Evaluating the Anderson-Darling Distribution

George Marsaglia*
The Florida State University
John C. W. Marsaglia
Western Oregon University

Abstract

Except for $n=1$, only the limit as $n\to\infty$ for the distribution of the Anderson-Darling test for uniformity has been found, and that in so complicated a form that published values for a few percentiles had to be determined by numerical integration, saddlepoint or other approximation methods. We give here our method for evaluating that asymptotic distribution to great accuracy—directly, via series with two-term recursions. We also give, for any particular n, a procedure for evaluating the distribution to the fourth digit, based on empirical CDF's from samples of size 10^{10}.

1 Introduction

For an ordered set $x_1 < x_2 < \cdots < x_n$ of purported uniform [0,1) variates, the Anderson-Darling goodness-of-fit test uses the statistic

$$A_n = -n - \frac{1}{n}[\ln(x_1(1-x_n)) + 3\ln(x_2(1-x_{n-1})) + 5\ln(x_3(1-x_{n-2})) + \cdots + (2n-1)\ln(x_n(1-x_1))].$$

This is a special case of the Cramer-von Mises approach: use $\int_0^1 w(x)[F_n(x)–x]^2\,dx$, the squared area between the sample CDF $F_n(x)$ (the staircase) and the diagonal $y=x$, using a suitable weight function $w(x)$. That approach: use the weighted square of the area, is in contrast to the Kolmogorov approach: use the maximum distance from the staircase to the diagonal, (for which the distribution has only recently become available [6]). Choice of the weight function $w(x) = \frac{1}{x(1-x)}$ was suggested by L.J. Savage, as it divides the distances from the diagonal to the corners of the staircase by their standard deviations. That choice leads to the statistic A_n above, for which Anderson and Darling [1] derived a complicated expression for the asymptotic distribution.

Thus, with $F_n(x)$ the sample CDF of a set $\{X_1, ..., X_n\}$ of iid random variables in [0,1),

$$F_n(x) = \frac{\text{number of } X_1, X_2, \ldots, X_n \text{ that are } \le x}{n},$$

the Anderson-Darling statistic for testing that the X's came from a uniform distribution is

$$A_n = n\int_0^1 \frac{[F_n(x)-x]^2}{x(1-x)}dx = n\int_0^1 \frac{[F_n(x)-x]^2}{x}dx + n\int_0^1 \frac{[F_n(x)-x]^2}{1-x}dx.$$

Since $F_n(x)$ is a step function, expressing the above integral as a sum in two parts leads to a collection of elementary integrals from which a little manipulation provides the form

$$A_n = -n - \frac{1}{n}[\ln(x_1(1-x_n)) + 3\ln(x_2(1-x_{n-1})) + 5\ln(x_3(1-x_{n-2})) + \cdots + (2n-1)\ln(x_n(1-x_1))],$$

in which $x_1 < x_2 < \cdots < x_n$ are the n sample X values put into increasing order.

For $n=1$ the distribution of A_1 is

$$\Pr[A_1 < z] = \Pr[-1 - \ln(x_1(1-x_1)) < z] = \sqrt{1-4e^{-1-z}},\ \text{for } z > \ln(4) - 1 = .38629\cdots.$$

Even for $n=2$ the distribution is difficult to evaluate (via numerical integration), and for specific $n>2$, all that seems available is some tabled values for $n \le 8$ based on simulations by Lewis [4] —simulations limited by CPU speeds and sampling methods circa 1960. But current speeds of CPU's, and fast methods for generating an ordered sample of uniform variates by means of the ziggurat method for exponential variates, make it feasible to generate 10^{10} random values of A_n for n's in the hundreds, and thus determine the distribution with considerable accuracy. We do this for $n = 8, 16, 32, 64, 128$ and the results lead to formulas

*Professor Emeritus

for in-between n's that seem to provide $\Pr(A_n < z)$ with accuracies to the fifth digit, inferred from the size of the samples used to derive them and supported by extensive testing.

