An expert module to improve the consistency of AHP matrices

A. Ishizaka and M. Lusti

WWZ/Information Systems, University of Basel, 4003 Basel, Switzerland
E-mail: Alessio.Ishizaka@unibas.ch

Received 10 March 2002; received in revised form 14 March 2003; accepted 12 May 2003

Abstract

In AHP (Analytical Hierarchy Process), the calculated priorities are plausible only if the comparison matrices are consistent or near consistent. This condition is reached if—within the pairwise comparison process—the transitivity and reciprocity rules are respected. We describe a Prolog application which helps the decision-maker to build a consistent matrix or a matrix with a controlled error. An expert module detects rule transgressions, explains them (How-explanations), suggests alternatives (What If-explanations) and gives hints on how to continue the comparison process.

Keywords: Multiple criteria decision; AHP; comparison matrix; consistency; transitivity; reciprocity; expert module; explanations.

Introduction

The Analytic Hierarchy Process (AHP) evaluates decision alternatives by pairwise comparison, thus allowing more accurate judgements than the simple weighted product model (Saaty, 1994, pp. 8–9). The calculated priorities are plausible only if the comparison matrices are consistent or near consistent. Especially for high-order matrices, consistency is difficult to reach if the alternatives can only be measured on an ordinal scale.

To improve an inconsistent matrix, a user can be urged to reconsider pairwise comparisons until the consistency measure proves to be satisfactory (e.g. Harker, 1997). Feedback after the completion of the comparison matrix is frustrating to the user, because it gives no hints about the comparisons to reconsider. This paper presents an expert module intervening after each pairwise comparison, which contradicts the comparisons made so far, explains it, and suggests a consistent alternative. The user is guided through a sequence of four steps which lead to a consistent or near consistent matrix.
Consistency

The elements \(a_{i,j}\) of a comparison matrix \(A\) compare the alternatives \(i\) and \(j\) of a decision problem. The \(a_{i,j}\) are said to be consistent if they respect the following transitivity (1) and reciprocity (2) rules:

\[
a_{i,j} = a_{i,k} \cdot a_{k,j}\quad \text{where } i, j \text{ and } k \text{ are any alternatives of the matrix}
\]

(1)

**Example 1.** Suppose you like an apple twice as much as an orange \((a_{1,2} = 2)\) and an orange three as much as a banana \((a_{2,3} = 3)\). If you like an apple six times as much as a banana \((a_{1,3} = 6)\), the transitivity rule is respected.

\[
a_{ij} = \frac{1}{a_{ji}}
\]

(2)

**Example 2.** If you like an apple twice as much as an orange \((a_{1,2} = 2)\), then you like an orange half as much as an apple \((a_{2,1} = 1/2)\).

A comparison matrix is reciprocal if its inferior part is reciprocal to the superior part and all the elements of the principal diagonal are 1. Therefore a transitivity test of one of the two parts of the matrix is sufficient:

\[
a_{i,j} = a_{i,k} \cdot a_{k,j}\quad \text{for } j < k < i
\]

(3)

Hence, for each element \(a_{i,j}\) a number of \(j - (i+1)\) equations (3) have to be respected.

**Example 3.** The comparison \(a_{2,5}\) has to conform to the equations \(a_{2,5} = a_{2,3} \cdot a_{3,5}\) and \(a_{2,5} = a_{2,4} \cdot a_{4,5}\).

These equations can be expressed only by terms of the first diagonal above the principal diagonal (ie. \(a_{1,2}, a_{2,3}, \ldots, a_{n-1,n}\)):

\[
a_{i,j} = a_{i,i+1} \cdot a_{i+1,i+2} \cdot \ldots \cdot a_{j-1,j}
\]

(4)

**Example 4.**

\[
a_{2,5} = a_{2,3} \cdot a_{3,5}
\]
\[
a_{3,5} = a_{3,4} \cdot a_{4,5}
\]
\[
a_{2,5} = a_{2,3} \cdot a_{3,4} \cdot a_{4,5}
\]

An \(n \times n\) matrix contains \(n^2\) comparisons. \(n - 1\) of the elements can be chosen freely, the other elements have to consider the reciprocity and transitivity rules (2, 4).

