Under consideration for publication in Knowledge and Information Systems

Cutting to the Chase with Warm-Start Contextual Bandits[1]

Bastian Oetomo, R. Malinga Perera, Renata Borovica-Gajic and Benjamin I. P. Rubinstein
School of Computing and Information Systems, The University of Melbourne

Abstract. Multi-armed bandits achieve excellent long-term performance in practice and sublinear cumulative regret in theory. However a real-world limitation of bandit learning is poor performance in early rounds due to the need for exploration—a phenomenon known as the cold-start problem. While this limitation may be necessary in the *general* classical stochastic setting, in practice where "pre-training" data or knowledge is available, it is natural to attempt to "warm start" bandit learners. This paper provides a theoretical treatment of warm-start contextual bandit learning, adopting Linear Thompson Sampling as a principled framework for flexibly transferring domain knowledge as might be captured by bandit learning in a prior related task, a supervised pre-trained Bayesian posterior, or domain expert knowledge. Under standard conditions we prove a general regret bound. We then apply our warm-start algorithmic technique to other common bandit learners—the ϵ-greedy and upper-confidence bound contextual learners. An upper regret bound is then provided for LinUCB. Our suite of warm-start learners are evaluated in experiments with both artificial and real-world datasets, including a motivating task of tuning a commercial database. A comprehensive range of experimental results are presented, highlighting the effect of different hyperparameters and quantities of pre-training data.

Keywords: multi-armed bandits, warm-start, pre-training

[1] A preliminary abridged version of this paper appeared as: Bastian Oetomo, R. Malinga Perera, Renata Borovica-Gajic, and Benjamin I. P. Rubinstein. "Cutting to the Chase with Warm-Start Contextual Bandits." *2021 IEEE 21th International Conference on Data Mining (ICDM)*, pp. 459–468, 2021.

Received 28 Feb 2022
Revised 24 Nov 2022
Accepted 04 Feb 2023

1. Introduction

Multi-armed bandits have undergone a renaissance in machine learning research (Slivkins, 2019; Lattimore and Szepesvári, 2020) with a range of deep theoretical results discovered, while applications to real-world sequential decision making under uncertainty abound, ranging from news (Li, Chu, Langford and Schapire, 2010) and movie recommendation (Qin, Chen and Zhu, 2014), to crowd sourcing (Tran-Thanh, Stein, Rogers and Jennings, 2014) and self-driving databases (Perera, Oetomo, Rubinstein and Borovica-Gajic, 2021; Marcus, Negi, Mao, Tatbul, Alizadeh and Kraska, 2020). The relative simplicity of the stochastic bandit setting, as compared to more general partially-observable Markov decision processes (POMDPs), typically admits regret analysis where bandit learners enjoy bounded cumulative regret—the gap between a learner's cumulative reward to time T and the cumulative reward possible with a fixed but optimal-with-hindsight policy. While many bandit learners are celebrated for attaining sublinear regret or average regret converging to zero, such *long-term performance goals* say little about the *short-term performance* of today's popular bandit algorithms.

Indeed the bandit setting is well known to be the simplest Markov decision process setting to require balancing of *exploration*—attempting infrequent actions in case of higher-than-expected rewards—with *exploitation*—greedy selection of actions that so far appear fruitful. Even in the stochastic setting, where rewards are drawn from stationary (context conditional) distributions, the underlying distributions are unknown and considered adversarially chosen. In other words, there's no free lunch (in the worst case) without significant exploration in early rounds.

The relatively poor early round performance of bandit learners is known as the *cold start problem*, and can be costly in high-stakes domains. Li et al. (2010) suggested that bandit learners be *warm started* or pre-trained somehow prior to such deployment, in the context of online media recommendation and advertising where poor performance leads to user dissatisfaction and financial loss. However little systematic research has explored the cold start problem. Intuitively, warm start is related to transfer learning (Cao, Pan, Zhang, Yeung and Yang, 2010) and domain adaptation (Csurka, 2017) while Shivaswamy and Joachims (2012) proposed warm-starting methods for non-contextual bandits and Zhang, Agarwal, Daumé III, Langford and Negahban (2019) modify any bandit policy to make use of pre-training from (batch) supervised learning via manipulation of the policy's importance sampling and weighting, which determines the relative importance of one data $(\boldsymbol{x}, y)$ over the other data—ultimately resulting in a weighted linear regression. Another work by Li, Xie, Lin and Lui (2021) employs virtual plays before committing to an action in every round, which implicitly assumes that the existing logged data is perfectly aligned with the unknown bandit data. A similar assumption is made implicitly by Bouneffouf, Parthasarathy, Samulowitz and Wistub (2019), who combine prior historical observations and clustering information. Other works have proposed approaches to the item-user cold-start problem, such as that proposed by Wang, Wang, Wang and He (2017), who passively assign a user to each item on top of the usual bandit which selects an item for a user. The warm-start problem is also related to the conservative bandit problem, where the usual bandit setting applies under the existence of a baseline policy and a performance constraint (Kazerouni, Ghavamzadeh, Abbasi Yadkori and Van Roy, 2017). This paper advocates for Thompson Sampling

(TS) (Thompson, 1933) as a natural framework for warm start bandits. Although the prior used in Thompson Sampling can be misspecified, as discussed by Liu and Li (2015), our extension to the LinTS contextual bandit not only affords more flexible forms of warm start, but quantifies prior uncertainty, and admits regret analysis. Furthermore, this idea can be extended into other bandit algorithms, such as ϵ-greedy and LinUCB.

Flexibility in warm start is paramount, as not all settings requiring warm start will necessarily admit prior supervised learning as assumed previously (Zhang et al., 2019). Indeed, bandits are typically motivated when there is an absence of direct supervision, and only indirect rewards are available. Our framework offers unprecedented flexibility. We advocate that prior knowledge could come from: bandit learning on a previous, related task; domain expert knowledge or knowledge extracted from a rule-based, non-adaptive baseline system; or indeed prior supervised learning.

We introduce a new motivation for warm start bandits from the database systems domain. Database indices, a data structure used by database management systems to execute queries more rapidly, may be formed on any combination of table columns. Unfortunately the best choice of index depends on unknown query workloads and potentially unstable system performance. Offline solutions to index selection have been the foundations of the automated tools provided by database vendors (Agrawal, Chaudhuri, Kollár, Marathe, Narasayya and Syamala, 2004; Zilio, Rao, Lightstone, Lohman, Storm, Garcia-Arellano and Fadden, 2004; Dageville, Das, Dias, Yagoub, Zaït and Ziauddin, 2004). Recognising that database administrators cannot practically foresee future database loads, *online* solutions, where the choice of the representative workload and the cost-benefit analysis of materialising a configuration are automated, have been proposed (Schnaitter, Abiteboul, Milo and Polyzotis, 2007; Sattler, Schallehn and Geist, 2004; Bruno and Chaudhuri, 2007; Bruno and Chaudhuri, 2006; Das, Grbic, Ilic, Jovandic, Jovanovic, Narasayya, Radulovic, Stikic, Xu and Chaudhuri, 2019; Ma, Van Aken, Hefny, Mezerhane, Pavlo and Gordon, 2018). Unfortunately most such approaches lack any form of performance guarantee. Recent work has demonstrated compelling potential for linear bandits for index selection (Perera et al., 2021) complete with regret bound guarantees, however the cold start problem is likely to limit deployment as vendors and users alike may be concerned about out-of-box performance. We demonstrate that a warm start bandit can deliver strong short-term improvement for database index selection without costing long-term results.

In summary, this paper makes the following contributions:

- We propose a framework for warm starting contextual bandits based on LinTS and extend our technique to ϵ-greedy and LinUCB;
- Unlike past efforts to warm-start bandit learners, which strictly apply to supervised learning only, our Warm Start Linear Bandit seen in Algorithms 2, 3 and 4 can incorporate prior knowledge from any form of prior learning, such as: supervised learning (Zhang et al., 2019), prior bandit learning, or manual construction of a prior by a domain expert. Notably our warm start approach incorporates uncertainty quantification;
- We introduce a method to automatically tune the hyperparameters used in Algorithms 2, 3 and 4;
- We present regret bounds for Warm Start LinTS and LinUCB that demonstrate sublinear regret for long-term performance;

– Experiments on database index selection (using data derived from standard system benchmarks), classification task data and synthetic data demonstrates performance improvement in the short term with performance competitive with baselines (where such baselines are able to be run); and
– We have expanded experiments to demonstrate the effect of increased pre-training on the performance in both accurate and misspecified settings.

2. Background: Contextual Bandits and Linear Thompson Sampling

The stochastic contextual multi-armed bandit (MAB) problem is a game proceeding in rounds $t \in [T] = \{1, 2, \ldots, T\}$. In round t the MAB learner,

1. observes k possible actions or *arms* $i \in [k]$ each with adversarially chosen *context vector* $\boldsymbol{x}_t(i) \in \mathbb{R}^d$;
2. selects or *pulls* an arm $i_t \in [k]$;
3. observes random reward $R_{i_t}(t)$ for the pulled arm i_t, where each $R_i(t) \mid \boldsymbol{x}_t(i) \sim P_{i|\boldsymbol{x}_t(i)}$ independently over $i \in [k], t \in [T]$.

The MAB learner's goal is to maximise its cumulative expected reward—the total expected reward over all rounds—which is equivalent to minimising the *cumulative regret* up to round T:

$$Reg(T) = \sum_{t=1}^{T} \mathbb{E}\left[R_{i_t^\star}(t) \mid \boldsymbol{x}_t(i_t^\star)\right] - \mathbb{E}\left[R_{i_t}(t) \mid \boldsymbol{x}_t(i_t)\right] ,$$

where $i_t^\star \in \arg\max_{i \in [k]} \mathbb{E}\left[R_i(t) \mid \boldsymbol{x}_t(i)\right]$, that is, an optimal arm to pull at round t. When a MAB algorithm's cumulative regret $Reg(T)$ is sub-linear in T, the average regret $Reg(T)/T$ goes to zero. Such an algorithm is said to be a "no regret" learner or *Hannan consistent*.

Thompson Sampling (TS), a Bayesian approach within the family of *randomised probability matching* algorithms, is one of the earliest design patterns for MAB learning (Thompson, 1933). Each modeled arm's reward likelihood is endowed with a prior. Arms are then pulled based on their posteriors: *e.g.*, parameters for each arm can be drawn from the corresponding posteriors, and then arm selection may proceed (greedily) by maximising reward likelihood.

Linear Thompson Sampling (LinTS) (Agrawal and Goyal, 2013; Abeille, Lazaric et al., 2017) is an algorithm with sub-linear cumulative regret, when the context-conditional reward satisfies a linear relationship

$$r_t(i_t) = R_{i_t}(t) \mid \boldsymbol{x}_t(i_t) = \boldsymbol{\theta}_\star^T \boldsymbol{x}_t(i_t) + \epsilon_t(i_t) ,$$

where additive noise $\epsilon_t(i_t)$ is conditionally R-subgaussian and $\boldsymbol{\theta}_\star \in \mathbb{R}^d$ is an unknown vector-valued parameter shared among all of the k arms.

Like most approaches to linear contextual bandit learning, LinTS adopts (online) ridge regression fitting for estimating the unknown parameter. For any regularisation parameter $\lambda \in \mathbb{R}^+$, define the matrix $\boldsymbol{V}_t$ as

$$\boldsymbol{V}_t = \lambda \boldsymbol{I} + \sum_{s=1}^{t-1} \boldsymbol{x}_s(i_s) \boldsymbol{x}_s^T(i_s) . \tag{1}$$

Algorithm 1 Linear Thompson Sampler

Input: $\hat{\boldsymbol{\theta}}_1, \lambda, \delta, T$
Initialize $\boldsymbol{V}_1 \leftarrow \lambda \boldsymbol{I}_d$, $\delta' = \frac{\delta}{4T}$, $\boldsymbol{b}_1 \leftarrow \boldsymbol{0}$
for $t = 1, \ldots, T$ **do**
  Sample $\boldsymbol{\eta}_t \sim \mathcal{D}^{TS}$
  $\tilde{\boldsymbol{\theta}}_t \leftarrow \hat{\boldsymbol{\theta}}_t + \beta_t(\delta')\boldsymbol{V}_t^{-1/2}\eta_t$ {perturbed parameter}
  $i_t \leftarrow s \in \arg\max_{i \in [k]} \tilde{\boldsymbol{\theta}}_t^T \boldsymbol{x}_t(i)$ {optimal arm}
  Pull arm i_t and observe reward $r_t(i_t)$
  $\boldsymbol{V}_{t+1} \leftarrow \boldsymbol{V}_t + \boldsymbol{x}_t(i_t)\boldsymbol{x}_t^T(i_t)$ {update Eq. (1)}
  $\boldsymbol{b}_{t+1} \leftarrow \boldsymbol{b}_t + r_t(i_t)\boldsymbol{x}_t(i_t)$
  $\hat{\boldsymbol{\theta}}_{t+1} \leftarrow \boldsymbol{V}_{t+1}^{-1}\boldsymbol{b}_{t+1}$ {update Eq. (2)}
end for

Then Abeille et al. (2017) demonstrated that we can estimate the unknown parameter $\boldsymbol{\theta}_\star$ as

$$\hat{\boldsymbol{\theta}}_t = \boldsymbol{V}_t^{-1} \sum_{s=1}^{t-1} \boldsymbol{x}_s(i_s) r_t(i_s) \ . \tag{2}$$

Earlier versions of LinTS (Agrawal and Goyal, 2013) do not include a tunable regularisation parameter.

A result due to Abbasi-Yadkori, Pál and Szepesvári (2011) is used within LinTS. Assuming $\|\boldsymbol{\theta}_\star\| \leq S$, then with probability at least $1 - \delta \in (0, 1)$:

$$\|\hat{\boldsymbol{\theta}}_t - \boldsymbol{\theta}_\star\|_{\boldsymbol{V}_t} \leq \beta_t(\delta) \ ,$$
$$\beta_t(\delta) = R\sqrt{2 \log \frac{\det(\boldsymbol{V}_t)^{1/2} \det(\boldsymbol{V}_1)^{-1/2}}{\delta}} + \sqrt{\lambda} S \ .$$

In Thompson Sampling, we may introduce a perturbation parameter $\boldsymbol{\eta}_t \in \mathbb{R}^d$, which, after rotation and scaling by the inverse square root of the matrix $\boldsymbol{V}_t^{-1/2}$, and scaling by oversampling factor $\beta_t(\delta')$, promotes exploration around the point estimate $\hat{\boldsymbol{\theta}}_t$:

$$\tilde{\boldsymbol{\theta}}_t = \hat{\boldsymbol{\theta}}_t + \beta_t(\delta')\boldsymbol{V}_t^{-1/2}\boldsymbol{\eta}_t \ .$$

Moreover, Abeille et al. (2017) have shown, that if $\boldsymbol{\eta}_t$ follows distribution $\mathcal{D}^{TS}$ with the following properties:

1. There exists $p > 0$ such that, for all $\|\boldsymbol{u}\| = 1$ we have $\mathbb{P}_{\boldsymbol{\eta} \sim \mathcal{D}^{TS}}(\boldsymbol{u}^T \boldsymbol{\eta} \geq 1) \geq p$; and
2. There exist positive constants c and c' such that, for all $\delta \in (0, 1)$ we have the inequality $\mathbb{P}_{\boldsymbol{\eta} \sim \mathcal{D}^{TS}}\left(\|\boldsymbol{\eta}\| \leq \sqrt{cd \log \frac{c'd}{\delta}}\right) \geq 1 - \delta$,

then LinTS is Hannan consistent. We adopt a standard multivariate Gaussian for $\boldsymbol{\eta}_t$ which satisfies the above properties (Abeille et al., 2017). With all of these definitions in mind, the version of LinTS used in this paper can be summarised as shown in Algorithm 1.

3. Warm Starting Linear Bandits

We now detail our flexible algorithmic framework for warm starting contextual bandits, beginning with Linear Thompson Sampling for which we derive a new regret bound.

3.1. Thompson Sampling

Given the foundation of Thompson Sampling in Bayesian inference, it is natural to look to manipulating the prior as a means to injecting *a priori* knowledge of the reward structure before the bandit is put into operation. The Algorithm 1 implementation of LinTS due to Abeille et al. (2017) decomposes the prior and posterior distributions on $\boldsymbol{\theta}_t$ as a Gaussian centred at the point estimate $\hat{\boldsymbol{\theta}}_t$ with covariance based on oversampling factor $\beta_t(\delta')$ and the matrix $\boldsymbol{V}_t$ via the random perturbation vector $\boldsymbol{\eta}_t$. Our approach to warm start is to focus on manipulating the initial point estimate $\hat{\boldsymbol{\theta}}_1$ and the matrix $\boldsymbol{V}_1$ to incorporate available prior knowledge into LinTS.

Remark 1. *Although Algorithm 1 appears to offer the freedom to select any* $\hat{\boldsymbol{\theta}}_1$*, Equations* (1) *and* (2) *do not present an immediate route to adapting subsequent point estimates* $\hat{\boldsymbol{\theta}}_t$*. Generalising Equation* (2) *to point estimate* $\hat{\boldsymbol{\theta}}_t = \boldsymbol{V}_t^{-1}(\lambda\hat{\boldsymbol{\theta}}_1 + \sum_{s=1}^{t-1} \boldsymbol{x}_s(i_s)r_t(i_s))$ *is unintuitive and does not clearly admit regret analysis.*

We adopt an intuitive approach of adapting Algorithm 1 to model the difference between an initial guess derived from some process occurring before bandit learning, and the actual parameter. **This pre-deployment process could be batch supervised learning, an earlier bandit deployment on a related decision problem, or simply a prior manually constructed by a domain expert. Our general framework is completely agnostic** and generalises earlier approaches to warm-starting bandits such as (Zhang et al., 2019). Without loss of generality we refer to this earlier process as the *first phase* and the basis for which initial parameters are designed as the *first phase dataset*. Let $\boldsymbol{\theta}_\star = \boldsymbol{\mu}_\star + \bar{\boldsymbol{\delta}}_\star$, where $\boldsymbol{\mu}_\star$ is the true parameter of the first phase dataset and $\bar{\boldsymbol{\delta}}_\star$ represents the *concept drift* between first phase and bandit deployment. With this reparametrisation, our linear model becomes:

$$\begin{aligned} r_t(i_t) &= \boldsymbol{\theta}_\star^T \boldsymbol{x}_t(i_t) + \epsilon_t(i_t) = (\boldsymbol{\mu}_\star + \bar{\boldsymbol{\delta}}_\star)^T \boldsymbol{x}_t(i_t) + \epsilon_t(i_t) \\ r_t(i_t) - \boldsymbol{\mu}_\star^T \boldsymbol{x}_t(i_t) &= \bar{\boldsymbol{\delta}}_\star^T \boldsymbol{x}_t(i_t) + \epsilon_t(i_t) \\ y_t(i_t) &= \bar{\boldsymbol{\delta}}_\star^T \boldsymbol{x}_t(i_t) + \epsilon_t(i_t)\ . \end{aligned}$$

Therefore, our problem has reduced from estimating $\boldsymbol{\theta}_\star$ to estimating $\bar{\boldsymbol{\delta}}_\star$.

