mathcal{D}$ is the area that $t_i < t_j$ $\forall j, j \neq i$.

If we assume that density functions exist for the conditional random variables T_i/X and they are $f_i(t_i/x)$, $i = 1, \ldots, n$, the probability $P(\omega_j/x)$ may be found by the similar way via functions $f_i(t_i/x)$ and equation (I) needn't be used.

The above consideration shows that using pattern recognition methods for the programs comparison is concerned with the determination of the functions.

The various decision rules defined via the statistic characteristics of the classes and the different situations when the information about characteristics is absent are considered in [3]. The application of pattern recognition methods to the problem of comparison of programs is very difficult from the point of view [3] because not only the distribution parameters but the form of the functions are unknown. In accordance with [3] the optimum methods are absent for this situation and various empirical methods are used.

One similar method is introduced below. The method is based on using random decision rule and allows us to choose one from n alternative programs by random way. The vector $(\lambda_1(x), \ldots, \lambda_n(x))$ corresponds to each point of the space of determinant parameters and vector components are the probabilities that satisfy the conditions

$$\lambda_i(x) \geq 0, \quad \sum_{i=1}^{n} \lambda_i(x) = 1.$$

When the next example of input data comes in, the pattern vector of the determinant parameters are counted and the program i is chosen in accordance with the probability $\lambda_i(x)$, $i = 1, \ldots, n$.

The probability $P(\omega_i/x)$ may be used as $\lambda_i(x)$. However the function $P(\omega_i/x)$ from argument x is unknown and therefore the next way was chosen. The space of determinant parameters is disjoined on the intervals. The vector $\lambda = (\lambda_1, \ldots, \lambda_n)$ is defined for every interval.

The component λ_i is function

$$\lambda_i = \frac{1/n + P\omega_i(L-1)}{L},$$

where L is the number of examples of the input data and the values of determinant parameters are enclosed in the interval.

If $\sum_{i=1}^{n} P\omega_i = 1$, then $\sum_{i=1}^{n} \lambda_i = 1$. If $L \to \infty$, then $\lambda_i \to P\omega_i$. If $L = 1$, then $\lambda_i = 1/n$.

The correlations above show that at the beginning of the works the programs are chosen with equal probability but as far as the number of examples is increased the programs are chosen with probabilities $P\omega_i$. The probabilities $P\omega_i$, $i = 1, \ldots, n$ are defined in the process of the program execution.

In order to check this approach the experiment with two programs was carried out. The next assumption was made for this case. First, two programs are compared and $n = 2$. Secondly, efficiency evaluations $t_1, t_2 > 0$. It corresponds to the case, when efficiency evaluations are the times of the program execution. Thirdly, the density functions have the form

$$f_i(t_i) = \frac{t_i - t_i^{\min}}{t_i^{\max} - t_i^{\min}}, \quad i = 1, 2.$$

The variables $t_i^{\min}$, $t_i^{\max}$, $i = 1, 2$ are the parameters of the distributions and their values are defined during the program execution.

Equation (2) for this case is expressed in the form

$$P\omega_1 = \iint\limits_{t_1 < t_2} \frac{t_1 - t_1^{\min}}{t_1^{\max} - t_1^{\min}} \cdot \frac{t_2 - t_2^{\min}}{t_2^{\max} - t_2^{\min}} \, dt_1 \, dt_2.$$

The solution of this integral results the next expressions depending on the values $t_i^{\min}$, $t_i^{\max}$, $i = 1, 2$.

If $t_1^{\min} > t_2^{\max}$ then $P\omega_1 = 0$.

If $t_2^{\min} > t_1^{\max}$ then $P\omega_1 = 1$.

If $t_2^{\max} > t_1^{\min}$, $t_2^{\min} < t_1^{\min}$, $t_1^{\max} > t_2^{\max}$ then

$$P\omega_1 = \frac{(t_2^{\max} - t_1^{\min})^2}{2(t_1^{\max} - t_1^{\min})(t_2^{\max} - t_2^{\min})}.$$

If $t_2^{min} > t_1^{min}$, $t_2^{max} < t_1^{max}$, then

$$P\omega_1 = \frac{(t_2^{max} - t_2^{min})(t_2^{min} - t_1^{min}) + 0.5\,(t_2^{max} - t_2^{min})^2}{(t_1^{max} - t_1^{min})(t_2^{max} - t_2^{min})}.$$

If $t_1^{max} > t_2^{min}$, $t_2^{min} > t_1^{min}$, $t_2^{max} > t_1^{max}$, then

$$P\omega_1 = 1 - \frac{(t_1^{max} - t_2^{min})^2}{2\,(t_1^{max} - t_1^{min})(t_2^{max} - t_2^{min})}.$$

If $t_1^{min} > t_2^{min}$, $t_2^{max} > t_1^{max}$, then

$$P\omega_1 = \frac{(t_2^{max} - t_1^{max})(t_1^{max} - t_1^{min}) + 0.5\,(t_1^{max} - t_1^{min})^2}{(t_1^{max} - t_1^{min})(t_2^{max} - t_2^{min})}.$$

$P\omega_2$ is defined from $P\omega_1 + P\omega_2 = 1$.

