KAIST AI
Kim Jaechul Graduate School

Optimization and Statistical Inference LAB

OptiML
Optimization & Machine Learning

JEONBUK NATIONAL UNIVERSITY 1947 KOREA

KSC 2025 Oral Session

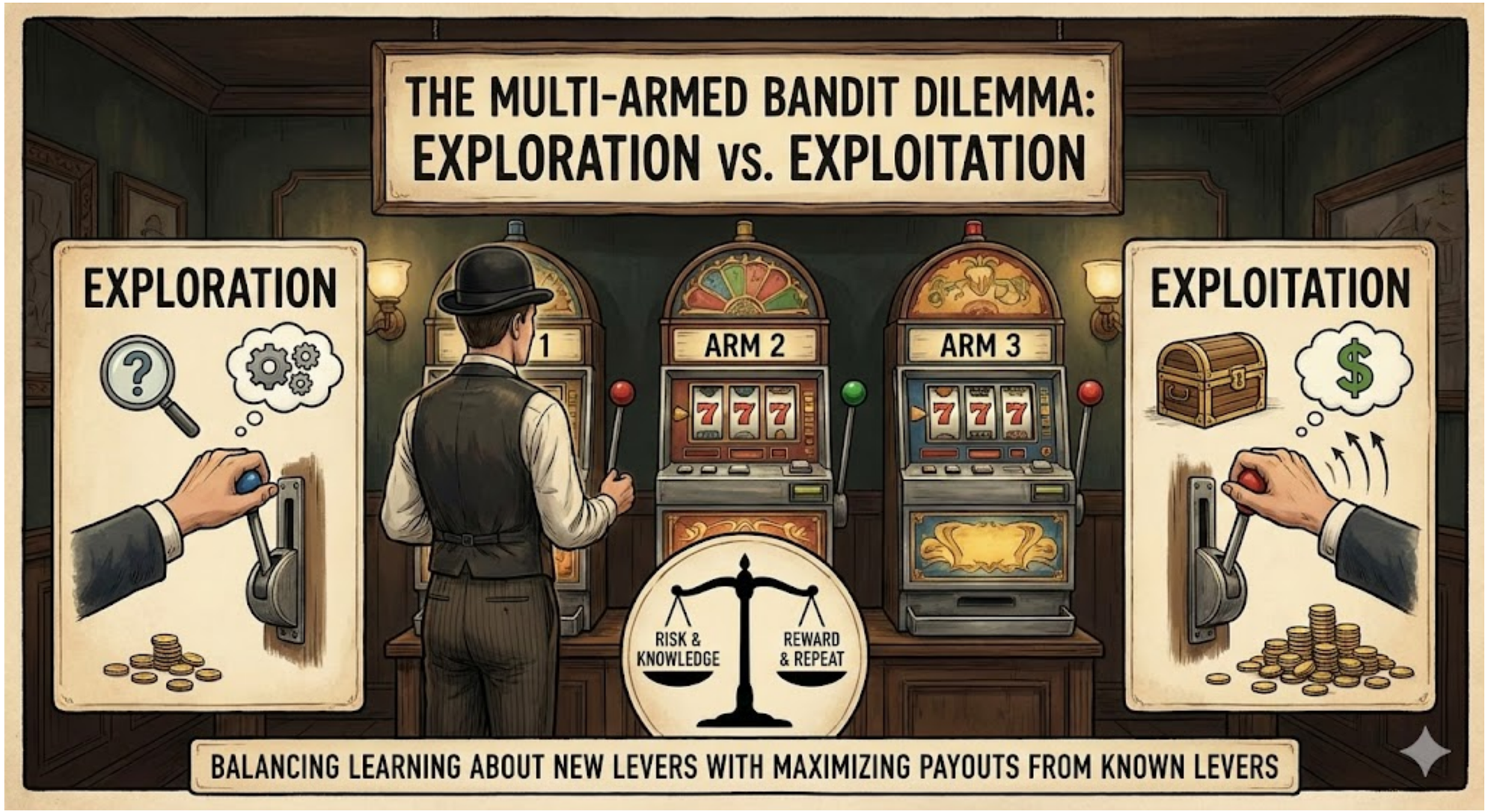
Preliminary Empirical Study of Low-Rank, Hierarchical Gaussian Linear Bandits

