Stat 5421 Lecture Notes: Fuzzy P-Values and Confidence Intervals

Charles J. Geyer

October 30, 2021

1 License

2 Discreteness versus Hypothesis Tests

You cannot do an exact level α test for arbitrary α when the data are discrete. For example, consider a lower-tailed test for binomial data with null hypothesis $H_0 : \pi = \pi_0$ and sample size n given by

```
n <- 20
pi0 <- 0.3
```

The P-value is $\Pr_{\pi_0}(X \le x)$, and the only P-values that can occur that are less than 0.2 are

```
x <- 0:n
p <- pbinom(x, n, pi0)
foo <- cbind(x, p)
colnames(foo) <- c("data", "P-value")
rownames(foo) <- rep("", nrow(foo))
round(foo[p < 0.2, ], 4)
```

```
##  data P-value
##     0  0.0008
##     1  0.0076
##     2  0.0355
##     3  0.1071
```

This behavior clearly has nothing to do with the particular values chosen for n and π_0. The code will always produce a finite set of possible P-values. This behavior also clearly has nothing to do with the binomial distribution. Any discrete distribution will do the same.

Strictly speaking, a hypothesis test like this should not be called "exact" but rather only "conservative-exact". An "exact" test should have the property that the P-value is uniformly distributed on $(0, 1)$ when the null hypothesis is correct, that is, we have

$$\Pr_{\theta_0}(P \le \alpha) = \alpha, \qquad 0 < \alpha < 1,$$

where P is the random variable that is the P-value. Exact tests whose test statistics are continuous random variables (t tests, F tests) have this property. Tests whose test statistics are discrete random variables cannot have this property, as we saw above. They only have the property

$$\Pr_{\theta_0}(P \le \alpha) \le \alpha, \qquad 0 < \alpha < 1.$$

The difference between these properties means that a so-called "exact" (but better called "conservative-exact") hypothesis test with a discretely distributed test statistic is not really analogous to an exact test

with a continuously distributed test statistic. The language we use to talk about them may mislead us into thinking they are analogous, but they aren't.

If you do the test in the example and observe $x = 3$ and report the P-value $P = 0.1071$ with no indication that the next possible lower P-value is 0.0355, then that is highly misleading. Yes, the reported P-value is far above 0.05 (if that is the standard for significance you are using), but it is as close to 0.05 as it could be without being below 0.05. And that is something that merely reporting $P = 0.1071$ does not even hint at.

3 Randomized Hypothesis Tests

The standard theory of hypothesis testing taught in all PhD-level theory classes (this can be found in Chapters 3 and 4 of the textbook *Testing Statistical Hypotheses* by Lehmann and Romano) fixes up this defect of hypothesis tests for discrete data by allowing *randomized tests*. The test does not deterministically map data values to decisions (accept or reject the null hypothesis). Rather for each data value x it rejects the null with probability $\phi(x)$ and accepts with probability $1 - \phi(x)$, and ϕ is chosen so that

$$\Pr_{\theta_0}(\text{reject}) = E_{\theta_0}\{\phi(x)\} = \alpha, \qquad 0 < \alpha < 1. \tag{1}$$

and thus the test rejects the null with probability α for all α just like hypothesis tests with continuous test statistics.

Then there is a lot of elaborate theory surrounding the Neyman-Pearson lemma that shows that it is possible to choose ϕ so that the test is *uniformly most powerful* (UMP), that is, the graph of the power function of the UMP test lies on or above the graph of the power function of any other test with the same level. In short, the UMP (randomized) test is provably better than any other test!

