Article

Phase recognition in network evolution

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 8 February 2016; *Accepted 21 March 2016*; *Published 1 September 2017*

Abstract
In present study I used one dimensional ordered cluster method to recognize different phases in network evolution by clustering time points in a time series. Matlab codes of the algorithm were provided.

Keywords network evolution; time series; ordered clustering; phase recognition; algorithm.

Selforganizology
ISSN 2410-0080
URL: http://www.iaees.org/publications/journals/selforganizology/online-version.asp
RSS: http://www.iaees.org/publications/journals/selforganizology/rss.xml
E-mail: selforganizology@iaees.org
Editor-in-Chief: WenJun Zhang
Publisher: International Academy of Ecology and Environmental Sciences

1 Introduction

Network evolution is a time series. In the process of evolution, network properties (topological structure, etc.) change as time (Zhang, 2012, 2015, 2016), and the change is not homogeneous. Sometimes we want to classify the time series into several phases according to amplitude of the network change. In present study, I use one dimensional ordered cluster method to recognize different phases in network evolution by clustering time points in a time series. Matlab codes of the algorithm are presented for further use.

2 Algorithm

I use one dimensional ordered cluster method (Zhang and Fang, 1982; Qi, 2005) to recognize different phases in network evolution by clustering time points in a time series. Suppose a network has m properties and it evolves through n time points in the time series. The raw data for network evolution is $(x_{ij})_{m*n}$. First, if we choose to standardize the raw data into the matrix (a_{ij}), calculate

$$a_{ij} = (x_{ij} - x_{bi})/s_i$$

where

$$x_{bi} = \sum_{j=1}^{n} x_{ij} / n$$

$$s_i = (\sum_{j=1}^{n} (x_{ij} - x_{bi})^2/(n-1))^{1/2} \qquad i=1,2,\ldots,m$$

If data standardization is not needed, let $a_{ij} = x_{ij}$, i=1,2,…,m; j=1,2,…,n.

Different properties have different importance in the network. Thus each property can be differently weighted. Suppose the weights of m properties are w_i, i=1,2,…,m, where $\sum w_i = 1$. The weighted data matrix is $a_{ij}=w_i \times a_{ij}$, i=1,2,…,m; j=1,2,…,n.

Calculate the distance between adjacent two time points in network evolution, i.e., Euclidean distance, Manhattan distance, Pearson correlation based distance, or Jaccard distance

$$e_{i\,i+1}=(\sum_{k=1,\ldots,m}(a_{ki}-a_{k\,i+1})^2/m)^{1/2}$$
$$m_{i\,i+1}=\sum_{k=1,\ldots,m}|a_{ki}-a_{k\,i+1}|/m$$
$$c_{i\,i+1}=\sum_{k=1,\ldots,m}(a_{ki}\times a_{k\,i+1})/(\sum_{k=1,\ldots,m}a_{ki}^2\times\sum_{k=1,\ldots,m}a_{k\,i+1}^2)^{1/2}$$
$$d_{i\,i+1}=(b_i+b_{i+1})/(c_i+c_{i+1}-e)$$
$$i=1,2,\ldots,n-1$$

where b_i is the number of properties of value 1 for time point i but value 0 for time point i+1; b_{i+1} is the number of properties of value 1 for time point i+1 but value 0 for time point i; c_i and c_{i+1} are the numbers of properties of value 1 for time point i and i+1 respectively. e is the number of properties of value 1 for both time point i and time point i+1.

According to different distance measures chosen, let $r_{i\,i+1}=e_{i\,i+1}$, or $r_{i\,i+1}=m_{i\,i+1}$, or $r_{i\,i+1}=1-c_{i\,i+1}$, or $r_{i\,i+1}=d_{i\,i+1}$. Find the shortest distance $r_{i\,i+1}$, i=1,2,…,n-1, and combine the two corresponding time points i and i+1 into the same cluster. Similarly, take the minimum between-time point distance as the between-cluster distance. Find the two clusters with minimum between-cluster distance and combine into a single cluster. Finally all time points are combined into a single cluster. The minimum between-cluster distance in a clustering hierarchical level is defined as the between-cluster distance of this hierarchical level.

