

FUZZY RANKING FROM FUZZY PAIRWISE COMPARISONS WITH
APPLICATIONS

by

YUNFENG WANG

August 2015

A Thesis Submitted
in Partial Fulfillment of
the Requirements for Degree of

MASTER OF SCIENCE

The Graduate Mathematics Program
Applied and Computational Mathematics Option
Department of Mathematics and Statistics
Texas A&M University-Corpus Christi

APPROVED: _____ Date: _____
Dr. George Tintera, Chair

Dr. Jose Guardiola, Member

Dr. Lei Jin, Member

Dr. George Tintera, Chair
Department of Mathematics and Statistics

Dr. Frank Pezold, Dean
College of Science and Engineering

Style: Proceedings of the American Mathematical Society

Table of Contents

1	Introduction	1
2	Basic Notions	3
2.1	Complete vs. Incomplete:	3
2.2	Crisp vs. Fuzzy	4
3	Known Methods	6
3.1	The Colley Method	6
3.2	The PageRank Algorithm	9
3.3	Saaty's Priority Ranking	14
4	Investigations and Results	20
4.1	A Fuzzy Version of the Colley Method	20
4.2	A Fuzzy version of the PageRank Algorithm	26
4.3	Incomplete Information in the Saaty's Priority Ranking	30
5	Conclusion	41
	References	42
	Appendix A	
	Fuzzy Arithmetic	45
	Appendix B	
	PageRank Code	47
	Appendix C	
	Completing Ranking Matrix	49
	Appendix D	
	Matlab Code: Calculating Error Matrix	51
	Appendix E	
	Matlab Code: Fuzzy Operation	52
	Appendix F	
	Completing Missing Data	53

Abstract

One method of ranking items is to score all of them against a standard scale. Sometimes it is difficult to create or use such a scale. As an alternative, it is possible to make side-by-side comparisons of some or all of the pairs of items. Then the problem is to convert the collective pairwise comparisons into a ranking. This problem has been studied previously in many contexts. This thesis addresses several methods, including where fuzzy comparisons are made for some, but not all, of the pairs. The Colley method and PageRank algorithm both use pairwise comparisons for some pairs to rank all items in a set. This thesis shows how those pairwise comparisons can be fuzzy. It also shows how Saaty's method for ranking alternatives can be completed when not all comparisons are used.

Acknowledgements

For the purpose of this study, I have learned how to design questions and to research the topics. I would like to thank many people here, especially the Chair of my research paper, Dr. George Tintera. No matter the week days and weekends, he guided me and directed me with all his patience to my paper, “Thanks” is not enough for me to tell him.

Dr. Jose Guardiola and Dr. Lei Jin, my thesis committee members, also contributed their time and knowledge to guide me to build up this research paper.

The paper would not be finished without the major support from my parents, Yufeng Wang and Xin Jiang. I want to dedicate this paper to my parents for their spiritual and financial support from the bottom of my heart.

Also, I want to thank many of my friends Dr. Charlene Tintera and Sulram Song. Without your inspiration, I would not have the encouragement to finish this paper. Thanks for all the people who helped me during this long process to get my master degree in Mathematics.

1 Introduction

When a job opens, many applications will be ranked for intermediate and final consideration. A sports fan wants to rank teams in a league. A student wishes to rank prospective schools. A internet user wants to rank sites found on the web. This paper will show how partial comparisons and fuzzy math methods can be used to convert comparisons to rankings of the various items.

One method of ranking items is to score them against a standard scale. Sometimes it is difficult to create or use such a scale. In that case, it may be better to compare pairs of items to each other. Then there is the problem of converting the collective pairwise comparisons into a total ranking.

To restate the problem in terms of general objects, if S is a collection of objects $\{A_1, \dots, A_n\}$, we want to assign a rating r_i to each A_i . The result is a total ranking in which there is a top ranked object A_i with $r_i \geq r_j$, all j . A consequence of the rating is a comparison

$$v_{ij} = \begin{cases} 1, & r_i > r_j \\ 0, & r_i = r_j \\ -1, & r_i < r_j \end{cases}$$

for any pair A_i, A_j .

Alternatively, it might start with some pairwise comparisons, say $A_i > A_j$ and record them with values v_{ij} . Then the problem is how to combine the pairwise comparisons into a ranking. This problem is well known and has many applications. We propose to study the case where not all pairwise comparisons have been made and cases where the values of those comparisons are fuzzy.

Wesley Colley created the Colley Method in 2002 and this method is famously used to rank college football teams. Not all teams need to play each other to generate a total ranking[16].

PageRank was originally created for counting and ranking websites by one of the founders of Google Company. This method analyzes and evaluates every web page and then gives them a ranking score[7].

Saaty's Priority Ranking theory uses pairwise comparisons of all objects to make a ranking of those objects. The solution uses the eigenvalue equation $A*w = n*w$ and complete crisp data to evaluate intangible things. The analysis also finds consistency of the individual judgements in the pairwise comparisons[1].

2 Basic Notions

2.1 Complete vs. Incomplete:

In order to record the results of comparing objects, data will be arranged into a matrix. Under this system, each pair of objects will be compared and scored easily. Comparisons that are made with complete data will be assigned as P while the missing comparisons assigned as M , Since every object is identical to itself, the comparison is recorded as I .

The ideal case is where objects all pairwise comparisons between the objects A , B , and C can be made. The ideal matrix for complete information is assigned as matrix (1) .

$$\begin{array}{c} \begin{array}{ccc} & A & B & C \\ A & \left(\begin{array}{ccc} I & P & P \\ P & I & P \\ P & P & I \end{array} \right) \\ B \\ C \end{array} \end{array} \quad (1)$$

It is possible that two resumes have very different strengths and weaknesses, two teams in a league have not played with each other, or two prospective schools that have not yet been compared. In some cases, it is possible that items A and B and items B and C can be compared, but item C cannot be compared with A . This incomplete comparisons of a matrix will be record for all present and missing data in matrix (2).

$$\begin{array}{c} \begin{array}{ccc} & A & B & C \\ A & \left(\begin{array}{ccc} I & P & M \\ P & I & P \\ M & P & I \end{array} \right) \\ B \\ C \end{array} \end{array} \quad (2)$$

Specific algorithms have different methods for recording the missing comparisons. The Colley and PageRank algorithms essentially use the value of 0 to record missing comparisons. This study will use blanks to indicate missing information in the Saaty's Priority Ranking.

2.2 Crisp vs. Fuzzy

Yager(1981) proposed the purpose of ordering fuzzy comparison for the quantities is to analyze either discrete or continuous data. He also suggested that the objects are separated by ranking the data between 0 and 1. The first use of fuzzy information was to generalize set membership. In exactly the same way, Yager generalized comparisons from being true or false to relative numbers r , $0 \leq r \leq 1$.

$$v(A > B) = \begin{cases} 1 & \text{if } True \\ 0 & \text{if } False \end{cases}$$

Fuzzy comparisons express a lack of precision. Yager's original interpretation was to let the value be a degree of comparison. With that, write $v(A > B) = x$ where $0 \leq x \leq 1$ The value of 0 or 1 is crisp and otherwise is fuzzy

As a second interpretation, we can let the comparison f be a fuzzy number. Specifically, write $v(A > B) = f$, where $f = (l, m, r)$ where m is the real number of greatest weight of the comparison and the interval (l, r) includes all weights, decreasing at the endpoints.

Here is one example to help understand the difference between crisp and fuzzy. Two teams, Carolina and New Orleans, from the National Football League played twice. Carolina won the first game, so the result is recorded as 1 in the Table 1. Carolina lost the second game, and the result is recorded as 0.

In addition to fuzzy ratings, it is possible to have fuzzy numbers, along with appropriate rules for arithmetic. This makes it possible to emulate the calculations

Table 1: NFL Crisp Example

Win Team		Lose Team	Crisp
Carolina	VS.	New Orleans	1
New Orleans	VS.	Carolina	0

done with crisp numbers. Following [8], a triangular fuzzy number is a triple of real numbers (l, m, r) representing a piecewise-linear membership function with $f(x) = 0$ for $x \leq l$ and $x \geq r$ and $f(m) = 1$.

We can make fuzzy comparisons using the score difference for the same two games. In the first game, Carolina beat New Orleans by a score difference of only 3 points. With a small difference, the middle number is closer to 0 than 1: 0.3. In the second the, score difference is much bigger, 25 points. With that large difference, the middle number is closer to 1 than 0: 0.8. Finally, 0.1 is added and subtracted from the middle number to get the left and right endpoints for the fuzzy numbers.

Table 2: NFL Fuzzy Example

Win Team	Score	Lose Team	Score	(l,m,r)
Carolina	16	New Orleans	13	(.2,.3,.4)
New Orleans	31	Carolina	6	(.7,.8,.9)

3 Known Methods

3.1 The Colley Method

Wesley Colley created the Colley Method in 2002 for computing rankings of the National Football League (NFL) teams. The model only considers the count of the wins and losses for each team and the number of games each team plays. Since not all teams play each other, there is incomplete data in this method.

The data and ratings of the teams are recorded in the linear system $Cr = b$ [16], where b records the net number of wins and losses, C records the record of number of games played and $r = C^{-1}b$ shows the ratings of the teams. Teams with higher ratings are ranked higher. Notation for the method is given below.

- n : number of teams to be rank = order of C .
- $r_{n \times 1}$: general rating vector produced by the Colley system is $[r_1, r_2, \dots, r_n]^t$
- t_i : total number of games played by team i
- n_{ij} : number of times team i faced team j
- C is an $n \times n$ matrix with $C_{ij} = \begin{cases} 2 + t_i & i = j \\ -n_{ij} & i \neq j \end{cases}$
- w_i : total number of wins accumulated by team i
- l_i : total number of losses accumulated by team i
- $b_{n \times 1}$: $n \times 1$ matrix on the right with $b_i = 1 + \frac{1}{2}(w_i - l_i)$

$$r = C^{-1}b \tag{3}$$

The original example (Table 4) C is

$$C = \begin{pmatrix} 5 & 0 & -1 & -2 & 0 & 0 \\ 0 & 5 & 0 & 0 & -1 & -2 \\ -1 & 0 & 4 & -1 & 0 & 0 \\ -2 & 0 & -1 & 6 & -1 & 0 \\ 0 & -1 & 0 & -1 & 6 & -2 \\ 0 & -2 & 0 & 0 & -2 & 6 \end{pmatrix}$$

$$C^{-1} = \begin{pmatrix} 0.2607 & 0.0087 & 0.0917 & 0.1059 & 0.0226 & 0.0104 \\ 0.0087 & 0.2638 & 0.0068 & 0.0183 & 0.0859 & 0.1166 \\ 0.0917 & 0.0068 & 0.2935 & 0.0824 & 0.0176 & 0.0081 \\ 0.1059 & 0.0183 & 0.0824 & 0.2237 & 0.0477 & 0.0220 \\ 0.0226 & 0.0859 & 0.0176 & 0.0477 & 0.2233 & 0.1030 \\ 0.0104 & 0.1166 & 0.0081 & 0.0220 & 0.1030 & 0.2399 \end{pmatrix}$$

$$b = \begin{pmatrix} 1/2 \\ 3/2 \\ 2 \\ 0 \\ 1 \\ 1 \end{pmatrix} \quad r = C^{-1}b = \begin{pmatrix} 0.3597 \\ 0.6160 \\ 0.6687 \\ 0.3149 \\ 0.5015 \\ 0.5392 \end{pmatrix}$$

and get ranking r .

