

Convex Hull Test for Ordinal Categorical Data Using the SAS[®] System

Dayong Li, Ph.D., Medex Clinical Trial Services, Inc., Ardmore, PA
 Huanhong Xia, Villanova University, St. Davids, PA

Abstract

This paper reviews the newly developed statistical procedure Convex Hull Test (CHT) for categorical data analysis, and provides an algorithm and SAS codes to implement the statistical procedure.

Introduction

Analysis of ordinal categorical data arises quite frequently in clinical trials and other studies. A typical example in clinical trials is to determine whether one treatment is superior to another based on ordered categorical responses. In this case and many other similar ones, stochastic order is a natural hypothesis. Among all available statistical tests, the newly developed Convex Hull Test (CHT) (Berger (1997)) is unbiased and admissible, and has substantially higher overall power than linear rank tests. It is also the best test for the conditional problems in some cases. Since CHT is a relatively new method, it is not readily available in the SAS system. This paper fills the gap by presenting a ready-to-use SAS program based only on SAS/BASE software so that users can apply the powerful CHT with ease.

The next section provides a brief introduction to the concept of CHT, illustrated using an example. An algorithm and SAS macros will then be presented. The paper ends with a discussion followed by references.

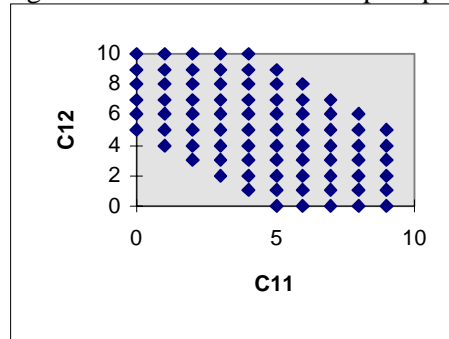
Example of Convex Hull Test

Convex Hull Test is intended to test for independence against stochastic order in a 2-by-j contingency table, where j is an integer. Assume that we are comparing an active treatment with a placebo in a clinical trial and the data are summarized in the following 2x3 contingency table (Table 1) with ordered categories C1, C2 and C3 such as “Worse”, “No change” and “Better”. We want to know if the active treatment has any effect.

Table 1. A 2x3 contingency with ordered categories C1, C2, C3

	C1	C2	C3	Row Total
Placebo	7	6	1	14
Treatment	2	4	8	14
Col. Total	9	10	9	28

Figure 1. The Conditional Sample Space



Let π_{ij} , $i = 1, 2$, $j = 1, 2, 3$, be the probability that a patient given treatment i results in outcome j and denote the probability vectors $(\pi_{11}, \pi_{12}, \pi_{13})$ and $(\pi_{21}, \pi_{22}, \pi_{23})$ as π_1 and π_2 , respectively. The statistical test would be (Berger (1997)):

$$H_0: \pi_1 = \pi_2 \text{ vs. } H_a: \sum_{j=1}^k \pi_{1j} \geq \sum_{j=1}^k \pi_{2j}, k = 1, 2, 3,$$

with at least one strict inequality in H_a . Let C_{ij} , $i = 1, 2$, $j = 1, 2, 3$, be the cell count of treatment i and output j . Note from Table 1 that, conditional on the margins, all cell counts would be fixed after fixing C_{11} and C_{12} . Therefore, the conditional sample space can be presented by all possible points (C_{11}, C_{12}) . Figure 1 is the plot of the sample space for our example. CHT is then realized through a process of peeling off convex points (extreme points) iteratively to form the test's critical region R . For example, we peel off points P1, P2 and P3 in the first round, then (3,10), (5, 9), (8,6), (9,4), (9,1) and (8,0) in the second round, and so on. Here P1 = (4,10), P2 = (9,5) and P3 = (9,0). The peeling process continues until a certain α level, say 0.05, is reached. For the p-value, the process should be stopped as soon as the given table is in R . The construction of whole R is not necessary if only the p-value is desired.