The ϕ function for the UMP lower-tailed test at level α has the obvious form

$$\phi(x) = \begin{cases} 1, & x < x^* \\ p^*, & x = x^* \\ 0, & x > x^* \end{cases}$$

where x^* is the largest x value such that

$$\Pr_{\theta_0}(X < x) < \alpha$$

and

$$p^* = \frac{\alpha - \Pr_{\theta_0}(X < x^*)}{\Pr_{\theta_0}(X = x^*)}. \tag{2}$$

For example, if we want to conduct a level 0.05 level lower-tailed UMP test for the situation above we want

```
alpha <- 0.05
xstar <- qbinom(0.05, n, pi0) - 1
xstar
```

```
## [1] 2
```

```
pbinom(xstar, n, pi0)
```

```
## [1] 0.03548313
```

```
pstar <- (alpha - pbinom(xstar, n, pi0)) / dbinom(xstar, n, pi0)
pstar
```

```
## [1] 0.5213292
```

so the UMP (randomized) lower-tailed test rejects the null with probability one if the observed data is less than $x^* = 2$ and with probability $p^* = 0.5213$ if the observed data is equal to $x^* = 2$ and otherwise accepts the null.

This is beautiful theory. The existence of UMP (randomized) tests is an important theoretical result. As mentioned above, it is taught as the standard theory of hypothesis testing to all PhD students in statistics.

But it is weird from an applied statistics point of view. Because of the artificial randomization, if you and I both do the UMP (randomized) hypothesis test of the same hypotheses for the same distribution and the same data, we may get different decisions for the same data. If we observe $X = x^*$, then we have to randomize. So I generate a Uniform$(0, 1)$ random variable and say "reject the null" if it is less than p^* given by the formula above, and you do the same. If we each generate a different Uniform$(0, 1)$ random variable, then we can get different results, even though the data are the same.

Because of this weirdness, every PhD statistician learns this theory in theory class and then never uses it for a real data analysis!

4 Fuzzy P-Values

Geyer and Meeden (2005, DOI:10.1214/088342305000000340, herein after just referred to as Geyer and Meeden) propose to keep the theory but jettison the weirdness by the simple device of not actually doing the randomization but only describing it. If we are both doing the UMP (randomized) test and observe data x^*, then we both say the UMP test rejects the null with probability p^* given by the formula above and leave it at that.

We also know that for many reasons the modern tendency is to report P-values rather than decisions. So Geyer and Meeden figured out what the corresponding P-value notion is. They call it a *fuzzy* P-value, something that is spread out rather than a single number.

For the lower-tailed UMP (randomized) test for the binomial distribution (or other distributions satisfying the conditions for a UMP test to exist) the fuzzy P-value is uniformly distributed on the interval from $\Pr_{\theta_0}(X < x)$ to $\Pr_{\theta_0}(X \le x)$. If one were to actually generate such a uniformly distributed random variable U and then say reject the null at level alpha when $U < \alpha$, this would be the UMP (randomized) test. But Geyer and Meeden say you shouldn't generate such a U. Instead you should just report that the the fuzzy P-value is uniformly distributed on the interval from $\Pr_{\theta_0}(X < x)$ to $\Pr_{\theta_0}(X \le x)$.

For example, if we want to report a fuzzy P-value for the lower-tailed UMP test for the situation above it is uniformly distributed on the interval $(0.0355, 0.1071)$. This is what is really analogous to an exact (not just conservative-exact) test like a t test.

The conservative-exact P-value is the upper end point of the fuzzy P-value. The fuzzy P-value makes precise how conservative the conservative-exact P-value is. The fuzzy P-value is exact-exact in the sense that if P is a random variable having the (uniform) distribution of the fuzzy P-value, then

$$\Pr(P \le \alpha) = \alpha, \qquad \text{for all } \alpha$$

which is just another way of stating the exactness property a hypothesis test is supposed to have, what we said above in other notation.

Of course, for an upper-tailed UMP test, the fuzzy P-value is uniformly distributed on the interval from $\Pr_{\theta_0}(X > x)$ to $\Pr_{\theta_0}(X \ge x)$, which is just the same as the formulas for the lower-tailed test but with the inequalities reversed.

All of this theory applies to hypothesis tests with continuous test statistics but doesn't do anything unconventional for them. A UMP (randomized) test with a continuous test statistic isn't actually randomized because any point occurs with probability zero, hence x^* occurs with probability zero. The formula (2) gives $0/0$, which is undefined, in that case, but it doesn't matter because we land in that case with probability zero. The corresponding fuzzy P-value isn't actually fuzzy because $\Pr_{\theta_0}(X < x) = \Pr_{\theta_0}(X \le x)$, and the interval over which the fuzzy P-value is distributed is degenerate, collapsed to a single point.

