Symbolic Computations for Exact Distributions of Rank Statistics

A. Di Bucchianico,

Eindhoven University of Technology
Department of Mathematics

Dortmund, January 13, 2005

Contents of this talk

- General remarks on rank-based methods
- Computer algebra
- Direct evaluations of generating functions
- Double generating function
- Distributions under alternative hypotheses
- Use of symmetries
- Branch-and-bound algorithm

Parametric statistical inference

sample $X_1, \ldots, X_n$

assumption: $X_i \sim N(\mu, \sigma^2)$

goal: knowledge about parameters μ and/or σ^2

null hypothesis: $H_0 : \mu = \mu_0$

alternative hypothesis:$H_1 : \mu \neq \mu_0$

test statistics: under H_0, we have

$$\overline{X} \sim N(\mu_0, \sigma^2) \text{ and } \frac{\overline{X} - \mu_0}{S/\sqrt{n}} \sim t_{n-1}$$

Drawback: one cannot always assume normality in practice.

Nonparametric statistical inference

sample $X_1, \ldots, X_n$

assumption: continuous distribution

median m : $\Pr(X_i \leq m) = 1/2$

null hypothesis: $H_0 : m = m_0$

test statistic: $T = \{\#i : X_i \leq m_0\}$ (sign test)

under H_0, we have $T \sim \text{Bin}(n, 1/2)$

more sophisticated: $W = \sum_{X_i \leq m} \text{rank}(X_i)$ (Wilcoxon signed rank test)

null distribution of W?

Null distribution of signed rank statistic

null hypothesis: $\Pr(X_i \leq m) = 1/2$

$W = \sum_{X_i \leq m} \text{rank}(X_i)$

Example: $m = 0$, ordered observations $-3, 4, 6, -7$, thus $W = 1 + 4 = 5$

Under H_0: $W \overset{d}{=} \sum_{i=1}^{n} iW_i$ with W_i i.i.d. Bin(0,1/2)

$$\text{pgf}(W) = \sum_{k=0}^{\frac{1}{2}n(n+1)} P(W = k)z^k = \prod_{i=1}^{n} \text{pgf}(W_i) = \frac{(1+x)\,(1+x^2)\,\ldots(1+x^n)}{2^n}$$

Can be evaluated directly using computer algebra.

Computer algebra: session in Mathematica

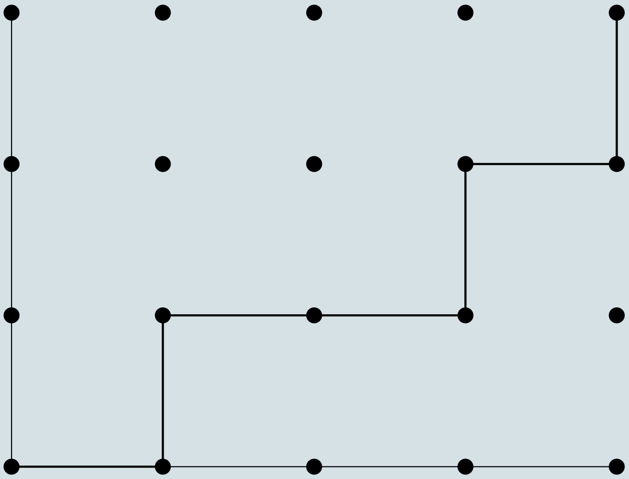
```
In[1]:= Expand[(1+x)*(1+x^2)]
                 2    3
Out[1]= 1 + x + x  + x

In[2]:= Coefficient[Product[1 + x^n,{n, 0, 30}],x,400]

Out[2] = 28964

In[3]:= <<DiscreteMath‘RSolve‘

In[4]:=  Assuming[n>0,SeriesTerm[1/(1-x^2),{x,0,n}]

        1           n
Out[4]= - ( 1 + (-1) )
        2
```

General remarks on rank-based methods

- Practical problems
  - tables (limited, errors, not exact,. . .)
  - limited availability in general statistical software
  - procedures in statistical software often based on asymptotics
  - exact procedures for practical sample sizes require dedicated software (StatXact, SPSS module)
- Mathematical problems
  - in general no closed expression for distribution function
  - evaluation distribution by direct enumeration only feasible for small sample sizes
  - evaluation distribution by recurrences is time-consuming

Another example: Mann-Whitney statistic

Mann-Whitney: $M_{m,n} = \#\{(i,j) \mid (Y_j < X_i)\}$
rank configuration: X Y X X Y X Y $\Rightarrow M_{4,3} = 4$



Note: $M_{4,3}$ equals area beneath path

Combinatorial interpretation of Mann-Whitney statistic

$M_{m,n} = k$

Statistical terminology: $M_{m,n}$ is area beneath Gnedenko path

Combinatorial terminology: $M_{m,n}$ is restricted partition of k with at most m parts, each of which $\leq n$ (Ferrers diagram)

Generating functions for restricted partitions go back to Gauss.