Junghyun Lee* (KAIST AI), Sanghwa Kim* (KAIST AI),
Gwangsu Kim (JBNU Statistics), Se-Young Yun (KAIST AI)

Bandits

Nano banana

Bandits (= MDP with a Single State)



Nano banana

Multi-Task Bandits

Solving multiple bandit instances (“tasks”) **simultaneously**

- **Task similarity:** The tasks share some “common structure” => faster learning!

Nano banana

Multi-Task Gaussian Linear Bandits

There are M bandit instances, each parametrized by $\theta_i \in \mathbb{R}^d$.

The user interacts with M instances **simultaneously** as follows:

for $t = 1,2,\cdots$

- Choose arms $\{x_{t,i}\}_{i\in[M]}$ for each task $i \in [M]$ **simultaneously**
- Observe the task-wise reward $r_{t,i} \sim \mathcal{N}(\langle x_{t,i}, \theta_i \rangle, \sigma^2)$

Question. How to formalize the "common structure" across θ_i's?

Nano banana

Two Approaches to Multi-Task Linear Bandits

Frequentist Approach. The parameters lie in a common low-dimensional space!

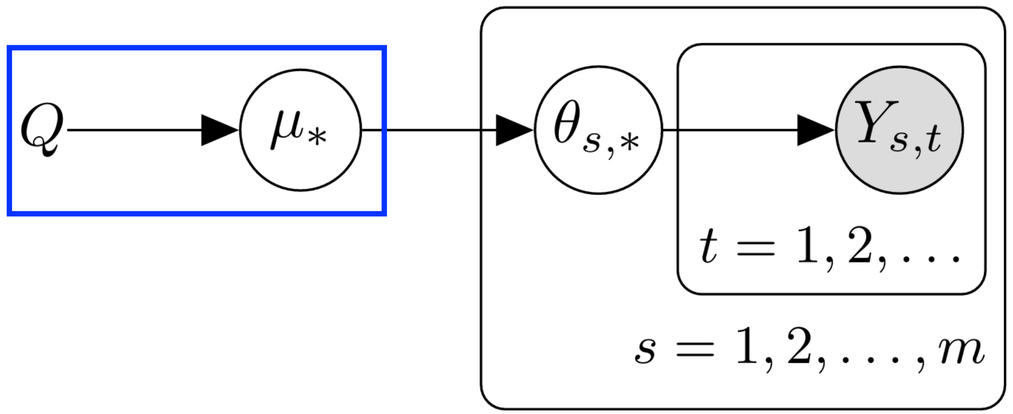
[Hu+21]

$$\theta_i = B w_{i,*} \qquad Y_{i,t} \sim \mathcal{N}(\langle X_{i,t}, \theta_i \rangle, \sigma^2)$$

d-dim task vector; $d \times k$ linear projector; k-dim latent task vector

Bayesian Approach. The parameters are sampled from a common prior

[Hong+22]



A New Problem Setting :)

Question. Can we combine those two approaches?

Yes! **Low-rank Hierarchical Gaussian Linear Bandits (Lr-HGLB)**

- Start from the Frequentist model
- Place independent priors on B *and* $w_{s,*}$, i.e., full hierarchical Bayesian model for the low-rank structure as well

---

Probabilistic Matrix Factorization

NIPS 2007

Ruslan Salakhutdinov and Andriy Mnih
Department of Computer Science, University of Toronto
6 King's College Rd, M5S 3G4, Canada
{rsalakhu,amnih}@cs.toronto.edu

