# Reinforcement Learning for Business, Economics, and Social Sciences

Unit 1-5: Inference with Batched Bandits

Davud Rostam-Afschar (Uni Mannheim)

# How to get bandits normal?

# Stylized data structure

| Obs | Selected arm | Reward |
|---|---|---|
| 1 | A | 0 |
| 2 | B | 0 |
| 3 | A | 1 |
| 4 | B | 0 |
| 5 | A | 0 |
| 6 | B | 1 |
| 7 | A | 1 |
| 8 | B | 0 |
| 9 | A | 0 |
| 10 | A | 1 |
| 11 | A | 1 |
| 12 | B | 0 |
| 13 | A | 1 |
| 14 | A | 0 |
| 15 | A | 1 |
| 16 | B | 0 |

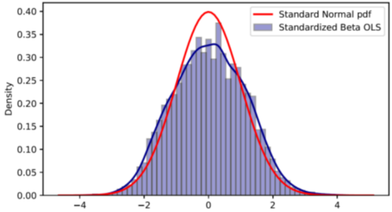
- Does arm A or arm B perform better?
- Which arm to play in next trial (round 17)?

# Bandits >> A/B Tests

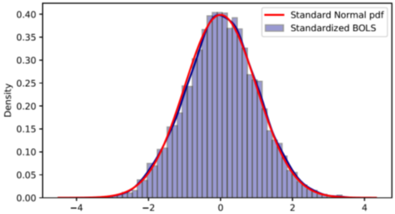
- Push to replace non-adaptive randomized trials with bandits
  - In development and labor economics, finance, biostats, health, ...
  - Can improve outcomes for participants (optimize regret)
  - Can improve policies learned at the end of trial (best-arm identification)
- **Problem**:
  - Bandits are not easy to implement
    Not available in statistical software like Stata
  - Bandits break inference
    Adaptive arm allocations
    $\rightarrow$ breaks asymptotics of usual estimators
    $\rightarrow$ wrong confidence intervals
- **Solution: Batched OLS (BOLS) for Batched Bandits**

# A Simple Example

- OLS and BOLS under Beta-Bernoulli two-arm Thompson Sampling with batch size $N_t = 100$ at batch $t = 10$
- All simulations are with no margin ($\beta_1 = \beta_0 = 0$)




(a) Empirical distribution of standardized OLS estimator for the margin




(b) Empirical distribution of standardized BOLS estimator for the margin

# Batchwise Data Collection

# Stylized data structure

| Obs | Selected arm | Batch | Reward |
|---|---|---|---|
| 1 | A | 0 | . |
| 2 | B | 0 | . |
| 3 | A | 0 | . |
| 4 | B | 0 | . |
| 5 | . | 1 | . |
| 6 | . | 1 | . |
| 7 | . | 1 | . |
| 8 | . | 1 | . |
| 9 | . | 2 | . |
| 10 | . | 2 | . |
| 11 | . | 2 | . |
| 12 | . | 2 | . |
| 13 | . | 3 | . |
| 14 | . | 3 | . |
| 15 | . | 3 | . |
| 16 | . | 3 | . |

# Stylized data structure

| Obs | Selected arm | Batch | Reward |
|---|---|---|---|
| 1 | A | 0 | 0 |
| 2 | B | 0 | 0 |
| 3 | A | 0 | 1 |
| 4 | B | 0 | 0 |
| 5 | . | 1 | . |
| 6 | . | 1 | . |
| 7 | . | 1 | . |
| 8 | . | 1 | . |
| 9 | . | 2 | . |
| 10 | . | 2 | . |
| 11 | . | 2 | . |
| 12 | . | 2 | . |
| 13 | . | 3 | . |
| 14 | . | 3 | . |
| 15 | . | 3 | . |
| 16 | . | 3 | . |

