Tight, Rigorous, and Automated Type I Error Proofs with Simulation

Michael Sklar [1] James Yang [2]

[1]Confirm Solutions CEO

[2]Confirm Solutions; Stanford University

August 8, 2022

Introduction

Figure: Michael Sklar

Table of Contents

Our Goals

- Make the innovation process of trial designs **predictable** and **fast**!
- Introduce a new technique: "proof by simulation".
- Guarantee Type I Error control not just on simulated points in the null space, but also *on the whole null space*.
- Consequences:
  - Automate the Type I Error mathematical proofs.
  - Allow more complex designs to be used in practice.
  - Faster iteration of creating new designs.

Simulation Challenges

- Some challenges raised by FDA at the start of the Complex Innovative Design (CID) Pilot Program:
  - **How many points in the null hypothesis space do simulate?**
  - **What is the computational complexity? Does it scale with multiple hypothesis testing?**
- Other challenges:
  - Simulation on a finite number of points in the null hypothesis space does not give guarantees for the whole space. **How do we deal with composite nulls?**
  - Simulation has Monte Carlo error. **How do we control Type I Error (on the whole space) accounting for the stochastic error?**

Methodology: High-Level Sketch

- Key steps:
  1. Simulate design on a finite number of points in the null hypothesis space.
  2. Construct upper bound estimates of the true Type I Error *on a compact subset of the null space*.
  3. Prove that such estimates are above the true Type I Error with high confidence, pointwise *on the compact subset*.
- Idea: if the upper bound estimates are under level α, the Type I Error is highly likely to be under α as well.

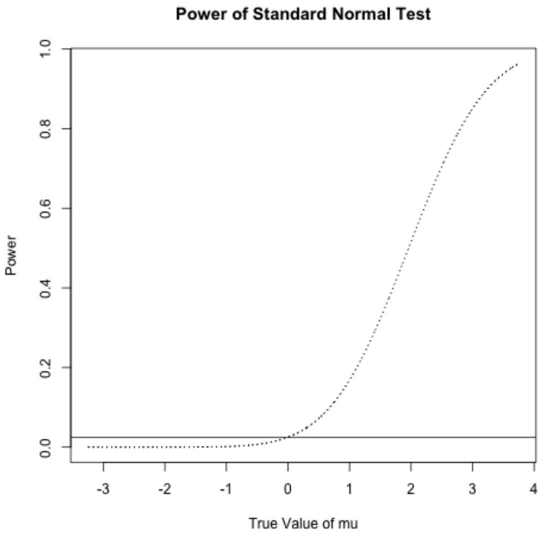
Methodology: High-Level Sketch

- Why do we assume compact subset?
- In practice, it is sufficient to study compact subsets.
- Theoretical arguments can often show that Type I Error is small in far regions.

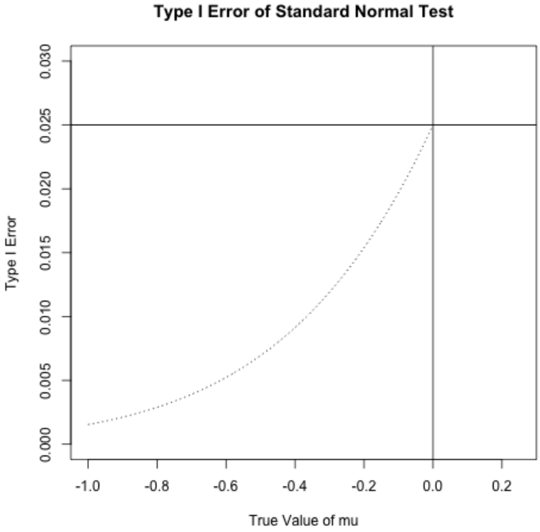
Simple Example: One-Sided Z-Test

- $X \sim \mathcal{N}(\mu, 1)$.
- $H_0 : \mu \leq 0$, $H_1 : \mu > 0$.
- Reject if $X > z_{1-\alpha}$.

