Limitations and Opportunities of Bandit Algorithms for Feature Selection

Presentation by: Eddie (Yidi) Wu

Brown University

April 13, 2026

Overview

Introduction

- Researchers and policymakers are interested in identifying which variables are most influential in determining outcomes.
- Understanding the key drivers of outcomes provides helps design more effective and targeted policies.
- In high-dimensional settings, selecting a small subset of relevant covariates improves interpretability and reduces model complexity.

Feature Selection Example

- Let $(y_i, x_{1,i}, ...x_{K,i}), i = 1, ...n$ be i.i.d random vectors indexed by i. Predict y_i by $x_i^T \beta$.
- LASSO estimator solves:

$$\hat{\beta} = argmin_\beta(\frac{1}{n}||Y - X\beta||_2^2 + \lambda||\beta||_1)$$

- Suppose $\frac{1}{n}X^T X = I_K$, the $\hat{\beta}_j = sign(z_j)(|z_j| - \frac{\lambda}{2})_+$ where z_j is the j-th row of $z = \frac{1}{n}X^T Y$.
- L1 penalty encourages sparsity in the coefficients. Feature selection based on the non-zero coefficients in $\hat{\beta}$.

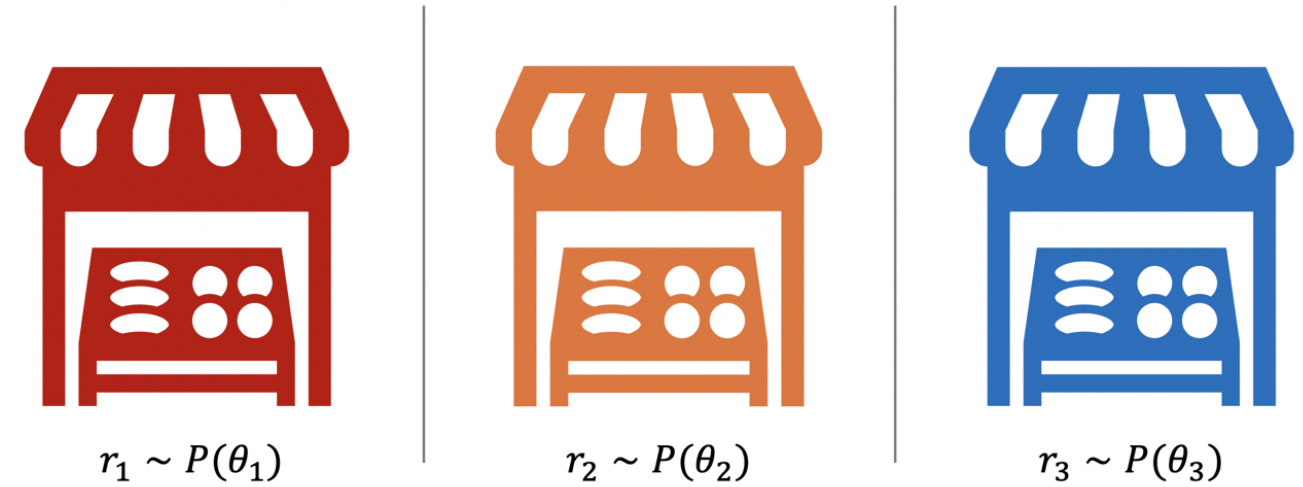
Stability Selection (Meinshausen & Buhlmann (2010))

Algorithm: Stability Selection

Input: penalty parameter λ; number of iterations T; data D; selection threshold π.
Output: stable feature set $\hat{S} \subseteq 2^{[K]}$.
Initialize $Count_k \leftarrow 0$ for $k = 1, \ldots, K$
for $t = 1, \ldots, T$ **do**
    Draw a subsample D' from D without replacement.
    Run LASSO on D' with penalty λ.
    Let $\hat{S}_t$ be the selected feature set.
    $Count_k \leftarrow Count_k + 1$ for all $k \in \hat{S}_t$.
end for
$\Pi_k \leftarrow Count_k / T$ for $k = 1, \ldots, K$.
$\hat{S} \leftarrow \{k : \Pi_k \geq \pi\}$.

Overview

Bandit Introduction



- A policy can aim at one of the following objectives:
  - Maximize expected cumulative reward $E[\sum_{t=1}^{T} r_{k,t}]$.
  - Finding the optimal arm $argmax_{k \in [1,\ldots K]} \theta_k$ with high confidence.

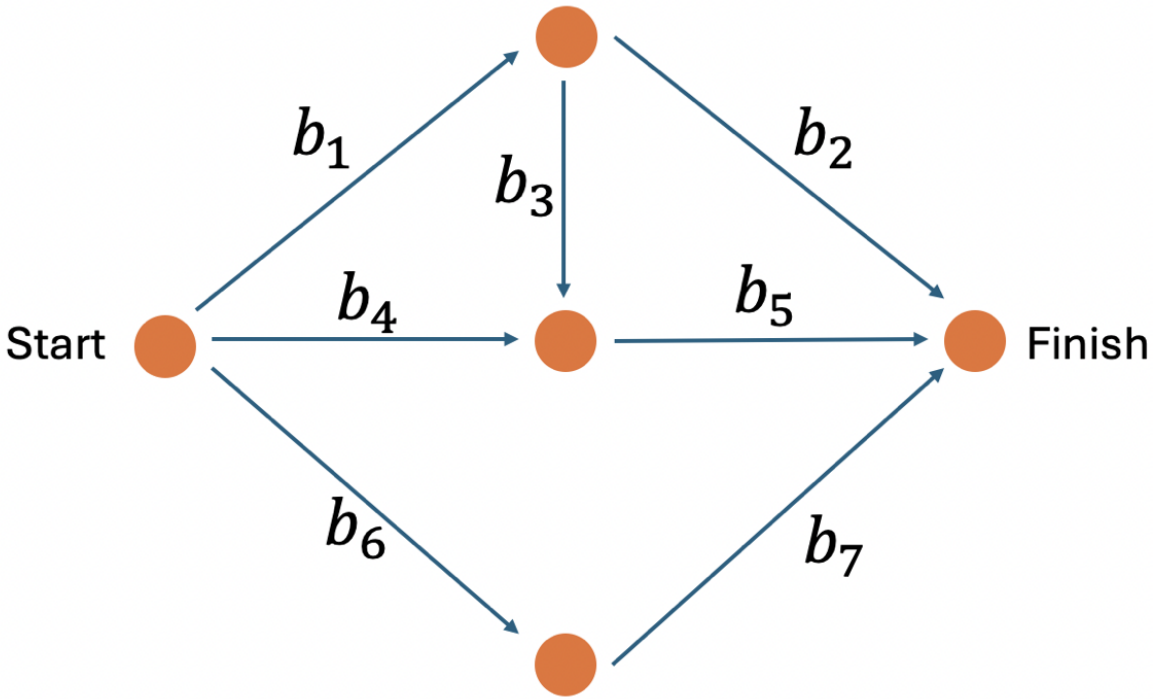
Stochastic Multi-armed Bandit

- Suppose there are K actions: $\{a_k, k = 1, ...K\}$.
- Each action yields a reward $r_k \sim \mathcal{P}(\theta_k)$.
- At each time step, the agent chooses one action to play.
- Reward maximization (or equiv. regret minimization): trade-off between exploration and exploitation.
- Best-arm identification under fixed confidence or fixed budget setting: pure exploration.

