OWA and PCA Integrated Assessment Model in Software Project

Duen-Yian Yeh*, Taiwan, yeh@mail.tit.edu.tw
Ching-Hsue Cheng, Taiwan, chcheng@pine.yuntech.edu.tw
Hwei-Wun Yio, Taiwan, yio@pine.yuntech.edu.tw

Abstract
This research proposed the OWA and PCA integrated assessment model for easily overcoming the complexity and for appropriately evaluating the performance of software project. First, the amount of evaluation data was reduced by PCA. As a result, the number of criteria was cut down, but the originally contained information was preserved. Next, OWA was employed to analytically obtain the weights of resultant criteria. In empirical validation, three software projects of one famous hospital in Taiwan were selected as the objects to examine the model appropriateness. Furthermore, a comparison was taken to reveal the superiority of this model.

KEYWORDS: Principal Component Analysis (PCA), Ordered Weighted Averaging (OWA), Decision Making, Software Project Management.

1. Introduction
The main goal of software project management is to proceed with the prevention and adjustment in advance with regard to the uncertain factors that might affect the schedule and quality in the duration of project development [1]. Expectedly, the occurrence of damages on the credit and profit of software industry will be avoided [2]. From the viewpoint of management, the function of evaluating software development process cannot only assist the deciders with the prediction of feasibility and the impact of benefits in advance, but also offer excellent help to the improvement process of software project and the strategy making of management. In fact, software project management is a complicated issue. Its performance evaluation will also be a complicated multi-criteria decision making topic. However, two difficulties will be inevitably encountered. One is the high number of criteria needed for properly evaluating the performance of software project, while the other lies in deciding the weight distribution of criteria.

This research treated the performance evaluation of software projects as the topic with the aim at developing an appropriate software project evaluation model, PCA and OWA integrated evaluation model. In this model, a suitable criterion structure and an integrated evaluation algorithm were established to obtain the proper evaluation results for best decision making. First, this model adopts the Principle Component Analysis operator (PCA) with the expectation at using a few criteria to explain the original complex condition. As a result, the degree of complexity of project evaluation would be considerably reduced. Second, the
Ordered Weighted Averaging operator (OWA) [3, 4] was used to analytically decide the weight distribution of criteria. In addition, this model particularly considers the situation parameter, \( \alpha \), for flexibly adjusting the weight distribution. Finally, in empirical validation, three software projects belong to one famous hospital in Taiwan were selected as the targets to examine the appropriateness of this model. Furthermore, a model comparison was taken to reveal the superiority of this model.

The following structures of this research are as follows. Chapter 2 is the literature review. Chapter 3 describes in detail the construction of “the PCA and OWA integrated evaluation model,” while chapter 4 contains the case validation and the model comparison with regard to three software projects. Finally, chapter 5 proposes concise conclusions.

### 2. Literature Review

Software project refers to the project that develops the requirement of certain function and accomplishes the software within the agreed deadline. The software development is generally divided into three phases: planning, development and application [5]. The first phase proceeds with system requirement analysis according to the data collected. The missions of this phase are to accomplish the planning of project management, the draft of system requirement and the contract with the clients. The second phase is the one of development that includes system analysis, software requirement analysis and the series of sub-phases, design, encoding and test. The main target of this phase lies in accomplishing the definition of system, specification of software requirement, specification of system design, software program, test data and historical records of test. In the third phase, the activities include system installation, users’ training and system maintenance. The main mission of this phase is to accomplish the users’ manual, training courses, teaching materials, system disposition management, etc.

The software development models that are commonly used include: waterfall model, Boehm's spiral model, V pattern model, saw tooth model, shark tooth model, issue-based model, etc. [6]. In actual utilization, there is no definite answer with regard to which development model is better. Basically, the adoption of software development model is a project-based problem [6]. Thus, it is difficult to determine the appropriate evaluation structure and the criteria for all models. Figure 1 collects an evaluation structure and its associated criteria from related websites and literatures [7-9]. In this structure, two criteria and ten associated sub-criteria are included.