**How to build a consistent matrix**

When filling a comparison matrix, we distinguish four types of comparisons: comparisons on the principal diagonal, independent comparisons, transitive comparisons, and reciprocal comparisons.
1) The principal diagonal contains all comparisons of an alternative with itself (Fig. 1):

![Fig. 1. Principal diagonal of a comparison matrix.](image1)

2) Independent comparisons are not linked to other comparisons by the transitivity or reciprocity rules. This is the case if the first \( n - 1 \) comparisons are distributed over all columns or lines respectively (counter example in Fig. 2).

![Fig. 2. Instead of a single comparison the fourth column contains two entries. This way, the first \( n - 1 \) comparisons are dependent, because of \( a_{1,4} = a_{1,2} \cdot a_{2,4} \) (transitivity rule). To keep the comparisons independent each entry has to be in a new column.](image2)

Beginning with the first line \( (a_{1,2}, a_{1,3}, \ldots, a_{1,n}) \) is questionable. It can be argued that comparing alternatives linewise compromises the (psychological) independence of the comparisons, an advantage of the pairwise AHP method compared to the simultaneous procedure. We therefore choose the first diagonal as a starting point (Fig. 3). Furthermore, this choice allows the calculation of the other comparisons of the upper matrix by multiplication instead of division \( (a_{i,k} = \frac{a_{ij}}{a_{ki}}) \).

![Fig. 3. Choosing the first diagonal above the principal diagonal as a starting point.](image3)

3) Transitive comparisons can be deduced from the first diagonal entered in the second step (4) (Fig. 4). The comparison between alternative 1 and alternative 5, for example, is given by: \( a_{1,5} = a_{1,2} \cdot a_{2,3} \cdot a_{3,4} \cdot a_{4,5} = 0.5 \cdot 2 \cdot 1 \cdot 0.25 = 0.25 \).
4) Each entry in the lower part of the comparison matrix is the reciprocal of the corresponding upper part entry (2). In Fig. 5, for example, $a_{5,1}$ is $\frac{1}{a_{1,5}}$.

\begin{align*}
\begin{array}{cccc}
1 & 0.5 & 1 & 1 \\
2 & 1 & 2 & 0.5 \\
1 & 1 & 1 & 0.25 \\
1 & 1 & 0.25 & 1 \\
4 & 2 & 4 & 4 \\
\end{array}
\end{align*}

Fig. 5. The lower part is the reciprocal of the upper part.

**Inconsistency tolerance**

For the construction of a consistent matrix, only the second step is necessary (section ‘How to build a consistent matrix’). The first step is trivial; the last two steps can be deduced by the transitivity (4) and the reciprocity rules (2). This was proposed by Wedley, Schoner, and Tang (1993) for the calculation of missing comparisons.

This procedure is not reliable for the transitive comparisons (step three) because it introduces values which are not necessarily present in the decision-maker’s mind. For instance, a weak preference value of 3 (on the fundamental scale) for the comparisons $A$ versus $B$ and $B$ versus $C$ would imply a predominant or absolute value for $A$ versus $C$ of $9 = 3 \cdot 3$. To overcome this problem, our system lets the user choose the comparison scale and the error tolerated in the translation rule:

- The fundamental scale one to nine has shortcomings (Dyer, 1990). For some problems a wider scale (e.g. 1–15) and for others a shorter scale is more appropriate. A description of various scales can be found in Triantaphyllou (2000, pp. 23–45).
- To allow a certain degree of inconsistency, a tolerated error $e$ (percentage of the height value of the comparison scale) is introduced. Hence, the transitivity rule (4) is supplemented by an error term:

$\begin{align*}
a_{i,j} = a_{i,i+1} \cdot a_{i+1,i+2} \cdot \ldots \cdot a_{j-1,j+1} \pm \frac{e \cdot h}{100}
\end{align*}$

where $e$ is the tolerated error $h$ and is the height value of the comparison scale.
If the extended transitivity rule (5) is violated, the user is required to modify either the value entered (left part of (5)) or the comparisons of the first diagonal (right part of (5)). Modifying the values of the first diagonal induces changes on the transitive comparisons. The user is offered both matrices and can adopt the most appropriate (example 5).