In making these calculations, it is apparent that C and its inverse always have nonnegative entries. In fact, they turn out to be nonnegative matrices. We say, a matrix A is nonnegative, or $A \geq 0$, if all its entries are nonnegative, and say a matrix A is positive, $A > 0$, if all of its entries are positive [13]. The results we need follow a

theorem proved by Oscar Perron as well as the Gershgorin Circle Theorem [14]. The PageRank Algorithm is also an application of Perron's Theorem.

Following [13], we denote the following for a square matrix, A :

- $\rho(A) = \max \{ \lambda; \lambda \text{ is an eigenvalue of } A \}$ and
- if there is more than one eigenvalue for A , $\mu(A) = \max \{ \lambda; \lambda \text{ is an eigenvalue of } A, \lambda \neq \rho(A) \}$.
- $\rho(A)$ is called the spectral radius of (A) .

Perron's Theorem [13] can now be stated as below.

Theorem: If A is a square positive matrix then

- $\rho(A) > 0$,
- $\rho(A)$ is a simple eigenvalue of A ,
- to $\rho(A)$ corresponds a positive eigenvector,
- $\mu(A) < \rho(A)$,
- $\lim_{(m \rightarrow \infty)} (A/\rho(A))^m \equiv L \equiv xy^T$, where $Ax = \rho(A)x, x > 0; A^T y = \rho(A)y, y > 0; x^T y = 1$,
- for every $r, \mu(A)/\rho(A) < r < 1$, there exists a constant $C = C(r, A)$, such that for every $m, \|(A - \rho(A))^m - L\|_\infty \leq Cr^m$, where $\|A\|_\infty = \max |a_{ij}|$.

To complete the demonstration that C^{-1} is non-negative, Govan [7] applies Gershgorin's Theorem [2] using the fact that the Colley matrix C is diagonally dominant $|C_{ii}| = |1 + n_i| > \sum_{(j \neq i, j=1)}^n |C_{ij}| = n_i$.

Gershgorin's Theorem: Every eigenvalue of matrix A_n satisfies: $|\lambda - A_{ii}| \leq \sum_{j \neq i} |A_{ij}| ; i \in 1, 2, \dots, n$

Because C is nonsingular and symmetric matrix all eigenvalues are real, so by the Gershgorin's Theorem "all eigenvalues of C must be real and positive making C a Stieltjes matrix, which is a symmetric nonsingular M -matrix" [7]. If matrix X satisfy every eigenvalue of X is positive and $X \in M = (M_{ij}) \in R^{n \times n} M_{ij} \leq 0, i \neq j$, then this matrix name is nonsingular M matrix.

Therefore, the matrix C^{-1} is nonnegative. Since C^{-1} is not negative, the ranking r will depend on the information b , if b is nonnegative, then ranking r will be nonnegative.

Theorem: The vector r , solution to the $Cr = b$ exists, is unique, and nonnegative.

3.2 The PageRank Algorithm

The internet plays a big role in our daily life as a source of a vast quantity of information. Search engines, including Google, help us find information on a daily basis. However, do you know how this search engine works? In 1998, two popular search algorithms were introduced, the Hypertext Induced Topic Search(HITS) by Jon Kleinberg and PageRank by Sergey Brin and Larry Page, the future founders of Google. Brin and Page use their method to rank every web page by giving each page a rating score.

In this study, we will change the crisp comparisons of web pages to fuzzy comparison and then give them ranking score. We will demonstrate that the Power method can still be applied to the modified Google matrix to get a ranking.

Consider the example from Langville and Meyer [15] in Figure 1. It is the graph of a tiny web, or network, in which nodes are linked to one another as follows. There are 6 nodes which are linked as node 1 can go to 2 and 3, node 2 cannot go to other node, and node 3 can go back 1, 2, and 5. Node indicated by the arrows. For example node 2 and 3 are linked from node1. It is important to note that all nodes in this example are linked in some way.

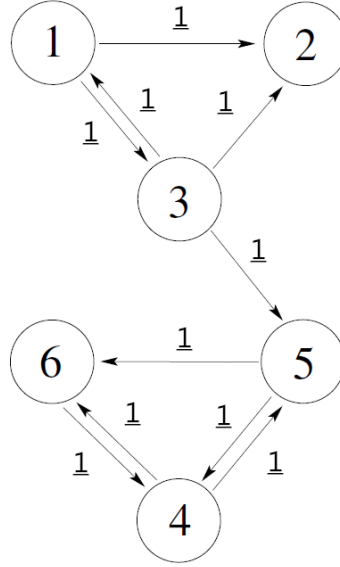


Figure 1: Tiny web example from Langville and Meyer[15]

Matrices are used to represent information and calculation with them is a primary tool used in the PageRank Algorithm. Following [7], the first step in the PageRank algorithm is to record the links from the graph in an adjacency matrix. The links in Figure 1 are recorded in the matrix A below using the definition that follows.

$$A_{ij} = \begin{cases} 1 & \text{if there is a link from } P_i \text{ to } P_j \\ 0 & \text{otherwise} \end{cases}$$

$$A = \begin{matrix} & \begin{matrix} 1 & 2 & 3 & 4 & 5 & 6 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{matrix} & \begin{pmatrix} 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix} \end{matrix}$$

The next step is to calculate the degree or weight of a node, or the number of links

starting at that node. Let node P_i have a weight equal to $deg^-(i)$ which is a total number of links from the original node P_i to other nodes. That is, $deg^-(i) = \sum_j A_{ij}$. In this example $deg^-(1) = 2$ because the node 1 goes to 2 and 3. As in [7], we create a new matrix H using the definition below.

$$H_{ij} = \begin{cases} 1/deg^-(i) & \text{if there is a link from } P_i \text{ to } P_j \\ 0 & \text{otherwise} \end{cases}$$

As you can see below the matrix H is not stochastic because the second row is all zeros. This is because of there not being any links from node 2.

$$H = \begin{matrix} & \begin{matrix} 1 & 2 & 3 & 4 & 5 & 6 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{matrix} & \begin{pmatrix} 0 & 1/2 & 1/2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1/3 & 1/3 & 0 & 0 & 1/3 & 0 \\ 0 & 0 & 0 & 0 & 1/2 & 1/2 \\ 0 & 0 & 0 & 1/2 & 0 & 1/2 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix} \end{matrix}$$

The third step, is to create a stochastic matrix from H . Define the vector a by a_i is zero if row i of A is nonzero and a_i is 1 otherwise [7]. Also, let e be the column of all ones. Then replace the i th row of matrix H with $(1/n)ae^T$. After replacing the matrix H will become stochastic matrix S

$$a_i = \begin{cases} 1 & \text{if } H_i^T = 0 \\ 0 & \text{otherwise} \end{cases}$$

and

$$S = H + \frac{1}{n}ae^T \tag{4}$$

In the case of our example with the 2nd row of H being zero, we would use $a = [0 \ 1 \ 0 \ 0 \ 0 \ 0]^T$ and $e^T = [1 \ 1 \ 1 \ 1 \ 1 \ 1]$. The resulting Stochastic matrix S is below.

$$S = \begin{matrix} & \begin{matrix} 1 & 2 & 3 & 4 & 5 & 6 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{matrix} & \begin{pmatrix} 0 & 1/2 & 1/2 & 0 & 0 & 0 \\ 1/6 & 1/6 & 1/6 & 1/6 & 1/6 & 1/6 \\ 1/3 & 1/3 & 0 & 0 & 1/3 & 0 \\ 0 & 0 & 0 & 0 & 1/2 & 1/2 \\ 0 & 0 & 0 & 1/2 & 0 & 1/2 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix} \end{matrix}$$

The last step is to make sure the Stochastic matrix S is irreducible. According to [7], “A nonnegative matrix is called irreducible if and only if the corresponding directed graph $G(A)$ is strongly connected” [7]. There are many ways to get result, but only consider about the simplest way like author Govan paper wrote that “make a rank 1 update to S and ensure that the result stays stochastic by making the sum into a convex combination” [7]. Therefore, the Google matrix G is

$$G = S\alpha + (1 - \alpha)ev^T \quad for \quad 0 < \alpha < 1 \quad (5)$$

which $e^T = [1 \ 1 \ 1 \ 1 \ 1 \ 1]$ and $v^T = [1/n \ 1/n \ 1/n \ 1/n \ 1/n \ 1/n]$.

At here e is a vector value 1 and v is a probability distribution vector which $v > 0$. The resulting G is what is generally called the Google matrix.

$$G = \begin{matrix} & \begin{matrix} 1 & 2 & 3 & 4 & 5 & 6 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{matrix} & \left(\begin{array}{cccccc} \frac{1-\alpha}{n} & \frac{\alpha}{2} + \frac{1-\alpha}{n} & \frac{\alpha}{2} + \frac{1-\alpha}{n} & \frac{1-\alpha}{n} & \frac{1-\alpha}{n} & \frac{1-\alpha}{n} \\ \frac{\alpha}{6} + \frac{1-\alpha}{n} & \frac{\alpha}{6} + \frac{1-\alpha}{n} & \frac{\alpha}{6} + \frac{1-\alpha}{n} & \frac{\alpha}{6} + \frac{1-\alpha}{n} & \frac{\alpha}{6} + \frac{1-\alpha}{n} & \frac{\alpha}{6} + \frac{1-\alpha}{n} \\ \frac{\alpha}{3} + \frac{1-\alpha}{n} & \frac{\alpha}{3} + \frac{1-\alpha}{n} & \frac{1-\alpha}{n} & \frac{1-\alpha}{n} & \frac{\alpha}{3} + \frac{1-\alpha}{n} & \frac{1-\alpha}{n} \\ \frac{1-\alpha}{n} & \frac{1-\alpha}{n} & \frac{1-\alpha}{n} & \frac{1-\alpha}{n} & \frac{\alpha}{2} + \frac{1-\alpha}{n} & \frac{\alpha}{2} + \frac{1-\alpha}{n} \\ \frac{1-\alpha}{n} & \frac{1-\alpha}{n} & \frac{1-\alpha}{n} & \frac{\alpha}{2} + \frac{1-\alpha}{n} & \frac{1-\alpha}{n} & \frac{\alpha}{2} + \frac{1-\alpha}{n} \\ \frac{1-\alpha}{n} & \frac{1-\alpha}{n} & \frac{1-\alpha}{n} & \frac{1-\alpha}{n} & \frac{1-\alpha}{n} & \frac{1-\alpha}{n} \end{array} \right) \end{matrix}$$

For this example, the n is 6 and $\alpha(\alpha)$ will be given. Brin and Page [3] originally used $\alpha = 0.85$. As a check of the stochastic condiation for the Google matrix G : the sum across row 1: $6 * (1 - \alpha)/6 + 2 * (\alpha/2) = 1 - \alpha + \alpha = 1$. Therefore, the dominant eigenvalue of the matrix G is 1.

With G being a stochastic matrix, the rating vector π is a positive left eigenvector of G for with the dominant eigenvalue 1. The simplest way to compute π is to apply the power method to G [17]. That is to start with a postive vector x and iterate the product $G^T x, (G^T)^2 x, \dots$. The sequence will converge to a rating vector π . That is to say, the PageRank ranking is solved by the Power method [10]:

- Pick an initial guess $x^0, ||x^0||_1 = 1, k = -1$
- Repeat $[x^{k+1}]^T = [x^k]^T G$, until “termination criterion satisfied” $||x^{k+1} - x^k|| \leq \tau$.

