Journal of Statistical Software

January 2007, Volume 17, Issue 8. http://www.jstatsoft.org/

A C++ Program for the Cramér-von Mises Two-Sample Test

Yuanhui Xiao
University of Rochester
Georgia State University

Alexander Gordon
University of Rochester
University of North Carolina at Charlotte

Andrei Yakovlev
University of Rochester

Abstract

As larger sets of high-throughput data in genomics and proteomics become more readily available, there is a growing need for fast algorithms designed to compute exact p values of distribution-free statistical tests. We present a program for computing the exact distribution of the two-sample Cramér-von Mises test statistic under the null hypothesis that the two samples are drawn from the same continuous distribution. The program makes it possible to handle substantially larger sample sizes than earlier proposed computational tools. The C++ source code for the program is published with this paper, and an R package is under development.

Keywords: Cramér-von Mises test, exact p values, distribution-free tests.

1. Introduction

High-throughput technologies such as gene expression microarrays and proteomics are opening up a new era in biology and medicine. Such technologies yield an abundance of valuable quantitative information posing new challenges to the statistician. It has become common practice to use microarray technology for selecting "interesting" genes by comparing expression levels of thousands of genes in two different phenotypes. Modern methods for finding such differentially expressed genes typically employ two-sample statistical tests combined with multiple testing procedures to guard against Type 1 errors (see Dudoit *et al.* 2003; Simon *et al.* 2003; Speed 2003; McLachlan *et al.* 2004; Lee 2004; Wit and MacClure 2004, for reviews). As larger sets of microarray gene expression data become more readily available, nonparametric methods for testing multiple two-sample hypotheses in microarray data analysis are beginning

to be more appreciated (Stamey *et al.* 2001; Grant *et al.* 2002; Troyanskaya *et al.* 2002; Xiao *et al.* 2004; Guan and Zhao 2005; Lee *et al.* 2005; Qiu *et al.* 2006, to name a few). For example, the Cramér-von Mises test appears to be quite competitive with the t test even when its power is assessed by simulating normally distributed log-expression levels under location alternatives (Qiu *et al.* 2006), conditions under which the t test is known to be optimal. The Cramér-von Mises test can provide a substantial gain in power as compared to the t test under some other departures from the null hypothesis.

In applications to microarrays and other types of high-throughput data, distribution-free statistical tests call for fast algorithms for computing exact p values because relevant asymptotic results (see Csörgö and Faraway 1996, for the Cramér-von Mises test) do not provide the required accuracy of approximation in the tail region of the corresponding limiting distribution. The reason is that the widely-used multiple testing procedures controlling the family-wise error rate (FWER) are focused on the region of very small p values. To illustrate this point, consider the following example. Suppose that one wishes to find changes in the marginal distributions of expression levels of 12558 genes when comparing two groups of patients of equal size $n = m = 43$ as in the application reported by Qiu *et al.* (2006). In other words, there are 12558 two-sample hypotheses to be tested in this setting. Consider a particular gene for which the observed Cramér-von Mises statistic value equals $A = 2.2253921$ with the exact and asymptotic p values being equal to 2.115×10^{-6} and 3.994×10^{-6}, respectively. The Bonferroni-adjusted p values are, therefore, equal to .02656 and .05015 respectively. Let the statistic value for another gene be equal to $B = 2.1193889$ so that the exact and asymptotic Bonferroni-adjusted p values are .0493 and .0866, respectively. As a result, all the genes with values of the Cramér-von Mises test statistic falling in the interval (B, A) will be declared differentially expressed when using exact p values, but they will not be selected if asymptotic p values are used at the same FWER level of 0.05. This example shows that the development of efficient numerical algorithms for computing exact p values has no sound alternative. Such algorithms should be designed to handle large sample sizes, that is situations where direct rearrangement methods are computationally prohibitive. In this paper, we present a C++ software that implements a numerical algorithm for computing the exact distribution of the two-sample Cramér-von Mises test statistic.

The Cramér-von Mises two-sample test is one of the best-known distribution-free two-sample tests. It is based on the statistic T_2 defined below. Suppose two independent samples $x_1, x_2, \ldots, x_m$ and $y_1, y_2, \ldots, y_n$ are drawn independently from two distributions with continuous cumulative distribution functions $F(\cdot)$ and $G(\cdot)$, respectively. Based on those samples, we want to test the null hypothesis $H_0 \colon F = G$ against the two-sided alternative $H_1 : F \neq G$. The Cramér-von Mises test statistic T_2 is given by

$$T_2 = \frac{mn}{(m+n)^2} \left\{ \sum_{i=1}^{m} (F_m(x_i) - G_n(x_i))^2 + \sum_{j=1}^{n} (F_m(y_j) - G_n(y_j))^2 \right\}, \tag{1}$$

where F_m and G_n are the empirical distribution functions associated with the respective samples (x_i) and (y_j). This statistic and the test based on it (rejecting H_0 if the value of T_2 is "too large") were first studied by Anderson (1962) as a 2-sample variant of the goodness-of-fit test introduced by Cramér (1928) and von Mises (1931).