So when the test statistic has a continuous distribution randomized and fuzzy tests produce nothing new. They are only interesting for discrete data.

5 Two-Tailed Tests

There are no UMP (randomized) two-tailed tests. There are no tests that are uniformly (in the true unknown parameter value) better than all other tests. But if we add a side condition, then there are. The extra condition is that the test be *unbiased*, which means the power is always greater than the significance level (the probability of rejecting the null hypothesis H_0 is always greater when H_0 is false than when H_0 is true). This is a reasonable criterion for a test to satisfy.

Then there is a lot of elaborate theory surrounding the Neyman-Pearson lemma that shows that it is possible to choose ϕ so that the test is *uniformly most powerful unbiased* (UMPU), that is, the graph of the power function of the UMPU test lies on or above the graph of the power function of any other unbiased test with the same level. In short, the UMPU (randomized) test is provably better than any other unbiased test!

We won't even describe the UMPU test but just show how to calculate its fuzzy P-value (see Geyer and Meeden for a thorough explanation of UMPU theory). The fuzzy P-value for the UMPU (randomized) two-tailed test is sometimes uniform on an interval and sometimes non-uniform, but its PDF is always a step function. The R function `arpv.binom` in CRAN package `ump` does fuzzy two-tailed tests for the binomial distribution.

For example, if we want to report a fuzzy P-value for the two-tailed UMP test for data

```
n <- 20
x <- 2
pi0 <- 1 / 3
```

```
library(ump)
print(arpv.binom(x, n, pi0, plot = FALSE))
```

```
## $alpha
## [1] 0.006145670 0.008236254 0.028054957 0.033270042
##
## $phi
## [1] 3.312001e-17 8.426339e-02 8.242187e-01 1.000000e+00
```

What this returns is a specification of the cumulative distribution function of the (random variable that is) the fuzzy P-value. Since the function is piecewise linear, it just reports the "knots" which separate the pieces. If we do not say `plot = FALSE` we get the plot

```
arpv.binom(x, n, pi0)
```

With a little more effort it is possible to plot the probability density function (PDF) rather than the cumulative density function (CDF). Your humble author prefers PDF; some of his co-authors prefer CDF. Use whichever you like.

Here is how to plot the PDF.

```
foo <- arpv.binom(x, n, pi0, plot = FALSE)
arpv.plot(foo$alpha, foo$phi, df = FALSE)
```

6 Interpretation of Fuzzy P-Values.

Geyer and Meeden claim that fuzzy P-values are no harder to interpret than conventional P-values. Very low P-values are still strong evidence against the null hypothesis. Very large P-values are still no evidence against the null hypothesis. In between P-values are still in between.

Despite people's desires for definite answers, P-values don't give definite answers. I once wrote

> Anyone who thinks there is an important difference between $P = 0.049$ and $P = 0.051$ understands neither science nor statistics.