The the parameter τ often lies between 10^{-8} and 10^{-4} .

As in [10], we use the Power method to solve PageRank because it is simple to implement. Second, the calculations do not have overflow and are accurate and reliable. Third, the rate of convergence does not depend on the matrix dimension. Next, few vectors have minimal storage. Finally, the convergence behavior is really strong (robust).

3.3 Saaty's Priority Ranking

In the last two sub-sections, we described methods for generating rankings from incomplete data. In this sub-section, we will review for Saaty's Priority Ranking. It uses complete, pairwise information to generate rankings. It differs from the other methods presented since it uses human judgements as data instead of objective information. The general conditions for Saaty's Priority Ranking [1] can be used with crisp or fuzzy comparisons. Before presenting the conditions, we review some background about judgements and appropriate scales.

People making comparisons normally find a scale to evaluate each individual item and then compare the scale numbers. However, not all attributes can be measured with a scale to use this way to make comparisons. Examples include people's mind, judgment, love, emotion and other intangible things. The measure of intangible things cannot absolutely set because the opinion and value will change from one person to another. If you want to use ruler and scale to this, you should find common standard. Otherwise, there always someone will not agree with you. Therefore, the only things we can do is let this intangible object compare side by side. Then find which one is better or higher.

Many mathematicians have thought about the problem of measurement. The paper of Saaty [1] cited the great mathematician Henri Lebesgue [11]: "It would seem that the principle of economy would always require that we evaluate ratios directly and not as ratios of measurements. However, in practice, all lengths are measured in meters, all angles in degrees, etc.; that is we employ auxiliary units and, as it seems, with only the disadvantage of having two measurements to make instead of one. Sometimes, this is because of experimental difficulties or impossibilities that prevent the direct comparison of lengths or angles. But there is also another reason". That reason is the subject of Saaty's Priority Ranking and is discussed in this paper.

If the object is intangible, how can we measure those? What scale can we give

to measure the intagibles? Saaty's paper develops a relative measure to compare two objects in mind. This relieves us of the need to have zero and negative units.

Because the fundamental scale can be define different meaning for every situation, so there are going to use same table with author Saaty's section 4. Given two objects, i and j , we cannot direclty know the weights those objects have in a ranking, w_i and w_j , respectively. We humans can compare the two otherwise intangible objects, by a relative ratio recording the ratio of the weights of the two elements, $A_{ij} = w_i/w_j$ instead of measuring the two individual weights directly. The Table 5 fundamental scale from 1 to 9 means the ratio $(w_i/w_j)/1$ For example, if we have two objects, 1 and 2, and write $A_{12} = 5$ to mean that the importance of object 1 is 5 times that of the importance of object 2, or the intensity of the importance of object 1 over object 2 is 'strong.'

Table 3: Fundamental Scale

Intensity of Importance	Definition	
1	Equal Importance	Two activities contribute equally to the objective
2	Weak or slight	
3	Moderate importance	Experience and judgment slightly favor one activity over another
4	Moderate plus	
5	Strong importance	Experience and judgment strongly favor one activity over another
6	Strong plus	
7	Very strong	An activity is favored very strongly over another; its dominance demonstrated in practice
8	Very,very strong	
9	Extreme importance	The evidence favoring one activity over another is of the highest possible order of affirmation
1.1-1.9	When activities are very close a decimal is added to 1 to show their difference as appropriate	
Reciprocals of above	If activity i has one of the above nonzero numbers assigned to it when compared with activity j , then j has the reciprocal value when compared with i	A logical assumption
Measurements from ratio scales		When it is desired to use such numbers in physical applications. Alternatively, often one estimates the ratios of such magnitudes by using judgment

Like the author Saaty's paper, we assume each element weight will not be zero. Also, any pairwise comparison must be a positive ratio because the weights are both

positive. Also, the ratio cannot be zero, because the reciprocal must also be a ratio, and zero has no meaningful reciprocal.

Now we are in a position to give details on how Saaty mathematically formats and solves his Saaty's Priority Ranking. This is done with the criteria below.

- n , positive integer greater than 1 is the number of objects to ranked.
- The ranking is a positive vector $w = [w_1, w_2, \dots, w_n]^t$
- Ranking w determines a ranking matrix A , with elements $A_{ij} = w_i/w_j$.
- $A_{ij} > 0$, for all $1 \leq i, j \leq n$
- $A_{ij} = 1/A_{ji}$, for all $1 \leq i, j \leq n$
 - Special case: $A_{ii} = 1$, for all $i, 1 \leq i \leq n$
- $A_{ij} \times A_{jk} = A_{ik}$, for all $1 \leq i, j, k \leq n$
- $Aw = nw$

Note that the last line means that the ranking vector, w can be recovered from A as the eigenvector corresponding to the eigenvalue, n . We illustrate the properties and recovery in the 2x2 case.

$$A_{2 \times 2} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} = \begin{bmatrix} 1 & \frac{w_1}{w_2} \\ \frac{w_2}{w_1} & 1 \end{bmatrix}$$

$$A * w = n * w \tag{6}$$

$$\begin{bmatrix} 1 & \frac{w_1}{w_2} \\ \frac{w_2}{w_1} & 1 \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} = 2 \begin{bmatrix} w_1 \\ w_2 \end{bmatrix}$$

In the previous discussion, the matrix of pairwise weights is $A_{ij} = w_i/w_j$ and linear algebra equation is $Aw = nw$. Solving the equation and finding the weights w is this paper's purpose. In the Saaty Priority Ranking, the condition $A_{ij} \times A_{jk} = A_{ik}$ enforces consistency among the entries. It is common, however for human judgement and the use of Saaty's scale to not be perfectly consistent. The most we can expect of a priority matrix A is that $A_{ij} = 1/A_{ji}$, for all $1 \leq i, j \leq n$. A matrix satisfying this condition is called a 'reciprocal matrix.' However, before solving eigenvalues, the problem need check the consistency of a matrix. We will call a reciprocal matrix 'consistent' if it satisfies the condition $A_{jk} = A_{ik}/A_{ij}$, or expressed another way, $A_{ij} * A_{jk} = A_{ik}$. Here are two examples to illustrate consistency.

A Consistent Matrix

$$\begin{array}{c} A \quad B \quad C \\ A \left(\begin{array}{ccc} 1 & 2 & 2/3 \\ 1/2 & 1 & 1/3 \\ 3/2 & 3 & 1 \end{array} \right) \\ B \\ C \end{array}$$

This 3-by-3 matrix, satisfies the condition $A_{12} * A_{23} = A_{13}$ and $A_{23} * A_{31} = A_{21}$ demonstrating that the matrix is consistent.

An Inconsistent Matrix:

$$\begin{array}{c} A \quad B \quad C \\ A \left(\begin{array}{ccc} 1 & 2 & 4/5 \\ 1/2 & 1 & 1/3 \\ 5/4 & 3 & 1 \end{array} \right) \\ B \\ C \end{array}$$

In this example, the $A_{12} * A_{23} \neq A_{13}$ and $A_{23} * A_{31} \neq A_{21}$. So this matrix is not consistent.

To measure the lack of consistency of a reciprocal matrix, Saaty created a Consistency ratio (C. R.)[20] of a pairwise comparison matrix as the ratio of its consistency index μ to the corresponding random index value as in Saaty's paper. According to

the Saaty's paper, allowable consistency ratio should not be more than 10 percentage and also cannot less than 1 percentage or 0.1 percentage without trivializing the impact of inconsistency.

$$\text{Consistency index } \mu \equiv (\lambda_{max} - n)/(n - 1) \quad (7)$$

In this study, if a matrix has a tolerable consistency ratio, then the matrix is meaningful. So, if a matrix is inconsistent, one must find the consistency index μ and determine this matrix is meaningful or not.

Finally, we note, Saaty's Priority Ranking can be used in either the fuzzy or crisp case. Use linear algebra to solve eigenvector w . In the case that the entries of A are fuzzy, the vector w is found as in a paper by Broek and Noppen [1]. The four steps in that paper are

- Step 1: Obtain triangular fuzzy numbers (l_{ij}, m_{ij}, r_{ij}) for the logarithms of the fuzzy pairwise comparisons A_{ij} for $1 \leq j < i \leq N$
- Step 2: Solve the linear programming problem (using the simplex algorithm) to maximize α with variables α and $u_i (2 \leq i \leq N)$ and the following constraints:
 - $u_i - u_j \leq r_{ij} - \alpha(r_{ij} - m_{ij})l_{i1} - \alpha(m_{i1} - l_{i1}) + l_{j1} + \alpha(m_{j1} - l_{j1})$ for all $2 \leq j < i \leq N$
 - $u_i - u_j \geq l_{ij} - \alpha(r_{ij} - m_{ij})l_{i1} - \alpha(m_{i1} - l_{i1}) + l_{j1} + \alpha(m_{j1} - l_{j1})$ for all $2 \leq j < i \leq N$
 - $u_i \leq r_{i1} - l_{i1} - \alpha(r_{i1} - l_{i1})$ for all $2 \leq i \leq N$
- Step 3: $w_1 = 1$ and $w_i = \exp(u_i + l_{i1} + \alpha(m_{i1} - l_{i1}))$ for all $2 \leq i \leq N$
- Step 4: normalize w

4 Investigations and Results

There are many methods applied to the ranking problem. Examples include the PageRank method, the Hyperlink-Induced Topic search model, the Colley Matrix method, the Keener Ranking model, and Massey Least Squares Rating. This paper only focuses on fuzzy versions of the Colley method and the PageRank method and completing information for the Saaty's Priority Ranking.

4.1 A Fuzzy Version of the Colley Method

This section provides an opportunity to generalize a crisp incomplete method. The Colley method as described in Section 3.1 uses incomplete, crisp pairwise comparisons to rank objects. We will show how to adapt the method to use incomplete, fuzzy pairwise comparisons to rank objects .

The Colley method relies on the outcomes of games in the form of wins and losses for each team. Think of that as the accumulation of comparisons between that team and the others that it has played. We will apply the method to other situations, such as pairwise comparison of resumes. Where the result of the comparisons are fuzzy. Recall that the record of wins and losses for team i is recorded at b_i in the formula (8). We will adapt b_i to express fuzzy information.

$$b_i = 1 + \frac{1}{2}(w_i - l_i) \quad (8)$$

The Matrix C will remain a crisp number. Then we can solve ranking $r = C^{-1}b$ and consider whether the product $C^{-1}b$ is meaningful.

We have considered the case that fuzzy comparisons could be a degree x and x between 0 and 1. We will review it for completeness sake to indicate how it could also be used with the method of this section. For this section, we will make comparisons using fuzzy numbers (l, m, r) or $(m - \Delta, m, m + \Delta)$ in the vector b in the Colley Method.

Having done that, we need to consider whether the product $C^{-1}b$ is meaningful.

Situation One: Lets assume $A > B$ and $v(A > B) = x$ where x is the degree of comparison and $0 \leq x \leq 1$. It can write $w = x$ and $l = -x$. Find each comparison x , then it can generalize the Colley method as

$$b_i = 1 + \frac{1}{2}(\sum w_i + \sum l_i) \quad (9)$$

where the sum is indexed over all possible comparisons between object i and other objects.