The statistic (1) has a simple meaning. Move the $m+n$ points $x_1, x_2, \ldots, x_m$ and $y_1, y_2, \ldots, y_n$, without changing their mutual order, to their new positions, which are $1/(m+n)$, $2/(m+$

$n), \ldots, (m+n)/(m+n) = 1$. Let $\{\xi_1, \ldots, \xi_m\}$ and $\{\eta_1, \ldots, \eta_n\}$ be two subsets of the set $\{1/(m+n), 2/(m+n), \ldots, 1\}$ formed by the x_i's and y_j's, respectively, and let F_m^* and G_n^* be the corresponding empirical distribution functions. Then T_2 equals, up to a constant factor depending only on m and n, the squared L_2-distance between F_m^* and G_n^*.

Anderson (1962) and Burr (1963) studied the distribution of T_2 under H_0 (which, obviously, does not depend on the distribution $F = G$) and provided some tables. In particular, Anderson (1962) listed some percentiles for T_2 when $m, n \leq 7$. Burr (1964) did the same for $m + n \leq 17$. However, the methods they used rely on the listing of all splittings of the set $\{1, 2, \ldots, m+n\}$ into a pair of subsets of cardinalities m and n (such splittings are often imprecisely called "permutations"). As a result, these methods are only applicable to small samples.

Burr (1963) proposed an iterative method, which does not use permutations and is less computationally intensive. Using it, he found the exact distribution of T_2 for $m = n = 10$. Burr presented his algorithm for the case $m = n$ only, but it works for the general case as well (Hájek and Šidák 1967). It is worth mentioning that the set of selected quantiles for T_2 with $m = n \leq 23$ by Zajta and Pandikow (1977) are still based on permutations. The software proposed in the present paper makes it possible to handle substantially larger sample sizes m and n.

The paper is organized as follows. In section 2, the necessary technical details of the basic numerical algorithm (based on Burr's idea) are given. Section 3 describes the usage of the software. The C++ source code for the software is published with this paper and an R package is under development.

2. Basic ideas behind the design of the software

As we have mentioned in the Introduction, the design of the software is based on ideas originally explored by Burr (1963). The recurrence relations we use are akin to those given by Hájek and Šidák (1967).

2.1. Tabulating the distribution function

Let $z_1, z_2, \ldots, z_{m+n}$ be the pooled and ordered sample of $x_1, x_2, \ldots, x_m$ and $y_1, y_2, \ldots, y_n$. Under the assumption that each sample is drawn from a continuous population, the pooled sample almost surely has no ties. The test statistic only depends on the differences $F_m(z_t) - G_n(z_t)$ of the two empirical distribution functions at the observed values. Let $h_t = L[F_m(z_t) - G_n(z_t)]$, where L is the least common multiple of m and n. It follows that h_t's ($t = 0, 1, 2, \ldots, m+n$) are integers and T_2 differs only by a constant factor from the integer-valued random variable

$$\zeta = \sum_{t=1}^{m+n} h_t^2. \tag{2}$$

Specifically, $T_2 = (mn/(m+n)^2 L^2)\zeta$. Then, the purpose is to find the distribution of ζ. To do that, it suffices to find the frequency function of ζ.

Throughout the paper, by a *frequency function* we understand a non-negative integer-valued function f on the set $\mathbf{Z}$ of all integers, such that the set $\{i \in \mathbf{Z}\colon f(i) \neq 0\}$ is non-empty and finite.

In computer, such a frequency function can be represented either by an array of values $f(i)$ on a sufficiently long "interval" $\{0, 1, 2, \dots, I\}$ (with I so large that $f(i) = 0$ if $i > I$), or by an array of pairs of integers $(i,\ f(i))$, where we may only store the pairs with $f(i) \neq 0$. We will call an array of either kind a *frequency table* and will use the terms "frequency function" and "frequency table" interchangeably. Note that in the actual program the second method of storage of frequency tables is used.

Let B be a finite set with the uniform probability measure on it, so that every element $b \in B$ has probability $1/|B|$. (Note that $|\cdot|$, for a finite set $\cdot$, stands for its cardinality.) Let η be an integer-valued random variable on B, i.e., a mapping $\eta\colon B \to \mathbf{Z}$. Then by the *frequency function of* η (over B) we understand the function $f_\eta\colon \mathbf{Z} \to \mathbf{Z}_+$ defined by

$$f_\eta(i) = |\{b \in B\colon \eta(b) = i\}|.$$

Obviously, the probability of the event $\eta = i,\ i \in \mathbf{Z}$, equals $f_\eta(i)/|B|$. Note that if the random variable η is non-negative (which will always be the case in this paper), then $f_\eta(i) = 0$ for all $i < 0$.

Since $F_m(z_{m+n}) = G_n(z_{m+n}) = 1$, we have $h_{m+n} = 0$. Let $h_0 = 0$, then the sequence $(h_t)_{t=0}^{m+n}$, for a given pair of samples (x_i) and (y_j), satisfies the following relation:

$$h_t = \begin{cases} h_{t-1} + a, & \text{if } z_t = x_i \text{ for some } i, \\ h_{t-1} - b, & \text{if } z_t = y_j \text{ for some } j, \end{cases} \tag{3}$$

for $1 \le t \le m + n$, where $a = L/m$ and $b = L/n$.