Interpretations in terms of Gnedenko paths or restricted partitions easily yield recurrence relations.

Distribution Mann-Whitney statistic: recurrences

Under $H_0 : F = G$, we have $P(M_{m,n} = k) = \dfrac{f(m,n,k)}{\dbinom{m+n}{n}}$

Recursion 1 (Mann-Whitney 1947):

$$f(m,n,k) = f(m-1,n,k-n) + f(m,n-1,k)$$

Proof: check whether path passes through $(m-1,n)$ or $(m,n-1)$.

Recursion 2 (Brus 1989):

$$f(m,n,k) = \sum_{i=0}^{n} f(m-1,i,k-i)$$

Proof: condition on last right-turn $(m-1,i)$ of path.

More serious recurrences

Recursion 3 (Brus 1989)

$$f(m,n,k) = \sum_{i_1=0}^{n} \sum_{i_2=0}^{i_1} \dots \sum_{i_j=0}^{i_{j-1}} f(m-j, i_j, k-i_1-\dots-i_j)$$

Proof: condition on last j right-turns of path.

Instead of looking at the end of the path we may also look at the other end of the path.

Recursion 4

$$f(m,n,k) = f(m-1,n,k) + f(m,n-1,k-m)$$

Proof: path must go through $(1,0)$ or $(0,1)$.

More serious recurrences: continued

Recursion 5 (Brus 1989)

$$f(m,n,k) = \sum_{i=0}^{n} f(m-1, n-i, k-im)$$

Proof: condition on first right-turn of path; count paths from $(0,i)$ to (m,n).

Recursion 6

$$f(m,n,k) = \sum_{i_j=0}^{n} \sum_{i_{j-1}=0}^{i_j} \ldots \sum_{i_1=0}^{i_2} f(m-j, n-i_j, k-i_1-\ldots-i_{j-1}-i_j(m-j+1))$$

Proof: condition on first j right-turns of path.

Generating function for Mann-Whitney null distribution

$$\sum_{k=0}^{mn} P(M_{m,n} = k)\, x^k = \frac{1}{\binom{m+n}{n}} \, \frac{\prod_{i=n+1}^{m+n}(1 - x^i)}{\prod_{j=1}^{m}(1 - x^j)}$$

Example 1:

exact: $P(M_{5,5} \leq 4) = \frac{1}{21} \approx 0.0476$; normal approximation: 0.0387

computing time: 0.00 sec (gen. function is polynomial of degree 25)

Example 2:

exact: $P(M_{20,20} \leq 138) = \frac{237538006}{4923090315} \approx 0.0482$; normal approximation: 0.0475

computing time: 0.28 sec (gen. function is polynomial of degree 400)

Computations performed on laptop (Pentium: 2 GHz).

Recurrences versus generating functions

m	n	k	Method 1	Method 2	Method 5
5	5	6	0.2	0.2	0.3
5	5	18	0.3	0.3	0.3
5	10	10	0.8	0.9	0.3
5	10	40	1.1	1.0	0.4
10	5	10	0.9	0.8	0.3
10	5	25	4.2	4.0	0.3
10	5	40	1.0	1.1	0.4
10	10	25	28	27	0.7
10	10	50	140	140	0.9
10	10	75	31	31	1.2

Computing time in Mathematica on (old) SunSPARCstation 5 in seconds.

Method 1 and 2 are based on recurrences; method 5 is direct evaluation of generating function.

Rank statistics with closed form generating function

- Wilcoxon signed rank statistic
- Wilcoxon rank sum statistic = Mann-Whitney statistic
- Kendall rank correlation statistic
- Kolmogorov one-sample statistic
- Smirnov two-sample statistic (for many combinations of sample sizes)
- Jonckheere-Terpstra statistic

Methods when direct evaluation of pgf is not possible

- Fourier methods: Pagano and Tritchler, Baglivo
- various shift-algorithms: Streitberg and Röhmel , Edgington
- network algorithms developed: Mehta and co-workers, commercial implementation in StatXact
- recursive computation of generating functions: Hirji and Johnson, Di Bucchianico and Van de Wiel

All these methods may be described as efficient methods to calculate generating functions.

