Proceedings of CONF-MLA 2024 Workshop: Semantic Communication Based Complexity Scalable Image Transmission System for Resource Constrained Devices
DOI: 10.54254/2755-2721/83/2024GLG0071

A Survey on Variants of Thompson Sampling

Junqing Yang

Shanghai Jiao Tong University (SJTU), 800 Dongchuan RD. Minhang District, Shanghai, China

shjtdxyjq@sjtu.edu.cn

Abstract. Thompson Sampling has become a prominent algorithmic approach in recent years. This review focuses on the evolution of TS and its variants, showing the innovative aspects of Neural Thompson Sampling (NeuralTS) and Meta-Thompson Sampling (Meta-TS), explaining the aggressive strategy used by Feel-Good Thompson Sampling (FGTS) and the introduction to Safe-LTS for Linear Thompson Sampling (LTS) problem. The survey first systematically review the literature, then examine the theoretical underpinnings, algorithmic frameworks and innovations of those TS variants, in the end provide our insights in future directions. In short, NeuralTS handles high-dimensional reward functions through deep learning integration, Meta-TS takes advantage of meta-learning for adapting to unknown prior distributions, FGTS applies an aggressive exploration strategy to handle pessimistic scenarios. In the end, this paper suggests that future research should emphasis on enhancing generalizability, bridging the gap between theory and practice, and improving adaptability to complex and dynamic environments.

Keywords: Thompson Sampling, Neural Thompson Sampling, Meta-Thompson Sampling, Feel-Good Thompson Sampling, Sliding-Window Thompson Sampling.

1. Introduction

The Multi-Armed Bandits Problem (MABP) is a core issue in decision theory and is widely applied in fields such as recommendation systems, medical decision-making, and online advertising. Thompson Sampling (TS) algorithm is introduced to handle MABP, because it can achieve a good balance between exploration and exploitation. The research on TS and Linear TS (LTS) is important because TS leverages Bayesian inference for decision-making and adapts to environmental changes due to its reliance on linear models to predict rewards, making it particularly useful in dynamic settings. In the past few years, scholars have conducted in-depth research on TA and proposed various improvements.

In 2019, Phan et al. examined the impact of approximate inference errors and proposed forced exploration to mitigate performance degradation [1]. This approach helps to enhance the robustness of models when faced with uncertainty and complexity. In 2020, significant progress was made across multiple fields. In the field of posterior distribution updates, Zhang et al. introduced a deep learning-based approach called Neural Thompson Sampling [2], which uses neural networks for the reward's posterior distribution, leveraging neural tangent features for variance estimation. In the field non-stationary MABP, Trovo et al. raised an algorithm called Sliding-Window Thompson Sampling (SW-TS) [3], using a sliding-window technique to handle abrupt and smooth changes. SW-TS provides upper bounds on the dynamic pseudo-regret for non-stationary environments, offering a reliable solution for decision-making problems in non-stationary environments. In the field of LTS under safety constraints, Moradipari et al. studied the impact of unknown linear safety constraints and proposed a safe algorithm called Safe-LTS to ensure constraints are met [4]. A year later, Moradipari et al. further explored safe exploration under linear constraints, providing a new algorithm based on LTS [5]. Also in 2020, Vernade et al. addressed the challenge of delayed and partially observable feedback with OTFLinUCB and OTFLinTS, which integrate information as it becomes available [6]. In the same year, Hamidi and Bayati proposed a data-driven version of LTS that adjusts posterior inflation, achieving minimax optimal regret under certain conditions [7]. In 2021, Kveton et al. proposed a meta-learning variant called Meta-Thompson Sampling [8], which adapts to different bandit instances, demonstrating the benefits of meta-learning with a novel regret bound. The main contribution of this research lies in proposing a novel Bayesian regret bound, which is not only an important theoretical support for the Meta-TS algorithm itself but also offers a new perspective for the entire Bayesian optimization and decision-making process. By treating problem instances as samples from an unknown prior, Meta-TS is able to learn and improve its exploration strategy through continuous interaction. In 2022, Zhang suggested a modification to standard TS called Feel-Good Thompson Sampling [9], to be more aggressive in exploring high-reward models, addressing the issue of suboptimality in pessimistic scenarios. This theoretical framework can be extended to address some Markov Decision Process (MDP) problems, offering a simple yet comprehensive mathematical structure for analysing TS and its variants. In 2024, Gigli explored TS for optimizing hierarchical digital marketing campaigns, proposing a parametric model that enhances efficiency and convergence [10]. In the same year, Zheng et al. introduced an approximate strategy using Underdamped Langevin Monte Carlo for high-dimensional posteriors [11], providing a more efficient way to sample from complex distributions, allowing TS to be applied to more sophisticated problems.