The limiting distribution of A_n also suffers from lack of a method for its accurate determination. Anderson,Darling [2] and Lewis [4] both give the same values for the 90,95 and 99 percentiles (their 99th percentile, 3.857 is actually 3.878125...). Other approximations are given by Sinclair and Spurr [7], while Giles [3] has recently suggested a saddlepoint method.

We provide here a method for determining that asymptotic distribution to arbitrary precision, with a C version to double precision accuracy, roughly 13-15 digits.

We first describe the method for evaluating the asymptotic distribution, $\lim_{n\to\infty} \Pr(A_n < z)$, to desired accuracy, and then show how it may be used to determine, for given n, $\Pr(A_n < z)$ by means of adjustments based on simulations of size 10^{10}.

2 The limiting distribution of A_n.

An expression for the limiting distribution of A_n was given by Anderson and Darling [1]. The method was based on a development of Doob for the absorption probability of a diffusion model. They gave

$$\lim_{n\to\infty} \Pr(A_n < z) = \frac{\sqrt{2\pi}}{z} \sum_{j=0}^{\infty} \binom{-\frac{1}{2}}{j} (4j+1)\, e^{-(4j+1)^2\pi^2/(8z)} \int_0^{\infty} e^{\frac{z}{8(1+w^2)} - w^2(4j+1)^2\pi^2/(8z)} d\, w.$$

This is a strange distribution function. Anderson and Darling [2] used numerical integration to find the 90, 95 and 99 percentiles. (They are reported as 1.933,2.492 and 3.857; the true values to 20 places are 1.9329578327415937304, 2.4923671600494096176 and 3.8781250216053948842.) Lewis [4], also using numerical integration, published a table giving $\lim \Pr(A_n < z)$ with 4-place accuracy for selected z values, as well as the same three percentiles with the wrong 3.857 value for 3.878125.... Other values have been provided by Sinclair and Spurr [6], (approximate inversion of the characterisitic function), and Giles [3], (saddlepoint approximations). In all, it seems that relatively few values or percentiles have been provided, all by approximation methods and sometimes giving less than the claimed 3-4 digits of accuracy. Note that Sinclair and Spur report a better value, 3.880 as the 99 percentile 3.878125..., which Giles disputes in a footnote, sticking to 3.857 as the 'true' value, presumably because it was given by both Anderson-Darling [2] and Lewis [4].

We will provide a method for evaluating the above distribution with accuracy limited to the computer's ability to distinguish between floating point numbers, give a C program for implementing it, and also give a quick-and-easy approximation that gives accuracy better than .000002 for probabilities less than .9 and .0000008 for those beyond.

We designate the limiting distribution by ADinf(z), and put it in the form

$$\lim_{n\to\infty} \Pr(A_n < z) = \text{ADinf}(z) = \frac{1}{z} \sum_{j=0}^{\infty} \binom{-\frac{1}{2}}{j} (4j+1) f(z,j),$$

where

$$f(z,j) = \sqrt{2\pi} e^{-t_j} \int_0^{\infty} e^{\frac{z}{8(1+w^2)} - w^2 t_j} d\, w, \qquad t_j = (4j+1)^2\pi^2/(8z).$$

The difficulty is in evaluating $f(z,j)$. To do that, we expand $\exp(\frac{z}{8(1+w^2)})$ in a series:

$$f(z,j) = c_0 + c_1\frac{z}{8} + c_2(\frac{z}{8})^2/2! + c_3(\frac{z}{8})^3/3! + c_4(\frac{z}{8})^4/4! + \cdots,$$

with, for given j, and t_j,

$$c_n = \sqrt{2\pi} e^{-t_j} \int_0^{\infty} \frac{e^{-w^2 t_j}}{(1+w^2)^n} d\, w.$$