# Stylized data structure

| Obs | Selected arm | Batch | Reward |
|---|---|---|---|
| 1 | A | 0 | 0 |
| 2 | B | 0 | 0 |
| 3 | A | 0 | 1 |
| 4 | B | 0 | 0 |
| 5 | A | 1 | . |
| 6 | B | 1 | . |
| 7 | A | 1 | . |
| 8 | B | 1 | . |
| 9 | . | 2 | . |
| 10 | . | 2 | . |
| 11 | . | 2 | . |
| 12 | . | 2 | . |
| 13 | . | 3 | . |
| 14 | . | 3 | . |
| 15 | . | 3 | . |
| 16 | . | 3 | . |

# Stylized data structure

| Obs | Selected arm | Batch | Reward |
|---|---|---|---|
| 1 | A | 0 | 0 |
| 2 | B | 0 | 0 |
| 3 | A | 0 | 1 |
| 4 | B | 0 | 0 |
| 5 | A | 1 | 0 |
| 6 | B | 1 | 1 |
| 7 | A | 1 | 1 |
| 8 | B | 1 | 0 |
| 9 | . | 2 | . |
| 10 | . | 2 | . |
| 11 | . | 2 | . |
| 12 | . | 2 | . |
| 13 | . | 3 | . |
| 14 | . | 3 | . |
| 15 | . | 3 | . |
| 16 | . | 3 | . |

# Stylized data structure

| Obs | Selected arm | Batch | Reward |
|---|---|---|---|
| 1 | A | 0 | 0 |
| 2 | B | 0 | 0 |
| 3 | A | 0 | 1 |
| 4 | B | 0 | 0 |
| 5 | A | 1 | 0 |
| 6 | B | 1 | 1 |
| 7 | A | 1 | 1 |
| 8 | B | 1 | 0 |
| 9 | A | 2 | . |
| 10 | A | 2 | . |
| 11 | A | 2 | . |
| 12 | B | 2 | . |
| 13 | . | 3 | . |
| 14 | . | 3 | . |
| 15 | . | 3 | . |
| 16 | . | 3 | . |

# Stylized data structure

| Obs | Selected arm | Batch | Reward |
|---|---|---|---|
| 1 | A | 0 | 0 |
| 2 | B | 0 | 0 |
| 3 | A | 0 | 1 |
| 4 | B | 0 | 0 |
| 5 | A | 1 | 0 |
| 6 | B | 1 | 1 |
| 7 | A | 1 | 1 |
| 8 | B | 1 | 0 |
| 9 | A | 2 | 0 |
| 10 | A | 2 | 1 |
| 11 | A | 2 | 1 |
| 12 | B | 2 | 0 |
| 13 | . | 3 | . |
| 14 | . | 3 | . |
| 15 | . | 3 | . |
| 16 | . | 3 | . |

# Stylized data structure

| Obs | Selected arm | Batch | Reward |
|---|---|---|---|
| 1 | A | 0 | 0 |
| 2 | B | 0 | 0 |
| 3 | A | 0 | 1 |
| 4 | B | 0 | 0 |
| 5 | A | 1 | 0 |
| 6 | B | 1 | 1 |
| 7 | A | 1 | 1 |
| 8 | B | 1 | 0 |
| 9 | A | 2 | 0 |
| 10 | A | 2 | 1 |
| 11 | A | 2 | 1 |
| 12 | B | 2 | 0 |
| 13 | A | 3 | . |
| 14 | A | 3 | . |
| 15 | A | 3 | . |
| 16 | B | 3 | . |

# Stylized data structure

| Obs | Selected arm | Batch | Reward |
|---|---|---|---|
| 1 | A | 0 | 0 |
| 2 | B | 0 | 0 |
| 3 | A | 0 | 1 |
| 4 | B | 0 | 0 |
| 5 | A | 1 | 0 |
| 6 | B | 1 | 1 |
| 7 | A | 1 | 1 |
| 8 | B | 1 | 0 |
| 9 | A | 2 | 0 |
| 10 | A | 2 | 1 |
| 11 | A | 2 | 1 |
| 12 | B | 2 | 0 |
| 13 | A | 3 | 1 |
| 14 | A | 3 | 0 |
| 15 | A | 3 | 1 |
| 16 | B | 3 | 0 |