Stochastic Combinatorial Bandits

- Suppose there are K base arms: $\{a_k, k = 1, ...K\}$.
- The set of actions is now $\mathcal{S} \subseteq 2^{[K]}$. At each time step, the agent chooses a superarm $S \in \mathcal{S}$ to play.
- Either observe reward $R_t(S)$ for the superarm, or reward $r_t^k(S)$ for each base arm in the superarm, or both.
- Superarm reward can be a simple summation of base arm rewards, or a more sophisticated nonlinear function.
- Maximize cumulative reward or finding the best superarm with high confidence.

Stochastic Combinatorial Bandit Example



- Base arms: $\{b_1, b_2, b_3, b_4, b_5, b_6, b_7\}$.
- Super arms: $\{\{b_1, b_2\}, \{b_1, b_3, b_5\}, \{b_4, b_5\}, \{b_6, b_7\}\}$.
- Reward: negative of the cost required to travel via b_k.
- Objective: maximize cumulative reward is like finding the path that takes the least amount of time.

Bandit for Feature Selection

- Each feature is a base arm.
- At every iteration, select a subset of features as a superarm. Fit the base learner using the subset of features.
- Observe reward, update each base arm.
- Repeat for a pre-specified number of iterations or till convergence.
- Extension of Durand & Gagne (2014) and Liu & Rockova (2021).

Thompson Sampling

Algorithm: Thompson sampling

Input: Beta prior α_0^k, β_0^k for $k = 1, ..., K$; number of iterations T; data D.

Output: Beta posterior α_T^k, β_T^k for $k = 1, ..., K$.

for $t = 1, \ldots, T$ **do**

Draw $\theta_t^k \sim \text{Beta}(\alpha_{t-1}^k, \beta_{t-1}^k)$ for $k = 1, ..., P$.

$S_t \leftarrow \{k : \theta_t^k \geq 0.5\}$.

Observe $r_t^k \sim Q_{s_t}$, from the reward distribution of S_t.

Update Beta parameters:

for all $k \in S_t$ **do**

$\alpha_t^k \leftarrow \alpha_{t-1}^k + r_t^k$

$\beta_t^k \leftarrow \beta_{t-1}^k + (1 - r_t^k)$

end for

end for

Overview

Linear Model with Isotropic Features

- Consider the linear model

$$y = X_{p,K}\beta_{p,K} + \epsilon$$

where $X_{p,K} = [X(p), X(K)]$, $X(p) \in \mathbb{R}^{n\times p}$ and $X(K) \in \mathbb{R}^{n\times(K-p)}$, and

$$\beta_{p,K} = \begin{pmatrix} \beta^* \\ 0 \end{pmatrix}, \qquad \beta^* \in \mathbb{R}^p$$

- $x_i \sim N(0, I_K)$, $\epsilon_i \sim N(0, \sigma_\epsilon^2)$, $\beta^* = [1, ..., 1]^T$.

Linear Factor Model

- Correlated features, a highly challenging DGP from Meinshausen & Buhlmann (2010) :

$$X_{p,K} = F\phi + \eta$$

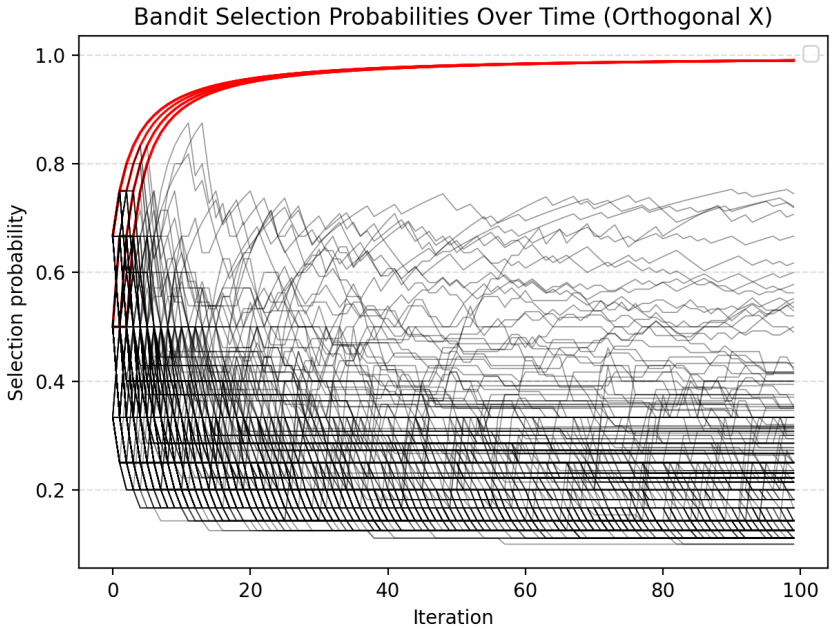
where $F \in \mathbb{R}^{n\times 2}$, $F_i \sim N(0, I_2)$ and $\phi \sim N(0, I_{2\times K})$, $\eta_i \sim N(0, \sigma_\eta^2)$.

