# Information Sharing in Hierarchical Bayesian Bandits using Meta Data

Jhanvi Garg [1] Fatemeh Doudi [2] Aayush Gautam [3] Andrew Porter [2]

[1]Dept. of Statistics

[2]Dept. of Electrical and Computer engineering

[3]Dept. of Computer Science and Engineering

Tuesday 2nd January, 2024

# Outline

- Motivation
- Prior Work
- Setting
- Algorithm
- Regret Bound Analysis
- Experiments
- Conclusion
- Future Directions
- Acknowledgement
- References

# Motivation

- Devising exploration techniques that are more efficient.
- Developing a model and an algorithm for leveraging metadata and similarity between tasks to enhance decision-making in multi-task MAB environment
- By utilizing the similarity, the effectiveness of information sharing can be enhanced by assigning greater weights to tasks that are close.
- Enhance the quality and effectiveness of recommendation systems

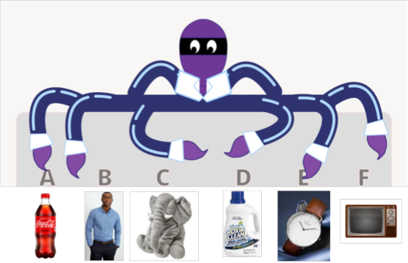



Figure: Choosing the Optimal Advertisement

# Prior Works

## Hierarchical Bayesian Bandits[1]

- Method
  - Share information of all the tasks using the hierarchical Bayesian model
- Drawback
  - Sharing information without utilizing metadata and taking into account the similarity
  - Only capable of dealing with situations where tasks originate from a single distribution

## Metadata-based Multi-Task Bandit with Bayesian Hierarchical Model[3]

- Leverages metadata to capture task specific details
- Incorporates a Bayesian hierarchical model to capture inter-task dependencies and share knowledge across tasks

# Setting

- Environment:

$$
\begin{gathered}
Y_{s,t} \mid A_{s,t}, \theta_{s,*} \sim N\left(A_{s,t}^{\top}\theta_{s,*}, \sigma^2\right), \quad \forall t \geqslant 1,\ s \in \mathcal{S} \\
\theta_{s,*} \mid \mu_{s*}, \gamma^* \sim N\left(\lambda\mu_{s*} + (1-\lambda)\gamma^*, \Sigma_0\right), \quad \forall s \in \mathcal{S} \\
\mu_{s*} \sim N\left(MB_s, \Sigma_a\right), \quad \forall s \in \mathcal{S} \\
\gamma_* \sim N\left(\mu_q, \Sigma_q\right)
\end{gathered}
$$

- Action space $\mathcal{A} \subset \mathbb{R}^d$
- $\mathcal{S}$ is the task space.
- $A_{s,t}$ and $Y_{s,t}$ denotes the action and reward at time t for task s respectively
- $\sigma \in \mathbb{R}^+$, $\Sigma_0, \Sigma_a, \Sigma_q \in \mathbb{R}^{d\times d}$ are positive definite matrices, $\lambda \in [0,1]$, $M \in \mathbb{R}^{|\mathcal{S}|\times d}$ and $B \in \mathbb{R}^{|\mathcal{S}|\times|\mathcal{S}|}$ are all known.
- $\mu_q, \theta_{s*}, \mu_{s*}, \gamma* \in \mathbb{R}^d$
- B is the similarity matrix and $B_s$ is a vector containing the similarity of task s with every other task.

# Setting

**Calculation for Similarity matrix**

- Defining feature vector $f_s$ for task $s \in \mathcal{S}$
- Weighted Gaussian kernel for similarity measurement
  - Weighing of features is important because they may have different scales.
  - Similarity between task $s$ and task $s1$ is given by:

$$B_{s,s1} \propto e^{-\frac{1}{2}\|w_i^T(f_s - f_{s1})\|}$$

- We normalize each row to have a constant sum.

# Algorithm

**Calculating Posterior Distribution**

Let us denote the history by $\mathcal{H}_t$. ie $\mathcal{H}_t = \{(A_{s_\tau,\tau}, Y_{s_\tau,\tau}), \tau \leqslant t\}$ where $s_\tau$ is the task chosen at time $\tau$.