# Stylized data structure

| Obs | Selected arm | Batch | Reward | True Expected Reward |
|---|---|---|---|---|
| 1 | A | 0 | 0 | 0.5 |
| 2 | B | 0 | 0 | 0.2 |
| 3 | A | 0 | 1 | 0.5 |
| 4 | B | 0 | 0 | 0.2 |
| 5 | A | 1 | 0 | 0.5 |
| 6 | B | 1 | 1 | 0.2 |
| 7 | A | 1 | 1 | 0.5 |
| 8 | B | 1 | 0 | 0.2 |
| 9 | A | 2 | 0 | 0.5 |
| 10 | A | 2 | 1 | 0.2 |
| 11 | A | 2 | 1 | 0.5 |
| 12 | B | 2 | 0 | 0.2 |
| 13 | A | 3 | 1 | 0.5 |
| 14 | A | 3 | 0 | 0.2 |
| 15 | A | 3 | 1 | 0.5 |
| 16 | B | 3 | 0 | 0.2 |

# Stylized data structure

| Obs | Selected arm | Batch | Reward | True Expected Reward | OLS |
|---|---|---|---|---|---|
| 1 | A | 0 | 0 | 0.5 | 0.600 |
| 2 | B | 0 | 0 | 0.2 | 0.167 |
| 3 | A | 0 | 1 | 0.5 | 0.600 |
| 4 | B | 0 | 0 | 0.2 | 0.167 |
| 5 | A | 1 | 0 | 0.5 | 0.600 |
| 6 | B | 1 | 1 | 0.2 | 0.167 |
| 7 | A | 1 | 1 | 0.5 | 0.600 |
| 8 | B | 1 | 0 | 0.2 | 0.167 |
| 9 | A | 2 | 0 | 0.5 | 0.600 |
| 10 | A | 2 | 1 | 0.2 | 0.600 |
| 11 | A | 2 | 1 | 0.5 | 0.600 |
| 12 | B | 2 | 0 | 0.2 | 0.167 |
| 13 | A | 3 | 1 | 0.5 | 0.600 |
| 14 | A | 3 | 0 | 0.2 | 0.600 |
| 15 | A | 3 | 1 | 0.5 | 0.600 |
| 16 | B | 3 | 0 | 0.2 | 0.167 |

# Stylized data structure

| Obs | Selected arm | Batch | Reward | True Expected Reward | OLS | Batch-Wise OLS |
|---|---|---|---|---|---|---|
| 1 | A | 0 | 0 | 0.5 | 0.600 | 0.500 |
| 2 | B | 0 | 0 | 0.2 | 0.167 | 0.000 |
| 3 | A | 0 | 1 | 0.5 | 0.600 | 0.500 |
| 4 | B | 0 | 0 | 0.2 | 0.167 | 0.000 |
| 5 | A | 1 | 0 | 0.5 | 0.600 | 0.500 |
| 6 | B | 1 | 1 | 0.2 | 0.167 | 0.500 |
| 7 | A | 1 | 1 | 0.5 | 0.600 | 0.500 |
| 8 | B | 1 | 0 | 0.2 | 0.167 | 0.500 |
| 9 | A | 2 | 0 | 0.5 | 0.600 | 0.667 |
| 10 | A | 2 | 1 | 0.2 | 0.600 | 0.667 |
| 11 | A | 2 | 1 | 0.5 | 0.600 | 0.667 |
| 12 | B | 2 | 0 | 0.2 | 0.167 | 0.000 |
| 13 | A | 3 | 1 | 0.5 | 0.600 | 0.667 |
| 14 | A | 3 | 0 | 0.2 | 0.600 | 0.667 |
| 15 | A | 3 | 1 | 0.5 | 0.600 | 0.667 |
| 16 | B | 3 | 0 | 0.2 | 0.167 | 0.000 |

