

Calculating the Power Function of the Wilcoxon Rank Sum Test in 2xK Tables

Mark Von Tress

July 29, 1991

Abstract

An IML program is described that calculates the power function of the Wilcoxon Rank Sum test in 2xk tables. The program may be used for sample size requirements in experiments involving 2xk Tables.

1 Introduction

Many clinical trials use ordinal categorical responses as primary efficacy variables. The ordinal response is used to categorize a subjective measurement. Typically the scale is accompanied by a detailed description of each category which helps reduce misclassification error. In parallel group clinical trials, two treatments are given in a blinded fashion, and the investigator, or patient, is asked to rate the response on the ordinal scale.

The Wilcoxon rank sum is a well known choice for determining treatment differences in such clinical trials. The power function for this test statistic must be used to determine the sample sizes for these trials. The problem is slightly more complicated than designing a study based on the t-test. Several questions arise:

- How do we calculate the power function?
- How do we calculate a meaningful sequence of alternative distributions?

This paper will take a simple approach to answering these questions. A SAS IML monte-carlo program is presented for calculating the power function. Some examples will also be presented.

2 Simulation Algorithm

The power calculation counts the number of times a simulated Wilcoxon rank sum statistic falls in the critical region. The algorithm is

1. Calculate the reference and alternative profiles
2. Calculate the cumulative step function for each of the profiles.
- 2b. Set count to zero.
3. Generate a table for a trial using the step function in 2.
4. Calculate a Wilcoxon rank sum statistic from the table in 3
5. Increment count if the test statistic is in the critical region
6. Repeat 3-5 for the desired number of trials.

The algorithm for generating a table uses the step function in 2. The tables are generated by the probability integral transform for discrete random variables: find the range in the step function where a uniform random number falls, and add 1 to that class in the table.

The rank sum statistic is generated from a 2xk table,

Category	1	2	...	k	row totals
reference	n_1	n_2	...	n_k	n
alternative	m_1	m_2	...	m_k	m
column totals	t_1	t_2	...	t_k	$N = n + m$

Also define the Wilcoxon scores as $r_1 = (t_1 + 1)/2$, and $r_i = t_1 + t_2 + \dots + t_{i-1} + (t_i + 1)/2$. Throughout this paper, p_i denotes n_i/n for the reference distribution and m_i/m for the alternative. The Wilcoxon rank sum is given by

$$W = \sum_{i=1}^k r_i m_i \quad (1)$$

The test statistic adjusted for ties is given by

$$Z = \frac{W - m(N + 1)/2}{\sqrt{(mn(N + 1)/12) - ((mn \sum (t_i^3 - t_i))/(12N(N - 1)))}} \quad (2)$$

In large samples, this test statistic has a standard normal distribution. See [2] for more information on the Wilcoxon rank sum test in 2xk tables.

The power function may be approximated by the percent of simulated test statistics falling into the critical region. A 95% confidence interval may be calculated on this percent using normal theory, where T is the number of tables generated:

$$\hat{p} \pm 1.96 \sqrt{\frac{\hat{p}(1 - \hat{p})}{T}} \quad (3)$$

The program in the Appendix implements this algorithm in an IML macro. The parameters in the macro call are the data set name to open, the number of tables to simulate, and the number of elements to generate per row. Increasing the number of tables improves the precision of the estimate of the power function (see equation 3). Increasing the *rowsize* parameter increases the power function for given reference and alternative distributions. Thus, the program may be used for calculating a sample size by searching for the value of *rowsize* that produces the desired power at a fixed level. The critical region may be modified by altering the line containing 1.96. An easy way to check that the program has been entered correctly is to use it with identical profiles. The power should be the type 1 error rate in large samples.

3 Smoothing Options

Smoothing techniques may be used if the step function is based on small sample sizes. Power calculations may be overly optimistic when using small samples. The difference between the alternative and reference distribution may be exaggerated by proportions calculated from small samples. Differences in proportions may be larger than would be expected if larger samples had been available. The smoothing attempts to compensate for this extra variability.

3.1 Ad-Hoc Methods

Ad-hoc methods allow direct specification of the profile distribution. The advantage to these methods is their directness and simplicity. They allow the user to directly specify the alternative distribution. One method is to simply specify the probabilities.

Another method shifts probabilities recursively. This shifts the profile probabilities in a recursive manner from right to left. The profile vector is shifted by $p'_i = p_i + \Delta p_{i+1}$, for $i=k-1, \dots, 1$. Also, $p'_k = p_k$. Then the p'_i are divided by their sum to produce a new probability profile. The value of Δ is chosen so that the difference between the probabilities in the first column differ by a predetermined amount, say 0.2. Probabilities may also be rolled forward using a similar technique.