The above literature has witnessed a surge in modifications and enhancements to the traditional TS algorithm, each tailored to address specific problems or to leverage new theoretical insights. However, a cohesive understanding of these developments and their interplay is still lacking. This review takes a multi-dimensional approach to bridge the knowledge gap. It starts with a systematic literature review of advanced TS developments from prominent conferences or preprints. Then it chooses some of the most promising new algorithms or optimizations, looks into how these innovations have been adapted to new problem areas, and examines their mathematical bases and theoretical support for their effectiveness. In the end, this paper assesses the strengths and limitations and suggests directions for future work. Hopefully, this review can provide ideas for improving the universality of related algorithms and provide references for the selection of Thompson Sampling algorithms in complex situations.

2. Thompson Sampling Algorithm for Multi-Armed Bandit Problem

Multi-armed Bandits problem is a leading problem in the field of reinforcement learning, since a lot of advanced algorithms anchor in this problem settings. Thompson Sampling algorithm is one of the most efficient algorithms that are designed to solve this problem. Research on this is always valuable.

2.1. An introduction to Thompson Sampling Algorithm

MABP is a foundational dilemma in machine learning where an agent must choose between exploring unknown options for potentially greater reward or exploiting known options to maximize immediate gains. It's often framed as a gambler choosing which of several slot machines (each a "bandit") to play, with the goal of maximizing total winnings over time. Thompson Sampling is an online decision-making algorithm widely used in MABP. Its core principle is using Bayesian inference to estimate the reward probabilities for each arm and randomly sample from these probability distributions at each step, thereby balancing the trade-off between exploration and exploitation.

2.2. The Advantages of Thompson Sampling Algorithm

Many kinds of algorithms are created to solve MABP, such as greedy algorithm and Upper Confidence Bound (UCB) algorithm, but TS edges over them for certain reasons. For example, greedy algorithms always choose the currently estimated optimal arm, which can lead to the algorithm getting stuck in a local optimum and neglecting exploration of other arms, while TS introduces an exploration mechanism through random sampling, helping to avoid this situation. Another example is UCB algorithms, which use a deterministic strategy to select actions by calculating the confidence upper bounds for each arm, sometimes too conservative in some cases. TS algorithm deals the problem by offering better exploration capabilities since it is based on probabilistic random strategy.

2.3. Evaluation Metrics for Multi-Armed Bandit Algorithms

When comparing the performance of different multi-armed bandit algorithms, several key metrics are commonly used in the literature. Here are some of the most important ones:

1) Regret (or Expected Regret): This is perhaps the most fundamental metric in the bandit literature. Regret measures the difference between the total reward that could have been achieved by always choosing the best arm (the one with the highest expected reward) and the actual total reward achieved by the algorithm. It's a measure of the opportunity cost of not always choosing the optimal arm. Mathematically, if r_{max} is the reward of the best arm and r_t is the reward of the chosen arm at round t, the cumulative regret R_T after T rounds is given by:

$$R_T = E\left[\sum_{t=1}^{T}(r_{max} - r_t)\right] \tag{1}$$

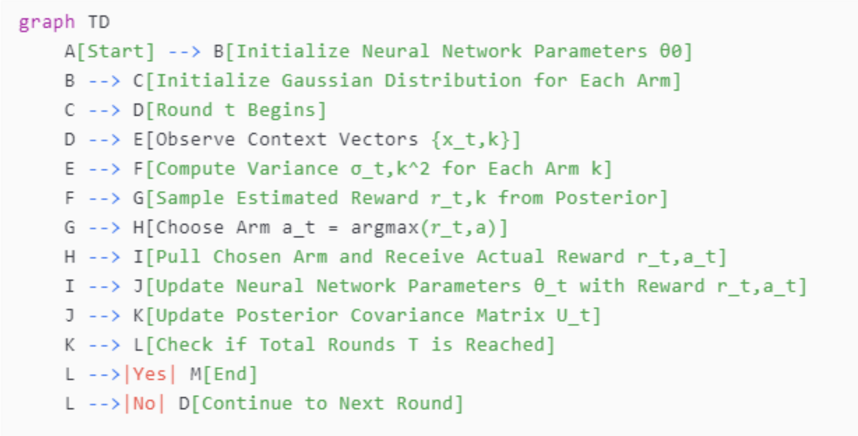
2) Cumulative Reward: This metric simply sums up the rewards obtained by the algorithm. It's a direct measure of the algorithm's performance but does not take into account the potential rewards that could have been obtained by choosing other arms.
3) Average Reward per Round: Sometimes, especially in problems where the number of rounds T is very large, the average reward per round is used instead of the cumulative reward. This normalizes the performance across different numbers of rounds.
4) Exploration Probability: This metric measures the proportion of times the algorithm chooses to explore (i.e., choose an arm other than the one with the highest current estimated reward). It's a direct measure of the algorithm's exploration behaviour.
5) Convergence Time: This is the number of rounds it takes for the algorithm to consistently choose the optimal arm. It's a measure of how quickly the algorithm learns the best arm.
6) Variance of Rewards: Some algorithms achieve high rewards but with high variability. The variance of the rewards gives insights into the stability of the algorithm's performance.

