Cutting to the Chase with Warm-Start Contextual Bandits

Bastian Oetomo, R. Malinga Perera, Renata Borovica-Gajic, Benjamin I. P. Rubinstein
School of Computing and Information Systems, The University of Melbourne
{b.oetomo, malinga.perera}@student.unimelb.edu.au, {renata.borovica, brubinstein}@unimelb.edu.au

***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 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. 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.**

***Index Terms*—multi-armed bandits, warm-start, pre-training**

I. Introduction

Multi-armed bandits have undergone a renaissance in machine learning research [1], [2] with a range of deep theoretical results discovered, while applications to real-world sequential decision making under uncertainty abound, ranging from news [3] and movie recommendation [4], to crowd sourcing [5] and self-driving databases [6], [7]. The relative simplicity of the stochastic bandit setting, as compared to more general POMDPs (partially observable Markov decision processes), regularly 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 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. [3] 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 [8] and domain adaptation [9] while [10] modify any bandit policy to make use of pre-training from (batch) supervised learning via manipulation of its rewards weight and importance sampling. This paper advocates for Thompson Sampling (TS) [11] as a natural framework for warm start bandits. Although the prior used in Thompson Sampling can be misspecified, as discussed by [12], our extension to the Linear TS 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 [10]. 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 [13], [14], [15]. 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 [16], [17], [18], [19],

[20], [21]. Unfortunately most lack any form of performance guarantee. Recent work has demonstrated compelling potential for linear bandits for index selection [6] 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 Linear Thompson Sampling and extend our technique to ϵ-greedy and LinUCB;
- Our Warm Start Linear Bandit algorithm can incorporate prior knowledge from supervised learning (like [10]), but also prior bandit learning, or manual construction of a prior by a domain expert, for example. Notably our warm start approach incorporates uncertainty quantification;
- We present a regret bound for Warm Start Linear TS that demonstrates sublinear regret for long-term performance; and
- 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).

II. 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:

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

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 [11]. 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.

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

Linear Thompson Sampling (TS) [22], [23] 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, Linear TS 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

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

Then [23] 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 Linear TS [22] do not include a tunable regularisation parameter.

A result due to [24] is used within Linear TS: 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, [23] 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 $\mathbb{P}_{\boldsymbol{\eta}\sim\mathcal{D}^{TS}}\left(\|\boldsymbol{\eta}\| \leq \sqrt{cd\log\frac{c'd}{\delta}}\right) \geq 1-\delta$,

then Linear TS is Hannan consistent. We adopt a standard multivariate Gaussian for $\boldsymbol{\eta}_t$ which satisfies the above properties [23]. With all of these definitions in mind, the version of Linear TS used in this paper can be summarised as shown in Algorithm 1.

III. Warm Starting Linear Bandits

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

A. 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 Linear TS due to [23] 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 Linear TS.

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 $\hat{\boldsymbol{\theta}}_t = \boldsymbol{V}_t^{-1}(\lambda\hat{\boldsymbol{\theta}}_1 + \sum_{s=1}^{t-1}\boldsymbol{x}_s(i_s)r_t(i_s))$ 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 prior to bandit learning, and the actual parameter. **This prior 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 [10]. 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 $\boldsymbol{\delta} \sim \mathcal{N}(\boldsymbol{0}, (\boldsymbol{\Sigma}_\mu + \alpha^{-1}\boldsymbol{I}_d))$.

The next result derives a generalisation of 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 $\boldsymbol{\theta} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{V}_1^{-1})$. 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:

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

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

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

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 in setting $\bar{\boldsymbol{\delta}} \sim \mathcal{N}(\boldsymbol{0}, \alpha^{-1}\boldsymbol{I})$: 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 V.

Our simple reduction of warm start bandit learning to Linear TS admits a regret bound. We follow the pattern of the regret analysis of [23] with differences detailed next.

