Dip Test Distributions, P-values, and other Explorations

Martin Mächler
ETH Zurich

Abstract

...
...

Keywords: MPFR, Abitrary Precision, Multiple Precision Floating-Point, R.

1. Introduction

FIXME: Need notation

D_n :=`dip( runif(n) );`

but more generally,

$$D_n(F) := D(X_1, X_2, \ldots, X_n), \quad \texttt{where } X_i \text{ i.i.d. } , X_i \sim F. \tag{1}$$

Hartigan and Hartigan (1985) in their "seminal" paper on the dip statistic D_n already proved that $\sqrt{n}\, D_n$ converges in distribution, i.e.,

$$\lim_{n \to \infty} \sqrt{n}\, D_n \overset{\mathcal{D}}{=} D_\infty. \tag{2}$$

A considerable part of this paper is devoted to explore the distribution of D_∞.

2. History of the `diptest` **R** package

Hartigan (1985) published an implementation in Fortran of a concrete algorithm, where the code was also made available on Statlib[1]

On July 28, 1994, Dario Ringach, then at NY University, asked on Snews (the mailing list for S and S-plus users) about distributions and was helped by me and then about `dyn.load` problems, again helped by me. Subsequently he provided me with S-plus code which interfaced to (a `f2c`ed version of) Hartigan's Fortran code, for computing the dip statistic. and ended the (then private) e-mail with

[1]Statlib is now a website, of course, `http://lib.stat.cmu.edu/`, but then was *the* preferred way for distributing algorithms for statistical computing, available years before the existence of the WWW, and entailing e-mail and (anonymous) FTP

> I am not going to have time to set this up for submission to StatLib. If you want to do it, please go ahead.
>
> Regards, Dario

- several important bug fixes; last one Oct./Nov. 2003

However, the Fortran code file http://lib.stat.cmu.edu/apstat/217, was last changed Thu 04 Aug 2005 03:43:28 PM CEST.

We have some results of the dip.dist of *before* the bug fix; notably the "dip of the dip" probabilities have changed considerably!!

- see rcs log of ../../src/dip.c

3. 21st Century Improvement of Hartigan[2]'s Table

((

Use listing package (or so to more or less "cut & paste" the nice code in `../../stuff/new-simul.Rout-1e6`

))

4. The Dip in the Dip's Distribution

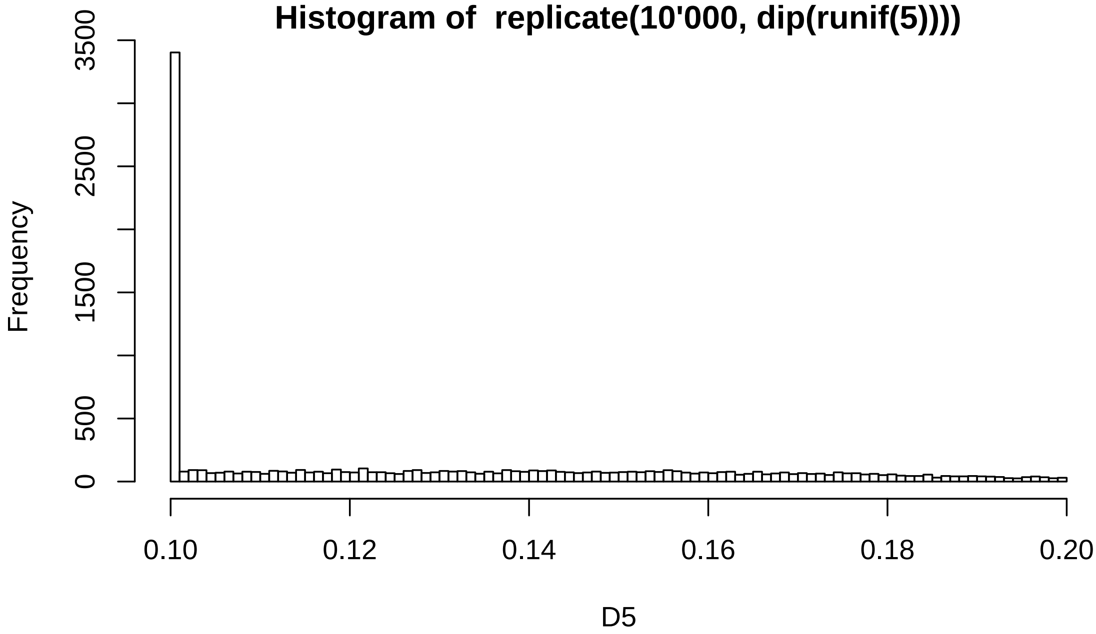
We have found empirically that the dip distribution itself starts with a "dip". Specifically, the minimal possible value of D_n is $\frac{1}{2n}$ *and* the probability of reaching that value,

$$\mathrm{P}\left[D_n = \frac{1}{2n}\right], \tag{3}$$

is large for small n.