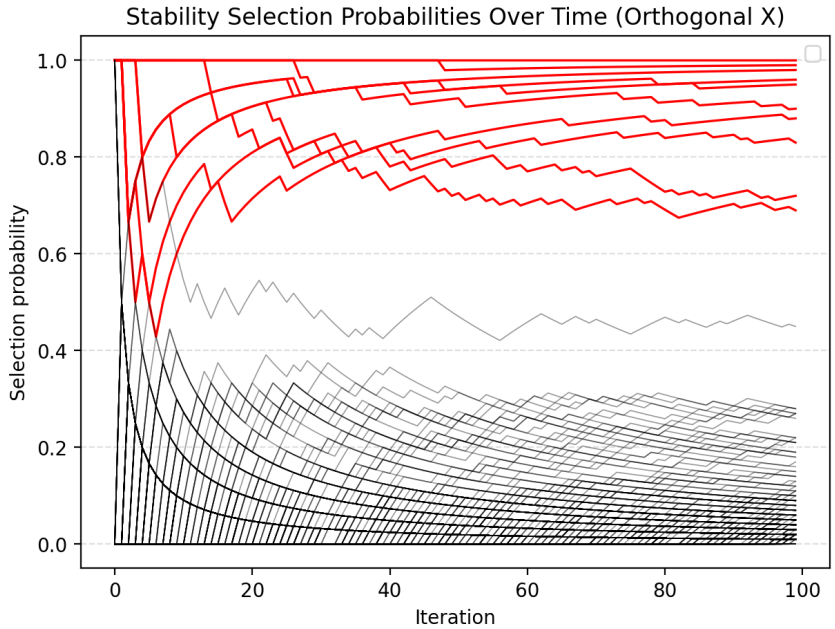
- $y = X_{p,K}\beta_{p,K} + \epsilon$ and $\beta^* = [1, ..., 1]^T$.
- Correlations of features in the range of $[-0.81, 0.80]$, with an average absolute correlation of 0.2 across all pairs of features.

Simulation details

- Sample size $n = 200$.
- Number of features $K = 1000$.
- Number of true signals $p = 10$, noise signals $K - p = 990$.
- Max iterations ranging from 100 to 500.
- Either bootstrap sample 200 or sub-sample 100 in each iteration.

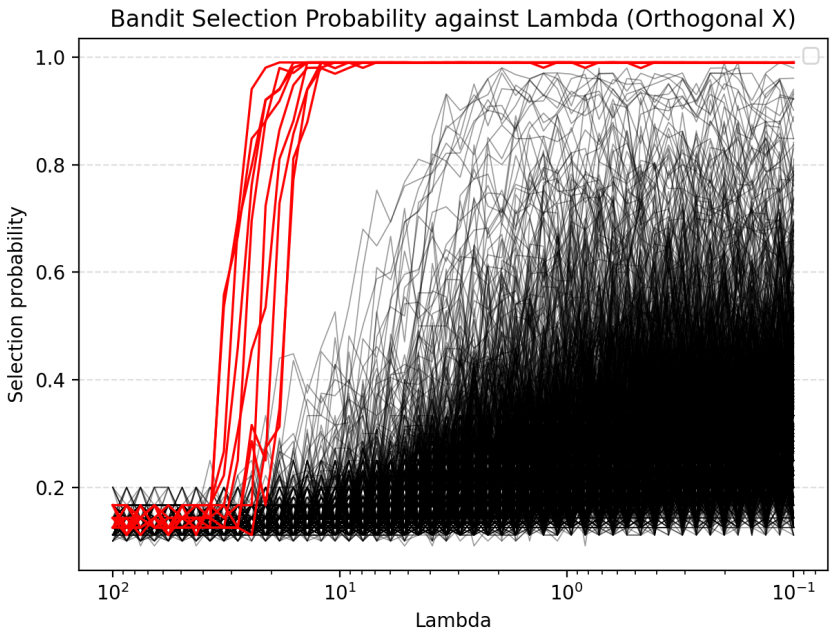
Selection Probabilities for a Fixed λ

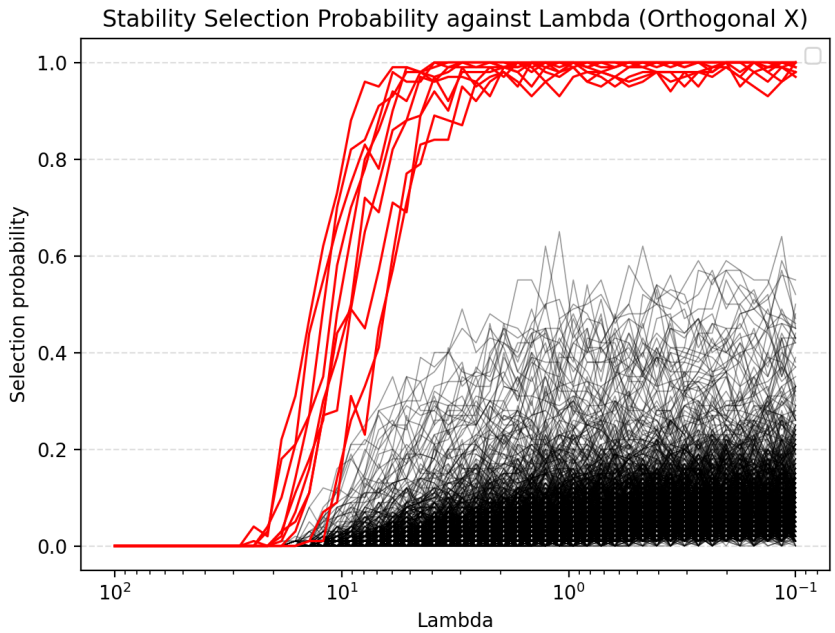




- High false positive rate (FPR) for bandit selection.
- Increasing max iteration doesn't help as selection probabilities converge to the levels shown above.