Bayesian Probabilistic Matrix Factorization
using Markov Chain Monte Carlo

ICML 2008

Ruslan Salakhutdinov RSALAKHU@CS.TORONTO.EDU
Andriy Mnih AMNIH@CS.TORONTO.EDU
Department of Computer Science, University of Toronto, Toronto, Ontario M5S 3G4, Canada

Lr-HGLB

The environment is sampled as follows: with a known rank $k \leq d$,

- $\mu_* \sim \mathcal{N}_k(\mu_q, \Sigma_q)$ (referred to as *hyper-prior*)

k-dim *latent* task vector

- $w_{i,*} \sim \mathcal{N}_k(\mu_*, \Sigma_0)$, i.i.d. over $i \in [M]$

$d \times k$ linear projector

- $B_{m,n} \sim \mathcal{N}(\mu_B, \sigma_B^2)$, i.i.d. over $1 \leq m \leq d, 1 \leq n \leq k$

d-dim task vector

- $\theta_i := Bw_{i,*}$, $r_{t,i} \sim \mathcal{N}(\langle x_t, \theta_i \rangle, \sigma^2)$

Interaction Protocol, Bayes Regret

Interaction Protocol. At the beginning, sample an environment. Then, for $t = 1,2,\cdots$

- Choose arms $\{x_{t,i}\}_{i\in[M]}$ for each task $i \in [M]$ **simultaneously**
- Observe the task-wise reward $r_{t,i} \sim \mathcal{N}(\langle x_{t,i}, \theta_i \rangle, \sigma^2)$

Interaction Protocol, Bayes Regret

Interaction Protocol. At the beginning, sample an environment. Then, for $t = 1,2,\cdots$

- Choose arms $\{x_{t,i}\}_{i\in[M]}$ for each task $i \in [M]$ **simultaneously**
- Observe the task-wise reward $r_{t,i} \sim \mathcal{N}(\langle x_{t,i}, \theta_i \rangle, \sigma^2)$

The performance of an algorithm is measured via *total Bayes regret* [14]:

$$BReg(T) := \mathbb{E}\left[\sum_{t=1}^{T}\sum_{i=1}^{M}\langle x_i^\star - x_{t,i}, \theta_i\rangle\right], \quad x_i^\star := \operatorname*{argmax}_{x\in\mathcal{A}}\langle x, \theta_i\rangle,$$

where the expectation $\mathbb{E}$ is taken with respect to the randomness of the environment and the algorithm.

Thompson Sampling for Lr-HGLB

Thompson Sampling for Lr-HGLB

In such Bayesian bandit scenario, natural algorithm is **Thompson sampling:**

Thompson Sampling for Lr-HGLB

In such Bayesian bandit scenario, natural algorithm is **Thompson sampling:**

- **Sample** an environment from the current posterior

Thompson Sampling for Lr-HGLB

In such Bayesian bandit scenario, natural algorithm is **Thompson sampling:**

- **Sample** an environment from the current posterior
- Choose arms $\{x_{t,i}\}_{i\in[M]}$ for each task $i \in [M]$ **simultaneously,** greedily from the sampled environment

Thompson Sampling for Lr-HGLB

In such Bayesian bandit scenario, natural algorithm is **Thompson sampling:**

- **Sample** an environment from the current posterior
- Choose arms $\{x_{t,i}\}_{i\in[M]}$ for each task $i \in [M]$ **simultaneously,** greedily from the sampled environment
- Observe the task-wise reward $r_{t,i} \sim \mathcal{N}(\langle x_{t,i}, \theta_i \rangle, \sigma^2)$ and **update** posterior

Thompson Sampling for Lr-HGLB

In such Bayesian bandit scenario, natural algorithm is **Thompson sampling:**

- **Sample** an environment from the current posterior
- Choose arms $\{x_{t,i}\}_{i\in[M]}$ for each task $i \in [M]$ **simultaneously,** greedily from the sampled environment
- Observe the task-wise reward $r_{t,i} \sim \mathcal{N}(\langle x_{t,i}, \theta_i \rangle, \sigma^2)$ and **update** posterior

