Article

A comprehensive risk assessment system for probabilistic problems
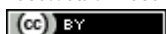
WenJun Zhang[1], **YanHong Qi**[2], **Xin Li**[3]

[1]School of Life Sciences, Sun Yat-sen University, Guangzhou 510275, China

[2]Sun Yat-sen University Libraries, Sun Yat-sen University, Guangzhou 510275, China

[3]College of Plant Protection, Northwest A & F University, Yangling 712100, Shaanxi, China

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

Abstract

In present study we provided a comprehensive risk assessment system. The system consists of four representative methods, namely probability-deterministic assessment, probability-interval assessment, probability-ranking assessment, and probability-ranking with optimal mix strategy. Among them, the probability-interval assessment was proposed by us. In the risk assessment, there are multiple available states, but only one of them can occur in nature. The occurrence probability of each state is the determined value and interval respectively for the first two methods; and for the latter two methods it is the probability difference of each pair of adjacent states. Known the benefit matrix, plans ranking can be derived from the first three methods according to the expected benefit; the fourth method can be used to obtain the optimal mix proportion of plans. The Matlab full code of the assessment system was given for further application and improvement.

1 Introduction

Risk assessment problems are widely found in various fields such as industry and agriculture, economy and trade, and military affairs. For example, the weather in the following year may be wet, moderate, or dry. The probability of occurrence of these weathers, according to our current level of understanding, may be a certain value, interval, or probability difference, etc., and we want to determine the planting plan to maximize the benefits on the basis of the occurrence probability, which is categorized to multi-attribute and risk assessment problems (von Neumann and Morgenstern, 1944; Navarrete et al., 1979; Hwang and Yoon, 1981; Kmietowicz and Pearman, 1981; Chen, 1987; Huang et al., 1988; Qi et al., 1990; Zhang et al., 1991a-b, 1996, 2004, 2017; Zhang and Gu, 1996; Qi, 2002, 2003; Zhang, 2007, 2012, 2016; Ferrarini, 2012). Based on the past and present research, we provided a comprehensive risk assessment system to improve the scientific nature of risk assessment and reduce and avoid the influence of subjective factors. Matlab is a well-known and large-scale

software package for scientific and engineering calculation, suitable for large-scale data calculation and analysis. Therefore, the risk assessment system in present research is implemented by Matlab.

2 Methods

In the assessment system, benefits and costs are two aspects of the same problem. For the cost type problem, all values in cost matrix change signs and thus the cost matrix is converted into a benefit matrix.

2.1 Probability-deterministic assessment

There are m available states, only one of which can occur in nature. The probability of occurrence of each state, p_i, is known, $\sum p_i = 1$, $i = 1, 2, ..., m$. Known n plans, and the benefit matrix is (x_{ij}), $i = 1, 2, ..., n$; $j = 1, 2, ..., m$. Calculate the expected benefit of each plan

$$d_i = \sum_{j=1}^{m} x_{ij} p_j \qquad i = 1, 2, ..., n$$

According to d_i, $i = 1, 2, ..., n$, the plans are ranked from large to small values, and the closer to the front, the better the plan. The top plan in the ranking list is the most preferred option.

2.2 Probability-interval assessment

In this study, we propose the following probability-interval assessment. The basic principle is that, through the randomization process, the probability-interval is converted to a certain probability, and then the expected benefit of the given plan is calculated.

There are m available states, but only one of which can occur in nature. The probability of occurrence of each state, p_i, is unknown, but falls in the interval $a_i \leq p_i \leq b_i$, where, $\sum a_i \leq 1$, $\sum b_i \geq 1$, $a_i \leq b_i < 1$, $a_i > 0$, $i = 1, 2, ..., m$.

Known n plans, and the benefit matrix is (x_{ij}), $i = 1, 2, ..., n$; $j = 1, 2, ..., m$. First, let the number of randomizations be s and the number of initial randomizations $t = 1$. The method is as follows:

(1) Generate m uniformly distributed (0, 1) random values p_i, satisfying $a_i \leq p_i \leq b_i$, $i = 1, 2, ..., m$. To calculate

$$r_j = \sum_{i=1}^{j} p_i \qquad j = 1, 2, ..., m$$
$$q_i = r_i / r_m \qquad i = 1, 2, ..., m$$
$$q_0 = 0$$

(2) Generate a uniformly distributed (0, 1) random value v. If $q_{i-1} < v \leq q_i$, $i = 1, ..., m$, the state i occurs, let $u_{it} = 1$, $u_{jt} = 0$, $j = 1, 2, ..., m$, $j \neq i$.

(3) If the number of randomizations $t < s$, let $t = t+1$ and return (1). If $t = s$, go to the next step.