<table>
<thead>
<tr>
<th>Criterion</th>
<th>Sub-criterion</th>
</tr>
</thead>
<tbody>
<tr>
<td>Software project design</td>
<td>Specification analysis((C_1))</td>
</tr>
<tr>
<td></td>
<td>System architecture design</td>
</tr>
<tr>
<td></td>
<td>System decomposition((C_2))</td>
</tr>
<tr>
<td></td>
<td>System architecture document((C_3))</td>
</tr>
<tr>
<td>Detailed design</td>
<td>Data structures and algorithms design((C_4))</td>
</tr>
<tr>
<td></td>
<td>Structure quality measurement((C_5))</td>
</tr>
<tr>
<td></td>
<td>Program language/tools((C_6))</td>
</tr>
<tr>
<td></td>
<td>Design walk-through((C_7))</td>
</tr>
<tr>
<td></td>
<td>Validation((C_8))</td>
</tr>
<tr>
<td></td>
<td>Test plan((C_9))</td>
</tr>
<tr>
<td></td>
<td>Design documentation((C_{10}))</td>
</tr>
</tbody>
</table>

*Figure 1 Criterion structure and the associated criteria for evaluating software project*
Since Yager has proposed OWA in 1988 [10], various related algorithms for determining the criterion weights for many research fields were published in the literatures [11]. The most important characteristic of this operator lies in deciding the associated weights of criteria by analytical procedures and, simultaneously, possesses the reasonable and flexible advantages at weight decision by means of the situation parameter, orness $\alpha$. The basic definitions and operations are briefly described as follows [3, 12, 13]:

The mapping function of OWA operator of dimension $n$: $f : R^n \rightarrow R$, and

$$f(a_1, a_2, ..., a_n) = \sum_{i=1}^{n} W_i b_i, \quad (1)$$

in which $W_i$ belongs to the vector of associated weights: $W = [W_1, W_2, ..., W_n]^T$ with

$$\sum_{i=1}^{n} W_i = 1 \quad \text{and} \quad W_i \in [0,1]. \quad b_i \quad \text{is the } i^{th} \text{ largest element of aggregated object set } \{a_1, a_2, ..., a_n\}.$$

In addition, two measures were introduced to characterize the aggregation. The first one, the measure of orness of the aggregation, is defined as

$$Orness(W) = \frac{1}{n-1} \sum_{i=1}^{n} (n-i)W_i \quad (2)$$

and it characterizes the degree to which the aggregation is like an or operation. The second one, the measure of dispersion of the aggregation, is defined as

$$Disp(W) = -\sum_{i=1}^{n} W_i \ln W_i \quad (3)$$

and it measures the degree to which $W$ takes into account all information in the aggregation.

O’Hagan [13] proposed an OWA operator that the associated weight vector is obtained with respect to the optimal condition having maximal entropy for a given level of orness $\alpha$. The computations of this operator merely require $\alpha$ value and the number of criteria $n$. The procedures follow the mathematical programming problem:

$$\text{Maximize: } -\sum_{i=1}^{n} W_i \ln W_i$$

$$\text{Subject to: } \alpha = \frac{1}{n-1} \sum_{i=1}^{n} (n-i)W_i, \quad 0 \leq \alpha \leq 1 \quad (4)$$

$$\sum_{i=1}^{n} W_i = 1 \quad \text{and} \quad W_i \in [0,1]$$

Using the method of Lagrange multipliers, the maximal entropy weights can be determined from the following polynomial formulas [13]:

3
\[ W_i^\alpha [(n-1)\alpha + 1 - nW_i]^\alpha = [(n-1)\alpha]^{\alpha-1} \left[ ((n-1)\alpha - n)W_i + 1 \right], \quad (5) \]

\[ W_n = \{[(n-1)\alpha - n]W_i + 1\} / \{[(n-1)\alpha + 1 - nW_i]\}, \quad \text{and (6)} \]

\[ W_j = n^{-\frac{1}{2}}W_i^{n-j}W_n^{j-1}, \quad 2 \leq i \leq n-1. \quad (7) \]

Basically, once \( n \) and \( \alpha \) are given then all the associated weights can analytically be obtained from formulas (5)- (7).

PCA [14, 15] is the method obtains \( n \) new variables by means of linear combinations of the original \( p \) variables existing in available data. In order to reduce the amount of data, \( n \) is usually much less than \( p \). In selecting \( n \), one must consider the explanatory capacity of the \( n \) new variables upon the covariance structure of the original variables. On the other hand, the characterizing function of PCA lies in the zero correlative relationship between the \( n \) new variables. As a result, the \( n \) new variables refer to \( n \) principle components. In practical implementation, if the \( p \) variables have the same scale of measurement, one can directly use the correlation matrix to obtain principle components. On the contrary, the standardization of data should be taken in advance if the scales of measurement are not the same.