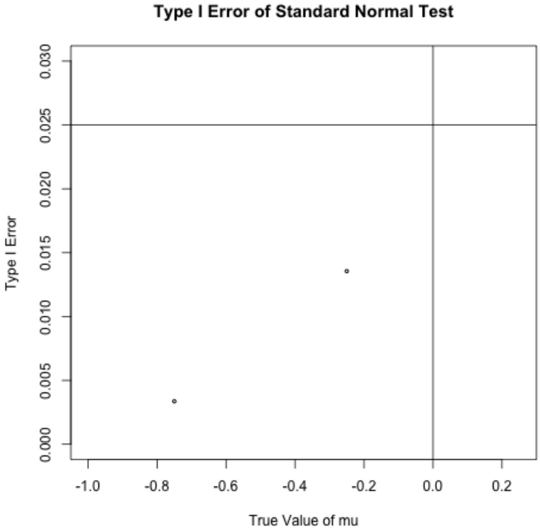
Simple Example: One-Sided Z-Test



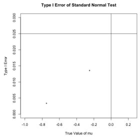
Simple Example: One-Sided Z-Test



Simple Example: One-Sided Z-Test

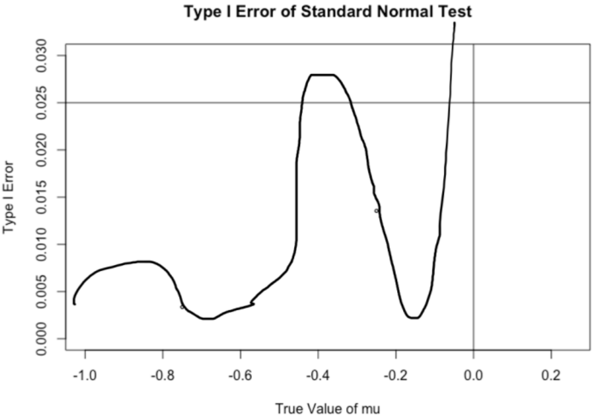


Simple Example: One-Sided Z-Test



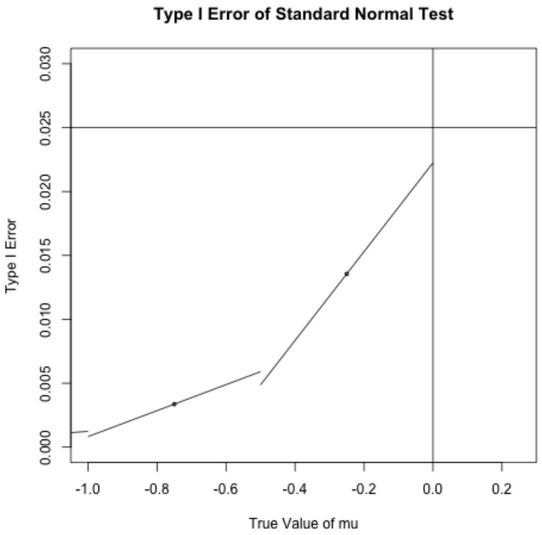
- When we simulate, we have to pick finitely many point nulls (2 here).
- For simplicity, let's say we knew exactly the Type I Error at these points.
- So, these points are on the curve from the last slide.
- Our problem is now: can we reconstruct the dotted line to show that our design controls Type I Error?
- This is a problem, because if we only know Type I Error at these points, the Type I Error could be wiggly elsewhere.

Simple Example: One-Sided Z-Test



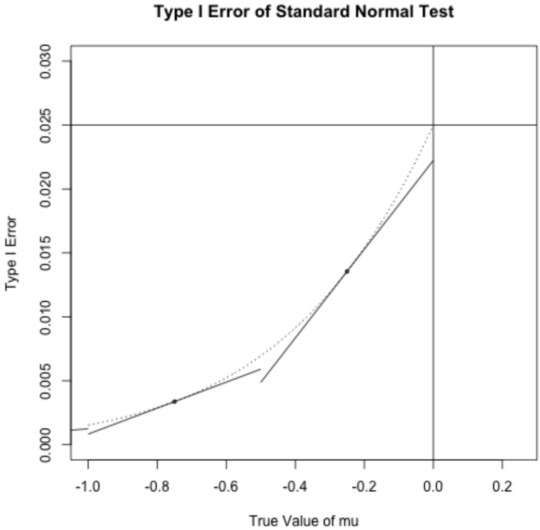
Simple Example: One-Sided Z-Test

- What if we knew the derivative of the true Type I Error at these points?
- Linear approximation?