Obviously if the sample size is large, doing CHT by hand would be a tremendous work. Thus computer assistance becomes necessary. The following algorithm simulates the peeling process and provides a feasible solution to the problem.

Algorithm for P-value Calculation

Note that the peeling process is iterative in nature and can always start from the rightmost convex points, which are P1, P2, and P3 in our example (see Figure 1). To let computer identify those convex points, we first select the points with largest C_{11} -value in each C_{12} -group. Then from those already selected data, we choose to work on the points with extreme C_{12} -value(s) in each C_{11} -group, and term these points "candidates". If connecting all adjacent points of these candidates, we would see line section(s). Then the convex points to be in the critical region are the end-points of all these line section(s). Based on that observation, the general algorithm is as follows.

Step 1. Determine the sample space S , conditional on the table margins.

Step 2. Let $W = S - R'$, where R' is formed exactly like the critical region R . Identify the "candidates" in W .

Step 3. Detect if three or more adjacent points from Step 2 form a straight line and identify the end points of each line section.

Step 4. Delete any "concave" end points; Append the rest of the end points to R' .

Step 5. Detect whether the given table is already in R' . If not, go to Step 2. Otherwise, add up all null probabilities associated with all tables in R' to get the p-value.

SAS Macros for Implementing CHT

The critical point to implement the above algorithm in SAS is how to do the Step 3 and 4. In other words, we must find means to let SAS know who are in a straight line and who are then the right points to pick. Amazingly, these can be accomplished simply by using the ratio of the difference of the two variates C_{12} and C_{11} . The sorting of the dataset also plays an important role. Let's look at the first round with candidates (4,10), (5,9), (6,8), (7,7), (8,6), (9,5) and (9,0), sorted by C_{12} in descending order. Calculate the differences of the two variates of every point to that of its previous one, and the ratio corresponding to each point is: *missing*, -1, -1, -1, -1, -1, and *missing* (defined when denominator equals zero). Now looking at the ratios from right to left, the rule is to pick the point where the ratio changes, plus the first one we first look at. Then we have (9,5), (4,10) plus (9,0). That finishes Step 3. Interestingly, Step 4 can be done similarly as Step 3. Assume Step 3 produced (4,10), (6,8), (7,6), (9,5) and (8,1). Again the data are sorted by C_{12} in descending order. Note that the ratios

should constitute an increasing sequence if every point was “convex”. The actual ratios here are: *missing*, -1, -2, -0.5, 4, which is not an increasing sequence because of -2, so (7,6), the point associated with -2, is “concave” and should be kicked out. Having gone through Step 3 and 4, the actual coding in SAS can be carried out smoothly.

Two SAS macros are used. One is SAMPLE, shown in Figure 2; the other is PVALUE shown in Figure 3. The macro SAMPLE takes the six cell counts in a 2x3 table as input, produces the whole sample space and calculates null probability for each table in the sample space, all conditional on the fixed margins. The PVALUE macro utilizes the algorithm to produce the p-value. Using the two macros, the main SAS program only needs a couple of lines as follows:

```
/*including macro SAMPLE*/
%INCLUDE 'sample.sas';
/*including macro PVALUE*/
%INCLUDE 'pvalue.sas';
OPTIONS LS=120 PS=59;
PROC DATASETS LIBRARY=WORK;
  DELETE tables cr tmp;
RUN;
%pvalue(count11=7,count12=6,count13=1,
count21=2,count22=4,count23=8);
```

A sample output of this ready-to-use program is presented as Figure 4 at the end of this paper. The test’s p-value for our example is 0.0092, so we reject the null hypothesis and recognize the effect of the treatment over placebo.

Discussion

Only the p-value calculation has been presented in detail. Construction of the critical region can be done in a very similar way, and thus the power of CHT can be

computed. This paper focuses on the so-called standard convex hull test. There are some variations of the test resulting from posing limitations on the peeling process. For example, we could start to peel only from point P2, or only from P1, P2 or from P2, P3. Fortunately, the algorithm remains valid for all these tests. The trick is to split the sample space. Another advantage of the proposed procedure is that it only utilizes the SAS/BASE, as mentioned before.