Consider a Bayesian linear regression model with the unknown true value of first phase dataset $\boldsymbol{\mu}_\star$ modeled by random variable $\boldsymbol{\mu} \sim \mathcal{N}(\hat{\boldsymbol{\mu}}, \boldsymbol{\Sigma}_\mu)$ with conjugate context-conditional Gaussian likelihood. We then model the difference parameter $\bar{\boldsymbol{\delta}}_\star$ as $\bar{\boldsymbol{\delta}} \sim \mathcal{N}(\boldsymbol{0}, \alpha^{-1}\boldsymbol{I})$. If $\boldsymbol{\theta} = \boldsymbol{\mu} + \bar{\boldsymbol{\delta}}$ is the random variable modelling $\boldsymbol{\theta}_\star$, then $\boldsymbol{\theta} \sim \mathcal{N}(\hat{\boldsymbol{\mu}}, \boldsymbol{\Sigma}_\mu + \alpha^{-1}\boldsymbol{I})$ owing to the Gaussian's stability property. Finally, since $\hat{\boldsymbol{\mu}}$ is known, we can model $\boldsymbol{\theta}$ as $\boldsymbol{\theta} = \hat{\boldsymbol{\mu}} + \boldsymbol{\delta}$, that is, a random variable centred at $\hat{\boldsymbol{\mu}}$ which is shifted by drift $\boldsymbol{\delta} \sim \mathcal{N}(\boldsymbol{0}, (\boldsymbol{\Sigma}_\mu + \alpha^{-1}\boldsymbol{I}_d))$.

We next generalise the coupled recurrence Equations (1) and (2) for efficient incremental computation of the generalised posterior estimates.

Proposition 1. *Consider linear regression likelihood* $y_i = \boldsymbol{\theta}^T \boldsymbol{x}_i + \epsilon_i$*, where* $\epsilon_i \sim \mathcal{N}(0, R^2)$*, and prior* $\boldsymbol{\theta} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{V}_1^{-1})$*. Then the posterior conditioned on data* $\boldsymbol{z}_i = (\boldsymbol{x}_i, y_i)$ *for* $i \in [t]$ *is given by* $\mathcal{N}(\hat{\boldsymbol{\theta}}_{t+1}, R^2 \boldsymbol{V}_{t+1}^{-1})$ *where* $\boldsymbol{\theta}_t$ *point estimates are defined by Equation* (2)*, and we replace Equation* (1) *for* $\boldsymbol{V}_t$ *with*

$$\boldsymbol{V}_t = R^2 \boldsymbol{V}_1 + \sum_{s=1}^{t-1} \boldsymbol{x}_s(i_s) \boldsymbol{x}_s^T(i_s) \ , \tag{3}$$

where R^2 *is the variance of the measurement noise.*

Proof. The posterior distribution is:

$$\begin{aligned}
&p(\boldsymbol{\theta} \mid y_1, \cdots, y_n) \\
&\quad \propto \exp\left\{ -\frac{1}{2} \left[\sum_{i=1}^{n} \left(\frac{y_i - \boldsymbol{\theta}^T \boldsymbol{x}_i}{R} \right)^2 + \boldsymbol{\theta}^T \boldsymbol{V}_1 \boldsymbol{\theta} \right] \right\} \\
&\quad \propto \exp\left\{ -\frac{1}{2} \left[\boldsymbol{\theta}^T \left(\frac{1}{R^2} \sum_{i=1}^{n} \boldsymbol{x}_i \boldsymbol{x}_i^T \right) \boldsymbol{\theta} - \frac{2}{R^2} \boldsymbol{\theta}^T \sum_{i=1}^{n} y_i \boldsymbol{x}_i + \boldsymbol{\theta}^T \boldsymbol{V}_1 \boldsymbol{\theta} \right] \right\} \\
&\quad = \exp\left\{ -\frac{1}{2} \left[\boldsymbol{\theta}^T \left(\boldsymbol{V}_1 + \frac{1}{R^2} \sum_{i=1}^{n} \boldsymbol{x}_i \boldsymbol{x}_i^T \right) \boldsymbol{\theta} \right.\right. \\
&\qquad \left.\left. - \boldsymbol{\theta}^T \left(\frac{1}{R^2} \sum_{i=1}^{n} y_i \boldsymbol{x}_i \right) - \left(\frac{1}{R^2} \sum_{i=1}^{n} y_i \boldsymbol{x}_i \right)^T \boldsymbol{\theta} \right] \right\} \ .
\end{aligned}$$

To avoid clutter, let $\bar{\boldsymbol{V}}_{n+1} = \boldsymbol{V}_1 + \frac{1}{R^2} \sum_{i=1}^{n} \boldsymbol{x}_i \boldsymbol{x}_i^T$ and $\bar{\boldsymbol{b}}_{n+1} = \frac{1}{R^2} \sum_{i=1}^{n} y_i \boldsymbol{x}_i$. Therefore, our posterior distribution can be rewritten as

$$\begin{aligned}
&p(\boldsymbol{\theta} \mid y_1, \cdots, y_n) \\
&\propto \exp\left\{ -\frac{1}{2} \left[\boldsymbol{\theta}^T \bar{\boldsymbol{V}}_{n+1} \boldsymbol{\theta} - \boldsymbol{\theta}^T \bar{\boldsymbol{b}}_{n+1} - \bar{\boldsymbol{b}}_{n+1}^T \boldsymbol{\theta} \right] \right\} \\
&\propto \exp\left\{ -\frac{1}{2} \left[\boldsymbol{\theta}^T \bar{\boldsymbol{V}}_{n+1} \boldsymbol{\theta} - \boldsymbol{\theta}^T \bar{\boldsymbol{V}}_{n+1} \bar{\boldsymbol{V}}_{n+1}^{-1} \bar{\boldsymbol{b}}_{n+1} - \bar{\boldsymbol{b}}_{n+1}^T \bar{\boldsymbol{V}}_{n+1}^{-T} \bar{\boldsymbol{V}}_{n+1} \boldsymbol{\theta} \right.\right. \\
&\qquad \left.\left. + \ \bar{\boldsymbol{b}}_{n+1}^T \bar{\boldsymbol{V}}_{n+1}^{-T} \bar{\boldsymbol{V}}_{n+1} \bar{\boldsymbol{V}}_{n+1}^{-1} \bar{\boldsymbol{b}}_{n+1} \right] \right\} \\
&= \exp\left\{ -\frac{1}{2} \left(\boldsymbol{\theta} - \bar{\boldsymbol{V}}_{n+1}^{-1} \bar{\boldsymbol{b}}_{n+1} \right)^T \bar{\boldsymbol{V}}_{n+1} \left(\boldsymbol{\theta} - \bar{\boldsymbol{V}}_{n+1}^{-1} \bar{\boldsymbol{b}}_{n+1} \right) \right\} \ ,
\end{aligned}$$

which is proportional to $\mathcal{N}(\bar{\boldsymbol{V}}_{n+1}^{-1} \bar{\boldsymbol{b}}_{n+1}, \bar{\boldsymbol{V}}_{n+1}^{-1})$. Therefore, our estimator for θ would be

$$\hat{\boldsymbol{\theta}}_{n+1} = \bar{\boldsymbol{V}}_{n+1}^{-1} \bar{\boldsymbol{b}}_{n+1} = \boldsymbol{V}_{n+1}^{-1} \boldsymbol{b}_{n+1} \ ,$$

where we have defined

$$\boldsymbol{V}_{n+1} = R^2 \boldsymbol{V}_1 + \sum_{i=1}^{n} \boldsymbol{x}_i \boldsymbol{x}_i^T, \quad \boldsymbol{b}_{n+1} = \sum_{i=1}^{n} y_i \boldsymbol{x}_i \ .$$

This completes the proof. □

Our approach comes with an appealing interpretation when setting $\bar{\boldsymbol{\delta}} \sim$

$\mathcal{N}(\mathbf{0}, \alpha^{-1}\boldsymbol{I})$: when we are confident that our pre-training guess is very close to the true parameter, we can set drift α^{-1} to be very small and close to 0. However, when we are not as confident, α^{-1} is naturally set large. Large α^{-1} creates more "deviation" or error from our first phase parameter $\boldsymbol{\mu}_\star$. This suggests a promising new direction which we highlight in future work Section 6.

Our simple reduction of warm start bandit learning to LinTS admits a regret bound. We follow the pattern of the regret analysis of Abeille et al. (2017) with differences detailed next.

Observe first that $\|\hat{\boldsymbol{\theta}}_t - \boldsymbol{\theta}_\star\|_{\boldsymbol{V}_t} = \|(\hat{\boldsymbol{\theta}}_t - \hat{\boldsymbol{\mu}}) - (\boldsymbol{\theta}_\star - \hat{\boldsymbol{\mu}})\|_{\boldsymbol{V}_t} = \|\hat{\boldsymbol{\delta}}_t - \boldsymbol{\delta}_\star\|_{\boldsymbol{V}_t} \leq \beta_t(\delta')$. Accordingly the argument yielding the confidence ellipsoid $\beta_t(\delta')$ stated in (Abbasi-Yadkori et al., 2011, Theorem 2) bounding $\|\hat{\boldsymbol{\theta}}_t - \boldsymbol{\theta}_\star\|_{\boldsymbol{V}_t}$ applies in our case, whose full proof of its modification can be found in the Appendix. However, as our initial matrix $\boldsymbol{V}_1$ generalises $\lambda\boldsymbol{I}$, we must alter the penultimate proof step of Abeille et al. (2017) as follows:

– the inequality proposed by Abbasi-Yadkori et al. (2011) which is used to define $\beta_t(\delta)$ in their paper is not valid in our scenario. This is corrected by using the version of $\beta_t(\delta)$ presented in this paper, removing the assumption that $\boldsymbol{V}_1 = \frac{\lambda}{R^2}\boldsymbol{I}$ and leave it in terms of $\boldsymbol{V}_1$:

$$R\sqrt{2\log\frac{\det(\boldsymbol{V}_t)^{1/2}\det(R^2\boldsymbol{V}_1)^{-1/2}}{\delta}} + \sqrt{\lambda_{max}(R^2\boldsymbol{V}_1)}S$$

– the inequality of (Abeille et al., 2017, Proposition 2) is no longer valid in our case. However, the last inequality in (Oetomo, Perera, Borovica-Gajic and Rubinstein, 2019) has modified (Abeille et al., 2017, Proposition 2) into:

$$\sum_{s=1}^{t}\|\boldsymbol{x}_s\|^2_{\boldsymbol{V}_s^{-1}} \leq 2\log\left(\frac{\det(\boldsymbol{V}_{t+1})}{\det(R^2\boldsymbol{V}_1)}\right)$$

and hence serves our purpose; and

– in proving (Abeille et al., 2017, Theorem 1) the authors used the fact that $\boldsymbol{V}_t^{-1} \leq \frac{1}{\lambda}\boldsymbol{I}$. This is not the case in our setting, but we can generalise the result with similar reasoning yielding $\boldsymbol{V}_t^{-1} \leq \frac{1}{\lambda_{min}(R^2\boldsymbol{V}_1)}\boldsymbol{I}$, where $\lambda_{min}(R^2\boldsymbol{V}_1)$ denotes the minimum eigenvalue of the matrix $R^2\boldsymbol{V}_1$.

We also need to change the definition of S, since our problem has shifted from estimating $\boldsymbol{\theta}$ to estimating $\boldsymbol{\delta}$. Therefore, after modifying the framework, the Warm Start Linear Thompson Sampling bandit can be summarised as in Algorithm 2, and admits the following regret bound.

Theorem 2 (Warm Start LinTS Regret Bound)**.** *Under the assumptions that:*

1. $\|\boldsymbol{x}\| \leq 1$ *for all* $x \in \mathcal{X}$*;*
2. $\|\boldsymbol{\delta}\| \leq S$ *for some known* $S \in \mathbb{R}^+$*; and*
3. *the conditionally* R*-subgaussian process* $\{\epsilon_t\}_t$ *is a martingale difference sequence given the filtration* $\mathcal{F}_t^x = (\mathcal{F}_1, \sigma(\boldsymbol{x}_1, r_1, \cdots, r_{t-1}, \boldsymbol{x}_t))$ *with* $\mathcal{F}_1$ *denoting any information on prior knowledge,*

along with the definition of $\mathcal{D}^{TS}$ *given in Section 2, then with probability at least* $1-\delta$*, with* $\delta' = \delta/(4T)$ *and* $\gamma_t = \beta_t(\delta')\sqrt{cd\log((c'd)/\delta)}$*, the regret of LinTS can*

Algorithm 2 Warm Start Linear Thompson Sampler

Input: $\hat{\boldsymbol{\mu}}, \alpha, \boldsymbol{\Sigma}_\mu, \delta, T, R$
Initialize $\hat{\boldsymbol{\delta}}_1 \leftarrow \mathbf{0}$, $\boldsymbol{V}_1 \leftarrow R^2(\boldsymbol{\Sigma}_\mu + \alpha^{-1}\boldsymbol{I}_d)^{-1}$, $\delta' \leftarrow \frac{\delta}{4T}, \boldsymbol{b}_1 \leftarrow \mathbf{0}$
for $t = 1, \ldots, T$ **do**
  Sample $\boldsymbol{\eta}_t \sim \mathcal{D}^{TS}$
  $\tilde{\boldsymbol{\theta}}_t \leftarrow \hat{\boldsymbol{\mu}} + \hat{\boldsymbol{\delta}}_t + \beta_t(\delta')\boldsymbol{V}_t^{-1/2}\eta_t$ {perturbed parameter}
  $i_t \leftarrow s \in \arg\max_{i\in[k]} \tilde{\boldsymbol{\theta}}_t^T \boldsymbol{x}_t(i)$ {optimal arm}
  Pull arm i_t and observe reward $r_t(i_t) = R_{i_t}(t)|\boldsymbol{x}_t(i_t)$
  $y_t(i_t) \leftarrow r_t(i_t) - \hat{\boldsymbol{\mu}}^T \boldsymbol{x}_t(i_t)$
  $\boldsymbol{V}_{t+1} \leftarrow \boldsymbol{V}_t + \boldsymbol{x}_t(i_t)\boldsymbol{x}_t^T(i_t)$ {update Eq. (3)}
  $\boldsymbol{b}_{t+1} \leftarrow \boldsymbol{b}_t + y_t(i_t)\boldsymbol{x}_t(i_t)$
  $\hat{\boldsymbol{\delta}}_{t+1} \leftarrow \boldsymbol{V}_{t+1}^{-1}\boldsymbol{b}_{t+1}$ {update Eq. (2)}
end for

be decomposed as

$$Reg(T) = R^{TS}(T) + R^{RLS}(T)\,,$$

with each of the term bounded as

$$R^{TS}(T) \leq \frac{4\gamma_T(\delta')}{p}\left(\sqrt{2T\log\frac{\det(\boldsymbol{V}_{t+1})}{\det(R^2\boldsymbol{V}_1)}} + \sqrt{\frac{8T}{\lambda_{min}(R^2\boldsymbol{V}_1)}\log\frac{4}{\delta}}\right)$$

$$R^{RLS}(T) \leq (\beta_T(\delta') + \gamma_T(\delta'))\sqrt{2T\log\frac{\det(\boldsymbol{V}_{t+1})}{\det(R^2\boldsymbol{V}_1)}}\,.$$

3.2. Extension to ϵ-Greedy and LinUCB Learners

The core idea of our warm-starting method as derived for Linear Thompson Sampling, lies in the method of setting up the initial phase of the bandit. The same expression of initial set up can be applied to other contextual bandit algorithms such as ϵ-Greedy and LinUCB.

