Statistical tests

Martin Lüscher

December 2010

Introduction

This note contains a short description of a set of ISO C programs for the χ^2 distribution and the Kolmogorov-Smirnov test. Further for further details see refs. [1–3] for example.

χ^2 distribution

Suppose $X_1,\ldots,X_\nu$ are independent statistical variables with mean zero and unit variance. The probability for the sum $\sum_{i=1}^{\nu} X_i^2$ to be less than χ^2 is then given by

$$P(\chi^2|\nu)=\left[2^{\frac{1}{2}\nu}\Gamma\bigl(\tfrac{1}{2}\nu\bigr)\right]^{-1}\int_0^{\chi^2}\mathrm{d}x\,x^{\frac{1}{2}\nu-1}\mathrm{e}^{-\frac{1}{2}x}.$$

It is easy to check by differentiation that

$$P(x|\nu)=\sum_{k=0}^{\infty}a_{\nu+2k},\qquad a_\mu=\left[2^{\frac{1}{2}\mu}\Gamma\bigl(\tfrac{1}{2}\mu+1\bigr)\right]^{-1}x^{\frac{1}{2}\mu}\mathrm{e}^{-\frac{1}{2}x}. \tag{1}$$

Although this series is absolutely convergent for all x, it is not suitable for numerical evaluation when ν and x are large. In this range of parameters it is better to use the representations

$$P(x|\nu)=1-\sum_{k=1}^{\frac{1}{2}\nu}a_{\nu-2k}\quad\text{if } \nu \text{ is even}, \tag{2}$$

$$P(x|\nu) = P(x|1) - \sum_{k=1}^{\frac{1}{2}(\nu-1)} a_{\nu-2k} \quad \text{if } \nu \text{ is odd.} \tag{3}$$

Specifically these formulae are employed if $\nu \geq 2$ and $x > \nu$, while in all other cases eq. (1) is used.

The terms in the sums are calculated by applying the recursion

$$(\mu+2)a_{\mu+2} = xa_\mu, \qquad \mu \geq 0,$$

either in the forward or the backward direction. To compute the first term, the approximation

$$\ln\Gamma(z) \simeq (z - \tfrac{1}{2})\ln(z) - z + \tfrac{1}{2}\ln(2\pi) + \frac{1}{12z} - \frac{1}{360z^3} + \frac{1}{1260z^5} - \frac{1}{1680z^7}$$

is used for $z \geq 20$, the relative error being less than 5×10^{-17} in this range of the argument. For half-integer values of z smaller than 20 the exact expression for $\ln\Gamma(z)$ is numerically safe.

Before the summation is started one should make sure that the parameters are in a range where the sums are greater than the desired level of precision. In this way one can avoid that over- or underflow occurs for extreme values of the arguments. The summations are stopped when all terms have been summed or if the next term satisfies

$$a_\mu \leq (1-p)\epsilon, \quad \text{where} \quad p = \begin{cases} x/(\mu+2) & \text{in eq. (1),} \\ \mu/x & \text{in eqs. (2) and (3).} \end{cases}$$

This criterion guarantees that the sum is obtained with an absolute precision better than ϵ.

Kolmogorov-Smirnov test

Let X be a statistical variable with probability distribution

$$\mathrm{Prob}(X \leq x) = F(x).$$

If $x_0, x_1, \ldots, x_{n-1}$ is a sequence of values of X which have been obtained in the course of an experiment, one may ask how well the empirical distribution

$$F_n(x) = \frac{1}{n} \times \{\text{number of values } x_i \leq x\}$$

compares with $F(x)$. A useful measure for the deviation between the distributions is provided by the Kolmogorov-Smirnov statistics

$$K_n^+ = \sqrt{n} \sup_x \{F_n(x) - F(x)\},$$

$$K_n^- = \sqrt{n} \sup_x \{F(x) - F_n(x)\}.$$

If $F(x)$ is smooth and if the observed differences are due to random fluctuations only, these variables can be shown to be distributed according to

$$\mathrm{Prob}(K_n^\pm \leq s) = 1 - \mathrm{e}^{-2s^2} \left\{1 - \tfrac{2}{3} s/\sqrt{n} + \mathrm{O}(1/n)\right\}. \tag{4}$$

By calculating $K_n^\pm$ one thus obtains an indication how probable it is that the outcome of the experiment is consistent with the expected distribution.

An interesting fact about the Kolmogorov-Smirnov test is that the array

$$f_0, f_1, \ldots, f_{n-1}, \qquad f_i = F(x_i), \tag{5}$$

is all what is needed to compute $K_n^\pm$. Traditionally a sorting routine is used in this calculation, but one can avoid this by noting that [3]

$$K_n^+ = \sqrt{n} \max_{k=0,\ldots,n;n_k>0} \left\{\frac{1}{n} \sum_{j=0}^{k} n_j - v_k\right\},$$

$$K_n^- = \sqrt{n} \max_{k=0,\ldots,n;n_k>0} \left\{u_k - \frac{1}{n} \sum_{j=0}^{k-1} n_j\right\}.$$

In these equations u_k and v_k denote the minimum and maximum of the f_i's satisfying

$$k \leq nf_i < k+1$$

and n_k is the number of these values. The computational effort required to calculate $K_n^{\pm}$ along these lines is proportional to n, because u_k, v_k and n_k can be obtained for all k in a single pass through the data array.

Programs

The programs described below (files `pchi_square.c` and `ks_test.c`) are written in ISO C and should thus be portable. On many machines one has to set an option such as `-ansi` to indicate to the compiler that the programs are not written in traditional C. To make the programs available to the calling program, an appropriate header file should be included that lists the prototypes of the externally accessible functions.

χ^2 *distribution*

The prototype of the program that calculates $P(\chi^2|\nu)$ is

```
   double pchi_square(double chi_square,int nu)
```

For $\chi^2 \geq 0$ and $1 \leq \nu \leq 1000$ the value returned by this function deviates from the true distribution by less than 10^{-8} if $\nu \geq 2$ and less than 10^{-9} if $\nu = 1$. When the parameters are not in this range, the program exits with an error message.

Kolmogorov-Smirnov test

The program for the Kolmogorov-Smirnov statistics is

```
   void ks_test(int n,double f[],double *pkp,double *pkm)
```

where `pkp` and `pkm` point to K_n^+ and K_n^- after execution. The program checks that `n` is greater than 0 and that the array values `f[0],f[1],...,f[n-1]` are contained in $[0,1]$.

For large n the expression on the right-hand side of eq. (4) provides an accurate approximation for the probability distribution of the Kolmogorov-Smirnov statistics. The function

```
   void ks_prob(int n,double kp,double km,double *pp,double *pm)
```

computes these probabilities for given values of n, K_n^+ and K_n^- (first three arguments). After execution `pp` and `pm` point to the associated probabilities.

References

[1] M. Abramowitz and I. A. Stegun, Handbook of mathematical functions (Dover Publications, New York, 1965)

[2] R. J. Barlow, Statistics (John Wiley & Sons, Chichester, 1989)

[3] D. E. Knuth, Semi-Numerical Algorithms, *in*: The Art of Computer Programming, vol. 2 (Addison-Wesley, Reading MA, 1981)