Institut für Produktion und Industrielles Informationsmanagement

Universität Duisburg-Essen / Standort Essen
Fachbereich 5: Wirtschaftswissenschaften
Universitätsstraße 9, 45141 Essen
Tel.: ++ 49 (0) 201 / 183 - 4007
Fax: ++ 49 (0) 201 / 183 - 4017

Arbeitsbericht Nr. 18

A heuristic algorithm to improve the consistency of judgments in the Analytical Hierarchy Process (AHP)

Dipl.-Kfm. Malte L. Peters

Univ.-Prof. Dr. Stephan Zelewski

E-Mail: { malte.peters | stephan.zelewski }@pim.uni-essen.de

Internet: http://www.pim.uni-essen.de/mitarbeiter

Essen 2003

Summary

The Analytical Hierarchy Process is a technique to solve multi-criteria decision problems requiring paired comparison judgments concerning the dominance of one element over another. This paper considers the problem of the consistency of judgments in matrices in the Analytical Hierarchy Process. A heuristic which adjusts an existing matrix with inconsistent judgments in an iterative process to improve the consistency of the judgments is presented.

Contents:

1. Introduction

The Analytical Hierarchy Process[1] is a technique to solve multi-criteria decision problems. It requires paired comparison judgments concerning the dominance of one element over another for each of *n* elements with respect to an element on the next higher level using a 1-9 scale. The paired comparison judgments are entered in a square matrix *A* of dimension *n*. For example, if an element *i* is judged to be *moderately important* by comparison with element *j* with respect to the common element on the next higher level, a 3 is entered as value for the paired comparison judgment a_{ij} in the matrix *A* while the reciprocal value is entered for the paired comparison judgment a_{ji}.

Saaty defines a matrix *A* to be consistent, if the following condition is satisfied[2]:

$$a_{ik} * a_{kj} = a_{ij} \qquad \forall\ i, j, k = 1, \ldots, n. \qquad (1)$$

For example, if $a_{23} = 2$ and $a_{34} = 3$ are fulfilled, in a consistent matrix the value of a_{24} has to be 6. The essential idea of the AHP is that a matrix *A* of rank *n* is only consistent if it has one positive eigenvalue $\lambda_{\max} = n$ while all other eigenvalues are zero[3]. Further, Saaty developed the *consistency index* (*CI*) to measure the deviation from a consistent matrix[4]:

$$CI = (\lambda_{\max} - n)/(n-1) \qquad (2)$$

1 Saaty (2000); Saaty (1994)

2 Saaty (2000), 48.

3 Saaty (2000), 54.

4 Saaty (2000), 84; Saaty (1994), 41.

The *consistency ratio* (*CR*) is introduced to aid the decision on revising the matrix or not. It is defined as the ratio of the *CI* to the so-called *random index* (*RI*) which is a *CI* of randomly generated matrices[5]:

$$CR = \frac{CI}{RI} \tag{3}$$

For $n = 3$ the required consistency ratio (CR^{Goal}) should be less than 0.05, for $n = 4$ it should be less than 0.08 and for $n \geq 5$ it should be less than 0.10 to get a sufficient consistent matrix[6]. Otherwise the matrix should be revised.

The heuristic algorithm presented in this paper is based upon Saaty's idea to change the judgments a_{ij}^{real} in an inconsistent evaluation matrix *A* with the greatest deviation to the $a_{ij}^{correct}$ calculated with the consistency condition (1).

---

5 Saaty (2000), 84; Saaty (1994), 42.

6 Saaty (2000), 85.

2. Heuristic algorithm to improve the consistency of judgments

The proposed heuristic requires as input an inconsistent matrix and delivers a sufficient consistent matrix. The rank order of the elements in the input matrix need not be the same as in the delivered matrix because the heuristic adjusts the paired comparison judgments in the matrix *A*. Thus the decision maker has to make up his mind, whether he wants to adopt the delivered matrix.