# Stylized data structure

| Obs | Selected arm | Batch | Reward | True Expected Reward | OLS | Batch-Wise OLS | $\omega_t$ |
|---|---|---|---|---|---|---|---|
| 1 | A | 0 | 0 | 0.5 | 0.600 | 0.500 | $\sqrt{\frac{2\times 2}{2+2}}$ |
| 2 | B | 0 | 0 | 0.2 | 0.167 | 0.000 | $\sqrt{\frac{2\times 2}{2+2}}$ |
| 3 | A | 0 | 1 | 0.5 | 0.600 | 0.500 | $\sqrt{\frac{2\times 2}{2+2}}$ |
| 4 | B | 0 | 0 | 0.2 | 0.167 | 0.000 | $\sqrt{\frac{2\times 2}{2+2}}$ |
| 5 | A | 1 | 0 | 0.5 | 0.600 | 0.500 | $\sqrt{\frac{2\times 2}{2+2}}$ |
| 6 | B | 1 | 1 | 0.2 | 0.167 | 0.500 | $\sqrt{\frac{2\times 2}{2+2}}$ |
| 7 | A | 1 | 1 | 0.5 | 0.600 | 0.500 | $\sqrt{\frac{2\times 2}{2+2}}$ |
| 8 | B | 1 | 0 | 0.2 | 0.167 | 0.500 | $\sqrt{\frac{2\times 2}{2+2}}$ |
| 9 | A | 2 | 0 | 0.5 | 0.600 | 0.667 | $\sqrt{\frac{1\times 3}{1+3}}$ |
| 10 | A | 2 | 1 | 0.2 | 0.600 | 0.667 | $\sqrt{\frac{1\times 3}{1+3}}$ |
| 11 | A | 2 | 1 | 0.5 | 0.600 | 0.667 | $\sqrt{\frac{1\times 3}{1+3}}$ |
| 12 | B | 2 | 0 | 0.2 | 0.167 | 0.000 | $\sqrt{\frac{1\times 3}{1+3}}$ |
| 13 | A | 3 | 1 | 0.5 | 0.600 | 0.667 | $\sqrt{\frac{1\times 3}{1+3}}$ |
| 14 | A | 3 | 0 | 0.2 | 0.600 | 0.667 | $\sqrt{\frac{1\times 3}{1+3}}$ |
| 15 | A | 3 | 1 | 0.5 | 0.600 | 0.667 | $\sqrt{\frac{1\times 3}{1+3}}$ |
| 16 | B | 3 | 0 | 0.2 | 0.167 | 0.000 | $\sqrt{\frac{1\times 3}{1+3}}$ |

# Point estimates OLS vs. BOLS

Aggregate or batched OLS (BOLS) estimator

$$\Delta^{\text{BOLS}} = \frac{\sum_t^T \omega_t \times \Delta_t^{BOLS}}{\sum_t^T \omega_t},$$

where $\omega_t = \sqrt{\frac{N_{t,k} \times N_{t,b}}{N_{t,k} + N_{t,b}}}$.

- $N_{t,k}$ is the number of times that comparison arm $k$ was played
- $N_{t,b}$ is the number of times that baseline arm $b$ was played
- weights batchwise estimates
- such that the aggregate margins are consistent and asymptotically **normally** distributed (Zhang, Janson, and Murphy, 2020)

# Point estimates OLS vs. BOLS

Example from stylized data structure

| | |
|---|---|
| OLS | $\widehat{\text{Reward}} = 0.6 - 0.433 \times \mathbb{1}_{\text{arm B}}$ |
| BOLS | $-0.443 = \frac{1\times 0.5 + 1\times 0 + \sqrt{\frac{1\times 3}{1+3}}\times 0.667 + \sqrt{\frac{1\times 3}{1+3}}\times 0.667}{1+1+\sqrt{\frac{1\times 3}{1+3}} + \sqrt{\frac{1\times 3}{1+3}}}$ |
| | $\widehat{\text{Reward}} = 0.6 - 0.443 \times \mathbb{1}_{\text{arm B}}$ |

