Computing Min and Max Scorings for Two-Sample Ordinal Data

Lyn R. Whitaker, Naval Postgraduate School, Monterey, CA
Michael D. Whitaker, CTB/McGraw-Hill, Monterey, CA

ABSTRACT

Ordinal response variables often occur in practice. For example, in clinical trials a subject's response to a drug regime might be categorized as negative, none, fair, or good. There are several common approaches to analyzing two-sample ordinal response data. These procedures applied to the same data can lead to contradictory conclusions. In an attempt to reconcile contradictory results and provide guidance to the practitioner, Kimledorf, Sampson and Whitaker (1992) propose an alternative approach. They find the scores which when assigned to the levels of the ordinal response variable maximize a two-sample test statistic and the scores that minimize that same statistic. Since many of the two-sample statistics are related by monotonic transformations, these extreme scores can in fact be used to find extreme test statistics for several different two-sample tests.

Let $x_1 \leq x_2 \leq \ldots \leq x_k (x_j \neq x_k)$ be the nondecreasing scores assigned to the levels of an ordinal response variable. The KSW procedure encompasses several of the common methods. The Wilcoxon-Mann-Whitney statistic is a special case of the two-sample statistic with marginal midrank scores assigned to the $x_1, \ldots, x_k$ (e.g., Conover and Iman (1981)). The Cochran-Mantel-Haenszel (CMH) statistic is usually calculated using uniform or equal spacing scores for the $x_1, \ldots, x_k$, marginal midrank scores (ridits), or modified ridit scores. The FREQ procedure allows the choice of these scores for calculating the CMH statistic as well as arbitrary user-provided scores. In addition, both the signed CMH statistic and the two-sample t-statistic are increasing functions of Pearson's correlation coefficient $\rho(x_1, \ldots,x_k)$ between the scores assigned to the ordinal variable and the binary variable indicating whether the response is from Treatment 1 or not.

Thus, by finding the scores $s_1, \ldots, s_k$ which maximize $\rho(x_1, \ldots,x_k)$ and the scores $t_1, \ldots, t_k$ which minimize $\rho(x_1, \ldots,x_k)$ among $x_1 \leq x_2 \leq \ldots \leq x_k$ where $x_j \neq x_k$, we have also found the maximum and minimum of the two-sample t-statistic and the CMH statistic. If both of the extreme values of the statistic lie in the rejection region then it is clear that no matter how the levels of the ordinal response are scored, the test statistic will be significant. When both of the extreme values of the test statistic fail to lie in the rejection region then the result is also clear, no matter what scores are assigned to the ordinal variable, the test statistics will always fail to reject the null hypotheses. In the third case, when the scores straddle a critical value, the conclusion becomes more difficult because some non-decreasing scores assigned to the data will result in rejecting the null hypothesis and yet another assignment of scores will result in acceptance of the null hypothesis.

In the next chapter we outline the KSW procedure for finding the minimum and maximum scores and present a SAS macro used to implement this procedure. In Chapter 3 we give a numerical example and in Chapter 4 we provide a conclusion.
2. THE KSW PROCEDURE AND ITS IMPLEMENTATION

The SAS code is a single macro. This macro needs only base SAS software to run and is implemented within a DATA step. The macro uses data in contingency table form, and does all the computations needed to report the minimum and maximum scores and their corresponding t-statistics, CMH statistics, and Pearson’s correlations. The complete code is available from the authors.

The two-sample data with scores \(x_1 \leq x_2 \leq \ldots \leq x_k\) where \(x_1 \neq x_k\) assigned to the levels of the ordinal response variable can be represented as:

<table>
<thead>
<tr>
<th>TREATMENT</th>
<th>(x_1)</th>
<th>(x_2)</th>
<th>(\ldots)</th>
<th>(x_k)</th>
<th>TOTAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>(m_1)</td>
<td>(m_2)</td>
<td>(\ldots)</td>
<td>(m_k)</td>
<td>(m)</td>
</tr>
<tr>
<td>1</td>
<td>(n_1)</td>
<td>(n_2)</td>
<td>(\ldots)</td>
<td>(n_k)</td>
<td>(n)</td>
</tr>
<tr>
<td>TOTAL</td>
<td>(m_1+n_1)</td>
<td>(m_2+n_2)</td>
<td>(\ldots)</td>
<td>(m_k+n_k)</td>
<td>(N)</td>
</tr>
</tbody>
</table>