Situation Two: Lets assume $A > B$ and $v(A > B) = (a, b, c)$ fuzzy number f . Then $v(B > A)$ is $-f, (-c, -b, -a)$. Which $c > b > a$, find $f + (-f) = (a - c, 0, c - a)$. Generalizing the Colley method to get b_i

$$b_i = 1 + \frac{1}{2}(\sum f(w_i) + \sum f(l_i)) \quad (10)$$

Again, the sum is indexed over all comparisons of object i with other objects. As in Section 3.1, the matrix C and its inverse are positive matrices. All that remains is to determine whether $r = C^{-1} * b$ is meaningful or not.

For the Colley matrix method, there has no bias toward conference, no pre-season poll, history, etc. This method uses a minimum of assumptions and no *ad hoc* adjustments. Nonetheless, it adjusts for strength of schedule, ignores runaway scores, and produces common sense results [4]. It can be reproducible.

Colley's ranking algorithm utilizes an idea from probability, known as Laplace's rule of succession [19]. The proof part will not be in this paper. We will example from author Govan's paper

For this time study, use the fuzzy comparison to check that can this method get same ranking r with a crisp number? But this time does not consider about the alpha number and boundary meaningful or not. Fuzzy numbers just according to the two

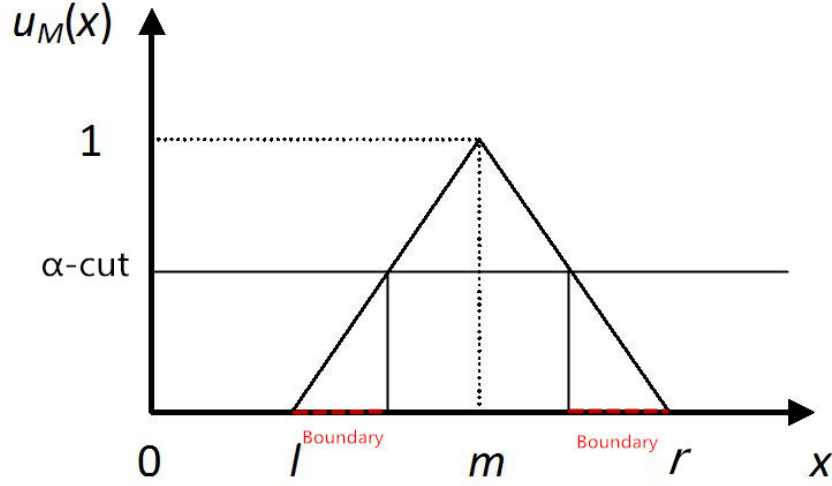


Figure 2: Alpha-cut and Boundary

Table 4: NFL Game

Team	Score	Team	Score	Difference	(l,m,r) $\Delta = 0.1$
Carolina	16	New Orleans	13	3	(.2,.3,.4)
Dallas	38	Philadelphia	17	21	(.6,.7,.8)
Dallas	28	Washington	23	5	(.2,.3,.4)
Houston	34	Carolina	21	13	(.5,.6,.7)
Houston	23	New Orleans	10	13	(.5,.6,.7)
New Orleans	31	Carolina	6	25	(.7,.8,.9)
Philadelphia	33	Washington	25	8	(.4,.5,.6)
Philadelphia	38	New Orleans	23	6	(.5,.6,.7)
Washington	27	Dallas	6	21	(.6,.7,.8)
Washington	20	Philadelphia	12	8	(.4,.5,.6)

Table 5: Each Team Played Times

Team	Played Times
Carolina (g1)	3
Dallas (g2)	3
Houston (g3)	2
New Orleans (g4)	4
Philadelphia (g5)	4
Washington (g6)	4

team score difference.

This study will keep the same value for matrix C . The games will still be 10 games, but this time will set Carolina versus New Orleans fuzzy score be $f_1=[0.2,0.3,0.4]$,

Dallas versus Philadelphia $f_2=[0.6,0.7,0.8] \dots f_{10}=[0.4,0.5,0.6]$. However, there one condition need consider about that is if one team losing for one game, then the fuzzy number will be reversed left to right and get a negative sign. Like Carolina versus New Orleans losing fuzzy number will be $z_1 = [-0.4, -0.3, -0.2]$, Dallas versus Philadelphia losing will be $z_2 = [-0.8, -0.7, -0.6]$ and so on. Next step, set Carolina be (g_1) , Dallas be (g_2) , etc (Table 5). Then find each team winning and losing the game for total games. Using fuzzy arithmetic to get each teams total fuzzy number. For example, Carolina won the first game and lost the fourth and sixth games, so g_1 is the net sum of those three results, giving $(-1.4, -1.1, -0.8)$. The results for each team are recorded in the fuzzy information matrix, b .

$$b = \begin{pmatrix} -1.4 & -1.1 & -0.8 \\ 0 & 0.3 & 0.6 \\ 1 & 1.2 & 1.4 \\ -1.1 & -0.7 & -0.3 \\ -0.5 & -0.1 & 0.3 \\ 0 & 0.4 & 0.8 \end{pmatrix}$$

Using the equation $r = C^{-1}b$, the fuzzy ratings of the teams are in the fuzzy vector r with the ranking of the teams being found by comparing ratings.

$$r = \begin{pmatrix} -0.4012 & -0.2464 & -0.0917 \\ -0.0685 & 0.1029 & 0.2743 \\ 0.0658 & 0.1972 & 0.3287 \\ -0.3358 & -0.1647 & 0.0064 \\ -0.1781 & 0.0075 & 0.1932 \\ -0.0822 & 0.1035 & 0.2892 \end{pmatrix}$$

The ranking of the teams in the original, crisp example based on the ratings is Houston, Dallas, Washington, Philadelphia, Carolina, and New Orleans. However, the ranking based on the fuzzy rating r ranks the teams from first to last as Houston, Dallas, Washington, Philadelphia, New Orleans, and Carolina. The result is a little different because the big game score in the game of New Orleans versus Carolina and the set of fuzzy number also relative.

While the ranking of the top two teams, Houston with rating $[0.0658, 0.1972, 0.3287]$ and Dallas with rating $[-0.0822, 0.1035, 0.2892]$, is based on the middle of each team's fuzzy rating. There is some overlap between the fuzzy ratings. We will need to use α -cuts to determine a level needed to distinguish the two teams. In general, we need to find the smallest α for triangular fuzzy number for which Team 1 $>_{\alpha}$ Team 2, or $(l_1, m_1, r_1)^{\alpha} > (l_2, m_2, r_2)^{\alpha}$. The solutions for various comparisons between teams in this example are shown in Table 6.

Using the example above and fuzzy number (l, m, r) or $(m - \Delta, m, m + \Delta)$, then we can draw the triangular fuzzy number and using linear equation find the two lines intersection. This intersection point y value is α value. Then find every intersection and compare them. Find the biggest one and using the biggest α to operation α -cut. For the Table 4 example ($\Delta=0.1$), there had 10 group fuzzy numbers, but only three intersections exist (Figure 3):

Table 6: Find the α

α	Point	Equation	x,y	α
α_1	$(0.6,1)(0.5,0);(0.3,1)(0.6,0)$	$y=10x-5;y=-10x+6$	$(11/20,1/2)$	$1/2$
α_2	$(0.6,1)(0.7,0);(0.7,1)(0.6,0)$	$y=10x-6;y=-10x+7$	$(13/20,1/2)$	$1/2$
α_3	$(0.8,1)(0.7,0);(0.7,1)(0.8,0)$	$y=10x-7;y=-10x+8$	$(3/4,1/2)$	$1/2$

Therefore the $\alpha = 1/2$. (Table 6). This result, along with the minimum *alpha* for various values of Δ number is recorded in Table 7.

In general, we find the relationship between α and Δ to follow the formula $\alpha = 1 - 0.05/\Delta$. If Δ become smaller, then intersection decreasing and the alpha value

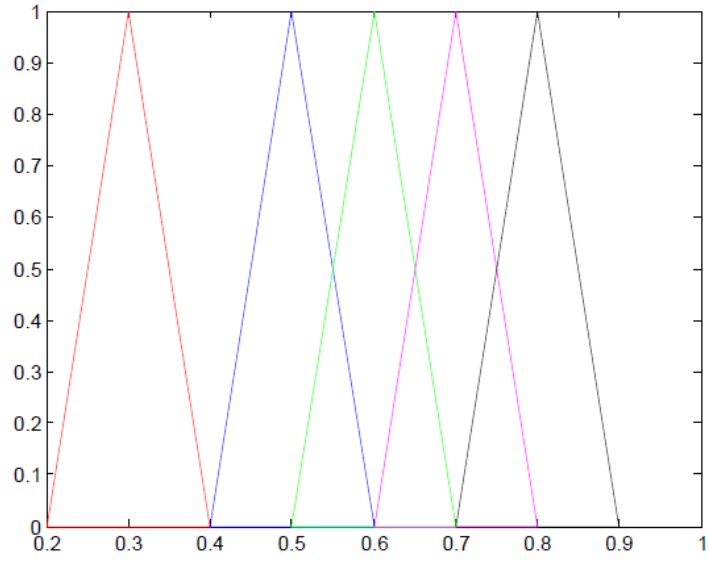


Figure 3: Alpha cut

Table 7: Delta affect Alpha Value

Delta	Intersection	α
0.05	0	0
0.1	3	1/2
0.15	6	2/3
0.2	8	3/4

closer to 0. If Δ become bigger, then intersection increasing and α will increase.

4.2 A Fuzzy version of the PageRank Algorithm

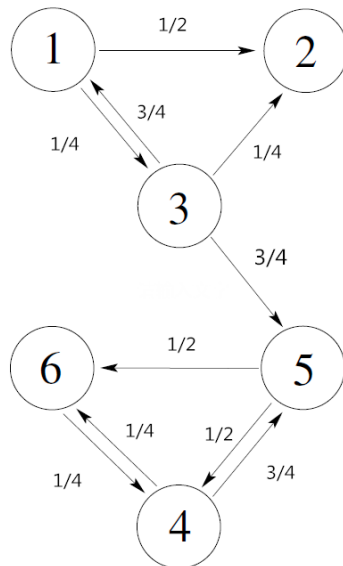


Figure 4: Example of Langville and Meyer [15] with Fuzzy Weights

The Google matrix uses the presence or absence of links to rank nodes. A famous example from Langville and Meyer [15] is studied by changing the crisp numbers to the fuzzy numbers. For example: the weight in the link from 1 to 3 is $1/4$ and from 3 to 1 is $3/4$ that means from 3 to 1 is more important than 1 to 3. The first step in this adaptation of the PageRank algorithm[15] is to change the weights in the graph. See the Figure 4.

