ISSN No. 0976-5697

Volume 6, No. 1, Jan-Feb 2015

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

Programming Development on Risk Function

Igbinehi,E. M.*
Department of mathematics/statistics
Federal Polytechnic Offa Nigeria
akeemwaju@yahoo.com

Adeoye Akeem .O
Department of mathematics/statistics
Federal Polytechnic Offa Nigeria
akeemwaju@yahoo.com

Ojo .O.D
Department of mathematics/statistics
Federal Polytechnic Offa Nigeria
ifemi24@yahoo.com

Taofeek-Ibrahim Fatimoh Abidem
Department of Computer Science
Federal Polytechnic Offa Nigeria
fatty_fatty2@yahoo.com.au

Abstract: This research, programming development on risk function was aimed to simplify how Loss table, minimax action, maxmini action ,minimini action and maxmax action can be obtain.. C++ program was use to developed a programming on the Loss table, minimax action, maxmini action , minimini action, maxmax action and Risk function.This program was discovered to be accurate. We thereby recommend the progamm for uses.

Keywords: Minimax, Maxmini, Minimini, Maxmax and Loss table.

I. INTRODUCTION

Statistical inference is the process of drawing conclusions from data that are subject to random variation, Statistical inference, drawing conclusions about the whole population on the basis of a sample[4],[6]. The outcome of statistical inference may be an answer to the question "what should be done next?", where this might be a decision about making further experiments or surveys, or about drawing a conclusion before implementing some organizational or governmental policy[3]. Decision procedures: A decision procedure or decision rule specifies how we use the data to choose an action a. A decision procedure δ is a function $\delta(\boldsymbol{X})$ from the sample space of the experiment to the action space[1],[2],[5].

For Example 1, decision procedures include

$$\delta(\boldsymbol{X}) = \frac{\sum_{i=1}^{n} X_i}{n}$$

and

$$\delta(\boldsymbol{X}) = \frac{1+\sum_{i=1}^{n} X_i}{n+1}.$$

a. Risk function: The loss of a decision procedure will vary over repetitions of the experiment because the data from the experiment $\boldsymbol{X}$ is random. The risk function $R(\boldsymbol{\theta},\delta)$ is the expected loss from using the decision procedure δ when the true parameter vector is $\boldsymbol{\theta}$:

$$R(\boldsymbol{\theta},\delta) = E_{\theta}[l(\boldsymbol{\theta},\delta(\boldsymbol{X}))]$$