**Example 5.** Suppose the user chooses 4 for the comparison \(a_{14}\) of the matrix of Fig. 6 and 0 for the tolerated error. But according to the transitivity rule (5) the user is supposed to enter \(a_{1,4} = 0.5 \cdot 2 \cdot 1 = 1\). If the user maintains his entry, he can modify the comparisons \(a_{1,2}, a_{2,3},\) and \(a_{3,4}\) of the first diagonal and see the effects on the comparisons above the first diagonal. Then he can either adopt the solution proposed by the program (Fig. 7) or proceed with his own matrix (Fig. 8).

![Fig. 6. Comparison matrix. The comparison \(a_{1,4}\) is requested.](image)

![Fig. 7. Matrix with proposed comparison \((a_{1,4} = 1)\).](image)

![Fig. 8. Matrix after changing premises \((a_{1,2} = 1\) and \(a_{3,2} = 2)\).](image)

**Expert module design**

The expert module is implemented in Visual Prolog\(^1\) and assists the user in the construction of a consistent or a near consistent matrix. It guides him through the comparison steps (section ‘How to build a consistent matrix’), indicates errors, explains them, and proposes an alternative.

\(^1\)Visual Prolog is a compiled variant of Standard Prolog (Clocksin and Mellish, 1987).
Figure 9 shows the flow chart of the algorithm (if the user specifies that the first and the last steps should automatically be taken by the system the errors two and seven will not appear).

![Flow chart of the algorithm](image)

**Error 1:** The entered comparison is outside the scale.
**Error 2:** The comparisons on the main diagonal must be 1.
**Error 3:** Since the entries of step 3 are linked to those of step 2 by the transitivity rule (5), each comparison of step 2 must not take a value which would lead to a step 3 value outside the comparison scale.
**Error 4:** The transitivity rule (5) is not obeyed.
**Error 5:** You skipped a step.
**Error 6:** The matrix is completed. The comparisons are frozen.
**Error 7:** The reciprocity rule (2) is not obeyed.

**Explanations**

When entering comparisons, rule violations are signalled and explained (Fig. 10).
Our expert module offers three types of explanations:

- A **Stereotype**-explanation is a static text linked to a determinate piece of knowledge. It is used for simple, ‘flat’ explanations, which cannot be hierarchically detailed.
- A **How**-explanation justifies an answer by a top-down tree from the requirements to the solution. A How-explanation is based on a recursive trace of the processed Prolog clauses (Clark and McCabe, 1982). The entire calculation can be represented as a tree whose root contains the end result. In Fig. 11 for example, the root calculated comparison with tolerated error is the sum of its two children, the calculated comparison without error and the tolerated error. Each non-leaf child can be viewed again as a sub-tree. The user can navigate freely through the entire tree, gradually looking for deeper explanations.
- A **What if**-explanation or hypothetical reasoning gives a hypothetical solution under modified premises. If the user rejects the deduced comparisons of step 3, he can change the first diagonal (step 2) and see how the comparisons of the upper diagonal are modified (example 5).

The explanation type given to the user depends on the type of error (Table 1).

**Conclusion and further work**

We have described the implementation of an expert module assisting the decision-maker in the construction of a consistent comparison matrix under AHP. The user can choose the comparison scale and the tolerated error. In our experience, the time to reach a consistent or near consistent matrix has considerably decreased with the help of the program. Further work will integrate this expert module in an Intelligent Tutoring System teaching the theoretical foundation of AHP.
Fig. 11. Trace of the transitivity calculation of formula (5).

Table 1
Explanation and error types

<table>
<thead>
<tr>
<th>Error 1</th>
<th>x</th>
</tr>
</thead>
<tbody>
<tr>
<td>Error 2</td>
<td>x</td>
</tr>
<tr>
<td>Error 3</td>
<td>x</td>
</tr>
<tr>
<td>Error 4</td>
<td>x</td>
</tr>
<tr>
<td>Error 5</td>
<td>x</td>
</tr>
<tr>
<td>Error 6</td>
<td>x</td>
</tr>
<tr>
<td>Error 7</td>
<td>x</td>
</tr>
</tbody>
</table>

References


