

Article

Analytic Hierarchy Process (AHP): Matlab computation software

WenJun Zhang

School of Life Sciences, Sun Yat-sen University, Guangzhou 510275, China; International Academy of Ecology and Environmental Sciences, Hong Kong

E-mail: zhwj@mail.sysu.edu.cn, wjzhang@iaees.org

Received 10 July 2017; Accepted 23 December 2018; Published 1 March 2019



Abstract

How to determine relative importance (e.g., weight) of nodes or links in a network is one of the focus in network biology. Analytic Hierarchy Process (AHP) can be used to determine weights of edges (i.e., links) and vertices (i.e., nodes) of a network, which is suitable for the hierarchical structure problems hard to be quantified. In present article, the Matlab computation software of AHP was presented for practical uses.

Keywords Analytic Hierarchy Process (AHP); computation software; Matlab.

Network Biology

URL: <http://www.iaees.org/publications/journals/nb/online-version.asp>

RSS: <http://www.iaees.org/publications/journals/nb/rss.xml>

E-mail: networkbiology@iaees.org

Editor-in-Chief: WenJun Zhang

Publisher: International Academy of Ecology and Environmental Sciences

1 Introduction

How to determine relative importance (e.g., weight) of nodes or links in a network is one of the focus in network biology (Zhang and Zhang, 2008; Zhang and Zhan, 2011; Li and Zhang, 2012; Choura et al., 2015; Jiang et al., 2015; Jiang and Zhang, 2015; Zhang, 2007, 2010, 2012a-c, 2016, 2018; Zhang et al., 2014; Zhang and Liu, 2008; Zhang and Feng, 2017; Zhang and Zhang, 2019). Analytic Hierarchy Process (AHP) can be used to determine weights of edges (i.e., links) and vertices (i.e., nodes) of a network. It is suitable for the hierarchical structure problems that are hard to be quantified (Jiang, 1988; Zhang, 2012a, 2016, 2018). In this article, I present the Matlab computation software of AHP.

2 Analytic Hierarchy Process (AHP)

In AHP of network analysis, we need to organize all vertices into several layers according to their attributes and relations. The top layer contains one vertex. Between-layer vertices are linked by directed edges. In general, the number of layers is infinite. However, too many adjacent vertices or edges will result in a difficulty in pairwise comparisons.

AHP can be conducted in the following way (Zhang, 2012a, 2016, 2018):

(1) Establish a hierarchical structure model. Determine the adjacent vertices at lower layer for every vertex. Starting from the top layer, for every layer, label all vertices from left to right at each layer.

(2) Construct judgement matrix of every layer. A judgement matrix is a weight matrix in which an element

is the weight of a vertex over its adjacent vertices (or its associated edges) at lower layer. The lower triangular vertices in a judgement matrix are the reciprocals of corresponding vertices in upper triangular matrix, and the diagonal elements are 1's.

(3) Consistency test of judge matrix. For a judgement matrix, $B=(b_{ij})_{m \times m}$, calculate maximum eigenvalue λ_{\max} , and the eigenvector.

Consistency CI of a judgement matrix B is

$$CI=(\lambda_{\max}-m)(m-1)$$

A smaller CI means that the judgement matrix is close to complete consistency. When $m=1, 2, 3, 4, 5, 6, 7, 8, 9$, the means of random consistency RI are 0, 0, 0.58, 0.9, 1.12, 1.24, 1.32, 1.41, 1.45, respectively (Jiang, 1988).

The proportion of random consistency CR is

$$CR=CI/RI$$

If $CR < 0.1$, then the judgement matrix possesses a better consistency. Otherwise, we need to adjust the judgement matrix, until the judgement matrix possesses a satisfied consistency.

(4) Calculate the weights of present layer's vertices over next layer's adjacent vertices, which are also weights of edges of every vertex.

(5) Calculate the weights of lower layer's vertices.

(6) Return to (3), and repeat above procedures, until the weights of bottom layer's vertices are obtained.

(7) Starting from bottom layer, for each layer search for the vertex with maximal weight at present layer.

The vertex is the most important vertex of the present layer relative to the top layer vertex.