Observe first that $\|\hat{\boldsymbol{\theta}}_t - \boldsymbol{\theta}_\star\|_{\boldsymbol{V}_t} = \|(\hat{\boldsymbol{\theta}}_t - \hat{\boldsymbol{\mu}}) - (\boldsymbol{\theta}_\star - \hat{\boldsymbol{\mu}})\|_{\boldsymbol{V}_t} = \|\hat{\boldsymbol{\delta}}_t - \boldsymbol{\delta}_\star\|_{\boldsymbol{V}_t} \le \beta_t(\delta')$. Accordingly the argument yielding the confidence ellipsoid $\beta_t(\delta')$ stated in [24, 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 [23] as follows:

- the inequality proposed by [24] 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 [23, Proposition 2] is no longer valid in our case. However, the last inequality in [25] has modified [23, Proposition 2] into:

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

  and hence serves our purpose; and
- in proving [23, Theorem 1] the authors used the fact that $\boldsymbol{V}_t^{-1} \le \frac{1}{\lambda}\boldsymbol{I}$. This is not the case in our setting, but we can generalise the result with similar reasoning yielding $\boldsymbol{V}_t^{-1} \le \frac{1}{\lambda_{min}(R^2\boldsymbol{V}_1)}\boldsymbol{I}$, 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 Linear TS Regret Bound)**.** *Under the assumptions that:*

Algorithm 2 Warm Start Linear Thompson Sampler

Input: $\hat{\boldsymbol{\mu}}, \alpha, \boldsymbol{\Sigma}_\mu, \delta, T, R$
Initialize $\hat{\boldsymbol{\delta}}_1 \leftarrow \boldsymbol{0}$, $\boldsymbol{V}_1 \leftarrow R^2(\boldsymbol{\Sigma}_\mu + \alpha^{-1}\boldsymbol{I}_d)^{-1}$, $\delta' \leftarrow \frac{\delta}{4T}, \boldsymbol{b}_1 \leftarrow \boldsymbol{0}$
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

1) $\|\boldsymbol{x}\| \le 1$ *for all* $x \in \mathcal{X}$*;*
2) $\|\boldsymbol{\delta}\| \le S$ *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 II, 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 Linear TS can be decomposed as*

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

with each of the term bounded as

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

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

B. 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 phases. 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 $S = \{s : s \in \arg\max_{i\in[k]} \hat{\boldsymbol{\theta}}_t^T\boldsymbol{x}_t(i)\}$ 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

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

is a powerful result. It was proposed in [3] 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:

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

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

IV. 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 [26]. 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 [10]. 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 Linear TS. 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.

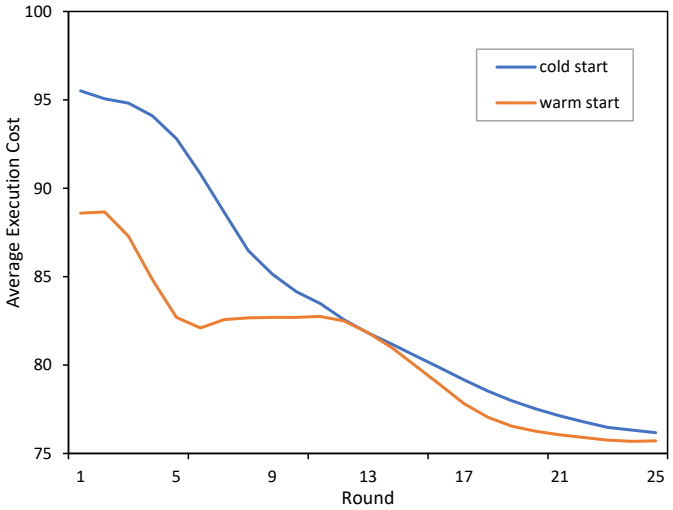


Fig. 1: Cold Start vs. Warm Start Linear TS for database index selection on the the TPC-H benchmark.