The sequence $(h_t)_{t=0}^{m+n}$ can be represented by a broken line (or, briefly, a *path*) on the plane $\mathbf{R}^2$ (see Figure 1), joining the points $(t, h_t),\ t = 0, 1, 2, \dots, m + n$ (see Figure 1). Note that the path starts at the point $(0, 0)$, ends at the point $(m+n,\ 0)$, and all of its $m+n+1$ vertices belong to the lattice $\mathbf{Z}^2$; m legs of the path are parallel translations of the vector $(1, a)$, and the other n legs – translations of the vector $(1, -b)$ (see Figure 1). There are totally $\binom{m+n}{m}$ such paths, and under the null hypothesis all of them are equally likely.

It follows from (3) and $h_0 = h_{m+n} = 0$ that the possible values of h_t (for a fixed t, $0 \le t \le m + n$), form a finite arithmetic progression H_t with the common difference $a + b$:

$$H_t = \{l_t, l_t + (a + b), l_t + 2(a + b), \dots, u_t\}, \tag{4}$$

where

$$l_t = \max\{-bt,\ -L + a(t - n)\} \tag{5}$$

and

$$u_t = \min\{at,\ L - b(t - m)\}.$$

The set H_t contains at most $s = \min(m, n) + 1$ points (exactly s points if t is between m and n) and $H_{m+n} = H_0 = \{0\}$ (see Figure 1).

For $t \in \{0, 1, 2, \dots, m + n\}$ and $d \in H_t$, let $f^+_{t,d}$ denote the frequency function of

$$\sum_{i=0}^{t} h_i^2 \tag{6}$$

over all paths joining the points $(0, 0)$ and (t, d). In other words, $f^+_{t,d}(s)$ is the number of paths from $(0, 0)$ to (t, d), such that the sum (6) equals s.

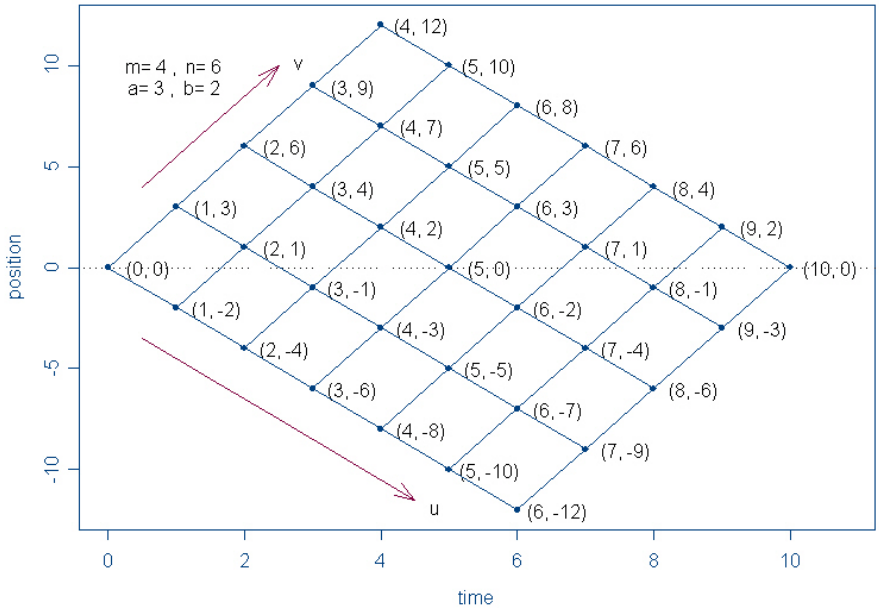

Figure 1: The paths ($m = 4,\ n = 6;\ a = 3,\ b = 2$)

Apparently, $f^+_{m+n,0}$ is the frequency function of (2), and the sum of all its values (frequencies) is the total number of paths joining $(0, 0)$ and $(m + n, 0)$, i.e., $\binom{m+n}{m}$. Hence, the probability mass function of (2) at point $i \in \mathbf{Z}$ (i.e., the probability that $\zeta = i$) equals

$$\binom{m+n}{m}^{-1} f^+_{m+n,0}(i). \tag{7}$$

For $z \in \mathbf{Z}$, denote by $I[z]$ the frequency function which is one at z, and zero elsewhere. It is obvious that $f^+_{0,0} = I[0]$. More generally,

$$\left\{ \begin{array}{rcl} f^+_{t,l_t} \equiv f^+_{t,-bt} & = & I[b^2 t(t+1)(2t+1)/6], \quad t = 0,\ 1,\ 2,\ \cdots,\ n; \\ f^+_{t,u_t} \equiv f^+_{t,at} & = & I[a^2 t(t+1)(2t+1)/6], \quad t = 0,\ 1,\ 2,\ \cdots,\ m, \end{array} \right. \tag{8}$$

because there is exactly one path joining the points $(0, 0)$ and $(t, -bt)$, and similarly for the points $(0, 0)$ and (t, at).