The matrix A is now has weights in the interval $[0, 1]$ instead of indicators of 1 or 0.

$$A_{ij} = \begin{cases} 0 < \text{Fuzzy} \leq 1 & \text{if there is a link from } P_i \text{ to } P_j \\ 0 & \text{otherwise} \end{cases}$$

$$A = \begin{matrix} & \begin{matrix} 1 & 2 & 3 & 4 & 5 & 6 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{matrix} & \left(\begin{array}{cccccc} 0 & 1/2 & 1/4 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 3/4 & 1/4 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 3/4 & 1/4 \\ 0 & 0 & 0 & 1/2 & 0 & 1/2 \\ 0 & 0 & 0 & 1/4 & 0 & 0 \end{array} \right) \end{matrix}$$

Next, as in the original PageRank algorithm, let node P_i have a weight equal to $deg^-(i)$ which is a total number from the original node p_i to other node. The $deg^-(i)$ will be $deg^-(i) = \sum_j A_{ij}$

Second step, the Hyperlink matrix is made into an almost stochastic matrix[15] as in

$$H_{ij} = \begin{cases} a_{ij}/deg^-(i) & \text{if there is a link from } P_i \text{ to } P_j \\ 0 & \text{otherwise} \end{cases}$$

$$\text{where } a_i = \begin{cases} Fuzzy & \text{if } H_i^T = 0 \\ 0 & \text{otherwise} \end{cases} \text{ and } e = [1 \ 1 \ 1 \ 1 \ 1 \ 1]^T$$

$$H = \begin{matrix} & \begin{matrix} 1 & 2 & 3 & 4 & 5 & 6 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{matrix} & \left(\begin{array}{cccccc} 0 & 2/3 & 1/3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 3/7 & 1/7 & 0 & 0 & 3/7 & 0 \\ 0 & 0 & 0 & 0 & 3/4 & 1/4 \\ 0 & 0 & 0 & 1/2 & 0 & 1/2 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{array} \right) \end{matrix}$$

The matrix H is stochastic except for a row of zeros. That row of H is completed

to make a stochastic matrix by means of the update as in [15].

$$S = H + \frac{1}{n}ae^T \quad (11)$$

$$S = \begin{matrix} & \begin{matrix} 1 & 2 & 3 & 4 & 5 & 6 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{matrix} & \begin{pmatrix} 0 & 2/3 & 1/3 & 0 & 0 & 0 \\ 1/6 & 1/6 & 1/6 & 1/6 & 1/6 & 1/6 \\ 3/7 & 1/7 & 0 & 0 & 3/7 & 0 \\ 0 & 0 & 0 & 0 & 3/4 & 1/4 \\ 0 & 0 & 0 & 1/2 & 0 & 1/2 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix} \end{matrix}$$

Next, the Google matrix, G , is found by means of

$$G = S\alpha + \frac{(1-\alpha)}{n}ee^T \quad for \quad 0 < \alpha < 1 \quad (12)$$

$$G = \begin{matrix} & \begin{matrix} 1 & 2 & 3 & 4 & 5 & 6 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{matrix} & \begin{pmatrix} \frac{1-\alpha}{n} & \frac{2\alpha}{3} + \frac{1-\alpha}{n} & \frac{\alpha}{3} + \frac{1-\alpha}{n} & \frac{1-\alpha}{n} & \frac{1-\alpha}{n} & \frac{1-\alpha}{n} \\ \frac{\alpha}{6} + \frac{1-\alpha}{n} & \frac{\alpha}{6} + \frac{1-\alpha}{n} & \frac{\alpha}{6} + \frac{1-\alpha}{n} & \frac{\alpha}{6} + \frac{1-\alpha}{n} & \frac{\alpha}{6} + \frac{1-\alpha}{n} & \frac{\alpha}{6} + \frac{1-\alpha}{n} \\ \frac{3\alpha}{7} + \frac{1-\alpha}{n} & \frac{\alpha}{7} + \frac{1-\alpha}{n} & \frac{1-\alpha}{n} & \frac{1-\alpha}{n} & \frac{3\alpha}{7} + \frac{1-\alpha}{n} & \frac{1-\alpha}{n} \\ \frac{1-\alpha}{n} & \frac{1-\alpha}{n} & \frac{1-\alpha}{n} & \frac{1-\alpha}{n} & \frac{3\alpha}{4} + \frac{1-\alpha}{n} & \frac{\alpha}{4} + \frac{1-\alpha}{n} \\ \frac{1-\alpha}{n} & \frac{1-\alpha}{n} & \frac{1-\alpha}{n} & \frac{\alpha}{2} + \frac{1-\alpha}{n} & \frac{1-\alpha}{n} & \frac{\alpha}{2} + \frac{1-\alpha}{n} \\ \frac{1-\alpha}{n} & \frac{1-\alpha}{n} & \frac{1-\alpha}{n} & \frac{1-\alpha}{n} & \frac{1-\alpha}{n} & \frac{1-\alpha}{n} \end{pmatrix} \end{matrix}$$

Check the Google matrix again and still find same eigenvalues that are 1.

The main result of this section is to randomly sample the rankings from graphs following the example of Langville and Meyer with the indicators 1 replaced by fuzzy weights from the set $\{1/4, 1/2, 3/4, 1\}$. That sampling was done using Matlab 10,000 times. The average ranking of the nodes is given in the vector r . See (Appendix B) for details of the code.

$$r = \begin{pmatrix} 5.7065 \\ 4.1914 \\ 5.1021 \\ 1.0003 \\ 2.7821 \\ 2.2176 \end{pmatrix}$$

Table 8: Crisp VS. Fuzzy

Num	Crisp Rank	Fuzzy Rank
1	6	5.7065
2	4	4.1914
3	5	5.1021
4	1	1.0003
5	3	2.7821
6	2	2.2176

From this example, node 4 get the first place, node 6 get second and node 1 get last position. In this 10,000 time, each time they can get different rankings because the change of initial fuzzy data. Compare to original Google matrix, this fuzzy number will affect the final answer ranking r . This example is not going to further because we are not studying the PageRank method, only checks what difference exist when use fuzzy instead of crisp number.

4.3 Incomplete Information in the Saaty's Priority Ranking

For this study, data collected was comparisons of MBA program using the Fundamental Scale (Table 3). This example data expressed a personal judgement. In this section, we will investigate using Saaty Priority Ranking of subsets of that data.

One person wants to read Master of Business Administration (MBA) programs in Texas State. Find website suggest from <http://find-mba.com/schools/usa/texas> and filter for Schools with AACSB, AMBA and/or EQUIS accreditation only and Schools ranked by FT, BusinessWeek and/or Economist. After the filter choose there are only ten business school left (Table 9).

Table 9: MBA Programs Example

Number	School Name
1	University of Texas at Austin - McCombs School of Business
2	Texas A&M University - Mays Business School
3	The University of Texas at Dallas - Naveen Jindal School of Management
4	Rice University - Jesse H. Jones Graduate School of Business
5	Southern Methodist University - Cox School of Business
6	University of Houston - C. T. Bauer College of Business
7	University of Texas at San Antonio - College of Business
8	Texas Christian University - Neeley School of Business
9	Baylor University - Hankamer School of Business
10	Texas Tech University - Rawls College of Business

Next, every two universities were compared using personal judgement following the Fundamental Scale shown in Table 3. Again, this matrix will use one personal's opinion and may vary with individuals. For example, comparing University of Texas at Austin(1) compare with Texas A&M University(2), these two university both have better rank, better reputation, nice school environment, and otherwise similar. Therefore, I compare these university and give them a score 1 because they are of equal importance to me. If University of Texas at Austin (#1 on the list) compares with Baylor University (#9), the score will be 9 because this person did not known much about Baylor University information, so University of Texas at Austin is extremly

important to me. It is a coincidence that the listing numbers and scale scores expressing the intensity of relative preference coincide. Therefore, this complete data from personal opinions implementing the Fundamental Scale in Table 3.

$$\text{Complete Matrix : } \begin{matrix} & \begin{matrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \\ 10 \end{matrix} & \left(\begin{matrix} 1 & 1 & 7 & 3 & 8 & 5 & 6 & 8 & 9 & 5 \\ 1 & 1 & 7 & 3 & 8 & 5 & 6 & 8 & 9 & 5 \\ 1/7 & 1/7 & 1 & 3/7 & 8/7 & 5/7 & 6/7 & 8/7 & 9/7 & 5/7 \\ 1/3 & 1/3 & 7/3 & 1 & 8/3 & 5/3 & 2 & 8/3 & 3 & 5/3 \\ 1/8 & 1/8 & 7/8 & 3/8 & 1 & 5/8 & 6/8 & 1 & 9/8 & 5/8 \\ 1/5 & 1/5 & 7/5 & 3/5 & 8/5 & 1 & 6/5 & 8/5 & 9/5 & 1 \\ 1/6 & 1/6 & 7/6 & 1/2 & 8/6 & 5/6 & 1 & 8/6 & 9/6 & 5/6 \\ 1/8 & 1/8 & 7/8 & 3/8 & 1 & 5/8 & 6/8 & 1 & 9/8 & 5/8 \\ 1/9 & 1/9 & 7/9 & 1/3 & 8/9 & 5/9 & 6/9 & 8/9 & 1 & 5/9 \\ 1/5 & 1/5 & 7/5 & 3/5 & 8/5 & 1 & 6/5 & 8/5 & 9/5 & 1 \end{matrix} \right) \end{matrix}$$

Eigenvalues $\lambda_{max} = 11.4296$ (calculated using MATLAB)

Consistency index $\mu \equiv \frac{\lambda_{max}-n}{n-1} = \frac{11.4296-10}{10-1} = \frac{1.4296}{9} = 0.1588$

Consistency ratio = $\mu/RI = 0.1588/1.49 = 0.1066 \approx 0.1$

Generating the complete matrix above was time consuming. It is natural to wonder what results we can get if only partial information exists. Can we still get the same or similar result? To study this, we assume there is a 10-by-10 consistent matrix, and we only use the first row of data from the original matrix.

$$\begin{matrix} & \begin{matrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \end{matrix} \\ 1 & \left(\begin{matrix} 1 & 1 & 7 & 3 & 8 & 5 & 6 & 8 & 9 & 5 \end{matrix} \right) \end{matrix}$$

To complete the matrix, use the consistency condition. In this partial example, first row already known. Then the first column can be computed by taking reciprocals

and the diagonal values are $a_{11}, a_{22}, \dots, a_{nn} = 1$, so the matrix will be as follows. The graph of the matrix shows that all items are linked by a comparison.

Link Matrix :

$$\begin{array}{c}
 \begin{matrix} & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \end{matrix} \\
 \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \\ 10 \end{matrix} \left(\begin{array}{cccccccccc}
 1 & 1 & 7 & 3 & 8 & 5 & 6 & 8 & 9 & 5 \\
 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 1/7 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 1/3 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
 1/8 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
 1/5 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
 1/6 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
 1/8 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
 1/9 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
 1/5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
 \end{array} \right)
 \end{array}$$

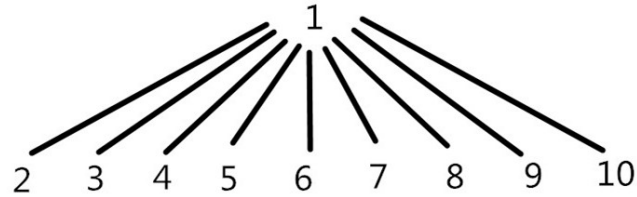


Figure 5: Link

Next, complete the matrix using the consistency condition $a_{ij} * a_{jk} = a_{ik}$. We calculate a_{23} and others below as examples. Notice that when there is more than one way to calculate a comparison, as for a_{34} , the results are identical.