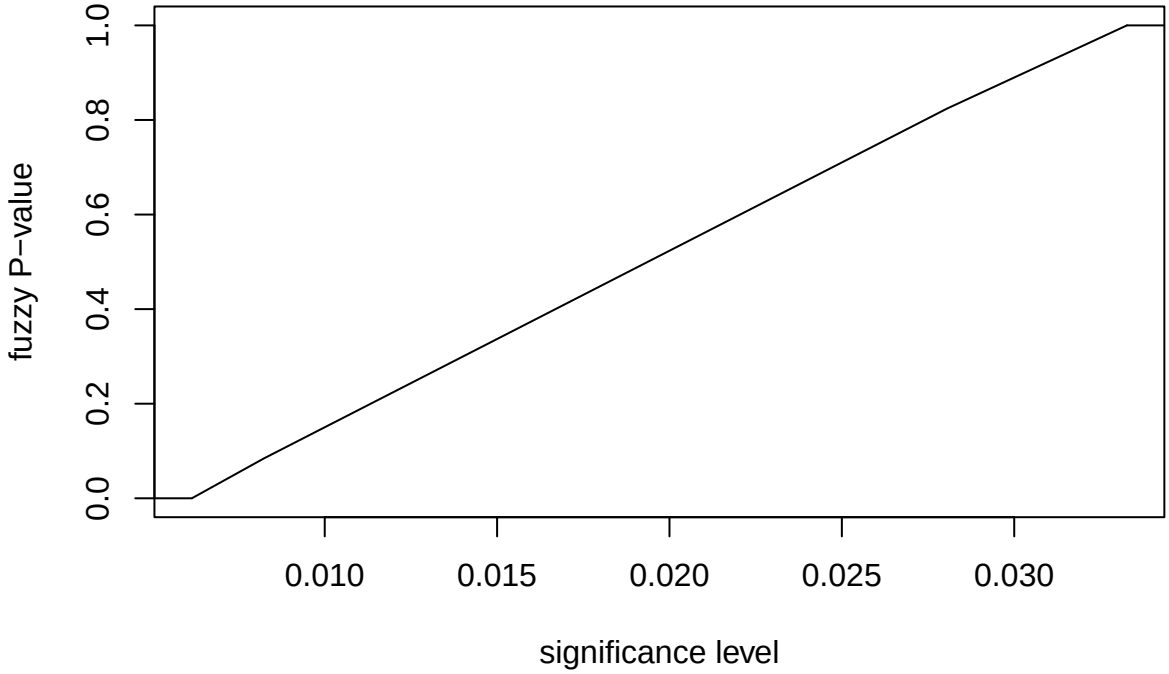


Figure 1: CDF of Fuzzy P-Value

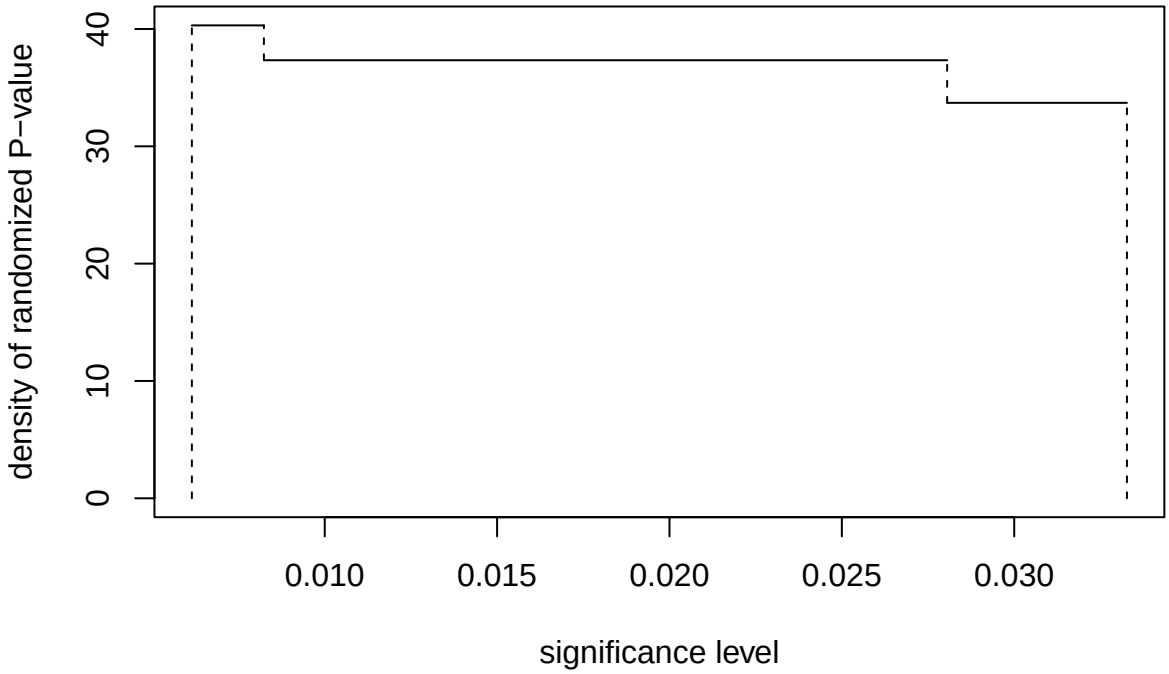


Figure 2: PDF of Fuzzy P-Value

But a co-author made me cut it from the paper, not because it was wrong, but because it might offend.