The algorithm is presented in pseudocode. Fig. 1 outlines the function "Find Greatest Deviation" which includes the part of the algorithm to compute the judgment a_{ij}^{real} in a matrix *A* with the greatest deviation to the $a_{ij}^{correct}$ calculated with the consistency condition. In every iteration three conditions are checked. The first one ensures that the deviation *cgmax* is the greatest of all deviations between the a_{ij}^{real} and $a_{ij}^{correct}$ which are stored in the array named *cg*. The second one ensures that the deviation *cgmax* is greater than all deviations calculated in the previous iterations. The purpose of the third condition is to avoid having diagonal elements chosen in the matrix because the *n* 1's on the matrix diagonal are for comparing the elements with themselves. Thus the index *i* has to be unequal to the index *j*. Moreover the appropriate values of the indices *i*, *k*, and *j* are saved to the variables *maxi*, *maxk*, and *maxj*, respectively, when the three conditions are satisfied.

FUNCTION *FIND_GREATEST_DEVIATION*

$h := 1$

$cgmax := 0$

FOR $i := 1, \ldots, n$

  FOR $j := 1, \ldots, n$

    FOR $k := 1, \ldots, n$

      $a_{ij}^{correct} := a_{ik} * a_{kj}$

      $cg(h) := \left| a_{ij}^{real} - a_{ij}^{correct} \right|$

        IF $cg\,(h) = \boldsymbol{MAX}\{cg(m) \mid m = 1,2,...,h\}$ ***AND*** $cg\,(h) > cgmax$ ***AND*** $i \neq j$ ***THEN***

          $cgmax := \boldsymbol{MAX}\{cg(m) \mid m = 1,2,...,h\}$

          $maxi := i$; $maxk := k$; $maxj := j$

        END IF

        $h := h + 1$

    NEXT k

  NEXT j

NEXT i

END FUNCTION

Fig. 1: Function Find Greatest Deviation

The part of the algorithm, given in Fig. 2, uses the variables *maxi*, *maxk*, and *maxj* to change the appropriate a_{ij}^{real}. The difference *cgrevise* between the judgment $a_{(\max i, \max j)}^{real}$ in a matrix and the judgement $a_{(\max i, \max j)}^{correct}$ calculated with the consistency condition indicates if $a_{(\max i, \max j)}^{real}$ has to be raised or reduced to improve the consistency of the judgments. If *cgrevise* is less than zero, $a_{(\max i, \max j)}^{correct}$ is greater than $a_{(\max i, \max j)}^{real}$. This means that $a_{(\max i, \max j)}^{real}$ has to be raised to improve the consistency of the judgments. The proposed algorithm raises $a_{(\max i, \max j)}^{real}$ by the value 1 and changes $a_{(\max j, \max i)}^{real}$ accordingly. Analogously if *cgrevise* is greater than zero, the algorithm reduces $a_{(\max i, \max j)}^{real}$ by the value 1 and changes $a_{(\max j, \max i)}^{real}$ accordingly.

The adjusted matrix is saved as the matrix A^{TEST}. Only if the consistency ratio $CR(A^{TEST})^{After\ Adjustment}$ is less than $CR(A)^{Before\ Adjustment}$ are the adjustments taken into the matrix A (See next page for $CR(A^{TEST})^{After\ Adjustment} \geq CR(A)^{Before\ Adjustment}$).