Any path joining the points $(0, 0)$ and (t, d) $(1 \le t \le m + n)$ should pass one of the two points $(t - 1, d - a)$ and $(t - 1, d + b)$. Therefore,

$$f^+_{t,d}(i) = f^+_{t-1,d-a}(i - d^2) + f^+_{t-1,d+b}(i - d^2), \quad i \in \mathbf{Z}, \tag{9}$$

which can also be rewritten as:

$$f^+_{t,d} = S_{d^2}[f^+_{t-1,d-a} + f^+_{t-1,d+b}]. \tag{10}$$

Here $S_r,\ r \in \mathbf{Z}$, is the *right shift* operator: given a function $f(\cdot)$ on $\mathbf{Z}$, the function $S_r f$ may be defined as follows:

$$(S_r f)(i) = f(i-r) \quad \text{for all} \quad i \in \mathbf{Z}.$$

Recurrence relation (9), together with equations (8), allows to compute frequency functions $f^+_{t,d}$ recursively, starting from $f^+_{0,0} = I[0]$ and ending up with the function $f^+_{m+n,0}$. However, the algorithm can be simplified if we change the "coordinates" t, d in the parallelogram formed by all paths from $(0,0)$ to $(m+n,0)$ (see Figure 1) to new coordinates u, v. These coordinates are defined as follows.

For a given path $h_i,\ i = 0, 1, \ldots, t$, from $(0,0)$ to (t,d), let $u := |\{j\colon 1 \le j \le t,\ h_t - h_{t-1} = -b\}|$ and $v := |\{j\colon 1 \le j \le t,\ h_t - h_{t-1} = a\}|$ (in other words, u and v are the number of y's and x's, respectively, among the first t z's). It follows that, given t and d, we have $v+u = t,\ av-bu = d$, which implies that u and v are uniquely determined by t and d. It is also clear that the pair (u,v) takes all values in $\mathbf{Z}^2$, such that

$$0 \le u \le n,\ 0 \le v \le m.$$

We can use the new coordinates u and v to label the frequency functions $f^+_{t,d}$, putting

$$g^+_{u,v}(i) := f^+_{u+v,av-bu}(i), \quad i \in \mathbf{Z},$$

for all u, v, such that $0 \le u \le n,\ 0 \le v \le m$. Then (10) becomes

$$g^+_{u,v} = S_{d^2}(g^+_{u-1,v} + g^+_{u,v-1}), \quad \text{where} \quad d = av - bu, \tag{11}$$

while (8) transforms into equalities

$$g^+_{u,0} = I[b^2 u(u+1)(2u+1)/6], \quad 0 \le u \le n \tag{12}$$

and

$$g^+_{0,v} = I[a^2 v(v+1)(2v+1)/6], \quad v \le u \le m. \tag{13}$$

Note that $g^+_{n,m} = f^+_{m+n,\,0}$. Relations (11), (12) and (13) lead to the following algorithm for computing the frequency functions $g^+_{u,v}$.

Algorithm 1.

for $(v \leftarrow 0;\ v \le m;\ v \leftarrow v+1)$
  $g^+_{0,v} \leftarrow I[a^2 v(v+1)(2v+1)/6]$
for $(u \leftarrow 1;\ u \le n;\ u \leftarrow u+1)$
  $g^+_{u,0} \leftarrow I[b^2 u(u+1)(2u+1)/6]$
  for $(v \leftarrow 0;\ v \le m;\ v \leftarrow v+1)$
    $g^+_{u,v} \leftarrow S_{(av-bu)^2}[g^+_{u-1,v} + g^+_{u,v-1}]$

Note that, as soon as the inner v-loop is complete, the frequency tables $g^+_{u-1,v},\ 0 \le v \le m$, are not needed any longer, and the memory they occupy can be freed.

In the case $m = n$, the following algorithm allows to store only half of necessary frequency tables using the identity $f^+_{t,-d} \equiv f^+_{t,d}$.

Algorithm 1* (case m=n).

$f^+_{0,0} \leftarrow I[0]$
for $(t \leftarrow 1;\ t \le 2n;\ t \leftarrow t+1)$
  $x \leftarrow \min(t, 2n-t)$
  for $(d \leftarrow x;\ d \ge 0;\ d \leftarrow d-2)$
    if $(d = t)$
      $f^+_{t,d} \leftarrow I[t(t+1)(2t+1)/6]$
    else
      $f^+_{t,d} \leftarrow f^+_{t-1,d+1} + f^+_{t-1,|d-1|}$

We have conducted numerical experiments to compare the computed tail probabilities with those tabulated by Burr (1963). In Burr's tables, the values of the probability mass function are represented as ordinary fractions and they are in complete agreement with the results of our computations.

2.2. Computing p values

In practice, one usually needs tail probabilities of the test statistic (p values), rather than probabilities of individual values. Of course, we can compute the individual probabilities and then find the tail probabilities by summation. In this subsection, we describe an alternative way of computing them, which allows larger sample sizes.

