

A Method of Software Quality Comparison based on PCA

Wenpeng Li, Wei Su, Jianxun Guo, Kuan Feng, Xiuming Yu*

China Electronics Standardization Institute, Beijing, China

Abstract

The platform software has a large number of functional and performance efficiency quality indicators, as well as differences in fixed basic hardware and software environments, making it impossible to effectively compare the quality. Considering that different users have different concerns about the product indicators of platform software, it brings certain difficulties to the selection of users. This paper proposes a software quality comparison method based on PCA, which extracts principal components by analyzing the correlation between data, reasonably allocates and evaluates software quality through dimensionality reduction and weighting, avoids errors caused by subjective experience, and can effectively adapt to changes in evaluation dimensions and the number of software products. Achieve the goal of horizontal comparison between products through a score.

Keywords

quality comparison, quality models, PCA

1. Introduction

With the rapid advancement of information technology, the level of information digitization in all aspects of social life is continuously improving, and software has emerged as a modern infrastructure. However, software is a product of human intellectual labor with poor visibility in terms of quality. The complexity and fuzziness of software make it challenging to quantify, thereby complicating users' ability to objectively comprehend its quality. The construction of a software quality model establishes a framework for measuring software quality attributes, establishing the relationship between measurable attributes and software quality, thereby providing a basis for evaluating and comparing the quality of software products.

In 1991, ISO/IEC JTC1/SC7 issued the ISO/IEC 9126 standard, which is based on McCall and Boehm's quality model. It reformulated the quality of software into 6 main attributes and 21 sub-attributes, marking a significant milestone in the standardization of software quality.

In the research of software quality evaluation, methods such as Delphi method, fuzzy comprehensive evaluation, topsis, evidential theory, and so on are often used. Most of these methods rely on subjective experience or fuzzy theory to construct quality measurement models, making them susceptible to human subjective influence in determining weights, introducing a level of uncertainty [1, 2, 3].

ISO/IEC 25010 provides the software product quality model (as shown in Fig. 1), offering eight

functional characteristics: Functional Suitability, Performance efficiency, Compatibility, Usability, Reliability, Security, Maintainability, and Portability, along with their respective sub-characteristics and properties [4].

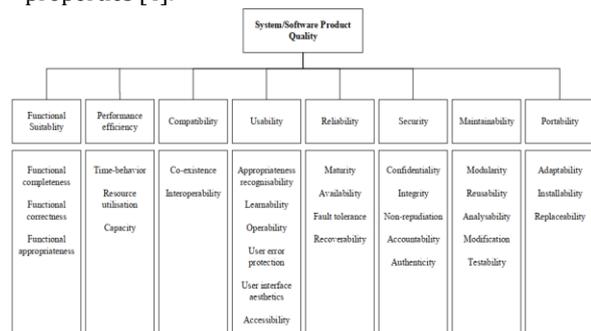


Figure 1: Product quality model

ISO/IEC 25023 provides quality measures for sub-characteristics of the software product quality model, which are widely used for measurement functions. These include mean response time and mean turnaround time for time behavior measurement; mean processor utilization, mean memory utilization, mean I/O utilization, and bandwidth utilization for resource utilization measurement; transaction processing capacity, user access capacity, and adequacy of user access increase in capacity measures [5].

The platform software has a large number of functional and performance efficiency quality indicators, as well as differences in fixed basic hardware and software environments, making it impossible to effectively compare the quality.

5th International Workshop on Experience with SQuaRE Series and its Future Direction, December 04, 2023, Seoul, Korea

liwp@cesi.cn (W. Li); yuxiuming@cesi.cn (X. Yu)

0009-0001-4065-2527 (W. Li)

© 2023 Copyright for this paper by its authors. The use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR Workshop Proceedings (CEUR-WS.org)



Considering that different users have different concerns about the product indicators of platform software, it brings certain difficulties to the selection of users. In the quality testing of large-scale platform product, function indicators of platform product capability and performance indicators are widely concerned. The results of test records are usually a numerical value (e.g. how many algorithms the platform supports, and the maximum concurrency supported by the performance result is 10,000). Inconsistent data dimensions make it difficult to compare and analyze products, and inconvenience users in comparing products.

This paper introduces a method for analyzing data correlation. This approach eliminates the need for subjective experience and is suitable for conducting large-scale comparisons of product quality.

2. Principal Component Analysis

Principal Component Analysis (PCA) accomplishes the objective of eliminating correlations between features by transforming a set of potentially correlated variables into a set of linearly independent variables through orthogonal transformation. This process retains crucial features while minimizing information loss. PCA generates two types of coefficients to achieve these goals: 'weights' that define the transformation from raw data to summary scores, and 'loadings' that indicate the strength of association between the raw variables and the low-dimensional representations [6].