Posterior Distribution of $\gamma_*$: $Q_t$

We denote $\gamma_* \mid \mathcal{H}_t \sim N\left(\bar{\mu}_t, \bar{\Sigma}_t\right)$

$$\bar{\mu}_t = \bar{\Sigma}_t \left( \Sigma_q^{-1}\mu_q + \sum_{s\in[m]} (\Sigma_0 + G_{s,t}^{-1})^{-1} G_{s,t}^{-1} R_{s,t} \right),$$

$$\bar{\Sigma}_t^{-1} = \Sigma_q^{-1} + (\Sigma_0 + G_{s,t}^{-1})^{-1}$$

- $R_{s,t} = \sigma^{-2} \sum_{l<t} 1_{S_l}(s) A_{s,l} Y_{s,l}$ is the weighed reward of state s until time t, weighted by features of taken actions
- $G_{s,t} = \sigma^{-2} \sum_{l<t} 1_{S_l}(s) A_{s,l} A_{s,l}^T$ is the outer product of the features of taken actions in task s up to round t

# Algorithm

Update of $\mu_{s*}$ : $P_{s,t}$

$\mu_{s*} \mid \mathcal{H}_t \sim N\left(\hat{\mu}_{s,t}, \hat{\Sigma}_{s,t}\right)$, where

$$\hat{\mu}_{s,t} = M_{s,t} B_s$$
$$M_{s,t} = \hat{\Sigma}_{s,t}(\Sigma_0^{-1} M + X)$$

where X is a $d \times |S|$ matrix and ith column of X is defined as follows:

$$X_{:,i} = (\Sigma_0 + G_{i,t}^{-1})^{-1} G_{i,t}^{-1} R_{i,t}$$
$$\hat{\Sigma}_{s,t}^{-1} = \Sigma_a^{-1} + \sum_{s' \in [m]} (\Sigma_0 + G_{s',t}^{-1})^{-1} B_{(s,s')}^{-2}$$

Posterior for $\theta_{s*}$

$\theta_{s,*} | \mathcal{H}_t, \gamma_t, \mu_{s,t} \sim N\left(\tilde{\mu}_{s,t}, \tilde{\Sigma}_{s,t}\right)$

$$\text{Define } \mu'_{s,t} = \lambda \gamma_t + \mu_{s,t}(1-\lambda)$$
$$\tilde{\mu}_{s,t} = \tilde{\Sigma}_{s,t}(\Sigma_0^{-1} \mu'_{s,t} + R_{s,t})$$
$$\tilde{\Sigma}_{s,t}^{-1} = \Sigma_0^{-1} + G_{s,t}$$

The calculation for posterior distributions are similar to the one presented in a paper [2]

# Algorithm

**Algorithm 1: MetaHierTS** - Meta Hierarchical Thompson Sampling

**Data:** Task set $S$ where $|S| = m$ and $n$ is the number of times the agent interacts with each task.

**Result:** Task recommendation for each round

Initialize the prior for $Q_1$ and $P_{s,1}$ for all s $\in \mathcal{S}$, set $\mathcal{H}_0 = \{\}$ and $S' = S$

**for** *each round t* **do**

- Choose task s at random from the set of tasks $\mathcal{S}'$;
- Sample $\gamma_t$ from the posterior $Q_t$ and $\mu_{s,t}$ from posterior $P_{s,t}$;
- Sample $\theta_{s,t} \sim \theta_{s,*} \mid \mathcal{H}_t, \gamma_t, \mu_{s,t}$;
- Sample the reward $Y_{s,t,a} \mid \theta_{s,t}, a$ for all action $a \in \mathcal{A}$;
- $a_m = \underset{a \in \mathcal{A}}{\mathrm{argmax}} Y_{s,t,a}$
- Observe the true reward corresponding to the action $a_m$ and call it $\tilde{Y}_{s,t,a_m}$
- $\mathcal{H}_{t+1} = \mathcal{H}_t \cup \left(a_m, \tilde{Y}_{s,t,a_m}\right)$
- if task s is taken n times: $S' = S' \backslash$s

# Regret Bound

The following theorem provides a regret bound for the sequential setting where $\Sigma_0 = \sigma_0^2 I_d$, $\Sigma_a = \sigma_a^2 I_d$, and $\Sigma_q = \sigma_q^2 I_d$.