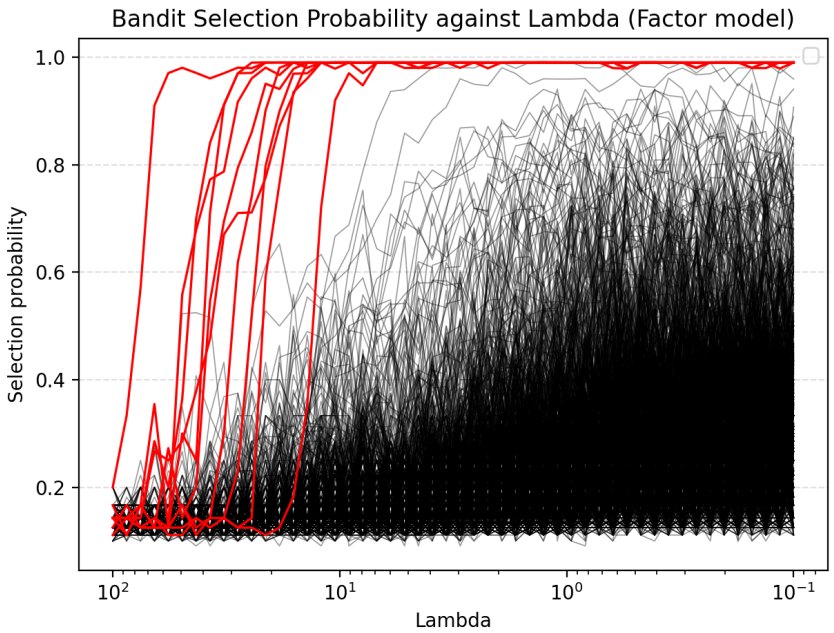
Selection Probabilities over λ's

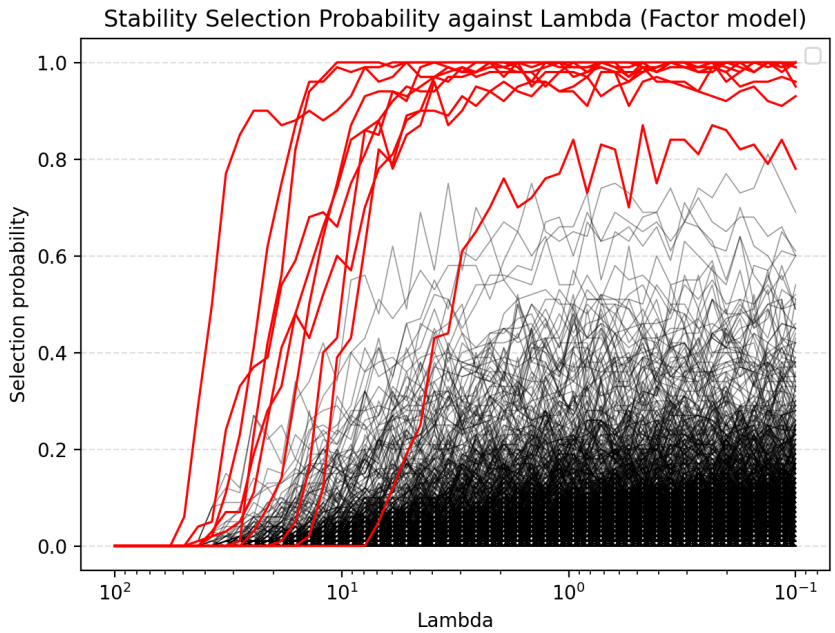




- Comparing bandit vs stability selection across a grid of λ.
- Bandit achieves weak separation between signal and noise, high FPR for a range of λ values, vs stability selection.

Selection Probabilities over λ's in Factor Model





- Both methods attain worse separation in the more challenging factor model.
- Stability selection appears better across λ.

Theoretical Insights

- Sign equality: $sign(\hat{\beta}) = sign(\beta)$, $sign()$ maps positive to 1, negative to -1, and 0 to 0.
- Consider the linear model

$$y = X_{p,K}\beta_{p,K} + \epsilon$$

where $X_{p,K} = [X(p), X(K)]$, $X(p) \in \mathbb{R}^{n\times p}$ and $X(K) \in \mathbb{R}^{n\times(K-p)}$,

$$\beta_{p,K} = \begin{pmatrix} \beta^* \\ 0 \end{pmatrix}, \qquad \beta^* \in \mathbb{R}^p$$

- Let

$$\Sigma^n_{p,K} := \begin{pmatrix} C^n_{11} & C^n_{12} \\ C^n_{21} & C^n_{22} \end{pmatrix} = \frac{1}{n} X^T_{p,K} X_{p,K}$$

denote the sample covariance matrix of $X_{p,K}$ partitioned according to true and noise features.

- Let $\hat{\beta}(\lambda)_{p,K}$ denote the LASSO estimate of y on $[X(p), X(K)]$.

Effect of Dropping Noise Features

Corollary

Assuming that the Irrepresentable Condition (IC) holds for the full design matrix $X_{p,K}$ i.e.

$$|C_{21}^n (C_{11}^n)^{-1} sign(\beta^*)| \leq \mathbf{1} - \eta$$

with a positive constant vector η, then the IC also holds for $X_{p,k}$ where $p \leq k \leq K$ i.e. using all the true features but a subset of noise features. Moreover, the lower bound of the probability of correct sign identification i.e. $\inf P(sign(\hat{\beta}(\lambda)_{p,k}) = sign(\beta_{p,k}))$ *is non-increasing in the number of noise features $(k - p)$.*

- IC is the sufficient and almost necessary condition for consistent LASSO sign identification (Zhao & Yu (2006)).

Dropping True Features might Violate IC

Corollary

Consider the same linear model set-up, let $X_{q,k}$ be the design matrix which contains a subset $q < p$ of true features and a subset $(k - q)$ of noise features.
There exists a distribution $(x_i, \epsilon_i) \sim \mathcal{P}$ and a coefficient vector $\beta_{p,K}$ such that, given a sample of size n drawn from $\mathcal{P}$, the sample covariance matrix of the full design matrix $\Sigma^n_{p,K} = \frac{1}{n} X^T_{p,K} X_{p,K}$ satisfies the IC but the sample covariance matrix of the design matrix with omitted true features $X_{q,k}$, where $q < p$, violates the IC.

Effect of Dropping True Features

Proposition

Consider the same linear model set-up, let $X_{q,k}$ be the design matrix which contains a subset $q < p$ of true features and a subset $(k - q)$ of noise features. Let

$$\beta_{q,k} = \begin{pmatrix} \tilde{\beta} \\ 0 \end{pmatrix}, \qquad \tilde{\beta} \in \mathbb{R}^q$$

be the pseudo-truth coefficient vector from omitting some true features. Suppose that both the sample covariance matrix of the full design matrix $X_{p,K}$ and the sample covariance matrix with omitted features $X_{q,k}$ satisfy the IC, there exists a distribution $(x_i, \epsilon_i) \sim \mathcal{P}$ and a coefficient vector $\beta_{p,K}$ such that, given a sample of size n drawn from $\mathcal{P}$,

$$\inf P(sign(\hat{\beta}_{q,k}) = sign(\beta_{q,k})) < \inf P(sign(\hat{\beta}_{p,K}) = sign(\beta_{p,K}))$$

Overview

Strongly Identifiable Assumption

- Sufficient Condition for Bandit Selection consistency using Thompson sampling (Liu & Rockova (2021)):

Assumption

Define $S^* := argmax_S E[R_t(S)]$ *as the optimal superarm.* S^* *is strongly identifiable if* $\exists \alpha$ *where* $0 < \alpha < 0.5$ *such that:*

- $\forall k \in S^*$, $E[r_t^k | S^*] \geq E[r_t^k | S_t] > 0.5 + \alpha$ *for all* S_t *which contains k, and for all t.*
- $\forall k \notin S^*$, $E[r_t^k | S_t] < 0.5 - \alpha$ *for all* S_t *which contains k, and for all t.*

- Whenever $k \in S^*$ is played, probability of positive reward is greater than 0.5. Whenever $k \notin S^*$ is played, probability of positive reward is less than 0.5.