3 Matlab Computation Software

The Matlab codes for AHP algorithm are as follows:

```
% AHP algorithm
data=input('Input the file name of raw data (e.g., raw.txt, raw.xls, etc. The matrix is data=(xij)mm*5.):','s');
data=xlsread(data);
mm=size(data,1);
d1=zeros(1,mm); d11=zeros(1,mm); d2=zeros(1,mm);
d22=zeros(1,mm); d3=zeros(1,mm); d33=zeros(1,mm);
d4=zeros(1,mm); d44=zeros(1,mm); u=zeros(1,mm);
lab=zeros(1,mm); d=zeros(1,mm); dd=zeros(1,mm);
ww=zeros(4,mm); w=zeros(1,mm);
d1=data(:,1); d2=data(:,2); d3=data(:,3); d4=data(:,4); d=data(:,5);
wa=zeros(1,max(d)); wb=zeros(1,max(d));
k=0; ii=0; rr=0;
while (ii<mm)
nn=0;
for j=k+1:mm
if ((d1(j)==d1(k+1)) & (d2(j)==d2(k+1))) nn=nn+1;
else break;
end
end
for i=k+1:k+nn
d11(i-k)=d1(i);
d22(i-k)=d2(i);
d33(i-k)=d3(i);
d44(i-k)=d4(i);
dd(i-k)=d(i);
```

```

end
m=floor(floor(1+sqrt(1+8*nn))/2);
iss='\n';
[w,u,iss]=weightAHP(m,d11,d22,d33,d44,dd,iss);
rr=rr+m;
for i=rr-m+1:rr
ww(1,i)=d11(i-rr+m);
ww(2,i)=u(i-rr+m);
ww(3,i)=w(i-rr+m);
end
k=k+nn;
ii=ii+nn;
end
la=0;
for i=1:rr
if (ww(1,i)>la) la=floor(ww(1,i)); end
end
ls=0;
for i=1:rr
if ((floor(ww(1,i))==la) & (ww(2,i)>ls)) ls=floor(ww(2,i)); end
end
for i=1:la
ii=0;
k=0;
for j=1:rr
if (floor(ww(1,j))==i) k=k+1; end
if ((floor(ww(1,j))==i) & (ww(2,j)>ii)) ii=ii+1; end
end
d(i)=ii;
u(i)=k;
end
u(la+1)=ls;
%d(0)=1;
d(la+1)=ls;
iss=strcat(iss,'Number of vertices for every layer:\n');
for i=0:la
if (i==0) tem=1;
else tem=d(i);
end
iss=strcat(iss,['Layer-',num2str(i),'=',num2str(floor(tem)),' ']);
end
iss=strcat(iss,'\n');
iss=strcat(iss,'Weight of layer-1 vertex:\n1.0\n');
iss=strcat(iss,'Weights of layer-2 vertices:\n');
opt=0;
d11(2)=1;
for i=1:u(1)
wa(i)=ww(3,i);
iss=strcat(iss,[num2str(wa(i)),' ']);
if (wa(i)>opt)
opt=wa(i); d11(2)=i;

```

```

end
end
iss=strcat(iss,'\n');
for i=2:la
ss=0; jj=0;
for j=u(i-1)+1:u(i-1)+u(i)
ss=ss+ww(3,j);
if (abs(ss-1)<1e-06)
jj=jj+1; lab(jj)=j; ss=0;
end
end
for k=1:d(i)
ss=0;
for j=u(i-1)+1:u(i-1)+u(i)
if (floor(ww(2,j))==k)
for ii=0:jj-1
if (ii==0) tem=0;
else tem=lab(ii);
end
if ((j<=lab(ii+1)) & (j>tem)) break; end
end
ss=ss+wa(ii+1)*ww(3,j);
end; end
wb(k)=ss;
end
iss=strcat(iss,['Weights of layer-',num2str(i+1),' vertices:\n']);
opt=0;
d11(i+1)=1;
for k=1:d(i)
wa(k)=wb(k);
iss=strcat(iss,[num2str(wa(k)),', ']);
if (wa(k)>opt)
opt=wa(k); d11(i+1)=k;
end
end
iss=strcat(iss,'\n');
end
iss=strcat(iss,'Most important vertex in every layer in respect to the start vertex:\n');
for k=2:la+1
iss=strcat(iss,['Layer-',num2str(k),'=',num2str(d11(k)),', ']);
end
fprintf(iss);

function [w,u,iss]=weightAHP(m,d11,d22,d33,d44,dd,iss)
ri=[0.00,0.00,0.58,0.90,1.12,1.21,1.32,1.41,1.45];
ci=0; cr=0; lamda=0; s=0;
w1=zeros(1,m);
b=zeros(m,m);
vv=0;
for i=1:m
for j=i:m