**Theorem** (Sequential regret) - Let $|\mathcal{S}_t| = 1$ for all rounds t and action space is finite ($|\mathcal{A}| = K$). The Bayes regret of MetaHierTS environment is given by

$$\mathcal{BR}(m,n) \leq K\sqrt{\frac{2}{\pi}}\sigma_{max} + K\sqrt{2log(mn)mn[c_1 m + c_2]}$$

where

$\sigma_{max} = \lambda^2\sigma_q^2 + (1-\lambda)^2\sigma_a^2 + \sigma_0^2$

$c_1 = \frac{\sigma_0^2}{\log(1+\sigma_0^2\sigma^{-2})}\log\left(1 + \frac{n\sigma_0^2}{\sigma^2 K}\right)$

$c_2 = \frac{c_q c}{\log(1+c_q\sigma^{-2})}\log\left(1 + \frac{m\sigma_q^2}{\sigma_0^2}\right)$, here $c_q = \lambda^2\sigma_q^2 + (1-\lambda)^2\sigma_a^2$ and $c = 1 + \frac{\sigma_0^2}{\sigma^2}$

In the proof, we make an implicit assumption that $\sigma_a^2 \leq \sigma_q^2$

# Regret Bound - Observations

- Proof of the regret bound for our case is closely analogous to the proof presented in a paper [1].
- The previous slide implies that the regret bound scales linearly with the number of tasks and sublinearly with respect to the number of rounds agent interacts with each task.
- Linearity in number of tasks is because the task parameters $\theta_{s,*}$ are generated independently from its distribution given $\mu_{s,*}$ and $\gamma*$.
- Sublinear scaling of the regret bound with respect to number of rounds agent interacts with each task is because the Thompson sampling algorithm is generally sublinear.
- This type of hierarchical model will not facilitate information sharing among tasks.

# Regret Bound - Conclusions

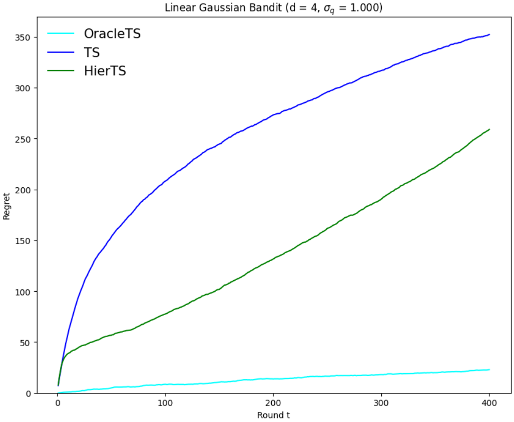
- To achieve a sublinear regret with respect to the number of tasks, the tasks should be inherently correlated with each other.
- Some assumptions that might achieve this are as follows:
  - **Clustering** - Multiple tasks share the same task parameter.
  - **Adding correlation between tasks parameter**- For every task $s$ in task space $\mathcal{S}$, the task parameters $\theta_{s,a}$ jointly follows a multivariate normal distribution where the covariance matrix is not diagonal for every action $a$ in $\mathcal{A}$.
  - **Sequential Tasks** - Here we have a series of related tasks that must be completed in a specific order, and the task parameter of each task influence the parameter of subsequent tasks.

# Experiments – Simulation Environment

- Define the environment parameters
  1. `num_clusters`: Number of clusters from which $\mu_*$ is sampled
  2. sigma: Reward noise, $\sigma$
  3. mu_q: Mean of the Gaussian hyperprior, known to the algorithm and from which `num_clusters` number of $\mu_*$s are sampled
  4. `sigma_q` standard deviation of the Gaussian hyperprior, $\sigma_q$
  5. `sigma_0` standard deviation of the Gaussian prior (with mean $\mu_*$ ) from which $\theta_*$ is sampled
- Sample `num_clusters` number of $\mu_*$s from a Gaussian hyperprior with mean $\mu_q$ and standard deviation $\sigma_q$
- For each task, $s \in S$
  1. Randomly choose a $\mu_*$
  2. Sample parameters $\theta_{s,*}$ from a Gaussian using $\mu_*$ as the mean and $\sigma_0$ as the standard deviation
  3. Generate metadata by adding Gaussian noise to $\theta_{s,*}$
  4. Sample parameters $A_{s,*}$ for each arm by sampling from a unit ball

