Using simulation to inspect the performance of a test

in particular tests of the parallel regressions assumption in ordered logit models

Maarten L. Buis[1] Richard Williams[2]

[1]WZB Berlin Social Research Center
Research unit: Skill Formation and Labor Markets
`www.maartenbuis.nl`

[2]University of Notre Dame
Department of Sociology
`www.nd.edu/~rwilliam/`

Parallel lines assumption in Ordered logit

- We have a dependent variable consisting of three ordered categories: 1, 2, and 3
- So we can look at the effect of a variable X on the comparison 1 versus 2 and 3 and the comparison 2 versus 3.
- An ordered logit results in one effect of X by assuming that these effects are the same
- A generalized version of this model allows some or all of these effects to be different. This model is implemented by Richard Williams in `gologit2`.

5 Tests of the parallel lines assumption after ordered logit

Tests of the parallel lines assumption compare the ordered logit model with a full generalized ordered logit model. There are 5 tests implemented in Stata (soon) in `oparallel`

- likelihood ratio test
- Wald test
- score test
- Wolfe-Gould test (approximate likelihood ratio test)
- Brant test (approximate Wald test)

What do I mean with ‘inspect the performance of a test’?

A test is based on the following process:

1. We think of a null hypothesis
2. We have drawn a sample
3. We imagine a world in which the null hypothesis is true and can that we draw many samples from this population
4. The p-value is the proportion of these samples that deviate from the null hypothesis at least as much as the observed data
5. It is the probability of drawing a sample that is at least as ‘weird’ as the observed data if the null hypothesis is true

What do I mean with ‘inspect the performance of a test’?

- The p-values returned by a test are often approximate, e.g. many are based on asymptotic arguments
- A valid question might be: Does the approximation work well enough for my dataset?
- To answer that question I am going to take the process of testing literally:
  1. I am going to change my data such that the null hypothesis is true
  2. I am going to draw many samples from this ‘population’ and perform the test in each of these samples
  3. I am going to compare the p-value returned by that test with the proportion of samples that are more extreme than that sample.

The distribution of p-values

- The p-value is one way to measure the difference between the data and the null-hypothesis, such that smaller values represent larger difference.
- If we find a p-value of α, than the probability of drawing a dataset with a p-value $\leq \alpha$ if the null hypothesis is true should itself be α, and this should be true for all possible values of α.
- So the sampling distribution of the p-values if the null hypothesis is true should be a standard uniform distribution.

The basic simulation (preparation)

```
clear all
use "http://www.indiana.edu/~jslsoc/stata/spex_data/ordwarm2.dta",
ologit warm white ed prst male yr89 age

predict double pr1 pr2 pr3 pr4, pr
forvalues i = 2/3 {
    local j = `i' - 1
    replace pr`i' = pr`i' + pr`j'
}
replace pr4 = 1
gen pr0 = 0
keep if e(sample)

gen ysim = .
gen u = .
```


The basic simulation (actual simulation)

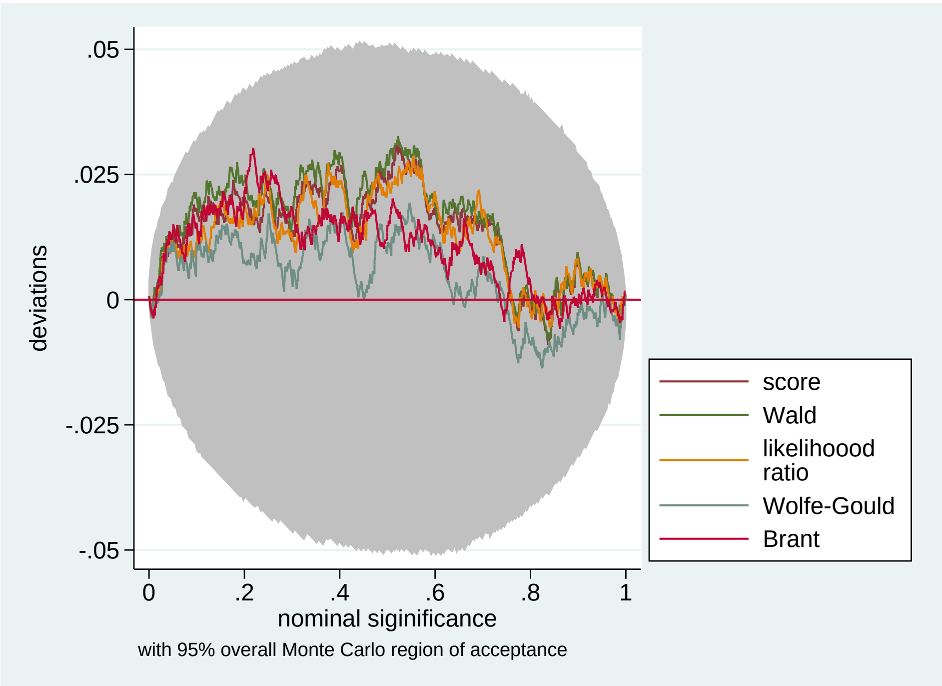
```
program define sim, rclass
    replace u = runiform()
    forvalues i = 1/4 {
        local j = `i' - 1
        replace ysim = `i' if u > pr`j' & u < pr`i'
    }
    ologit ysim white ed prst male yr89 age
    oparallel
return scalar s = r(p_s)
return scalar w = r(p_w)
return scalar lr = r(p_lr)
return scalar wg = r(p_wg)
return scalar b = r(p_b)
end

simulate s=r(s) w=r(w) lr=r(lr) wg=r(wg) b=r(b), reps(1000) : sim
```