In PCA, the first principle component is obtained from the condition its variance is equal to the maximal eigenvalue of the covariance matrix and, as a result, contains the most amount of variance of the original variables. The first principle component can be represented as follows:

\[ Y_1 = a_{i_1}x_1 + a_{i_2}x_2 + \ldots + a_{i_p}x_p, \quad (8) \]

in which the coefficients of \( Y_1 \) are the corresponding elements of eigenvector having maximal eigenvalue. Next, the second principle component can be obtained by the same procedure with the additional condition of zero correlative relationship between \( Y_1 \) and \( Y_2 \). Similarly, the remainder can be obtained following the same procedure. Based upon the above, the new synthetic variables \( Y_1, Y_2, \ldots, Y_n \) are respectively called the first, second…\( n^{th} \) principle components of the original variables and their respectively contained variances are gradually diminished.

The allocation of evaluation data in two-dimensional space could be used to demonstrate the geometric meaning of PCA. Assume that \( k \) evaluators are requested to evaluate the target with respect to the two criteria \( X(1) \) and \( X(2) \). Figure 2 shows the allocation of evaluation results, in which each black dot represents an evaluation data come from one of \( k \) evaluators. Obviously, the profile of allocation in \( X(1) \) and \( X(2) \) coordinates is approximately an oblique ellipse. After an orthogonal rotation of angle \( \theta \) in coordinates, one can obtain a new set of coordinates as Figure 2 shown. In Figure 2, the long axis of ellipse is termed as \( Y(1) \), while the short axis is termed as \( Y(2) \). The formula of rotation is given as:

\[ Y(1) = X(1)\cos \theta + X(2)\sin \theta \quad (9) \]

\[ Y(2) = X(1)(-\sin \theta) + X(2)\cos \theta \quad (10) \]

In the new coordinates \( Y(1) \) and \( Y(2) \), the following properties stand:

1. The correlation of the \( k \) evaluation data points in the principle component
coordinates $Y(1)$ and $Y(2)$ is zero.

2. Most of variance of the evaluation data points on the two dimensional plane distributes on $Y(1)$ axis, while less on $Y(2)$ axis.

In practice, $Y(1)$ and $Y(2)$ are two new variables transformed from the original criteria $X(1)$ and $X(2)$. Owing to the fact the maximal variance exists in variable $Y(1)$, $Y(1)$ could be the most suitable variable used to represent the original evaluation data. As expected, the loss of messages would be minimized. Therefore, $Y(1)$ axis is called the first principle component. On the other hand, $Y(2)$ is orthogonal to $Y(1)$ and has less variance. It is called the second principle component.

In all, PCA is the method generates few new variables to represent the original variable data. During the generation of new variables, one must simultaneously focus upon the following goals: properly reducing the number of variables, attempting to remain the content of the original variable data and remaining new variables independent to one another.

3. PCA and OWA Integrated Evaluation Model

The “PCA and OWA integrated evaluation model” was proposed to provide an appropriate model for evaluating the performance of software projects. The characterizing integration of PCA and OWA lies in the capture of respective advantages of them. Two phases involve in the evaluation model, i.e., PCA mode and OWA mode and performance aggregation. All the 9 computation steps needed to implement this model are shown as follows:

**Phase 1: PCA mode**

After obtaining the $k$ evaluation data of $m$ software projects with respect to each criterion, the steps of PCA mode include:

- Step 1: Collecting the evaluation data in matrix form
- Step 2: Obtaining the correlation matrix of criteria
- Step 3: Obtaining the eigenvalues and eigenvectors of the correlation matrix
- Step 4: Deciding the number ($n$) of principle components based upon the rule of totally accumulated variance $\geq 90\%$
- Step 5: Transforming linearly the original evaluation data onto the selected eigenvectors
- Step 6: Obtaining the PCA scores of software projects ($m \times n$) based upon the above linear transformation

**Phase 2: OWA mode and performance aggregation**

- Step 7: Obtaining the weights of $n$ principle components from formulas (5)- (7) according
to the situation parameter $\alpha$

Step 8: Aggregating the PCA scores and the OWA weights to obtain the aggregated performances of software projects. The aggregation algorithm is given as follows:

$$[P_{ij}]_{m \times n} \times [W_j]_{n \times 1} = A_{m \times 1}, (11)$$

in which $P_{ij}$ represents the PCA score of $i$th software project with respect to $j$th principle component. $W_j$ is the weight of $j$th principle component. $A_{m \times 1}$ represents the aggregated performance vector of $m$ software projects.