We then have, all with fixed j and $t = t_j = (4j+1)^2\pi^2/(8z)$:

$$c_0 = \pi e^{-t} (2t)^{-1/2},$$

$$c_1 = \pi(\pi/2)^{1/2} \text{erfc}(t^{1/2}),$$

and—the key to accurate evaluation of ADinf()—the recursion:

$$c_{n+1} = \frac{(n - \frac{1}{2} - t)c_n + t c_{n-1}}{n}.$$

Using these recursions, for fixed j in the series for $f(z,j)$, we may then evaluate ADinf(z) as a series,

$$\text{ADinf}(z) = \frac{1}{z}[f(z,0) - \frac{1}{2}\frac{5}{1!}f(z,1) + \frac{1}{2}\frac{3}{2}\frac{9}{2!}f(z,2) - \frac{1}{2}\frac{3}{2}\frac{5}{2}\frac{13}{3!}f(z,3) + \frac{1}{2}\frac{3}{2}\frac{5}{2}\frac{7}{2}\frac{17}{4!}f(z,4) - \cdots]$$

The accompanying C version will evaluate ADinf(z) to around fifteen places, checked with a 30-digit Maple version. Note that for the two initial values of the recursions, the second requires erfc, the complementary error function. Our version of the complementary normal distribution function, cPhi(x), is included and recommended for use, as it is more accurate than many of the available erf(x) or erfc(x) routines in C compiler libraries.

Figure 2 shows the distribution and density of A_∞. The distribution is infinitely flat (every derivative is zero) at $z = 0$ and nearly flat as z passes 6 or so, with a slow approach to 1. The C (or Maple) procedure gives ADinf(9) = .999960465988611 (.999960465988612484992562014458), and ADinf(10) = .999986184964588 (.999986184964589314168018038088).

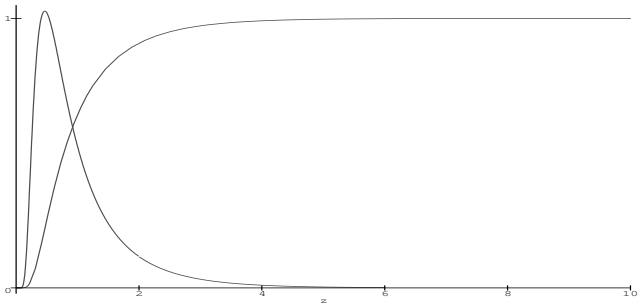

Figure 1: The density and distribution function of A_∞.

The above method for evaluating ADinf(z) to unlimited accuracy, requiring the normal integral and a loop within a loop, might be more complicated and/or have more power than many users want to invoke—certainly in applications where 6-digit accuracy may be more than adequate. For that reason we offer the following two-piece formula, with accuracies indicated, (using lowercase adinf rather than the mixed ADinf):

$$\text{adinf}(z) = \begin{cases} \text{for } \ 0 < z < 2, \text{ with } |\text{error}| < .000002: \\ z^{-1/2}e^{-1.2337141/z}(2.00012 + (0.247105 - (.0649821 - (.0347962 - (.0116720 - .00168691z)z)z)z)z)z) \\ \text{for } \ 2 \le z < \infty, \text{ with } |\text{error}| < .0000008: \\ \exp(-\exp(1.0776 - (2.30695 - (.43424 - (.082433 - (.008056 - .0003146z)z)z)z)z)). \end{cases}$$

The function ADinf(z) starts like $2z^{-1/2}e^{\pi^2/(8z)}$, and $\pi^2/8 = 1.23370055$ is changed to 1.2337141 to ensure (7-place) continutity at $z = 2$, with a fifth degree polynomial in Horner form as a multiplier for the range $0 < z \le 2$. With increasing z, we might expect a standard extreme-value form: $e^{-e^{a-bz}}$, and examination shows that $\ln(-\ln(\text{ADinf}(z)))$ looks quite linear. But a linear form in that top exponent does not provide the accuracy we seek, and a Horner polynomial to degree 5 is used.