But, the joint conjugacy of $(B, \{w_i\}_{i\in[M]})$ fails! => **intractable** to sample/update

Thompson Sampling for Lr-HGLB

In such Bayesian bandit scenario, natural algorithm is **Thompson sampling:**

- **Sample** an environment from the current posterior
- Choose arms $\{x_{t,i}\}_{i\in[M]}$ for each task $i \in [M]$ **simultaneously,** greedily from the sampled environment
- Observe the task-wise reward $r_{t,i} \sim \mathcal{N}(\langle x_{t,i}, \theta_i \rangle, \sigma^2)$ and **update** posterior

But, the joint conjugacy of $(B, \{w_i\}_{i\in[M]})$ fails! => **intractable** to sample/update

$$B \rightarrow r_{t,i} \leftarrow w_{i,*}$$ => *posterior dependency between* B *and* $w_{i,*}$

Gibbs Sampling to the Rescue!

Gibbs Sampling to the Rescue!

Still, we can utilize the product structure, namely, *coordinate-wise* update:

Gibbs Sampling to the Rescue!

Still, we can utilize the product structure, namely, *coordinate-wise* update:

- **Conditioned on** B, each w_i has a closed-form Gaussian posterior

Gibbs Sampling to the Rescue!

Still, we can utilize the product structure, namely, *coordinate-wise* update:

- **Conditioned on** B, each w_i has a closed-form Gaussian posterior
- **Conditioned on** $\{w_i\}_{i\in[M]}$, B has a closed-form Gaussian posterior

Gibbs Sampling to the Rescue!

Still, we can utilize the product structure, namely, *coordinate-wise* update:

- **Conditioned on** B, each w_i has a closed-form Gaussian posterior
- **Conditioned on** $\{w_i\}_{i\in[M]}$, B has a closed-form Gaussian posterior

In Bayesian statistics, this is known as **Gibbs Sampling**

IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL. PAMI-6, NO. 6, NOVEMBER 1984 721

Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images

STUART GEMAN AND DONALD GEMAN

IEEE TPAMI 1984

Sampling-Based Approaches to Calculating Marginal Densities

ALAN E. GELFAND AND ADRIAN F. M. SMITH*

Journal of the American Statistical Association (JASA) 1990

Gibbs, Low-rank Hierarchical Thompson Sampling

for $t = 1,2, \ldots$ **do**

Sample $\hat{b}_t \sim \mathcal{N}_k\left(\tilde{\mu}_t^B, \tilde{\Sigma}_t^B\right)$ and set $\hat{B}_t \leftarrow \mathrm{vec}^{-1}\left(\hat{b}_t\right)$

Sample $\mu_t \sim Q_t = \mathcal{N}_k(\bar{\mu}_t, \bar{\Sigma}_t)$;

for $i = 1,2, \ldots, M$ **do**

Task-wise posterior of w's:

$$\tilde{\Sigma}_{t,i}^w \leftarrow \left(\Sigma_0^{-1} + \hat{B}_t^T V_{t,i} \hat{B}_t\right)^{-1}, \qquad \tilde{\mu}_{t,i}^w \leftarrow \tilde{\Sigma}_{t,i}^w \left(\Sigma_0^{-1} \mu_t + \hat{B}_t^T y_{t,i}\right)$$

Sample $\hat{w}_{t,i} \sim \mathcal{N}_k\left(\tilde{\mu}_{t,i}^w, \tilde{\Sigma}_{t,i}^w\right)$;

Choose action: $x_{t,i} \leftarrow \underset{x \in \mathcal{A}}{\mathrm{argmax}} \langle \hat{B}_t^T x, \hat{w}_{t,i} \rangle$;

Receive a reward $r_{t,i}$

Update task-wise matrices:

$$V_{t+1,i} \leftarrow V_{t,i} + x_{t,i} x_{t,i}^T, \qquad y_{t+1,i} \leftarrow y_{t,i} + r_{t,i} x_{t,i}$$