- $a_{23} = a_{21} * a_{13} = 1 * 7 = 7$

$$\begin{aligned}
\bullet a_{34} &= \begin{cases} a_{31} * a_{14} = 1/7 * 3 = 3/7 \\ a_{32} * a_{24} = 1/7 * 3 = 3/7 \end{cases} \\
\bullet a_{45} &= \begin{cases} a_{41} * a_{15} = 1/3 * 8 = 8/3 \\ a_{42} * a_{25} = 1/3 * 8 = 8/3 \\ a_{43} * a_{35} = 7/3 * 8/7 = 8/3 \end{cases}
\end{aligned}$$

The result of determining all comparisons using the consistency condition is recorded in a New Matrix.

$$\text{New Matrix : } \begin{matrix} & \begin{matrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \\ 10 \end{matrix} & \left(\begin{array}{cccccccccc} 1 & 1 & 7 & 3 & 8 & 5 & 6 & 8 & 9 & 5 \\ 1 & 1 & 7 & 3 & 8 & 5 & 6 & 8 & 9 & 5 \\ 1/7 & 1/7 & 1 & 3/7 & 8/7 & 5/7 & 6/7 & 8/7 & 9/7 & 5/7 \\ 1/3 & 1/3 & 7/3 & 1 & 8/3 & 5/3 & 2 & 8/3 & 3 & 5/3 \\ 1/8 & 1/8 & 7/8 & 3/8 & 1 & 5/8 & 6/8 & 1 & 9/8 & 5/8 \\ 1/5 & 1/5 & 7/5 & 3/5 & 8/5 & 1 & 6/5 & 8/5 & 9/5 & 1 \\ 1/6 & 1/6 & 7/6 & 1/2 & 8/6 & 5/6 & 1 & 8/6 & 9/6 & 5/6 \\ 1/8 & 1/8 & 7/8 & 3/8 & 1 & 5/8 & 6/8 & 1 & 9/8 & 5/8 \\ 1/9 & 1/9 & 7/9 & 1/3 & 8/9 & 5/9 & 6/9 & 8/9 & 1 & 5/9 \\ 1/5 & 1/5 & 7/5 & 3/5 & 8/5 & 1 & 6/5 & 8/5 & 9/5 & 1 \end{array} \right) \end{matrix}$$

Eigenvalues $\lambda_{max} = 10$

Consistency index $\mu \equiv \frac{\lambda_{max} - n}{n - 1} = \frac{10 - 10}{10 - 1} = \frac{0}{9} = 0$

Consistency ratio = $\mu / RI = 0 / 1.49 = 0$

The results of completing the comparison matrix from the partial data shows us a consistency ratio of 0. This indicates the entire matrix is perfectly consistent. There are no judgments that are out of line.

This example shows that partial information can be completed to total information. Notice, also, that the matrix from a subset of all information is more consistent

than the matrix from all of the data. .

For a second example, we randomly pick any 9 random numbers from the upper triangle of the matrix, then complete the rest part to see the consistency ratio is still acceptable or not.

Random Matrix :

$$\begin{array}{c}
 \begin{array}{cccccccccc}
 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \\
 \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \\ 10 \end{array} & \left(\begin{array}{cccccccccc}
 1 & 0 & 7 & 0 & 8 & 5 & 0 & 0 & 9 & 0 \\
 0 & 1 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 \\
 1/7 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
 0 & 1/2 & 0 & 1 & 0 & 0 & 8 & 0 & 0 & 0 \\
 1/8 & 0 & 0 & 0 & 1 & 0 & 0 & 3 & 0 & 0 \\
 1/5 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 1/8 & 0 & 0 & 1 & 0 & 0 & 2 \\
 0 & 0 & 0 & 0 & 1/3 & 0 & 0 & 1 & 0 & 0 \\
 1/9 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 1/2 & 0 & 0 & 1
 \end{array} \right)
 \end{array}
 \end{array}$$

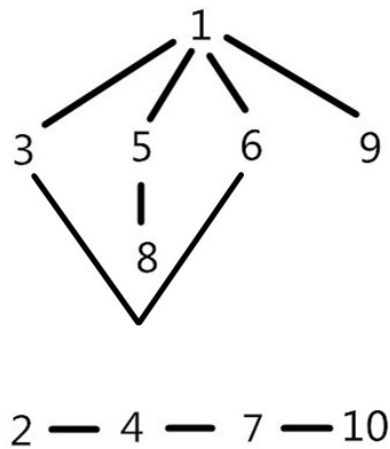


Figure 6: Random

Eigenvalues $\lambda_{max} = ?$

Consistency index $\mu \equiv \frac{\lambda_{max}-n}{n-1} = \frac{?-10}{10-1} = \frac{?}{9} = ?$

Consistency ratio = $\mu/RI = ?/1.49 = ?$

This incomplete matrix was chosen at random from the complete matrix. The diagonal data and lower triangular reciprocal numbers are also filled in the matrix. Then only a few missing entries are still there. The rest of the missing data are there because there are no links between those two groups, then the matrix cannot be completed. Like: $a_{18} = a_{15} * a_{58} = 8 * 3 = 24$, but a_{28} cannot be computed. The eigenvalue, Consistency Index and Consistency Ratio cannot be calculated.

The graph makes it clear what the problem is. There is no comparison between objects in the two subgraphs consisting of items $\{1, 3, 5, 6, 8, 9\}$ and $\{2, 4, 7, 10\}$. That is, the graph is not connected. This is further illustrated in the next example.

The next example will be Left corner data

$$\text{Left Corner Matrix : } \begin{matrix} & \begin{matrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \\ 10 \end{matrix} & \left(\begin{array}{cccccccccc} 1 & 1 & 7 & 3 & 8 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 3 & 2 & 8 & 0 & 0 & 0 & 0 & 0 \\ 1/7 & 1/3 & 1 & 1/5 & 1/2 & 0 & 0 & 0 & 0 & 0 \\ 1/3 & 1/2 & 5 & 1 & 9 & 0 & 0 & 0 & 0 & 0 \\ 1/8 & 1/8 & 1/2 & 1/9 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{array} \right) \end{matrix}$$

Eigenvalues $\lambda_{max} = ?$

Consistency index $\mu \equiv \frac{\lambda_{max}-n}{n-1} = \frac{?-10}{10-1} = \frac{?}{9} = ?$

Consistency ratio = $\mu/RI = ?/1.49 = ?$

This similar example is filled with incomplete data, the left corner partial infor-

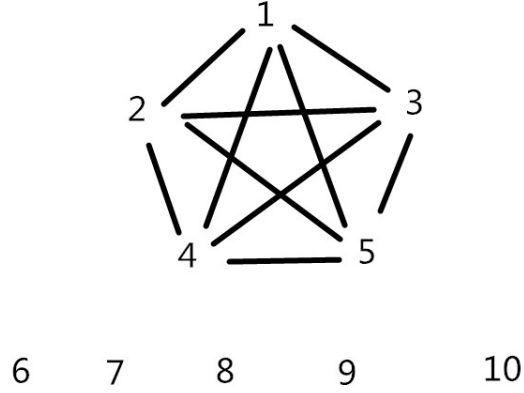


Figure 7: Left Corner

mation is already completed, and there are no links between one group to another. Therefore, this example cannot be completed.

The partial information for the next example is from above the main diagonal. Looking at the graph in Figure 8 shows that the graph is connected. We expect, therefore, to be able to complete the matrix in this case.

Diagonal Matrix :

$$\begin{array}{c}
 \begin{matrix} & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \end{matrix} \\
 \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \\ 10 \end{matrix} \begin{pmatrix}
 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 1 & 1 & 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 1/3 & 1 & 1/5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 5 & 1 & 9 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 1/9 & 1 & 1/6 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 6 & 1 & 3 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 1/3 & 1 & 6 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 1/6 & 1 & 1 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1/5 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 5 & 1 & 1
 \end{pmatrix}
 \end{array}$$

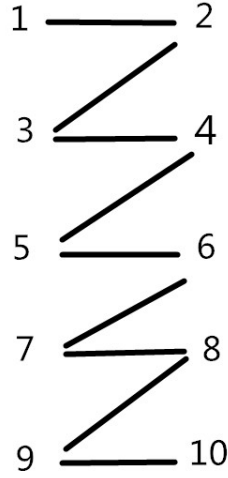


Figure 8: Diagonal

Eigenvalues $\lambda_{max} = 10$

Consistency index $\mu \equiv \frac{\lambda_{max}-n}{n-1} = \frac{10-10}{10-1} = \frac{0}{9} = 0$

Consistency ratio = $\mu/RI = 0/1.49 = 0$

So, indeed, the partial information in this example can be completed. Again, the matrix is perfectly consistent.

In a last example with partial information, the graph in Figure 9 is connected, but also contains loops. This means pairs of objects, say 1 and 5, can be connected in two different ways. Objects 1 and 5 are connected through 1-2-4-5 and 1-3-5.

Loop Matrix :

$$\begin{matrix}
 & \begin{matrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \end{matrix} \\
 \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \\ 10 \end{matrix} & \left(\begin{array}{cccccccccc}
 1 & 1 & 7 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 1 & 1 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 \\
 1/7 & 0 & 1 & 0 & 2 & 0 & 0 & 0 & 0 & 0 \\
 0 & 1/2 & 0 & 1 & 9 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 1/2 & 1/9 & 1 & 1/6 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 6 & 1 & 3 & 7 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 1/3 & 1 & 0 & 6 & 0 \\
 0 & 0 & 0 & 0 & 0 & 1/7 & 0 & 1 & 0 & 1/5 \\
 0 & 0 & 0 & 0 & 0 & 0 & 1/6 & 0 & 1 & 1/5 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 5 & 5 & 1
 \end{array} \right)
 \end{matrix}$$

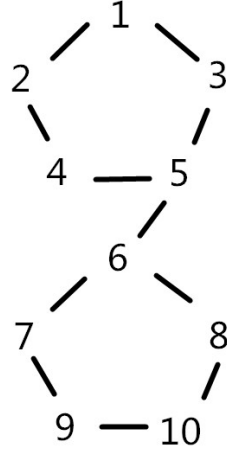


Figure 9: Loop

Eigenvalues $\lambda_{max} = 10.0351$

Consistency index $\mu \equiv \frac{\lambda_{max} - n}{n-1} = \frac{10.0351 - 10}{10-1} = 0.0039$

Consistency ratio $= \mu / RI = 0.0039 / 1.49 = 0.0026$

The result show us the partial example get consistency ratio = 0.003. This ratio is acceptable since it is less than 10%. This is very special example. The next part

shows how to solve these missing data. This is a very special example.

From the Saaty's paper, if the consistency ratio number is bigger than people are expected or above the scale boundary, they can do three things: first, find the most inconsistent judgment which is $a_{ij} = w_j/w_i$ in the matrix. Second, determine the range of value. Third, ask the judge to reconsider a certain value in the matrix and fix that value.

From partial pairwise information to get full expect full information, we can get our theorem:

Theorem : Partial matrix get full matrix iff there is connected graph. The solution is unique iff there is a connected tree.