Exact distributions under alternative hypotheses

For control charts, we need distributions under alternative hypotheses. Literature only gives some special cases (Lehmann alternatives).

sample $X_1, \ldots, X_n$ from continuous distribution with symmetric density

$$H_0 : m = 0 \quad \text{and} \quad H_1 : m = \delta.$$

$X_{|j|}$ is jth order statistic of $|X_1|, \ldots, |X_n|$

$\overline{\pi}_j = 1$ if $X_{|j|}$ corresponds to positive observation and -1 otherwise.
Linear signed rank statistic:

$$T_{p,a} = \sum_{j=1}^{p} a(j)\overline{\pi}_j,$$

where $a(j)$ is the jth rank score and $1 \leq p \leq n$.

Recursions for alternative distributions

Problem: under H_1 rank configurations are no longer equiprobable, hence nice generating functions no longer exist.

Idea: embed recursions on probabilities (Klotz (1962) and Arnold (1965)) into recursions for generating function

$$(\pi, 0) = (\pi_1, \ldots, \pi_p, 0), \quad (\pi, 1) = (\pi_1, \ldots, \pi_p, 1).$$

$$A_{p,\pi}(u) = \mathrm{P}(\bar{\pi} = \pi, |X_i| \leq u \text{ for } i = 1, \ldots, p)$$

$$A_{p+1,(\pi,1)}(u) = (p+1) \int_0^u A_{p,\pi}(v) f(v)\, dv$$

$$A_{p+1,(\pi,0)}(u) = (p+1) \int_0^u A_{p,\pi}(v) f(-v)\, dv$$

where $f(w) = f_0(\delta - w)$. (f_0 is density under H_0).

Generating function

Now define generating function.

$$\Pi_p := \{0,1\}^p$$

$$A_{p,\pi}(u) = \mathrm{P}(\bar{\pi} = \pi, |X_i| \leq u \text{ for } i = 1, \ldots, p)$$

$$H_p(u,x) = \sum_{\ell} \sum_{\pi \in \Pi_{p,\ell}} A_{p,\pi}(u) x^{\ell}$$

$$H_0(u,x) = 1,$$

where $\Pi_{p,\ell} = \{\pi \in \Pi_p | T_p(\pi) = \ell\}$.
Note that

$$\sum_{\ell} \mathrm{P}(T_n = \ell) x^{\ell} = \sum_{\ell} \sum_{\pi \in \Pi_{n,\ell}} A_{n,\pi}(\infty) x^{\ell} = H_n(\infty, x).$$

Recursion for generating function

$$H_p(u,x) = p\left(x^{a(p)} \int_0^u H_{p-1}(v,x) f(v)\, dv + \int_0^u H_{p-1}(v,x) f(-v)\, dv \right).$$

Evaluation of recursion:

- symbolically if f has primitive in closed form
- numerically using midpoint algorithm adapted from Milton (1970)

Note on integrals in recursion

$$
\begin{aligned}
H_1(u,x) =& p\left(x^{a(1)} \int_0^u f(v)\,dv + \int_0^u f(-v)\,dv\right) \\
=& p\left(x^{a(1)}\,(F(u) - F(0)) + F(0) - F(-u)\right)
\end{aligned}
$$

$$
H_2(u,x) = 2\left(x^{a(2)} \int_0^u H_1(v,x) f(v)\,dv \int_0^u H_1(v,x) f(-v)\,dv\right).
$$

Some simplification is possible by using

$$
\int_0^u F^j(v)\,f(v)\,dv = F^{j+1}(u) - F^{j+1}(0).
$$

However, we also need

$$
\int_0^u F^j(v)\,f(-v)\,dv.
$$

Nonparametric CUSUM chart

King and Longnecker calculate (using time-consuming numerical integration) the ARL of the following CUSUM chart:

observations $X_i = (X_{i1}, \ldots, X_{in})$ from continuous distribution, target value θ_0

$$S_0 = 0, S_i = \max\{0, S_{i-1} + T_{n,a}(X_i) - \theta_0 - k\}$$

Signal if $S_i > h$.

Out-of-control run length distribution of S under location shift can be obtained, since $(S_0, S_1, \ldots)$ is Markov chain with finite state space (cf. Brook and Evans (1972)).