Top-two Thompson Sampling for CMAB

- Algorithms in existing work focuses on regret minimization, where the regret is directly indicative of the cost of not selecting the true optimal superarm.
- Often learning occurs before implementation, greater exploration might be preferred to exploitation e.g. experimental regret is negligible relative to simple regret which includes future deployment costs.
- Motivated by top-two Thompson sampling in Russo (2018), we propose the Top-two Thompson sampling for combinatorial bandits.
- Feature selection remains consistent under the strong identifiability assumption.

Top-two TS for CMAB

Algorithm: Top-two Thompson sampling

Input: Beta prior α_0^k, β_0^k for $k = 1, ..., K$; max iterations T; data D; exploration probability δ.

Output: Beta posterior α_T^k, β_T^k for $k = 1, ..., K$.

for $t = 1, \ldots, T$ **do**
  Draw $\theta_t^k \sim \text{Beta}(\alpha_{t-1}^k, \beta_{t-1}^k)\ \forall k$. Draw $u_t \sim U[0, 1]$.
  $S_t \leftarrow \{k : \theta_t^k \geq 0.5\}$.
  if $u_t \leq \delta$ **then**
    Draw S_t' in the same way as S_t until $S_t' \neq S_t$.
    $S_t \leftarrow (S_t \cup S_t') \setminus (S_t \cap S_t')$
  end if
  Observe $r_t^k \sim Q_{s_t}$ from the reward distribution of S_t.
  for all $k \in S_t$, **do**
    Update α_t^k, β_t^k
  end for
end for

Top-two TS for CMAB

- Reward function can be adapted to the base learner, e.g. tree split probability, permutation importance, SHAP.
- Allocates more measurement effort to explore uncertain regions of the feature space.
- Faster concentration of posterior mass on the optimal set.

Top-two TS for CMAB Consistency

Proposition (Posterior concentration)

Under the strong identifiability assumption, in the top-two Thompson sampling for CMAB algorithm, the following holds for each base arm k:

1. *The number of times it is played at time t satisfies:*

$$N_k(t, \delta) \to \infty \ a.s.$$

2. *The posterior probability of selection converges to:*

$$\Pi_t(\{\theta_k \geq 0.5\}) \to \mathbf{1}\{i \in S^*\} \ a.s.$$

where θ_k is posterior mean. This implies that $\Pi_t(S^*) \to 1$ a.s.

3. *Moreover, there exists $c_k > 0$ and a finite $T_k > 0$ s.t. for all $t > T_k$*

$$\Pi_t(\{k \text{ misclassified}\}) \leq e^{-c_k N_k(t,\delta)}$$

.

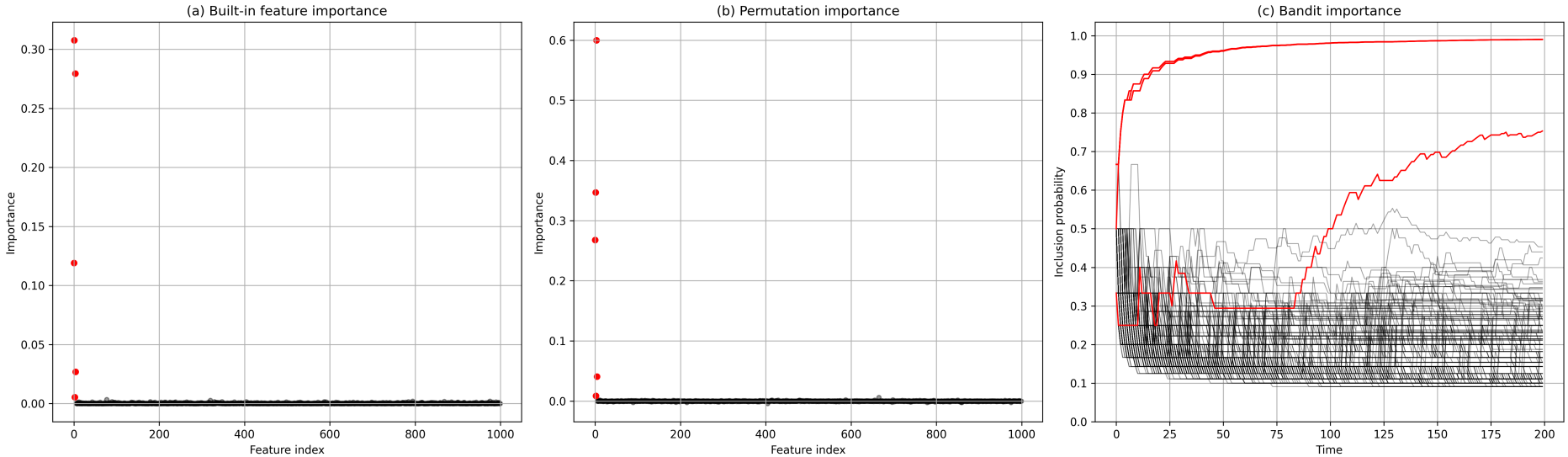
Proof Sketch

1. Proof by contradiction. Suppose $N_k(t, \delta)$ is finite, then posterior stops getting updated for k. But our sampling rule implies that k has a non-zero probability of being played.
2. Established by the strong identifiability condition and using martingale strong law.
3. The exponential rate of convergence depending on $N_k(t, \delta)$ comes from the Beta distribution exponential tail KL-divergence type bound.

