Math of AHP

The Math of AHP -- Computing priorities from pairwise comparisons

Eigenvector computation

Priorities for the elements in a cluster are derived from the relative pairwise comparisons by normalizing the principle right eigenvector of the matrix consisting of the pairwise comparisons. An insight into why this is so "natural" a computation can be obtained by considering *n* rocks of known weights w_1, w_2,.. w_n. We form a matrix of the ratio of the known weights of the rocks as follows:

$$\underline{A} = \begin{bmatrix} w_1/w_1 & w_1/w_2 & w_1/w_3 & -- & w_1/w_n \\ w_2/w_1 & w_2/w_2 & w_2/w_3 & -- & w_2/w_n \\ w_3/w_1 & w_3/w_2 & w_3/w_3 & -- & w_3/w_n \\ -- & -- & -- & -- & -- \\ -- & -- & -- & -- & -- \\ -- & -- & -- & -- & -- \\ w_n/w_1 & w_n/w_2 & w_n/w_3 & -- & w_n/w_n \end{bmatrix}$$

If we wanted to "recover" or find the vector of weights [w_1, w_2, w_3, ... w_n] given these ratios, we can take the matrix product of the matrix A with the vector *w* to obtain [1]:

$$\begin{bmatrix} w_1/w_1 & w_1/w_2 & w_1/w_3 & -- & w_1/w_n \\ w_2/w_1 & w_2/w_2 & w_2/w_3 & -- & w_2/w_n \\ w_3/w_1 & w_3/w_2 & w_3/w_3 & -- & w_3/w_n \\ - & - & - & -- & -- \\ - & - & - & -- & -- \\ - & - & - & -- & -- \\ w_n/w_1 & w_n/w_2 & w_n/w_3 & -- & w_n/w_n \end{bmatrix} * \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ - \\ - \\ - \\ w_n \end{bmatrix} = \begin{bmatrix} nw_1 \\ nw_2 \\ nw_3 \\ - \\ - \\ - \\ nw_n \end{bmatrix}$$

$\underline{A} \quad * \underline{w} = n\,\underline{w}$

If we knew A, but not *w*, we could solve the above for *w*. The problem of solving for a nonzero solution to this set of equations is very common in engineering and physics and is known as an eigenvalue equation:

$\underline{Aw} = \lambda\,\underline{w}$

The solution to this set of equations is, in general, found by solving an *n*th order equation for l. Thus, in general, there can be up to *n* unique values for l, with an associated *w* vector for each of the *n* values.

In this case, however, the matrix A has a special form since each row is a constant multiple of the first row. For such a matrix, the rank of the matrix is one, and all the eigenvalues of A are zero, except one. Since the sum of the eigenvalues of a positive matrix is equal to the trace of the matrix (the sum of the diagonal elements), the non-zero eigenvalue has a value of *n*, the size of the matrix. This eigenvalue is referred to as λ_{max}.

Notice that each column of $\underline{A}$ is a constant multiple of $\underline{w}$. Thus, $\underline{w}$ can be found by normalizing any column of $\underline{A}$.

The matrix $\underline{A}$ is said to be strongly consistent in that

$a_{ik}a_{kj} = a_{ij}$ for all i,j.

Now let us consider the case where we do *not* know $\underline{w}$, and where we have only estimates of the $a_{ij's}$ in the matrix $\underline{A}$ and the strong consistency property most likely does not hold. (This allows for small errors and inconsistencies in judgments.) It has been shown that for any matrix, small perturbations in the entries imply similar perturbations in the eigenvalues, thus the eigenvalue problem for the inconsistent case is:

$\underline{\mathbf{A}\boldsymbol{w}} = \boldsymbol{\lambda}_{\max}\underline{\boldsymbol{w}}$,

where $\boldsymbol{\lambda}_{\max}$ will be close to *n* (actually greater than or equal to *n*) and the other l's will be close to zero. The estimates of the weights for the activities can be found by normalizing the eigenvector corresponding to the largest eigenvalue in the above matrix equation.

Aside: The eigenvalue equation, when solved, produces one or more eigenvalues with associated eigenvectors. The word eigen, in German, means characteristic, or one's own. This is appropriate because eigenvalue equations arise naturally in many instances of mathematics and science -- on their own -- as a characteristic of the system itself. Eigenvalue equations are among the most frequent and universally occurring concepts in math and science; others include *pi*, the natural number *e*, fibonacci numbers, and the golden ratio. Even the Google search algorithm uses eigenvectors. (See 25 billion dollar eigenvector)

[1] The matrix product is formed by multiplying, element by element, each row of the first factor, $\underline{A}$, by corresponding elements of the second factor, $\underline{w}$, and adding. Thus, the first element of the product would be: $(w_1/w_1)*w_1 + (w_1/w_2)*w_2 + ...+ (w_1/w_n)*w_n = nw_1$. Similarly, the second element would be $(w_2/w_1)*w_1 + (w_2/w_2)*w_2 + ...+ (w_2/w_n)*$

$_n = nw_2$. The n^{th} element would be nw_n. Thus, the resulting vector would be $n\underline{w}$.