The method for comparison of two programs was obtained above. The application of the method is carried out in the following way.

The set of determinant parameters and efficiency evaluation are defined by means of qualitative analysis of programs and the input data. The space of the determinant parameters is disjoined on the intervals. The values of L, t_i^{max}, t_i^{min}, $i = 1, 2$ are derived during the programs execution for every interval.

When the next example of the input data comes in the values of the determinant parameters are calculated and the corresponding interval is chosen. The value L is incremented and the probabilities λ_1, λ_2 are calculated for its interval. In accordance with values λ_1, λ_2 the program is chosen by random way and applied to the example of the input data. After that the value of the efficiency evaluation t_i is derived and the values t_i^{min} or t_i^{max} are corrected if $t_i < t_i^{min}$ or $t_i > t_i^{max}$ accordingly.

The experiment was carried out with two programs intended for the minimization of the Boolean functions. The number of variables of the Boolean functions, the number of terms N and the average letters in the terms divided by the number of variables C were chosen as the determinant parameters. The efficiency evaluation was the time of the program execution.

The stream of examples of input data was generated by the random way and the method of comparison of programs was applied. The application of the method results in the decision rule introduced in Table I.

The analysis of Table 1 shows that the first program is more efficient when the values of N are small and the values of C are large. When the values of C are small or the values of N are large, the second program is more efficient.

The second part of the experiment meant the application of the three programs to the stream of the input data. At the beginning the programs U_1 and U_2 were applied separately. Then the third program, U_3 was applied. This program consisted of the programs U_1 and U_2 which were applied to the examples of input data in accordance with the decision rule introduced in Table 1.

Table 1.

	0 – 6	6 – 12	12 – 18	18 – 24
1 – 0.8	$L = 50$ $\lambda_1 = 0.96$ $\lambda_2 = 0.04$	$L = 81$ $\lambda_1 = 0.543$ $\lambda_2 = 0.457$	$L = 80$ $\lambda_1 = 0.45$ $\lambda_2 = 0.55$	$L = 51$ $\lambda_1 = 0.476$ $\lambda_2 = 0.524$
0.8 – 0.6	$L = 28$ $\lambda_1 = 0.964$ $\lambda_2 = 0.036$	$L = 88$ $\lambda_1 = 0.58$ $\lambda_2 = 0.42$	$L = 79$ $\lambda_1 = 0.19$ $\lambda_2 = 0.81$	$L = 33$ $\lambda_1 = 0.03$ $\lambda_2 = 0.97$
0.6 – 0.4	$L = 7$ $\lambda_1 = 0.29$ $\lambda_2 = 0.71$	$L = 22$ $\lambda_1 = 0.09$ $\lambda_2 = 0.91$	$L = 20$ $\lambda_1 = 0.05$ $\lambda_2 = 0.95$	–

The results of this experiment are introduced in Table 2.

Table 2.

N	10	15	20	25	30	35	40	45	50
U_1	2.04	6.38	18.7	32.3	68.2	47.9	68.6	72.6	–
U_2	1,16	3.59	12.0	80.7	106	133	–	–	–
U_3	0.91	3.5	5.99	5.44	13.9	13.9	18.8	42	54.6

The strings of Table 2 correspond to the programs U_1, U_2, U_3, the columns correspond to the values N. The elements of the table show the average time of program execution for examples with intended value N. The results in Table 2 show that the application of the program U_3 is more efficient than U_1 or U_2.

The experiment delineated above allows us to speak about the possibilities to apply methods of comparison of programs based on learning without teacher and also it allows us to expect to profit from their application.

REFERENCES

[1] *Закревский, А. Ю.:* Алгоритмы синтеза дискретных автоматов, М., «Наука», 1971. с. 512.

[2] *Толмачева, А. Ю.:* Распознавание алгоритмов структурного синтеза дискретных устройств. «Автоматика и вычислительная техника», 1974, № 6, с. 8 – 10.

[3] *Миленький, А. В.:* Классификация сигналов в условиях неопределенности. М., «Советское радио», 1975, с. 328.

[4] *Tou, J. T., Gonzales, P. C.:* Pattern recognition principles. Addison Wesley Publ. Comp,. Ontario, 1974.