Transition probabilities can be expressed in terms of the alternative distribution.

Spearman's rank correlation test

Together with Kendall's τ the most popular rank test statistic for testing correlation.

Bivariate data: $(X_1, Y_1), \ldots, (X_n, Y_n)$

R_i : rank of X_i in $X_1, \ldots, X_n$; S_i : rank of Y_i in $Y_1, \ldots, Y_n$.

H_0 : X and Y are not correlated.

Test statistic: $\rho = \sum_{i=1}^{n} (R_i - S_i)^2$.

Tail probabilities: $P(\rho \geq d)$ or $P(\rho \leq d)$.

Goal: fast computation of tail probabilities and critical values

All permutations: too time and memory consuming.

Permutations and Spearman's ρ

$\mathcal{S}_n$ denotes the symmetric group of n elements

null distribution of ρ equivalent to enumeration of statistic on $\mathcal{S}_n \times \mathcal{S}_n$:

$$S_2 : (\sigma, \tau) \longmapsto \sum_{j=1}^{n} (\sigma(j) - \tau(j))^2.$$

equivalence relation on $\mathcal{S}_n \times \mathcal{S}_n$:

$$(\sigma, \tau) \sim (\rho, \varsigma) \longrightarrow \exists \nu \in \mathcal{S}_n : \rho = \sigma \circ \nu \text{ and } \zeta = \tau \circ \nu$$

$$S_2(\sigma \circ \nu, \tau \circ \nu) = S_2(\sigma, \tau)$$

$$S_1 : \sigma \longmapsto \sum_{j=1}^{n} (\sigma(j) - j)^2$$

S_1 is statistically equivalent to $\widetilde{S}_1 := \sum_{j=1}^{n} j\,\sigma(j)$.

Generating function and permanent

$$\mathrm{per}(A) = \sum_{\sigma \in \mathcal{S}_n} \prod_{j=1}^{n} a_{\sigma(j),j}.$$

Olds (1938) and Kendall (1939):

$$\sum_{k=0}^{\infty} \Pr(S_1 = k)\, x^k = \frac{1}{n!}\, \mathrm{per}(P) \text{ and } \sum_{k=0}^{\infty} \Pr(\widetilde{S}_1 = k)\, x^k = \frac{1}{n!}\, \mathrm{per}(\widetilde{P})$$

where

$$P_{ij} = x^{(i-j)^2} \quad \text{and} \quad \widetilde{P}_{ij} = x^{ij}, \quad i,j = 1, \ldots, n.$$

- permanent does not share nice properties of determinant
- Ryser's algorithm not fast enough for permanents with monomial entries of size >10
- exploit symmetries to break down permanents into smaller permanents

Reduction 1: Laplace expansion + symmetries

$$\operatorname{per}\begin{pmatrix} 1 & x & x^4 & x^9 \\ x & 1 & x & x^4 \\ x^4 & x & 1 & x \\ x^9 & x^4 & x & 1 \end{pmatrix} = \operatorname{per}\begin{pmatrix} 1 & x \\ x & 1 \end{pmatrix} \operatorname{per}\begin{pmatrix} 1 & x \\ x & 1 \end{pmatrix} + \operatorname{per}\begin{pmatrix} 1 & x^4 \\ x & x \end{pmatrix} \operatorname{per}\begin{pmatrix} x & x \\ x^4 & 1 \end{pmatrix}$$

$$+ \operatorname{per}\begin{pmatrix} 1 & x^9 \\ x & x^4 \end{pmatrix} \operatorname{per}\begin{pmatrix} x & 1 \\ x^4 & x \end{pmatrix} + \operatorname{per}\begin{pmatrix} x & x^4 \\ 1 & x \end{pmatrix} \operatorname{per}\begin{pmatrix} x^4 & x \\ x^9 & 1 \end{pmatrix}$$

$$+ \operatorname{per}\begin{pmatrix} x & x^9 \\ 1 & x^4 \end{pmatrix} \operatorname{per}\begin{pmatrix} x^4 & 1 \\ x^9 & x \end{pmatrix} + \operatorname{per}\begin{pmatrix} x^4 & x^9 \\ x & x^4 \end{pmatrix} \operatorname{per}\begin{pmatrix} x^4 & x \\ x^9 & x^4 \end{pmatrix}$$

$$\text{symmetry: } \operatorname{per}\begin{pmatrix} 1 & x^9 \\ x & x^4 \end{pmatrix} \operatorname{per}\begin{pmatrix} x & 1 \\ x^4 & x \end{pmatrix} = \operatorname{per}\begin{pmatrix} x & x^4 \\ 1 & x \end{pmatrix} \operatorname{per}\begin{pmatrix} x^4 & x \\ x^9 & 1 \end{pmatrix}$$