The basic simulation (interpret the results)

```
simpplot s w lr wg b,            ///
 main1opt(ms(none) c(l) sort )  ///
 main2opt(ms(none) c(l) sort )  ///
 main3opt(ms(none) c(l) sort )  ///
 main4opt(ms(none) c(l) sort )  ///
 main5opt(ms(none) c(l) sort )  ///
 legend(order(2 "score"         ///
              3 "Wald"          ///
  4 "likelihoood"  ///
    "ratio"        ///
  5 "Wolfe-Gould"  ///
  6 "Brant" ))     ///
 overall reps(100000)           ///
 scheme(s2color)                ///
 ylab(-.05(.025).05,angle(horizontal))
```


The basic simulation (interpret the results)

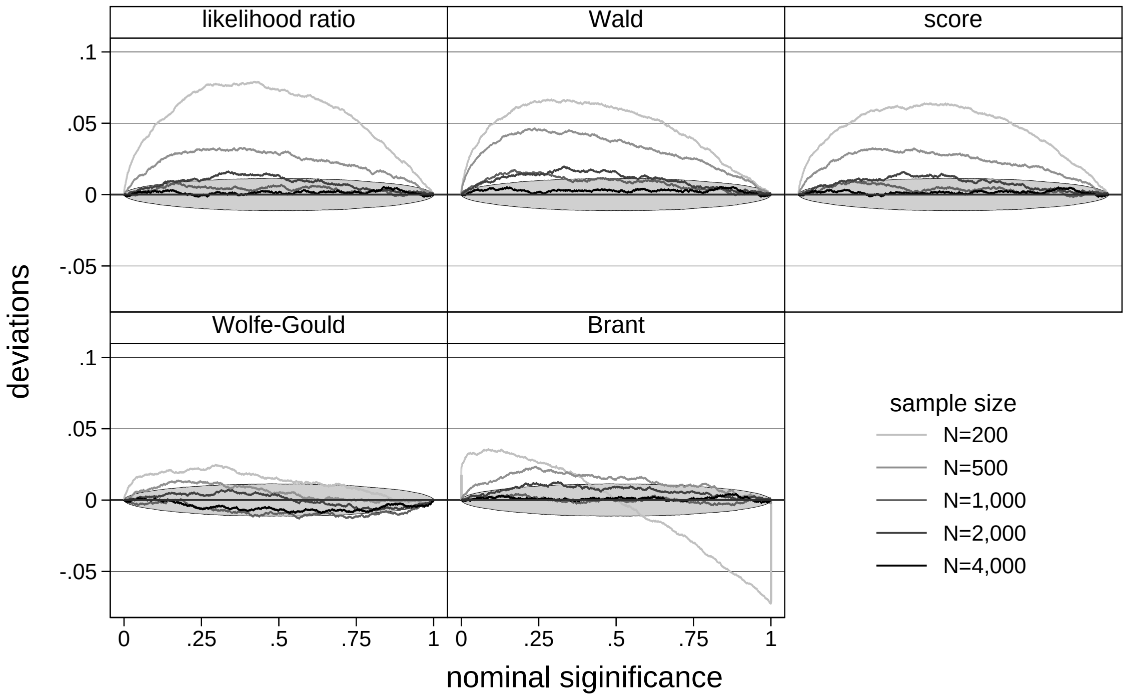


Sample size

- So, all three tests seem to work well in the current dataset, which contains 2,293 observations
- What if I have a smaller dataset?
- Adapt the basic example by sampling say 200 observations, like so:

```
<prepare data>
save prepared_data
program define sim, rclass
    use prepared_data
    bsample 200
    ...
```