In the ϵ-Greedy Algorithm, we balance exploration and exploitation by means of relatively naïve randomness: in each round we (uniformly) explore with probability ϵ and exploit with probability $1-\epsilon$. Specifically, by incorporating warm start, this means that at each round we choose an arm at random uniformly from the set $[k]$ with probability ϵ, and choose an arm at random uniformly from the set $S = \arg\max_{i\in[k]} \hat{\boldsymbol{\theta}}_t^T \boldsymbol{x}_t(i)$ with probability $1-\epsilon$. We summarise the Warm Start ϵ-Greedy Algorithm in Algorithm 3

We can also extend our warm-starting technique to LinUCB using the fact that $\boldsymbol{\theta} \sim \mathcal{N}(\hat{\boldsymbol{\mu}} + \boldsymbol{V}_t^{-1}\boldsymbol{b}_t, R^2\boldsymbol{V}_t^{-1})$, which is a powerful result. It was proposed by Li et al. (2010) that one way to interpret their algorithm is to look at the distribution of the expected payoff $\boldsymbol{\theta}_\star^T \boldsymbol{x}_t$. With the affine transformation property of multivariate Gaussian distributions, we have that $\boldsymbol{\theta}^T\boldsymbol{x} \sim \mathcal{N}(\hat{\boldsymbol{\theta}}_t^T\boldsymbol{x}, R^2\boldsymbol{x}^T\boldsymbol{V}_t^{-1}\boldsymbol{x})$. Therefore, the upper bound of such a quantity is:

$$\hat{\boldsymbol{\mu}}^T\boldsymbol{x} + (\boldsymbol{V}_t^{-1}\boldsymbol{b}_t)^T\boldsymbol{x} + \rho R\sqrt{\boldsymbol{x}^T\boldsymbol{V}_t^{-1}\boldsymbol{x}}$$

Algorithm 3 Warm Start ϵ-Greedy

Input: $\hat{\boldsymbol{\mu}}, \alpha, \boldsymbol{\Sigma}_\mu, \epsilon, T, R$
Initialize $\hat{\boldsymbol{\delta}}_1 \leftarrow \mathbf{0}$, $\boldsymbol{V}_1 \leftarrow R^2(\boldsymbol{\Sigma}_\mu + \alpha^{-1}\boldsymbol{I}_d)^{-1}, \boldsymbol{b}_1 \leftarrow \mathbf{0}$
for $t = 1, \ldots, T$ **do**
  Sample $u_t \sim \mathcal{U}(0,1)$
  if $u_t < \epsilon$ **then**
    choose $i_t \in [k]$ uniformly at random
  else
    $\hat{\boldsymbol{\theta}}_t \leftarrow \hat{\boldsymbol{\mu}} + \hat{\boldsymbol{\delta}}_t$
    $i_t \leftarrow s \in \arg\max_{i\in[k]} \hat{\boldsymbol{\theta}}_t^T \boldsymbol{x}_t(i)$ {optimal arm}
  end if
  Pull arm i_t and observe reward $r_t(i_t) = R_{i_t}(t)|\boldsymbol{x}_t(i_t)$
  $y_t(i_t) \leftarrow r_t(i_t) - \hat{\boldsymbol{\mu}}^T \boldsymbol{x}_t(i_t)$
  $\boldsymbol{V}_{t+1} \leftarrow \boldsymbol{V}_t + \boldsymbol{x}_t(i_t)\boldsymbol{x}_t^T(i_t)$ {update Eq. (3)}
  $\boldsymbol{b}_{t+1} \leftarrow \boldsymbol{b}_t + y_t(i_t)\boldsymbol{x}_t(i_t)$
  $\hat{\boldsymbol{\delta}}_{t+1} \leftarrow \boldsymbol{V}_{t+1}^{-1}\boldsymbol{b}_{t+1}$ {update Eq. (2)}
end for

Algorithm 4 Warm Start LinUCB

Input: $\hat{\boldsymbol{\mu}}, \alpha, \boldsymbol{\Sigma}_\mu, \rho, T, R$
Initialize $\hat{\boldsymbol{\delta}}_1 \leftarrow \mathbf{0}$, $\boldsymbol{V}_1 \leftarrow R^2(\boldsymbol{\Sigma}_\mu + \alpha^{-1}\boldsymbol{I}_d)^{-1}, \boldsymbol{b}_1 \leftarrow \mathbf{0}$
for $t = 1, \ldots, T$ **do**
  $\hat{\boldsymbol{\theta}}_t \leftarrow \hat{\boldsymbol{\mu}} + \hat{\boldsymbol{\delta}}_t$
  $i_t \leftarrow s \in \arg\max_{i\in[k]} \hat{\boldsymbol{\theta}}_t^T \boldsymbol{x}_t(i) + \rho R\sqrt{\boldsymbol{x}_t^T \boldsymbol{V}_t^{-1}\boldsymbol{x}_t}$
  Pull arm i_t and observe reward $r_t(i_t) = R_{i_t}(t)|\boldsymbol{x}_t(i_t)$
  $y_t(i_t) \leftarrow r_t(i_t) - \hat{\boldsymbol{\mu}}^T \boldsymbol{x}_t(i_t)$
  $\boldsymbol{V}_{t+1} \leftarrow \boldsymbol{V}_t + \boldsymbol{x}_t(i_t)\boldsymbol{x}_t^T(i_t)$ {update Eq. (3)}
  $\boldsymbol{b}_{t+1} \leftarrow \boldsymbol{b}_t + y_t(i_t)\boldsymbol{x}_t(i_t)$
  $\hat{\boldsymbol{\delta}}_{t+1} \leftarrow \boldsymbol{V}_{t+1}^{-1}\boldsymbol{b}_{t+1}$ {update Eq. (2)}
end for

for some value ρ, which is left as a hyperparameter. The summary of our Warm Start LinUCB Algorithm can be seen in Algorithm 4.

Theorem 3 (Warm Start LinUCB Regret Bound)**.** *The regret bound of warm-started LinUCB follows an argument of Lattimore and Szepesvári (2020) very closely. The regret, whose complete derivation is provided in the appendix, admits bound*

$$Reg(T) \leq \left(R\sqrt{2\log\left(\frac{\det(\boldsymbol{V}_T)^{\frac{1}{2}}\det(R^2\boldsymbol{V}_1)^{-\frac{1}{2}}}{\delta}\right)} + \sqrt{\lambda_{max}(R^2\boldsymbol{V}_1)}S \right) \cdot \sqrt{8T\log\left(\frac{\det(\boldsymbol{V}_{T+1})}{\det(R^2\boldsymbol{V}_1)}\right)}.$$

3.3. A Regret Lower Bound

We here present a lower bound for the warm-started bandit linear contextual ϵ-greedy algorithm. Consider the best-case scenario for ϵ-Greedy with constant ϵ, that is, that we have the true weight as our initial guess *i.e.*, $\hat{\boldsymbol{\mu}} = \boldsymbol{\theta}_\star$. Assume that we use the hyperparameter $\alpha \to \infty$, which ensures the weight's resistance to changes from observations, *i.e.*, $\hat{\boldsymbol{\theta}}_t = \hat{\boldsymbol{\mu}} = \boldsymbol{\theta}_\star$ for all t. With this setting, denoting $\Delta_{i,t} \geq 0$ as the difference between the expected rewards of the optimal arm and arm i at round t, the regret is $\frac{\epsilon}{K}\sum_{t=1}^{T}\sum_{i=1}^{K}\Delta_{i,t}$. This argument, detailed in Lemma 4, proves a lower bound since it is derived from a best case scenario.

Lemma 4. *The regret for warm-started ϵ-greedy is at best $\frac{\epsilon}{K}\sum_{t=1}^{T}\sum_{i=1}^{K}\Delta_{i,t}$.*

Proof. Since $\hat{\boldsymbol{\theta}}_t = \boldsymbol{\theta}_\star$ for all t, each exploitation round will yield one of the optimal arms with probability 1. Assume that there are K arms in total. Let E denote the event that exploration occurs, and A_i be the event that arm i is chosen. Then the expected cumulative regret for the linear contextual ϵ-greedy is:

$$\begin{aligned} R(T) &= \sum_{t=1}^{T}[0P(E^c) + \sum_{i=1}^{K}\Delta_{i,t}P(E \cap A_i)] \\ &= \frac{\epsilon}{K}\sum_{t=1}^{T}\sum_{i=1}^{K}\Delta_{i,t} \\ &= \frac{\epsilon T}{K}\sum_{i=1}^{K}\bar{\Delta}_i \ , \end{aligned}$$

where $\bar{\Delta}_i$ is the average of $\Delta_{i,t}$ over t, *i.e.*, $\bar{\Delta}_i = \frac{1}{T}\sum_{t=1}^{T}\Delta_{i,t}$. □

Note that in this analysis we have used a constant ϵ for our ϵ-Greedy algorithm. In practice the value of ϵ can be scheduled to recede over time. Auer, Cesa-Bianchi and Fischer (2002) have shown that in the case of non-contextual bandits, this regime enjoys a sub-linear upper regret bound.

Reduction From Non-Contextual to Contextual Bandits. The above lower bound of the contextual ϵ-Greedy algorithm leads naturally to a lower bound for non-contextual bandits. The non-contextual bandit is different from its contextual counterpart where it does not provide any context. In each round, the true means of each non-contextual arm remain constant and are independent of each other (*i.e.*, $\theta_{i,t} = \theta_i$ for all t), thus the parameters to estimate are θ_i for arm $i \in [K]$. A non-contextual bandit can be formulated as a contextual bandit, as shown in Lemma 5. By performing such a reduction, essentially using a contextual bandit to act in a non-contextual setting, we can relate lower bounds between the settings.

Lemma 5. *A non-contextual bandit can be formulated as a contextual bandit. Therefore, any fundamental limitations for non-contextual bandits must also hold for contextual bandits.*

Proof. Let the non-contextual bandit arm be $i = 1, \ldots, K$ and let the expected reward for arm i be θ_i. A contextual bandit equivalent can be constructed by

setting the context for arm i as $\boldsymbol{x}(i) = \boldsymbol{e}_i$, which is the standard basis of $\mathbb{R}^K$, *i.e.*, the vector whose element is 1 in its i^{th} element and 0 otherwise. Furthermore, assuming that the shared model is used, then the i^{th} element of the true weight $\boldsymbol{\theta}_\star$ can be taken to be θ_i. This setting leads us to set the initial weight $\hat{\boldsymbol{\mu}} = \begin{bmatrix}\hat{\mu}_1 & \cdots & \hat{\mu}_K\end{bmatrix}^T$ to provide an initial guess of the true mean of each arm μ_i for $i \in [K]$, with $\boldsymbol{V}_1 = \text{diag}(\lambda_1, \cdots, \lambda_K)$ reflecting the confidence we have for our initial estimate. A diagonal matrix is particularly chosen for this purpose since the means of each arm are independent of each other. Thus, the (contextual) estimate of $\boldsymbol{\theta}_\star$ is

$$\hat{\boldsymbol{\theta}}_{t+1} = \hat{\boldsymbol{\mu}} + \boldsymbol{V}_{t+1}^{-1}\boldsymbol{b}_{t+1} = \hat{\boldsymbol{\mu}} + \left(\boldsymbol{V}_1 + \sum_{s=1}^{t} \boldsymbol{x}_s \boldsymbol{x}_s^T\right)^{-1} \sum_{s=1}^{t} (r_s - \hat{\boldsymbol{\mu}}^T \boldsymbol{x}_s)\boldsymbol{x}_s \ .$$

Now since $\boldsymbol{x}_s = \boldsymbol{e}_{i_s}$, and noticing that $\boldsymbol{e}_i \boldsymbol{e}_i^T = \text{diag}(\mathbb{1}(i=1), \cdots, \mathbb{1}(i=K))$ for all $i \in [K]$, *i.e.*, a matrix with all zero entries except at entry (i,i) with value 1, we have

$$\sum_{s=1}^{t} \boldsymbol{x}_s \boldsymbol{x}_s^T = \text{diag}\left(\sum_{s=1}^{t} \mathbb{1}(i_s = 1), \cdots, \sum_{s=1}^{t} \mathbb{1}(i_s = K)\right) = \text{diag}(T_1, \cdots, T_K),$$

$$r_s - \hat{\boldsymbol{\mu}}^T \boldsymbol{x}_s = r_s - \hat{\mu}_{i_s}$$

and

$$\sum_{s=1}^{t} (r_s - \hat{\boldsymbol{\mu}}^T \boldsymbol{x}_s)\boldsymbol{x}_s = \begin{bmatrix} w_1 & \ldots & w_K \end{bmatrix}^T \ ,$$

where T_i is the number of times arm i is pulled and $w_i = \sum_{s=1}^{t} (r_s - \hat{\mu}_i)\mathbb{1}(i_s = i) = \sum_{s=1}^{t} r_s \mathbb{1}(i_s = i) - T_i \hat{\mu}_i$ is the total sum of all the reward differences observed by arm i. Therefore, the estimate of the weight is

$$\begin{aligned}
\hat{\boldsymbol{\theta}}_{t+1} &= \hat{\boldsymbol{\mu}} + \boldsymbol{V}_{t+1}^{-1}\boldsymbol{b}_{t+1} \\
&= \begin{bmatrix}\hat{\mu}_1 & \cdots & \hat{\mu}_K\end{bmatrix}^T + \\
&\qquad [\text{diag}(\lambda_1, \cdots, \lambda_K) + \text{diag}(T_1, \cdots, T_K)]^{-1} \begin{bmatrix} w_1 & \cdots & w_K \end{bmatrix}^T \\
&= \begin{bmatrix}\hat{\mu}_1 & \cdots & \hat{\mu}_K\end{bmatrix}^T + [\text{diag}(\lambda_1 + T_1, \cdots, \lambda_K + T_K)]^{-1} \begin{bmatrix} w_1 & \cdots & w_K \end{bmatrix}^T \\
&= \begin{bmatrix}\hat{\mu}_1 + \frac{w_1}{\lambda_1 + T_1} & \cdots & \hat{\mu}_K + \frac{w_K}{\lambda_K + T_K}\end{bmatrix}^T \\
&= \begin{bmatrix}\frac{\hat{\mu}_1 \lambda_1 + \sum_{s=1}^{t} r_s \mathbb{1}(i_s=1)}{\lambda_1 + T_1} & \cdots & \frac{\hat{\mu}_K \lambda_K + \sum_{s=1}^{t} r_s \mathbb{1}(i_s=K)}{\lambda_K + T_K}\end{bmatrix}^T \\
&= \begin{bmatrix}\hat{\theta}_1 & \ldots & \hat{\theta}_K\end{bmatrix}^T \ .
\end{aligned}$$

This result can be interpreted such that for each arm $i \in [K]$, our estimate of the true mean θ_i is its sample mean with a pseudo-observation of mean $\hat{\mu}_i$ worth of λ_i observations. Indeed, when we choose $\lambda_i = 0$ for all $i \in [K]$, we recover each arm's mean estimate typically calculated by a non-contextual bandit ϵ-greedy algorithm. With this, when we exploit, we choose an arm which maximise $\hat{\boldsymbol{\theta}}^T x(i) = \hat{\boldsymbol{\theta}}^T \boldsymbol{e}_i = \hat{\theta}_i$, which is the same as what is performed in the non-contextual case.

□

Since a non-contextual bandit can be formulated as a contextual bandit, our approach may be applied to warm-start a non-contextual bandit. Its lower bound when the ϵ-greedy algorithm is used follows the lower bound of contextual ϵ-greedy, with $\Delta_{i,t} = \bar{\Delta}_i$ for all t since the mean reward (hence the regret each arm) is stationary across t. In other words, Lemma 4 is a fundamental lower bound on our warm-start setting also.

4. Experiments

We now report on a comprehensive suite of experimental evaluations of our warm start framework against a number of baselines and different datasets. We are interested in the benefit of warm start over cold start—in such cases we focus on short-term performance differences, as this is a practical limitation of bandits in high-stakes applications. We also explore the impact of prior misspecification as a potential risk of incorrect warm start. We summarise our experiments next, and then describe them with results in more detail below.

Datasets. Experiments in database index selection explore the effect of warm start in selecting a single index per round where queries arrive to the database in batches and rewards correspond to (negative) execution time. We use a commercial database system, and the standard TPC-H benchmark (TPC, n.d.). Results on two OpenML datasets (Letters and Numbers) test bandits on online multi-class classification, as a benchmark previously used to evaluate the ARRoW warm-start technique (Zhang et al., 2019). These datasets are advantageous to ARRoW in that they supply the (restrictive) kind of prior knowledge needed—supervised pre-training. Experiments on synthetic data provide sufficient control of the environment to explore limitations of our warm start approach.

Baselines. On the database index selection task, we use cold start TS as a natural and fair baseline. On the OpenML datasets we include the ARRoW warm-start framework, which was originally tested in the same way. We also demonstrate the performance of both frameworks on the ϵ-greedy and LinUCB learners, as well as LinTS. Where *cold start* corresponds throughout to having no pre-training dataset (*i.e.*, Algorithm 1), *hot start* in the synthetic experiment corresponds to having 100% accuracy on the pre-training parameter $\boldsymbol{\mu}_\star$, and *warm start* corresponds to having an estimate on the pre-training parameter $\boldsymbol{\mu}_\star$, namely $\hat{\boldsymbol{\mu}}$. By its very nature, we can only produce hot start results with the artificial dataset, since 100% accuracy on the pre-training parameter requires an infinite amount of observation in the real world database index selection problem.

Hardware. All experiments are performed on a commodity laptop equipped with Intel Core i7-6600u (2 cores, 2.60GHz, 2.81GHz), 16 GB RAM, and 256 GB disk (Sandisk X400 SSD) running Windows 10. In database experiments, we report cold runs only: we clear database buffer caches prior to query execution—the memory setting thus does not impact our findings.

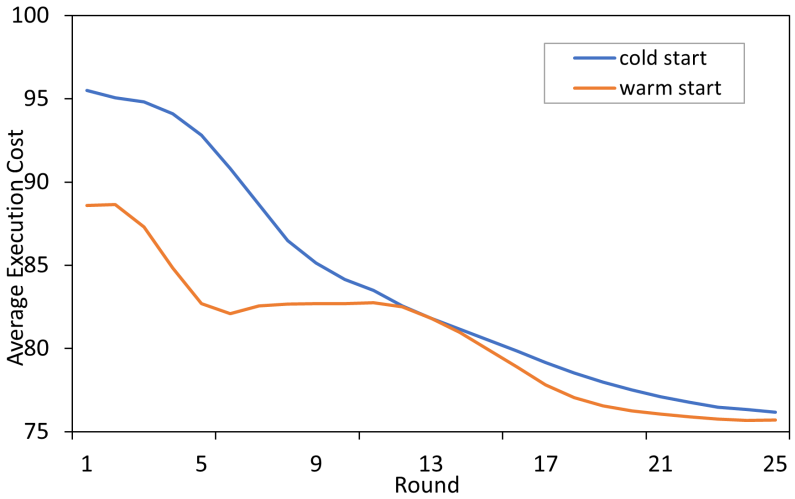


Fig. 1. Cold Start vs. Warm Start LinTS for database index selection on the the TPC-H benchmark.

4.1. Database Index Selection

As the real-world problem of database index selection motivated this work, we begin with a demonstration in this setting. In a database management system, an index is a data structure used to speed up database execution of a set of queries (*a.k.a workload*). While a huge space of possible indices could be considered, only a few can actually be created due to memory constraints (since each index occupies space in memory). With a tremendous number of indices, it is impractical for humans to decide which indices to create without assistance. A recent effort has been made to automate this task by using bandits (Perera et al., 2021) to propose an optimal set of indices to boost the workload execution. This recent framework we will be adopted in our work and expanded to support warm start. The aim of this experiment is to demonstrate that the warm-started bandit will yield similar performance as the cold-started bandit in the long run whilst having better performance in earlier rounds. The consequence of such a demonstration, is a system more suitable for deployment.