## Inference OLS vs. BOLS

$$\mathbb{P}\left(\Delta^{\text{BOLS}} - c\sigma w_t \leq \mu \leq \Delta^{\text{BOLS}} + c\sigma w_t\right) = 1 - \alpha,$$

- where $\Delta^{\text{BOLS}}$ is the weighted estimated marginal effect
- $\mu$ is the hypothesized difference between means of the arms
- $c$ is a critical value, e.g., the $1 - \alpha/2 = 97.5$th percentile of a normal
- $\sigma$ reflects the sampling error
- $w_t$ is a weight correcting the bias due to adaptive sampling

$$w_t = \sqrt{T} / \sum_{t=1}^{T} \omega_t.$$

- $T$ is the total number of batches

# Batched Bandits in Practice

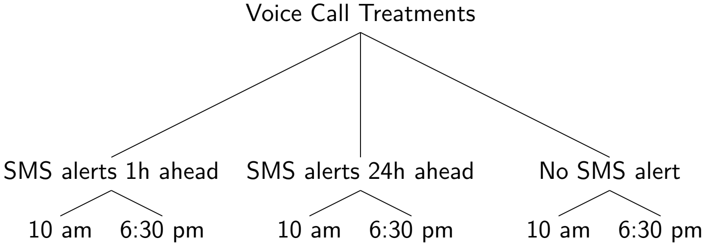
# Empirical application (Kasy and Sautmann, 2021)

**Six call methods to enroll rice farmers**

- Kasy and Sautmann (2021) designed an experiment using *exploration sampling* for Precision Agriculture for Development
- NGO that works with government partners to provide a phone-based personalized agricultural extension service to farmers in India
- Aim is to choose best call methods to enroll rice farmers in one state

**Six call methods to enroll rice farmers**

- The outcome (reward) is a binary variable for call completion:
    - $= 1$ if call recipient answered five questions asked during call
    - $= 0$ otherwise



# Empirical application (Kasy and Sautmann, 2021)

- *Exploration sampling* replaces the Thompson assignment shares
- modification shifts weight away from the best performing option to competing treatments
- 10,000 valid phone numbers randomly assigned to one of 16 batches
- batch size was 600 numbers each (and one with 400)
- From June 3, 2019 batches run every other day, completed next day

# Empirical application (Kasy and Sautmann, 2021)

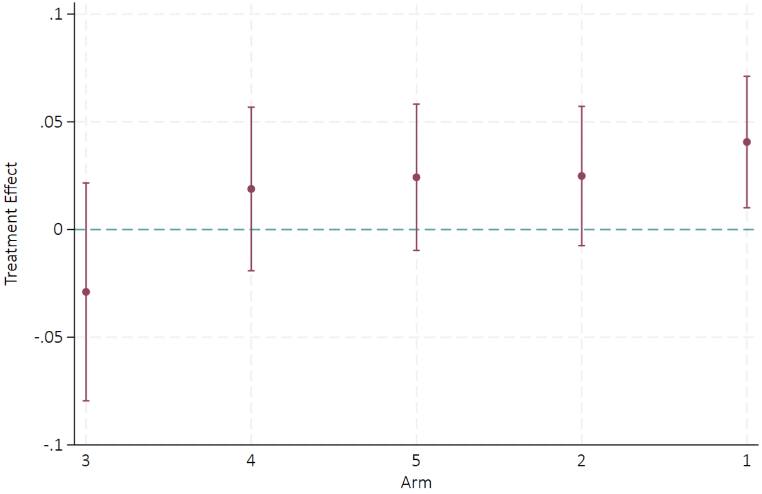
```
.
. use "example data\kasy_sautmann_2021.dta", clear
. bbandits outcome treatment date
```

| | | | | | |
|---|---|---|---|---|---|
| Number of obs | = | 10000 | | | |
| Est. Rewards only best arm | = | 1926 | Mean reward best arm | = | 0.1926 |
| Actual total reward | = | 1804 | Actual mean reward | = | 0.1804 |
| Est. reward uniformly chosen arms | = | 1709 | Mean reward uniform | = | 0.1709 |