(4) Calculate the probability of occurrence of each state

$$w_i = \sum_{t=1}^{s} u_{it} / s \qquad i = 1, 2, ..., m$$

(5) Calculate the expected benefit of each plan

$$d_i = \sum_{j=1}^{m} x_{ij} w_j \qquad i = 1, 2, ..., n$$

According to d_i, $i = 1, 2, ..., n$, the plans are ranked from large to small values, and the closer to the front, the better the plan. The top plan in the ranking list is the most preferred option.

The general principle for setting the number of randomizations (s) is, s should increase as m and max (b_i-a_i), e.g., let $s=100m(0.1+max (b_i-a_i))$.

2.3 Probability-ranking assessment

There are m available states, but only one of which can occur in nature. The probability of occurrence of each state, p_i, is unknown but, $p_1 \geq p_2 \geq \ldots \geq p_m > 0$, $i=1, 2, ..., m$, and

$$p_i - p_{i+1} \geq m_i \qquad i=1, 2, ..., m$$
$$p_{m+1} = 0$$

where, $m_i \geq 0$, $\sum i \times m_i < 1$. Under the state p_i, the benefit of a plan is x_i, $i=1, 2, ..., m$. We pursue the maximal benefit

$$max\ E = \sum p_i \times x_i$$
$$\sum p_i = 1$$
$$p_i - p_{i+1} \geq m_i \qquad i=1, 2, ..., m$$

Let $r_i = q_i - m_i$, $i=1, 2, ..., m$; $q_i = p_i - p_{i+1}$, $i=1, 2, ..., m-1$; $q_i = p_m$, $i=m$, then the solution of the problem is

$$r_k = 1 - \sum i \times m_i / k \qquad k=1, 2, ..., m$$
$$r_j = 0 \qquad j=1, 2, ..., m; j \neq k$$

Thereafter, replace r_i in $max\ E = \sum r_i \times y_i + \sum m_i \times y_i$ with the solution, and achieve n pairs of ($max\ E$, $min\ E$). Solve $Emax=max\ max\ E$, $Emin=min\ min\ E$. Suppose $0 \leq \alpha \leq 1$, and the expected benefit of eachplan is calculated

$$f = \alpha \times Emax + (1-\alpha) \times Emin$$

Then, for the problem $max\ E$, the plan with the larger f value is a better solution. Here, α is the compromise factor. If the ranking of occurrence of each state is more believable and the value of m_i is larger, then a larger α can be taken (Kmietowicz and Pearman, 1981; Huang et al., 1988; Zhang et al., 2004; Zhang, 2007; Zhang et al., 2017).

2.4 Probability-ranking with optimal mix strategy

There are m available states, n plans, and the benefit matrix is (x_{ij}), $i=1, 2, ..., n; j=1, 2, ..., m$. The proportion of each plan is q_i, $i=1, 2, ..., n$, $\sum q_i = 1$; only one of the states can occur in nature, and the probability of occurrence of each state, p_j, is unknown, but $p_1 \geq p_2 \geq \ldots \geq p_m > 0$, $j=1, 2, ..., m$, and

$$p_j - p_{j+1} \geq M_j \qquad j=1, 2, ..., m$$
$$p_{m+1} = 0$$

where, $M_j \geq 0$, $\sum j \times M_j < 1$. We pursue the maximal benefit (or minimal cost)

$$min\ F$$
$$\sum_{j=1}^{m} p_j \times x_{ij} - F \leq 0$$
$$p_j - p_{j+1} \geq M_j \qquad j=1, 2, ..., m$$
$$\sum p_j = 1$$

$$p_j \geq 0 \qquad j=1, 2, ..., m$$

This problem is equivalent to solving the optimal mix proportion q_i, $i=1, 2, ..., n$, $\sum q_i=1$, which is a linear programming problem (Huang et al., 1988)

$$\min Z = -F - \sum M_i \times h_i$$
$$-\sum q_i \times x_{i1} + h_1 + F \leq 0$$
$$-\sum q_i \times x_{ik} - h_{k-1} + h_k + F \leq 0 \qquad k=2, 3, ..., m$$
$$\sum q_i = 1$$
$$q_i \geq 0$$
$$h_j \geq 0 \qquad i=1, 2, ..., n; \quad j=1, 2, ..., m$$