In particular, our problem setting is as follows. At round $t = 1, 2, \ldots, T$, we observe a workload W_t with a set of queries, and the system recommends one index i_t out of the set of all possible indices $\mathcal{I}$. After index i_t is created, we execute the queries in workload W_t. Our chosen aim is to minimise the query execution time, noting we do not take into account the time it takes to create the index i_t. After q_t is executed, the index i_t is dropped and the buffer is cleaned.

In this paper, the adopted database comes from the TPC-H benchmark (TPC, n.d.). This publicly available industrial benchmark comes with a set of predefined query templates. A query template is a parameterized query whose parameter values (a.k.a conditions) are missing, keeping only the structure of the query and leaving number and string values as variables. We chose five query templates at random and instantiated them with actual parameter values in each round. These queries will be used as the workload in both pre-training and deployment phase.

It should be noted that the value of R and S are unknown in the real-world

dataset. In this case, we treat these as hyperparameters which need to be chosen, adding to α.

In running this experiment, we have used the context features as described by Perera et al. (2021), with the reward being the performance gain, described as $t_{no\_index} - t_i$, where t_{no_index} corresponds to the execution time of the whole workload without any indices and t_i the execution time of the whole queries in the workload using index i.

Due to the lack of information on the most optimal index, it is impossible to retrieve the regret for each round. Therefore, with this real-world experiment, we present the average execution time (loss) of workload W_t based on what both algorithms recommend, which can be found in Figure 1.

Results. It can be seen that the warm-started LinTS outperforms the cold-started LinTS, in short-term rounds and cumulatively. This can be explained by the query templates used to pre-train the warm-started bandit resembling the templates used in the testing dataset. This leads the warm-started bandit's guess of the initial weight $\boldsymbol{\theta}_1 = \hat{\boldsymbol{\mu}}$ being closer to the actual weight $\boldsymbol{\theta}_\star$ compared to the initial guess of $\boldsymbol{\theta}_1 = \mathbf{0}$ by the cold-started bandit.

4.2. OpenML Classification Dataset

We chose two of the datasets used in (Zhang et al., 2019), which correspond to letters and numbers identification respectively. We split the data such that 10% is used as the supervised learning examples and the other 90% used as the actual bandit rounds. This advantages ARRoW (Zhang et al., 2019) as the only form of permissible prior knowledge. We try all learners presented in this paper for this dataset: ϵ-greedy, LinUCB and LinTS. As for the hyperparameters, we used $\epsilon = 0.0125$ for ϵ-greedy, $\rho R = 0.2$ for LinUCB, $\beta_t(\delta) = 1$ for LinTS in Letter dataset and $\beta_t(\delta) = 0.05$ for LinTS in Numbers dataset with $R = 0.25$. All of these hyperparameters were found iteratively by grid search.

As described in (Zhang et al., 2019), we transform the dataset into a dataset capable of evaluating bandit algorithm by mapping the classes as the arms and the cost of each class as $c(a) = \mathbb{1}(a \neq y)$ given example (x, y). For the classification problem, we also modify our bandit algorithm which usually shares its parameter across the arms. However, since the context of each arm is the same for the classification task, we distinguish the value by making the parameter different, leading to the disjoint bandit with arm i having the weight $\boldsymbol{\theta}_{\boldsymbol{i,t}}$. As such its reward is modelled by the equation $r_t(i) = \boldsymbol{\theta}_{i,\star}^T \boldsymbol{x}_t(i) + \epsilon_t(i)$

We have used the term cost instead of rewards in this dataset, which requires minor modification of the learners: we change the argmax operation into argmin and in the case of LinUCB, the Upper Confidence Bound in Line 5 to Lower Confidence Bound $\hat{\boldsymbol{\theta}}_{i,t}^T \boldsymbol{x}_t(i) - \rho R \sqrt{\boldsymbol{x}_t^T(i) \boldsymbol{V}_t^{-1} \boldsymbol{x}_t(i)}$.

The ARRoW algorithm presented in (Zhang et al., 2019) is also executed partially, with the size of the class $|\Lambda|$ set to 1. We chose the best performing λ to be compared against our algorithm, for fairness. We note that sensitivity analysis in Figure 3 and Figure 4, demonstrate that the choices are generally not very important.

We follow a suggestion of the original ARRoW paper to evaluate (Zhang

et al., 2019, Algorithm Line 5), evaluating

$$\arg\min_{f\in\mathcal{F}} \left\{ (1-\lambda) \sum_{(x,c)\in S} \sum_{a=1}^{K} (f(x,a) - c(a))^2 + \lambda \sum_{\tau=1}^{t} \frac{1}{p_{\tau,a_\tau}} (f(x_\tau, a_\tau) - c_\tau(a_\tau))^2 \right\}$$

where $f(x,a)$ is a linear function and $\mathcal{F}$ is the class of all linear functions. The solution of which can be obtained via the weighted linear regression.

Another algorithm we used for comparison is by Li et al. (2021), hereby labelled as WWW'21 for convenience (denoting the publication venue). This algorithm employs virtual plays in every round by sampling the context according to a cdf $F_X(\boldsymbol{x})$, estimated by its empirical cdf $\hat{F}_X(\boldsymbol{x})$, ultimately equivalent to random sampling of the seen contexts with replacement. A feedback is provided by an offline evaluator whenever the online confidence band is wider than the offline counterpart. The virtual plays are continued indefinitely until the offline evaluator does not give a feedback.

We present the results for the OpenML Dataset in Figure 2, where we have labelled our algorithm *diff* for the fact that our algorithm models the difference between the true parameter from the guessed weight. It can be seen that our algorithm performs as well as previous algorithms, whilst still offering the flexibility to choose the initial guess.

Sensitivity analysis for this experiment (with accurate prior) is presented in Figure 3 and Figure 4. As mentioned, neither ARRoW nor our warm start approach are very sensitive to their hyperparameters, while the algorithm proposed by Li et al. (2021) does not require any hyperparameter tuning. These results also support our choice of $\alpha = 10^7$ across these experiments.

Effect of Warm-Start on Exploration Hyperparameters. In this section, we present the final cumulative cost as a means of measuring the performance of warm-started bandit under different exploration hyperparameters. As previously observed from Figures 3 and 4, the temperature hyperparameter does not appear to have a significant impact on final performance. Thus, for this analysis, we again fixed the value $\alpha = 10^7$. We reran the experiment for both Letters and Numbers datasets using the ϵ-greedy, LinUCB, and LinTS algorithms, varying the value of the exploration hyperparameters ϵ, ρR and β respectively. The results, as shown in Figure 5, suggest that lower values of the exploration hyperparameters are preferred. This is intuitive since a goal of warm-starting bandits is to reduce the demand on exploration during initial rounds. This effect is very prominent especially in the ϵ-greedy algorithm. This can be explained by the fact that exploration in the ϵ-greedy is strictly dictated by the value of ϵ, while in LinUCB and LinTS the exploration terms are partly influenced by the matrix $\boldsymbol{V}_t$, which initially depends on the covariance matrix $\boldsymbol{\Sigma}_\mu$. Therefore, in ϵ-greedy, we recommend 'manually' reducing the exploration hyperparameter ϵ, while in LinUCB the exploration is partially automatically reduced thanks to the lower exploration boost when $\boldsymbol{\Sigma}_\mu$ has smaller eigenvalues.

Effect of Pre-Training Data Ratio on Performance. As previously done in Zhang et al. (2019), we can explore the fraction of the dataset available for

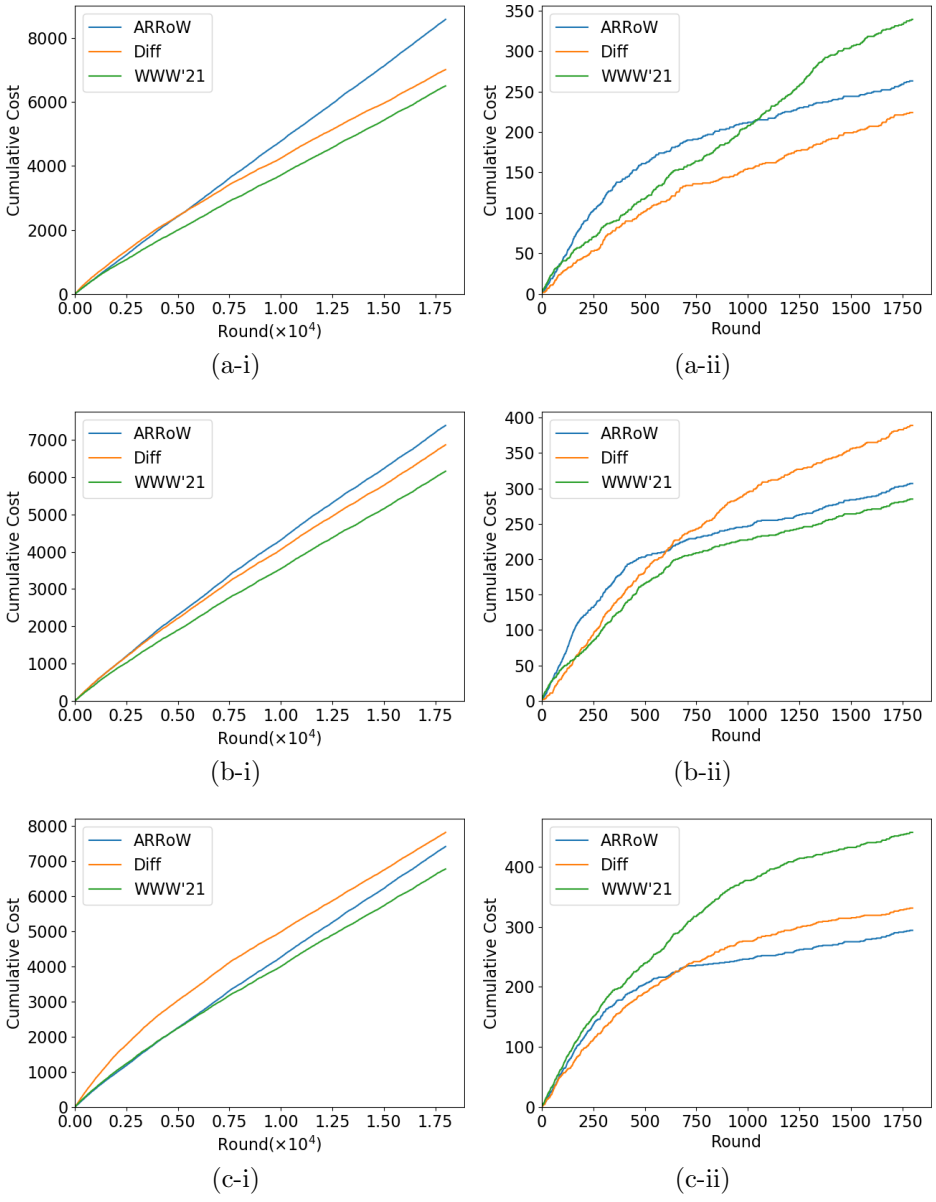
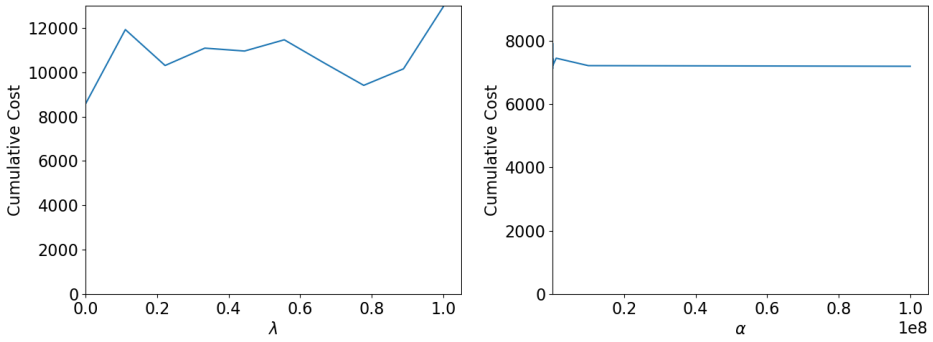
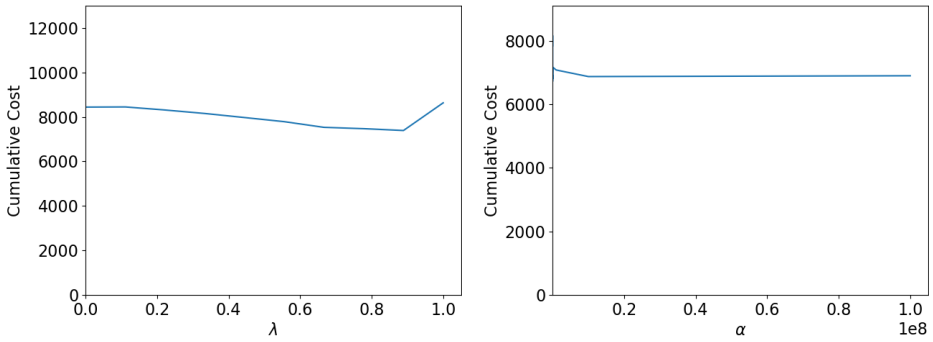


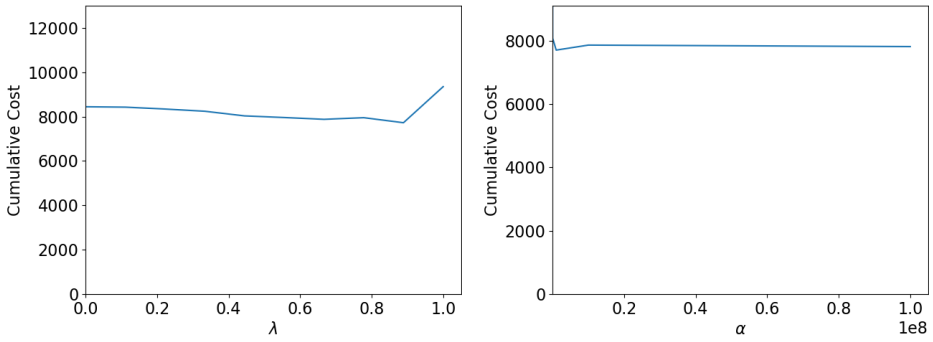
Fig. 2. Comparisons of both our and ARRoW warm-start frameworks on the (column i) Letters and (ii) Numbers datasets, with learners (row a) ϵ-greedy, (b) LinUCB and (c) LinTS.



(a-i) (a-ii)

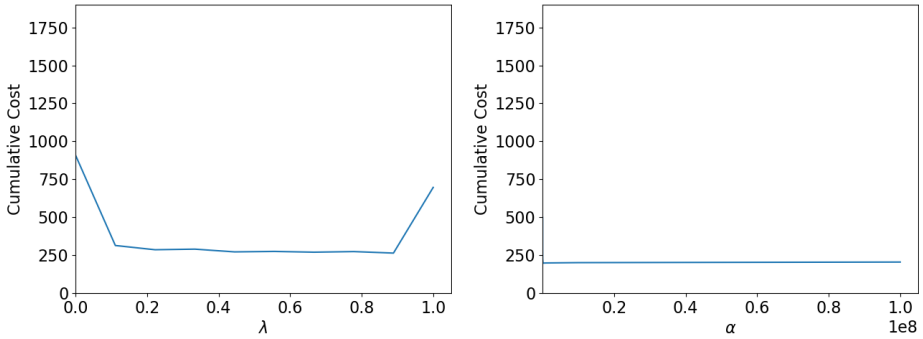


(b-i) (b-ii)

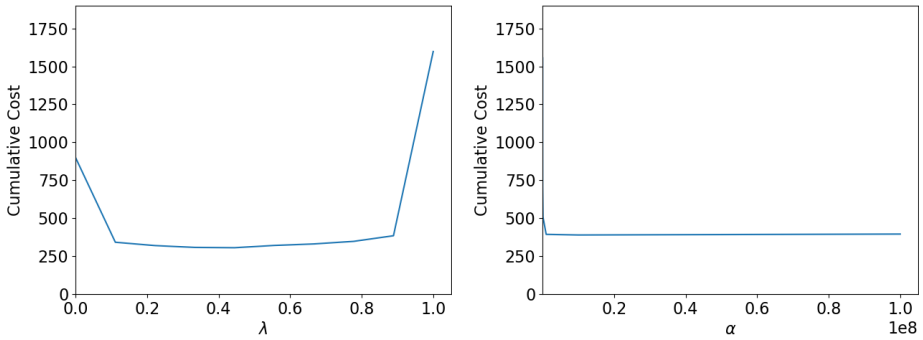


(c-i) (c-ii)

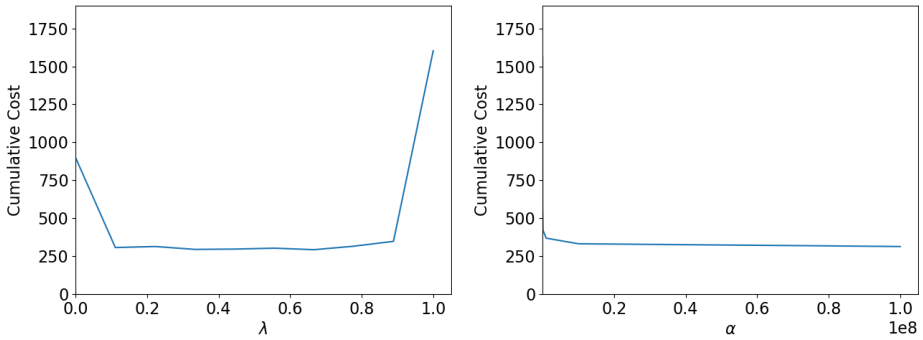
Fig. 3. Sensitivity analysis showing total cumulative cost achieved vs. hyperparameter on the Letters dataset. Column (i) demonstrates ARRoW results with varying λ while column (ii) shows our warm start approach Diff with varying α. Finally the learners vary over (row a) ϵ-greedy, (b) LinUCB, (c) LinTS.