3. The Combination of Neural Networks and Thompson Sampling

Deep neural networks are widely used in fields such as image recognition, speech recognition, natural language processing, gamers, and robot control. It is believed that deep neural networks could also be applied to the upgrade of TS algorithm. NeuralTS, which integrates deep neural networks with TS to balance exploration and exploitation in complex, real-world applications, was proposed on this idea.

3.1. The Core Principle of NeuralTS

Unlike traditional bandit algorithms, which might sometimes struggle with high-dimensional reward function approximation, NeuralTS leverages the expressive power of neural networks to effectively handle such complexity. The novelty of NeuralTS lies in its approach to uncertainty estimation, where it considers weight uncertainty across all layers of the neural network, not just the last layer as in some previous methods. This allows NeuralTS to provide a more accurate posterior distribution of rewards, which is crucial for the algorithm's performance. The algorithm is built around a novel posterior distribution of the reward, with the mean represented by the neural network approximator and the variance derived from the neural tangent features of the network. The pseudocode of a simple flowchart for NeuralTS is shown in **Figure 1**.



```
graph TD
    A[Start] --> B[Initialize Neural Network Parameters θ0]
    B --> C[Initialize Gaussian Distribution for Each Arm]
    C --> D[Round t Begins]
    D --> E[Observe Context Vectors {x_t,k}]
    E --> F[Compute Variance σ_t,k^2 for Each Arm k]
    F --> G[Sample Estimated Reward r_t,k from Posterior]
    G --> H[Choose Arm a_t = argmax(r_t,a)]
    H --> I[Pull Chosen Arm and Receive Actual Reward r_t,a_t]
    I --> J[Update Neural Network Parameters θ_t with Reward r_t,a_t]
    J --> K[Update Posterior Covariance Matrix U_t]
    K --> L[Check if Total Rounds T is Reached]
    L -->|Yes| M[End]
    L -->|No| D[Continue to Next Round]
```

Figure 1. Pseudocode of a simple flowchart for NeuralTS.

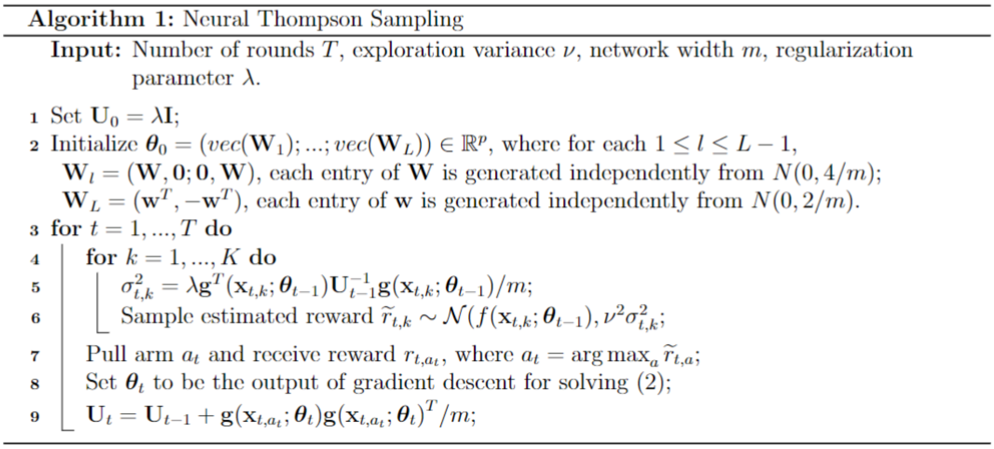
3.2. Algorithmic Framework of NeuralTS

NeuralTS requires a set of parameters including the number of rounds T, exploration variance v, network width m, and regularization parameter λ, used to balance exploration and exploitation and control model complexity. The algorithm commences by initializing the posterior covariance matrix U_0 and the neural network parameters θ_0 with selected random values to promote diverse exploration. During each round, it computes the variance of the reward's posterior distribution for each arm, samples an estimated reward from this distribution, and selects the arm that maximizes the expected reward. The phase of post-reward observation refines the neural network parameters through gradient descent, optimizing a loss function that incorporates an ℓ_2-regularization term to raise generalization. The covariance matrix is concurrently updated to reflect the newly acquired data, encapsulating the uncertainty in the reward estimation.