Hyper-posterior: denoting $\Lambda_{t+1,i} \coloneqq \left(\hat{B}_t^T V_{t+1,i} \hat{B}_t\right)^{-1}$,

$$\bar{\Sigma}_{t+1} \leftarrow \left(\Sigma_q^{-1} + \sum_{i=1}^{M} \left(\Sigma_0 + \sigma^2 \Lambda_{t+1,i}\right)^{-1}\right)^{-1}$$

$$\bar{\mu}_{t+1} \leftarrow \bar{\Sigma}_{t+1} \left(\Sigma_q^{-1} \mu_q + \sum_{i=1}^{M} \left(\Sigma_0 + \sigma^2 \Lambda_{t+1,i}\right)^{-1} \Lambda_{t+1,i} \hat{B}_t^T y_{t+1,i}\right)$$

Posterior of vec(B):

$$\tilde{\Sigma}_{t+1}^B \leftarrow \left(\sigma_B^{-2} I_{dk} + \sigma^{-2} \sum_{i=1}^{M} \left(\hat{w}_{t,i} \hat{w}_{t,i}^T\right) \otimes V_{t+1,i}\right)^{-1}$$

$$\tilde{\mu}_{t+1}^B \leftarrow \tilde{\Sigma}_{t+1}^B \left(\sigma_B^{-2} \mu_B 1_{dk} + \sigma^{-2} \sum_{i=1}^{M} \mathrm{vec}\left(y_{t+1,i} \hat{w}_{t,i}^T\right)\right)$$

Gibbs, Low-rank Hierarchical Thompson Sampling

for $t = 1,2, \ldots$ **do**

Sample $\hat{b}_t \sim \mathcal{N}_k\left(\tilde{\mu}_t^B, \tilde{\Sigma}_t^B\right)$ and set $\hat{B}_t \leftarrow \mathrm{vec}^{-1}\left(\hat{b}_t\right)$

Sample $\mu_t \sim Q_t = \mathcal{N}_k(\bar{\mu}_t, \bar{\Sigma}_t)$;

for $i = 1,2, \ldots, M$ **do** **Posterior of w_i's, *conditioned* on B**

Task-wise posterior of w's:

$$\tilde{\Sigma}_{t,i}^w \leftarrow \left(\Sigma_0^{-1} + \hat{B}_t^T V_{t,i} \hat{B}_t\right)^{-1}, \qquad \tilde{\mu}_{t,i}^w \leftarrow \tilde{\Sigma}_{t,i}^w \left(\Sigma_0^{-1} \mu_t + \hat{B}_t^T y_{t,i}\right)$$

Sample $\hat{w}_{t,i} \sim \mathcal{N}_k\left(\tilde{\mu}_{t,i}^w, \tilde{\Sigma}_{t,i}^w\right)$;

Choose action: $x_{t,i} \leftarrow \underset{x \in \mathcal{A}}{\mathrm{argmax}} \langle \hat{B}_t^T x, \hat{w}_{t,i} \rangle$;

Receive a reward $r_{t,i}$

Update task-wise matrices:

$$V_{t+1,i} \leftarrow V_{t,i} + x_{t,i} x_{t,i}^T, \qquad y_{t+1,i} \leftarrow y_{t,i} + r_{t,i} x_{t,i}$$

Hyper-posterior: denoting $\Lambda_{t+1,i} := \left(\hat{B}_t^T V_{t+1,i} \hat{B}_t\right)^{-1}$,

$$\bar{\Sigma}_{t+1} \leftarrow \left(\Sigma_q^{-1} + \sum_{i=1}^{M} \left(\Sigma_0 + \sigma^2 \Lambda_{t+1,i}\right)^{-1}\right)^{-1}$$

$$\bar{\mu}_{t+1} \leftarrow \bar{\Sigma}_{t+1} \left(\Sigma_q^{-1} \mu_q + \sum_{i=1}^{M} \left(\Sigma_0 + \sigma^2 \Lambda_{t+1,i}\right)^{-1} \Lambda_{t+1,i} \hat{B}_t^T y_{t+1,i}\right)$$