(a-i) (a-ii)



(b-i) (b-ii)



(c-i) (c-ii)

Fig. 4. Sensitivity analysis showing total cumulative cost achieved vs. hyperparameter on the Numbers dataset. Column (i) demonstrates ARRoW results with varying λ while (ii) shows our warm start approach Diff with varying α. Finally the learners vary over (row a) ϵ-greedy, (b) LinUCB, (c) LinTS.

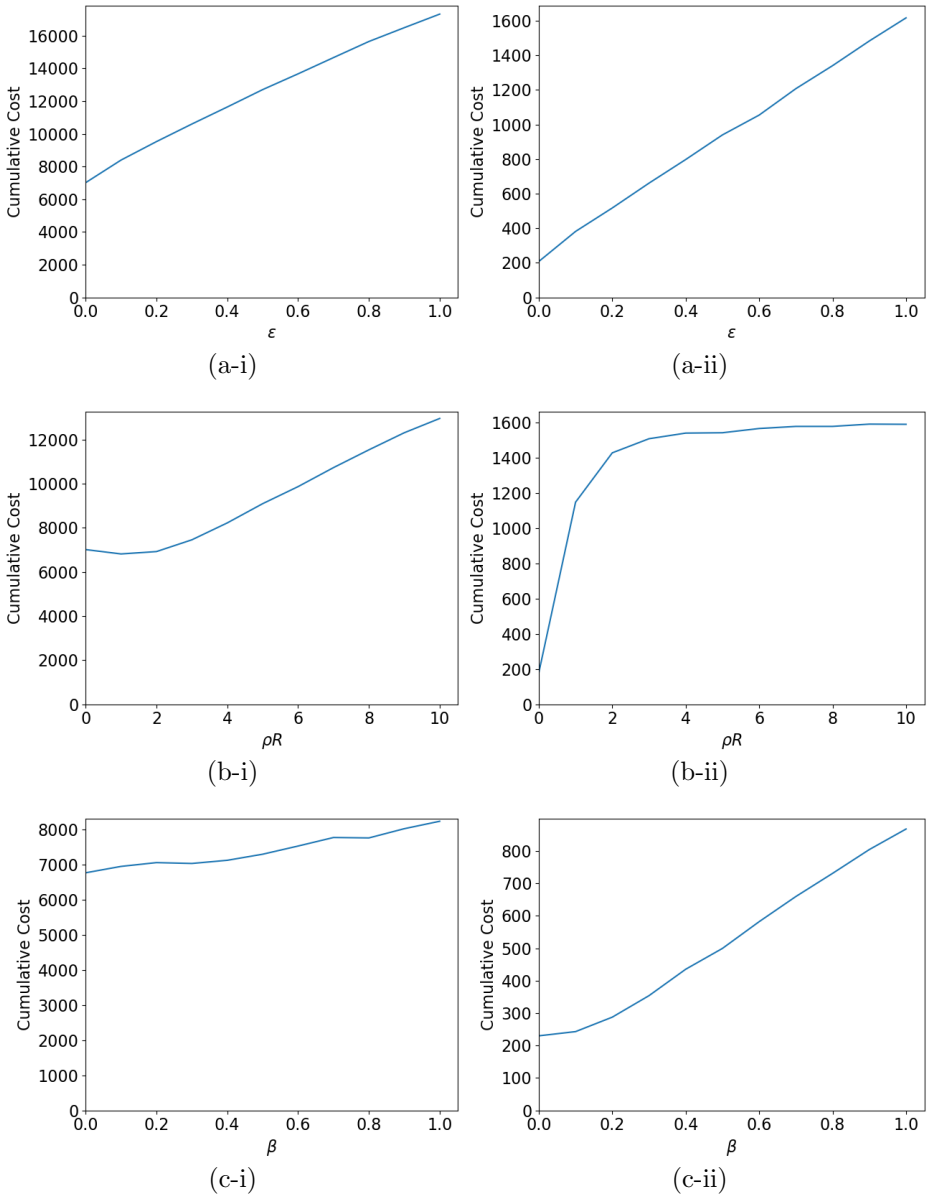


Fig. 5. Effect of warm-starting the bandits showing total cumulative cost achieved vs. exploration hyperparameter. Column (i) is on the Letters dataset while column (ii) is on Numbers. The learners vary over (row a) ϵ-greedy, (b) LinUCB and (c) LinTS. The performance appears better when the exploration hyperparameter is relatively small.

pre-training. In this section, we present how the cumulative cost evolves as the pre-training dataset to total dataset ratio changes. Here the total dataset refers to the union between the pre-training dataset and the bandit deployment dataset. In particular, we investigate the performance for each of the ratios in $\{0, 0.001, 0.002, 0.003, 0.004, 0.005, 0.01, 0.05, 0.1\}$. For fairness, all experiments from the different ratios in the same dataset share the same deployment data, thus the maximum ratio in the experiment, which is 0.1, is used to determine the deployment dataset. Since there are 20000 data in Letters Dataset and 2000 data in Numbers dataset, we used the last 18000 and 1800 data in Letter and Number Dataset, respectively. Figure 6 supports the intuition that higher ratios likely lead to better performance. This effect is particularly apparent during the initial increase, while the gain gradually fades away as the ratio is increased further. This diminishing return can be explained since the biggest improvement in the correctness of θ occurs in the beginning of the supervised learning, whereas its accuracy, while increasing, improves more slowly as more data is observed.

Effect of Misspecified Pre-Training Data Ratio on Performance. A series of experiments investigating sensitivity to the warm-start temperature and exploration hyperparameters was carried out. We also investigated the effect of the fraction of dataset used as pre-training in both settings: accurate prior and misspecified prior.

We investigated the effect of a misspecified prior with both datasets. For this, we need to create another dataset in which the true weight $\theta_\star$ is different from the deployment dataset's. To do this, we have trained a linear regression for the whole dataset for each arm i, giving us the disjoint parameter $\boldsymbol{\theta}_1(i)$, which is then transformed by a rotation matrix $\boldsymbol{R}_\gamma$ to give a new parameter $\boldsymbol{\theta}_2(i) = \boldsymbol{R}_\gamma \boldsymbol{\theta}_1(i)$. For each datum at round t used for pre-training, we extracted the context $\boldsymbol{x}_t(i)$ for all arms, then calculate $d_r(i) = (\boldsymbol{\theta}_2(i) - \boldsymbol{\theta}_1(i))^T \boldsymbol{x}_t(i)$. This acts as the perturbation of the original reward $r_t(i)$, yielding the inaccurate reward $r'_t(i) = r_t(i) + d_r(i)$. In our data generation, we have calculated the similarities between the two parameters, yielding the similarities $\cos(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2) = \frac{\langle \boldsymbol{\theta}_1, \boldsymbol{\theta}_2 \rangle}{\|\boldsymbol{\theta}_1\| \|\boldsymbol{\theta}_2\|} = \frac{1}{\sqrt{2}}$ for all arms and both datasets. This consistent rotation attempts to maintain a similar amount of misspecification across datasets, however as we shall see, properties of the data interact with the magnitude of perturbation.

Due to the nature of the semi-synthetic dataset generation process, the reward might no longer be in $\{0, 1\}$ as previously generated from the classification problem. This observation does not effect the validity of the model, or appropriateness of warm-start in this setting thanks to the flexibility of reward structures accommodated.

We present our result in Figure 7. Differing to the previous experiment, we no longer have the privilege to have a very similar dataset as our pre-training data. It can be seen that for the Letter dataset, some warm-starting provides a modest initial boost to performance, while warm-starting appears to hurt the performance in Numbers dataset.

4.3. Synthetic Experiments

In generating the artificial dataset, we started off by choosing a value for $\boldsymbol{\theta}_\star$. In this case, we chose the value to be $\boldsymbol{\theta}_\star^T = [0.1 \quad 0.3 \quad 0.5 \quad 0.7 \quad 0.9]$, with the

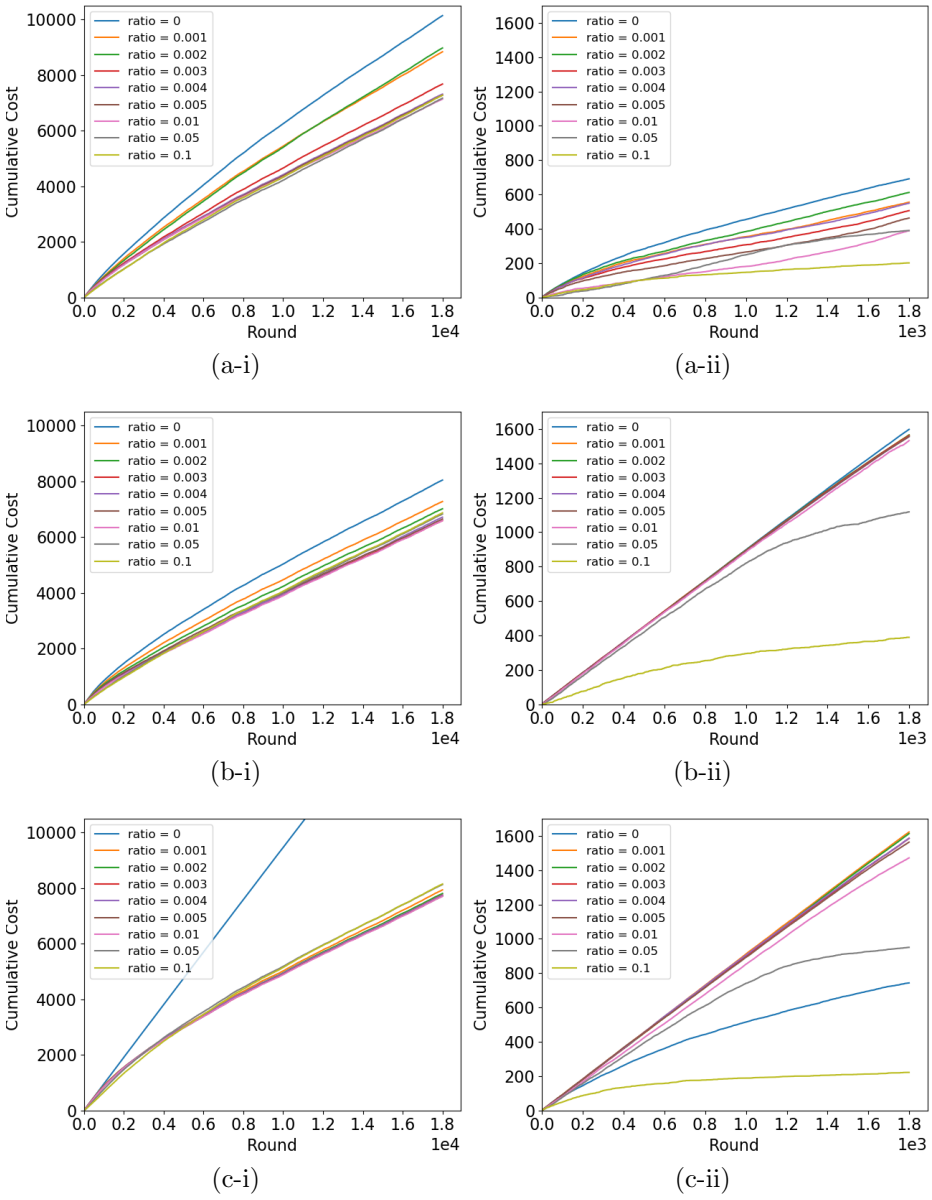


Fig. 6. Effect of different ratios of pre-training data (fraction of full dataset used in pre-training). Column (i) is on the Letters dataset while column (ii) is on Numbers. The learners vary over (row a) ϵ-greedy, (b) LinUCB and (c) LinTS. The 2001^{st} to 20000^{th} data and the 201^{st} to 2000^{th} data is used as the deployment data in Letter and Number Dataset respectively regardless of the ratio used.

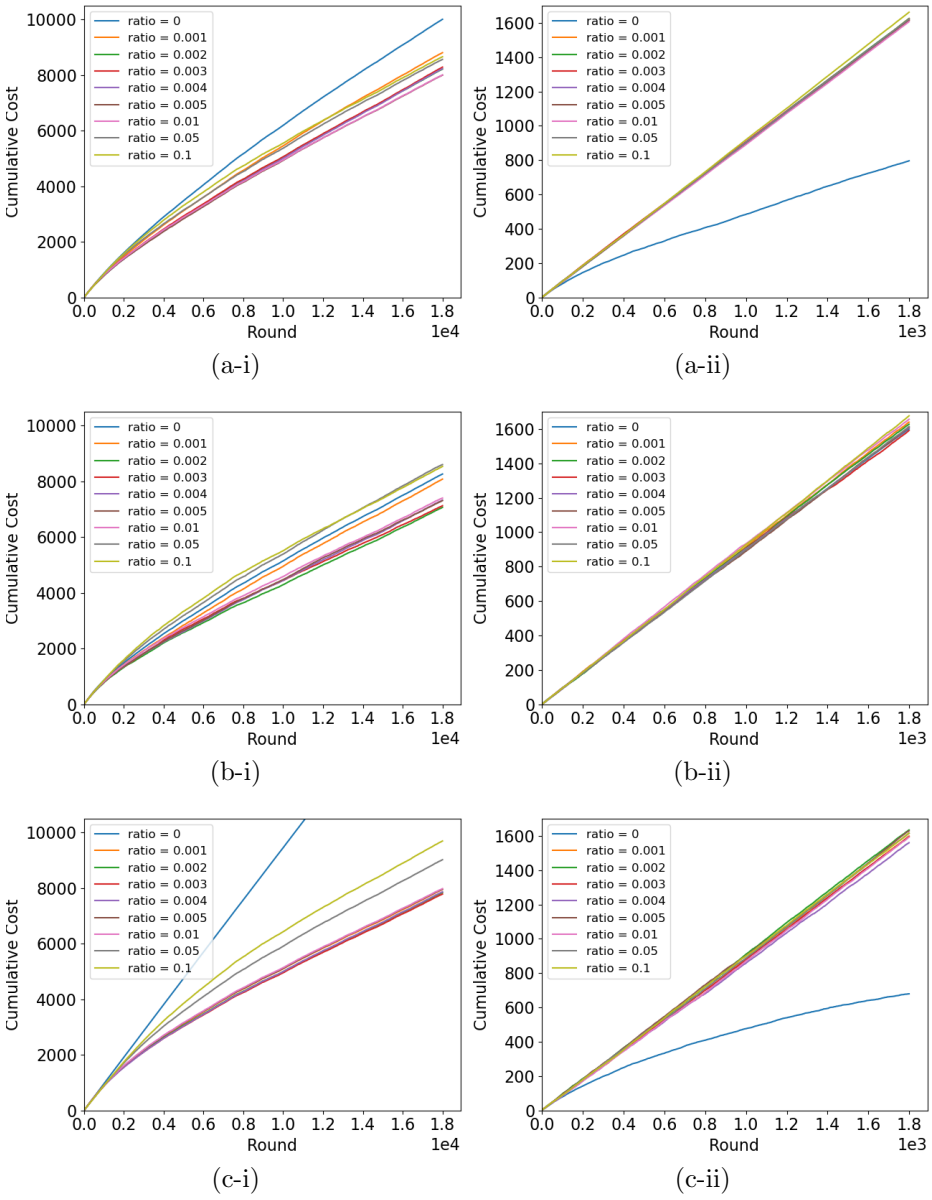
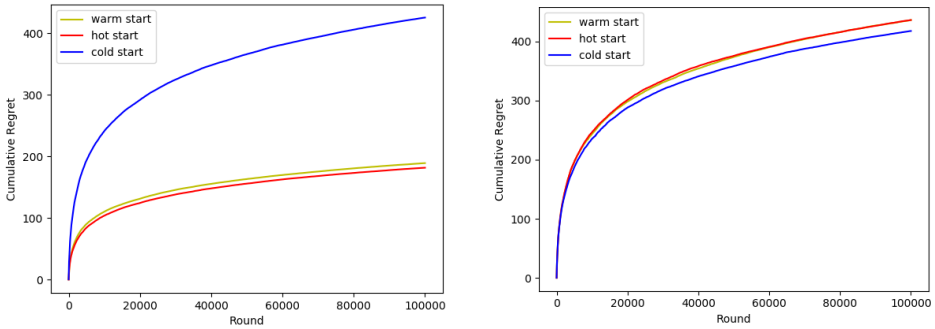


Fig. 7. Effect of different fractions of misspecified pre-training data. Column (i) is on the Letters dataset while column (ii) is on Numbers. The learners vary over (row a) ϵ-greedy, (b) LinUCB and (c) LinTS. The 2001^{st} to 20000^{th} data and the 201^{st} to 2000^{th} data is used as the deployment data in Letter and Number Dataset respectively regardless of the ratio used.



(a) (b)

Fig. 8. Artificial dataset experimental results for (a) an accurate prior and (b) a misspecified prior, comparing cold-, warm- and hot-start LinTS.

bandit having 10 arms. After the value of $\boldsymbol{\theta}_\star$ is chosen, we generate a random vector $\boldsymbol{x}_t(i) \in \mathbb{R}^d$, $d = 5$ where each element is drawn from uniform distribution $U(0, 1)$ for each $i = 1, 2, \cdots, 10$, followed by taking the inner product and adding the Gaussian noise $\epsilon_i(t) \sim \mathcal{N}(0, R^2)$, $R = 0.25$, independent on the arm i and round number t. The noisy reward $r_i(t) = \boldsymbol{\theta}_\star^T \boldsymbol{x}_t(i) + \epsilon_i(t)$ is saved, as well as the regret of pulling arm i, namely $\boldsymbol{\theta}_\star^T \boldsymbol{x}_t(i) - \max_{i \in [k]} \boldsymbol{\theta}_\star^T \boldsymbol{x}_t(i)$. This makes it possible to compare all bandit algorithms equally without needing *off-policy evaluation*. We repeat this process 100,000 times, which corresponds to 100,000 rounds of the second phase dataset.