NeuralTS is given in **Algorithm 1** as is shown in **Figure 2**. It maintains a Gaussian distribution for each arm's reward. When selecting an arm, it samples the reward of each arm from the reward's posterior distribution, and then pulls the greedy arm. Once the reward is observed, it updates the posterior. The mean of the posterior distribution is set to the output of the neural network, whose parameter is the solution to the following minimization problem:

$$(\min_{\theta} L(\boldsymbol{\theta}) = \sum_{i=0}^{t} \left[f\left(\boldsymbol{x}_{i,a_i};\boldsymbol{\theta}\right) - r_{i,a_i}\right]^2/2 + m\lambda\|\boldsymbol{\theta} - \boldsymbol{\theta}_{\mathbf{0}}\|_2^2/2 \qquad (2)$$

It's obvious that (1) is an ℓ_2-regularized square loss minimization problem, where the regularization term centres at the randomly initialized network parameter $\boldsymbol{\theta}_0$. The algorithm adapts gradient descent to solve (2) with step size η and total number of iterations $\boldsymbol{J}$.



```
Algorithm 1: Neural Thompson Sampling
  Input: Number of rounds $T$, exploration variance $\nu$, network width $m$, regularization
         parameter $\lambda$.
Set $\mathbf{U}_0 = \lambda\mathbf{I}$;
Initialize $\boldsymbol{\theta}_0 = (vec(\mathbf{W}_1); ...; vec(\mathbf{W}_L)) \in \mathbb{R}^p$, where for each $1 \leq l \leq L-1$,
  $\mathbf{W}_l = (\mathbf{W}, \mathbf{0}; \mathbf{0}, \mathbf{W})$, each entry of $\mathbf{W}$ is generated independently from $N(0, 4/m)$;
  $\mathbf{W}_L = (\mathbf{w}^T, -\mathbf{w}^T)$, each entry of $\mathbf{w}$ is generated independently from $N(0, 2/m)$.
for $t = 1, ..., T$ do
    for $k = 1, ..., K$ do
        $\sigma_{t,k}^2 = \lambda \mathbf{g}^T(\mathbf{x}_{t,k}; \boldsymbol{\theta}_{t-1}) \mathbf{U}_{t-1}^{-1} \mathbf{g}(\mathbf{x}_{t,k}; \boldsymbol{\theta}_{t-1})/m$;
        Sample estimated reward $\widetilde{r}_{t,k} \sim \mathcal{N}(f(\mathbf{x}_{t,k}; \boldsymbol{\theta}_{t-1}), \nu^2 \sigma_{t,k}^2$;
    Pull arm $a_t$ and receive reward $r_{t,a_t}$, where $a_t = \arg\max_a \widetilde{r}_{t,a}$;
    Set $\boldsymbol{\theta}_t$ to be the output of gradient descent for solving (2);
    $\mathbf{U}_t = \mathbf{U}_{t-1} + \mathbf{g}(\mathbf{x}_{t,a_t}; \boldsymbol{\theta}_t) \mathbf{g}(\mathbf{x}_{t,a_t}; \boldsymbol{\theta}_t)^T/m$;
```

Figure 2. Pseudocode for Neural Thompson Sampling algorithm.

3.3. Theoretical Guarantees and Empirical Validation

The theoretical bases of NeuralTS are robust. Theoretical analysis guarantees that, assuming a bounded reward function, NeuralTS achieves a cumulative regret of *O*(*T*), which aligns with the regret bounds of other leading contextual bandit algorithms in terms of the total number of rounds *T*. This *O*(*T*) regret bound is a testament to the algorithm's efficiency in balancing exploration and exploitation.

The impact of the neural network's architecture and the variance of the exploration noise on the regret bound can be considered to further substantiate the theoretical guarantees. By adjusting the network's width and depth, the approximation capability of the neural network can be modulated, thus influencing the rate at which the algorithm converges to the optimal policy. Additionally, the choice of exploration variance is critical. A higher variance promotes exploration at the cost of increased short-term regret, while a lower variance facilitates more rapid exploitation but may delay discovering the optimal arm.

Empirical validation has been conducted across a diverse array of datasets, and the performance of NeuralTS is benchmarked against several cutting-edge algorithms, such as linear and kernelized TS, UCB methods, and neural network-based approaches like Bootstrapped DQN. The results consistently demonstrate the competitive performance of NeuralTS, showcasing its ability to generalize across different problem domains and data distributions. It can be stated that NeuralTS' seamless integration of probabilistic exploration with deep learning's predictive prowess positions it as a formidable approach to sequential decision-making in the presence of uncertainty.

3.4. Limitations of NeuralTS and Future Directions

Despite NeuralTS demonstrates strong theoretical appeal and empirical performance, it has limitations such as relatively high computational complexity, especially when updating deep and wide networks through multiple gradient descent steps. Besides, the theoretical guarantees of the algorithm rely on the assumption of a bounded reward function, which may not always hold true in practical applications. Future research should focus on optimizing the algorithm to reduce computational costs, extending its adaptability to unknown or dynamically changing time horizons, and enhancing its robustness to the unbounded reward functions or potential model misspecifications.