3 Getting the distribution of A_n by simulation.

Until now, little was known about the distribution of A_n for various n. Lewis [4] gave tables for small n, based on simulations conducted at the IBM Research Center, but was unable to go beyond $n = 8$. He concluded that the convergence of the distribution of A_n to that of A_∞ is "quite rapid". Such confidence in a rapid approach seems to hold only for probabilities greater than about .8. We think that use of A_∞'s distribution for that of A_n is not good enough for modern computer-aided statistical procedures. The worst error in using A_∞ for a particular A_n is about $.044/n$, near the 33rd percentile. But there is some support for Lewis's confidence: the approach is more rapid than that for the limiting form of Kolmogorov's distribution, which has worst error of about $.278/\sqrt{n}$, also around the 33rd percentile [6].

By developing very fast ways to generate a sequence of ordered uniform variables, we will be able to get samples of A_n large enough (typically 10^{10} with each of $n = 8, 16, 32, 64, 128$) to provide accuracies better than .00005 for determining $\Pr(A_n < z)$ or for converting an observed A_n to a p-value.

To do this, we need a way to partition the possible values of A_n into cells with approximately equal probabilities. But this need not be done directly. When applying a goodness-of-fit test to the values $x_1 < x_2 < \cdots < x_n$, whatever version of the Kolmogorov-Smirnov class of tests we use, it is necessary to convert the resulting statistic to a uniform variable, a p-value, in [0,1). We do not need $\Pr(A_n < z)$ directly, but, rather, we need a way to convert the observed value to a uniform [0,1) value.

We do that in the following way: Since the distribution of A_n is close to that of A_∞, if Z is a random value with distribution of A_n, then ADinf(Z) should be close to uniformly distributed in [0,1). Therefore,

if we divide the unit interval into 1000 cells, 0 to .001, .001 to .002, ...,.999 to 1, then count the number of times that ADinf(Z) falls into those intervals. We need only make a small adjustment to make the resulting empirical distribution quite close to uniform in [0,1)—provided our mean cell counts are large enough to provide the necessary accuracy. Our mean cell counts are around 10^7.

Thus there is an error function, say errfix(n, x), $0 < x < 1$, such that

$$\Pr(A_n < z) = \text{ADinf}(z) + \text{errfix}(n, \text{ADinf}(z)).$$

A computer approximation to the true errfix(n, x) needs to have sufficient accuracy to ensure that the composite result, ADinf(z) + errfix$(n,$ ADinf$(z))$ will be close enough to uniform for practical applications.

The error function errfix$(n, x), 0 < x < 1$ given below is based on extensive simulation. For a given value of the random variable $Z = A_n$, we evaluate $x =$ ADinf(Z), then convert that x to a uniform [0,1) random variable p by means of $p = x +$ errfix(n, x). Graphs of errfix$(n, x), 0 < x < 1$ are given in Figure 3, for $n = 8, 16, 32, 64, 128$.