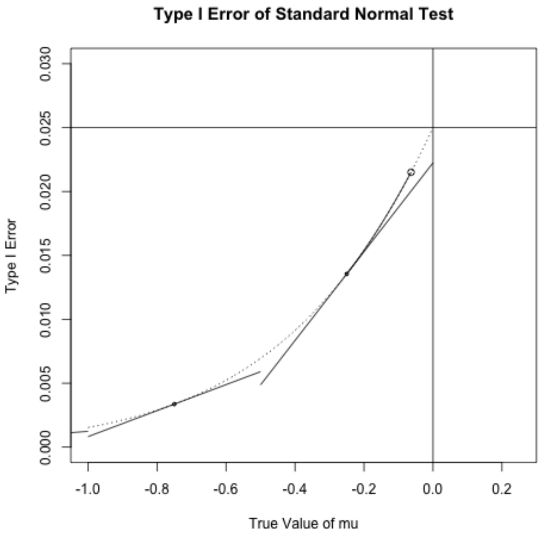
Simple Example: One-Sided Z-Test

- Always under the true Type I Error in this case due to convexity.



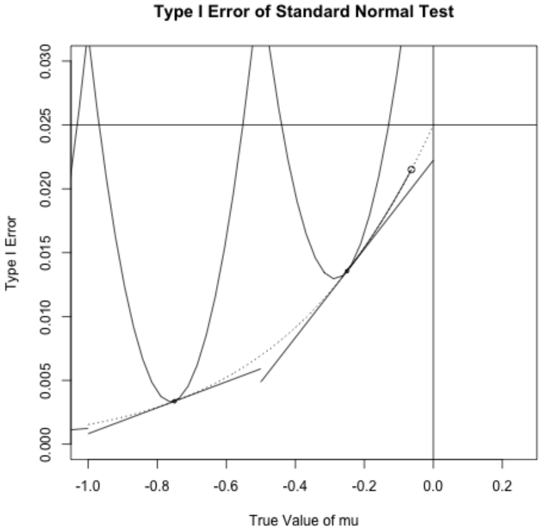
Simple Example: One-Sided Z-Test

- Quadratic approximation?



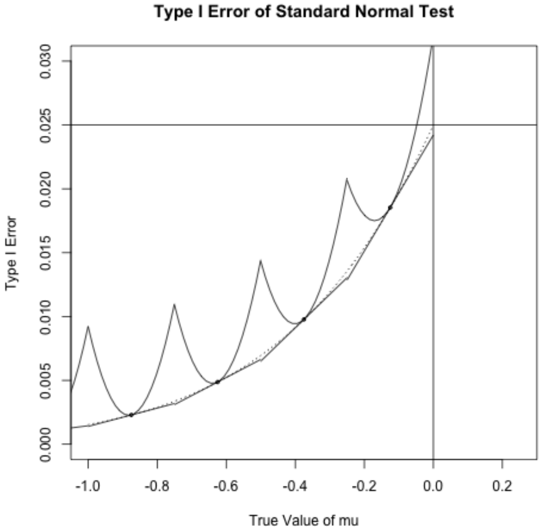
Simple Example: One-Sided Z-Test

- Conservative estimate using a bound on the second derivative.
- Consequence of Taylor's Theorem.
- This particular bound is pretty bad!



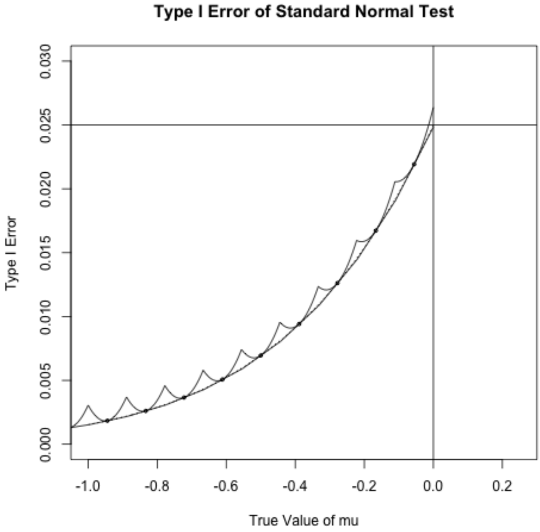
Simple Example: One-Sided Z-Test

- Increase number of simulation points!