Posterior of vec(B):

$$\tilde{\Sigma}_{t+1}^B \leftarrow \left(\sigma_B^{-2} I_{dk} + \sigma^{-2} \sum_{i=1}^{M} \left(\hat{w}_{t,i} \hat{w}_{t,i}^T\right) \otimes V_{t+1,i}\right)^{-1}$$

$$\tilde{\mu}_{t+1}^B \leftarrow \tilde{\Sigma}_{t+1}^B \left(\sigma_B^{-2} \mu_B 1_{dk} + \sigma^{-2} \sum_{i=1}^{M} \mathrm{vec}\left(y_{t+1,i} \hat{w}_{t,i}^T\right)\right)$$

Posterior of B, *conditioned* on w_i's

Gibbs, Low-rank Hierarchical Thompson Sampling

for $t = 1,2,\ldots$ **do** Sample environment from current posterior

Sample $\hat{b}_t \sim \mathcal{N}_k\left(\tilde{\mu}_t^B, \tilde{\Sigma}_t^B\right)$ and set $\hat{B}_t \leftarrow \mathrm{vec}^{-1}\left(\hat{b}_t\right)$

Sample $\mu_t \sim Q_t = \mathcal{N}_k(\bar{\mu}_t, \bar{\Sigma}_t)$;

for $i = 1,2,\ldots,M$ **do** Posterior of w_i's, *conditioned* on B

Task-wise posterior of w's:

$$\tilde{\Sigma}_{t,i}^w \leftarrow \left(\Sigma_0^{-1} + \hat{B}_t^T V_{t,i} \hat{B}_t\right)^{-1}, \qquad \tilde{\mu}_{t,i}^w \leftarrow \tilde{\Sigma}_{t,i}^w \left(\Sigma_0^{-1}\mu_t + \hat{B}_t^T y_{t,i}\right)$$

Sample $\hat{w}_{t,i} \sim \mathcal{N}_k\left(\tilde{\mu}_{t,i}^w, \tilde{\Sigma}_{t,i}^w\right)$;

Choose action: $x_{t,i} \leftarrow \underset{x \in \mathcal{A}}{\mathrm{argmax}} \langle \hat{B}_t^T x, \hat{w}_{t,i} \rangle$;

Receive a reward $r_{t,i}$ Sample actions simultaneously

Update task-wise matrices:

$$V_{t+1,i} \leftarrow V_{t,i} + x_{t,i} x_{t,i}^T, \qquad y_{t+1,i} \leftarrow y_{t,i} + r_{t,i} x_{t,i}$$

Hyper-posterior: denoting $\Lambda_{t+1,i} := \left(\hat{B}_t^T V_{t+1,i} \hat{B}_t\right)^{-1}$,

$$\bar{\Sigma}_{t+1} \leftarrow \left(\Sigma_q^{-1} + \sum_{i=1}^{M} \left(\Sigma_0 + \sigma^2 \Lambda_{t+1,i}\right)^{-1}\right)^{-1}$$

$$\bar{\mu}_{t+1} \leftarrow \bar{\Sigma}_{t+1}\left(\Sigma_q^{-1}\mu_q + \sum_{i=1}^{M} \left(\Sigma_0 + \sigma^2 \Lambda_{t+1,i}\right)^{-1} \Lambda_{t+1,i} \hat{B}_t^T y_{t+1,i}\right)$$

