# Temporal-aware Bayesian inference for optimizing experimental returns

## STATS 209 - Project Presentation

Mark Robinson

Stanford University

2025 12 03

1. Introduction
2. Model
3. Results
4. Outlook

## Background & Motivation

- Representative problem: Experimentation program design in the technology industry
- Competing approaches to program design:
  - Traditional frequentist approach: searching for statistically significant results
  - Bayesian approach: directly optimizing for business metrics (e.g. user satisfaction; click-through rate)
- Potential gains from improvements?
  - Faster and more accurate testing at lower cost
    - *Preview*: This is achievable!

# Model set-up (outline)

Model for experimental program design (e.g. Azevedo et al. (2020), Sudijono et al. (2025)) –

- Set-up:
  - Organization generates a set of ideas over time; can choose which ideas to implement.
  - Ideas have some true (unobserved) value, organization would only like to implement good ideas.
  - Organization can carry out testing to learn about the true value of ideas, but faces limited testing resources.
- Organization problem: Choose –
  1. **Testing procedure**: Which ideas to test, and how many resources to allocate to each test
  2. **Decision rule**: Given testing results, which ideas to 'shelve' and which to 'ship'

# Model set-up (notation)

- Set-up:
  - Discrete time periods $t$: $1, \ldots, \infty$
  - New ideas generated each period $i$: $1, \ldots, I_t$
  - Testing resources each period $n$: $1, \ldots, N_t$
- Ideas and testing:
  - True idea values $\Delta_i$; prior over idea values: $\Delta_i \sim_{iid} (\mu, \tau^2)$.
  - Signal for idea $i$ generated according to $\hat{\Delta}_i \sim (\Delta_i, \frac{\sigma^2}{N_i})$, where $N_i$ is the number of units allocated for testing for unit $i$, and $\sigma^2$ is some baseline level of noise.
- Payoff:
  - Organization payoff each period equal to:

$$U_t = \sum_{\{i \in S_t\}} u(\Delta_i) = \sum_{\{i \in S_t\}} \Delta_i$$

# Extending decisions to be 'temporal-aware'

- New organization problem: Choose –
  1. **Testing procedure**: Which ideas to test, and how many resources to allocate to each test
  2. **Decision rule**: Given testing results across all previous periods, which ideas to 'shelve', which to 'ship' and which to continue testing
- Formulating the new problem:
  - Let $I_t^c$ denote cumulative ideas up to period $t$, and similar for $\Delta_{i,t}^c, n_{i,t}^c$. Let $\gamma$ be a discount factor over the future. Then total expected reward from shipping ideas with positive posterior mean is:

$$\sum_{t=1}^{\infty} \gamma^t \sum_{i=1}^{I_t^c} \mathbb{E}[\mathbb{E}[u(\Delta_{i,t})|\hat{\Delta}_{i,t}^c; n_{i,t}^c] 1(n_{i,t}^c)]$$

where $1(n_{i,t}^c)$ is an indicator of whether the inner expectation is larger than the 'ship' threshold.

# Optimal organization decision-making

The optimal organization decisions will cover the following:

1. Decision rule thresholds: Optimal shelve/continue testing/ship thresholds $(\alpha;\beta)$ over $\tilde{\Delta}_i$ for an idea, dependent on how times the idea has previously been tested: $\tilde{\Delta}_i < \alpha \implies$ shelve; $\tilde{\Delta}_i > \beta \implies$ ship; $\tilde{\Delta}_i \in [\alpha, \beta] \implies$ continue testing
   - Optimal thresholds determine $\beta$ by $U_t(\beta) = \beta$; and $\alpha$ by $U_t(\alpha) = U_0$, where $U_0$ is the expected return from an untested idea.
2. Number of ideas to test each period $(k)$
   - Assumption 1: Test every idea assigned to 'continue testing'
   - Assumption 2: Test the same number of ideas overall each period $(k_t = k_{t+1})$
   - Assumption 3: Allocate resources equally between tested ideas each period $(n_{i,t} = \frac{N_t}{k})$. Motivation – Sudijono et al. (2025).

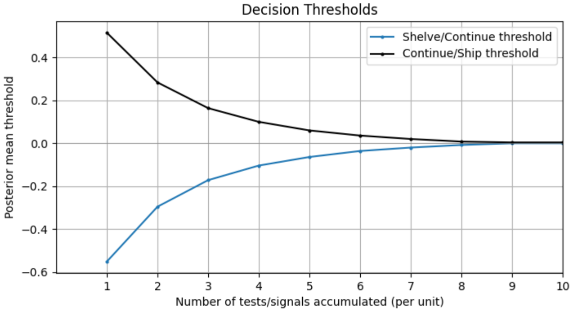
# Solution approach

Constructing a solution –

- Solve using numerical methods. For simplicity:
  - Normal prior distribution over ideas derived from Netflix experimentation data (Sudijono et al., 2025) [Work in progress, placeholder values used for now]
  - Risk-neutral utility function $U(x) = x$
- Solving proceeds iteratively as follows:
  1. Deriving decision rule thresholds:
     - Given prior and parameters, $k$, and a guess/estimate for $U_0$, identify optimal shelve/continue/ship thresholds
     - Given optimal thresholds, derive $U_0$.
     - Repeat until convergence.
  2. Deriving optimal number of ideas to test:
     - Given decision rule thresholds, calculate implied expected utility from each possible number of ideas. Choose $n^*$ to maximize expected utility.