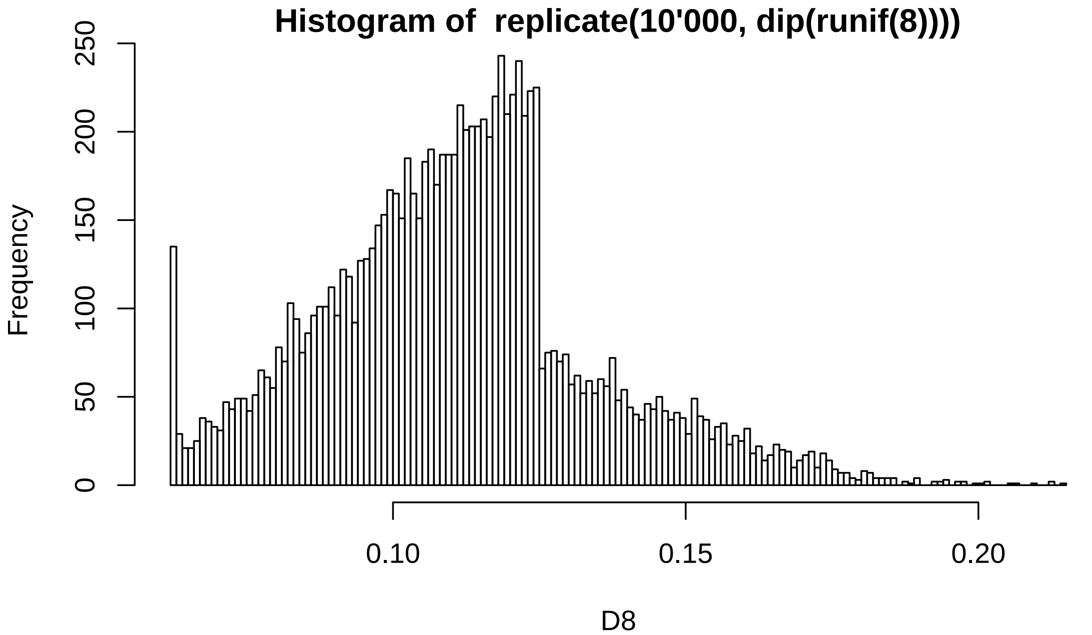
E.g., consider an approximation of the dip distribution for $n = 5$,

```
R> require("diptest") # after installing it ..
R> D5 <- replicate(10000, dip(runif(5)))
R> hist(D5, breaks=128, main = "Histogram of  replicate(10'000, dip(runif(5))))")
```



which looks as if there was a bug in the software — but that look is misleading! Note how the phenomenon is still visible for $n = 8$,

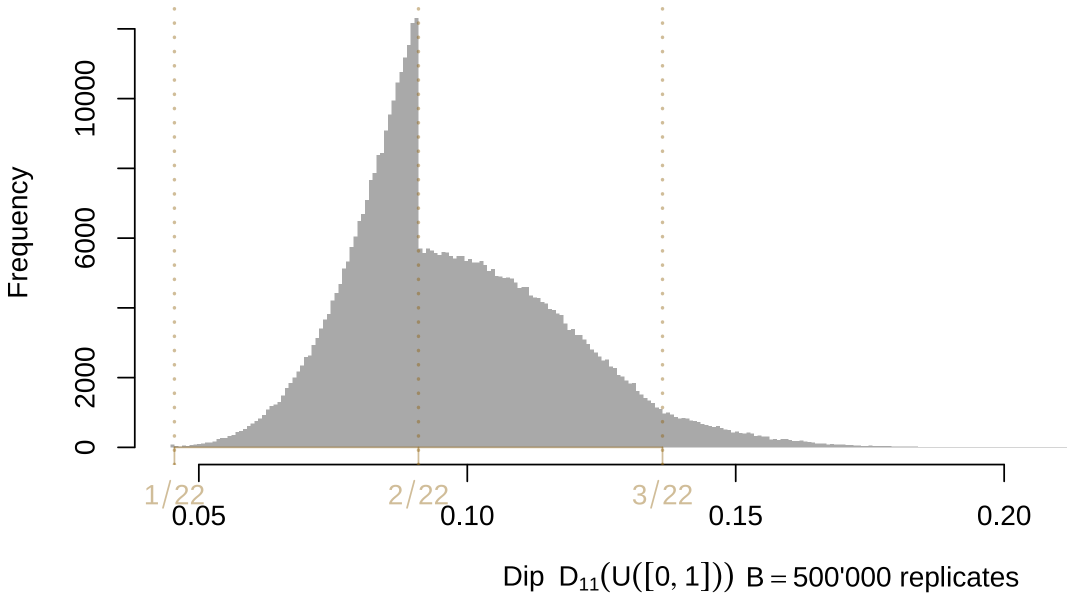
```
R> D8 <- replicate(10000, dip(runif(8)))
R> hist(D8, breaks=128, main = "Histogram of  replicate(10'000, dip(runif(8))))")
```