II. PROGRAM SOURCE CODE

```
 #include<iostream.h>
#include<math.h>
#include<conio.h>
#include<cstring.h>
#include<stdlib.h>
void menu_name();
void menu_name()
{
string skey;
menu_name();
 int dtable[20][20];
 int theta, alpha,a,num_d, num_k,i,j,k,tmp;
 float                                    R_Q1_D1,prob1,prob2,
a_d[8][32],max_val[32],min_val[32],d[32];
 int x1=0; int x2=1;
 string keep_chm[5],keep_chx[5],tmp_ch;
 alpha=2;
 theta=2;
 cout<<"Enter probability 1=";cin>>prob1;
 cout<<"Enter probability 2=";cin>>prob2;
 //determine the number of d
 num_d=pow(2,theta);
 //compute number of iteration
 num_k=2*(theta-1);
 //drawing the data table;
 gotoxy(5,5);cout<<"   | Q1 | Q2|";
 gotoxy(5,6);cout<<"------------";
 gotoxy(5,7);cout<<"@1 |   |  |";
 gotoxy(5,8);cout<<"------------";
 gotoxy(5,9);cout<<"@1 |   |  |";

 //accepting value for data table
 int col=10;
 int row=7;
 for(i=1; i<=alpha; i++)
           {
             for(j=1; j<=theta; j++)
```

```
            {

        gotoxy(col,row);cin>>dtable[i][j];
                    col=col+4;
            }
            row=row+2;
            col=10;
        }
 cout<<"\n";
 //computation of Risk function of R(Qn@n)
a=2;
int rw=1;
int cl=1;
        for(k=1; k<=num_k; k++)
        {
        for(i=1; i<=theta; i++)
            {
                for(j=1; j<=alpha; j++)
                {

R_Q1_D1=(dtable[i][j]*prob1)+(dtable[a][j]*prob2);
                        a_d[rw][cl]=R_Q1_D1;
                        cout<<"R(Q"<<j<<"
D"<<i<<")="<<R_Q1_D1<<endl;
                        rw=rw+1;
                }
                cl=cl+1;
                rw=1;
                a=a-1;
                if(a==0){a=2;}
            }
        a=a-1;
        }
col=10;
row=22;
//drawing the risk table;
 gotoxy(5,20);cout<<"   | d1 | d2| d3 | d4 |";
 gotoxy(5,21);cout<<"----------------------";
 gotoxy(5,22);cout<<"Q1 |    |   |   |    |";
 gotoxy(5,23);cout<<"----------------------";
 gotoxy(5,24);cout<<"Q2 |    |   |   |    |";
 gotoxy(5,25);cout<<"----------------------";
 gotoxy(5,26);cout<<"   |    |   |   |    |";

        for(i=1; i<=theta; i++)
            {
                for(j=1; j<=num_d; j++)
                {

        gotoxy(col,row);cout<<a_d[i][j];
       if(i==num_d)
                        {
                        col=col+5;
                        }
                        else
                        {
                        col=col+4;
                        }
                }
                row=row+2;
                col=10;
            }
//---------------computation of the risk table to get D-----------
--------
 a=2;
        for(i=1; i<theta; i++)
            {
                for(j=1; j<=num_d; j++)
                {

        d[j]=(a_d[i][j]*prob1)+(a_d[a][j]*prob2);
                }
            }
 row=26;
 col=10;
 //----------output of d1 to d4-------------------------
        for(i=1; i<=num_d; i++)
        {
            gotoxy(col,row);cout<<d[i];
            if(i==num_d)
            {
             col=col+5;
            }
            else
            {
            col=col+4;
            }
        }
//--------------determine the maximum of each d column------
-------------
        for(i=1; i<=4; i++)
        {
            if(a_d[1][i]>a_d[2][i])
            {
                max_val[i]=a_d[1][i];
            }
            else
            {
                max_val[i]=a_d[2][i];
            }
        }
//--------------determine the minimum of each d column-------
------------
        for(i=1; i<=4; i++)
        {
            if(a_d[1][i]<a_d[2][i])
            {
                min_val[i]=a_d[1][i];
            }
            else
            {
                min_val[i]=a_d[2][i];
            }
        }
//--------------output of all the minimum for each d column---
----------------
cout<<"\n\noutput of all the minimum for each d column\n";
        for(i=1; i<=4; i++)
        {
        cout<<"d"<<i<<"="<<min_val[i]<<",  ";
        }
keep_chm[1]="d1";
keep_chm[2]="d2";
keep_chm[3]="d3";
keep_chm[4]="d4";
```

```
//--------------output of all the maximum for each d column---
----------------
cout<<"\n\noutput of all the maximum for each d
column\n";
            for(i=1; i<=4; i++)
            {
            cout<<"d"<<i<<"="<<max_val[i]<<",  ";
            }
keep_chx[1]="d1";
keep_chx[2]="d2";
keep_chx[3]="d3";
keep_chx[4]="d4";
//--------------determine the maxmin of risk function table----
---------------
            for(i=1; i<=num_d; i++)
            {
                        for(j=i+1; j<=num_d; j++)
                                    {
            if(min_val[i]<min_val[j])
                                                {

tmp=min_val[i];

tmp_ch=keep_chm[i];

min_val[i]=min_val[j];

keep_chm[i]=keep_chm[j];

min_val[j]=tmp;

keep_chm[j]=tmp_ch;
                                                }
                                    }
            }
//--------------determine the minmax of risk function table----
---------------
            for(i=1; i<=num_d; i++)
            {
                        for(j=i+1; j<=num_d; j++)
                                    {

            if(max_val[i]>max_val[j])
                                                {

tmp=max_val[i];

tmp_ch=keep_chx[i];

max_val[i]=max_val[j];

keep_chx[i]=keep_chx[j];

max_val[j]=tmp;

keep_chx[j]=tmp_ch;
                                                }
                                    }
            }
            k=1;
//----------determined if the output appears more than one----
-----------------------
for(j=1; j<=num_d; j++)
{
            if(min_val[j]==min_val[j+1])
            {
                        k=k+1;
            }
            else
            {
             j=num_d;
            }
}
//----------output of maxmin---------------------------
cout<<"\n\noutput of the maxmin\nmaxmin : ";
            for(i=1; i<=k; i++)
                        {

            cout<<keep_chm[i]<<"="<<min_val[i]<<", ";
                        }
k=1;
//----------determined if the output appears more than one----
-----------------------
for(j=1; j<=num_d; j++)
{
            if(max_val[j]==max_val[j+1])
            {
                        k=k+1;
            }
            else
            {
             j=num_d;
            }
}
 //----------output of minmax---------------------------
cout<<"\n\noutput of the minmax\nminmax : ";
            for(i=1; i<=k; i++)
                        {

            cout<<keep_chx[i]<<"="<<max_val[i]<<", ";
                        }
k=4;
//----------determined if the output appears more than one----
-----------------------
for(j=4; j>=1; j--)
{
            if(min_val[j]==min_val[j-1])
            {
                        k=k-1;
            }
            else
            {
             j=1;
            }
}
//----------output of minmin---------------------------
cout<<"\n\noutput of the minmin\nminmin : ";
            for(i=4; i>=k; i--)
                        {

            cout<<keep_chm[i]<<"="<<min_val[i]<<", ";
                        }
k=4;
//----------determined if the output appears more than one----
-----------------------
for(j=4; j>=1; j--)
```

```
{
            if(max_val[j]==max_val[j-1])
            {
                        k=k-1;
            }
            else
            {
             j=1;
            }
}
//----------output of maxmax---------------------------
cout<<"\n\noutput of the maxmax\nmaxmax : ";
            for(i=4; i>=k; i--)
                        {

            cout<<keep_chx[i]<<"="<<max_val[i]<<", ";
      }
}
```

III. CONCLUSION

This program make it easier to obtain Loss table, minimax action, maxmini action , minimini action, maxmax action and Risk function.

IV. REFERENCES

[1]. Berry, D. A. (1996). Statistics: A Bayesian Perspective. Duxbury. ISBN 0-534-23476-3

[2]. Carlin, B. P. and Louis, T. A. (2008). Bayesian Methods for Data Analysis, Third Edition. Boca Raton, FL: Chapman and Hall/CRC. ISBN 1-58488-697-8.

[3]. Jaynes E. T. (2003) Probability Theory: The Logic of Science, CUP. ISBN 978-0-521-59271-0 (Link to Fragmentary Edition of March 1996).

[4]. Stone, J.V (2013), "Bayes' Rule: A Tutorial Introduction to Bayesian Analysis", Download first chapter here, Sebtel Press, England.

[5]. Bolstad, W. M. (2007) Introduction to Bayesian Statistics: Second Edition, John Wiley ISBN 0-471-27020-2

[6]. Winkler, R. L (2003). Introduction to Bayesian Inference and Decision (2nd ed.). Probabilistic. ISBN 0-9647938-4-9. Updated classic textbook. Bayesian theory clearly presented.