For $t = 0, 1, \ldots, m+n$ and $d \in H_t$, denote by $f^-_{t,d}$ the frequency function of

$$\sum_{i=t}^{m+n} h_i^2 \tag{14}$$

over all paths connecting the points (t, d) and $(m+n, 0)$. In view of symmetry,

$$f^-_{t,d} = f^+_{m+n-t,-d}. \tag{15}$$

Proposition . Let $M = [(m+n)/2]$ ($[z]$ stands for the integer part of z). If $m+n$ is even, the frequency function $f^+_{m+n,0}$ is

$$f^+_{m+n,0} = \sum_{d \in H_M} S_{-d^2}[f^+_{M,d} * f^+_{M,-d}]. \tag{16}$$

If $m+n$ is odd,

$$f^+_{m+n,0} = \sum_{d \in H_M} f^+_{M,d} * f^+_{M,-d-a} + \sum_{d \in H_M} f^+_{M,d} * f^+_{M,-d+b}. \tag{17}$$

Here $*$ denotes the convolution of frequency functions: $(g * h)(i) = \sum_{k \in \mathbf{Z}} g(k)h(i-k)$.

Remark. Note that $f^+_{M,d} \equiv 0$ if $d < l_M$ or $d > u_M$ (see (4)).

Proof. Let f_d^M $(d \in H_M)$ stand for the frequency function of (2) over all paths that connect the points $(0,0)$ and $(m+n,0)$ through the point (M,d). Any path connecting the points $(0,0)$ and $(m+n,0)$ should pass one and only one of the points (M,d), $d \in H_M$. Therefore, the function $f^+_{m+n,0}$ can be decomposed into the sum of functions f_d^M, $d \in H_M$:

$$f^+_{m+n,0} = \sum_{d \in H_M} f_d^M. \tag{18}$$

Suppose $m+n$ is even, so that $m+n = 2M$; since the path's part connecting $(0,0)$ with (M,d) and the part connecting (M,d) with $(m+n,0)$ can be chosen independently, we have

$$f_d^M(i-d^2) = \sum_{j \in \mathbf{Z}} f^+_{M,d}(j)\, f^-_{M,d}(i-j) \equiv (f^+_{M,d} * f^-_{M,d})(i).$$

(The argument $i-d^2$, rather than i, of f_d^M is due to the fact that, adding the two sums (6) and (14), we "almost" obtain the sum (2), but take h_t^2 into accout twice.) Equivalently, we have $S_{d^2} f_d^M = f^+_{M,d} * f^-_{M,d}$, or

$$f_d^M = S_{-d^2}[f^+_{M,d} * f^-_{M,d}]. \tag{19}$$

Applying (18), (19) and (15), we obtain (16).

Suppose now that $m+n$ is odd, so that $m+n = 2M+1$. Any path joining the points $(0,0)$ and $(m+n,0)$ through (M,d) should pass exactly one of the two points $(M+1, d+a)$ and $(M+1, d-b)$; therefore,

$$f_d^M = f^+_{M,d} * f^-_{M+1,d+a} + f^+_{M,d} * f^-_{M+1,d-b}\ . \tag{20}$$

Applying (18), (20) and (15), we obtain (17). □

From now on we will assume that

$$m \le n. \tag{21}$$

Formulas (16) and (17) show that, in order to tabulate the probability mass function of (2), it is sufficient to have the frequency tables $f^+_{M,d}$, $d \in H_M$. Since $f^+_{M,d} = f^+_{u+v,av-bu} = g^+_{u,v}$ for some pair (u,v) with $u+v = M$ and $av-bu = d$, these frequency tables actually are $\{g_{0,M},\ g_{1,M-1},\ \ldots, g_{m,M-m}\}$. Therefore, they can be obtained using the following algorithm.

Algorithm 2.

for $(v \leftarrow 0;\ v \le m;\ v \leftarrow v+1)$
  $g^+_{0,v} \leftarrow I[a^2 v(v+1)(2v+1)/6]$
for $(u \leftarrow 1;\ u \le M;\ u \leftarrow u+1)$
  $g^+_{u,0} \leftarrow I[b^2 u(u+1)(2u+1)/6]$
  for $(v \leftarrow 0;\ v \le \min(m, M-u);\ v \leftarrow v+1)$
    $g^+_{u,v} \leftarrow S_{(av-bu)^2}[g^+_{u-1,v} + g^+_{u,v-1}]$

Algorithm 2* (case m=n).

$f^+_{0,0} \leftarrow I[0]$
for $(t \leftarrow 1;\ t \le n;\ t \leftarrow t+1)$
  $x \leftarrow \min(t, 2n-t)$
  for $(d \leftarrow x;\ d \ge 0;\ d \leftarrow d-2)$
    if $(d = t)$
      $f^+_{t,d} \leftarrow I[t(t+1)(2t+1)/6]$
    else
      $f^+_{t,d} \leftarrow f^+_{t-1,d+1} + f^+_{t-1,|d-1|}$

It follows from the assumption (21) that $l_M = -bM$ (see (5)), so that each integer $d \in H_M$ equals one of the numbers

$$d_v = -bM + (b+a)v \tag{22}$$

with some $v \in \{0, 1, \ldots, m\}$. Consequently, $d_{m-v} + d_v$ equals 0 or b, if $m+n$ is even or odd, respectively. Therefore, $-d_v = d_{m-v}$ in the former case, while $-d_v + b = d_{m-v}$ and $-d_v - a = d_{m-1-v}$ in the latter case.