Middling P-values are already equivocal. Making them fuzzy doesn't make them much more equivocal.

But making them fuzzy does make them truly exact rather than just conservative-exact (for one-tailed tests). And making them fuzzy also makes them possible for two-tailed tests (there are no sensible nonrandomized two-tailed tests for non-symmetric binomial distributions, that is, unless the null hypothesis is $\pi_0 = 1/2$ there is no conservative-exact nonrandomized two-tailed test).

Also making them fuzzy makes them best possible (UMP or UMPU).

7 Fuzzy Confidence Intervals

Inverting a fuzzy hypothesis test gives a fuzzy confidence interval. We won't explain (see Geyer and Meeden) but just give examples and interpretations.

A fuzzy confidence interval for observed data x is a function m_x on the parameter space taking values between zero and one. The interpretation is that, if $m_x(\theta) = 0$, then θ is definitely out of the confidence interval, if $m_x(\theta) = 1$, then θ is definitely in the confidence interval, and otherwise θ is partly in and partly out and $m(\theta)$ says how much. We can think of $m_x(\theta)$ as being like "partial credit" on the question about θ. If θ_0 is the true unknown parameter value, then the fuzzy confidence interval gets full marks if $m_x(\theta_0) = 1$, it fails completely if $m_x(\theta_0) = 0$, and gets partial credit $m_x(\theta_0)$ whatever the value is.

When evaluating the performance of the fuzzy confidence interval, we say its coverage probability is

$$E_\theta\{w_X(\theta)\}$$

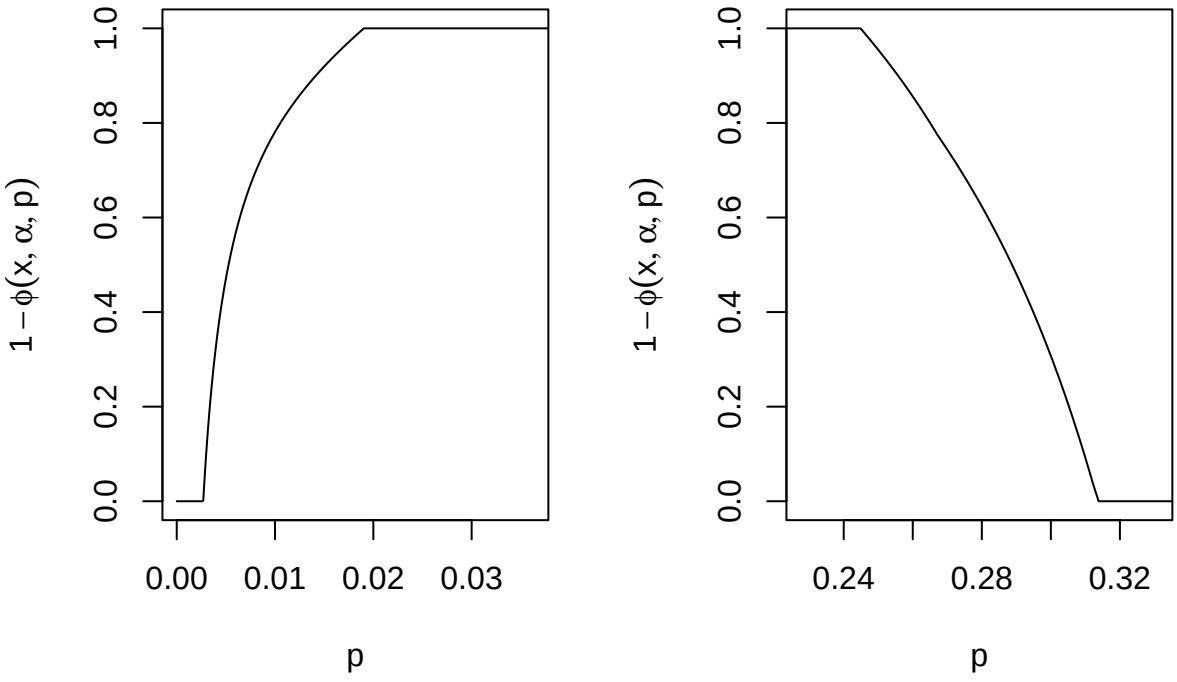
and we want to have this exactly at the specified confidence level.