sample size

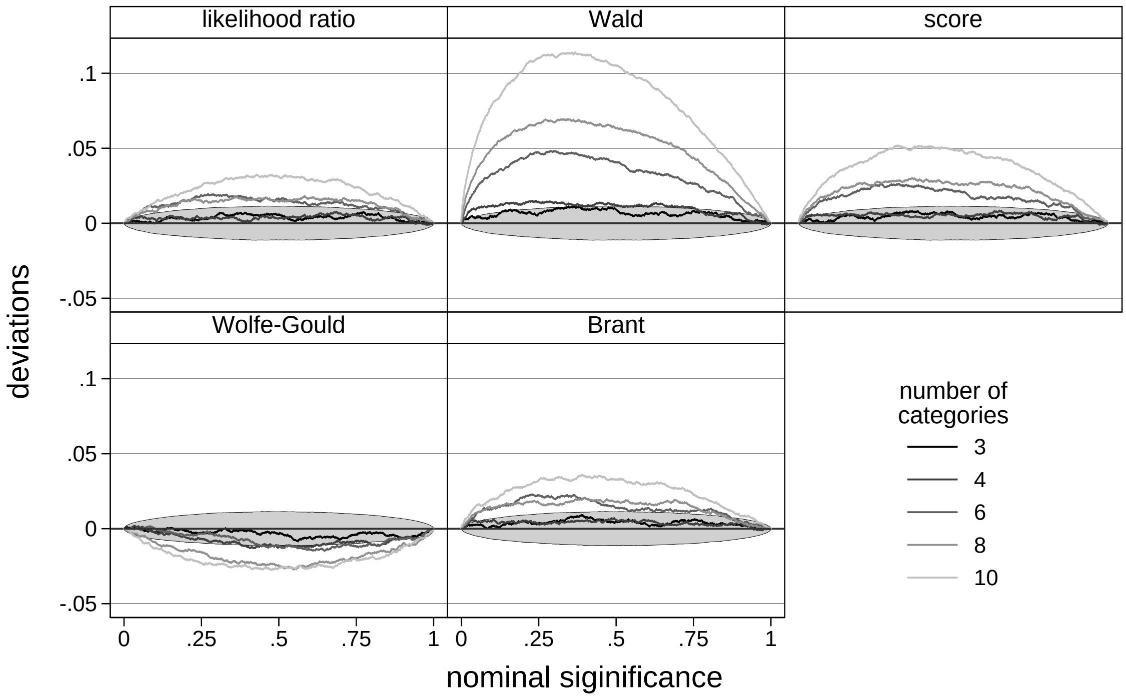


with 95% overall Monte Carlo region of acceptance

number of categories

- What if the number of observations remains constant at the observed number 2,293 but we increase the number of answer categories?
- We looked at 3, 4, 6, 8, and 10 categories, by changing the constants.
- These constants were chosen such that the proportion of observations in each of these categories are all the same