4. The Application of Meta Learning in Thompson Sampling

Meta learning is a concept in the field of machine learning that focuses on designing algorithms that can utilize past experience to learn new tasks faster. This idea is actually also beneficial for reinforcement learning. Meta-TS is one of the research results on this idea.

4.1. The Background of Meta-TS

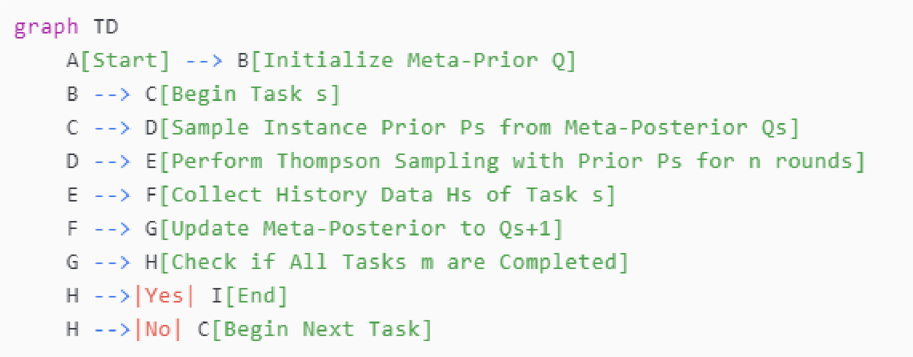
In the previous chapter, NeuralTS is discussed, which focuses on utilizing deep neural networks to better approximate the posterior distribution of reward functions, providing more accurate reward predictions when facing high-dimensional contextual information. But when dealing with multi-task environments, some preprocessing is needed. That's why Meta-TS is created, which focuses on adapting to unknown prior distributions through meta-learning and stresses the rapid adaptation capability of the algorithm to new tasks through meta-learning, the core of which is to enable models to extract valuable knowledge from previous data when facing new problems, thereby improving learning efficiency and performance.

The combined use of NeuralTS and Meta-TS can further enhance the performance of TS in complex circumstances. For example, when facing an unknown environment, Meta-TS can adjust its strategy to adapt to new tasks, while NeuralTS can provide more accurate reward predictions during the adaptation process. This combination not only expands the application range of TS but also provides new ideas for solving more complex reinforcement learning problems.

4.2. The Core Principle of Meta-TS

The idea of meta learning can be well applied to the estimation and updating of prior reward distributions. Meta-TS operates by maintaining a meta-posterior distribution Q_s, which reflects the algorithm's current estimate of the true prior P^*. For each task, it samples a potential prior, P_s, from this meta-posterior and employs it to guide the TS algorithm's actions. A simple flowchart is shown in **Figure 3**.

The sampling of P_s represents an optimistic exploration strategy, enabling Meta-TS to interact with the environment and gather rewards. After the interaction, Meta-TS updates its meta-posterior based on the observed rewards, refining its estimate of P^* in accordance with Bayesian principles. This iterative process of sampling and updating allows Meta-TS to progressively learn the underlying prior, leading to more informed decisions and efficient exploration over successive tasks.



```
graph TD
    A[Start] --> B[Initialize Meta-Prior Q]
    B --> C[Begin Task s]
    C --> D[Sample Instance Prior Ps from Meta-Posterior Qs]
    D --> E[Perform Thompson Sampling with Prior Ps for n rounds]
    E --> F[Collect History Data Hs of Task s]
    F --> G[Update Meta-Posterior to Qs+1]
    G --> H[Check if All Tasks m are Completed]
    H -->|Yes| I[End]
    H -->|No| C[Begin Next Task]
```

Figure 3. Pseudocode of a simple flowchart for Meta-TS.

4.3. Efficient Implementations of Meta-TS

In the paper by Kveton et al., efficient implementations of Meta-TS for both Bernoulli and Gaussian bandits are presented [8]. These implementations take advantage of the properties of these distributions to update the meta-posterior in a computationally efficient manner.

The authors evaluate the performance of Meta-TS using synthetic experiments. The results indicate that Meta-TS can quickly adapt to the unknown prior **P***, and its regret is comparable to that of TS with a known **P***. Particularly in the Gaussian bandit scenario, where the meta-prior width $\boldsymbol{\sigma}_0$ is significantly larger than the instance prior width $\boldsymbol{\sigma}_0$, substantial gains from meta-learning are expected.

The experimental outcomes are in line with this expectation, showing that after just a few tasks, the slope of the Meta-TS regret approaches that of OracleTS, the idealized TS with the true prior **P***. Furthermore, the experimental results demonstrate that the benefits of meta-learning are preserved as the number of arms **K** or dimensions **D** increases. However, these benefits diminish when the prior width $\boldsymbol{\sigma}_0$ approaches the meta-prior width $\boldsymbol{\sigma}_0$. In such cases, there is little advantage to adapting to **P***, and all methods perform similarly. The authors also experimented with mis-specified Meta-TS, showing that the impact of mis-specification is relatively minor, which attests to the robustness of Meta-TS.