3 Matlab Codes

The following are the Matlab full codes of the assessment system, riskAssess.m

```
valmat=input('Enter the excel file name of the benefit matrix, for example, raw.xls. The benefit matrix x=(xij)n*m, rows are the plans and columns are the available states: ','s');
x=xlsread(valmat);
m=size(x,2); n=size(x,1);
type=input('Enter the type of risk assessment (1: probability-deterministic assessment; 2: probability-interval assessment; 3: probability-ranking assessment; 4: probability-ranking with optimal mix strategy): ');
if (type==1)
prob=input('Enter the excel file name of the occurrence probabilities of available states, for example, prob.xls.: ','s');
disp('The file has 1 row and m columns, which store the occurrence probability of each state, pi, i=1,2,...,m.')
p=xlsread(prob);
if (abs(sum(p)-1)>0.1)
disp('The sum of the probabilities of all states is required to be equal to 1, please check and correct and start again.');
pause;
end
d=p*x';
rr=sortrows([(1:n)' d'],-2);
fprintf('Ranking   Plan   Expected benefit\n');
rk=[(1:n)' rr];
disp([rk])
end
if (type==2)
interv=input('Enter the excel file name of the probability intervals of available states, for example,, interv.xls.: ','s');
disp('The file has two rows and m columns. The first row and the second row store the lower limit and upper limit of the interval respectively; if the lower limit and upper limit are equal, then the two values in the first row and the second row are the same.')
interv=xlsread(interv);
de=sum(interv)==0;
if (sum(de)~=0)
disp('The following states are meaningless in this system, please delete them from the benefit matrix and the probability-interval matrix, and start again.');
find(de);
```

```
pause;

end

if (sum(interv(1,:))>1)

disp('The sum of the lower limits of the probability intervals must be less than or equal to 1, please check and correct and start

again.');

pause;

end

if (sum(interv(2,:))<1)

disp('The sum of the upper limit of the probability intervals must be greater than or equal to 1, please check and correct and start

again.');

pause;

end

s=round(100*m*(0.1+max(interv(2,:)-interv(1,:))));

r=zeros(1,m); q=zeros(1,m);

u=zeros(m,s); w=zeros(1,m);

d=zeros(1,m); tem=zeros(1,m);

t=1;

while (t<=s)

tem=rand()*(interv(2,:)-interv(1,:))+interv(1,:);

r(1)=tem(1);

for i=2:m

r(i)=r(i-1)+tem(i);

end

q=r/r(m);

ran=rand();

for i=1:m

if (i==1) re=0;

else re=q(i-1);

end

if ((ran>re) & (ran<=q(i))) u(i,t)=1; break; end

end

t=t+1;

end

w=sum(u')/s;

d=w*x';

rr=sortrows([(1:n)' d'],-2);

fprintf('Ranking    Plan    Expected benefit\n');

rk=[(1:n)' rr];

disp([rk])

end

if (type==3)

mi=input('Enter the excel file name of the probability difference of each pair of adjacent states, for example, mi.xls.: ','s');

disp('The file has 1 row and m columns, store the probability difference of each pair of adjacent states, mi, i=1,2,...,m.')

alpha=input('Enter the compromise coefficient, the value is within (0,1): ');

mi=xlsread(mi);
```

```
if (sum((1:m).*mi)>=1)

disp('The occurrence probability mi of each pair of adjacent states must satisfy ∑i*mi<1. Please make adjustment and start

again.');

pause;

end

w=zeros(1,m); s=zeros(1,n); d=zeros(1,n);

for k=1:n

for i=1:m

t=0;

for j=1:i

t=t+x(k,j);

end

w(i)=t/i;

end

g=w(1);

ll=w(1);

for i=1:m

if (w(i)>g) g=w(i); end

if (w(i)<ll) ll=w(i); end

end

pp=0; u=0;

for j=1:m

q=0;

for i=1:j

q=q+x(k,i);

end

pp=pp+mi(j)*q;

u=u+j*mi(j);

end

s(k)=(1-u)*g+pp;

d(k)=(1-u)*ll+pp;

end

d=alpha*s+(1-alpha)*d;

rr=sortrows([(1:n)' d'],-2);

fprintf('Ranking   Plan    Expected benefit\n');

rk=[(1:n)' rr];

disp([rk])

end

if (type==4)

mi=input('Enter the excel file name of the probability difference of each pair of adjacent states, for example, mi.xls.: ','s');

disp('The file has 1 row and m columns, store the probability difference of each pair of adjacent states, mi, i=1,2,...,m.')

mi=xlsread(mi);

if (sum((1:m).*mi)>=1)

disp('The occurrence probability mi of each pair of adjacent states must satisfy ∑i*mi<1. Please make adjustment and start

again.');
```

```
pause;
end
A=zeros(n+2*m+1,n+m+1);
b=zeros(n+2*m+1,1);
c=[zeros(1,n) -mi -1];
A(1,:)=[-x(:,1)' 1 zeros(1,m-1) 1];
for i=2:m
A(i,1:n)=-x(:,i)';
A(i,n+i-1)=-1; A(i,n+i)=1; A(i,n+m+1)=1;
end
Aeq=[ones(1,n) zeros(1,m+1)];
beq=1;
VLB(1:n+m)=zeros(n+m,1);
VLB(n+m+1)=-Inf;
VUB=[];
xx=linprog(c,A,b,Aeq,beq,VLB,VUB);
disp('Optimal mix proportion of each plan')
disp([xx(1:n)])
end
```