number of categories

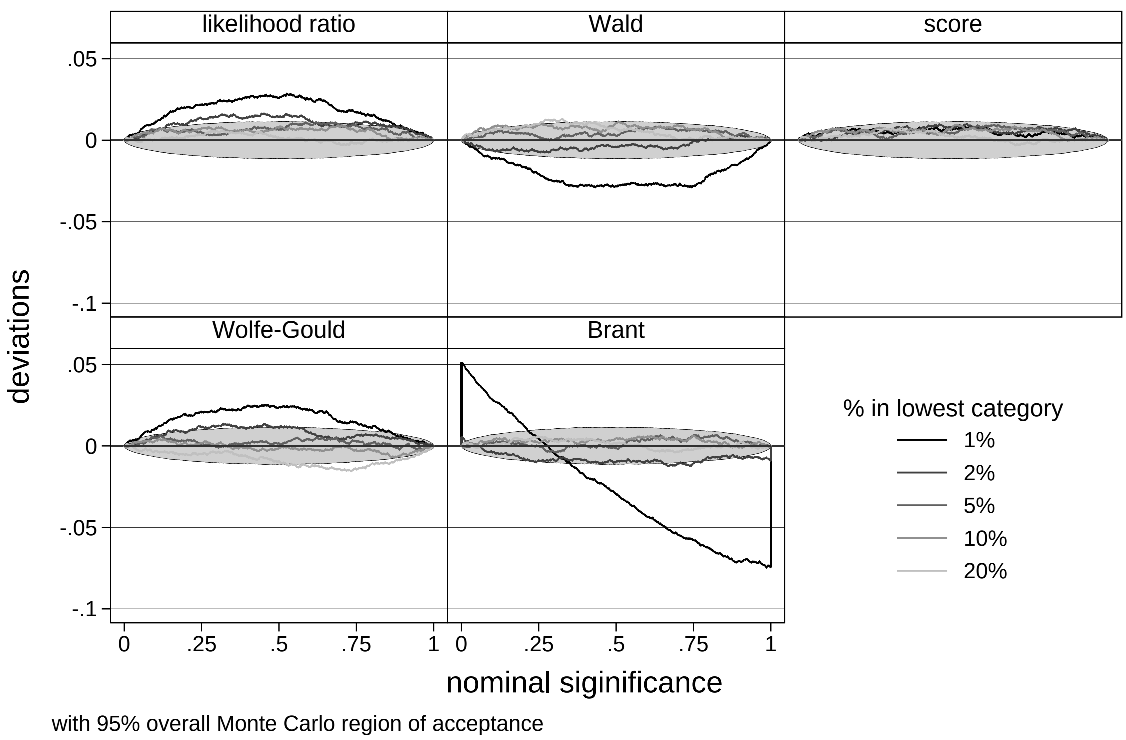


with 95% overall Monte Carlo region of acceptance

size of categories

- In this set-up the proportion in a category decreases as the number of categories increase
- Did we see an effect of the number of categories or of small categories?
- Such sparse categories are also common in real data and often cause trouble.
- We fix the number of categories at 4 but change the first constant such that the proportion of observations in the first two categories change
- We do that in such a way that the first category contains 1%, 2%, 5%, 10%, or 20% of the observations

size of categories



Bootstrap test

- Consider the basic simulation again.
- It creates a ‘population’ in which the null hypothesis is true, but is otherwise as similar to the data as possible
- It draws many times from this population, and in each of these draws it inspects how large the deviation from the null hypothesis is
- We could just count the number of samples in which that deviation is larger than in the observed data and we would have an estimate of the p-value
- This is a bootstrap test
- This is implemented in `oparallel` as the `asl` option

p-value = $\frac{k}{B}$ or $\frac{k+1}{B+1}$

- The ratio is of the number of samples that are at least as extreme as the observed data k over the the number of replications B is the natural estimate of the p-value. However...
- If the null hypothesis is true all possible values of a should be equally likely.
- If we draw B samples then there are $B+1$ possible outcomes: 0 , 1, $\cdots$, or B samples that are more extreme than the observed data.
- Each of these outcomes should be equally likely, so $\frac{1}{B+1}$
- So the probability of finding 0 or less samples that are more extreme than the observed data is $\frac{1}{B+1}$
- The probability of finding 1 or less samples that are more extreme than the observed data is $\frac{2}{B+1}$
- In general, the probability of finding k or less samples that are more extreme than the observed data is $\frac{k+1}{B+1}$

An alternative justification of $\frac{k+1}{B+1}$

- there is some ideal p-value based on an infinite number of bootstrap samples that we try to approximate.
- Based on B bootstrap one can determine the hypothetical rank i of the p-value in the observed data if it had occurred in one of the bootstrap samples.
- If there are no bootstrap samples with a p-value smaller than the observed p-value than the observed p-value would have been the smallest and would thus receive rank 1.
- Similarly, if there was only one bootstrap sample that produced a smaller p-value then the observed p-value would have received rank 2.
- In general, $i = k + 1$.
- We know that the underlying distribution of the ideal p-value must be a continuous standard uniform distribution.
- This means that the value of the i^{th} smallest value will follow a Beta distribution with parameters i and $B + 1 - i$
- The mean of this distribution is $i/(B + 1)$ = $(k + 1)/(B + 1)$.

uncertainty in the bootstrap estimate of the p-value

- There is randomness in our estimate of the p-value
- If we use the simple proportion as our estimate we can use the binomial distribution to compute a Monte Carlo confidence interval around our estimate (`cii` in Stata)
- If we use $(k + 1)/(B + 1)$ as our estimate we can use the Beta distribution
- The two are very similar

Concluding remarks

- Tests of the parallel lines assumption in ordered logit models tend to be a bit anti-conservative
- But it is nowhere near as bad as we expected
- Problematic situations are small sample sizes and a large number of categories in the dependent variable, but not so much a sparse categories.
- Surprisingly the Wolfe-Gould test seems to work best

Concluding remarks

- Does this mean that tests for the parallel lines is not anti-conservative?
- Not if you use it for model selection. If you are automatically going to reject your model when you find a significant deviation from the parallel lines assumptions you will reject to many useful models.
- A model is a simplification of reality. Simplification is another word for 'wrong in some useful way'. So, all models are by definition wrong.
- Finding that the parallel lines assumption does not hold tells you that the patterns you can see in a generalized ordered logit model are unlikely to be just random noise.
- It is now up to the researcher to determine whether these patterns are important enough to abandon the ordered logit model. This is a judgement call that cannot be delegated to a computer

Concluding remark

- Checking a test, we make sure we repeatedly draw from a population in which the null hypothesis is true
- in regression type problems it is usually enough to draw a new dependent variable from the distribution implied by the model
- The purpose is than to check whether the p-values follow a standard uniform distribution

Concluding remarks

- This idea can also be used to estimate p-values when the test itself does not behave as well as you would like.
- That is the bootstrap test, and it is a general idea. It has been applied in: `asl_norm` and `propcnsreg`