Because correlation is scale and location invariant we can, without loss of generality and for ease of use, optimize \(p(x_1, \ldots, x_k)\) over scores \(x_1=0, x_2, \ldots, x_k=1\). The notion of stochastic ordering plays an important role in the computations. The empirical distribution of Treatment 1 is said to be stochastically greater than that of Treatment 0 if

\[
(n_j + \ldots + n_k)/n \geq (m_j + \ldots + m_k)/m
\]

(2.1)

for \(j = 2, \ldots, k\). If the inequality (2.1) is reversed then Treatment 0 is said to be stochastically greater than Treatment 1. If neither hold, then the empirical distributions from the two treatments are stochastically incomparable. For simplicity, we compute the scores \(s_1, \ldots, s_k\) that maximize and the scores \(t_1, \ldots, t_k\) that minimize in three different cases:

Case 1, Treatment 0 data is stochastically greater than Treatment 1 data,
Case 2, Treatment 1 data is stochastically greater than Treatment 0 data,
Case 3, Treatment 0 and Treatment 1 data are stochastically incomparable.

Thus, the first step in computation is to decide in which of the three cases the data fall.

If the data fall into case 1, we find the maximum scores, \(s_1, \ldots, s_k\), by first finding the isotonic regression \(y_1, \ldots, y_k\) of \(n_{ij}(m_{ij}+n_{ij})\) with weights \((m_{ij}+n_{ij})\). There are several algorithms for computing the isotonic regression. In the SAS macro, we use the Pool Adjacent Violators Algorithm (PAVA) (see Robertson, Dykstra and Wright (1988)). The PAVA code is given in the Appendix. The scores \(s_1, \ldots, s_k\) are computed by re-scaling the isotonic regression as \(s_i = (y_i - y)/y(y_k - y_1)\). The minimum scores \(t_i\), \(t_k\) are found by computing \(p(x_1, \ldots, x_k)\) for the k-2 scores of the form \(0=x_1, =x_2\) and \(1=x_{j+1} = \ldots =x_k\) for \(j=2, \ldots, k-1\) and finding the one that gives the smallest \(p(x_1, \ldots, x_k)\).

If the data fall into case 2 then finding the maximum scores \(s_1, \ldots, s_k\) is similar to finding the minimum scores in case 1, i.e. the scores that maximize \(p(x_1, \ldots, x_k)\) among scores of the form \(0=x_1, =x_2, =x_k\) and \(1=x_{j+1} = \ldots =x_k\) for \(j=2, \ldots, k-1\). The minimum score \(t_1, \ldots, t_k\) are found as are the maximum score in case 1. Compute the isotonic regression \(y_1, \ldots, y_k\) of \(n_{ij}/(m_{ij}+n_{ij})\) with weights \((m_{ij}+n_{ij})\) and then re-scale to get \(t_i = (y_i - y)/(y_k - y_1)\) for \(i = 1, \ldots, k\).