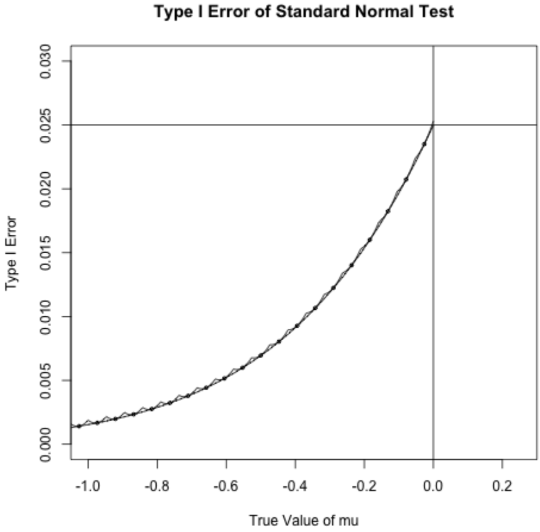
Simple Example: One-Sided Z-Test

- Increase number of simulation points!!



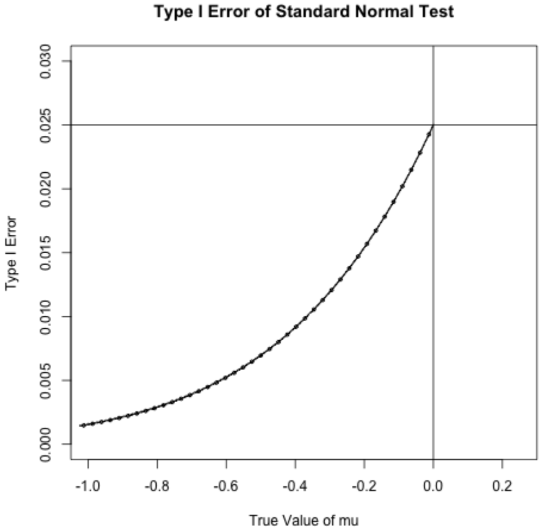
Simple Example: One-Sided Z-Test

- Increase number of simulation points!!!



Simple Example: One-Sided Z-Test

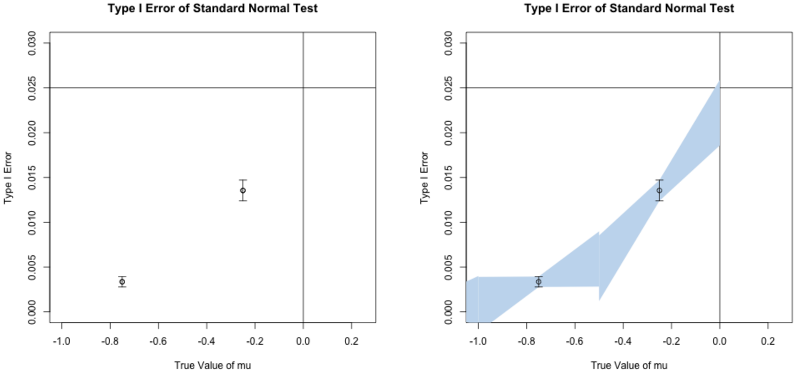
- Increase number of simulation points!!!!



Simple Example: One-Sided Z-Test

- Why did this quadratic approximation work so well?
- Problem is 1-dimensional.
  - More dimensions implies orders of magnitude more computation for the same level of approximation.
- Cheated!
  - Assumed knowledge of true Type I Error and its derivatives.

Simple Example: One-Sided Z-Test



- Monte Carlo estimates lie in confidence intervals around true value.
- Derivative estimates produce confidence bands at other values.
- Add a bound on second-order remainder term.

Less High-Level Sketch

- Let Θ_0 denote a compact subset of the null hypothesis space.
- Let $f(\theta)$ denote the true Type I Error of the design if θ were the true parameter.
- Key steps:
  1. Simulate the design on a (*finite*) set of grid-points in Θ_0.
  2. Construct a process, $\theta \mapsto \hat{U}(\theta)$, that depends on the simulated data.
  3. Prove that $\mathbb{P}\left[\hat{U}(\theta) \geq f(\theta)\right] \geq 1 - \delta$ for all $\theta \in \Theta_0$.