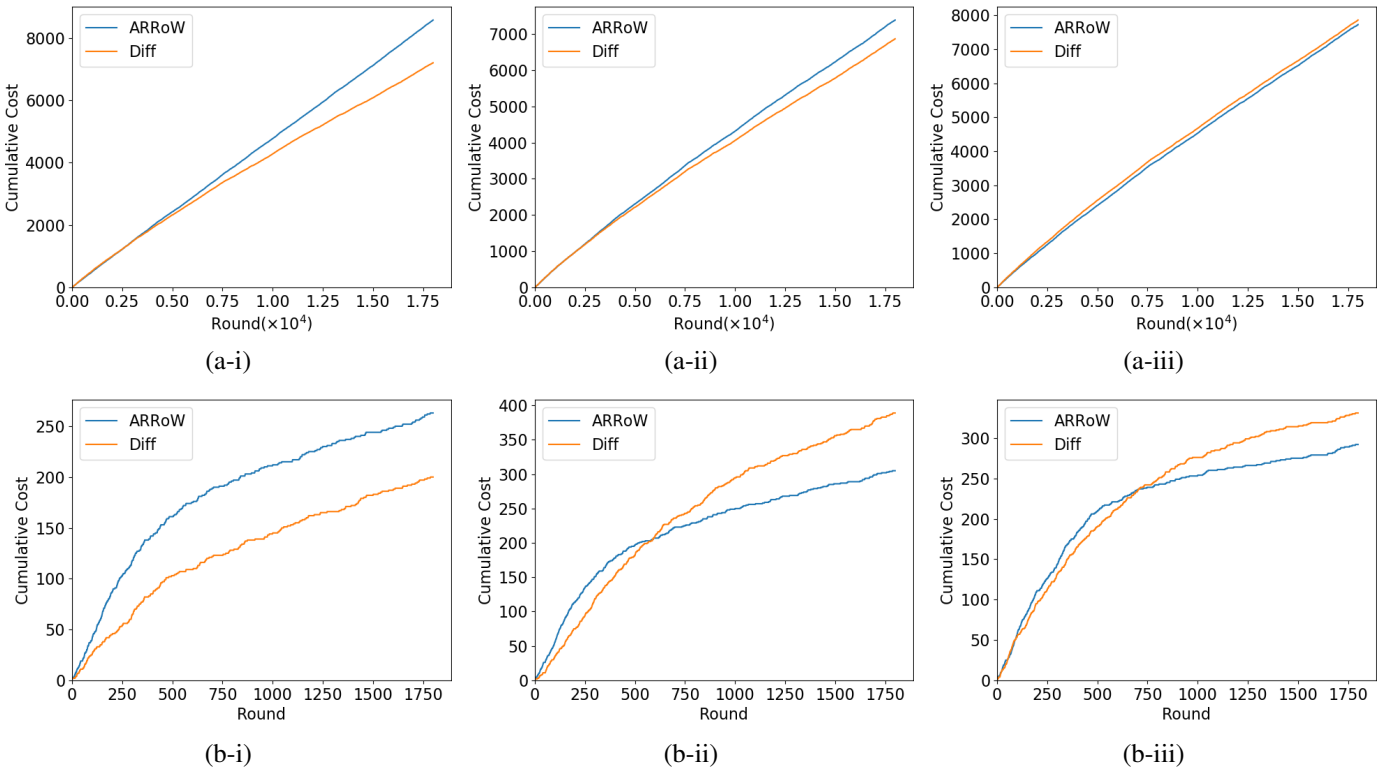


Fig. 2: Comparisons of both our and ARRoW warm-start frameworks on the (row a) Letters and (b) Numbers datasets, with learners (column i) ϵ-greedy, (ii) LinUCB and (iii) TS.

A. *Database Index Selection*

For the real-world experiment, we provide an application to a version of the database index selection problem, with problem setting 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 [26]. In each round, five TPC-H query templates are randomised to represent the workload at round t.

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 [6], 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 Linear TS outperforms the cold-started Linear TS, in short-term rounds and cumulatively. This can be explained by the query templates used to pre-train the warm-started bandit resemble the templates used in the testing dataset, leading the warm-started bandit to guess the initial weight $\boldsymbol{\theta}_1 = \hat{\boldsymbol{\mu}}$ to be closer to the actual weight $\boldsymbol{\theta}_\star$ compared to the initial guess of $\boldsymbol{\theta}_1 = \mathbf{0}$ by the cold-started bandit.

B. *OpenML Classification Dataset*

We chose two of the datasets used in [10], 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 [10] as the only form of permissible prior knowledge. We try all learners presented in this paper for this dataset: ϵ-greedy, LinUCB and Linear Thompson Sampling. As for the hyperparameters, we used $\epsilon = 0.0125$ for ϵ-greedy, $\rho R = 0.2$ for LinUCB, $\beta_t(\delta) = 1$ for TS in Letter dataset and $\beta_t(\delta) = 0.05$ for TS in Numbers dataset. All of these hyperparameters were found iteratively by grid search.