Reduction 2: More symmetries

$$P = \begin{pmatrix} 1 & x & x^4 & x^9 & x^{16} & x^{25} \\ x & 1 & x & x^4 & x^9 & x^{16} \\ x^4 & x & 1 & x & x^4 & x^9 \\ x^9 & x^4 & x & 1 & x & x^4 \\ x^{16} & x^9 & x^4 & x & 1 & x \\ x^{25} & x^{16} & x^9 & x^4 & x & 1 \end{pmatrix}$$

Take $S = (1, 4, 6)$

$\Lambda(S) = (1, 2, 3, 4, 5, 6) \setminus S = (2, 3, 5)$

$T(S) = (6 + 1 - 2, 6 + 1 - 3, 6 + 1 - 5) = (1, 3, 6)$

$T(\Lambda(S)) = (2, 4, 5)$

$$P(U|S) = \begin{pmatrix} 1 & x^9 & x^{25} \\ x & x^4 & x^{16} \\ x^4 & x & x^9 \end{pmatrix} = P(L|T(S)) = \begin{pmatrix} x^9 & x & x^4 \\ x^{16} & x^4 & x \\ x^{25} & x^9 & 1 \end{pmatrix}$$

Symmetry around mean

$S_1 : \sigma \longmapsto \sum_{j=1}^{n} (\sigma(j) - j)^2$ is symmetric around $n(n^2-1)/3$

$n = 3$: $$\begin{pmatrix} 1 & x & x^4 \\ x & 1 & x \\ x^4 & x & 1 \end{pmatrix}$$

$\Psi(x^b) = x^{(\frac{1}{3}n(n^2-1)-b)}$ + extension by linearity to polynomials

$\Psi(1+x^2) = x^8 + x^6 = x^4(x^4+x^2)$ Take $S = (1,2)$, then $\Lambda(S) = (3)$, $T(S) = (2,3), T(\Lambda(S)) = (1)$.

$$\Psi_8\Big(\operatorname{per}\big(P(U|S)\big)\operatorname{per}\big(P(L|\Lambda(S))\big)\Big) = \Psi_8(1+x^2) = x^6 + x^8$$

$$= \operatorname{per}\big(P(U|T(S))\big)\operatorname{per}\big(P(L|T(\Lambda(S)))$$

Page's L statistic

test for ordered alternatives in a randomised block design with b blocks.

β_j denotes the block effect of the jth treatment.

$$H_0 : \beta_1 = \ldots = \beta_t$$

$$H_1 : \beta_1 \leq \ldots \leq \beta_t,$$

with at least one strict inequality.

$$L = \sum_{j=1}^{t} j\, R_j,$$

$$G_{b,t}(x) = \sum_{k=0}^{\infty} P(L = k)\, x^k = \left(\frac{1}{t!} \operatorname{per}(\widetilde{P}) \right)^b$$

Remarks and results

- Cases with ties can be treated in a similar way.
- Cases with ties important, because no tables available.
- Existing tables of Spearman's ρ extended from $n = 18$ to 24 (extension of work of Franklin)
- Asymptotics are very accurate for $n = 24$
- Implementation faster than implementations in available statistical packages.

Linear rank score statistics

samples $X_1, \dots, X_m$ and $Y_1, \dots, Y_n$

$$Z_i = \begin{cases} 1 \text{ if } i\text{th order statistic in combined sample is an } x\text{-observation,} \\ 0 \text{ otherwise} \end{cases}$$

$$T = \sum_{i=1}^{n+m} a(i) Z_i$$

Examples:

- Wilcoxon rank sum statistic: $a(x) = x$
- Mood scale statistic: $a(x) = \left(x - \frac{n+m+1}{2}\right)^2$
- ...

Generating function for linear rank statistics

consider generating functions w.r.t. to *both* k (the values of T) and m (the sample size of the first sample).

Streitberg & Röhmel 1986 (cf. Euler 1748)

$$\sum_{m+n=N} \sum_{k} \binom{N}{m} P(T_{m,n} = k)\, x^k\, y^m = (1 + x^{a(1)}\, y) \ldots (1 + x^{a(N)}\, y),$$

where $T_{m,n} = \sum_{\ell=1}^{m+n} a(\ell)\, Z_\ell$.