# Optimal program design (decision thresholds)

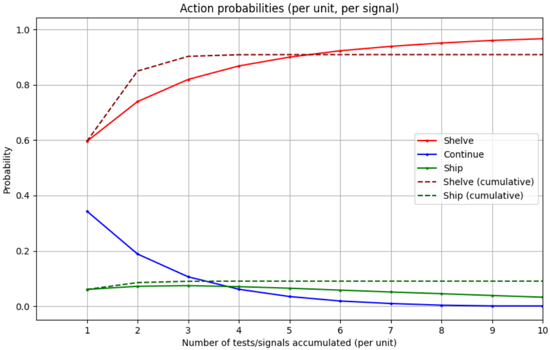
Optimal decision thresholds at $\mu = -1.0$; $\tau = 1.0$; $\sigma^2 = 2.0$; $N = 1000$; $k = 1$ and $\gamma = 0.99$




Intuition – early: prioritize further testing; later: only re-test very uncertain cases!

# Optimal program design (action probabilities)

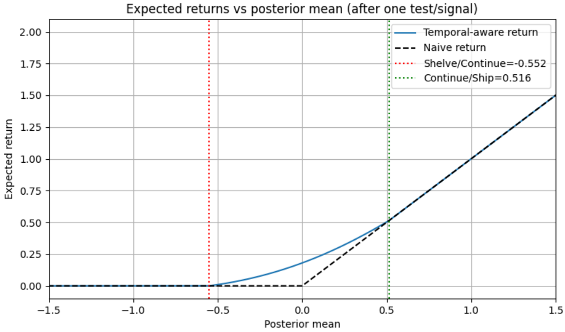
Probabilities of each action implied by decision thresholds:




Intuition? Given most ideas are bad, immediately discard most unless strong evidence in favor.

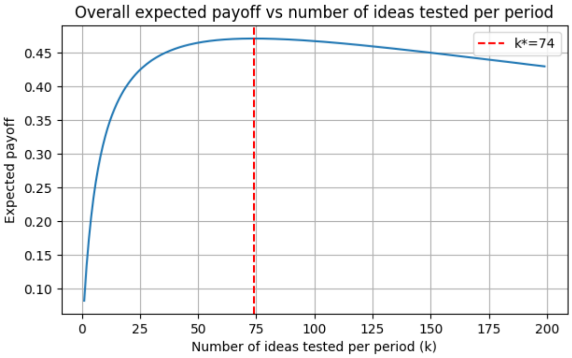
# Optimal program design (number of ideas)

- **Naive return**: Expected payoff if decision had to be made after one period. Equal to posterior mean
- **Temporal-aware return**: Expected payoff from following decision rule. Allows for higher return on average within 'continue' range.



# Optimal program design (payoff)

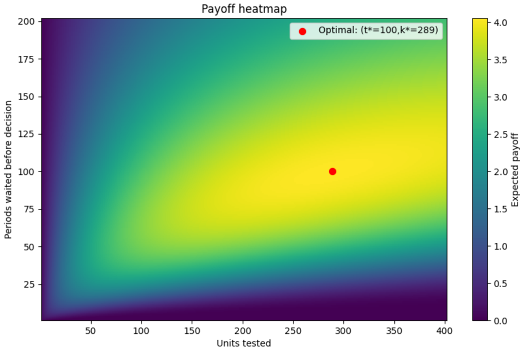
Expected payoff changing with number of ideas selected to test each period, given optimal decision thresholds.




Intuition? Trade-off between increasing the number of ideas tested each period and decreasing testing power due to finite resources.

# Defining a baseline strategy for comparison

- Compare 'temporal-aware' results to a restricted strategy – without the option to continue testing.
- Chooses $t^*$ periods over which to accumulate signals and $k^*$ ideas. 'Ship' / 'shelve' each tested unit after $t^*$ periods:



## Strategy comparison

Use average per-period naive strategy and temporal-aware payoffs to compare the two approaches. With these parameters, 'temporal-awareness' has:

- $\approx 70x$ faster testing per idea
- $\approx 20x$ higher per-period ideas tested
- $\approx 5x$ higher per-period expected return

Results are robust to different parameters. Temporal awareness exhibits even higher relative performance with lower $\mu$ and $\sigma^2$; and higher $\gamma$.

## Extensions

Various modifications / extensions are possible which lead to a richer model:

- Cost of ideation (increases 'continue' rate!)
- Risk-aversion (increases 'continue' rate!)
- Cost of implementation (?)
- Alternative distributions (?)

As well as considering these, my next steps are:

- Repeat analysis using real-world data from Netflix
- Consider links to wider literature including sequential frequentist testing

## Thanks

# Thanks for listening!

Even if you are not assigned to me for peer feedback, any thoughts/comments still welcome at mrobin10@stanford.edu (or in person).

References:

Azevedo, E. M., Deng, A., Montiel Olea, J. L., Rao, J. and Weyl, E. G. (2020), 'A/b testing with fat tails', *Journal of Political Economy* **128**(12), 4614–000.

Sudijono, T., Ejdemyr, S., Lal, A. and Tingley, M. (2025), Optimizing returns from experimentation programs, *in* 'Proceedings of the 26th ACM Conference on Economics and Computation', pp. 869–869.