To generate the pre-training dataset, we firstly choose the value of α^{-1}, before sampling the true parameter deviation $\boldsymbol{\delta}_\star \sim \mathcal{N}(\mathbf{0}, \alpha^{-1}\boldsymbol{I})$. After the value $\boldsymbol{\delta}_\star$ is sampled, we calculate $\boldsymbol{\mu}_\star = \boldsymbol{\theta}_\star - \boldsymbol{\delta}_\star$ and conducted the process exactly as we generated the second phase dataset. We generated two types of pre-training dataset: accurate prior, where we chose $\alpha^{-1} = 10^{-4}$ and misspecified prior, where we chose $\alpha^{-1} = 0.25$. We produced 10,000 rounds worth of pre-training dataset.

We observed that, with the dataset generated both from the accurate and misspecified prior regime, $\alpha = 10$ seems to be the cut-off point where all algorithms work quite well. Therefore, we plot for all warm-starting methods the cumulative regret for $\alpha = 10$, as shown in Figure 8.

Results. In the accurate prior regime, it is clear that the hot-started and warm-started bandits outperform the cold-started bandit. This can be explained by the fact that the value of $\boldsymbol{\theta}_\star$ is closer to $\hat{\boldsymbol{\mu}}$ or $\boldsymbol{\mu}_\star$ as opposed to $\mathbf{0}$. However, the opposite problem occurs when the prior is misspecified, as the cold-start bandit slightly outperforms the hot-started bandit and warm-started bandit, due to the fact that $\boldsymbol{\theta}_\star$ is closer to $\mathbf{0}$ compared to $\hat{\boldsymbol{\mu}}$ or $\boldsymbol{\mu}_\star$.

It should be noted as well, that we have held the hyperparameter α the same for all regimes here. When the hyperparameter α is tuned optimally, the hot-started and cold-started bandits are able to perform even better, as the pre-training dataset is treated as if they are the real dataset.

5. Towards Adaptive Drift Hyperparameter

In this section we take a closer look at a key hyperparameter of our warm-start algorithms: the drift hyperparameter α which controls how much exploration follows pre-training. While this has so far been set manually, based on how much the operator believes pre-training to be aligned with deployment time, in practice we believe this parameter may sometimes be difficult to set.

Limitations of the current approach. The advantage of our current approach of warm-starting as applied in Algorithms 2, 3 and 4 has been centralised around the selection of the drift hyperparameter. This drift hyperparameter α has been used as a means for temperature tuning: how much can we trust the initial weight guess? With an accurate prior, a sufficiently large value of α will give the bandit an early advantage in the deployment phase as unnecessary exploration is eliminated. On the other hand, although the warm-started bandit is somewhat insensitive to α with an accurate prior, its sensitivity will be largely augmented when the prior is highly misspecified; a large α value makes the bandit retain its highly misaligned initial guess and resist changes made from observations. Therefore, it is advantageous to choose a value of α which is not too far off from its optimum. Alternatively, we may attempt to *adapt* α based on data, which is the approach adopted in this section.

Empirical Bayes. We choose the value of α using the fact that even though this hyperparameter is completely unknown before the deployment phase starts, a better estimate can be made as we observe more data from the deployment phase. If the data matches with how the initial weight is chosen, we may decide to put more trust on $\hat{\boldsymbol{\mu}}$ (large α). On the other hand, we may decide to doubt our initial weight when the observed data does not support it (small α). This strategy invites adoption of *empirical Bayes*, a general method of using observations to estimate or set prior distributions.

Assumptions. In an attempt to do this, we make a hierarchical structure assumption such that $\bar{\boldsymbol{\delta}} \mid \alpha \sim \mathcal{N}(\mathbf{0}, \alpha^{-1}\boldsymbol{I}_d)$, where $\alpha \sim \Gamma(\bar{\alpha}, \bar{\beta})$ for convenience. Furthermore, in order to obtain a well-known distribution, we also assume that $\boldsymbol{\theta}_\star = \hat{\boldsymbol{\mu}} + \bar{\boldsymbol{\delta}}_\star$ as represented by the random variable $\boldsymbol{\theta} = \hat{\boldsymbol{\mu}} + \bar{\boldsymbol{\delta}}$ for deterministic $\hat{\boldsymbol{\mu}}$, where the dissimilarity between $\hat{\boldsymbol{\mu}}$ and $\theta_\star$ is captured by the random variable α embdedded in $\bar{\boldsymbol{\delta}}$. Compared to the initial assumption, α is now treated as random variable and the variance of the initial guess $\boldsymbol{\Sigma}_\mu$ is now absorbed and partially represented by α.

Lemma 6. *With the above assumptions, the marginal $\bar{\boldsymbol{\delta}}$ follows a multivariate student-t distribution with degrees of freedom ν_t, location $\boldsymbol{\mu}_t$ and scale matrix $\boldsymbol{\Sigma}_t$, denoted $St(\nu_t, \mu_t, \boldsymbol{\Sigma}_t)$, with $\nu_t = 2\,\bar{\alpha} = 2\,\bar{\beta}\,\alpha_t, \mu_t = \mathbf{0}, \boldsymbol{\Sigma}_t = \frac{\bar{\beta}}{\bar{\alpha}}\boldsymbol{I}_d = \alpha_t^{-1}\boldsymbol{I}_d$.*

Proof. Firstly, notice that in the case of one-dimensional (scalar) weight, the joint distribution of $(\bar{\delta}, \alpha)$ collapses to normal-gamma distribution. It is a standard result that the marginal distribution of $\bar{\delta}$ follows non-standardised Student-t distribution with degrees of freedom $\nu_t = 2\alpha$, location $\mu_t = \mu$ and scale $\sigma_t^2 = \frac{\beta}{\alpha}$, so we expect a similar result for multidmensional $\bar{\boldsymbol{\delta}}$.

To prove the main result, we compute the required marginal density by marginalising α out of the joint distribution itself found by multiplying the

model's likelihood and prior, noting the integrand of the fifth equation to be the pdf of a gamma distribution with shape $\bar{\alpha}+\frac{d}{2}$ and rate $\bar{\beta}+\frac{1}{2}\,\bar{\boldsymbol{\delta}}^T\,\bar{\boldsymbol{\delta}}$, hence integrates to 1:

$$
\begin{aligned}
p_{\bar{\boldsymbol{\delta}}}(\bar{\boldsymbol{\delta}}) &= \int_0^\infty p_{\bar{\boldsymbol{\delta}}|\alpha}(\bar{\boldsymbol{\delta}} \mid \alpha) p_\alpha(\alpha)\, d\alpha \\
&= \int_0^\infty (2\pi)^{-\frac{d}{2}} \det(\alpha^{-1}\boldsymbol{I}_d)^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}\,\bar{\boldsymbol{\delta}}^T(\alpha^{-1}\boldsymbol{I}_d)^{-1}\,\bar{\boldsymbol{\delta}}\right\} \cdot \\
&\qquad \frac{\bar{\beta}^{\bar{\alpha}}}{\Gamma(\bar{\alpha})}\alpha^{\bar{\alpha}-1}e^{-\bar{\beta}\,\alpha}\, d\alpha \\
&= \int_0^\infty (2\pi)^{-\frac{d}{2}}\alpha^{\frac{d}{2}} \exp\left\{-\frac{\alpha}{2}\,\bar{\boldsymbol{\delta}}^T\,\bar{\boldsymbol{\delta}}\right\} \frac{\bar{\beta}^{\bar{\alpha}}}{\Gamma(\bar{\alpha})}\alpha^{\bar{\alpha}-1} \exp\{-\bar{\beta}\,\alpha\}\, d\alpha \\
&= \frac{(2\pi)^{-\frac{d}{2}}\,\bar{\beta}^{\bar{\alpha}}}{\Gamma(\bar{\alpha})} \int_0^\infty \alpha^{\bar{\alpha}+\frac{d}{2}-1} \exp\left\{-\alpha(\bar{\beta}+\frac{1}{2}\,\bar{\boldsymbol{\delta}}^T\,\bar{\boldsymbol{\delta}})\right\} d\alpha \\
&= \frac{(2\pi)^{-\frac{d}{2}}\,\bar{\beta}^{\bar{\alpha}}}{\Gamma(\bar{\alpha})} \frac{\Gamma(\bar{\alpha}+\frac{d}{2})}{(\bar{\beta}+\frac{1}{2}\,\bar{\boldsymbol{\delta}}^T\,\bar{\boldsymbol{\delta}})^{\bar{\alpha}+\frac{d}{2}}} \cdot \\
&\qquad \int_0^\infty \frac{(\bar{\beta}+\frac{1}{2}\,\bar{\boldsymbol{\delta}}^T\,\bar{\boldsymbol{\delta}})^{\bar{\alpha}+\frac{d}{2}}}{\Gamma(\bar{\alpha}+\frac{d}{2})}\alpha^{(\bar{\alpha}+\frac{d}{2})-1} \exp^{-(\bar{\beta}+\frac{1}{2}\,\bar{\boldsymbol{\delta}}^T\,\bar{\boldsymbol{\delta}})\alpha}\, d\alpha \\
&= \frac{(2\pi)^{-\frac{d}{2}}\,\bar{\beta}^{\bar{\alpha}}\,\Gamma(\bar{\alpha}+\frac{d}{2})}{(\bar{\beta}+\frac{1}{2}\,\bar{\boldsymbol{\delta}}^T\,\bar{\boldsymbol{\delta}})^{\bar{\alpha}+\frac{d}{2}}\Gamma(\bar{\alpha})} \\
&= \frac{1}{2^{\frac{d}{2}}\pi^{\frac{d}{2}}}\,\bar{\beta}^{\bar{\alpha}}\,\frac{\Gamma(\frac{2\,\bar{\alpha}+d}{2})}{\Gamma(\frac{2\,\bar{\alpha}}{2})}(\bar{\beta}+\frac{1}{2}\,\bar{\boldsymbol{\delta}}^T\,\bar{\boldsymbol{\delta}})^{-\frac{2\,\bar{\alpha}+d}{2}} \\
&= \frac{1}{2^{\frac{d}{2}}\pi^{\frac{d}{2}}}\,\bar{\beta}^{\bar{\alpha}}\,\frac{\Gamma(\frac{2\,\bar{\alpha}+d}{2})}{\Gamma(\frac{2\,\bar{\alpha}}{2})}\,\bar{\beta}^{-\frac{2\,\bar{\alpha}+d}{2}}\left(1+\frac{1}{2}\,\bar{\boldsymbol{\delta}}^T\,\frac{1}{\bar{\beta}}\boldsymbol{I}_d\,\bar{\boldsymbol{\delta}}\right)^{-\frac{2\,\bar{\alpha}+d}{2}} \\
&= \frac{1}{2^{\frac{d}{2}}\pi^{\frac{d}{2}}}\,\frac{\Gamma(\frac{2\,\bar{\alpha}+d}{2})}{\Gamma(\frac{2\,\bar{\alpha}}{2})}\,\bar{\beta}^{-\frac{d}{2}}\left(1+\frac{1}{2\,\bar{\alpha}}\,\bar{\boldsymbol{\delta}}^T\,\frac{\bar{\alpha}}{\bar{\beta}}\boldsymbol{I}_d\,\bar{\boldsymbol{\delta}}\right)^{-\frac{2\,\bar{\alpha}+d}{2}} \\
&= \frac{1}{2^{\frac{d}{2}}\pi^{\frac{d}{2}}}\,\frac{\bar{\alpha}^{\frac{d}{2}}}{\bar{\alpha}^{\frac{d}{2}}}\,\frac{\Gamma(\frac{2\,\bar{\alpha}+d}{2})}{\Gamma(\frac{2\,\bar{\alpha}}{2})}\,\bar{\beta}^{-\frac{d}{2}}\left(1+\frac{1}{2\,\bar{\alpha}}\,\bar{\boldsymbol{\delta}}^T\left(\frac{\bar{\beta}}{\bar{\alpha}}\boldsymbol{I}_d\right)^{-1}\bar{\boldsymbol{\delta}}\right)^{-\frac{2\,\bar{\alpha}+d}{2}} \\
&= \frac{1}{(2\,\bar{\alpha})^{\frac{d}{2}}\pi^{\frac{d}{2}}}\left(\frac{\bar{\alpha}}{\bar{\beta}}\right)^{\frac{d}{2}}\,\frac{\Gamma(\frac{2\,\bar{\alpha}+d}{2})}{\Gamma(\frac{2\,\bar{\alpha}}{2})}\left[1+\frac{1}{2\,\bar{\alpha}}\,\bar{\boldsymbol{\delta}}^T\left(\frac{\bar{\beta}}{\bar{\alpha}}\boldsymbol{I}_d\right)^{-1}\bar{\boldsymbol{\delta}}\right]^{-\frac{2\,\bar{\alpha}+d}{2}} \\
&= \frac{\Gamma(\frac{2\,\bar{\alpha}+d}{2})}{\Gamma(\frac{2\,\bar{\alpha}}{2})(2\,\bar{\alpha})^{\frac{d}{2}}\pi^{\frac{d}{2}}\left[\left(\frac{\bar{\beta}}{\bar{\alpha}}\right)^d\right]^{\frac{1}{2}}}\left[1+\frac{1}{2\,\bar{\alpha}}\,\bar{\boldsymbol{\delta}}^T\left(\frac{\bar{\beta}}{\bar{\alpha}}\boldsymbol{I}_d\right)^{-1}\bar{\boldsymbol{\delta}}\right]^{-\frac{2\,\bar{\alpha}+d}{2}}
\end{aligned}
$$

$$= \frac{\Gamma(\frac{2\,\bar{\alpha}+d}{2})}{\Gamma(\frac{2\,\bar{\alpha}}{2})(2\,\bar{\alpha})^{\frac{d}{2}}\pi^{\frac{d}{2}}\left[\det\left(\frac{\bar{\beta}}{\bar{\alpha}}\boldsymbol{I}_d\right)\right]^{\frac{1}{2}}} \cdot$$

$$\left[1+\frac{1}{2\,\bar{\alpha}}(\bar{\boldsymbol{\delta}}-\mathbf{0})^T\left(\frac{\bar{\beta}}{\bar{\alpha}}\boldsymbol{I}_d\right)^{-1}(\bar{\boldsymbol{\delta}}-\mathbf{0})\right]^{-\frac{2\,\bar{\alpha}+d}{2}}$$

$$= \frac{\Gamma(\frac{\nu_t+d}{2})}{\Gamma(\frac{\nu_t}{2})\nu_t^{\frac{d}{2}}\pi^{\frac{d}{2}}\left(\det\boldsymbol{\Sigma_t}\right)^{\frac{1}{2}}}\left[1+\frac{1}{\nu_t}(\bar{\boldsymbol{\delta}}-\boldsymbol{\mu_t})^T\boldsymbol{\Sigma_t}^{-1}(\bar{\boldsymbol{\delta}}-\boldsymbol{\mu_t})\right]^{-\frac{\nu_t+d}{2}},$$

which is multivariate t-distribution with $\nu_t = 2\,\bar{\alpha}, \mu_t = \mathbf{0}$ and $\boldsymbol{\Sigma}_t = \alpha_t^{-1}\boldsymbol{I}_d = \frac{\bar{\beta}}{\bar{\alpha}}\boldsymbol{I}_d$. Therefore, we conclude that $\bar{\boldsymbol{\delta}} \sim St(2\,\bar{\alpha}, \mathbf{0}, \frac{\bar{\beta}}{\bar{\alpha}}\boldsymbol{I}_d)$, *i.e.*, a student-t distribution with zero mean and spherical covariance. Notice that we can express ν_t in terms of α_t and $\bar{\beta}$ as $\nu_t = 2\,\bar{\beta}\,\alpha_t$ since $\alpha_t = \frac{\bar{\alpha}}{\bar{\beta}}$. By setting the hyperparameters in terms of $(\alpha_t, \bar{\beta})$, we control the prior of α by its mean α_t and variance $\frac{\alpha_t}{\bar{\beta}}$, which is more intuitive instead of its shape and rate $(\bar{\alpha}, \bar{\beta})$. □

Following Song and Xia (2016), we adopt noise such that

$$\boldsymbol{\epsilon} \sim St\left(2\,\bar{\alpha}+d, \mathbf{0}, \frac{2\,\bar{\alpha}}{2\,\bar{\alpha}+d}\left(1+\frac{1}{2\,\bar{\beta}}\|\,\bar{\boldsymbol{\delta}}\,\|_2^2\right)\beta_t^{-1}\boldsymbol{I}_n\right).$$

Adaptive Hyperparameter Algorithm. Since $\bar{\boldsymbol{\delta}}$ follows a student-t distribution, our assumptions follow the premise laid out by Song and Xia (2016). By rewriting $\boldsymbol{X} = [\boldsymbol{x}_1 \quad \cdots \quad \boldsymbol{x}_n]^T$ and $\boldsymbol{y} = [y_1 \quad \cdots \quad y_n]^T$, the value of α_t and β_t can then be optimised by the q-EM algorithm following Song and Xia (2016), summarised in Algorithm 5. This algorithm takes $\bar{\beta}$ as its hyperparameter, which controls the degrees of freedom in the underlying distribution of $\bar{\boldsymbol{\delta}}$: when a Gaussian distribution of $\bar{\boldsymbol{\delta}}$ is preferred, we let $\nu_t \to \infty$ by letting $\bar{\beta} \to \infty$, recovering the Gaussian distribution from the t-distribution.