As described in [10], 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

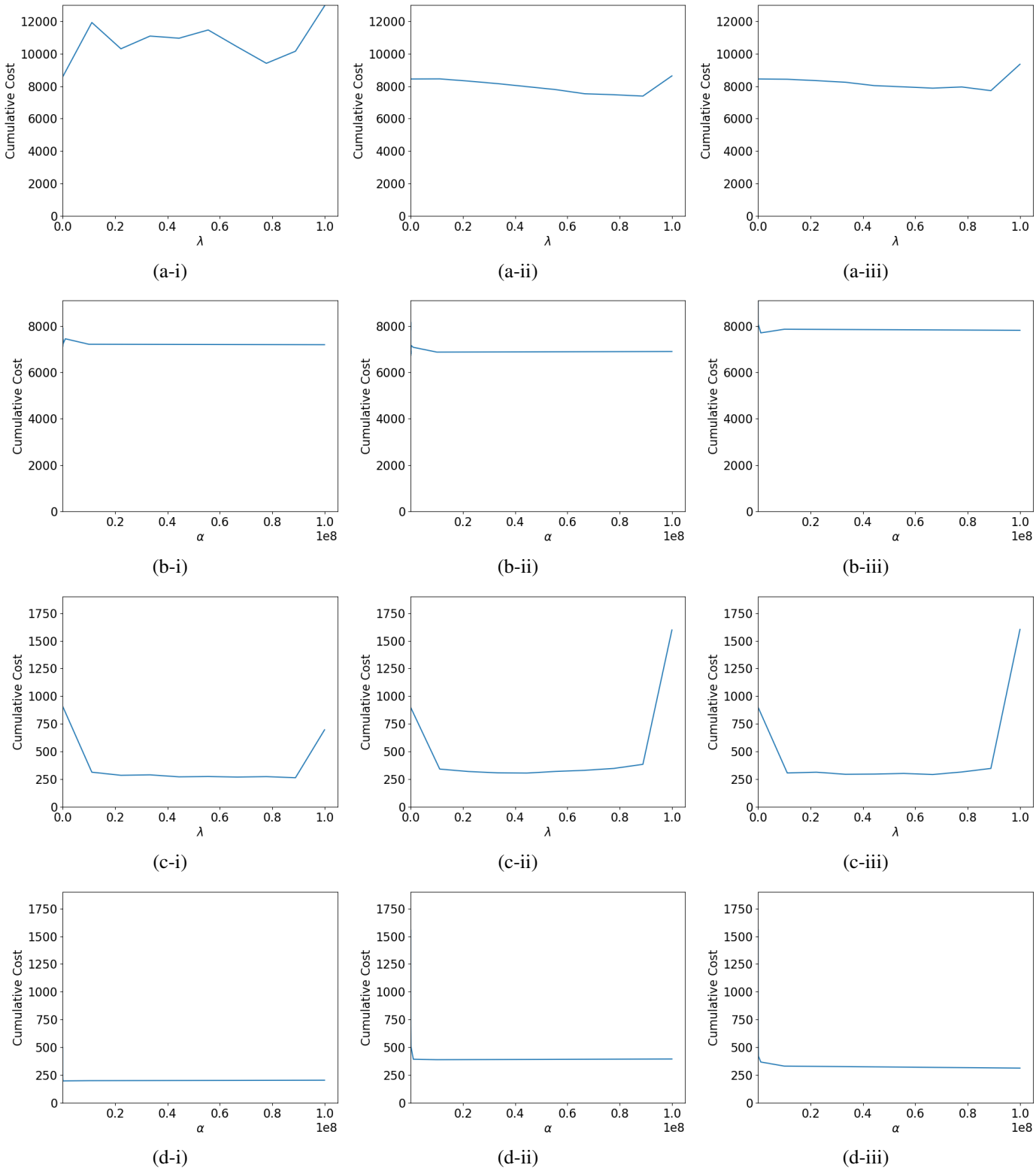
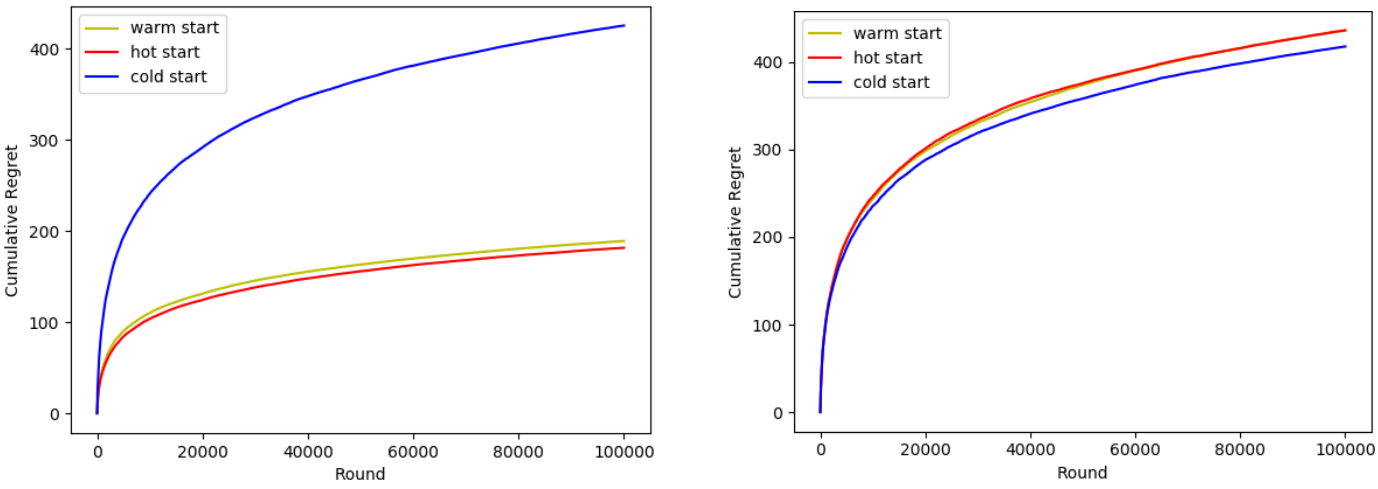