PCA can be represented by the following mathematical model:

$$\begin{cases} x_1 = a_{11}F_1 + a_{12}F_2 + \dots + a_{1m}F_m + a_1\varepsilon_1 \\ x_2 = a_{21}F_1 + a_{22}F_2 + \dots + a_{2m}F_m + a_2\varepsilon_2 \\ \dots \\ x_p = a_{p1}F_1 + a_{p2}F_2 + \dots + a_{pm}F_m + a_p\varepsilon_p \end{cases} \quad (1)$$

where, $x_1, x_2, x_3, \dots, x_p$ represent p primitive variables, $F_1, F_2, F_3, \dots, F_m$ represent m factor variables, matrix form can be expressed as:

$$X = AF + a\varepsilon \quad (2)$$

where, F represent common factors, A represent factor loading matrix, a_{ij} represent factor loading.

For determining the weights of indicators in principal component analysis, the first step is to calculate the coefficients (b_{ij}) of indicators in the linear combinations of each principal component and the variance contribution rate (c_{ij}) of each principal component. The coefficient of each indicator in different linear combinations of principal components (b_{ij}) equals the ratio of the loadings (a_{ij}) of each indicator to the square root of the eigenvalues (λ_i) of

each component, which is $b_{ij} = \frac{a_{ij}}{\sqrt{\lambda_i}}$. Secondly, the indicator weight is the weighted average of the

coefficients of indicators in the linear combinations of principal components, with the weights being the variance contribution rates of the principal components. Finally, the indicator weights are normalized.

In software quality comparison, the matrix used for PCA can be represented as:

$$X = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1m} \\ x_{21} & x_{22} & \dots & x_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ x_{p1} & x_{p2} & \dots & x_{pm} \end{bmatrix} = [x_1 \quad x_2 \quad \dots \quad x_p]^T$$

Where, p represents the number of software products, and m represents the number of quality characteristics.

3. Software Quality Comparison based on PCA

Analyzing data from some tested blockchain platforms, performance indicators of blockchain platform products, specifically as follows:

- Average response time: the average time it takes for a transaction to be processed and confirmed. This metric is measured by iterating multiple times (with a 1-second interval) and obtaining the average response time for each iteration.
- Transaction processing rate: the number of transactions that can be processed per second. This metric measures the overall performance of the blockchain product in terms of transaction processing speed.
- Concurrent user/request count: the maximum number of users or requests that can be processed simultaneously. This metric measures the scalability of the blockchain product and its ability to handle multiple concurrent requests.
- Data processing volume: the amount of data that needs to be processed for each transaction. This metric measures the size of the transactions being processed and the overall data processing capacity of the blockchain product.
- CPU utilization: the percentage of CPU resources being used by the blockchain product. This metric measures the efficiency of the blockchain product in utilizing the available CPU resources.
- Memory utilization: the percentage of memory resources being used by the blockchain product. This metric measures the efficiency of the blockchain product in utilizing the available memory resources.

And, core functional indicators as following:

- Supported consensus mechanisms: the number of different consensus mechanisms that the blockchain product supports.

- Supported smart contract development languages: the number of different smart contract development languages that the blockchain product supports.

- Supported key algorithms: the number of different key algorithms that the blockchain product supports.

Selecting data from six blockchain platform products, see Table 1 for details.

Table 1
Blockchain platform products test data

blockchain platform products	Average response time (ms)	Transaction processing rate	Concurrent user/request count	Data processing volume (MB)	CPU utilization (%)	Memory utilization (%)	consensus mechanisms	smart contract development languages	key algorithms
1	7	12882	167272	490.05	74.7	8.3	4	3	6
2	9	44200	45000	1560.99	18	15	1	4	5
3	2	150918	111040	73.35	23	16	3	5	7
4	59	693	17845	16	43	20	5	4	5
5	29	799	27424	9	50	61	3	5	6
6	66	59241	8514	367	10	15	4	4	7

Constructing a 9×6 matrix as following:

$$X = \begin{bmatrix} \text{product1} \\ \text{product2} \\ \text{product3} \\ \text{product4} \\ \text{product5} \\ \text{product6} \end{bmatrix}$$

Performing PCA analysis in the above matrix. The cumulative variance explanation rate of the first four eigenvalues in table 2 exceeds 95%. Ingredients, also known as predictors or independent variables refer to the original variables or features in the dataset that you want to reduce the dimensionality. Components are the new variables that are created by PCA to represent the original data in a lower-dimensional space. These components are linear combinations of the original ingredients and are ordered so that they capture the most variance in the data.