Here has one question that we need to solve. If partial matrix connects and there is a loop, then the matrix will become to an inconsistent. Here has some information should know $AW = nW \Leftrightarrow AW = \lambda W$ if λ belong to consistency matrix. The original condition is $[W_i/W_1, W_i/W_2, \dots, W_i/W_n] * [W_1, W_2, \dots, W_n]^T = n * W_i$ and $\sum W_i = 1$. However, for inconsistency to get $W' = 1/S * W \Rightarrow AW' = A * 1/S * W = 1/S * AW = 1/S * \lambda W = \lambda * (1/S * W) = \lambda W'$. Here is simple loop example

$$\text{Simple Loop Matrix : } \begin{matrix} & \begin{matrix} 1 & 2 & 3 & 4 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \end{matrix} & \left(\begin{array}{cccc} 1 & 2 & 7 & 0 \\ 1/2 & 1 & 0 & 3 \\ 1/7 & 0 & 1 & 1 \\ 0 & 1/3 & 1 & 1 \end{array} \right) \end{matrix}$$

According to the information above, using this partial information we can get full information first. Like $a_{14} = a_{12} * a_{24} = 2 * 3 = 6$ and $a_{14} = a_{13} * a_{34} = 7 * 1 = 7$. There are two different ways to get a different answer because the matrix is inconsistent. So we consider the geometric mean $a_{14} * a_{14} = a_{12} * a_{24} * a_{13} * a_{34}$, then get $a_{14} = \sqrt{a_{12} * a_{24} * a_{13} * a_{34}} = \sqrt{42} \approx 6.4807$ and $a_{41} = \sqrt{a_{14}} \approx 0.1543$

When we finish entries in the entire matrix, the next step is to use matlab to

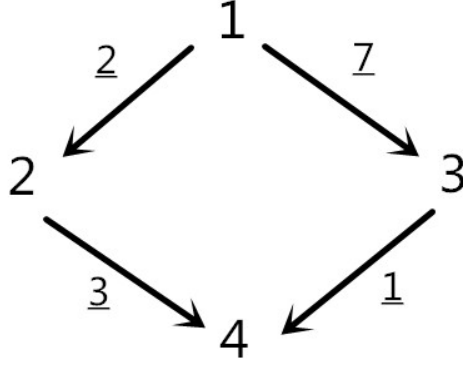


Figure 10: Simple Loop

find this eigenvalues $\lambda_{max} = 4.0015$ and eigenvectors $w_1 = 0.8848, w_2 = 0.4257, w_3 = 0.1314, w_4 = 0.1365$. The sum of W will be 1.5748. Using formula $W' = 1/S * W$ can get W'_1, W'_2, W'_3, W'_4 and $\sum W' = 1$. The new error \mathcal{E}_{ij} matrix will be

$$\mathcal{E}_{ij} = \begin{matrix} & \begin{matrix} 1 & 2 & 3 & 4 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \end{matrix} & \begin{pmatrix} 1 & 0.962251 & 1.03956 & 0.999798 \\ 1.03923 & 1 & 1.000199 & 0.961945 \\ 0.961948 & 0.999801 & 1 & 1.038813 \\ 1.000202 & 1.03956 & 0.962637 & 1 \end{pmatrix} \end{matrix}$$

In this new matrix, the biggest numbers are $\mathcal{E}_{42} = 1.03956$ and $\mathcal{E}_{13} = 1.03956$. Which means that the inconsistent part comes from the original data not from computed data. If the consistency ratio number is bigger than expected or above the scale boundary, we can do three things: First, find the most inconsistent judgment which is $\mathcal{E}_{ij} = a_{ij} * \frac{w_j}{w_i}$ in the matrix. Second, determine the range of values. Third, ask the judge to reconsider that the value in the matrix and fix that value.

5 Conclusion

In this thesis, the Colley Matrix Method for rankings based on partial information is generalized to allow for fuzzy comparisons based on score differences instead of just wins and losses. This resulted in fuzzy rankings for NFL teams. The results are similar to the crisp number result, but one part that needs to be careful is the boundaries for fuzzy number. By changing the Δ , a measure of precision in the fuzzy comparisons, we found that it is possible that too much imprecision makes the calculated rankings meaningless. Second, fuzzy numbers instead of crisp numbers are used in the PageRank method as weights for links. The result was that the initial fuzzy numbers can affect the results of the ranking. Random sampling of the fuzzy weights with this method showed that the rankings are mostly, but not completely, determined by the underlying graph. Third, a consistency criteria was used to complete partial pairwise intangible comparisons in Saaty's Priority Ranking. The result was a criteria for being able to make this completion, connectedness of the underlying graph. After completing the data, the new comparisons had less error than for the original data.

References

- [1] Broek, P. and Noppen, J. “*Rankings from Fuzzy Pairwise Comparisons*”, IEEE Xplore. Volume:2, pg.188 - 190, 28 - 30, Nov. 2005.
- [2] Brakken, S. “*Gershgorins Theorem for Estimating Eigenvalues*”, Retrieved from: <http://buzzard.ups.edu/courses/2007spring/projects/brakkenthal-paper.pdf>, 2007.
- [3] Brin, S. and Page, L. “*The anatomy of a large-scale hypertextual Web search engine*”, Computer Networks and ISDN System, Pg.107 - 117, 1998.
- [4] Colley, W.N “*Colleys Bias Free College Football Ranking Method: The Colley Matrix Explained*”, Retrieved from: <http://www.colleyrankings.com/matrate.pdf>.
- [5] Chutia,R. and Mahanta,S. and Datta,D. “*Arithmetic of Triangular Fuzzy Variable from Credibility Theory*”, International Journal of Energy, Information and Communications Vol. 2, Issue 3, August, 2011.
- [6] Dutta,P. and Boruah, H. and Ali,T. “*Fuzzy Arithmetic with and without using α -cut Method: A Comparative Study*”, International Journal of Latest Trends in Computing. Volume 2, Issue 1, March 2011.
- [7] Govan, A.Y. “*Ranking Theory with Application to Popular Sports*”, Retrieved from: <http://meyer.math.ncsu.edu/Meyer/REU/REU2009/GovanThesis.pdf>, 4 Dec, 2008.
- [8] Gao,S. and Zhang,Z. and Cao,C. “*Multiplication Operation on Fuzzy Numbers*”, Journal of Software, VOL. 4, NO. 4, June 2009.
- [9] Gani,A.N. and Assarudeen,S.N. “*A New Operation on Triangular Fuzzy Number for Solving Fuzzy Linear Programming Problem*”, Applied Mathematical Sciences, Vol. 6, 2012.

- [10] Ipsen,I. and Wills,R. “*Mathematical Properties and Analysis of Google’s PageRank*”, Retrieved from: <http://www4.ncsu.edu/~ipsen/ps/cedya.pdf>.
- [11] Lebesgue,H. “*Lecons Sur Lintegration*”, Second Ed, Gauthier-Villars, Paris, 1928.
- [12] MacCrimmon,K.R “*Multiple Criteria Decision Making*”, University of South Carolina Press, Columbia, S.C, 1973.
- [13] Mason, O. “*NonNegative Matrices and Related Topics*”, Hamilton Institute, July 9 - 12, 2006.
- [14] Meyer, C.D. “*Matrix Analysis and Applied Linear Algebra*”, SIAM: Society for Industrial and Applied Mathematics; Har/Cdr edition. Pg 498, February 15, 2001.
- [15] Meyer, C.D. and Langville,A.N. “*Google’s PageRank and Beyond: The Science of Search Engine Rankings*”, Princeton University Press, 2006.
- [16] Pilkington,A. “*Colley’s Method*”, Retrieved from: <http://www3.nd.edu/~apilking/Math10170/Information/Lectures202015/Topic8Colley.pdf>. University of Notre Dame, 2015.
- [17] Page,L. and Brin,S. and Motwani,R. and Winograd,T. “*The PageRank Citation Ranking: Bringing Order To the Web*”, Retrieved from: <http://ilpubs.stanford.edu:8090/422/1/1999-66.pdf>. Stanford Digital Library Technologies Projec, 1998.
- [18] Ribeiro, R.A. “*Fuzzy Multiple Attribute Decision Making: A Review and New Preference Elicitation Techniques*”, Fuzzy Sets and Systems - Special issue on fuzzy multiple criteria decision making, Pages 155 - 181, Volume 78 Issue 2, March 11, 1996.

- [19] Ross,S. “*A First Course in Probability*”, Pearson Prentice Hall, 8th edition, January 7, 2009.
- [20] Saaty,T.L. “*Relative Measurement and Its Generalization in Decision Making Why Pairwise Comparisons are Central in Mathematics for the Measurement of Intangible Factors The Analytic Hierarchy/Network Process*”, Royal Academy of Sciences, Spain, Series A. Mathematics.pp. 251 - 318,VOL. 102 (2), 2008.
- [21] Yager,R. “*On Ordered Weighted Averaging Aggregation Operators in Multicriteria Decisionmaking*”, IEEE Transactions on Systems, Man and Cybernetics Volume 18 Issue 1, Page(s):183 - 190, Feb 1988.
- [22] Zimmermann, H.J. “*Fuzzy Sets, Decision Making and Expert Systems*”, Kluwer Academic Publisher, International Series in Management Science/Operations Research, 1987.

Appendix A

There are some operations of the fuzzy number. Suppose the fuzzy number $A = (a_1, a_2, a_3)$ which $a_3 > a_2 > a_1$ and $B = (b_1, b_2, b_3)$ which $b_3 > b_2 > b_1$ be two positive fuzzy numbers, then

Addition operation:

$$A + B = (a_1, a_2, a_3) + (b_1, b_2, b_3) = (a_1 + b_1, a_2 + b_2, a_3 + b_3) \quad (13)$$

Symmetric operation:

$$-(A) = (-a_3, -a_2, -a_1) \quad (14)$$

Subtraction operation:

$$A - B = (a_1, a_2, a_3) - (b_1, b_2, b_3) = (a_1 - b_3, a_2 - b_2, a_3 - b_1) \quad (15)$$

Multiplication operation [9]:

$$A * B = (\min(a_1b_1, a_1b_3, a_3b_1, a_3b_3), a_2b_2, \max(a_1b_1, a_1b_3, a_3b_1, a_3b_3)). \quad (16)$$

Division:

$$A/B = (\min(a_1/b_1, a_1/b_3, a_3/b_1, a_3/b_3), a_2/b_2, \max(a_1/b_1, a_1/b_3, a_3/b_1, a_3/b_3)). \quad (17)$$

Square root of fuzzy number [5]

$$\sqrt{A} = [\sqrt{a_1}, \sqrt{a_2}, \sqrt{a_3}] \quad (18)$$

Operation of α - cut:

Fuzzy number $A = (a_1, a_2, a_3)$, and $\alpha \in [0, 1]$ then the membership functions will be

$$\mu_A(x) = \begin{cases} 0 & x \leq a_1 \\ \frac{x-a_1}{a_2-a_1} & \text{if } a_1 < x \leq a_2 \\ \frac{a_3-x}{a_3-a_2} & \text{if } a_2 < x \leq a_3 \\ 0 & x > a_3 \end{cases} \quad (19)$$

The inequality $\mu_A(x) = \alpha$ has two solutions, found by solving

$$\alpha = \frac{a_1^\alpha - a_1}{a_2 - a_1} = \frac{a_3 - a_3^\alpha}{a_3 - a_2} \quad (20)$$

The result is $a_1^\alpha = (a_2 - a_1)\alpha + a_1$ and $a_3^\alpha = -(a_3 - a_2)\alpha + a_3$. The interval $A_\alpha = [a_1^\alpha, a_3^\alpha]$ includes all numbers x with $\mu_A(x) \geq \alpha$.