FUNCTION *MATRIX_ADJUSTMENT*

CALCULATE $CR(A)^{Before\ Adjustment}$

$a^{correct}_{(\max i, \max k)} := a^{real}_{(\max i, \max k)} * a^{real}_{(\max k, \max j)}$

$cgrevise := a^{real}_{(\max i, \max j)} - a^{correct}_{(\max i, \max j)}$

$A^{TEST} := A$

IF *cgrevise < 0* ***THEN***

$a^{TEST}_{(\max i, \max j)} = a^{real}_{(\max i, \max j)} + 1$

$a^{TEST}_{(\max j, \max i)} = 1/(a^{real}_{(\max i, \max j)} + 1)$

END IF

IF *cgrevise > 0* ***THEN***

$a^{TEST}_{(\max i, \max j)} = a^{real}_{(\max i, \max j)} - 1$

$a^{TEST}_{(\max j, \max i)} = 1/(a^{real}_{(\max i, \max j)} - 1)$

END IF

CALCULATE $CR(A^{TEST})^{After\ Adjustment}$

IF $CR(A^{TEST})^{After\ Adjustment} < CR(A)^{Before\ Adjustment}$ ***THEN***

$A := A^{TEST}$

$CR(A)^{Before\ Adjustment} := CR(A^{TEST})^{After\ Adjustment}$

END IF

END FUNCTION

Fig. 2: Function Matrix Adjustment

Fig. 3 shows a first version of the algorithm to improve the consistency of judgments in a matrix using the two projected functions. This algorithm adjusts the matrix until the consistency ratio $CR(A)^{Before\ Adjustment}$ falls below the chosen goal consistency ratio $CR^{Goal}(n)$.

FUNCTION *MAIN_1*

$CR(A)^{Before\ Adjustment} := 1$

WHILE $CR(A)^{Before\ Adjustment} \geq CR^{Goal}(n)$

    CALL FUNCTION *FIND_GREATEST_DEVIATION*

    CALL FUNCTION *MATRIX_ADJUSTMENT*

END WHILE

END FUNCTION

Fig. 3: Function Main 1

This version of the algorithm works with matrices of dimension $n = 3$. The algorithm has to be extended to make it work for matrices of dimension $n > 3$ because the reduction of the greatest deviation by the value 1 may make another deviation the greatest and may worsen the consistency of the judgments.

To make the algorithm work for matrices of all dimensions the function “Next Greatest Deviation” (shown in Fig. 4) is added to the algorithm. The function “Main 1” is replaced with the function “Main 2” (shown in Fig. 5). This function calls the function “Next Greatest Deviation” if the consistency ratio after the adjustment is equal to or greater than the consistency ratio before the adjustment. The function “Next Greatest Deviation” sets the greatest deviation to zero and finds the next greatest deviation in the array *cg*. If two or more identical maximum values (greatest deviations) occur in the array *cg*, analogously to the function “Find Greatest Deviation” the first one found is chosen. The values for the variables *maxi*, *maxj*, and *maxk* are calculated in same way as in the function “Find Greatest Deviation”. Then the function “Matrix Adjustment” is used to change the a_{ij}^{real} of the next greatest deviation. This process is repeated until the consistency is improved.

FUNCTION *NEXT_GREATEST_DEVIATION*

WHILE $CR(A^{TEST})^{After\ Adjustment} > CR(A)^{Before\ Adjustment}$

MAX $\{cg(m) \mid m = 1, 2, ..., h\} := 0$

FIND (***FIRST MAX*** $\{cg(m) \mid m = 1, 2, ..., h\}$)

GET *maxi, maxj, maxk*

CALL FUNCTION *MATRIX_ADJUSTMENT*

END WHILE

END FUNCTION

Fig. 4: Function Next Greatest Deviation

FUNCTION *MAIN_2*

$CR(A)^{Before\ Adjustment} := 1$

WHILE $CR(A)^{Before\ Adjustment} \geq CR^{Goal}(n)$

CALL FUNCTION *FIND_GREATEST_DEVIATION*

CALL FUNCTION *MATRIX_ADJUSTMENT*

IF $CR(A^{TEST})^{After\ Adjustment} \geq CR(A)^{Before\ Adjustment}$ ***THEN***

CALL FUNCTION *NEXT_GREATEST_DEVIATION*

END IF

END WHILE

END FUNCTION

Fig. 5: Function Main 2

3. A numerical example

A numerical example is presented to illustrate the proposed algorithm. In Fig. 6 a sample matrix A with a consistency ratio of $CR(A)$ = 0.3483 = 0.3866/1.11 is shown.