| Arm b | Mean Reward | | | | | | Share arm b |
|---|---|---|---|---|---|---|---|
| | 0.1606 | | | | | | 0.0903 |
| k v. b | Margin OLS | Margin BOLS | z | P>\|z\| | [95% Conf. | Interval] | Share arm k |
| 1-0 | 0.0320 | 0.0406 | 2.61 | 0.009 | 0.0101 | 0.0711 | 0.3931 |
| 2-0 | 0.0185 | 0.0249 | 1.51 | 0.132 | -0.0075 | 0.0572 | 0.2234 |
| 3-0 | -0.0158 | -0.0289 | -1.12 | 0.262 | -0.0795 | 0.0216 | 0.0366 |
| 4-0 | 0.0078 | 0.0188 | 0.97 | 0.330 | -0.0191 | 0.0568 | 0.1081 |
| 5-0 | 0.0192 | 0.0243 | 1.40 | 0.161 | -0.0097 | 0.0582 | 0.1485 |

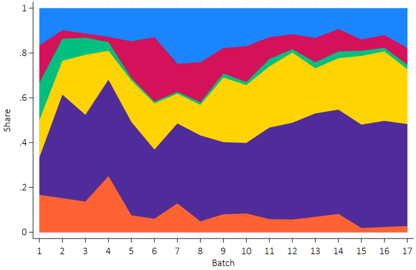
| Treatment | 0 | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|---|
| SMS | — | 1h ahead | 24h ahead | — | 1h ahead | 24h ahead |
| Call time | 10 am | 10 am | 10 am | 6:30 pm | 6:30 pm | 6:30 pm |

# Empirical application (Kasy and Sautmann, 2021)

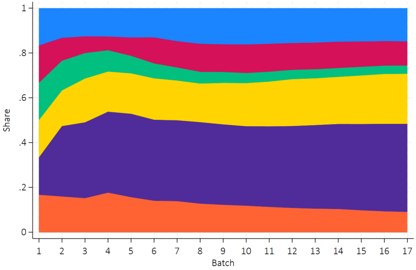



The figure was generated using `kasy_sautmann_2021.dta` and running `bbandits outcome treatment date`

# Empirical application (Kasy and Sautmann, 2021)




(a) Batchwise shares




(b) Cumulative shares

The figure was generated using `kasy_sautmann_2021.dta` and running `bbandits outcome treatment date`

## Empirical application (Kasy and Sautmann, 2021)

Takeaways

**Clear best and worst arms**

- Best: Calling farmers at 10 am after a message an hour ahead
- Worst: Calling at 6:30 pm without a text message alert

**Improvement of success rate**

- 18.04% success rates within the experiment
- 17.15% success rate with equal assignment

# References


Kasy, M., and A. Sautmann (2021): "Adaptive Treatment Assignment in Experiments for Policy Choice," *Econometrica*, 89(1), 113–132.

Kemper, J., and D. Rostam-Afschar (2025): "Earning While Learning: How to Run Batched Bandit Experiments," University of Mannheim.

Offer-Westort, M., A. Coppock, and D. P. Green (2021): "Adaptive Experimental Design: Prospects and Applications in Political Science," *American Journal of Political Science*, 65(4), 826–844.

Zhang, K., L. Janson, and S. Murphy (2020): "Inference for Batched Bandits," *Advances in neural information processing systems*, 33, 9818–9829.

# Takeaways

# How to run Batched Bandit Experiments?

- Bandits may improve learning and exploitation
- There is a push to use more bandits in real experiments in economics, biostats, health, psychology, political science (Offer-Westort, Coppock, and Green, 2021), ...
  and survey research methods!
- need for valid inference to support conclusions
  - bandits break inference
  - researchers want valid confidence intervals
- **Batched bandit inference (download BBandits)**
  - (Kemper and Rostam-Afschar, 2025)
  - BOLS allows valid statistical inference & correct coverage for batched bandits