Some steps in Algorithm 5 require expensive computations. To mitigate such costs, Song and Xia (2016) suggest to diagonalise the Gram matrix $\boldsymbol{X}^T\boldsymbol{X} = \boldsymbol{P}\boldsymbol{D}\boldsymbol{P}^T$ and compute the following quantities beforehand:

$$\boldsymbol{y}_p = \boldsymbol{X}^T\boldsymbol{y}, \quad \boldsymbol{y}_{pV} = \boldsymbol{P}^T\boldsymbol{y}_p, \quad \|\boldsymbol{y}\|_2^2 .$$

The required quantities in each iteration can then be calculated as:

$$\begin{aligned}
\boldsymbol{\mu}_{opt} &= \boldsymbol{P}\left(\boldsymbol{D}+\frac{\alpha_t}{\beta_t}\boldsymbol{I}_d\right)^{-1}\boldsymbol{y}_{pV} \\
\boldsymbol{y}^T\boldsymbol{B}_{opt}^{-1}\boldsymbol{y} &= \beta_t\left(\|\boldsymbol{y}\|_2^2 - \boldsymbol{y}_p^T\boldsymbol{\mu_{opt}}\right) \\
\mathrm{tr}(\boldsymbol{C}_{opt}) &= \frac{\nu+\boldsymbol{y}^T\boldsymbol{B}_{opt}^{-1}\boldsymbol{y}}{\nu+n}\,\mathrm{tr}((\alpha_t\boldsymbol{I}_d+\beta_t\boldsymbol{D})^{-1}) \\
\mathrm{tr}(\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{C}_{opt}) &= \frac{\nu+\boldsymbol{y}^T\boldsymbol{B}_{opt}^{-1}\boldsymbol{y}}{\nu+n}\,\mathrm{tr}(\boldsymbol{D}(\alpha_t\boldsymbol{I}_d+\beta_t\boldsymbol{D})^{-1}) \\
\|\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\mu}_{opt}\|_2^2 &= \|\boldsymbol{y}\|_2^2 - 2\boldsymbol{y}_p^T\boldsymbol{\mu}_{opt} + \|\boldsymbol{X}\boldsymbol{\mu}_{opt}\|_2^2 ,
\end{aligned}$$

where the Woodbury matrix identity is used in the second equation and the cyclic property of the trace operation is used in the fourth equation. We argue

Algorithm 5 Adaptive Optimisation of α_t and β_t

Input: $\boldsymbol{X}, \boldsymbol{y}, \bar{\beta}, \alpha_t, \beta_t, tol$
$\alpha_{t,old} \leftarrow \alpha_t, \beta_{t,old}, \leftarrow \beta_t$
while $\frac{|\alpha_t - \alpha_{t,old}|}{\alpha_t} > tol$ or $\frac{|\beta_t - \beta_{t,old}|}{\beta_t} > tol$ **do**
  $\nu \leftarrow 2\,\bar{\beta}\,\alpha_t$
  $\alpha_{t,old} \leftarrow \alpha_t, \beta_{t,old} \leftarrow \beta_t$
  $\boldsymbol{A_{opt}} \leftarrow (\alpha_t \boldsymbol{I}_d + \beta_t \boldsymbol{X}^T \boldsymbol{X})^{-1}$
  $\boldsymbol{\mu}_{opt} \leftarrow (\boldsymbol{X}^T \boldsymbol{X} + \frac{\alpha_t}{\beta_t} \boldsymbol{I}_d)^{-1} \boldsymbol{X}^T \boldsymbol{y}$
  $\boldsymbol{B}_{opt} \leftarrow \beta_t^{-1} \boldsymbol{I}_n + \alpha_t^{-1} \boldsymbol{X} \boldsymbol{X}^T$
  $\boldsymbol{C}_{opt} \leftarrow \frac{1}{\nu+n}(\nu + \boldsymbol{y}^T \boldsymbol{B}_{opt}^{-1} \boldsymbol{y}) \boldsymbol{A}_{opt}$
  $b_{opt} \leftarrow \|\boldsymbol{\mu}_{opt}\|_2^2 + \mathrm{tr}(\boldsymbol{C}_{opt})$
  $c_{opt} \leftarrow \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\mu}_{opt}\|_2^2 + \mathrm{tr}(\boldsymbol{X}^T \boldsymbol{X} \boldsymbol{C}_{opt})$
  $\alpha_t \leftarrow \frac{d}{b_{opt}}$
  $\beta_t \leftarrow \frac{n}{c_{opt}}$
end while
return α_t, β_t

that when one wishes to store $\boldsymbol{X}^T \boldsymbol{X}$ and not $\boldsymbol{X}$, then the second term of the last equality can be calculated as

$$\|\boldsymbol{X}\boldsymbol{\mu}_{opt}\|_2^2 = \boldsymbol{\mu}_{opt}^T (\boldsymbol{X}^T \boldsymbol{X}) \boldsymbol{\mu}_{opt} = \|\boldsymbol{\mu}_{opt}\|_{\boldsymbol{X}^T \boldsymbol{X}}^2 \,.$$

These quantities can then be used to calculate b_{opt} and c_{opt} which yield new α_t and β_t until convergence.

Regret Bound. Algorithm 5 may be invoked at the start of each round to give updated values of α_t and β_t. However, under this adaptive hyperparameter, α^{-1} is no longer independent of the other variables. This violates one of the assumptions made in (Abbasi-Yadkori et al., 2011), as the choice of λ in their scenario is independent of other variables. Therefore, the validity of the over-sampling factor becomes questionable. As the regret analysis for LinTS depends on the validity of the upper bound provided by (Abbasi-Yadkori et al., 2011), this in turns becomes invalid as well. As such, regret analysis for the adaptive case would become another open problem. A possible remedy for this problem may be to halt the hyperparameter optimisation update after a certain number of rounds, in which case α^{-1} might be viewed as constant in the long run as a direct consequence of Theorem 2 and 3.

Corollary 7 (Warm-Start Bandit with Adaptive Hyperparameters)**.** *Consider a multi-armed bandit agent with hyperparameters updated as per Algorithm 5 every round up to round n_s when no further update is invoked. Then, round $n_s + 1$ can be treated as the first bandit round with constant hyperparameter.*
For LinTS, this is equivalent to

$$Reg(T + n_s) = R^{TS}(T + n_s) + R^{RLS}(T + n_s)\,,$$

with each of the term bounded as

$$R^{TS}(T+n_s) \leq R^{TS}(n_s) + \frac{4\bar{\gamma}_T(\delta')}{p}\left(\sqrt{2T\log\frac{\det(\boldsymbol{V}_{n_s+T+1})}{\det(R^2\boldsymbol{V}_{n_s+1})}} + \sqrt{\frac{8T}{\lambda_{min}(R^2\boldsymbol{V}_{n_s+1})}\log\frac{4}{\delta}}\right)$$

$$R^{RLS}(T+n_s) \leq R^{RLS}(n_s) + \left(\bar{\beta}_T(\delta') + \bar{\gamma}_T(\delta')\right)\sqrt{2T\log\frac{\det(\boldsymbol{V}_{n_s+T+1})}{\det(R^2\boldsymbol{V}_{n_s+1})}}\ ,$$

where $R^{RS}(n_s)$ *and* $R^{RLS}(n_s)$ *are constant,* $\bar{\gamma}_T(\delta) = \bar{\beta}_T(\delta')\sqrt{cd\log((c'd)/\delta)}$ *and* $\bar{\beta}_T(\delta)$ *is the upper bound of the ellipsoid whose rounds start at* n_s*, defined as:*

$$\bar{\beta}_T(\delta) = R\sqrt{2\log\left(\frac{\det(\boldsymbol{V}_{n_s+T})^{\frac{1}{2}}\det(R^2\boldsymbol{V}_{n_s+1})^{-\frac{1}{2}}}{\delta}\right)} + \sqrt{\lambda_{max}(R^2\boldsymbol{V}_{n_s+1})}\bar{S}\ ,$$

where $\bar{S}$ *is defined such that* $\|\hat{\boldsymbol{\theta}}_{n_s+1} - \boldsymbol{\theta}_\star\| \leq \bar{S}$*.*
For LinUCB, this is equivalent to

$$Reg(T+n_s) \leq Reg(n_s) + \bar{\beta}_T(\delta)\sqrt{8T\log\left(\frac{\det(\boldsymbol{V}_{n_s+T+1})}{\det(R^2\boldsymbol{V}_{n_s+1})}\right)}\ ,$$

where $Reg(n_s)$ *is constant and* $\bar{\beta}_T(\delta)$ *is defined as above.*

Experimental Results. To demonstrate the advantage of the adaptive hyperparameter tuning, we repeated the experiment for the artificial dataset. We generated two types of pre-training data: accurate and misspecified. For the generation of accurate dataset, we chose true $\alpha^{-1} = 10^{-4}$ and for the misspecified dataset, we chose true $\alpha^{-1} = 100$. Notice that such a high value of α in the misspecified dataset is intentionally chosen to be extreme to demonstrate the capability of the adaptive hyperparameter algorithm, and hence does not reflect a real world setting. For the bandit, we have used LinUCB with $\rho = 0.2$, bandit hyperparameters initial $\alpha_t = 1$, initial $\beta_t = 1/R^2 = 16$ (both unchanged over time for bandits with manually chosen hyperparameters), and $\bar{\beta} = 1$ with $tol = 0.1$ for hyperparameter tuning convergence requirements of both α_t and β_t. As shown in Figure 9 (a), the adaptive hyperparameter algorithm is capable of exploiting the accurate prior, even outperforming its non-adaptive counterpart. On the other hand, when the prior is highly misspecified in Figure 9 (b), a disastrous result occurs for warm-started bandit without automatic hyperparameter, while our adaptive hyperparameter algorithm is able to detect the mismatch and ignore the initial guess, attempting to restore its performance should cold-start regime had been deployed.

6. Conclusions and Future Work

In this paper we have developed a flexible framework for warm starting linear contextual bandits that inherits the flexibility of Bayesian inference in incorporating prior knowledge. Our approach generalises the Linear Thompson

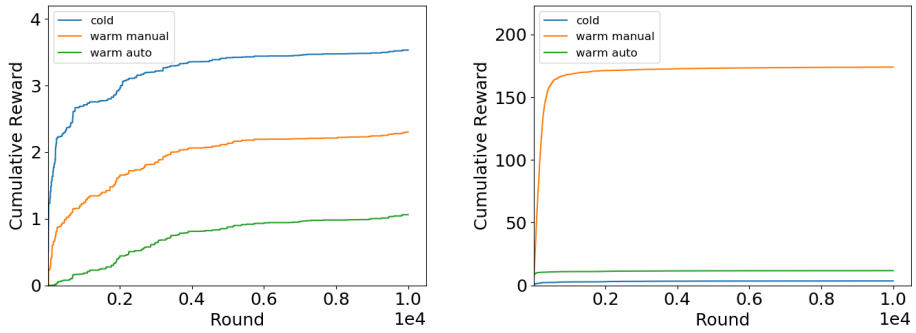


Fig. 9. Experimental results for (a) an accurate prior and (b) a misspecified prior, comparing cold-start (cold), warm-start with non-adaptive hyperparameters (warm manual) and warm-start with adaptive hyperparameters (warm auto) using LinUCB.

Sampler of Abeille et al. (2017), by permitting arbitrary Gaussian priors for potentially improving short-term performance, while maintaining the regret bound that guarantees the long-term performance of Hannan consistency. While little attention has been paid to the warm start problem since the direction was suggested by Li et al. (2010), the few existing works on warm start are far less flexible in catering to potential sources of prior knowledge, and in how uncertainty is quantified. We motivate the opportunity for warm start in the database systems domain where bandit-based index selection could be pre-trained prior to deployment by users, and we demonstrate the practical potential for warm start on a standard database benchmark. We have also contributed an approach to adapting the key hyperparameters responsible to the control of the exploration temperature based on misspecification of pre-training.

Being relatively unexplored, we believe that warm start bandits offer a number of intriguing future directions for research, well suited to the Thompson Sampling framework on which our approach was developed.

Adaptive Oversampling Factor. In this paper, it is assumed that the ℓ_2-norm of the parameter is bounded by S. However, this may not be known with confidence in some applications. In such cases the algorithms are still valid, but the bounds may not be. However, as more data is observed, we gain information (accuracy) about $\boldsymbol{\delta}_\star$: the variance of random variable $\boldsymbol{\delta}$ drops. Therefore, one may wish to bound $\|\boldsymbol{\delta}\|$ with some level of probability. It is interesting to note that how large the value of S is closely related on the drift hyperparameter—potentially both quantities could be optimised using one algorithm jointly.

Reward Unit Mismatch. When the pre-training data is provided, there is a potential difference between the units of the pre-training and deployed datasets. An interesting problem arises by noticing that the performance of the contextual bandit algorithm is not measured by how close the predicted reward is to the actual reward, but rather the *rank* of the arm values. As such it is the direction of the initial guess of $\boldsymbol{\theta}$ that is important, not its norm. A simple solution could be learning a constant scaling the size of the pre-training reward to the deployed

rewards. Ideally this scalar would be incorporated into the Warm Start LinTS, provided performance is not sacrificed.

A. Full Proof of the Confidence Ellipsoid of Warm-Started Bandit

We now detail the full proof of Theorem 2, by extending a previous analysis (Abbasi-Yadkori et al., 2011). We restate our estimate of the parameter for convenience:

$$\hat{\boldsymbol{\theta}}_n = \boldsymbol{V}_n^{-1}\boldsymbol{b}_n \,,$$

where for $n \geq 2$ we have defined

$$\boldsymbol{V}_n = \bar{\boldsymbol{V}}_1 + \sum_{i=1}^{n-1} \boldsymbol{x}_i \boldsymbol{x}_i^T, \quad \boldsymbol{b}_n = \sum_{i=1}^{n-1} y_i \boldsymbol{x}_i \,.$$

Let $\boldsymbol{X}_{1:t}$ and $\boldsymbol{Y}_{1:t}$ be matrices comprising the contexts and the rewards up to round t respectively and $\boldsymbol{\epsilon}_{1:t}$ be the vector containing their corresponding subgaussian noise, that is:

$$\boldsymbol{X}_{1:t} = \begin{bmatrix} \boldsymbol{x}_1^T \\ \vdots \\ \boldsymbol{x}_t^T \end{bmatrix}, \quad \boldsymbol{Y}_{1:t} = \begin{bmatrix} y_1 \\ \vdots \\ y_t \end{bmatrix}, \quad \boldsymbol{\epsilon}_{1:t} = \begin{bmatrix} \epsilon_1 \\ \vdots \\ \epsilon_t \end{bmatrix} \,.$$

Therefore, we can write $\hat{\boldsymbol{\theta}}_t$ as

$$\hat{\boldsymbol{\theta}}_t = (\boldsymbol{X}_{1:t-1}^T \boldsymbol{X}_{1:t-1} + \bar{\boldsymbol{V}}_1)^{-1} (\boldsymbol{X}_{1:t-1}^T \boldsymbol{Y}_{1:t-1}) \,.$$

To avoid clutter, let $\boldsymbol{X} = \boldsymbol{X}_{1:t-1}, \boldsymbol{Y} = \boldsymbol{Y}_{1:t-1}, \boldsymbol{\epsilon} = \boldsymbol{\epsilon}_{1:t-1}$. Then, we have $\boldsymbol{V}_t = \bar{\boldsymbol{V}}_1 + \boldsymbol{X}^T\boldsymbol{X}$. Therefore, we can expand the expression of $\boldsymbol{\theta}_t$ above as:

$$\begin{aligned}
\hat{\boldsymbol{\theta}}_t &= (\boldsymbol{X}^T\boldsymbol{X} + \bar{\boldsymbol{V}}_1)^{-1}(\boldsymbol{X}^T\boldsymbol{Y}) \\
&= (\boldsymbol{X}^T\boldsymbol{X} + \bar{\boldsymbol{V}}_1)^{-1}[\boldsymbol{X}^T(\boldsymbol{X}\boldsymbol{\theta}_\star + \boldsymbol{\epsilon})] \\
&= (\boldsymbol{X}^T\boldsymbol{X} + \bar{\boldsymbol{V}}_1)^{-1}\boldsymbol{X}^T\boldsymbol{\epsilon} + (\boldsymbol{X}^T\boldsymbol{X} + \bar{\boldsymbol{V}}_1)^{-1}\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{\theta}_\star \\
&= (\boldsymbol{X}^T\boldsymbol{X} + \bar{\boldsymbol{V}}_1)^{-1}\boldsymbol{X}^T\boldsymbol{\epsilon} + (\boldsymbol{X}^T\boldsymbol{X} + \bar{\boldsymbol{V}}_1)^{-1}(\boldsymbol{X}^T\boldsymbol{X} + \bar{\boldsymbol{V}}_1 - \bar{\boldsymbol{V}}_1)\boldsymbol{\theta}_\star \\
&= (\boldsymbol{X}^T\boldsymbol{X} + \bar{\boldsymbol{V}}_1)^{-1}\boldsymbol{X}^T\boldsymbol{\epsilon} + (\boldsymbol{X}^T\boldsymbol{X} + \bar{\boldsymbol{V}}_1)^{-1}(\boldsymbol{X}^T\boldsymbol{X} + \bar{\boldsymbol{V}}_1)\boldsymbol{\theta}_\star - \\
&\qquad (\boldsymbol{X}^T\boldsymbol{X} + \bar{\boldsymbol{V}}_1)^{-1}\bar{\boldsymbol{V}}_1\boldsymbol{\theta}_\star \\
&= (\boldsymbol{X}^T\boldsymbol{X} + \bar{\boldsymbol{V}}_1)^{-1}\boldsymbol{X}^T\boldsymbol{\epsilon} + \boldsymbol{\theta}_\star - (\boldsymbol{X}^T\boldsymbol{X} + \bar{\boldsymbol{V}}_1)^{-1}\bar{\boldsymbol{V}}_1\boldsymbol{\theta}_\star \,.
\end{aligned}$$

Next, we would like to obtain for any vector with appropriate size $\boldsymbol{c}$:

$$\begin{aligned}
&\boldsymbol{c}^T\hat{\boldsymbol{\theta}}_t - \boldsymbol{c}^T\boldsymbol{\theta}_\star \\
&\quad = \boldsymbol{c}^T(\boldsymbol{X}^T\boldsymbol{X} + \bar{\boldsymbol{V}}_1)^{-1}\boldsymbol{X}^T\boldsymbol{\epsilon} - \boldsymbol{c}^T(\boldsymbol{X}^T\boldsymbol{X} + \bar{\boldsymbol{V}}_1)^{-1}\bar{\boldsymbol{V}}_1\boldsymbol{\theta}_\star \\
&\quad = \langle \boldsymbol{c}, \boldsymbol{X}^T\boldsymbol{\epsilon} \rangle_{\boldsymbol{V}_t^{-1}} - \langle \boldsymbol{c}, \bar{\boldsymbol{V}}_1\boldsymbol{\theta}_\star \rangle_{\boldsymbol{V}_t^{-1}} \,.
\end{aligned}$$