In summary, Meta-TS showcases its adaptability and effectiveness under unknown prior conditions through its efficient implementation and theoretical analysis. These experimental results not only substantiate the practicality of the Meta-TS algorithm but also provide strong evidence for the potential application of meta-learning in more complex settings.

4.4. Limitations of Meta-TS and Future Directions

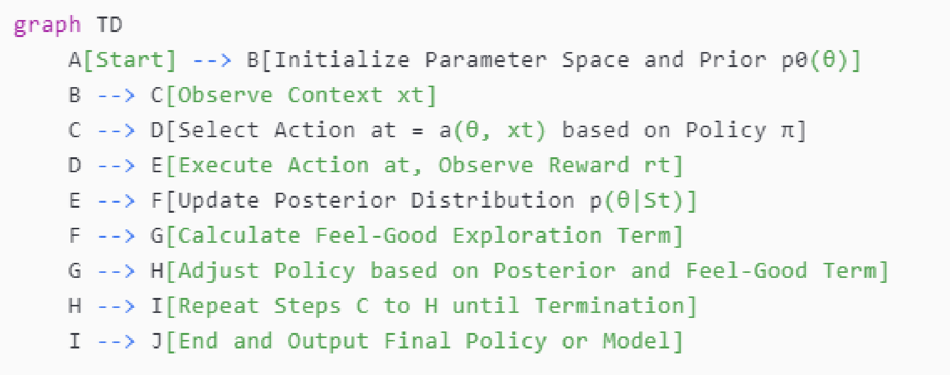
While Meta-TS shows considerable promise, there still exists limitations. Previously, the regret analysis is inherently conservative, relying on a single pull of each arm per task. Additionally, the analysis is currently limited to Gaussian bandits, indicating a need for further research to generalize the approach to other types of bandit problems. For future research, the analysis should be extended beyond Gaussian bandits. This would involve more complex algebra and a deeper understanding of different posterior distributions. Also, there is a need to move beyond the conservative assumption of pulling each arm at least once per task. A less conservative analysis can potentially show greater benefits from meta-learning. Another interest lies in analysing Meta-TS in the context of contextual bandits, where the arms are associated with different contexts or features. These new directions are all worth further exploration.

5. Feel-Good Thompson Sampling for More Aggressive Exploration

5.1. Theoretical Framework of FGTS

Sometimes the exploration strategy of TS is too conservative. To solve the problem, FGTS is proposed with an innovative idea that explores new actions more aggressively, thereby tackling the suboptimality issues in pessimistic scenarios and offering a significant step forward in the field of adaptive online learning. FGTS addresses the conservative nature of standard TS by introducing a theoretical framework that integrates aggressive exploration strategy. This framework utilizes the Decoupling Coefficient, a key concept that quantifies exploration complexity and enables the control of Bellman error within the algorithm. The Decoupling Coefficient allows FGTS to convert the regret analysis into an online least squares estimation problem and measures how aggressively it should explore by determining the minimum number of actions needed to keep the regret in check.

By applying the Decoupling Coefficient, FGTS introduces an optimistic exploration term, Feel-Good, which biases the model towards higher reward predictions based on historical data. This additional term is crucial for achieving better frequentist regret bounds, as it pushes the algorithm to explore more boldly. In essence, FGTS enhances the standard TS by providing a theoretical base for aggressive exploration, ensuring that the algorithm remains effective even when facing challenging, uncertain environments. The pseudocode of a simple flowchart for FGTS is shown in **Figure 4**.



```
graph TD
    A[Start] --> B[Initialize Parameter Space and Prior p0(θ)]
    B --> C[Observe Context xt]
    C --> D[Select Action at = a(θ, xt) based on Policy π]
    D --> E[Execute Action at, Observe Reward rt]
    E --> F[Update Posterior Distribution p(θ|St)]
    F --> G[Calculate Feel-Good Exploration Term]
    G --> H[Adjust Policy based on Posterior and Feel-Good Term]
    H --> I[Repeat Steps C to H until Termination]
    I --> J[End and Output Final Policy or Model]
```

Figure 4. Pseudocode of a simple flowchart for FGTS.

