Appendix: Validation of power results

September 28, 2021

Contents

Introduction **1**
Monte Carlo Simulations 1
Validation scenarios 2
Simulation parameters 2

Validation results **3**

Westfall-Young procedures **3**

Introduction

This appendix discusses our work to validate that the power estimation methods work as intended.

We compare three different methods of estimating power:

- PUMP
- PowerUpR (only comparable for D1individual, unadjusted power)
- Monte Carlo Simulations

We compare point estimates of power from PUMP and PowerUpR for D1 individual, unadjusted power estimates. For all other types of power definitions and adjustments, we are only able to compare PUMP to the estimated power from Monte Carlo simulations. The simulations produce a 95% confidence interval for power, and we check that the PUMP estimate is within the confidence interval.

Monte Carlo Simulations

The main work of this validation step was to design monte carlo simulations in order to estimate power. In order to estimate power by simulation, we follow the following steps.

For iteration $s = 1, \dots S$:

1. Generate simulated data according to the assumed data generating process (DGP)
2. Generate simulated treatment assignment
3. Calculate p-value given simulated data, treatment assignment, and assumed model

At the end, a 95% confidence interval for power is calculated, assuming a conservative standard error estimate of $\sqrt{0.25/S}$.

We also validate MDES and sample size calculations. For MDES, we choose one default scenario for each design and model, then input the already-calculated D1 individual power and see if the output MDES is the same as the original input MDES. Similarly, for sample size validation, we input the already-calculated D1 individual power and see if the output sample size (either J or K depending on design) is the same as the original sample size.

Validation scenarios

We use the following adjustment procedures:

- Bonferroni
- Benjamini Hochberg (BH)
- Holm
- Westfall-Young Single Step (WY-SS)
- Westfall-Young Step Down (WY-SD)

We calculate power under the following definitions:

- Individual power for each outcome (M = 3): D1indiv, D2indiv, D3indiv
- Mean individual power
- Minimum power: min1, min2
- Complete power

We consider the following designs:

Code	Design	Model	PowerUp
d1.1_m2cc	d1.1	m2cc	n/a
d2.1_m2fc	d2.1	m2fc	blocked_i1_2c
d2.1_m2ff	d2.1	m2ff	blocked_i1_2f
d2.1_m2fr	d2.1	m2fr	blocked_i1_2r
d3.1_m3rr2rr	d3.1	m3rr2rr	blocked_i1_3r
d2.2_m2rc	d2.2	m2rc	simple_2c_2r
d3.3_m3rc2rc	d3.3	m3rc2rc	simple_c3_3r
d3.2_m3ff2rc	d3.2	m3ff2rc	blocked_c2_3f
d3.2_m3rr2rc	d3.2	m3rr2rc	blocked_c2_3r

Simulation parameters

In order to validate that the method works in a wide range of scenarios, which vary the following parameters.

Parameters that vary:

Parameter	Default	Comparison values
school size $\bar{n}$	50	75, 100
R^2	0	0.6
ρ	0.5	0.2, 0.8
ATE (ES) true positives	(0.125, 0.125, 0.125)	(0.125, 0, 0)
ICC	0.2	0.7
ω	0.1	0.8

We do not vary:

- $M = 3$
- J and K are fixed for each scenario
- Scalar grand mean Ξ_0
- Correlations between school random effects and impacts κ
- ρ informs all correlations; we keep the same correlation between covariates, residuals, impacts, random effects for all levels and across all outcomes

Validation results

Below is an example of a graph we use for validation. The red dot is the PUM estimate of power, the green dot is the PowerUpR estimate of power, and the 95% confidence intervals based on the monte carlo simulations are shown in blue. To validate that PUMP produces the expected result, we want to see the red and green points match, and for the red point to be within the blue intervals. The plot shows the results across different types of power and different MTPs. We repeat this graph for each set of parameters for each design.

TODO include plot

Next, we validate MDES and sample size calculations. The first column shows the calculated MDES or sample size, the middle column is the power we plugged into the calculation, and the last column shows the MDES or sample size that we are targeting. Thus, ideally we want the first and last columns to match.

TODO include tables

Westfall-Young procedures

The Westfall-Young procedure was validated separately due to its unique complications and computational burden.

First, we wrote a series of careful unit tests ensuring that our code worked as expected for each step of the WY procedure. One of these tests compared results from our procedure to the WY procedure in the multtest package, and found it matched quite well.

Second, we also tested the WY procedures using the same simulation procedure described above. For constant effect and fixed effect models, the power estimation matches between PUM and the simulations. However, for random effects models, the two methods diverge in some cases. We find that the simulated power matches the PUMP-calculated power only when (1) there is a large number of blocks/clusters, and (2) the user has a large number of WY permutations. Below, we discuss the single-step procedure because it is simpler, although the same concepts hold for the step-down procedure. Although it is difficult to verify why this discrepancy occurs, our hypothesis is that this behavior occurs due to the combination of the sensitivity of the WY procedure, and the instability of the random effects model.

The goal of the WY procedure is to estimate the adjusted p-value for outcome i

$$\tilde{p}_i = Pr\left(\min_{1\leq j\leq k} P_j \leq p_i \mid H_0^C\right).$$

where P_j is a random variable representing a p-value for outcome j, and lowercase p_i is the observed realization for outcome i. We assume a set of k tests with corresponding null hypotheses H_{0i} for $i = 1, \cdots, k$. The complete null hypothesis is the setting where all the null hypotheses are true: $H_0^C = \cap_{i=1}^k H_{0i} = \{\text{all } H_i \text{are true}\}$. In order to estimate this p-value, the p-value is calculated across B permutations

$$\tilde{p}_i = \frac{1}{B}\sum_{b=1}^{B} \mathbb{1}(\min_{1\leq j\leq k} p^\star_{b,j} \leq p_i).$$

In order for the procedure to be correct, we should be generating the p-values P_j from the the true distribution of p-values under the null hypothesis. We generally are testing outcomes that are truly significant, so the observed p-values are small. Thus, looking at these expressions, we are concerned about tail behavior; we are testing whether a small observed p-value is less than the minimum of of the generated p-values of our multiple outcomes. This combination of factors means that a small discrepancy in the tails between the generated null distribution and the true distribution could have a large impact on the adjusted p-value.

Given that we are relying on tail behavior, this explains why we need both a large number of permutations, and a large number of blocks/clusters. The large number of permutations is required because we are trying to estimate a rare event. With a significant outcome, it is rare that the observed p-value will be less than the minimum of p-values generated from the null distribution. The large number of blocks/clusters is required to accurately estimate the tail behavior for random effects models. With a small number of blocks/clusters, we may not properly estimate the spread of the distribution, resulting in a poor estimate of the tail behavior.