Acknowledgments

We would like to thank Dr. Vance Berger for his helpful remarks on the Convex Hull Test.

References

- Berger, V.W. (1997), Convex Hull Test for Ordered Trichotomous Data. New Jersey, Annual Conference of International Chinese Statistical Association.
- SAS Institute Inc., (1990), SAS Guide to Macro Processing, Version 6, Second Edition.

Contacting the author

Dayong Li, Ph.D.
 Medex Clinical Trial Services, Inc.
 10 E. Athens Avenue
 Ardmore, PA 19003, USA
 Phone: (610)896-2340 Ext. 258
 Fax: (610)896-2750
 E-mail: dayong@juno.com

Figure 2. Macro SAMPLE, which takes cell counts in a 2X3 table and produces the sample space and the null probabilities.

```
%MACRO sample(c11=, c12=, c13=,
              c21=, c22=, c23= );
%LET n1= %EVAL(&c11.+&c12.+ &c13);
```

```

/*Calculate margins*/
%LET n2 =%EVAL(&c21.+&c22.+&c23.);
%LET t1 = %EVAL(&c11. + &c21.);
%LET t2 = %EVAL(&c12. + &c22.);
%LET t3 = %EVAL(&c13. + &c23.);

DATA tables;
  n = &n1. + &n2.; *Total # of subjects;
  min_n1t1 = MIN(&n1., &t1.);
  min_n1t2 = MIN(&n1., &t2.);
  min_sum = MIN(&n1,&t1.+&t2.);
  DO i=0 TO min_n1t1;
    x11 = i;
    DO j=0 TO min_n1t2;
      x12 = j;
      key = PUT(x11,$4.)||PUT(x12,$4.);
              /*To identify each table.*/
/*Compute null probabilities below*/
      nullprob =
        GAMMA(&t1+1)/GAMMA(x11+1)
        *GAMMA(&t2+1)/GAMMA(x12+1)
        *GAMMA(&t3+1)/GAMMA(&n1-
          x11-x12+1)*GAMMA(&n1+1)
        /GAMMA(&t1- x11+1)*
          GAMMA(&n2+1)/GAMMA(&t2-
x12+1)
          /GAMMA(&t3-&n1+x11+x12+1)
          /GAMMA(n+1);
      IF %EVAL(&n1.-&t3.) =< x11+x12
        =< min_sum THEN OUTPUT;
    END;
  END; RUN;
%MEND sample;

*****/
%MACRO pvalue(count11=, count12=,
              count13=, count21=,
              count22=, count23= );

/*flag when the given table inreject region*/
%LET reject = 0;

%sample(c11=&count11,c12=&count12,
        c13=&count13,c21=&count21,
        c22=&count22,c23=&count23);

%DO %WHILE (&reject. = 0);
  /*Refer to Figure 1, first to pick the*/
  /*rightmost ends. */
  PROC SORT DATA = tables
    (KEEP = x11 x12 key
nullprob);
    BY x12 x11;
  RUN
  DATA tmp; SET tables;
    BY x12 x11;
    IF last.x12;

  /*Then points w. largest/smallest X11*/
  PROC SORT;
    BY x11 x12;
  RUN;
  DATA tmp; SET tmp;
    BY x11;
    IF first.x11 OR last.x11;
  PROC SORT; BY key; RUN;

  /*Largest and smallest x12 frm the */
  /*remaining table for each x11. */
  PROC SORT DATA = tables;
    BY x11 x12;
  RUN;
  DATA lrg_sml;
    SET tables (KEEP=x11 x12 key);
    BY x11;
    IF first.x11 OR last.x11;
  PROC SORT; BY key; RUN;

/*Possible points to be peeled off*/
  DATA tmp;
    MERGE tmp (IN=a)
              lrg_sml (IN=b KEEP=key);