Setup

- Assume finite max number of patients in each arm.
- Let X, the full patient data, come from an exponential family P_θ:

$$dP_\theta(x) = \exp\left[T(x)^\top \theta - A(\theta)\right] d\mu(x)$$

- This assumption can be relaxed to densities with log-concave densities.
- Treat a design as a black-box.
  - Adaptive data collection is OK!
  - Censored data is OK!

Polytope Null

- Assume Θ_0 is a polytope and let θ_0 be a simulation point.
- Perform a Taylor expansion of the true Type I Error: for $v \in \Theta_0 - \theta_0$,

$$
\begin{aligned}
f\left(\theta_0 + v\right) = f\left(\theta_0\right) + \nabla f\left(\theta_0\right)^\top v \\
+ \int_0^1 (1-\alpha) v^\top \nabla^2 f(\theta_0 + \alpha v) v \, d\alpha
\end{aligned}
$$

- Goal: upper bound each term.

Polytope Null

- Let $F(x)$ denote the indicator that the test rejects with data x.
- **0th Order**: use $\frac{1}{n}\sum_{i=1}^{n} F(X_i)$ and upper bound with Clopper-Pearson to control $f(\theta_0)$.
- **1st Order**: use $\widehat{\nabla f}(\theta_0) := \frac{1}{n}\sum_{i=1}^{n} F(X_i)(T(X_i) - \nabla A(\theta_0))$ and upper bound $\nabla f\left(\theta_0\right)^\top v$ with Cantelli's Inequality to control $\nabla f\left(\theta_0\right)$. Since the null is a polytope, the worst case can be shown to occur at one of the corners of $\Theta_0 - \theta_0$.
- **2nd Order**: use $U(v) := \frac{1}{2}\sup_{\theta\in\Theta_0} v^\top \mathrm{Var}_\theta\left[T(X)\right] v$ to dominate the remainder term.
- Combine these estimates and their bounds to get a total upper bound on Θ_0.

Final case: Compact Null

- Assume Θ_0 is a compact subset of the null space.
- Create a disjoint polytope covering of Θ_0.
- Apply previous result to each element in the covering.

Possible Workflow Issue

- Declaring victory if $\hat{U} \leq \alpha$ may still be problematic.
- Due to randomness in $\hat{U}$, this decision rule does not lead to any meaningful statement about the true Type I Error being under α.

Solution: Tune Threshold

- Tune the rejection threshold!
- Key idea: find the critical value, $\hat{\lambda}$, where $\hat{U}(\theta)$ first hits level α exactly for some $\theta \in \Theta_0$.
- Then, letting $f_\lambda(\theta)$ denote the true Type I Error at θ with rejection threshold λ, we have that

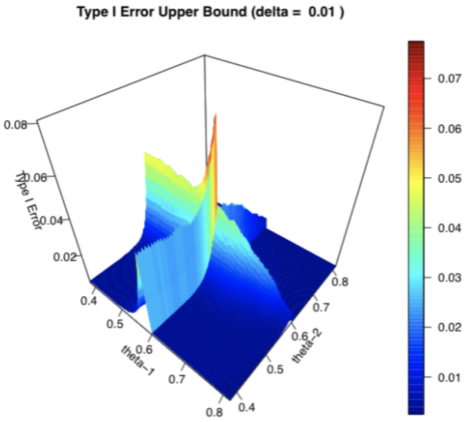
$$\mathbb{P}\left[f_{\hat{\lambda}}(\theta) \leq \alpha\right] \geq 1 - \delta$$

- Interpretation: if a trial were run with independent data from the simulated data used to compute $\hat{\lambda}$, the true Type I Error will be under α with high probability.

Thompson Sampling

- Two-arm trial.
- Outcomes $Y_{ij} \sim \mathit{Bernoulli}(\theta_j)$ $(i = 1, \ldots, 100)$.
- $H_0 : \theta_1 < 0.6$, $H_1 : \theta_1 \geq 0.6$.
- $\mathit{Beta}(1, 1)$ prior
- Reject arm 1 at the end if posterior $\mathbb{P}(\theta_1 > 0.6) > 0.7$.