Note that there is another phenomenon, in addition to the point mass at $1/(2n)$, particularly visible, if we use *many* replicates,

```
R>  set.seed(11)
R>  n <- 11
```

```
R>  B.s11 <- 500000
R>  D11 <- replicate(B.s11, dip(runif(n)))
```



FIXME:
use '../../stuff/sim-minProb.R'
and '../../stuff/minProb-anal.R'

Further, it can be seen that the *maximal* dip statistic is $\frac{1}{4} = 0.25$ and this upper bound can be reached simply (for even n) using the the data $(0, 0, \ldots, 0,\ 1, 1, \ldots, 1)$, a bi-point mass with equal mass at both points.

5. P-values for the Dip Test

Note that it is not obvious how to compute P-values for "the dip test", as that means determining the distribution of the test statistic, i.e., D_n under the null hypothesis, but a natural null, $H_o : F \in \{F cadlag \mid f := \frac{d}{dx} F is unimodal\}$ is too large. Hartigans'(1985) argued for using the uniform $U[0, 1]$ i.e., $F'(x) = f(x) = \mathbf{1}_{[0,1]}(x) = [0 \le x \le 1]$ (Iverson bracket) instead, even though they showed that it is not quite the "least favorable" one. Following Hartigans', we will define the P-value of an observed d_n as

$$P_{d_n} := \mathrm{P}\left[D_n \ge d_n\right] := \mathrm{P}\left[\mathrm{dip}(U_1, \ldots, U_n) \ge d_n\right], \quad \text{where } U_i \sim U[0, 1], \quad \text{i.i.d.} \tag{4}$$

5.1. Interpolating the Dip Table

Because of the asymptotic distribution, $\lim_{n\to\infty} \sqrt{n}\, D_n \overset{\mathcal{D}}{=} D_\infty$, it is makes sense to consider the "$\sqrt{n} D_n$"–scale, even for finite n values:

```
R> data(qDiptab)
R> dnqd <- dimnames(qDiptab)
R> (nn. <- as.integer(dnqd[["n"]]))
```

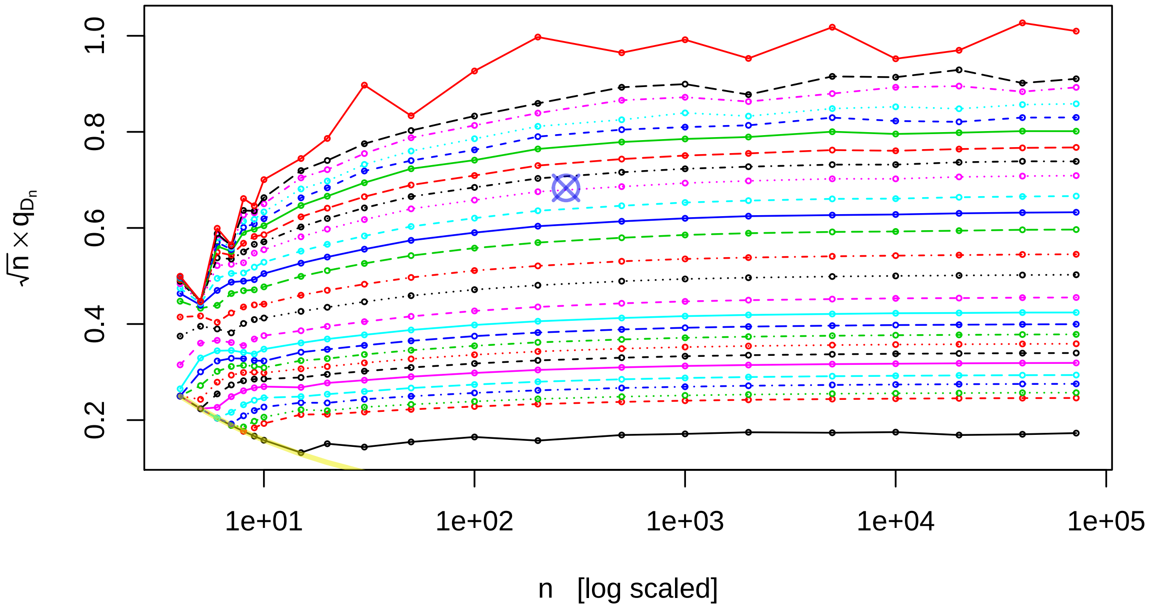
```
 [1]     4     5     6     7     8     9    10    15    20    30    50
[12]   100   200   500  1000  2000  5000 10000 20000 40000 72000
```

```
R> matplot(nn., qDiptab*sqrt(nn.), type ="o", pch=1, cex = 0.4,
          log="x", xlab="n   [log scaled]",
          ylab = expression(sqrt(n) %*% q[D[n]]))
R> ## Note that  1/2n  is the first possible value (with finite mass),,
R> ## clearly visible for (very) small n:
R> lines(nn., sqrt(nn.)/(2*nn.), col=adjustcolor("yellow2",0.5), lwd=3)
R> P.p <- as.numeric(print(noquote(dnqd[["Pr"]])))
```

```
 [1] 0        0.01     0.02     0.05     0.1      0.2      0.3      0.4
 [9] 0.5      0.6      0.7      0.8      0.9      0.95     0.98     0.99
[17] 0.995    0.998    0.999    0.9995   0.9998   0.9999   0.99995  0.99998
[25] 0.99999  1
```

```
R> ## Now look at one well known data set:
R> D <- dip(x <- faithful$waiting)
R> n <- length(x)
R> points(n, sqrt(n)*D, pch=13, cex=2, col= adjustcolor("blue2",.5), lwd=2)
R> ## a simulated (approximate) P-value for D  is
R> mean(D <= replicate(10000, dip(runif(n)))) ## ~ 0.002
```

```
[1] 0.0021
```