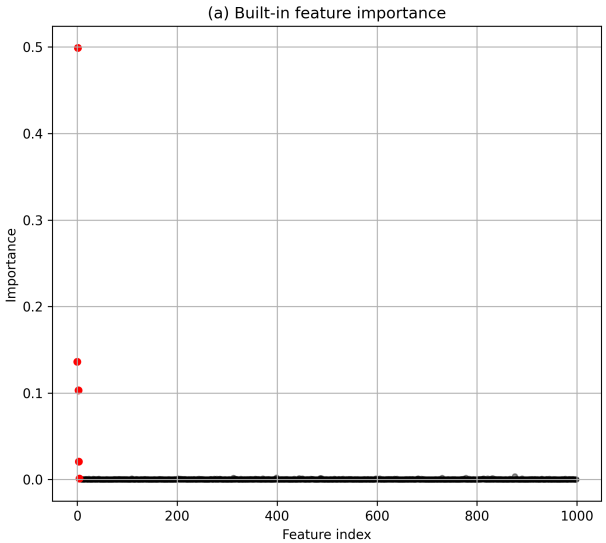
Simulation Set-up

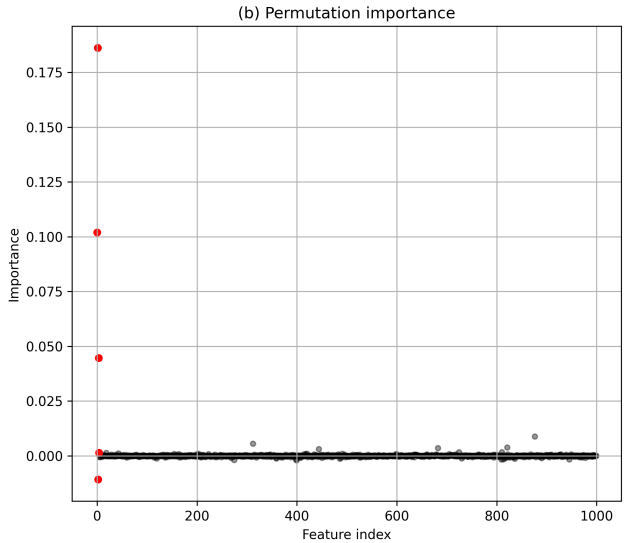
- Two nonlinear DGPs (Friedman):
- (1) $y_i = 10\sin(\pi X_{i,1} X_{i,2}) + 20(X_{i,3} - 0.5)^2 + 10X_{i,4} + 5X_{i,5} + \epsilon_i$
- (2) $y_i = 0.1e^{4x_{i,1}} + \frac{4}{1+e^{-20(x_{i,2}-0.5)}} + 4x_{i,3} + 3x_{i,4} + 2x_{i,5} + \epsilon_i$
- $X_i \sim Unif[0,1]^p$, $\epsilon \sim N(0, 0.5^2)$
- n = 300, p = 1000, random forest as base learner, permutation importance as reward.

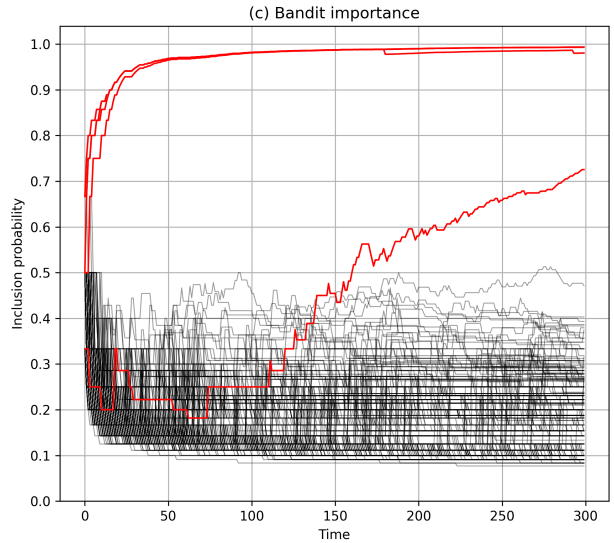
Simulation Results (1)



Simulation Results (2)