Thompson Sampling



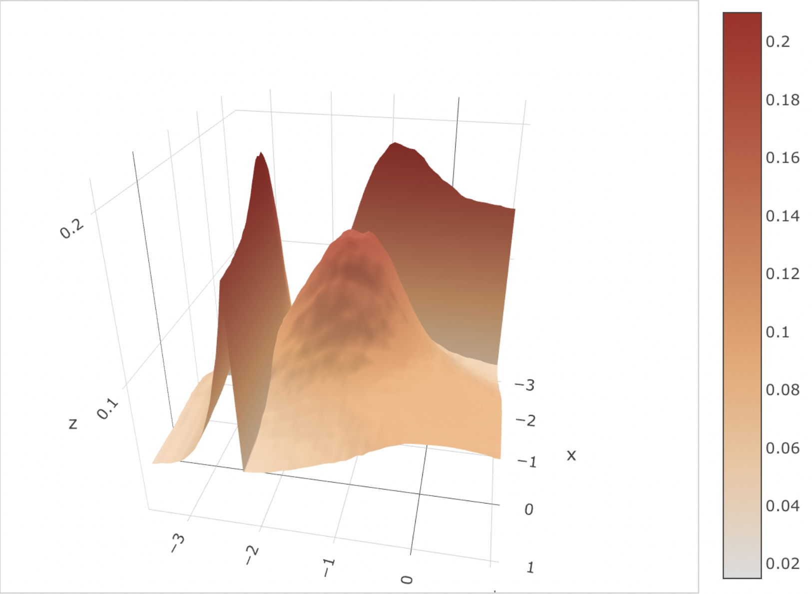
Berry et al (2013)

- Basket design:

$$
\begin{aligned}
Y_j &\sim Binom(n_j, p_j) \quad j = 1, \ldots, d \\
p_j &= \text{expit}(\theta_j + \text{logit}(q_j)) \\
\theta_j &\sim \mathcal{N}\left(\mu, \sigma^2\right) \\
\mu &\sim \mathcal{N}\left(\mu_0, \sigma_0^2\right) \\
\sigma^2 &\sim \Gamma^{-1}\left(\alpha_0, \beta_0\right)
\end{aligned}
$$

- Let $c \in [0,1]^{d-1}$ be a vector of fixed thresholds.
- Reject if $\mathbb{P}\left[p_i > p_0 | Y\right] > c_i$ for some null (treatment) arm i.

Berry Basket Trial



Computation Scale

- Number of parameters to grid should be small (≤ 6) to be tractable.
- Example designs with number of simulations and null points:

Design Type	Number of Sims	Number of Null Points	Total Sims (billion)
Thompson	100,000	16,384	1.6
Exponential Hazard	100,000	16,384	1.6
Binomial Selection	100,000	262,144	25
Berry Basket	10,000	5,308,416	100

Computation Bottleneck

- Most of the bottleneck is in speeding up the trial simulation itself.
- Lots of practical designs use Bayes.
- Traditional methods to obtain Bayes quantity is through MCMC or analytical formulas (conjugacy).
- Found *Integrated Nested Laplace Approximation* (INLA) to be **efficient** and **accurate**.
- Homegrown INLA library in development.

Main Contributor of INLA Library + Application

Figure: Ben Thompson

Simulations are Fast!

- JAX Python library and homegrown C++ codebase.
- Benchmark on modern Apple M1 Macbook (CPU) and cloud machine with NVIDIA V100 (GPU):

Design Type	Number of Sims	Number of Null Points	Hardware	Time (m)
Exponential Hazard	100,000	16,384	CPU	1.57
Binomial Selection	100,000	262,144	CPU	2.37
Thompson	100,000	16,384	CPU	4
Berry Basket	10,000	5,308,416	CPU	150
Berry Basket	10,000	5,308,416	GPU	1.16

Acknowledgements

(a) Alex Constantino

(b) Gary Mulder

(c) Daniel Kang

Recap

- Our methodology gives provable control of Type I Error via simulation.
- Allows complex designs to be studied seamlessly.
- Our software can make simulations extremely fast.