The Streitberg-Röhmel generating function can be applied directly to :

- Mood scale statistic
- Freund-Ansari-Bradley statistic
- percentile modified rank statistics

Halperin statistic

Sometimes the Streitberg-Röhmel formula is not redundant:

independent samples $X_1, \ldots, X_m$ and $Y_1, \ldots, Y_n$

continuous distribution functions F and G, resp.

fixed right-censoring at time point$T$.

r_m X-observations and r_n Y-observations censored.

Halperin statistic (1960) $H_{m,n}$ is lower bound for $M_{m,n}$:

$$H_{m,n} = \#\{(i,j) \mid (Y_j < X_i \leq T)\} + r_n * (m - r_m)$$

Halperin statistic: continued

$$H_{m,n} = \#\{(i,j) \mid (Y_j < X_i \leq T)\} + r_n * (m - r_m)$$

Using

$$\Pr(A|B) = \sum_{k=1}^{N} \Pr(A|B \cap C_k) \Pr(C_k|B)$$

we may rewrite

$$\sum_k P(H_{m,n} = k \mid r_m + r_n = r)\, x^k$$

in terms of Mann-Whitney statistics for sample sizes $(\ell, m+n-r_m-r_n-\ell)$ $(1 \leq \ell \leq m)$.

This sum coincides with the Streitberg & Röhmel generating function.

Branch-and-bound algorithm

It is too time-consuming to expand the Streitberg & Röhmel generating function directly for other cases like

- Van der Waerden (normal scores) statistic (for $m = n = 15$ the generating function contains approximately 60000 terms).
- Klotz scale statistic

For testing purposes one only needs tail of distribution, *e.g.*

$$P(T > c) = \alpha.$$

Branch-and-bound algorithm exploit this by step-wise expansion of generating function with throwing parts that will not contribute.

Toy example Branch-and-bound: signed-rank statistic

$$2^n \sum_{k=0}^{\frac{1}{2}n(n+1)} P(W=k)x^k = (1+x)\left(1+x^2\right)\ldots\left(1+x^n\right)$$

Expand product term by term. Suppose we have already expanded $(1+x)(1+x^2) = 1+x+x^2+x^3$ and $n=4$.

Minimal contribution of remaining terms to exponents equals 0 .

Maximal contribution of remaining terms to exponents equals $3+4=7$.

If interested in $16P(W \geq 9)$: drop 1 and x from further expansion.

If interested in $16P(W \geq 3)$: drop x^3 and add contribution 4.

Intermezzo: contingency tables

The branch-and-bound method has also been applied to tests for contingency tables, although sometimes in disguised form.

Current approaches:

- network algorithms of Mehta and co-workers (implemented in StatXact)
- Recursive Polynomial Generating Method for $2 \times K$ tables: Hirji and Johnson, Comp. Stat. Data Anal. 21 (1996), 419–429
- Recursive Domain Partitioning for (likelihood ratio statistic for $r \times K$ tables, Cressie-Read statistics): Bejerano et al., J. Comp. Biology 11 (2004), 867–886

Alternative: Markov Chain Monte Carlo sampling, random sampling of contingency tables using Gröbner bases (Diaconis and Sturmfels, Ann. Stat. 26 (1998), 173–193).

Rank statistics for ANOVA

t independent samples of sizes n_j, $j = 1, \ldots, t$; $N = \sum_{j=1}^{t} n_j$

observations X_{ij} ($i = 1, \ldots, n_j$) belonging to treatment j are mutually independent and have unknown distribution function $F(x - \gamma_j)$, $j = 1, \ldots, t$.

$$H_0 : \gamma_1 = \ldots = \gamma_t = 0$$

$$H_1 : \exists j, j = 2, \ldots, t : \gamma_j \neq 0.$$

$$R_j = \sum_{\ell=1}^{N} a(\ell) Z_{\ell j}$$

a is rank score function (increasing w.l.o.g.)

$$Z_{\ell j} = \begin{cases} 1 & \text{if } \ell \text{ th smallest observation belongs to the } j\text{th sample} \\ 0 & \text{otherwise.} \end{cases}$$

Kruskal-Wallis type statistics

$$Q_{\vec{n}} = \sum_{j=1}^{t} R_j^2 / n_j$$

where $\vec{n} = (n_1, \dots, n_t)$.