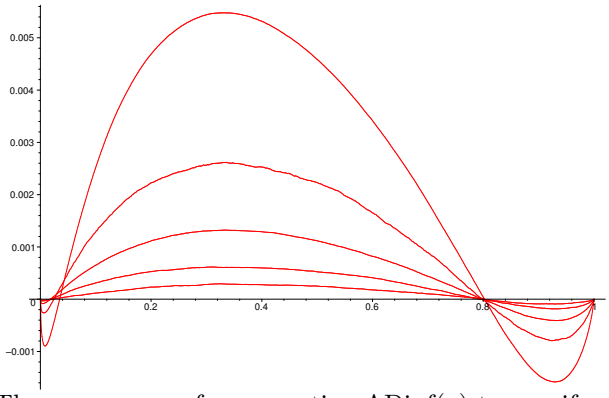

Figure 2: The error curves for converting ADinf(z) to a uniform distribution

Here, we get lucky. We are able to adequately represent, piecewise, any one of those error curves by means of three fixed functions g_1, g_2, g_3, adjusting the scale as a function of $1/n$ and providing stretch factors in the g's arguments. Each error curve is made of three parts, an initial dip that reaches the x-axis at a point $c = c(n)$ and can be adequately represented as $g_1(x/c)$ times a function of $1/n$. Then the second region from c to .8 can be adequately represented as a function of $1/n$ times $g_2(x - c)/(.8 - c)$, and finally, a last dip from .8 to 1 than can be represented as $g_3(x)/n$. For each error curve, the right crossing is satisfactorily close to $x = .8$ for all n, but the left crossing, c, varies enough to require an empirical function of n:

$$c(n) = .01265 + .1757/n.$$

With that $c(n)$, the following piecewise function serves to accurately represent, (to within $\pm.00005$), the error in converting an observed A_n into a uniform [0,1) variable, at least for specific $n = 8, 16, 32, 64, 128$:

$$\text{errfix}(n, x) = \begin{cases} (.0037/n^3 + .00078/n^2 + .00006/n)g_1(x/c(n)) & \text{if } x < c(n) \\ (.04213/n + .01365/n^2)g_2((x - c(n))/(.8 - c(n))) & \text{if } c(n) \le x < .8 \\ g_3(x)/n & \text{if } .8 < x \end{cases}$$

$$g_1(x) = \sqrt{x}(1 - x)(49x - 102),$$

$$g_2(x) = -.00022633 + (6.54034 - (14.6538 - (14.458 - (8.259 - 1.91864x)x)x)x)x,$$

$$g_3(x) = -130.2137 + (745.2337 - (1705.091 - (1950.646 - (1116.360 - 255.7844x)x)x)x)x.$$

A polynomial will not serve near the origin, but $\sqrt{x}$ times a simple polynomial works well.

For values of n between or beyond the designated 8,16,32,64,128, errfix(n,x) seems to give accuracies better than .0005. To use errfix() in applying the Anderson-Darling test: for given n and sample values $x_1 < x_2 < \cdots < x_n$, first compute

$$Z = A_n = -n - \frac{1}{n}[\ln(x_1(1 - x_n)) + 3\ln(x_2(1 - x_{n-1})) + \cdots + (2n - 1)\ln(x_n(1 - x_1))].$$

Then the random variable ADinf(Z) + errfix$(n,$ ADinf$(Z))$ should be uniformly distributed in the interval [0,1).

Alternatively, if $Z = A_n$ is the Anderson-Darling statistic arising from a (sorted) sample of n iid uniform variates in [0,1), then

$$\Pr(Z < z) = \text{ADinf}(z) + \text{errfix}(n, \text{ADinf}(z)).$$

Since the exact distribution of $Z = A_n$ is not known, but only approximated from samples of 10^{10} for various n, we base our claim on:

The 1000 times 10,000 Test: For various n, do this:
A: Generate a sample of 10,000 Z's, $Z = -n - \frac{1}{n}[\ln(x_1(1-x_n)) + \cdots + (2n-1)\ln(x_n(1-x_1))]$, converting each to values in [0,1) by $\mathrm{ADinf}(Z) + \mathrm{errfix}(n, \mathrm{ADinf}(Z))$ as above.
B: Apply a KS test (the Kolmogorov test [6]) to the resulting 10000 'uniform' values.
This will return a value in [0,1).
C: Repeat Steps **A** and **B** 1000 times to yield 1000 values in [0,1).
If those 1000 values, in turn, pass the Kolmogorov test,
then the error conversion formula may be considered suitable for practical purposes.
We found the error conversion procedure passed the '1000 times 10,000 test' for each of $n = 10, 20, 30, \ldots, 100$—indeed, numerous times for each of those n's and others.

4 Speeding up the simulation.

We describe here the method for generating an ordered set of n uniform [0,1) variates that makes feasible the generation of some 10^{10} ordered sets. These are used to find the distribution of A_n, or rather, find the function of A_n that converts it to a p-value for testing goodness-of-fit under the Anderson-Darling criterion.