Using these facts, we can re-write (16) as

$$f^+_{m+n,0} = \sum_{v=0}^{m} S_{-d_v^2}[f^+_{M,d_v} * f^+_{M,d_{m-v}}] \quad (m+n \text{ is even}) \tag{23}$$

and (17) as

$$f^+_{m+n,0} = \sum_{v=0}^{m-1} f^+_{M,d_v} * f^+_{M,d_{m-v-1}} + \sum_{v=0}^{m} f^+_{M,d_v} * f^+_{M,d_{m-v}} \quad (m+n \text{ is odd}) \tag{24}$$

Convoluting large frequency tables is computationally intensive. However, if we need to compute only several p values, this can be done fast.

Associate with each frequency function f its tail function $f^{\ge}$ defined as

$$f^{\ge}(i) = \sum_{j \ge i} f(j), \quad i \in \mathbf{Z}.$$

We have

$$(f+g)^{\ge} = f^{\ge} + g^{\ge}; \tag{25}$$

$$(S_r f)^{\ge} = S_r[f^{\ge}]; \tag{26}$$

$$(f * g)^{\ge} = f * g^{\ge} = g * f^{\ge}, \tag{27}$$

where

$$f * g^{\ge}(i) = \sum_{k \in \mathbf{Z}} f(k) g^{\ge}(i-k).$$

Assume Q is a given value of (2),then the corresponding p value, P, is

$$P = Pr\{\zeta \geq Q\} = \binom{m+n}{m}^{-1} \sum_{i \geq Q} f^{+}_{m+n,0}(i) = \binom{m+n}{m}^{-1} (f^{+}_{m+n,0})^{\geq}(Q). \tag{28}$$

Using (25) and (26), we can derive from (23) and (24) that

$$(f^{+}_{m+n,0})^{\geq}(Q) = \sum_{v=0}^{m} (f^{+}_{M,d_v} * f^{+}_{M,d_{m-v}})^{\geq}(Q + d^2), \tag{29}$$

if $m + n$ is even, and

$$(f^{+}_{m+n,0})^{\geq}(Q) = \sum_{v=0}^{m-1} (f^{+}_{M,d_v} * f^{+}_{M,d_{m-v-1}})^{\geq}(Q) + \sum_{v=0}^{m} (f^{+}_{M,d_v} * f^{+}_{M,d_{m-v}})^{\geq}(Q), \tag{30}$$

if $m + n$ is odd.

Now, the only need is to efficiently evaluate quantities of the following type:

$$R = (f * g)^{\geq}(c),$$

where f and g are two frequency functions and c an integer. We proceed further along the lines of the work by van de Wiel (2001).

Let $f_i = f(u_i)$, $i = 1, 2, \ldots, k$, be all the non-zero values of the frequency function f, and $g_j = g(v_j)$, $j = 1, 2, \ldots, l$, be those of g. We assume that both sequences u_i and v_j are strictly increasing. By the definition of R,

$$R = \sum_{q \geq c} [f * g](q) = \sum_{q \geq c} \sum_{u+v=q} f(u)g(v) = \sum_{u+v \geq q} f(u)g(v),$$

or

$$R = \sum_{u_i + v_j \geq c} f_i g_j = \sum_{i=1}^{k} f_i G_i, \tag{31}$$

where $G_i = \sum_{j:\, v_j \geq c - u_i} g_j$. For $i = 1, 2, \ldots, k$, let $r_i = \min\{j \colon v_j \geq c - u_i\}$ ($r_i = l + 1$ if no such j exist, i.e., $v_l < c - u_i$). Then $1 \leq r_k \leq r_{k-1} \leq \ldots \leq r_1 \leq k + 1$. Note that all r_i can be found using one linear run through both arrays u_i and v_j: we put $i = 1$, $j = k + 1$ and decrease j until the inequality $v_j \geq c - u_i$ fails or $j = 0$; the previous j is r_1. We increase i by 1 and continue to decrease j until the same inequality fails or $j = 0$; the previous j is r_2; etc.

Now we can compute the G_i's recursively:

$$\begin{cases} G_1 &= \sum_{j=r_1}^{k} g_j, \\ G_i &= G_{i-1} + \sum_{j=r_i}^{r_{i-1}-1} g_j, \quad \text{for } i = 2, 3, \ldots, k-1, k. \end{cases} \tag{32}$$

In view of equations (29) and (30), the p value $P = \Pr\{\zeta \geq Q\}$ can now be computed using the following algorithm.

Algorithm 3^e (even $m+n$).

$s = 0$
for $(v \leftarrow 0;\ v \le m;\ v \leftarrow v+1)$
  $w \leftarrow m - v$
  $q \leftarrow (f^+_{M,d_v} * f^+_{M,d_w})^{\ge}(Q+d^2)$
  $s \leftarrow s + q$
$P \leftarrow \binom{m+n}{m}^{-1} s$

Algorithm 3^o (odd $m+n$).