5.2. The Innovation Aspects of FGTS

Traditional TS selects actions by sampling from the posterior distribution, which might not be sufficient to discover optimal actions, especially in complex or high-dimensional action spaces. To address this, FGTS introduces an additional exploration term based on the maximum historical reward predicted by the model. Specifically, the exploration term is given by $-\lambda \min\big(b, f(\theta, x)\big)$, where $\boldsymbol{\lambda}$ is a non-negative tuning parameter that controls the intensity of exploration, $\mathbf{b}$ is a constant defining the boundary of rewards, and $\boldsymbol{f(\theta, x)}$ represents the maximum reward predicted by the model for given parameters $\boldsymbol{\theta}$ and context $\boldsymbol{x}$. This term encourages the algorithm to favour models that have demonstrated higher rewards in the past, thus promoting the exploration of actions that could lead to greater gains. Meanwhile, to make sure that the algorithm does not over-explore to the point of performance degradation, FGTS incorporates a penalty term that restrains the selection of models that deviate significantly from the optimal. This penalization mechanism helps the algorithm to maintain a reasonable balance between venturing into new actions and utilizing information effectively to make well-informed decisions.

5.3. Conclusion and Future Directions

FGTS marks a significant advancement over traditional TS by introducing a more aggressive exploration strategy, resulting in optimal regret bounds for finite action spaces. Theoretical analysis has proven that FGTS achieves a regret bound of $O(\sqrt{KT \ln N})$, substantially enhancing from the previous $\tilde{O}(d^{3/2}\sqrt{T})$, for high-dimensional problems. Despite these gains, FGTS may not be optimal for all structured bandit problems. Future work should focus on integrating structural information to refine regret bounds further, potentially improving performance across a wider range of applications. Future research could also dig into extending FGTS to more complex settings, such as those with random transitions in reinforcement learning, or incorporating multi-agent systems to explore emergent behaviours in dynamic environments.

6. Linear Thompson Sampling under Safety Constraints

6.1. Settings of the Linear Stochastic Bandit Problem

The Linear Stochastic Bandit Problem (LSBP) is a generalization of the Multi-Armed Bandit Problem in reinforcement learning and online optimization. In LSBP, the learner selects an action (or "arm") from a set of available actions at each step (or round), each associated with a feature vector. Upon executing an action, the learner observes a stochastic reward that is the inner product of the action's feature vector with an unknown parameter vector, perturbed by zero-mean noise. The goal is to maximize the total reward over a sequence of rounds, or equivalently, minimize the cumulative regret. LSBP is relevant in applications like online advertising, where each ad is an action, and the feature vector captures ad attributes like size, colour, and position.

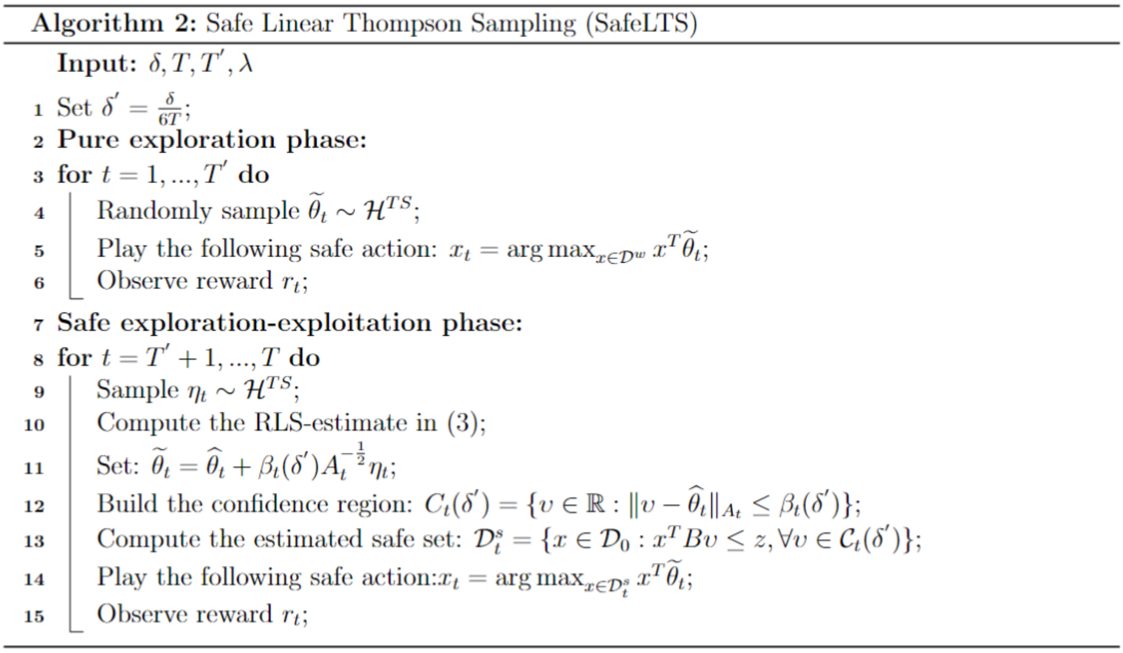
6.2. The Exploration Process and Proposal of Safe-LTS