Step 9: Ranking the $m$ software projects according to their aggregated performances

4. Case Validation and Comparison

This research used three software projects as the evaluation objects for verifying the proposed model and also for comparing the evaluation results with other models. The introduction of the three software projects is elaborated in Table 1. These software projects are belong to Talin Tzu-Chi Hospital that is one of the famous hospitals in Taiwan. For breaking through the evaluation limitation, this research took two expert groups and two time periods for evaluation. The first group contains 45 members of on-the-job program in information management department of national A-University in Taiwan. This group evaluated the performances of Project (I) and Project (II) in the first time period. On the other hand, the second group contains 47 members of two-year vocational program of the same department of national A-University. This group evaluated the performances of Project (I), Project (II) and Project (III) in the second time period.

This research used the criterion structure and its associated criteria shown in Figure 1 to evaluate the performance of these 3 software projects. A well-designed questionnaire was used as the instrument to express the experts’ opinions on the performance of software projects.

4.1 Case Verification

After the computations in phase 1, Figure 3 shows the respective percentages that could explain the individual accumulated variance of these 3 software projects. Based upon the rule of accumulated variance $\geq 90\%$, it could be recognized that the setting of number of principle components at 6 is enough to preserve the information contained in the original variables for each software project. Then, the PCA Scores of each software project with respect to the 6 principle components could be obtained by corresponding mappings.

In phase 2, the step 7 gives the OWA weight distributions of the 6 principle components that could be obtained from Formulas (5)-(7) under the different settings of situation parameter $\alpha$. Table 2 presents the results, in which 6 of typical values of $\alpha$ were taken. As seen, the difference does exist between each two weight distributions of different $\alpha$ values. Finally, for different $\alpha$ values, the aggregated performances of these 3 software projects could be respectively obtained by aggregating the OWA weights and the PCA scores with respect to principle components from Formula (11). As results, Figure 4 depicts the aggregated results. In this figure, the negative values of project (I) and project (III) come from the linear transformations of the original evaluation data onto the selected eigenvectors.
<table>
<thead>
<tr>
<th>Code of software project</th>
<th>Summary of system</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project (I)</td>
<td></td>
</tr>
<tr>
<td>Project Name</td>
<td>Medical management information system of Talin Tzu-Chi Hospital: Web system of the obstetrics and gynecology department</td>
</tr>
<tr>
<td></td>
<td>In selecting developmental instruments, the Microsoft Visual Studio.NET was chosen for conveniently employing WEB to save a sharing database, in which the case history of the patients and the doctors’ information would be safely kept and used. The data utility and sharing mechanism were thus established. The managers could enjoy the advantage of data centralized management under the structure of Client/Server and, in addition, the client side could operate everywhere upon the authority by the validation of codes through server which reduce the cost for maintaining the clients' software and upgrade the convenience of newel information for the complete sharing of data.</td>
</tr>
<tr>
<td>Project (II)</td>
<td></td>
</tr>
<tr>
<td>Project Name</td>
<td>XML Web Service practices: an example of the reservation operation system of Talin Tzu-Chi Hospital</td>
</tr>
<tr>
<td></td>
<td>This system assists the related doctors, medical and administrative personnel to process and manage the reservation matters to satisfy the medical requirement in the hospital and to comply with the related regulations of Bureau of National Health Insurance and Department of Health. The system was based upon Internet Web and XML Web Service techniques. As a result, it fulfills the goal to connect to the internet anywhere, anytime and any installations for convenient obtainment of the required information and service. Expectedly, the operational cost is reduced and the whole efficiency is then upgraded. In addition, the information personnel could centralize the management of database and, consequently, could fulfill the consistency and immediate aspect of data. Therefore, the maintenance of the system would be easier and faster.</td>
</tr>
<tr>
<td>Project (III)</td>
<td></td>
</tr>
<tr>
<td>Project Name</td>
<td>Web operation of medical management information system of Talin Tzu-Chi Hospital: Endoscope and Clinical Information System</td>
</tr>
<tr>
<td></td>
<td>The medical management information system was established in Web version to achieve the interaction of data and figures. The first sub-system, “friendly project,” functions as the medical and electronic platform between Talin Tzu-Chi Hospital and grassroots hospitals. It provides the direction of application and development of enterprise network through Web Computing. The second and third subsystems are “Endoscope” and “Clinical Information System”. Through the internet, the medical personnel are allowed to more efficiently manage the diagnosis function and control the patients’ diagnosis data. The development instrument mainly refers to Microsoft Visual Studio.NET. By means of Web operation, the business was not limited to the internal hospital. As long as the users access to the authority, they could use the system through the internet anywhere and anytime. As a result, it expands the scale of the service.</td>
</tr>
</tbody>
</table>