```

Figure 3. Macro PVALUE outputs the p-value for CHT

```

*****/
/*NAME: PVALUE */
/*DESC : used to compute p-value for a */
/*given table in Convex Hull test. Since p-*/
/*value is the smallest alpha level under */
/*the null, the program peels off extreme */
/*values to form a rejection region. Repeat*/
/*this peeling process until thegiven table */
/* is in the region. In the last peelling */
/*process, then add all null probalities in */
/* the region to get p-value. */

```

```

        BY key;
        IF a AND b;
PROC SORT; BY DESCENDING x12;
RUN;
DATA tmp; SET tmp;
  x11_dif = DIF(x11);
  x12_dif = DIF(x12);
  IF x12_dif ne 0 THEN
      ratio =
x11_dif/x12_dif;
  ind = 1;  /**Dummy**/
PROC SORT; BY x12; RUN;

DATA tmp;
  SET tmp;
  RETAIN test;
  IF _N_ = 1 THEN
      DO;
          test = ratio; OUTPUT;
      END;
  ELSE DO;
      IF ratio NE test THEN
          DO;
              test = ratio; OUTPUT;
          END;
      END;
  DROP test; RUN;

/*Get rid of concave points. Note that ratio
should be decreases with x12, so any jump
points are bad and will be delete here.*/
DATA tmp;
  SET tmp;
  RETAIN test;
  IF _N_ = 1 THEN test = ratio;
  IF ratio > test THEN DELETE;
  ELSE test = ratio;
PROC SORT; BY key; RUN;

/*Append data to the critical region R*/
PROC APPEND BASE = cr
  DATA = tmp (KEEP=x11 x12
              nullprob key);
  RUN;

/*To see if the given table already in R.*/
DATA cr;
  SET cr;
  IF &count11 = x11 AND &count12 = x12

```

```

  THEN CALL SYMPUT('reject',1);
  RUN;
  PROC SORT DATA=tables; BY key;
  RUN;
  DATA tables;
      MERGE tables
          tmp (KEEP=key ind);
  BY key;
  IF ind ne 1; DROP ind; RUN;
%END;  /*End of DO WHILE clause*/

DATA p1; SET cr END=last;
  IF _n_=1 THEN pvalu = 0;
  pvalu + nullprob;
  IF last THEN OUTPUT;

/*for printing, show the under study table */
DATA freq;
  treat = "Treat_1"; result = " C1";
  count = &count11;
OUTPUT;
  treat = "Treat_1"; result = " C2";
  count = &count12;
OUTPUT;
  treat = "Treat_1"; result = " C3";
  count = &count13;
OUTPUT;
  treat = "Treat_2"; result = " C1";
  count = &count21;
OUTPUT;
  treat = "Treat_2"; result = " C2";
  count = &count22;
OUTPUT;
  treat = "Treat_2"; result = " C3";
  count = &count23;
OUTPUT;

TITLE "CONVEX HULL TEST FOR
ORDINAL DATA";
PROC FREQ DATA = freqq;
  TABLES treat*result/NOCOL NOROW
          NOPERCENT;
  WEIGHT count;
  RUN;

TITLE3 "P-VALUES FOR TABLE
(&count11, &count12, &count13;
&count21, &count22, &count23)";
PROC PRINT DATA = p1 SPLIT = "/"

```

```

                                NOOBS
LABEL;
VAR pvalu;
LABEL pvalu =
    'P-VALUE/(Convex Hull Test)';
RUN;

%MEND pvalue;

/*----- End of MACRO pvalue. -----*/

```

Figure 4. Sample Output

Convex Hull Test For Ordinal Data

TABLE OF TREAT BY RESULT

TREAT	RESULT			
Frequency	C1	C2	C3	Total
Placebo	7	6	1	14
Treat	2	4	8	14
Total	9	10	9	28

Convex Hull Test For Ordinal Data

P-Value
(Convex Hull Test)

.0092377