Fig. 3: Sensitivity analysis showing total cumulative cost achieved vs. hyperparameter. Rows (a,b) are on the Letters dataset while rows (c,d) are on Numbers. Rows (a,c) demonstrate ARRoW results with varying λ while (b,d) show our warm start approach Diff with varying α. Finally the learners vary over (column i) ϵ-greedy, (ii) LinUCB, (iii) LinTS.

making the parameter different, leading to the disjoint bandit with arm i having the weight $\boldsymbol{\theta_{i,t}}$. 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



(a) (b)

Fig. 4: Artificial dataset experimental results for (a) an accurate prior and (b) a misspecified prior, comparing cold-, warm- and hot-start TS.

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 [10] 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, demonstrate that the choices are generally not very important.

We follow a suggestion of the original ARRoW paper to evaluate [10, 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.

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 the previous algorithm, whilst still offering the flexibility to choose the initial guess.

Sensitivity analysis is presented in Figure 3. As mentioned, neither warm start approach is very sensitive to their hyperparameters. These results also support our choice of $\alpha = 10^7$ across these experiments.

C. 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 $\boldsymbol{\theta}_\star^T = \begin{bmatrix} 0.1 & 0.3 & 0.5 & 0.7 & 0.9 \end{bmatrix}$, with the 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 4.

Results. In the accurate prior regime, it is clear that the hot-started and warm-started bandits overperform 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.

V. 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 Sampler [23], 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 [3], 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.

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 drift hyperparameter. Our result suggest tuning α^{-1}, which reflects drift during warm start. We expect large α^{-1} when tasks are dissimilar, and small values for similar tasks. It is intuitively appealing to attempt to use newly observed data for adapting α^{-1}. Approaches that might bear fruit include: Hierarchical Bayes via multi-level modelling [27], or using Empirical Bayes via evidence maximisation [28].

Under an adaptive hyperparameter, α^{-1} is no longer independent of the other variables. This violates one of the assumptions made in [24], as the choice of λ in their scenario is independent of other variables. Therefore, the validity of the oversampling factor becomes questionable. As the regret analysis for Linear TS depends on the validity of the upper bound provided by [24], this in turns becomes invalid as well. As such regret analysis for the adaptive case would become another open problem.

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 are not. 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 Lin TS, provided performance is not sacrificed.