Now as we have assumed that $\bar{\boldsymbol{V}}_1$ is positive definite, and since $\boldsymbol{V}_t$ is the sum of positive definite matrices, then $\boldsymbol{V}_t$ is also a positive definite matrix, thus the

inner products are well-defined. Therefore, we can invoke the Cauchy-Schwarz Inequality to obtain

$$\begin{aligned}|\boldsymbol{c}^T\hat{\boldsymbol{\theta}}_t - \boldsymbol{c}^T\boldsymbol{\theta}_\star| &\leq \|\boldsymbol{c}\|_{\boldsymbol{V}_t^{-1}}\|\boldsymbol{X}^T\boldsymbol{\epsilon}\|_{\boldsymbol{V}_t^{-1}} + \|\boldsymbol{c}\|_{\boldsymbol{V}_t^{-1}}\|\bar{\boldsymbol{V}}_1\boldsymbol{\theta}_\star\|_{\boldsymbol{V}_t^{-1}} \\ &= \|\boldsymbol{c}\|_{\boldsymbol{V}_t^{-1}}\left(\|\boldsymbol{X}^T\boldsymbol{\epsilon}\|_{\boldsymbol{V}_t^{-1}} + \|\bar{\boldsymbol{V}}_1\boldsymbol{\theta}_\star\|_{\boldsymbol{V}_t^{-1}}\right) \ .\end{aligned}$$

Now (Abbasi-Yadkori et al., 2011, Theorem 1), where $\boldsymbol{V} = \bar{\boldsymbol{V}}_1$, yields, with probability at least $1-\delta$ that

$$\|\boldsymbol{X}^T\boldsymbol{\epsilon}\|_{\boldsymbol{V}_t^{-1}} \leq R\sqrt{2\log\left(\frac{\det(\boldsymbol{V}_t)^{\frac{1}{2}}\det(\bar{\boldsymbol{V}}_1)^{\frac{1}{2}}}{\delta}\right)} \ .$$

Furthermore, since $\boldsymbol{c}$ can be any vector, we choose $\boldsymbol{c} = \boldsymbol{V}_t(\hat{\boldsymbol{\theta}}_t - \boldsymbol{\theta}_\star)$, which yields

$$\begin{aligned}\boldsymbol{c}^T\hat{\boldsymbol{\theta}}_t - \boldsymbol{c}^T\boldsymbol{\theta}_\star &= \boldsymbol{c}^T(\hat{\boldsymbol{\theta}}_t - \boldsymbol{\theta}_\star) \\ &= (\hat{\boldsymbol{\theta}}_t - \boldsymbol{\theta}_\star)^T\boldsymbol{V}_t(\hat{\boldsymbol{\theta}}_t - \boldsymbol{\theta}_\star) \\ &= \|\hat{\boldsymbol{\theta}}_t - \boldsymbol{\theta}_\star\|^2_{\boldsymbol{V}_t} \ ,\end{aligned}$$

and

$$\begin{aligned}\|\boldsymbol{c}\|_{\boldsymbol{V}_t^{-1}} &= \|\boldsymbol{V}_t(\hat{\boldsymbol{\theta}}_t - \boldsymbol{\theta}_\star)\|_{\boldsymbol{V}_t^{-1}} \\ &= \sqrt{(\hat{\boldsymbol{\theta}}_t - \boldsymbol{\theta}_\star)^T\boldsymbol{V}_t^T\boldsymbol{V}_t^{-1}\boldsymbol{V}_t(\hat{\boldsymbol{\theta}}_t - \boldsymbol{\theta}_\star)} \\ &= \|\hat{\boldsymbol{\theta}}_t - \boldsymbol{\theta}_\star\|_{\boldsymbol{V}_t} \ .\end{aligned}$$

Combining both expressions above, we have:

$$\|\hat{\boldsymbol{\theta}}_t - \boldsymbol{\theta}_\star\|_{\boldsymbol{V}_t} \leq \|\bar{\boldsymbol{V}}_1\boldsymbol{\theta}_\star\|_{\boldsymbol{V}_t^{-1}} + R\sqrt{2\log\left(\frac{\det(\boldsymbol{V}_t)^{\frac{1}{2}}\det(\bar{\boldsymbol{V}}_1)^{\frac{1}{2}}}{\delta}\right)} \ .$$

Now we use the fact that $\boldsymbol{V}_s \leq \boldsymbol{V}_t$ for $s \leq t$, thus we can bound:

$$\begin{aligned}\|\bar{\boldsymbol{V}}_1\boldsymbol{\theta}_\star\|_{\boldsymbol{V}_t^{-1}} &= \sqrt{\boldsymbol{\theta}_\star\bar{\boldsymbol{V}}_1^T\boldsymbol{V}_t^{-1}\bar{\boldsymbol{V}}_1\boldsymbol{\theta}_\star} \\ &\leq \sqrt{\boldsymbol{\theta}_\star\bar{\boldsymbol{V}}_1^T\bar{\boldsymbol{V}}_1^{-1}\bar{\boldsymbol{V}}_1\boldsymbol{\theta}_\star} \\ &= \|\boldsymbol{\theta}_\star\|_{\bar{\boldsymbol{V}}_1} \\ &\leq \sqrt{\lambda_{max}(\bar{\boldsymbol{V}}_1)}\|\boldsymbol{\theta}_\star\| \\ &\leq \sqrt{\lambda_{max}(\bar{\boldsymbol{V}}_1)}S \ .\end{aligned}$$

Thus, we conclude that

$$\|\hat{\boldsymbol{\theta}}_t - \boldsymbol{\theta}_\star\|_{\boldsymbol{V}_t} \leq R\sqrt{2\log\left(\frac{\det(\boldsymbol{V}_t)^{\frac{1}{2}}\det(\bar{\boldsymbol{V}}_1)^{\frac{1}{2}}}{\delta}\right)} + \sqrt{\lambda_{max}(\bar{\boldsymbol{V}}_1)}S \ .$$

B. Full Proof of the Regret Bound of Warm-Start LinUCB

The regret analysis for LinUCB is included here for completeness, and follows closely the proof laid out by Lattimore and Szepesvári (2020). Let $\mathcal{C}_t$ be a closed confidence set containing $\boldsymbol{\theta}_\star$ with high probability such that $\mathcal{C}_t \subseteq \{\boldsymbol{\theta} \in \mathbb{R}^d : \|\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}_t\|_{\boldsymbol{V}_t} \leq \beta_t\}$. Furthermore, let $\tilde{\boldsymbol{\theta}}_t \in \mathcal{C}_t$ be such that $\tilde{\boldsymbol{\theta}}_t^T \boldsymbol{x}_t = UCB_t(\boldsymbol{x}_t)$. This implies that

$$\boldsymbol{\theta}_\star^T \boldsymbol{x}_t^\star \leq UCB_t(\boldsymbol{x}_t^\star) \leq UCB_t(\boldsymbol{x}_t) = \tilde{\boldsymbol{\theta}}_t^T \boldsymbol{x}_t \ .$$

Therefore,

$$\begin{aligned}
reg_t &= \boldsymbol{\theta}_\star^T \boldsymbol{x}_t^\star - \boldsymbol{\theta}_\star^T \boldsymbol{x}_t \\
&\leq \tilde{\boldsymbol{\theta}}_t^T \boldsymbol{x}_t - \boldsymbol{\theta}_\star^T \boldsymbol{x}_t \\
&= (\tilde{\boldsymbol{\theta}}_t - \boldsymbol{\theta}_\star)^T \boldsymbol{x}_t \\
&\leq \|\boldsymbol{x}_t\|_{\boldsymbol{V}_t^{-1}} \|\tilde{\boldsymbol{\theta}}_t - \boldsymbol{\theta}_\star\|_{\boldsymbol{V}_t} \\
&= \|\boldsymbol{x}_t\|_{\boldsymbol{V}_t^{-1}} \|(\tilde{\boldsymbol{\theta}}_t - \hat{\boldsymbol{\mu}}) - (\boldsymbol{\theta}_\star - \hat{\boldsymbol{\mu}})\|_{\boldsymbol{V}_t} \\
&= \|\boldsymbol{x}_t\|_{\boldsymbol{V}_t^{-1}} \|\tilde{\boldsymbol{\delta}}_t - \boldsymbol{\delta}_\star\|_{\boldsymbol{V}_t} \\
&\leq 2\|\boldsymbol{x}_t\|_{\boldsymbol{V}_t^{-1}} \beta_t
\end{aligned}$$

where we have defined $\tilde{\boldsymbol{\delta}}_t$ as the vector $\tilde{\boldsymbol{\theta}}_t$ relative to $\hat{\boldsymbol{\mu}}$. The next step follows from Jensen's Inequality:

$$\begin{aligned}
Reg(T) &= \sum_{t=1}^{T} reg_t \\
&\leq \sqrt{T \sum_{t=1}^{T} reg_t^2} \\
&\leq \sqrt{T \sum_{t=1}^{T} 4\|\boldsymbol{x}_t\|^2_{\boldsymbol{V}_t^{-1}} \beta_t^2} \\
&\leq \beta_T \sqrt{4T \sum_{t=1}^{T} \|\boldsymbol{x}_t\|^2_{\boldsymbol{V}_t^{-1}}} \\
&\leq \beta_T \sqrt{8T \log\left(\frac{\det(\boldsymbol{V}_{T+1})}{\det(R^2 \boldsymbol{V}_1)}\right)} \ .
\end{aligned}$$

where

$$\beta_T = R\sqrt{2\log\left(\frac{\det(\boldsymbol{V}_T)^{\frac{1}{2}} \det(R^2\boldsymbol{V}_1)^{-\frac{1}{2}}}{\delta}\right)} + \sqrt{\lambda_{max}(R^2\boldsymbol{V}_1)}S \ .$$

References

Abbasi-Yadkori, Y., Pál, D. and Szepesvári, C. (2011), 'Improved algorithms for linear stochastic bandits', *Advances in Neural Information Processing Systems* **24**, 2312–2320.

Abeille, M., Lazaric, A. et al. (2017), 'Linear Thompson sampling revisited', *Electronic Journal of Statistics* **11**(2), 5165–5197.

Agrawal, S., Chaudhuri, S., Kollár, L., Marathe, A. P., Narasayya, V. R. and Syamala, M. (2004), Database tuning advisor for Microsoft SQL Server 2005, *in* 'VLDB'.

Agrawal, S. and Goyal, N. (2013), Thompson sampling for contextual bandits with linear payoffs, *in* 'International Conference on Machine Learning', pp. 127–135.

Auer, P., Cesa-Bianchi, N. and Fischer, P. (2002), 'Finite-time analysis of the multiarmed bandit problem', *Machine learning* **47**(2), 235–256.

Bouneffouf, D., Parthasarathy, S., Samulowitz, H. and Wistub, M. (2019), 'Optimal exploitation of clustering and history information in multi-armed bandit', *arXiv preprint arXiv:1906.03979* .

Bruno, N. and Chaudhuri, S. (2006), To tune or not to tune?: A lightweight physical design alerter, *in* 'VLDB'.

Bruno, N. and Chaudhuri, S. (2007), An Online Approach to Physical Design Tuning, *in* 'ICDE'.

Cao, B., Pan, S. J., Zhang, Y., Yeung, D.-Y. and Yang, Q. (2010), Adaptive transfer learning, *in* 'AAAI', p. 7.

Csurka, G. (2017), *Domain adaptation in computer vision applications*, Springer.

Dageville, B., Das, D., Dias, K., Yagoub, K., Zaït, M. and Ziauddin, M. (2004), Automatic SQL tuning in Oracle 10g, *in* 'VLDB'.

Das, S., Grbic, M., Ilic, I., Jovandic, I., Jovanovic, A., Narasayya, V. R., Radulovic, M., Stikic, M., Xu, G. and Chaudhuri, S. (2019), Automatically indexing millions of databases in Microsoft Azure SQL Database, *in* 'SIGMOD'.

Kazerouni, A., Ghavamzadeh, M., Abbasi Yadkori, Y. and Van Roy, B. (2017), 'Conservative contextual linear bandits', *Advances in Neural Information Processing Systems* **30**.

Lattimore, T. and Szepesvári, C. (2020), *Bandit algorithms*, Cambridge University Press.

Li, L., Chu, W., Langford, J. and Schapire, R. E. (2010), A contextual-bandit approach to personalized news article recommendation, *in* 'WWW'.

Li, Y., Xie, H., Lin, Y. and Lui, J. C. (2021), Unifying offline causal inference and online bandit learning for data driven decision, *in* 'Proceedings of the Web Conference 2021', WWW '21, Association for Computing Machinery, New York, NY, USA, p. 2291–2303.
URL: *https://doi.org/10.1145/3442381.3449982*

Liu, C.-Y. and Li, L. (2015), 'On the prior sensitivity of thompson sampling', *arXiv preprint arXiv:1506.03378* .

Ma, L., Van Aken, D., Hefny, A., Mezerhane, G., Pavlo, A. and Gordon, G. J. (2018), Query-based workload forecasting for self-driving database management systems, *in* 'SIGMOD'.

Marcus, R., Negi, P., Mao, H., Tatbul, N., Alizadeh, M. and Kraska, T. (2020), 'Bao: Learning to steer query optimizers'. arXiv:2004.03814 [cs.DB].

Oetomo, B., Perera, M., Borovica-Gajic, R. and Rubinstein, B. I. (2019), 'A note on bounding regret of the C^2UCB contextual combinatorial bandit', *arXiv preprint arXiv:1902.07500* .

Perera, R. M., Oetomo, B., Rubinstein, B. I. P. and Borovica-Gajic, R. (2021), DBA bandits: Self-driving index tuning under ad-hoc, analytical workloads with safety guarantees, *in* '2021 IEEE 37th International Conference on Data Engineering', ICDE.

Qin, L., Chen, S. and Zhu, X. (2014), Contextual combinatorial bandit and its application on diversified online recommendation, *in* 'SDM'.

Sattler, K.-U., Schallehn, E. and Geist, I. (2004), Autonomous query-driven index tuning, *in* 'IDEAS'.

Schnaitter, K., Abiteboul, S., Milo, T. and Polyzotis, N. (2007), On-Line Index Selection for Shifting Workloads, *in* 'ICDEW'.

Shivaswamy, P. and Joachims, T. (2012), Multi-armed bandit problems with history, *in* 'Artificial Intelligence and Statistics', PMLR, pp. 1046–1054.

Slivkins, A. (2019), 'Introduction to multi-armed bandits', *Foundations and Trends in Machine Learning* **12**(1-2), 1–286.

Song, C. and Xia, S.-T. (2016), 'Bayesian linear regression with Student-t assumptions', *arXiv preprint arXiv:1604.04434* .

Thompson, W. R. (1933), 'On the likelihood that one unknown probability exceeds another in view of the evidence of two samples', *Biometrika* **25**(3–4), 285–294.

TPC (n.d.), 'TPC-H benchmark', http://www.tpc.org/tpch/.

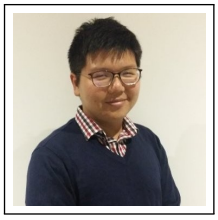
Tran-Thanh, L., Stein, S., Rogers, A. and Jennings, N. R. (2014), 'Efficient crowdsourcing of unknown experts using bounded multi-armed bandits', *Artificial Intelligence* **214**, 89–111.

Wang, L., Wang, C., Wang, K. and He, X. (2017), Biucb: A contextual bandit algorithm for cold-start and diversified recommendation, *in* '2017 IEEE International Conference on Big Knowledge (ICBK)', IEEE, pp. 248–253.

Zhang, C., Agarwal, A., Daumé III, H., Langford, J. and Negahban, S. (2019), Warm-starting contextual bandits: Robustly combining supervised and bandit feedback, *in* 'Proceedings of the 36th International Conference on Machine Learning', pp. 7335–7344.

Zilio, D. C., Rao, J., Lightstone, S., Lohman, G. M., Storm, A. J., Garcia-Arellano, C. and Fadden, S. (2004), DB2 design advisor: Integrated automatic physical database design, *in* 'VLDB'.

Author Biographies



Bastian Oetomo is a PhD candidate in the School of Computing and Information Systems at the University of Melbourne. His research is on multi-armed bandit applied to databases. He completed his DMathSc (Applied Mathematics), BSc (Mechanical Systems) and MEng (Electrical) at the University of Melbourne. During his studies, he received multiple student awards for his performance.

Malinga Perera is a PhD candidate in the School of Computing and Information Systems at the University of Melbourne. His research interests include machine learning and databases. He was part of the team designing multiple large scale big-data systems which won national-level awards in Sri Lanka (1st runner up 'Best Technology or Framework Innovation' in SLASSCOM Innovation Awards 2019). He completed his bachelor's in the University of Moratuwa, Sri Lanka (CS and Eng).

Renata Borovica-Gajic holds a position of Senior Lecturer in Data Analytics in the School of Computing and Information Systems at The University of Melbourne. Dr Borovica-Gajic received her Ph.D. degree in Computer Science from Swiss Federal Institute of Technology in Lausanne (EPFL), Switzerland. Renata's research focuses on solving data management problems when storing, accessing and processing massive data sets, enabling faster, more predictable, and cheaper data analysis as a result. She is also interested in the topics of scientific data management, data exploration, query optimization, physical database design, and hardware-software co-design. Her work has repeatedly appeared in premier data management outlets, including SIGMOD, VLDB, ICDE and VLDB Journal, and she is a recipient of the SIGMOD 2022 Test-of-Time Award as well as Australian Research Council (ARC) DECRA Fellowship.

Benjamin Rubinstein is a Professor in the School of Computing and Information Systems, at the University of Melbourne. His research interests span machine learning, security & privacy, and databases. He has been part of teams that have: analysed privacy of products at the Australian Bureau of Statistics, the financial industry, and Transport for NSW; robustness of translation systems to data poisoning attacks with Meta; helped identify and plug side-channel attacks against the Firefox browser; deanonymised Victorian Myki transport data and an unprecedented Australian Medicare data release; developed scalable Bayesian approaches to record linkage tested by U.S. Census; and shipped production systems for entity resolution in Bing and the Xbox360. Rubinstein completed a BSc (Pure Maths), BEng (Software Hons.), MCompSci (Research) at the University of Melbourne, and a PhD (CS) at UC Berkeley in 2010.