

NEW PRIORITY CALCULATIONS

Abstract

There are many methods of deriving priority vectors from pairwise comparison matrices, e.g. the standard largest eigenvector, the geometric mean, Harker's method, the least squares method, etc. Through our work on the SimpleAHP web application, we discovered an issue with most of these methods, that confused our users. The issue arises when there are two or more voters on a particular pairwise set, and those users have opposite votes on everything (we call them doppleganger voters). Given doppleganger voters one would expect the resulting priority sets to be inverses, or at least have reversed rankings. This is not always the case (this result has long been known, but the SimpleAHP web application made this idea more apparent).

While the geometric mean method does address this issue, it has its own shortcomings. Therefore in this paper we describe two new priority vector calculations that address doppleganger voters, while retaining the graph theoretic spirit of the eigenvector method. We compare and contrast the results of the priority calculations with some of the standard methods (eigenvector and geometric mean) on some differentiating examples. In addition we provide open source implementations of the new calculations in several languages (Python, R, Excel) in a free available github repository.

1 Introduction

The process of deriving a priority vector from a pairwise comparison matrix is a very well studied problem. However, our work creating the SimpleAHP web application (see Adams (2016a) for more information) brought to light some subtleties surrounding the standard largest eigenvector calculation. In the SimpleAHP web application, participants can provide symbolic votes of *better* and *much better*, and the facilitator can adjust the meaning of those symbolic votes. This structure allows for easy construction of doppleganger voters.

Definition 1 (Doppleganger voter pair). *Two voters on a pairwise comparison set are said to be dopplegangers if the pairwise comparison matrix of one user is the transpose of the other user.*

Doppleganger voters happened naturally with our participants because of the reduced amount of choice. However, a second natural way to construct doppleganger voters in the SimpleAHP web application is to change the meaning of *better* to be a number less than 1, and similarly for *much better*. Effectively we are making an inverse interpretation of what *better* and *much better* mean.

We constructed a doppleganger voter in this fashion on our first participant and attempted to verify our calculations by verifying that the doppleganger voter had reversed the order of the alternatives being compared. In fact we found that it did not reverse the order, and eventually we discovered this fact is indeed well known, (e.g. see Choo and Wedley (2004)). However this behavior of the largest eigenvector was troublesome for our target user, namely youth and those with no experience in AHP theory. Because this concept, of doppleganger voters having reversed priorities, we use the phrase the doppleganger property in the following sense.

Definition 2 (Doppleganger property). *We say that a priority vector calculation \mathcal{P} has the weak doppleganger property if, for all pairwise comparison matrices M the ranking of $\mathcal{P}(M)$ is the inverse ranking of $\mathcal{P}(M^T)$.*

We say that a priority vector calculation \mathcal{P} has the strong doppleganger property if, for all pairwise comparison matrices M , we have

$$\frac{1}{\mathcal{P}(M)} = \mathcal{P}(M^T).$$

In this paper we put forward two novel methods for calculating priority vectors from a pairwise comparison matrix. Both are adaptations of the standard largest eigenvector calculation, and both are the result of a limiting process very similar to the largest eigenvector's standard computational algorithm.

2 Literature Review

This work is heavily influenced by our work in Adams (2016a) where we create tools and techniques that allow youths, and those inexperienced in AHP, to nonetheless use AHP successfully on their own, without the need of facilitator or expert in the theory. Through that work we discovered some properties of the largest eigenvector priority vector calculation that confused our target audience (and in truth it confused us initially as well).

This behavior of the largest eigenvector, namely that the largest eigenvector of the transpose and the largest eigenvector of the original pairwise matrix need not be related, is well known and can be found in Choo and Wedley (2004) or Saaty (1990). However, because of the simplicity of changing symbolic votes in the SimpleAHP web interface, this property of the largest eigenvector is more readily observable and causes issues for those new to AHP theory. The geometric average is a standard method that remedies the transpose issue, but it has its own shortcomings (see Barzilai and Golany (1994) for details of using geometric averages of columns). Instead we opted to create two new eigenvector inspired, priority vector calculations, both of which respect the doppleganger property.

3 Hypotheses/Objectives

This paper defines two new priority vector calculations, algorithmically. We compare and contrast these new calculations with the standard eigenvector method and provide some examples that elucidate the differences in approach.

4 Research Design/Methodology

For this research, we develop programming libraries for several languages that implement the new algorithm. These libraries are freely available on our github page at Adams (2016b). In addition there is a publicly available gitter room for discussion of these algorithms and results (see Adams (2016c)).

The two new calculations are based upon the standard largest eigenvector calculation method. In particular we mean the following method for calculating the largest eigenvector.

Definition 3 (Largest eigenvector algorithm). *Let M be an $n \times n$ pairwise comparison matrix. The largest eigenvector is the limit of the sequence of vectors \mathbf{v}_k , where*

$$\begin{aligned}\mathbf{v}_0 &= (1, 1, \dots, 1) \\ \mathbf{v}_{k+1} &= \text{Normalize}(M \times \mathbf{v}_k) \\ \mathcal{E}(M) &= \lim_{k \rightarrow \infty} \mathbf{v}_k\end{aligned}$$

and $\text{Normalize}(\mathbf{v})$ means divide the vector \mathbf{v} by the sum of its entries. And by $M \times \mathbf{v}_k$ we mean standard matrix multiplication.