And R function `fci.binom` in R package `ump` does do this. It produces exact fuzzy confidence intervals by inverting the UMPU two-tailed test.

Here's how that works.

```
fci.binom(x, n)
```

```
## 95 percent fuzzy confidence interval
## core is [0.0191, 0.2448]
## support is (0.0027, 0.3138)
```

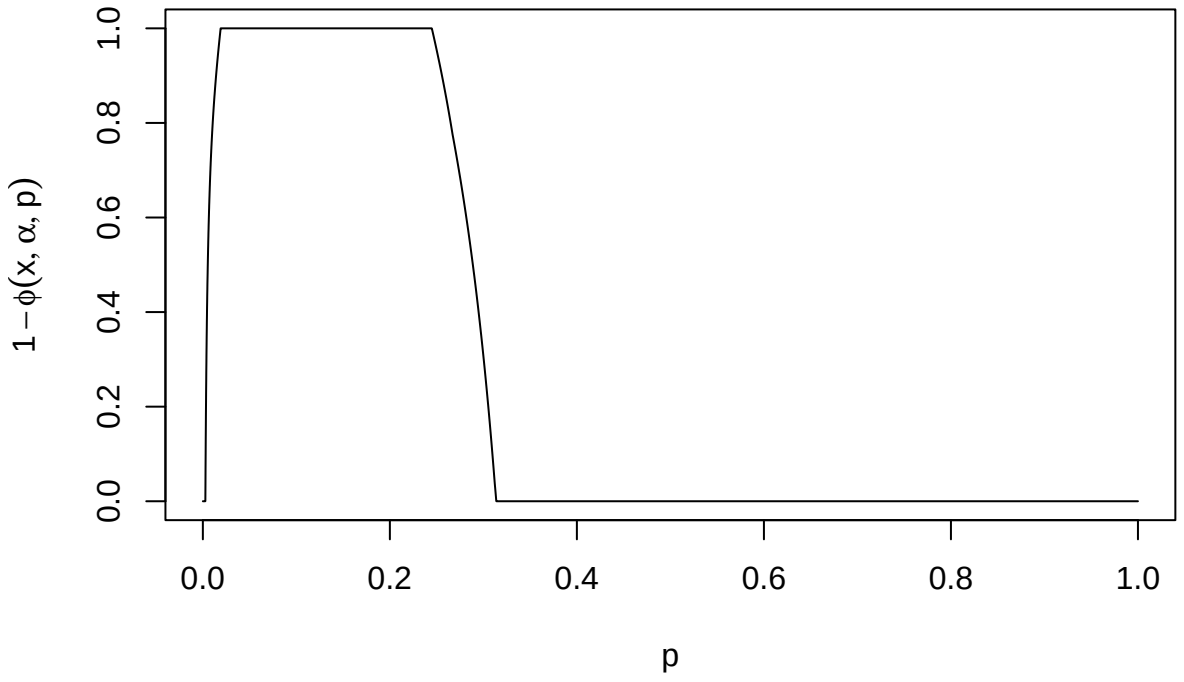


This function also blathers some. The "core" of the fuzzy confidence interval is the set of points that are definitely in the interval and the "support" is the set of points that are either definitely or partially in the interval.

If we don't want to just plot the edges of the fuzzy confidence interval (which it does by default). We can override that behavior with an optional argument. Here's how that works. is produced by the following code

```
fci.binom(x, n, flat = 100)
```

```
## 95 percent fuzzy confidence interval
## core is [0.0191, 0.2448]
## support is (0.0027, 0.3138)
```



We can see that a fuzzy confidence interval isn't all that different from a conventional confidence interval. In order to obtain exactness, we just need to allow for confidence intervals to have "partial credit". We know from the web page about coverage of confidence intervals that no conventional confidence interval can be exact-exact.