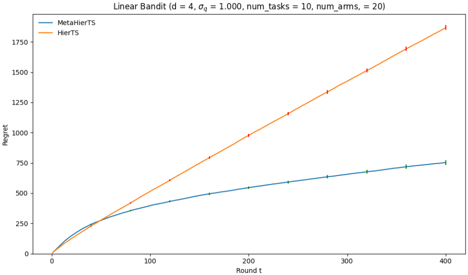
# Experiments – Results

- Bandits have Reward Probability Chosen Randomly
- Regret $= \mathbb{E}[\sum_{t=0}^{T}(\mu_* - \mu_s)]$
- Comparing HierTS [1] with vanilla Thompson sampling



# Experiments – Results

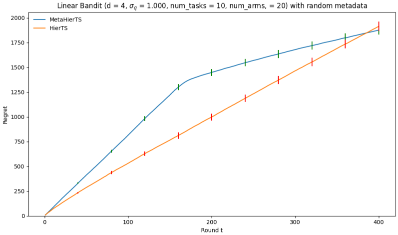
- Bandits have Reward Probability Chosen Randomly
- Regret $= \mathbb{E}[\sum_{t=0}^{T}(\mu_* - \mu_s)]$
- Number of clusters = 3, $\lambda = 0.1$



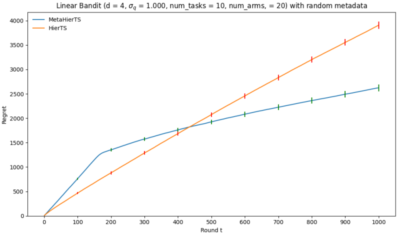

Comparison between HierTS [1] and MetaHierTS (ours)

# Experiments – Results

- When the metadata is random
- Regret $= \mathbb{E}[\sum_{t=0}^{T}(\mu_* - \mu_s)]$
- Number of clusters $= 3$, $\lambda = 0.1$



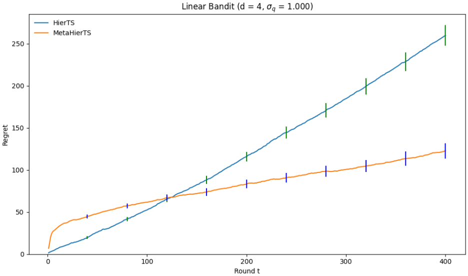

Number of rounds=400




Number of rounds $= 1000$

# Experiments – Results

- When there is only one cluster and metadata is random
- Environment generation is exactly the same as in HierTS [1]
- Regret $= \mathbb{E}[\sum_{t=0}^{T}(\mu_* - \mu_s)]$
- $\lambda = 0.1$




Comparison between HierTS [1] and MetaHierTS (ours)

# Conclusion and Future directions

- We considered the problem of metadata multi-task bandit as in [1]
- To have better utilization of metadata, we defined and used a similarity matrix in our framework
- We use the aforementioned framework as a special case and generalized their setup
- Simulation results show that our MetaHierTS algorithm achieves better regret bounds than the algorithm presented in [1]
- In future, we aim to improve the theoretical regret bound by including additional assumptions in the environment generation process.
- We aim to test our algorithm using a real-world dataset in future.
- We plan to broaden our formulation to include non-Gaussian scenarios.

# Acknowledgement and Contributions

We gratefully thank Nilson Chapagain for sharing his insights with us during the process of formulating the problem. Additionally, we would like to thank Prof. Dileep for referring us to relevant literature to guide our research.

The authors contribution are as follows:

- Problem Formulation[1] [2] [3] [4]
- Substantial contribution to designing and analysing the algorithm in theory[1] [2]
- Substantial contribution to data generation and implementation of the algorithm[3] [4]

---

[1] Jhanvi
[2] Fatemeh
[3] Aayush
[4] Andrew

# References

[1] J. Hong, B. Kveton, M. Zaheer, and M. Ghavamzadeh. Hierarchical bayesian bandits, 2022.

[2] B. Kveton, M. Konobeev, M. Zaheer, C.-W. Hsu, M. Mladenov, C. Boutilier, and C. Szepesvari. Meta-thompson sampling. In M. Meila and T. Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 5884–5893. PMLR, 18–24 Jul 2021.

[3] R. Wan, L. Ge, and R. Song. Metadata-based multi-task bandits with bayesian hierarchical models. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 29655–29668. Curran Associates, Inc., 2021.