The key to speed is the ability to generate exponential variates at the rate of 50 to 60 million per second. This can be done using the ziggurat method of Marsaglia and Tsang [5], with the #define feature of C that permits the fast part of the generating procedure to be done in-line. Given such a fast method for generating exponential variates, the required set of ordered uniform variates can be generated as

$$x_1 = y_1/S, x_2 = (y_1 + y_2)/S, \ldots, x_n = (y_1 + \cdots + y_n)/S,$$

with $y_1, y_2, \ldots, y_n, y_{n+1}$ exponential variates and $S = y_1 + \cdots + y_{n+1}$.

With such a fast method for generating each ordered set of uniform variates, we were able to get samples of size 10^{10} to find, empirically, the error adjustment that makes $AD(Z) + \mathrm{errfix}(n, AD(Z))$ as close to being uniform in [0,1) as practical applications are likely to require.
Summary for AD test: Generate a sample value Z of A_n. Then return $\mathrm{ADinf}(Z) + \mathrm{errfix}(n, \mathrm{ADinf}(Z))$ as the required p-value, uniformly distributed in [0,1) if the ordered set $x_1 < x_2 < \cdots < x_n$ came from a sample of n iid uniform's, or, alternatively, $\Pr(A_n < z) = \mathrm{ADinf}(z) + \mathrm{errfix}(n, \mathrm{ADinf}(z))$.

5 Attachments

The **browse files** section for this article contains files ADinf.c and AnDarl.c. The first file contains `ADinf(double z)` for the asymptotic distribution to full accuracy, with a main program for calling it.
The second file, AnDarl.c, provides `AD(int n,double z)` for finding $\Pr(A_n < z)$. It finds adinf(z), then adjusts that result, returning adinf(z)+errfix(n,adinf(z)). The quick-and-easy adinf() is used, rather than the full-precision `ADinf()`, since precision beyond 6-7 digits is likely to be wasted for the 4-5 digit accuracy of the code in errfix(n,x). (If you want greater accuracy, try the more elaborate ADinf(z), but the effort is likely to be wasted.)
The file AnDarl.c also contains the procedure `ADtest(int n,double *u)` that computes the Anderson-Darling statistic $Z = A_n$ from the (sorted) u array, then converts to a p-value by means of `AD(n,Z)`.
The main() program for the AD() and ADtest() routines contains two sorted arrays of size 10, to illustrate use of ADtest, and an infinite loop for arguments to AD(n,z), along with the adinf(z) and errfix(n,adinf(z)) that lead to the result.

References

[1] T. W. Anderson and D. A. Darling, (1952), Asymptotic theory of certain 'goodness-of-fit' criteria based on stochastic processes, *Ann. Math. Stat.,* **23** 193-212.

[2] T. W. Anderson and D. A. Darling, (1954), A test of goodness of fit, *J. Amer. Stat. Assn.*, **49** 765–769.

[3] D. E. A. Giles, (2000), A Saddlepoint Approximation to the Distribution Function of the Anderson-Darling Test Statistic, http://ideas.repec.org/p/vic/vicewp/0005.html

[4] P. A. Lewis, (1961), Distribution of the Anderson-Darling Statistic, *Ann. Math. Stat.*, **32** 1118–1124.

[5] G. Marsaglia and Wai Wan Tsang, (2000), The ziggurat method for generating random variables, *Journal Statistical Software*, **5**, Issue 8.

[6] G. Marsaglia, Wai Wan Tsang and Jingbo Wang, (2003), Evaluating Kolmogorov's distribution, *Journal Statistical Software* **8**, Issue 18.

[7] C. D. Sinclair and B. D. Spurr, (1988), Approximations to the Distribution Function of the Anderson-Darling Test Statistic, *Journal American Statistical Association*, **83** 1190–1191.