Kruskal-Wallis: $a(\ell) = \ell$

Direct enumeration of $Q_{\vec{n}}$ not feasible.

Iman 1975 derived recursion for probabilities (inefficient).

Alternative approach: recursive build-up of pgf + branch-and-bound

Recursion: at each step we assign a rank to one of the treatments.

R is set of ranks already assigned; recursion starts at $R = \emptyset$.

Recursion for pgf $Q_{\vec{n}}$

$$F^R_{\vec{u}} = \{\, f : R \to \{1,\ldots,t\} | \,,\ \ (|f^{-1}(1)|, |f^{-1}(2)|, \ldots, |f^{-1}(t)|) = \vec{u}\,\}$$

$$A_f = \left(\sum_{r_1 \in f^{-1}(1)} a(r_1), \ldots, \sum_{r_t \in f^{-1}(t)} a(r_t) \right).$$

$$S^R_{\vec{u}} = \{\, A_f \mid f \in F^R_{\vec{u}}\,\}.$$

Generating function: $$G^R_{\vec{u}}(\vec{x}) = \sum_{\vec{v} \in S^R_{\vec{u}}} \#(f \in F^R_{\vec{u}} \mid A_f = \vec{v}) \prod_{i=1}^{t} {x_i}^{v_i}$$

$\varepsilon_j(\vec{u}) = (u_1, \ldots, u_j - 1, \ldots, u_t)$ for $j : u_j > 0$.

Recursion: $$G^R_{\vec{u}}(\vec{x}) = \sum_{j : u_j > 0} G^{R\setminus\{r\}}_{\varepsilon_j(\vec{u})}(\vec{x}) {x_j}^{a(r)} \qquad \text{if } \vec{u} \neq \vec{0}$$

Exchangeability

Terms in recursion may be computed efficiently by permuting labels:

$$\pi(y) := (y_{\pi(1)}, \ldots, y_{\pi(t)}), \ \pi \in \mathcal{S}_t$$

$$G_{\pi(\vec{u})}(\vec{x}) = G_{\vec{u}}(\pi^{-1}(\vec{x}))$$

where π^{-1} is inverse permutation of π

Corollary We only need recursion for $u_1 \geq \ldots \geq u_t$.

Further reduction is necessary because polynomials involved in recursion grow fast.

Reduction is possible for computing tail probabilities. Idea is similar to network algorithm of Mehta et al., but more transparent because of setting in terms of generating functions.

Branch-and-bound

given $G_{\vec{u}}^{R}(\vec{x})$ at certain recursion step

k^{th} term of this polynomial is of the form: $c\, {x_1}^{v_1} \cdots {x_t}^{v_t}$

$$Q_{\vec{u}}^{R,k} = \sum_{j=1}^{t} \frac{{v_j}^2}{n_j}.$$

$P(Q_{\vec{n}} \geq g)$ requires computing $Q_{\vec{n}}^{R,k}$ for all k, adding the coefficients c of those terms for which $Q_{\vec{n}}^{R,k} \geq g$ and dividing by $N!/(n_1! \cdots n_t!)$.

It is *not* necessary to compute all $Q_{\vec{n}}^{R,k}$!

Bounding

$\vec{u} = (u_1, \ldots, u_t)$ keeps track of number of assigned ranks to treatments.

Assume $U = \sum_{j=1}^{t} u_j < N$ at certain step in recursion

$$Q_{\vec{u},\vec{n}}^{R,k} = \sum_{j=1}^{t} \frac{(v_j + w_j)^2}{n_j},$$

where $\vec{w} = (w_1, \ldots, w_t)$ are unknown unassigned ranks.

Key idea: obtain lower and upper bounds for $Q_{\vec{u},\vec{n}}^{R,k}$.

If our lower bound $\geq g$, then $Q_{\vec{u},\vec{n}}^{R,k} \geq g$ for every assignment of remaining ranks. Hence, we have fixed contribution $c \binom{N-U}{n_1-u_1,\ldots,n_t-u_t} \Big/ \binom{N}{n_1,\ldots,n_t}$ to $P(Q_{\vec{n}} \geq g)$.
If our upper bound is $< g$, then $Q_{\vec{u},\vec{n}}^{R,k} < g$ for every assignment of remaining ranks. Hence, this term may be deleted.