Also, the full Matlab codes listed above, an demo data file, and an executable file package (Fig. 1), as the supplementary material (Zhang-Supplementary-Material.rar) of the present article, can be downloaded at http://www.iaees.org/publications/ journals/ces/articles/2023-13(2)/2-Zhang-Abstract.asp.

Fig. 1 Command window of software running (Matlab environment).

4 Demostration

Suppose there are 5 available states and 6 plans. The benefit matrix is as the following

$$
\begin{array}{ccccc}
6 & 3 & 7 & 6 & 4 \\
5 & 4 & 5 & 7 & 5 \\
1 & 8 & 5 & 4 & 7 \\
5 & 5 & 2 & 2 & 3 \\
3 & 2 & 3 & 6 & 5 \\
3 & 6 & 7 & 5 & 2 \\
\end{array}
$$

For the probability-interval assessment, the probability interval of occurrence of each state is as follows

$$
\begin{array}{ccccc}
0.2 & 0.1 & 0.2 & 0.1 & 0.2 \\
0.5 & 0.2 & 0.6 & 0.3 & 0.4 \\
\end{array}
$$

, then the ranking for plans is

Ranking	Plan	Expected benefit
1	1	5.4800
2	2	5.1800
3	3	4.6440
4	6	4.5400
5	5	3.7480
6	4	3.3240

For the probability-ranking with optimal mix strategy, the probability difference of each pairs of adjacent states is 0.1, 0.1, 0.05, 0.05, 0.05, respectively (apparently $\sum i \times m_i = 0.9 < 1$), then the optimal mix proportion of plans is calculated as: 1.0, 0, 0, 0, 0, 0, that is, plan 1 is the only option.

References

Chen T. 1987. Decision-making Analysis. Science Press, Beijing, China

Ferrarini A. 2012. WORTHY: a new model for ecological ranking and evaluation. Computational Ecology and Software, 2(4): 213-219

Huang MF, Zhang H, Yuan W. 1988. New Practical Techniques for Decision-making. Guizhou People's Press, Guiyang, China

Kmietowicz ZW, Pearman AD. 1981. Decision Theory and Incomplete Knowledge. Gower Publ, Hampshire, UK

Qi YH. 2002. An improved algorithm and network computing software of mix preference map and its application in evaluation of books and journals. Modern Computer, 11: 14-16, 56

Qi YH. 2003. A network distributed computing software for assessment and analysis of books and periodicals. Journal of Intelligence, 1: 43-45

Qi YH, Wang DX, Zhang WJ. 1990. Analysis of resistance indices of wheat to Fusarium graminerum Schw. and identification in batches using computers, Acta Phytopathologica Sinica, 20(2): 147-152

Zhang WJ. 2007. Methodology on Ecology Research. Sun Yat-sen University Press, Guangzhou, China

Zhang WJ. 2012. Computational Ecology: Graphs, Networks and Agent-based Modeling. World Scientific, Singapore

Zhang WJ. 2016. Selforganizology: The Science of Self-Organization. World Scientific, Singapore

Zhang WJ, Feng YJ, Liu GH. 2004. Network program of probability ranking based decision making and its application in risk analysis of agri-biodiversity. Computer Application Research of Computers, 21(1): 28-30

Zhang WJ, Gu DX. 1996. Kind of algorithm for stochastic decision-making in insect pest management. Bulletin of Science and Technology, 12(5): 288-293

Zhang WJ, Li ZQ, Qi YH. 1991a. A study on appraising disease resistance of wheat using MOD Theory. Acta Phytopathologica Sinica, 21(3): 223-228

Zhang WJ, Qi YH, Li X. 2017. MATASS: the software for multi-attribute assessment problems. Computational Ecology and Software, 7(2): 38-48

Zhang WJ, Qi YH, Gu DX. 1996. Objective system and benefit evaluation for microbial control of pests. Journal of Biomathematics, 11(5): 43-51

Zhang WJ, Qi YH, Wang DX. 1991b. Algorithm of drought resistance attributes and evaluation of its decision-making in wheat seedling stage. Agricultural Research in the Arid Areas, 2: 100-105