Posterior of vec(B):

$$\tilde{\Sigma}_{t+1}^B \leftarrow \left(\sigma_B^{-2} I_{dk} + \sigma^{-2} \sum_{i=1}^{M} \left(\hat{w}_{t,i} \hat{w}_{t,i}^T\right) \otimes V_{t+1,i}\right)^{-1}$$

$$\tilde{\mu}_{t+1}^B \leftarrow \tilde{\Sigma}_{t+1}^B \left(\sigma_B^{-2} \mu_B 1_{dk} + \sigma^{-2} \sum_{i=1}^{M} \mathrm{vec}\left(y_{t+1,i} \hat{w}_{t,i}^T\right)\right)$$

Posterior of B, *conditioned* on w_i's

Experiments

- Oracle HTS vs. HTS vs. **GibbsLr-HTS (ours)**
- We consider two choices of *tempering hyper parameter* $c \in \{0.01, 0.05\}$

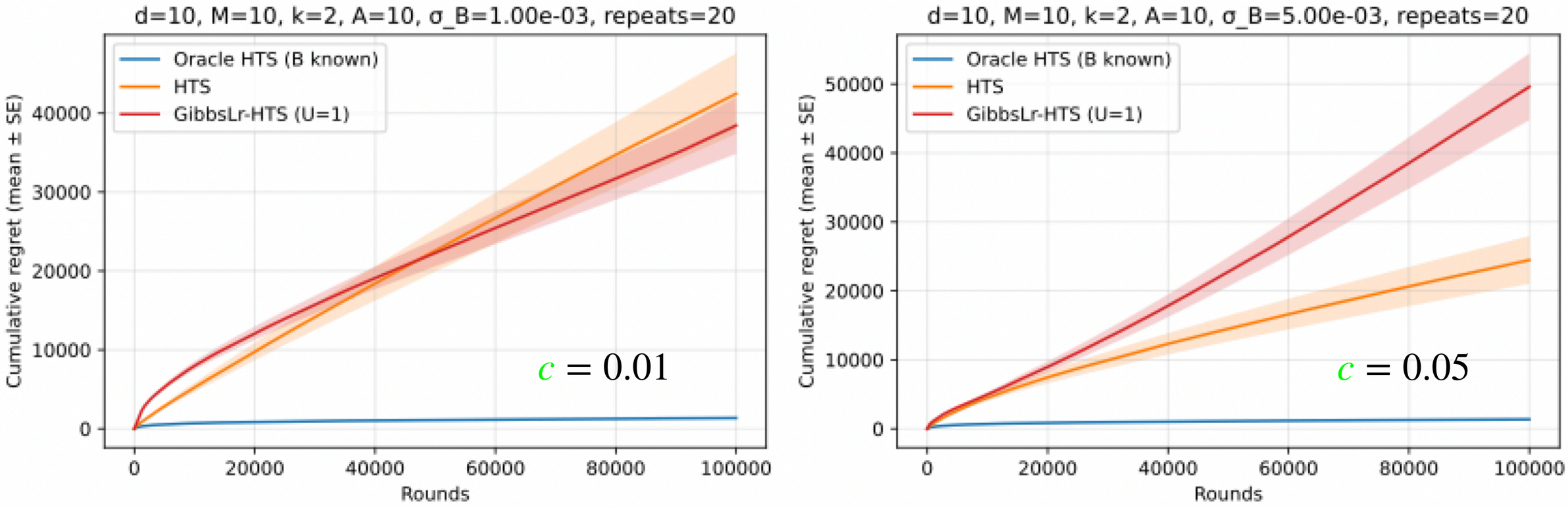


Figure 1. Empirical Bayes regret curves for the considered algorithms.

Thank you for your attention!

In summary:

1. **Low-rank Hierarchical Gaussian Linear Bandits:** A new setting that takes a Bayesian approach to multi-task linear bandits with a shared low-rank structure
2. **Gibbs, Low-rank Hierarchical Thompson Sampling:** Preliminary algorithm that combines Gibbs sampling with Thompson sampling
3. Preliminary experimental results

Future Directions.

1. Rigorous Bayes regret upper bound
2. Is this the “right” algorithm? How to set the tempering hyperparameter?