$s = 0$
for $(v \leftarrow 0;\ v \le m-1;\ v \leftarrow v+1)$
  $w \leftarrow m - v - 1$
  $q \leftarrow (f^+_{M,d_v} * f^+_{M,d_w})^{\ge}(Q)$
  $s \leftarrow s + q$
for $(v \leftarrow 0;\ v \le m;\ v \leftarrow v+1)$
  $w \leftarrow m - v$
  $q \leftarrow (f^+_{M,d_v} * f^+_{M,d_w})^{\ge}(Q)$
  $s \leftarrow s + q$
$P \leftarrow \binom{m+n}{m}^{-1} s$

We have made some additional improvements in the code to increase its efficiency. We do not describe these small modifications at length, lest the exposition become too cumbersome. Technical details are available from the corresponding author upon request.

3. Usage

The `XCVMTest` program allows to compute the exact null distribution of the Cramér-von Mises test statistic, as well as p values corresponding to given values of the statistic. The program works in command line mode, its C++ source code is available along with this paper. There are four ways to use the program.

A. `XCVMTest m n`

The program computes the distribution of the Cramér-von Mises statistic: its values, their probabilities and the corresponding p values. In addition, the output contains the related integers (see Section 2.1): ζ (the scaled statistic), $f(\zeta)$ (the frequency) and $\sum_{j \ge \zeta} f(j)$ (the cumulative frequency). The command line arguments m and n are the sample sizes. E.g., typing `XCVMTest 10 10` gives Table 2 in Burr (1963). The output is self-explanatory.

B. `XCVMTest [-d|-f]  m n` $t_1\ t_2\ \ldots\ t_r$

The arguments m and n are the sample sizes; $t_1, \ldots, t_r$ ($r \geq 1$) are the given values of the Cramér-von Mises statistic. The output contains pairs consisting of the statistic values and the corresponding p values. (More precisely, the given values are first re-scaled to the ζ-scale (see Section 2.1) and rounded to the nearest integer, then the corresponding p values are computed.) The values `-d` and `-f` of the optional parameter have the following meaning: the option `-d` implies computing the full distribution of the Cramér-von Mises statistic (see Section 2.1), the p values being then obtained by summation. The option `-f` implies a more efficient computation using convolutions (see Section 2.2), which reduces the memory load and, therefore, extends the range of pairs (m, n) for which the computation is possible. The default option is `-f`.

C. `XCVMTest [-d|-f]  m n  StatFileName`

The only difference from B is that the given values of the Cramér-von Mises statistic are being read from a text file (whose name is the last argument of the command) rather than from the command line. The file consists of the statistic values separated with delimiters; possible delimiters are: the space, the tab character, and the end-of-line character. The values `-d` and `-f` of the optional parameter have the same meaning as in B.

D. `XCVMTest [-d|-f]  DataFileName`

The program evaluates the Cramér-von Mises statistic and the corresponding p value for each pair of samples contained in the text file whose name is the last argument of the command. Here is the format of the file:

$$\begin{array}{cccccccc} m & n & & & & & & \\ X_{1,1} & X_{1,2} & \ldots & X_{1,m} & Y_{1,1} & Y_{1,2} & \ldots & Y_{1,n} \\ X_{2,1} & X_{2,2} & \ldots & X_{2,m} & Y_{2,1} & Y_{2,2} & \ldots & Y_{2,n} \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ X_{r,1} & X_{r,2} & \ldots & X_{r,m} & Y_{r,1} & Y_{r,2} & \ldots & Y_{r,n} \end{array}$$

The first line of the file contains the sample sizes m and n separated by a space. Any other line of the file consists of $m + n$ numbers separated by spaces or tab characters: the first sample (m numbers) followed by the second sample (n numbers).

The values `-d` and `-f` of the optional parameter have the same meaning as in B.

If the program, at some step of computation, cannot allocate enough memory, it displays a message to this effect and stops.

Our computation experiments were carried out on a UNIX workstation (Sunfire V480) with 16.3GB RAM, 4×8.0MB Cache and 4×1200MHz CPU. The computation was still successful for the following sample sizes m and n in the extreme cases $m = n$ ($L = n$) and $m = n-1$ ($L = n(n-1)$):

- A: `m=n=200` and `m=60`, `n=61`;
- B, C, D with optional parameter `-d`: `m=n=200` and `m=64`, `n=65`;

- B, C, D with optional parameter `-f`: `m=n=250` and `m=80`, `n=81`.

The following table presents the computing time for various pairs of sample sizes (m, n) for each option.

$m = n$	1	2	$m = n$	1	2	$m,\ n$	1	2
40	1.00	0.15	100	160.93	17.39	10, 11	0.01	0.00
50	3.21	0.44	110	282.00	29.14	20, 21	1.19	0.12
60	9.29	1.12	120	476.17	46.42	30, 31	23.63	2.60
70	21.94	2.48	130	774.07	71.28	40, 41	193.25	22.56
80	45.98	5.10	140	1212.94	107.19	50, 51	833.79	119.12
90	87.58	9.72	150	1792.01	154.98	60, 61	4053.00	435.77

Table 1: The CPU time used by the program. For mode A, or modes B, C, D with option `-d`, the time is in column 1; for modes B, C, D with option -f, the time is in column 2. The CPU time is measured in seconds.

Acknowledgements

The work was supported in part by NIH grant GM075299.