The following are the Matlab codes, NetEvolPhase.m, for the algorithm

```
raw=input('Input the excel file name of property-by-time point data (e.g., raw.xls, etc.)((xij)m*n, m: number of properties; n: number of time points): ','s');
sel=input('Data standardization or not (1: Data standardization; 2: No standardization): ');
selw=input('Weighting properties or not (1: No weighting; 2: Weighting properties): ');
sele=input('Choose distance measures (1: Euclidean distance; 2: Manhattan distance; 3: Pearson correlation; 4: Jaccard coefficient): ');
araw=xlsread(raw);
m=size(araw,1); n=size(araw,2);
a=araw';
if (selw==2)
weights=input('Input the excel file name of weights of properties (e.g., weights.xls, etc.)(a row of m weights of properties): ','s');
w=xlsread(weights);
end
q=zeros(1,m);
p=zeros(1,m);
r=zeros(1,n-1);
r1=zeros(1,n-1);
pp=zeros(1,n-1);
am=zeros(1,n-1);
x=zeros(n,n);
```

```
nu=zeros(1,n);
if (sel==1)
for i=1:m
p(i)=sum(a(:,i))/n;
q(i)=sum((a(:,i)-p(i)).^2);
q(i)=sqrt(q(i)/(n-1));
for j=1:n
a(j,i)=(a(j,i)-p(i))/q(i);
end; end; end
if (selw==2)
dum=sum(w);
if (dum~=1) w=w/dum; end
for i=1:m
a(:,i)=a(:,i)*w(i);
end; end
disp('Distance Between Two Adjacent Time Points');
for i=1:n-1
switch sele
    case 1
      aa=sum((a(i,:)-a(i+1,:)).^2);
      r1(i)=sqrt(aa/m);
    case 2
      aa=sum(abs(a(i,:)-a(i+1,:)));
      r1(i)=aa/m;
    case 3
      aa=0; bb=0; cc=0;
      aa=sum(a(i,:).*a(i+1,:));
      bb=sum(a(i,:).^2);
      cc=sum(a(i+1,:).^2);
      r1(i)=aa/sqrt(bb*cc);
    case 4
      aa1=0; bb1=0; cc1=0; rr1=0; nn1=0;
      for k=1:m
      if (abs(a(i,k))>=1e-08) aa1=aa1+1; end
      if (abs(a(i+1,k))>=1e-08) bb1=bb1+1; end
      if ((abs(a(i,k))>1e-08) & (abs(a(i+1,k))>=1e-08)) cc1=cc1+1; end
      if ((abs(a(i,k))<=1e-08) & (abs(a(i+1,k))>=1e-08)) rr1=rr1+1; end
      if ((abs(a(i,k))>=1e-08) & (abs(a(i+1,k))<=1e-08)) nn1=nn1+1; end
      end
      if ((aa1+bb1-cc1)==0) r1(i)=1;
      else r1(i)=(nn1+rr1)/(aa1+bb1-cc1);
      end
      end;
if ((sele==1) | (sele==2) | (sele==4)) r(i)=r1(i);
else if (sele==3) r(i)=1-r1(i); end
```

```
end; end
for i=1:n-1
fprintf(['r(' num2str(i) '    ' num2str(i+1) ')          ']);
end
fprintf('\n');
for i=1:n-1
fprintf([num2str(r(i)) '      ']);
end
fprintf('\n\n');
pp=r;
for i=1:n-2
k=i;
for j=i:n-2
if (pp(j+1)<=pp(k)) k=j+1; end
end
aa=pp(i);
pp(i)=pp(k);
pp(k)=aa;
end
bb1=1;
for k=0:n-1
if (k==0) u=0; else u=pp(k); end
lab1=0;
for i=0:k-1
if (i==0) v=0; else v=pp(i); end
if (v==u) lab1=1; break; end
end
if (lab1==1) continue; end
for i=1:n-1
if ((r(i)-u)<1e-06) am(i)=1;
else am(i)=0; end
end
i=1;
nu(bb1)=1;
lab3=0;
while (i<=n-1)
lab2=0;
if (am(i)==0) x(bb1,i)=nu(bb1); x(bb1,i+1)=nu(bb1)+1; end
if (am(i)==1)
for j=i:n-1
if (am(j)==0) i=j; lab2=1; break;
else x(bb1,j)=nu(bb1); x(bb1,j+1)=nu(bb1);
if (j==n-1) lab3=1; break; end
end; end;
if (lab2==1) continue; end
```

```
if (lab3==1) break; end
end
i=i+1;
nu(bb1)=nu(bb1)+1;
end
fprintf('\n');
fprintf(['Distance=' num2str(u) '\n']);
%u(bb1)=u;
for i=1:nu(bb1)
fprintf('(');
for j=1:n
if (x(bb1,j)==i)
fprintf([num2str(j) '    ']);
end; end
fprintf(')    ');
end
bb1=bb1+1;
end
fprintf('\n');
```

3 Application Example

Here I use a series of data on network evolution, which are derived from the model of Zhang (2016) (Table 1).