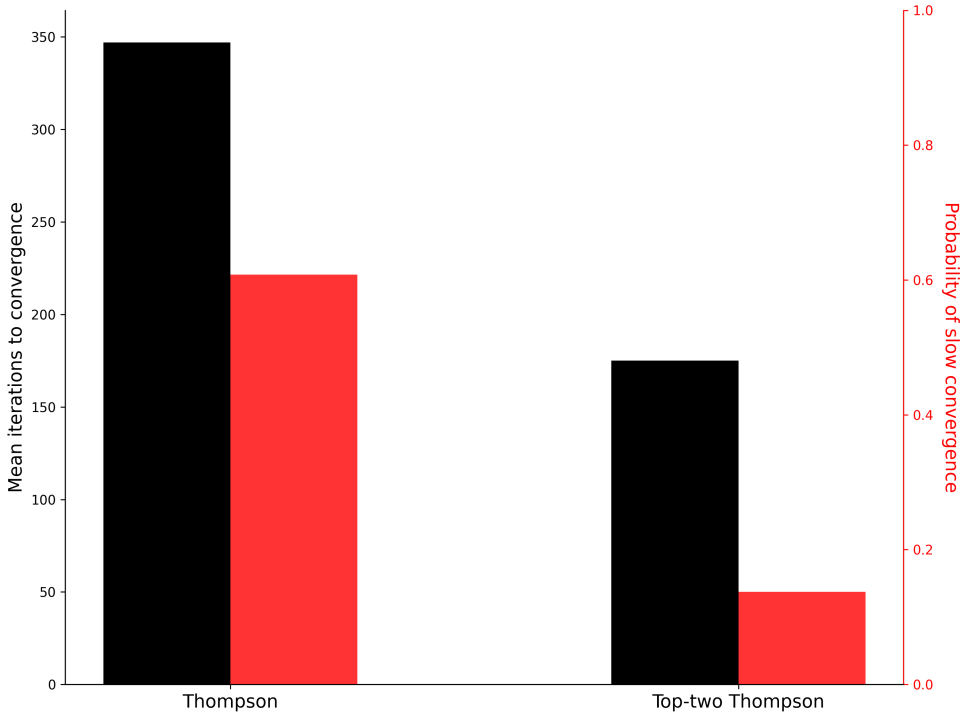


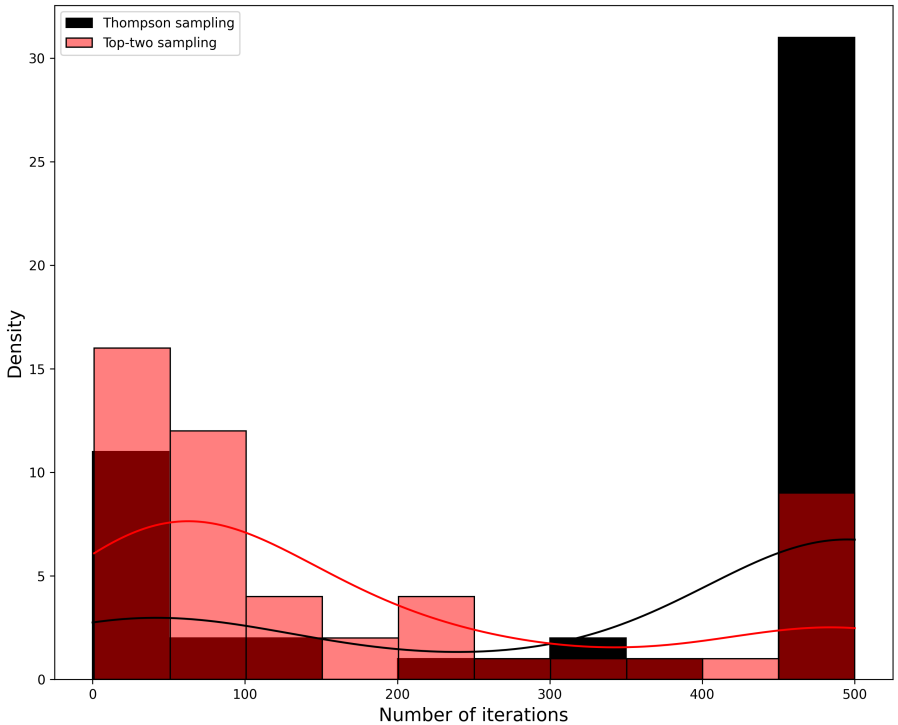


Simulation Comparing Convergence Rate

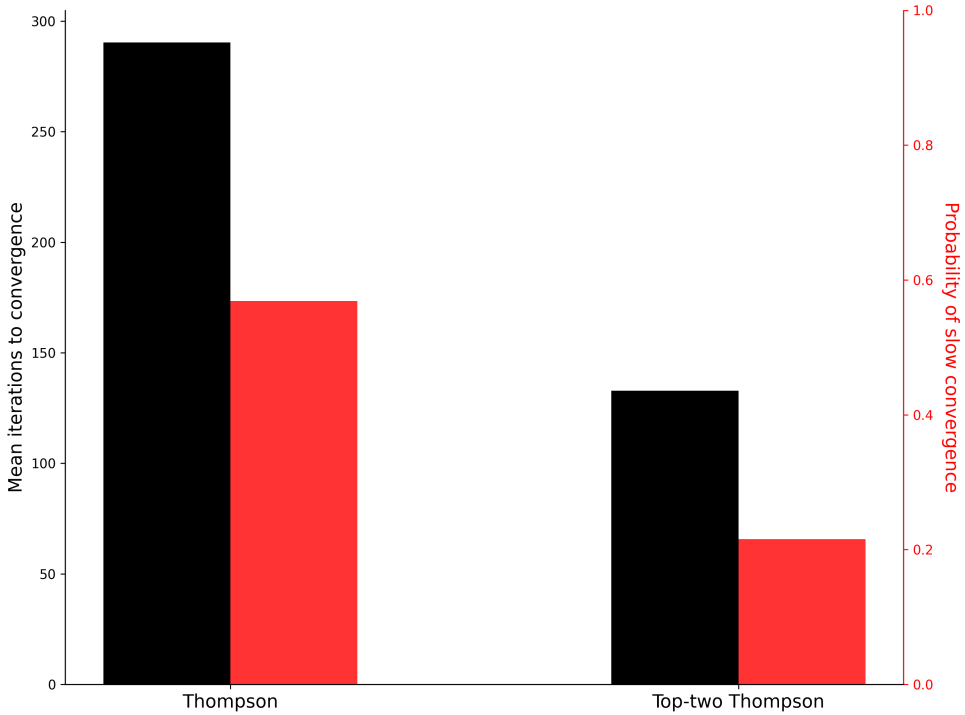
- Data generated from the same DGPs (1) and (2).
- Compare Thompson sampling versus top-two Thompson sampling.
- Convergence defined as all true features having posterior probabilities above 0.5 and all noise features having posterior probabilities below 0.5.
- Cap the max number of iterations at 500.
- Repeat over 50 trials.

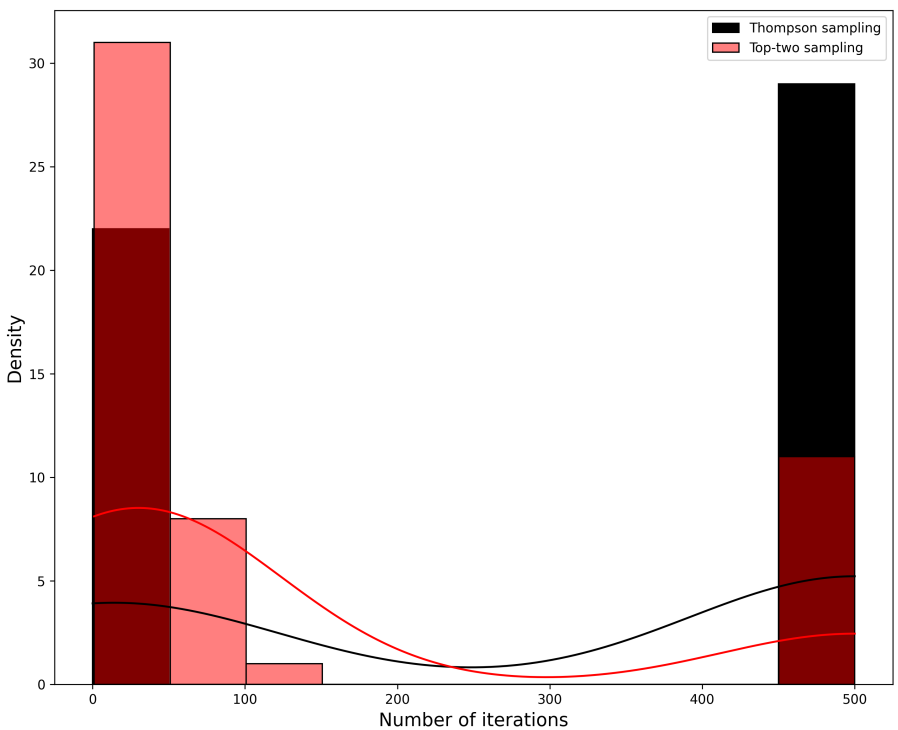
Convergence Results (1)





Convergence Results (2)





Overview

When Bandit Selection Helps

- More often than not, bandit algorithms are used in online settings.
- Full model evaluation is computationally expensive or intractable.
- Exploration budget is limited, need to adaptively allocate computation toward promising variables.
- Heterogeneous signal strength, weak signals difficult to be detected in the presence of strong signals given limited sample.

How Else to Do Bandit Selection

- K base arms, each associated with feature importance $X_{k,t}$ which is supported on $[0, 1]$ (can be relaxed).
- $X_{k,t}$ of different base arms may be dependent.
- Suppose we have a large feature space and feature evaluation is expensive, adaptively allocate evaluation to promising features while minimizing the cost of make wrong feature selection.
- UCB class of algorithms provides an alternative path from Thompson sampling based algorithm.

Reward assumptions

- A different set of reward assumptions.
- Expected reward requires monotonicity and bounded smoothness.