Optimization problems

$$\min_{\vec{w}} \; Q^{R,k}_{\vec{u},\vec{n}} \quad \text{and} \quad \max_{\vec{w}} \; Q^{R,k}_{\vec{u},\vec{n}},$$

$$\text{subject to} \quad : \quad w_j = \sum_{r_j \in S_j} a(r_j),$$

$$|S_j| = n_j - u_j$$

$$\bigcup_{j=1}^{t} S_j = R,$$

quadratic form of $Q^{R,k}_{\vec{u},\vec{n}}$ and complex constraints obstruct fast computation

non-classical optimization problem because of repeated optimization

exact bounds for L^* and L^* not necessary: $L \leq L^*$ and $U \geq U^*$ the better the safe bounds, the more efficient our computation.

Splitting the objective function

$$Q^{R,k}_{\vec{u},\vec{n}} = \sum_{j=1}^{t} \frac{(v_j + w_j)^2}{n_j},$$

where $\vec{w} = (w_1, \ldots, w_t)$ are unknown unassigned ranks.

$$\begin{aligned} L &:= C + 2\,L_1 + L_2 = \textstyle\sum_{j=1}^{t} \frac{{v_j}^2}{n_j} + 2\min_{\vec{w}} \sum_{j=1}^{t} \frac{v_j w_j}{n_j} + \min_{\vec{w}} \sum_{j=1}^{t} \frac{{w_j}^2}{n_j} \\ U &:= C + 2\,U_1 + U_2 = \textstyle\sum_{j=1}^{t} \frac{{v_j}^2}{n_j} + 2\max_{\vec{w}} \sum_{j=1}^{t} \frac{v_j w_j}{n_j} + \max_{\vec{w}} \sum_{j=1}^{t} \frac{{w_j}^2}{n_j} \end{aligned}$$

Computation of bounds requires:

1. constant C (easy)
2. linear optimization (closed form)
3. quadratic optimization (hard, but does not depend on $\vec{v}$)

Relaxation of quadratic optimization

$$\min_{\vec{w}} \sum_{j=1}^{t} \frac{w_j^2}{n_j} \quad \text{and} \quad \max_{\vec{w}} \sum_{j=1}^{t} \frac{w_j^2}{n_j},$$

subject to :

$$\sum_{i \in S} w_i \leq \sum_{j=|R|-N_S+1}^{|R|} a(j) \qquad \text{for all } S \subseteq T$$

$$w_1 + w_2 + \ldots + w_t = \sum_{j=1}^{|R|} a(j)$$

Next step is to identify the extreme points of the feasible region.

Solution can be generated in a simple way because of hierarchical structure of constraints.

Our solution is much faster than the network algorithm in StatXact 4.

Summary

- probability generating functions are convenient tool to compute exact (null) distributions of rank statistics
- use recursion on level of generating functions rather than individual probabilities
- ideas to compute probability generating functions:
  - direct evaluation
  - reduction using symmetries (cf. Spearman's ρ)
  - double generating function (Streitberg-Röhmel)
  - blocks can be handled by exponentiation
  - for p- values one does not need complete generating function: branch-and-bound

References

Tables and software are available at

- http://www.win.tue.nl/∼markvdw/software.html
- http://www.win.tue.nl/∼markvdw/tables.html

M. A. van de Wiel, A. Di Bucchianico and P. van der Laan, Exact distributions of nonparametric test statistics and computer algebra, J. Roy. Statist. Soc. D The Statistician, 48 (1999), 507-516

M.A. van de Wiel and A. Di Bucchianico, Fast computation of the exact null distribution of Spearman's rho and Page's L statistic for samples with and without ties, J. Stat. Plann. Inf. 92 (2001), 133-145.

M.A. van de Wiel, The split-up algorithm: a fast symbolic method for computing p-values of rank statistics, Comp. Statistics, 16 (2001), 519-538.

References continued

M.A. van de Wiel, Exact non-null distributions of rank statistics, Communications Statistics: Simulation and Computation, 16 (2001), 1011–1028.

M.A. van de Wiel, Exact distributions of multiple comparisons rank statistics, Journal of the American Statistical Association 97 (2002), 1081–1089.

M.A. van de Wiel, Exact null distributions of quadratic distribution-free statistics for two-way classification, J. Stat. Plann. Inf. 120 (2004), 29–40.

A. Di Bucchianico and M.A. van de Wiel, Exact null distributions of distribution-free quadratic t-sample statistics, J. Stat. Plann. Inf.127 (2005), 1–21 .