$$A = \begin{bmatrix} 1.0000 & 2.0000 & 2.0000 & 0.2500 & 0.5000 \\ 0.5000 & 1.0000 & 4.0000 & 6.0000 & 0.5000 \\ 0.5000 & 0.2500 & 1.0000 & 0.1111 & 0.1250 \\ 4.0000 & 0.1667 & 9.0000 & 1.0000 & 0.5000 \\ 2.0000 & 2.0000 & 8.0000 & 2.0000 & 1.0000 \end{bmatrix}$$

Fig. 6: Matrix before adjustment

The $CR(A)$ is calculated using equations (2) and (3) and the random index RI = 1.11 for n = 5. The algorithm undertakes 31 adjustments until the matrix $A^{'}$ in Fig. 7 with a consistency ratio of $CR(A^{'})$ = 0.0886 = 0.0984/1.11 is obtained.

$$A^{'} = \begin{bmatrix} 1.0000 & 0.1538 & 2.0000 & 0.2500 & 0.2500 \\ 6.5000 & 1.0000 & 18.0000 & 6.0000 & 0.5000 \\ 0.5000 & 0.0556 & 1.0000 & 0.1111 & 0.0625 \\ 4.0000 & 0.1667 & 9.0000 & 1.0000 & 0.5000 \\ 4.0000 & 2.0000 & 16.0000 & 2.0000 & 1.0000 \end{bmatrix}$$

Fig. 7: Matrix after adjustment

The $CR(A^{'})$ of 0.0886 which is lower than $CR^{Goal}(5) = 0.1$ indicates that the matrix is sufficient consistent. First the algorithm raises a_{23}^{real} by 1 to 5 because the consistency condition ($a_{24}^{real} * a_{43}^{real} = a_{23}^{correct}$) yields $a_{23}^{correct}$ = 6 * 9 = 54 and the function "Find Greatest Deviation" yields $cg = \left|a_{23}^{real} - a_{23}^{correct}\right| = |4 - 54|$ = 50 as the greatest deviation. In the next three iterations the deviation between a_{23}^{real} and $a_{23}^{correct}$ is also the greatest and a_{23}^{real} is raised to 7. If a_{23}^{real} is raised to 8 the consistency ratio worsens from 0.3402 to 0.3410. In this case the function "Main 2" calls the function "Next Greatest Deviation" which yields cg = $\left|a_{21}^{real} - a_{24}^{real} * a_{41}^{real}\right|$ = $|0.5 - 6 * 4|$= 23.5 as the next greatest deviation. Then a_{21}^{real} is raised to 1.5. The algorithm continues in this fashion until the consistency ratio $CR(A)^{Before\ Adjustment}$ falls below the goal consistency ratio $CR^{Goal}(5) = 0.1$ and thus the matrix is sufficient consistent.

4. Concluding remarks

This paper has demonstrated a heuristic algorithm to improve the consistency of paired comparison judgments in the Analytical Hierarchy Process. The algorithm has been implemented in the programmable mathematical software package Scilab[7]. The proposed algorithm has been applied to more than 50 inconsistent test matrices. It works for all test matrices.

[7] Scilab (2003)

References

Saaty (1994)

Saaty, T. L.: How to Make a Decision: The Analytic Hierarchy Process. In: Interfaces, Vol. 24 (1994), No. 6, S. 19-43.

Saaty (2000)

Saaty, T. L.: Fundamentals of Decision Making and Priority Theory with the Analytic Hierarchy Process. 2. Edition, Pittsburgh 2000.

Scilab (2003)

Scilab Group, Scilab 2.6, 2003 (Downloadable from website http://www-rocq.inria.fr/scilab/).