Arithmetic operation of fuzzy numbers using α -cut method [6]: Suppose the fuzzy number $A = (a_1, a_2, a_3)$ which $a_3 > a_2 > a_1$ and $B = (b_1, b_2, b_3)$ which $b_3 > b_2 > b_1$ $A^\alpha = [(a_2 - a_1)\alpha + a_1, a_3 - (a_3 - a_2)\alpha]$; $B^\alpha = [(b_2 - b_1)\alpha + b_1, b_3 - (b_3 - b_2)\alpha]$

Addition:

$$A^\alpha + B^\alpha = [((a_2 - a_1)\alpha + a_1) + ((b_2 - b_1)\alpha + b_1), (a_3 - (a_3 - a_2)\alpha) + (b_3 - (b_3 - b_2)\alpha)] \quad (21)$$

Subtraction:

$$A^\alpha - B^\alpha = [((a_2 - a_1)\alpha + a_1) - ((b_2 - b_1)\alpha + b_1), (a_3 - (a_3 - a_2)\alpha) - (b_3 - (b_3 - b_2)\alpha)] \quad (22)$$

Multiplication:

$$A^\alpha * B^\alpha = [((a_2 - a_1) + a_1) * ((b_2 - b_1)\alpha + b_1), (a_3 - (a_3 - a_2)\alpha) * (b_3 - (b_3 - b_2)\alpha)] \quad (23)$$

Division:

$$A^\alpha / B^\alpha = [((a_2 - a_1)\alpha + a_1) / (b_3 - (b_3 - b_2)\alpha), (a_3 - (a_3 - a_2)\alpha) / (b_2 - b_1)\alpha + b_1] \quad (24)$$

Square root:

$$\sqrt{A^\alpha} = [\sqrt{(a_2 - a_1)\alpha + a_1}, \sqrt{a_3 - (a_3 - a_2)\alpha}] \quad (25)$$

Appendix B

PageRank Code

%PaggRank Crisp Code

```

maxcount=10000;n=6;
Y=zeros(n,maxcount);
edge=[1 3 6 4 1 5 5 3 3 4;3 2 4 6 2 4 6 1 5 5];
p=0.85;
for k = 1:maxcount
    x2=ones(n,1);
    Gnew=zeros(n,n);
    for i = 1:10
        Gnew(edge(1,i),edge(2,i))=1;
    end
    degree=Gnew*x2;
    Gnew=Gnew./ repmat(degree,1,n);
    Gnew(2,:)=1/n;
    Gnew=p*Gnew+(1-p)*x2*(x2./n)';
    G=Gnew';
% Power method
xold = zeros(n,1);
while max(abs(xold-x2)) > .0001
    xold = x2;
    x2= G*x2;
end
% Gnew
%x
Y(:,k)=x2;
end
Z=Y;
for j =1:n
    [maxVal maxInd]=max(Z);
    for i =1:maxcount
        Z(maxInd(i),i)=-j;
    end
end
Z=-Z;
mean(Z,2)

```

% PaggRank Fuzzy 10,000 Trails Code

```

maxcount=10000;n=6;
Y=zeros(n,maxcount);
edge=[1 3 6 4 1 5 5 3 3 4;3 2 4 6 2 4 6 1 5 5];
p=0.85;

```

```

for k = 1:maxcount
    x2=ones(n,1);
    Gnew=zeros(n,n);
        for i = 1:10
            Gnew(edge(1,i),edge(2,i))=randi([1,4])/4;
        end
    degree=Gnew*x2;
    Gnew=Gnew./ repmat(degree,1,n);
    Gnew(2,:)=1/n;
    Gnew=p*Gnew+(1-p)*x2*(x2./n)';
    G=Gnew';
    % Power method
    xold = zeros(n,1);
    while max(abs(xold-x2)) > .0001
        xold = x2;
        x2= G*x2;
    end
    % Gnew
    %x
    Y(:,k)=x2;
end
    Z=Y;
    for j =1:n
        [maxVal maxInd]=max(Z);
        for i =1:maxcount
            Z(maxInd(i),i)=-j;
        end
    end
    Z=-Z;
    mean(Z,2)

```

Appendix C

Completing Ranking Matrix

%Diagonal

```
A=[1 1 0 0 0 0 0 0 0 0; 1 1 3 0 0 0 0 0 0 0; 0 1/3 1 1/5 0 0  
    0 0 0 0; 0 0 5 1 9 0 0 0 0 0; 0 0 0 1/9 1 1/6 0 0 0 0; 0 0  
    0 0 6 1 3 0 0 0; 0 0 0 0 0 1/3 1 6 0 0; 0 0 0 0 0 0 0 1/6  
    1 1 0; 0 0 0 0 0 0 0 1 1 1/5; 0 0 0 0 0 0 0 0 5 1; ];
```

```
A(1,3)=findmissing(A,1,3);  
A(1,4)=findmissing(A,1,4);  
A(1,5)=findmissing(A,1,5);  
A(1,6)=findmissing(A,1,6);  
A(1,7)=findmissing(A,1,7);  
A(1,8)=findmissing(A,1,8);  
A(1,9)=findmissing(A,1,9);  
A(1,10)=findmissing(A,1,10);  
A(2,4)=findmissing(A,2,4);  
A(2,5)=findmissing(A,2,5);  
A(2,6)=findmissing(A,2,6);  
A(2,7)=findmissing(A,2,7);  
A(2,8)=findmissing(A,2,8);  
A(2,9)=findmissing(A,2,9);  
A(2,10)=findmissing(A,2,10);  
A(3,5)=findmissing(A,3,5);  
A(3,6)=findmissing(A,3,6);  
A(3,7)=findmissing(A,3,7);  
A(3,8)=findmissing(A,3,8);  
A(3,9)=findmissing(A,3,9);  
A(3,10)=findmissing(A,3,10);  
A(4,6)=findmissing(A,4,6);  
A(4,7)=findmissing(A,4,7);  
A(4,8)=findmissing(A,4,8);  
A(4,9)=findmissing(A,4,9);  
A(4,10)=findmissing(A,4,10);  
A(5,7)=findmissing(A,5,7);  
A(5,8)=findmissing(A,5,8);  
A(5,9)=findmissing(A,5,9);  
A(5,10)=findmissing(A,5,10);  
A(6,8)=findmissing(A,6,8);  
A(6,9)=findmissing(A,6,9);  
A(6,10)=findmissing(A,6,10);  
A(7,9)=findmissing(A,7,9);  
A(7,10)=findmissing(A,7,10);  
A(8,10)=findmissing(A,8,10);
```

```

A(3,1)=1/A(1,3);
A(4,1)=1/A(1,4);
A(5,1)=1/A(1,5);
A(6,1)=1/A(1,6);
A(7,1)=1/A(1,7);
A(8,1)=1/A(1,8);
A(9,1)=1/A(1,9);
A(10,1)=1/A(1,10);
A(4,2)=1/A(2,4);
A(5,2)=1/A(2,5);
A(6,2)=1/A(2,6);
A(7,2)=1/A(2,7);
A(8,2)=1/A(2,8);
A(9,2)=1/A(2,9);
A(10,2)=1/A(2,10);
A(5,3)=1/A(3,5);
A(6,3)=1/A(3,6);
A(7,3)=1/A(3,7);
A(8,3)=1/A(3,8);
A(9,3)=1/A(3,9);
A(10,3)=1/A(3,10);
A(6,4)=1/A(4,6);
A(7,4)=1/A(4,7);
A(8,4)=1/A(4,8);
A(9,4)=1/A(4,9);
A(10,4)=1/A(4,10);
A(7,5)=1/A(5,7);
A(8,5)=1/A(5,8);
A(9,5)=1/A(5,9);
A(10,5)=1/A(5,10);
A(8,6)=1/A(6,8);
A(9,6)=1/A(6,9);
A(10,6)=1/A(6,10);
A(9,7)=1/A(7,9);
A(10,7)=1/A(7,10);
A(10,8)=1/A(8,10);
[V,D]=eig(A)

```

Appendix D

Matlab Code: Calculating Error Matrix E_{ij}

```
[V,D]=eig(A)

W=V(:, 1)

for i = 1:10
    for j = 1:10
        E(i,j) = A(i,j)* W(j)/ W(i);
    end
end

E
```

Appendix E

Matlab Code: Fuzzy Operation

Fuzzy operation with Matlab Code

% Addition with two variable :

```
function z = fuzadd(x,y);
```

```
z =[x(:) + y(:) ];
```

```
fuzadd=z;
```

% Three variable

```
function z = fuzadd(x,y,z);
```

```
z =[x(:) + y(:)+ z(:) ];
```

```
fuzadd=z;
```

% Four variable

```
function z = fuzadd(x,y,z,a);
```

```
z =[x(:) + y(:)+ z(:)+ a(:) ];
```

```
fuzadd=z;
```

% Negative operation:

```
function z = fuzneg(x);
```

```
z = fliplr(-x);
```

```
fuzneg =z;
```

% Subtraction operation:

```
function z = fuzsub(x,y);
```

```
z =[x(:) - y(:) ];
```

```
fuzsub=z;
```

% Constant Multiplication operation:

```
function z = fsprod(c,x);
```

```
if c>0
```

```
    z = c*x;
```

```
elseif c < 0
```

```
    z = c*fliplr(x);
```

```
end
```


Appendix F

Completing Missing Data

% Main Function Code

```
function p = findmissing(adj,src,snk)
    temp=pathbetweennodes(adj,src,snk);
    D=zeros(size(temp,1),8);
```

```
for i = 1:size(D,1)
    t = cell2mat(temp(i));
    n=size(t,2);
    for j = 1:n
        D(i,j)=t(j);
    end
end
```

```
p=1; r = size(D,1); root = 1/r;
for i = 1:r
    n = size(find(D(i,:)),2)-1;
    for j=1:n
        p=p*(adj(D(i,j),D(i,j+1)))^root;
    end
end
```

% Pathbetweennodes

```
function pth = pathbetweennodes(adj, src, snk, verbose)
% PATHBETWEENNODES Return all paths between two nodes of a
graph
% pth = pathbetweennodes(adj, src, snk)
% pth = pathbetweennodes(adj, src, snk, vflag)
% This function returns all simple paths (i.e. no cycles)
between two nodes in a graph. Not sure this is the most
efficient algorithm, but it seems to work quickly for
small graphs, and isn't too terrible for graphs with 50
nodes.
% Input variables:
% adj: adjacency matrix
% src: index of starting node
% snk: index of target node
% vflag: logical scalar for verbose mode. If true, prints
paths to screen as it traverses them (can be useful for
larger, time-consuming graphs). [false]
% Output variables:
```

*% pth: cell array, with each cell holding the indices of a
unique path of nodes from src to snk.
% Copyright 2014 Kelly Kearney*

```

if nargin < 4
    verbose = false;
end
n = size(adj,1);
stack = src;
stop = false;
pth = cell(0);
cycles = cell(0);

next = cell(n,1);
for in = 1:n
    next{in} = find(adj(in,:));
end

visited = cell(0);

pred = src;
while 1
    visited = [visited; sprintf('%d', stack)];
    [stack, pred] = addnode(stack, next, visited, pred);
    if verbose
        fprintf('%2d', stack);
        fprintf('\n');
    end

    if isempty(stack)
        break;
    end

    if stack(end) == snk
        pth = [pth; {stack}];
        visited = [visited; sprintf('%d', stack)];
        stack = popnode(stack);
    elseif length(unique(stack)) < length(stack)
        cycles = [cycles; {stack}];
        visited = [visited; sprintf('%d', stack)];
        stack = popnode(stack);
    end
end

function [stack, pred] = addnode(stack, next, visited, pred)

```

```

newnode = setdiff(next{stack(end)}, pred);
possible = arrayfun(@(x) sprintf('%d', [stack x]), newnode,
    'uni', 0);
isnew = ~ismember(possible, visited);

if any(isnew)
    idx = find(isnew, 1);
    stack = str2num(possible{idx});
    pred = stack(end-1);
else
    [stack, pred] = popnode(stack);
end

function [stack, pred] = popnode(stack)
stack = stack(1:end-1);
if length(stack) > 1
    pred = stack(end-1);
else
    pred = [];
end

```