Table 1 Introduction of three software projects under evaluation

Figure 3 Explanatory percentages of accumulated variance of three software projects
\( \alpha = 0.5 \quad \alpha = 0.6 \quad \alpha = 0.7 \quad \alpha = 0.8 \quad \alpha = 0.9 \quad \alpha = 1.0 \)

\begin{center}
\begin{tabular}{lcccccc}
\hline
 & \( \alpha = 0.5 \) & \( \alpha = 0.6 \) & \( \alpha = 0.7 \) & \( \alpha = 0.8 \) & \( \alpha = 0.9 \) & \( \alpha = 1.0 \) \\
\hline
\( W_1 \) & 0.16666 & 0.24676 & 0.34747 & 0.47811 & 0.66372 & 1.00000 \\
\( W_2 \) & 0.16666 & 0.20721 & 0.23977 & 0.25473 & 0.22396 & 0.00000 \\
\( W_3 \) & 0.16666 & 0.17401 & 0.16543 & 0.13571 & 0.07559 & 0.00000 \\
\( W_4 \) & 0.16666 & 0.14614 & 0.11416 & 0.07229 & 0.02548 & 0.00000 \\
\( W_5 \) & 0.16666 & 0.12274 & 0.07876 & 0.03851 & 0.00860 & 0.00000 \\
\( W_6 \) & 0.16666 & 0.10308 & 0.05437 & 0.02053 & 0.00290 & 0.00000 \\
\hline
\end{tabular}
\end{center}

Table 2 OWA weight distributions under the different settings of \( \alpha \)

As seen from Figure 4, all the results of 6 \( \alpha \) settings reveal that Project (II) had the best performance, while Project (I) had the worst. In other words, the performance ranking is Project (II) > Project (III) > Project (I). In addition, the least difference between the aggregated performances of these 3 software projects exists at the case of \( \alpha = 0.5 \). The reason lies in the fact that “\( \alpha = 0.5 \)” refers to the maximal entropy in data and, therefore, results in the most uniformity in criterion weights, i.e., all of the weights equal to \( 1/n \).

![Figure 4 Aggregated performances of the 3 software projects for different values of \( \alpha \)](image)

4.2 Validation and Comparison

In order to further validate the appropriateness of the proposed model, this research also conducted the Simple Weighted Average (SWA) operator to proceed with the comparison between the aggregated results. In implementing SWA, two types of criterion weights were used. One is the uniform type, while the other uses the weights obtained by the model of Lee et al. [7]. By means of the model of Lee et al., the criterion weight distribution is shown in Table 3. After aggregating these associated weights and the original evaluation data by evaluators, the aggregated performances of these 3 software projects are respectively presented in Table 4.

From Table 4, it can be found that all the aggregated results illustrate the same performance ranking: Project (II) > Project (III) > Project (I). This result provides a powerful evidence for validating this integrated evaluation model for evaluating the performance of these 3 software projects. On the other hand, as compared with SWA, this integrated evaluation model would be a useful evaluation model even by means of less amount of evaluation criteria and corresponding information data. In addition, owing to the adoption of situation parameter, orness \( \alpha \), this integrated evaluation model provides a flexible possibility in deciding the weight distribution of evaluation criteria with respect to the evaluator’s confident or conservative (pessimistic) attitude.
## 5. Conclusions

This research proposed the PCA and OWA integrated evaluation model to appropriately evaluate the performances of software projects. The characteristic of this model lies in the integration of respective advantages of the PCA and OWA operators. Finally, in case verification, this research used three software projects as the targets to validate the advantages of this model.

In this model, a well-designed questionnaire was used as the instrument to express the experts' opinions on the performances of targets with respect to each criterion. Then, the PCA operator was employed to reduce the handling amount of data and the OWA operator was used to analytically obtain the weight distribution of criteria. In the empirical case of three software projects, the original ten criteria were linearly transformed into the reduced six principle components by means of PCA, and also transformed were the PCA scores of 3 software projects with respect to the 6 principal components. After conducting OWA, the weight distribution of six principle components was obtained. As aggregating the PCA scores and the OWA weights, the ranking of aggregated performance, Project (II) > Project (III) > Project (I), was obtained for all different values of situation parameter $\alpha$.

Two superiorities contain in this model. One lies in the verified evidence that this
integrated evaluation model would be a useful model for evaluating the performances of software projects even by means of less amount of evaluation criteria and corresponding information data. As a result, it reduces the complexity of software project evaluation. The other lies in the fact that, owing to the adoption of situation parameter $\alpha$, this integrated evaluation model provides a flexible possibility in deciding the weight distribution of evaluation criteria.

6. Reference