References

Anderson TW (1962). "On the Distribution of the Two-Sample Cramèr-von Mises Criterion." *The Annals of Mathematical Statistics*, **33**(3), 1148–1159.

Burr EJ (1963). "Small-Sample Distribution of the Two-Sample Cramér-von Mises Criterion for Small Equals Samples." *The Annals of Mathematical Statistics*, **34**(1), 95–101.

Burr EJ (1964). "Distribution of the Two-Sample Cramér-von Mises W^2 and Watson's U^2." *The Annals of Mathematical Statistics*, **35**(3), 1091–1098.

Cramér H (1928). "On the Composition of Elementary Errors: II. Statistical Applications." *Skandinavisk Aktuarietidskrift*, **11**, 141–180.

Csörgö S, Faraway JJ (1996). "The Exact and Asymptotic Distributions of Cramér-von Mises Statistics." *Journal of the Royal Statistical Society B*, **58**, 221–234.

Dudoit S, Shaffer JP, Boldrick JC (2003). "Multiple Hypothesis Testing in Microarray Experiments." *Statistical Science*, **18**, 71–103.

Grant GR, Manduchi E, Stoeckert CJ (2002). "Using Non-parametric Methods in the Context of Multiple Testing to Determine Differentially Expressed Genes." In SM Lin, KF Johnson (eds.), "Methods of Microarray Data Analysis: Papers from CAMDA '00," pp. 37–55. Kluwer Acedamic, Norwell, Massachusetts.

Guan Z, Zhao H (2005). "A Semiparametric Approach for Marker Gene Selection Based on Gene Expression Data." *Bioinformatics*, **21**, 529–536.

Hájek J, Šidák Z (1967). *Theory of Rank Tests*. Academic Press, New York.

Lee MLT (2004). *Analysis of Microarray Gene Expression Data*. Kluwer, Boston.

Lee MLT, Gray R, Björkbacka H, Freeman MW (2005). "Generalized Rank Tests for Replicated Microarray Data." *Statistical Applications in Genetics and Molecular Biology*, **4**(11). Article 3.

McLachlan GL, Do KA, Ambroise C (2004). *Analyzing Microarray Gene Expression Data*. Wiley, New Jersy.

Qiu X, Xiao Y, Gordon A, Yakovlev A (2006). "Assessing Stability of Gene Selection in Microarray Data Analysis." *BMC Bioinformatics*, **7**. Article 50.

Simon RM, Korn EL, McShane LM, Radmacher MD, Wright GW, Zhao Y (2003). *Design and Analysis of DNA Microarray Investigations*. Springer, New York.

Speed TP (ed.) (2003). *Statistical Analysis of Gene Expression Microarray Data*. Chapman & Hall/CRC, Boca Raton, Florida.

Stamey TA, Warrington JA, Caldwell MC, Chen Z, Fan Z, Mahadevappa M, McNeal JE, Nolley R, Zhang Z (2001). "Molecular Genetic Profiling of Gleason Grade 4/5 Prostate Cancers Compared to Benign Prostatic Hyperplasia." *The Journal of Urology*, **166**, 2171–2177.

Troyanskaya OG, Garber ME, Brown PO, Botstein D, Altman RB (2002). "Nonparametric Methods for Identifying Differentially Expressed Genes in Microarray Data." *Bioinformatics*, **18**, 1454–1461.

van de Wiel MA (2001). "The Split-up Algorithm: A Fast Symbolic Method for Computing *P*-Values of Rank Statisitcs." *Computational Statistics*, **16**, 519–538.

von Mises R (1931). *Wahrscheinlichkeitsrechnung und ihre Anwendung in der Statistik und Theoretischen Physik*. Deuticke, Leipzig.

Wit E, MacClure J (2004). *Statistics for Microarrays*. Wiley, Chichester.

Xiao Y, Frisina R, Gordon A, Klebanov L, Yakovlev A (2004). "Multivariate Search for Differentially Expressed Gene Combinations." *BMC Bioinformatics*, **5**. Article 164.

Zajta AJ, Pandikow W (1977). "A Table of Selected Percentiles for the Cramér-von Mises-Lehmann Test: Equal Sample Sizes." *Biometrika*, **64**(1), 165–167.

Affiliation:

Yuanhui Xiao
Department of Biostatistics and Computational Biology
University of Rochester, and
Department of Mathematics and Statistics
Georgia State University
30 Pryor Street
Atlanta, GA 30303, United States of America
E-mail: matyxx@langate.gsu.edu

Alexander Gordon
Department of Biostatistics and Computational Biology
University of Rochester, and
Department of Mathematics and Statistics
University of North Carolina at Charlotte
9201 University City Blvd
Charlotte, NC 28223, United States of America
E-mail: aygordon@uncc.edu

Andrei Yakovlev
Department of Biostatistics and Computational Biology
University of Rochester
601 Elmwood Avenue, Box 630
Rochester, NY 14642, United States of America
E-mail: andrei_yakovlev@urmc.rochester.edu

Journal of Statistical Software http://www.jstatsoft.org/
published by the American Statistical Association http://www.amstat.org/
Volume 17, Issue 8 *Submitted:* 2006-03-17
January 2007 *Accepted:* 2006-12-18