Appendix

A. Full Proof of the Regret Bound

We now detail the full proof of Theorem 2, by extending a previous analysis [24]. 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:

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

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

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

Now [24, 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

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

and

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

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:

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

Thus, we conclude that

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

REFERENCES

[1] A. Slivkins, "Introduction to multi-armed bandits," *Foundations and Trends in Machine Learning*, vol. 12, no. 1-2, pp. 1–286, 2019.

[2] T. Lattimore and C. Szepesvári, *Bandit algorithms*. Cambridge University Press, 2020.

[3] L. Li, W. Chu, J. Langford, and R. E. Schapire, "A contextual-bandit approach to personalized news article recommendation," in *WWW*, 2010.

[4] L. Qin, S. Chen, and X. Zhu, "Contextual combinatorial bandit and its application on diversified online recommendation," in *SDM*, 2014.

[5] L. Tran-Thanh, S. Stein, A. Rogers, and N. R. Jennings, "Efficient crowdsourcing of unknown experts using bounded multi-armed bandits," *Artificial Intelligence*, vol. 214, pp. 89–111, 2014.

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

[7] R. Marcus, P. Negi, H. Mao, N. Tatbul, M. Alizadeh, and T. Kraska, "Bao: Learning to steer query optimizers," 2020. arXiv:2004.03814 [cs.DB].

[8] B. Cao, S. J. Pan, Y. Zhang, D.-Y. Yeung, and Q. Yang, "Adaptive transfer learning," in *AAAI*, p. 7, 2010.

[9] G. Csurka, *Domain adaptation in computer vision applications*. Springer, 2017.

[10] C. Zhang, A. Agarwal, H. D. Iii, J. Langford, and S. Negahban, "Warm-starting contextual bandits: Robustly combining supervised and bandit feedback," in *Proceedings of the 36th International Conference on Machine Learning*, pp. 7335–7344, 2019.

[11] W. R. Thompson, "On the likelihood that one unknown probability exceeds another in view of the evidence of two samples," *Biometrika*, vol. 25, no. 3–4, pp. 285–294, 1933.

[12] C.-Y. Liu and L. Li, "On the prior sensitivity of thompson sampling," *arXiv preprint arXiv:1506.03378*, 2015.

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

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

[15] B. Dageville, D. Das, K. Dias, K. Yagoub, M. Zaït, and M. Ziauddin, "Automatic SQL tuning in Oracle 10g," in *VLDB*, 2004.

[16] K. Schnaitter, S. Abiteboul, T. Milo, and N. Polyzotis, "On-Line Index Selection for Shifting Workloads," in *ICDEW*, 2007.

[17] K.-U. Sattler, E. Schallehn, and I. Geist, "Autonomous query-driven index tuning," in *IDEAS*, 2004.

[18] N. Bruno and S. Chaudhuri, "An Online Approach to Physical Design Tuning," in *ICDE*, 2007.

[19] N. Bruno and S. Chaudhuri, "To tune or not to tune?: A lightweight physical design alerter," in *VLDB*, 2006.

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

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

[22] S. Agrawal and N. Goyal, "Thompson sampling for contextual bandits with linear payoffs," in *International Conference on Machine Learning*, pp. 127–135, 2013.

[23] M. Abeille, A. Lazaric, *et al.*, "Linear Thompson sampling revisited," *Electronic Journal of Statistics*, vol. 11, no. 2, pp. 5165–5197, 2017.

[24] Y. Abbasi-Yadkori, D. Pál, and C. Szepesvári, "Improved algorithms for linear stochastic bandits," *Advances in Neural Information Processing Systems*, vol. 24, pp. 2312–2320, 2011.

[25] B. Oetomo, M. Perera, R. Borovica-Gajic, and B. I. Rubinstein, "A note on bounding regret of the C^{2}UCB contextual combinatorial bandit," *arXiv preprint arXiv:1902.07500*, 2019.

[26] TPC, "TPC-H benchmark." http://www.tpc.org/tpch/.

[27] G. M. Allenby and P. E. Rossi, "Hierarchical Bayes models," *The handbook of marketing research: Uses, misuses, and future advances*, pp. 418–440, 2006.

[28] C. Song and S.-T. Xia, "Bayesian linear regression with Student-t assumptions," *arXiv preprint arXiv:1604.04434*, 2016.