Table 2
Variance explained table

Ingredients	Eigenvalue	Explained Variance (%)	Cumulative explained variance (%)
1	2.621	29.122	29.122
2	2.499	27.762	56.884
3	1.998	22.204	79.088
4	1.535	17.055	96.143
5	0.347	3.857	100
6			100

The factor loading coefficients were calculated using the first four eigenvalues, and the results are shown in Table 3.

Table 3
Factor load factor

indicators	Principal component 1	Principal component 2	Principal component 3
Average response time	0.824	-0.053	-0.171

Transaction processing rate	-0.41	0.762	0.429
Concurrent user/request count	-0.678	-0.335	0.623
Data processing volume	-0.699	-0.036	-0.674
CPU utilization	0.008	-0.805	0.337
Memory utilization	0.57	0.182	-0.191
consensus mechanisms	0.631	-0.43	0.543
smart contract development languages	0.318	0.803	0.009
key algorithms	0.041	0.54	0.705

Obtain the value of a in formula (1), which shown in Table 4.

Table 4
Ingredient matrix table

indicators	Ingredient 1	Ingredient 2	Ingredient 3
Average response time	0.314	-0.021	-0.086
Transaction processing rate	-0.156	0.305	0.215
Concurrent user/request count	-0.259	-0.134	0.312
Data processing volume	-0.267	-0.014	-0.337
CPU utilization	0.003	-0.322	0.169
Memory utilization	0.218	0.073	-0.095
consensus mechanisms	0.241	-0.172	0.272

smart contract development languages key algorithms	0.121	0.321	0.004
	0.016	0.216	0.353

Obtain the formula (2) as follows:

$$F=(0.291/0.961) \times F1+(0.278/0.961) \times F2+(0.222/0.961) \times F3+(0.171/0.961) \times F4$$

Table 5
Comprehensive score table

blockchain platform products	score	Principal component 1	Principal component 2	Principal component 3	Principal component 4
5	0.584	0.967	0.213	-0.266	1.643
3	0.495	-0.616	1.286	1.187	0.205
6	0.13	0.648	0.576	0.079	-1.415
4	-0.051	1.05	-0.772	-0.258	-0.49
1	-0.472	-0.846	-1.524	0.888	0.108
2	-0.686	-1.203	0.221	-1.63	-0.051

Based on the score in Table 5, product 5 is the best, and product 2 is the worst.

4. Conclusion

Blockchain platform products, as a typical software system with functional and performance efficiency indicators, need to comprehensively consider the support of the platform for algorithms, languages, consensus mechanisms, and performance efficiency indicators for scoring.

This paper proposed a software quality comparison method based on PCA, which extracts principal components by analyzing the correlation between data, reasonably allocates and evaluates software quality through dimensionality reduction and weighting, avoids errors caused by subjective experience, and can effectively adapt to changes in evaluation dimensions and the number of software products. Achieve the goal of horizontal comparison between products through a score.

References

- [1] Shi H , Ma J , Zou F . A Fuzzy Comprehensive Evaluation Model for Software Dependability Based on Entropy Weight[C]. International Conference on Computer Science & Software Engineering. IEEE Computer Society, 2008.
- [2] Bo L , Yang C . An Improved Comprehensive Evaluation Model of Software Dependability based on Rough Set Theory[J]. Journal of Software, 2009, 4(10).
- [3] Ding S , Yang S L , Fu C . A novel evidential reasoning based method for software trustworthiness evaluation under the uncertain and unreliable environment[J]. Expert Systems with Applications, 2012, 39(3):2700-2709.
- [4] ISO/IEC 25010. Systems and software engineering —Systems and software Quality Requirements and Evaluation (SQuaRE) —

The principal component scores and comprehensive scores of the six products, which listing in TABLE 5.

Firstly, calculate the product of each principal component's linear combination coefficient and its corresponding variance explained rate, then accumulate these products, and finally divide the total variance explained rate by this sum to get the comprehensive score.

System and software quality models, International Standard Organization, 2011.

- [5] ISO/IEC 25023. Systems and software engineering —Systems and software Quality Requirements and Evaluation (SQuaRE) — Measurement of system and software product quality, International Standard Organization, 2016.
- [6] Yumeng C , Yinglan F .Research on PCA Data Dimension Reduction Algorithm Based on Entropy Weight Method[C]//2020 2nd International Conference on Machine Learning, Big Data and Business Intelligence (MLBDBI).2020.DOI:10.1109/MLBDBI51377.2020.00084.