For case 3, the scores \(s_1, \ldots, s_k\) are computed as in case 1 and the scores \(t_1, \ldots, t_k\) are computed as in case 2. The macro KSW, implementing this procedure is:

```
%macro ksw(\n lev, \n treat0=treat1, \n minscore=min_scr, \n maxscore=max_scr, \n min_r=min_r, \n max_r=max_r, \n min_t=min_t, \n max_t=max_t, \n cmh=min_cmh, \n cmh=min_cmh, \n macro:macro, \n PAVA, \n stoc_ord, \n Cov, \n Required: \n Proc: \n Comments: \n); \n %define the work arrays; \n array _x[\n lev, \n treat0=treat1, \n min_scr, \n max_scr, \n min_r, \n max_r, \n min_t, \n max_t, \n cmh, \n cmh, \n macro, \n PAVA, \n stoc_ord, \n Cov]; \n```

1432
%*; select(_case_); when(1) do;
%*; create the y from the empirical distrib; %*;

do _ksw_j_ = 1 to dim(y0); _w(_ksw_j_) = y0(_ksw_j_)/y0(1); end;

%*; Find the isotonic regression; %*;
%psv(max_els=an_lev.array=y0.weights=w_);
%*; Re-scale to include 0 and 1; %*;

do _ksw_j_ = 1 to dim(y0); _t0(_ksw_j_) = (y0(_ksw_j_) - y0(1))/(y0(1)-y0(1)); end;

%*; Compute the correlation, the t, and CMH for those scores; %*;
%cor(pop0=treat0, pop1=treat1, scores=t0, r=min_r, t=min_t, cmh=min_cmh);
%*; copy values into the output variables; %*;

do _ksw_k_ = 1 to dim(min_score);
%minscore(_ksw_k_) = _t0(_ksw_k_); end;

%*; This finishes the minimum score. % Now, construct scores of the form % 0x1(1),...x(j) and 1=x(j+1),...,x(k) for % j=2,...,k-1; % then pick the one that gives the minimum % correlation; %
% construct a score; %

do _ksw_j_ = dim(treat0) to 2 by -1;
do _ksw_k_ = _ksw_j_ to dim(treat1); _t1(_ksw_k_) = 1; end;

%*; compute the correlation, t and CMH; %
%cor(pop0=treat0, pop1=treat1, scores=t1, r=min_r, t=min_t, cmh=min_cmh);
%*; copy the score and statistics into an % array for later interrogation; %

_r1(_ksw_j_+1) = _x_; _stt0(_ksw_j_+1) = stud_t_; _cmh(_ksw_j_+1) = cmh_;
do _ksw_k_ = 1 to dim(_t0_);
_t1(_ksw_j_+1, _ksw_k_) = _t1(_ksw_k_); end;
%*; find score giving the max correlation; %

_max_r_ = -1;
do _ksw_k_ = 1 to dim(_r1_);
if (_max_r_ <= _r1(_ksw_k_)) then do;
_max_r_ = _r1(_ksw_k_); _in_max_ = _ksw_k_; end;
%*; copy values to the output variables; %

do _ksw_k_ = 1 to dim(_maxscore_);
%maxscore(_ksw_k_) = _t1(_in_max_, _ksw_k_); end;

%*; Create the yi from the empirical distrib; %

do _ksw_j_ = 1 to dim(treat0);
_x0(_ksw_j_) = (treat0(_ksw_j_)*
(stt0(_ksw_j_))+(treat1(_ksw_j_))
(sttt1(_ksw_j_))+(treat0(_ksw_j_))
(_t0(_ksw_j_))+(treat1(_ksw_j_))
(sttt1(_ksw_j_)));

end;
end;

%*; Find the isotonic regression; %
%
NOTE: The SAS System for Microsoft Windows, Release 6.10
Limited Production
1 options sasauto=c:sugi6\sasuser;
2 data agresti;
3 infile cards;
4 array treat0(*) a1-a4;
5 array treat1(*) b1-b4;
6 array minscr(4);
7 array maxscr(4);
8 input a1-a4;
9 input b1-b4;
10 %ksw(treat0=treat1, treat1=treat1, nlev=4, minlev=1, maxlev=4,
11 minmax=minmax, minscore=minscr);
12 output maxscr(*)=maxscr;
13 run;

3. EXAMPLE

We illustrate this procedure with an example using data from Agresti(1984), where two treatments are used to try to heal ulcer craters.

<table>
<thead>
<tr>
<th>Treatment</th>
<th>Larger</th>
<th>&lt; 2/3 Healed</th>
<th>≥ 2/3 Healed</th>
<th>Healed</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>12</td>
<td>10</td>
<td>4</td>
<td>6</td>
</tr>
<tr>
<td>B</td>
<td>5</td>
<td>8</td>
<td>8</td>
<td>11</td>
</tr>
</tbody>
</table>

The DATA step implementing the KSW procedure for this data is:

```
options sasauto=c:sugi6\sasuser;
data agresti;
  infile cards;
  array treat0(*) a1-a4;
  array treat1(*) b1-b4;
  array minscr(4);
  input a1 a4;
  input b1 b4;
  %ksw(treat0=treat1, treat1=treat1, nlev=4, minlev=1, maxlev=4,
    minmax=minmax, minscore=minscr);
run;
```

The log for this example is:

```
NOTE: The data set WORK.AGRESTII has 1 observations and 22 variables.
```

Note that there are 22 variables in this example. Eight are for the frequencies, 8 are the extreme scores, 2 are t-statistics, 2 are CMH statistics, and 2 are Pearson's correlations.

The empirical distribution of ulcer crater size for Treatment A is stochastically less than that for Treatment B. Thus, the minimum scores are found by searching through the scores of 0's and 1's and the maximum scores are found using the PAVA. The resulting output gives the minimum score $t_0=0$, and $t_1=1$ with minimum $t$ of 1.42 and the maximum score of $s_0=0$, $s_1=1$ with a corresponding maximum $t$ of 2.508. There are no scores which will accept the alternative that Treatment A is better than Treatment B. It is clear that there are some scores which lead to rejection of the null hypothesis that the two treatments are the same and that there are some scores that fail to reject the null hypothesis in favor of a difference in the two treatments (or that Treatment B is better than Treatment A). This straddling situation requires the practitioner to re-evaluate what differences in the treatments are of practical significance. Upon closer inspection of the minimum and maximum scores, we see that if the practitioner is interested in drugs that show any type of improvement regardless of the degree of improvement then the two treatments are very similar. On the other hand, if the practitioner is really interested in completely or almost completely healing ulcer craters then this data presents evidence that Treatment B is better than Treatment A.

4. CONCLUSION
The KSW procedure gives an approach for analyzing two-sample ordinal data. Most methods either explicitly or tacitly assign scores to the levels of the ordinal variable. For true ordinal variables there is no one underlying score that adequately describes the levels. Thus, practitioners often try different scores or different methods, often with conflicting results. KSW helps reconcile these differences by finding the scores which maximize and the scores which minimize both the CMH and the t-statistic. In this paper, we implement the KSW procedure. To enhance the portability of the KSW macro, the code is written using only basic SAS software.

The KSW statistics should not be thought of as test statistics. They are extreme values over a set of test statistics generated from all possible ordinal scorings (including scorings that pool levels of the ordinal variable). Thus, we have purposely left p-values out of the KSW macro. As seen in the undercrater example, even though there is no distribution theory for the KSW procedure, both the extreme t-statistics and the corresponding scores provide a deeper insight into the data than any one of the usual methods used alone.

The more general problem of finding extreme scores for ordinal response variables in an ANOVA setting is treated in Gautam(1991). Streitberg and Roehmel(1988) give a method for computing bounds for p-values for a class of permutation tests in the two-sample setting. They do not give extreme scores and their algorithm is implemented in TESTIMATE.

5. REFERENCES


6. APPENDIX

```sas
/*macro PAV(max_el, array, weight);*/
%macro PAV(max_el, array, weight);%global index; %if &quote(index)= %then %let index = 1; %else %let index = &index+1;%let pooled = pool&index.; %let parray = par&index.; %let pqwghts = pqwght&index.; array &pooled (max_el) temporary; array &parray (max_el) temporary; array &pqwghts (max_el) temporary; if &dim(array)=1 then go to &pavindex; do _pav_j = 1 to &dim(array); &pav_j = 0; &pqwghts(_pav_j) = 0; end; &pav_i = 2 to &dim(array); if (&parray(_pav_i)> &array(_pav_i)) then do; &pav_i = &pav_i; do &pav_j = 1 to &dim(array); &pav_j = &pav_j - 1; &pav_j > &pav_j then do; &pav_j = &pav_j - 1; do while((&parray(_pav_j)> &pav_j) & (pav_i < 2));
```

1435
_plval_ = _plval_; _plwght_ = _plwght_; do until(_spooled(_pav_j_)); _pav_j_ = _pav_j_ - 1; end; /* do until */ _plwght_ = _plwght_ + awghts(_pav_j_); _plwght_ = (_plval_ * _plwght_) / _plwght_; _plval_ = (_sparray(_pav_j_) * awghts(_pav_j_)) + (_plval_ * _plwght_) / _plwght_; _pav_j_ = _pav_j_ - 1; _pav_j_ = _pav_j_ + 1; end; /* if _pav_j_ > 1 */ &sparray(_pav_j_) = _plval_; &awghts(_pav_j_) = _plwght_; end; /* (sparray(_pav_i_) > &array(_pav_i_)) then */

else
  do;
    _pav_j_ = _pav_j_ + 1;
    &sparray(_pav_j_) = &array(_pav_i_);
    &awghts(_pav_j_) = &weights(_pav_i_);
    end;
    /* _pav_j_ = 2 to dim(&array); */ &array(1) = &sparray(1);
  _pav_j_ = 1;
  _pav_j_ = 1;
  do _pav_j_ = 2 to dim(&array);
    if _spooled(_pav_j_) then
      _pav_j_ = _pav_j_ + 1;
      &array(_pav_j_) = &sparray(_pav_j_);
    end;
    &pav_index:
    drop _pav_j_ _pav_i_ _pav_j_ _plval_ _plwght_ _plval_ _plwght_; &end;

SAS is a registered trademark of SAS Institute Inc. in the USA and other countries. © indicates USA registration.

You may contact the Authors at:

Lyn R. Whitaker , Code OR/WH, Naval Postgraduate School, Monterey, Ca. 93943. (408)656-3482. WHITAKER@NPS.NAVY.MIL

Michael D. Whitaker, CTB/McGraw-Hill, 20 Ryan Ranch Rd, Monterey, Ca. 93940. (408)393-7992. MWHITAKE@MCGRAW-HILL.COM