Assumption

1. $\forall k \in [K], \mu_k \leq \mu'_k$ *implies* $R_\mu(S) \leq R_{\mu'}(S)$ *for all* S.
2. *There exists a continuous strictly increasing* f *with* $f(0) = 0$ *s.t. for any* μ *and* μ' *and for any* $\Lambda > 0$, $\max_{i \in S} |\mu_i - \mu'_i| \leq \Lambda$ *implies* $|r_\mu(S) - r_{\mu'}(S)| \leq f(\Lambda)$ *for all* S.

Combinatorial Upper Confidence Bound Algorithm

- Based on Chen et al. (2016).

CUCB Feature Selection

Initiate: variable $T_k \leftarrow 0$ as the total number of times base arm k is played; variable $\hat{\mu}_k \leftarrow 1$ as the mean outcomes of arm k so far.
Output: $\hat{\mu}_k$ for all $k = 1, ...K$.
$t \leftarrow 0$

while true **do**
  $t \leftarrow t + 1$
  for all k **do**
    Set $\bar{\mu}_i = \min\{\hat{\mu}_i + \sqrt{\frac{3 \ln t}{2 T_k}}, \ 1\}$
  end for $S \leftarrow Oracle(\bar{\mu}_1, ..., \bar{\mu}_K)$
  Observe outcome from playing S and update T_k and $\hat{\mu}_k$ for all arms.
end while

CUCB for Feature Selection

- Suppose budget limits to you sample m features, oracle can be the top m features.
- Reward function can be model dependent or model agnostic, such as relative permutation importance which is bounded on $[0, 1]$.
- Other arbitrary oracle depends on your task objective.
- Chen et al. (2016) shows that the regret bound of CUCB is bounded by $O(\log T)$ given the two reward assumptions.

Overview

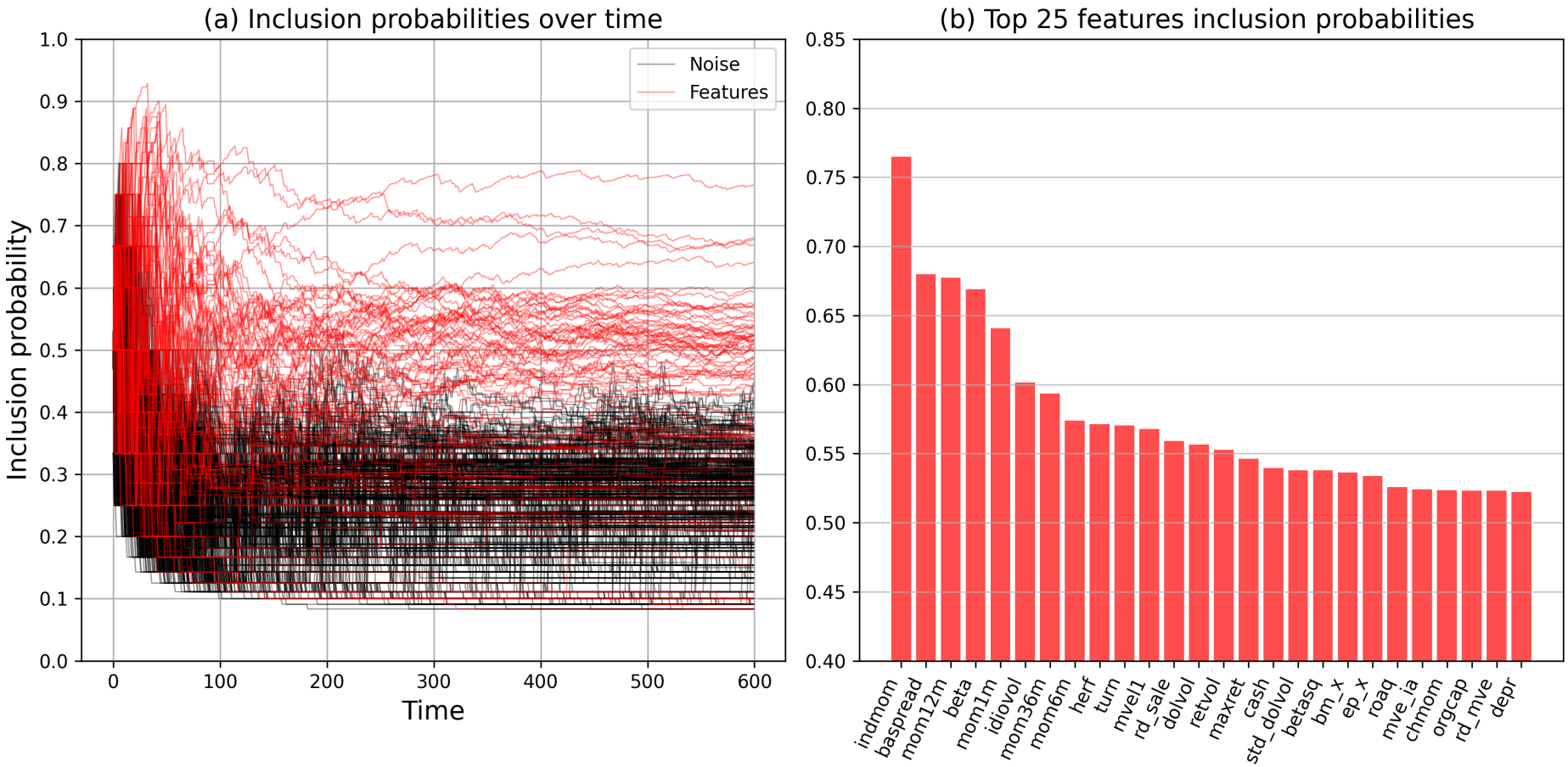
Understanding Empirical Asset Pricing

- Asset pricing using ML conducted by Gu, Kelly and Xiu (2020) on 30,000 stocks in the US between 1957 and 2016 (later extended to 2019).
- Compile a large collection of 94 stock level features covering characteristics related to momentum, liquidity, volatility, valuation, and fundamentals, and 8 macro predictors.
- Attained predictive R^2 of $0.34 - 0.40\%$ and $3.09 - 3.60\%$ for monthly and annual stock returns respectively.

My Set-up

- Online data rolling in monthly from 2010 to 2019.
- Top-two Thompson sampling with gradient boosting regressor as base learner, 5 iterations per month, totalling 600 iterations over 120 months from 2010 - 2019.
- Permutation importance as reward. Checked correlations to be low. Binary reward defined as 2% relative drop in predictive R^2.
- Artificially draw 5000 noise features on the same support as the true features to increase dimensionality, giving approximately 5000 stocks and 5100 features per month.
- Compare my feature selection to that of Gu, Kelly and Xiu (2020).

Empirical Results



Thank you.