but we can use our table to compute a deterministic (but still approximate, as the table is from simulation too) P-value:

```
R> ## We are in this interval:
R> n0 <- nn.[i.n <- findInterval(n, nn.)]
R> n1 <- nn.[i.n +1] ; c(n0, n1)
```

```
[1] 200 500
```

```
R> f.n <- (n - n0)/(n1 - n0)# in [0, 1]
R> ## Now "find" y-interval:
```

```
R> y.0 <- sqrt(n0)* qDiptab[i.n  ,]
R> y.1 <- sqrt(n1)* qDiptab[i.n+1,]
R> (Pval <- 1 - approx(y.0 + f.n*(y.1 - y.0),
                      P.p,
                      xout = sqrt(n) * D)[["y"]])
```

```
[1] 0.001809527
```

```
R> ## 0.018095
```

Finally, in May 2011, after several year of people asking for it, I have implemented a `dip.test` function which makes use of a — somewhat more sophisticated — interpolation scheme like the one above, to compute a P-value. As `qDiptab` has been based on 10^6 samples, the interpolation yields accurate P-values, unless in very extreme cases. Here is the small ($n = 63$) example from Hartigan[2],

```
R> data(statfaculty)
R> dip.test(statfaculty)
```

```
        Hartigans' dip test for unimodality

data:  statfaculty
D = 0.0595, p-value = 0.08672
alternative hypothesis: non-unimodal, i.e., at least bimodal
```

where, from a P-value of 8.7%, we'd conclude that there is not enough evidence against unimodality.

5.2. Asymptotic Dip Distribution

We have conducted extensive simulations in order to explore the limit distribution of D_∞, i.e., the limit of $\sqrt{n}\,D_n$, (2).

Our current R code is in ' `../../stuff/asymp-distrib.R` ' but the simulation results (7 Megabytes for each n) cannot be assumed to be part of the package, nor maybe even to be simply accessible via the internet.

6. Less Conservative Dip Testing

7. Session Info

```
R> toLatex(sessionInfo())
```

- R version 2.15.1 Patched (2012-08-11 r60235), `x86_64-unknown-linux-gnu`
- Locale: `LC_CTYPE=de_CH.UTF-8`, `LC_NUMERIC=C`, `LC_TIME=en_US.UTF-8`, `LC_COLLATE=C`, `LC_MONETARY=en_US.UTF-8`, `LC_MESSAGES=C`, `LC_PAPER=C`, `LC_NAME=C`, `LC_ADDRESS=C`, `LC_TELEPHONE=C`, `LC_MEASUREMENT=de_CH.UTF-8`, `LC_IDENTIFICATION=C`
- Base packages: base, datasets, grDevices, graphics, methods, stats, utils

- Other packages: diptest 0.75-4
- Loaded via a namespace (and not attached): tools 2.15.1

References

Hartigan JA, Hartigan PM (1985). "The Dip Test of Unimodality." *Annals of Statistics*, **13**, 70–84.

Hartigan PM (1985). "Computation of the Dip Statistic to Test for Unimodality." *Applied Statistics*, **34**, 320–325.

Affiliation:

Martin Mächler
Seminar für Statistik, HG G 16
ETH Zurich
8092 Zurich, Switzerland
E-mail: maechler@stat.math.ethz.ch
URL: http://stat.ethz.ch/people/maechler