Our first new priority vector calculation, which we call the Geometric Eigenvector calculation, tweaks Definition 3 by changing the notion of “matrix multiplication” to that of geometric average. We can state the definition as follows.

Definition 4 (Geometric Eigenvector algorithm). *Let M be an $n \times n$ pairwise comparison matrix. The Geometric Eigenvector is the limit of the following sequence of vectors $\mathbf{v}_0, \mathbf{v}_1, \dots$ where*

$$\begin{aligned}\mathbf{v}_0 &= (1, 1, \dots, 1) \\ \mathbf{v}_{k+1} &= (M \hat{\boxtimes} \mathbf{v}_k) \\ \mathcal{GE}(M) &= \lim_{k \rightarrow \infty} \mathbf{v}_k\end{aligned}$$

where $M \hat{\boxtimes} \mathbf{v}$ is a column vector, whose i^{th} entry is the geometric average of the vector

$$(v_1 m_{i,1}, v_2 m_{i,2}, \dots, v_n m_{i,n})$$

Our second new priority vector calculation follows the outline of Definition 3, changing the matrix multiplication, this time doing a power weighting type calculation. This leads us to our Power Weighted Eigenvector calculation.

Definition 5 (Power Weighted Eigenvector). *Let M be an $n \times n$ pairwise comparison matrix. The Power Weighted Eigenvector is the limit of the following sequence of vectors $\mathbf{v}_0, \mathbf{v}_1, \dots$ where*

$$\begin{aligned}\mathbf{v}_0 &= (1, 1, \dots, 1) \\ \mathbf{v}_{k+1} &= (M^{\mathbf{v}_k}) \\ \mathcal{PWE}(M) &= \lim_{k \rightarrow \infty} \mathbf{v}_k\end{aligned}$$

where $M^{\mathbf{v}}$ is a column vector, whose i^{th} entry is the weighted geometric average i^{th} row of M by \mathbf{v} , in other words:

$$(M^{\mathbf{v}})_i = \left(\prod_{m_{ij} \neq 0} m_{ij}^{v_j} \right)^{1 / \sum_{m_{ij} \neq 0} v_j}$$

5 Data/Model Analysis

In the `Jupyter` directory of the github repository at Adams (2016b) we have Python based Jupyter workbooks that perform the standard eigenvector calculation as well as Definition 4 and Definition 5. In the Jupyter workbook `ComparingEigenToNewCalc.ipynb` in Adams (2016b) we define the following matrices to perform the new calculations on.

$$S = \begin{pmatrix} 1 & 1/3 & 3 & 1 \\ 3 & 1 & 1/3 & 3 \\ 1/3 & 3 & 1 & 1/9 \\ 1 & 1/3 & 9 & 1 \end{pmatrix} \quad E = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 1/2 & 1 & 5 & 6 \\ 1/3 & 1/5 & 1 & 7 \\ 1/4 & 1/6 & 1/7 & 1 \end{pmatrix}$$

$$M = \begin{pmatrix} 1 & 2 & 0.5 & 1 \\ 0.5 & 1 & 1 & 1 \\ 2 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix} \quad M' = \begin{pmatrix} 1 & 2 & 1/2 & 1/1.1 \\ 1/2 & 1 & 1 & 1 \\ 2 & 1 & 1 & 1 \\ 1.1 & 1 & 1 & 1 \end{pmatrix}$$

The above matrices have the following reason for their inclusion in this paper:

- S:** This was the first user vote that we tried the doppleganger property on, and discovered that the largest eigenvector did not have the that property on this vote set.
- E:** This was a first working example based on the matrix S , where we reduced the inconsistency.
- M:** A truly minimal working example showing the largest eigenvector priority vector calculation does not satisfy either the strong or weak doppleganger property.
- M':** A slightly less minimal version of the matrix M .

6 Limitations

7 Conclusions

8 Key References

- Adams, B. (2016a). Ahp for student decisions in a montessori elementary class.
- Adams, B. (2016b). Github ahpnewpriority project. <https://github.com/wjladams/ahpNewPriority>.
- Adams, B. (2016c). Gitter discussion room for the ahpnewpriority project. <https://gitter.im/wjladams/ahpNewPriority>.
- Barzilai, J. and Golany, B. (1994). Ahp rank reversal, normalization and aggregation rules. *Infor*, 32(2):57.

Choo, E. U. and Wedley, W. C. (2004). A common framework for deriving preference values from pairwise comparison matrices. *Computers & Operations Research*, 31(6):893–908.

Saaty, T. L. (1990). Eigenvector and logarithmic least squares. *European journal of operational research*, 48(1):156–160.

9 Appendix 1

	One ring to rule them all One ring to find them One ring to bring them all And in the darkness bind them
---	--