```

```

if (j==i)
b(i,j)=1;
continue;
end;
if (j~i)
vv=vv+1;
b(i,j)=dd(vv);
end
b(j,i)=1/b(i,j);
end; end
s=0;
for i=1:m
w(i)=b(i,1);
for j=2:m
w(i)=w(i)*b(i,j);
end
w(i)=w(i)^(1.0/m);
s=s+w(i);
end
for i=1:m
w(i)=w(i)/s;
end
for i=1:m
w1(i)=0;
for j=1:m
w1(i)=w1(i)+b(i,j)*w(j);
end; end
eigv=eig(b);
lamda=max(eigv);
ci=(lamda-m)/(m-1);
cr=ci/ri(m);
ii=1;
u(ii)=d33(1);
for k=1:(m*m-m)/2
vv=ii;
jj=0;
for i=1:vv
if (u(i)~d33(k)) jj=jj+1; end
end
if (jj==vv)
ii=ii+1;
u(ii)=d33(k);
end
vv=ii;
jj=0;
for i=1:vv
if (u(i)~d44(k)) jj=jj+1; end
end
if (jj==vv)
ii=ii+1;
u(ii)=d44(k);

```

```

end
end
iss=strcat(iss,['Consistency CI and random consistency ratio CR of the matrix of next-layer vertices to the vertex
',num2str(d22(1)),' of layer ',num2str(d11(1)),'\n']);
iss=strcat(iss,['CI=',num2str(ci),' ',' CR=',num2str(cr),'\n']);
if (cr<0.1)
iss=strcat(iss,'The matrix is consistent. It is OK.\n');
iss=strcat(iss,['Weights of next-layer vertices to the vertex ',num2str(d22(1)),' of layer ',num2str(d11(1)),'\n']);
for i=1:m
iss=strcat(iss,['Weight of next-layer vertex ',num2str(u(i)),'=',num2str(w(i)),'\n']);
end
iss=strcat(iss,'\n');
else
iss=strcat(iss,['Due to CR>0.1, matrix of next-layer vertices to the vertex ',num2str(d22(1)),' of layer ',num2str(d11(1)),' is
not consistent. Please try to adjust this matrix and run one more time.\n']);
end

```

4 Application

For example, a food web has 3 trophic layers. Layer 1 is a predator, layer 2 are 5 prey species being predated by the predator, and layer 3 are 3 plant species being fed by preys (Fig. 1). We want to determine the most important plan species for the predator. The data for AHP problem in Fig. 1 is shown in Table 1.

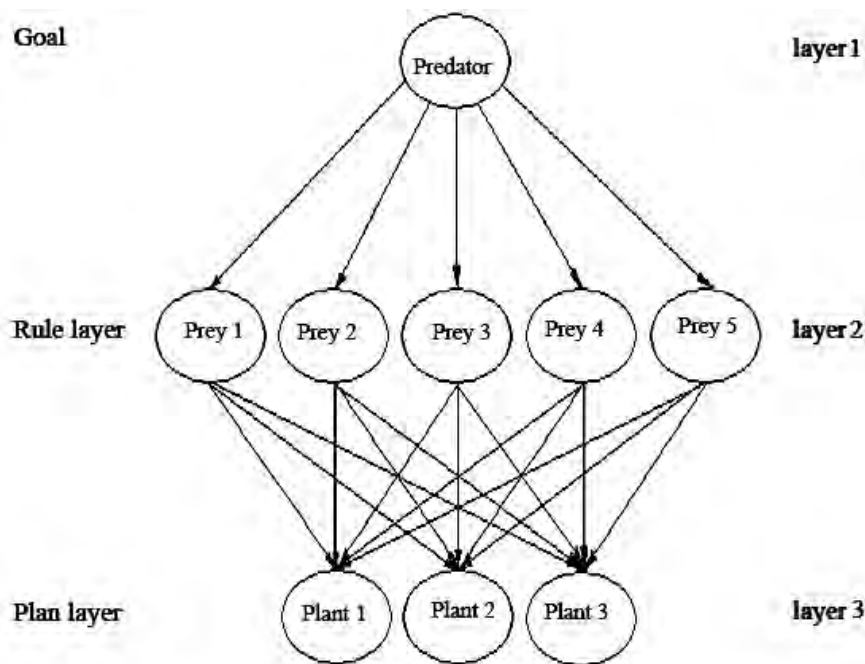


Fig. 1 The hierarchical structure for a food web (Zhang, 2012, 2016).

The results of the Matlab algorithm are

Consistency CI and random consistency ratio CR of the matrix of next-layer vertices to the vertex 5 of layer 2:

CI=0.74489 CR=1.2843

Due to $CR > 0.1$, matrix of next-layer vertices to the vertex 5 of layer 2 is not consistent. Please try to adjust this matrix and run one more time.