3.2 Parametric Methods: Beta Distribution

Parametric techniques replace the profile by probabilities from well known densities in the manner of the chi-square goodness of fit test. This is useful when the profile has a well known shape. The Beta distribution is useful because of its flexibility. The categories above may be rescaled onto the interval 0 to 1 by dividing by k . The Beta density function is given by

$$p(x|\alpha, \beta) = \frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha)\Gamma(\beta)} x^{\alpha-1} (1-x)^{\beta-1} \quad (4)$$

where x is in $[0,1]$, $\alpha > 0$, and $\beta > 0$. The categories are rescaled and the probabilities are calculated using the incomplete beta function. Let $x_i = i/k$, and $x_{i+1} = (i+1)/k$ for $i = 0, \dots, k-1$. The profile probability is the beta distribution probability for that bin, b_i , i.e.

$$P[b_i] = \int_{x_i}^{x_{i+1}} p(x|\alpha, \beta) dt \quad (5)$$

In the interest of simplicity, moment estimators may be used for the parameters. The parameter estimates are given by

$$\hat{\alpha} = \frac{\mu^2(1-\mu)}{\sigma^2} - \mu \quad \hat{\beta} = \frac{\hat{\alpha}(1-\mu)}{\mu} \quad (6)$$

where μ and σ^2 are the sample mean and variance. The elementary formulae may be used for the sample moments, i.e.:

$$\mu = \sum_{i=1}^k p_i x_i \quad \sigma^2 = \sum_{i=1}^k p_i (x_i - \mu)^2 \quad (7)$$

The estimators that give a change, Δ , in the mean while retaining a constant variance are given by

$$\hat{\alpha} = \frac{\mu^2(1+\Delta-\mu)}{\sigma^2(1+\Delta)^3} - \frac{\mu}{(1+\Delta)} \quad \hat{\beta} = \frac{\hat{\alpha}(1+\Delta-\mu)}{\mu} \quad (8)$$

Typically Δ ranges between -0.2 to 0.2.

3.3 Bayesian Smoothing

The profile probabilities may be replaced by a weighted average of the observed probabilities, p_i , and the prior probabilities, λ_i . The weighted average is given by

$$\hat{p}_i = \frac{N}{N + \hat{C}} p_i + \frac{\hat{C}}{N + \hat{C}} \lambda_i \quad (9)$$

where

$$\hat{C} = \frac{1 - \sum_{i=1}^k p_i^2}{\sum_{i=1}^k (p_i - \lambda_i)^2} \quad (10)$$

These smoothed probabilities result from assuming a Dirichlet prior and using squared error loss. The value of C comes from using the observed probabilities. Note that if C is large, then more weight is given to the prior distribution. Conversely if N is large, then the observed probabilities are given more weight.

The prior probabilities may be selected a-priori, or estimated from the data using pseudo Bayesian techniques. A simple estimate of the prior may be calculated from the column marginal probabilities as $\hat{\lambda}_i = t_i/N$. See [1] for more information on Bayesian smoothing of small frequencies.

4 Example

The program uses the frequencies below to illustrate some of the ideas in this paper.

Category	1	2	3	4	5	6	7	row totals
reference	0	2	4	7	5	0	0	18
alternative	1	1	2	6	2	3	1	16
column totals	1	3	6	13	7	3	1	34

Appendix A contains the data for these examples. In each case, the cell proportions were multiplied by 1000 to produce counts which are required as input into the program.

4.1 Parametric Smoothing

The alternative distribution above was smoothed by fitting a beta distribution to it. The power function was then calculated using the beta distribution that had the same variance, but the mean of the shifted alternative was increased 20% to produce the mean of the smoothed distribution. The following profiles were compared:

Category	1	2	3	4	5	6	7	row totals
smoothed alt.	0.068	0.125	0.156	0.174	0.180	0.171	0.126	1.000
shifted smooth alt.	0.132	0.174	0.181	0.173	0.154	0.121	0.065	1.000

The means of the rescaled categories were 0.5416 and 0.4514 with a common variance of 0.0643. The chi-square goodness of fit test did not reject the description of the alternative distribution by the beta distribution ($p=0.275$). Also note that the observed proportion of 0.375 ($=6/16$) was smoothed to 0.173. Approximately 130 observations per row are required to detect a difference between these distributions with 80% power and a 5% chance of a type 1 error in a two sided hypothesis test.