Table 1 A series of data on network evolution, derived from the model of Zhang (2016).

Time points	1	2	3	4	5	6	7	8	9	10
Connectance	0.0555	0.0588	0.0612	0.0637	0.0669	0.0686	0.0718	0.0751	0.08	0.0833
Entropy	0.8735	1.5951	1.9388	1.8857	2.0686	2.0588	2.3265	2.7869	3.5429	4.0359
Coefficient of variation	1.3211	1.5539	1.6463	1.6044	1.6307	1.6127	1.6609	1.7573	1.9038	1.9892
Aggregation index	1.1166	1.1898	1.2125	1.1911	1.1896	1.1798	1.1851	1.2028	1.2271	1.2388
Time points	11	12	13	14	15	16	17	18	19	20
Connectance	0.0882	0.0906	0.0939	0.0971	0.1004	0.1037	0.1078	0.1135	0.1184	0.12
Entropy	4.6367	4.9543	6.0531	6.2833	6.8694	7.4441	8.191	9.3037	9.4653	9.9012
Coefficient of variation	2.0733	2.1158	2.3159	2.32	2.3962	2.4654	2.5513	2.6733	2.6319	2.6839
Aggregation index	1.2446	1.2474	1.2816	1.2729	1.2792	1.2838	1.289	1.296	1.2767	1.2816

Use the algorithm above, with data standardization, no weighting properties (4 properties) and Euclidean distance, the resultant recognition of phases of network evolution is indicated in Table 2. In Table 2, for the same hierarchical level, the time points with the same background color belong to the same phase, and the time points without background color are independent time points.

Table 2 Recognition of phases of network evolution.

Hierar. Level	Distance	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	0.091	1	2	3	**4**	**5**	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
2	0.098	1	2	3	**4**	**5**	6	7	8	9	10	**11**	**12**	13	14	15	16	17	18	19	20
3	0.109	1	2	3	**4**	**5**	**6**	7	8	9	10	**11**	**12**	13	14	15	16	17	18	19	20
4	0.114	1	2	3	**4**	**5**	**6**	7	8	9	10	**11**	**12**	13	14	15	16	17	18	**19**	**20**
5	0.119	1	2	3	**4**	**5**	**6**	**7**	8	9	10	**11**	**12**	13	14	15	16	17	18	**19**	**20**
6	0.124	1	2	3	**4**	**5**	**6**	**7**	8	9	10	**11**	**12**	**13**	**14**	15	16	17	18	**19**	**20**
7	0.157	1	2	3	**4**	**5**	**6**	**7**	8	9	10	**11**	**12**	**13**	**14**	**15**	**16**	17	18	**19**	**20**
8	0.168	1	2	3	**4**	**5**	**6**	**7**	8	9	10	**11**	**12**	**13**	**14**	**15**	**16**	17	18	**19**	**20**
9	0.193	1	2	3	**4**	**5**	**6**	**7**	8	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	17	18	**19**	**20**
10	0.194	1	2	3	**4**	**5**	**6**	**7**	8	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	17	18	**19**	**20**
11	0.197	1	2	3	**4**	**5**	**6**	**7**	8	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	18	**19**	**20**
12	0.231	1	2	**3**	**4**	**5**	**6**	**7**	8	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	18	**19**	**20**
13	0.239	1	2	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**
14	0.267	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**
15	0.283	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**
16	0.346	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**
17	0.464	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**
18	0.805	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**

Acknowledgment

We are thankful to the support of Discovery and Crucial Node Analysis of Important Biological and Social Networks (2015.6-2020.6), from Yangling Institute of Modern Agricultural Standardization, and High-Quality Textbook *Network Biology* Project for Engineering of Teaching Quality and Teaching Reform of Undergraduate Universities of Guangdong Province (2015.6-2018.6), from Department of Education of Guangdong Province, China.

References

Qi YH. 2005. The web software for ordered cluster analysis of sequential information. Information Science, 23(Suppl.): 99-101

Zhang WJ. 2012. Modeling community succession and assembly: A novel method for network evolution. Network Biology, 2(2): 69-78

Zhang WJ, 2015. A generalized network evolution model and self-organization theory on community assembly. Selforganizology, 2(3): 55-64

Zhang WJ. 2016. A random network based, node attraction facilitated network evolution method. Selforganizology, 3(1): 1-9

Zhang YT, Fang KT. 1982. Introduction to Multivariate Statistics. Science Press, Beijing, China