Number of vertices for every layer:

Layer-0=1, Layer-1=5, Layer-2=3

Weight of layer-1 vertex:

1.0

Weights of layer-2 vertices:

0.43448, 0.21218, 0.080783, 0.22007, 0.052484

Weights of layer-3 vertices:

0.39957, 0.45452, 0.14591

Most important vertex in every layer in respect to the start vertex:

Layer-2=1, Layer-3=2

At layer 2, the weights of prey 1 to 5 are 0.43448, 0.21218, 0.080783, 0.22007, 0.052484, respectively. Three weights of plant species 1 to 3 at layer 3 are 0.39957, 0.45452, 0.14591, respectively.

Among all prey species, prey species 1 is the most important to the predator.

In present application example, the link weights are unavailable due to $CR > 0.1$. Matrix of next-layer vertices to the vertex 5 of layer 2 is not consistent. We may adjust this matrix and run one more time for perfect results.

Table 1 Data for AHP problem in Fig. 1.

d_1	d_2	d_3	d_4	d
1	1	1	2	3
1	1	1	3	5
1	1	1	4	3
1	1	1	5	4
1	1	2	3	5
1	1	2	4	1
1	1	2	5	3
1	1	3	4	0.3333
1	1	3	5	3
1	1	4	5	6
2	1	1	2	1
2	1	1	3	2
2	1	2	3	5
2	2	1	2	3
2	2	1	3	8
2	2	2	3	5
2	3	1	2	2
2	3	1	3	9
2	3	2	3	2
2	4	1	2	0.5
2	4	1	3	0.3333
2	4	2	3	6
2	5	1	2	2
2	5	1	3	0.25
2	5	2	3	4

d_1, d_2, d_3, d_4 , are labels of present layers, labels of present layer's vertices (from left to right), labels of next layer's vertices (from left to right), and labels of next layer's vertices (from left to right), and d is the weights of the every vertex at next layer over its followed vertex at the same layer.

References

- Choura M, Rebai A, Masmoudi K. 2015. Unraveling the WRKY transcription factors network in *Arabidopsis Thaliana* by integrative approach. *Network Biology*, 5(2): 55-61
- Jiang LQ, Zhang WJ. 2015. Determination of keystone species in CSM food web: A topological analysis of network structure. *Network Biology*, 5(1): 13-33
- Jiang LQ, Zhang WJ, Li X. 2015. Some topological properties of arthropod food webs in paddy fields of South China. *Network Biology*, 5(3): 95-112
- Jiang QF. 1988. *Practical Decision-making Analysis*. Guizhou People's Press, Guiyang, China
- Li JR, Zhang WJ. 2013. Identification of crucial metabolites/reactions in tumor signaling networks. *Network Biology*, 3(4): 121-132
- Zhang GL, Zhang WJ. 2019. Protein-protein interaction network analysis of insecticide resistance molecular mechanism in *Drosophila melanogaster*. *Archives of Insect Biochemistry and Physiology*, 100(1): e21523
- Zhang WJ. 2007. Supervised neural network recognition of habitat zones of rice invertebrates. *Stochastic Environmental Research and Risk Assessment*, 21: 729-735
- Zhang WJ. 2010. *Computational Ecology: Artificial Neural Networks and Their Applications*. World Scientific, Singapore
- Zhang WJ. 2012a. *Computational Ecology: Graphs, Networks and Agent-based Modeling*. World Scientific, Singapore
- Zhang WJ. 2012b. How to construct the statistic network? An association network of herbaceous plants constructed from field sampling. *Network Biology*, 2(2): 57-68
- Zhang WJ. 2012c. Several mathematical methods for identifying crucial nodes in networks. *Network Biology*, 2(4): 121-126
- Zhang WJ. 2016. *Selforganizology: The Science of Self-Organization*. World Scientific, Singapore
- Zhang WJ. 2018. *Fundamentals of Network Biology*. World Scientific Europe, London, UK
- Zhang WJ, Feng YT. 2017. Metabolic pathway of non-alcoholic fatty liver disease: Network properties and robustness. *Network Pharmacology*, 2(1): 1-12
- Zhang WJ, Liu GH, Dai HQ. 2008. Simulation of food intake dynamics of holometabolous insect using functional link artificial neural network. *Stochastic Environmental Research and Risk Assessment*, 22(1): 123-133
- Zhang WJ, Wang R, Zhang DL, et al. 2014. Interspecific associations of weed species around rice fields in Pearl River Delta, China: A regional survey. *Selforganizology*, 1(3-4): 143-205
- Zhang WJ, Zhan CY. 2011. An algorithm for calculation of degree distribution and detection of network type: with application in food webs. *Network Biology*, 1(3-4): 159-170
- Zhang WJ, Zhang XY. 2008. Neural network modeling of survival dynamics of holometabolous insects: a case study. *Ecological Modelling*, 211: 433-443