4.2 Bayesian Smoothing

The next example demonstrates how using small frequencies may produce overly optimistic estimates of the power function. The proportions in the exact table are

Category	1	2	3	4	5	6	7	row totals
ref. probs.	0.000	0.111	0.222	0.389	0.278	0.000	0.000	1.000
alt. probs.	0.062	0.062	0.125	0.375	0.125	0.188	0.062	1.000

At 75 observations per row, the power function is about 0.498. This may be attributed to the differences in the last three columns. The table that results from using the column marginal as a prior is

Category	1	2	3	4	5	6	7	row totals
bayes ref. probs.	0.020	0.095	0.190	0.384	0.228	0.061	0.020	1.000
bayes alt. probs.	0.040	0.080	0.161	0.380	0.181	0.119	0.040	1.000

The smoothing parameters for the reference and alternative distributions are $\hat{C} = 40.93$ and $\hat{C} = 35.59$ respectively. Although this smoothing appears benign, the power is reduced to 0.112 at 75 elements per row. This could have occurred for two reasons. The smoothing shrinks the category means closer together, and the differences in the tails are smaller.

References

- [1] Agresti, Alan (1990) *Categorical Data Analysis*, John Wiley and Sons, pp. 463-466
- [2] Mehta, C.R., Patel, N.R. and Tsiatis, A.A (1984) *Exact Significance Testing to Establish Treatment Equivalence with Ordered Categorical Data*, *Biometrics*, 40, pp 819-826.

A Wilpow.sas

```

/*
  Author:      Mark Von Tress, Ph.D.
  Program:     WILPOW.SAS
  Date:        12/10/90
  Version:     1.0

  Description: This program calculates the power function of
               Wilcoxon rank-sum test for 2xK tables using
               Monte-Carlo techniques

  System requirements: PROC IML of SAS v6.0+

*/

options ls=80;

%macro wilpow( dsname, ntables, rowsizes);
/*
  dsname is data set containing two rows of counts
  ntables is the number of tables to generate
  rowsizes is the number of elements to generate per row.
*/

```

```

proc iml;

start getpstar( nij, pstar ); * get probs from counts, nij, in dsname;
  nidot = nij[,+];
  pstar = nij / ( j(1,ncol(nij),1)@nidot );
  print pstar;
finish;

start getprofile( pstar, ps, altcdf ); * accumulate rows;
  p1 = pstar[1,];
  p2 = pstar[2,];
  ps = cusum( p1 );
  altcdf = cusum( p2 );
finish;

start getn( pdf, n);
  u = ranuni(1243);
  n=1;
  k=ncol(pdf);
  do while( n<=k & u>pdf[1,n]); n=n+1; end;
finish;

start gentable( ps, altcdf, trials, atable); * get a table;
  atable = j(2,ncol(ps),0);
  do i=1 to trials;
    run getn( ps, n);
    atable[1,n] = atable[1,n] + 1;
    run getn( altcdf, n);
    atable[2,n] = atable[2,n] + 1;
  end;
  * print atable;
finish;

start wilcox( atable, z); * calculate a wilcoxon test stat from a table;
  tot = atable[+,];
  n = atable[1,+];
  m = atable[2,+];
  NN= n+m;
  scores= cusum(tot)-0.5#tot+j(1,ncol(tot),0.5);
  tiesum = sum( tot#tot#tot - tot);
  num = atable[1,]*scores - m*(NN+1)/2;
  den = m#n*(NN+1)/12 - tiesum#m#n/(12*NN*(NN-1));
  if den <= 0 then z=0; else z=num/sqrt(den);
  * print tot n m scores num den;
finish;

start main;
  use &dsname;
  read all into nij;

  run getpstar( nij, pstar);          * calculate pstar from counts;
  run getprofile( pstar, ps, altcdf); * accumulate probs;

```

```

trials= &rowsizes;          * number of samples per treatment group;
nsamps= &ntables;           * number of tables to generate;
print ps altcdf;

phat = 0;                   * count number of rejections;
do i=1 to nsamps;
  run gentable( ps, altcdf, trials, atable); * generate a table;
  run wilcox( atable, z); * calculate a stat;
  if abs(z) > 1.96 then phat = phat + 1; * test for rejection;
end;
phat = phat / nsamps; * approximate power;
lci95 = phat-1.96*sqrt( phat*(1-phat)/nsamps ); * lower 95 pcnt ci;
uci95 = phat+1.96*sqrt( phat*(1-phat)/nsamps ); * upper 95 pcnt ci;
print trials lci95 phat uci95;
finish;

run main;
%mend wilpow;

data one;
  input c0-c6; * input two rows of counts;
cards;
068 125 156 174 180 171 126
132 174 181 173 154 121 065
%wilpow(one, ntables=500, rowsizes=130);
title "Beta Smoothing and 20% shift in mean from Alt to Ref";

data one;
  input c0-c6; * input two rows of counts;
cards;
000 111 222 389 278 000 000
062 062 125 375 125 188 062
%wilpow(one, ntables=500, rowsizes=75);
title "No Smoothing";

data one;
  input c0-c6; * input two rows of counts;
cards;
020 095 190 384 228 061 020
040 080 161 380 181 119 040
%wilpow(one, ntables=500, rowsizes=75);
title "Pseudo bayes Smoothing";

```