To find out a proper solution to the LSBP under additional linear safety constraints, Moradipari et al. proposed a novel safe algorithm in 2020, which achieves a regret bound comparable to the standard LTS without safety constraints, demonstrating the advantage of inherent randomness of TS in expanding the set of safe actions. A year later, they researched on the impact of unknown linear safety constraints on the performance of LTS and proposed Safe-LTS algorithm with pure exploration-exploitation phases, ensuring safety constraints met at each round. Safe-LTS is based on the necessity to uphold safety while seeking optimal reward, particularly in environments with stage-wise safety or reliability constraints. It achieves regret bounds comparable to Safe-UCB and demonstrates its effectiveness through numerical simulations. Safe-LTS The algorithm ensures compliance with safety constraints at each round, even amidst uncertainty. Its theoretical underpinning lies in its ability to construct confidence regions that encapsulate unknown parameters with high probability, thereby allowing for safe action selection.

6.3. Algorithm Framework and Analysis of Safe-LTS

The implementation of Safe-LTS basically involves two stages. At the pure exploration phase, it gathers preliminary data by selecting actions from a safe subset based on random parameter sampling. At the safe exploration-exploitation phase, Safe-LTS employs regularized least squares estimates to construct a confidence region that encapsulates the uncertainty around the parameter estimates. Actions are chosen that optimize the sampled parameter's expected reward while remaining within a computed safe set, ensuring that all decisions satisfy the imposed safety constraints. This allows Safe-LTS to balance the acquisition of knowledge with the maintenance of safety, making it well-suited for environments where risk mitigation is crucial.

The core of this algorithm lies in its sophisticated parameter settings and data structures. The choice of $\boldsymbol{\delta}$, $\mathbf{T}$, $\mathbf{T'}$, and $\boldsymbol{\lambda}$ are pivotal [5]: $\boldsymbol{\delta}$ serves as the confidence level, $\mathbf{T}$ denotes the total number of rounds, $\mathbf{T'}$ marks the duration of the pure exploration phase, and $\boldsymbol{\lambda}$ is the regularization parameter that controls the trade-off between exploration and exploitation. The algorithm meticulously maintains the Gram matrix $\mathbf{A_t}$ and the regularized least squares estimate $\boldsymbol{\widetilde{\theta}_t}$, which are instrumental in constructing the confidence region $\boldsymbol{C_t(\delta')}$. The algorithm dynamically computes the safe set $\boldsymbol{D_t^s}$, an inner approximation of the true safe set $\boldsymbol{D_0^s}$, ensuring that all chosen actions $\boldsymbol{x_t}$ are in compliance with the safety constraints. A pseudo-code for Safe-LTS is shown in **Figure 5**.



```
Algorithm 2: Safe Linear Thompson Sampling (SafeLTS)
  Input: $\delta, T, T', \lambda$
Set $\delta' = \frac{\delta}{6T}$;
Pure exploration phase:
for $t = 1, ..., T'$ do
    Randomly sample $\widetilde{\theta}_t \sim \mathcal{H}^{TS}$;
    Play the following safe action: $x_t = \arg\max_{x \in \mathcal{D}^w} x^T \widetilde{\theta}_t$;
    Observe reward $r_t$;
Safe exploration-exploitation phase:
for $t = T' + 1, ..., T$ do
    Sample $\eta_t \sim \mathcal{H}^{TS}$;
    Compute the RLS-estimate in (3);
    Set: $\widetilde{\theta}_t = \widehat{\theta}_t + \beta_t(\delta') A_t^{-\frac{1}{2}} \eta_t$;
    Build the confidence region: $C_t(\delta') = \{v \in \mathbb{R} : \|v - \widehat{\theta}_t\|_{A_t} \leq \beta_t(\delta')\}$;
    Compute the estimated safe set: $\mathcal{D}_t^s = \{x \in \mathcal{D}_0 : x^T B v \leq z, \forall v \in \mathcal{C}_t(\delta')\}$;
    Play the following safe action:$x_t = \arg\max_{x \in \mathcal{D}_t^s} x^T \widetilde{\theta}_t$;
    Observe reward $r_t$;
```

Figure 5. Pseudocode for Safe-LTS.

Specifically, at each round $\mathrm{t} = \mathrm{T}' + 1, \dots, \mathrm{T}$ of the safe exploration-exploitation phase, Safe-LTS uses the previous action-observation pairs to compute the Gram matrix At and the RLS-estimate $\tilde{\theta}_t$ of θ_* defined as follows:

$$A_t = \lambda I + \sum_{s=1}^{t-1} x_s x_s^T \,, \hat{\theta}_t = A_t^{-1} \sum_{s=1}^{t-1} r_s x_s. \tag{3}$$

6.4. Limitations of Safe-LTS and Future Directions

While Safe-LTS demonstrates promising results in addressing the linear stochastic bandit problem with safety constraints, it does have limitations. Thise is because the theoretical guarantees provided are predicated on certain assumptions that may not hold in all practical scenarios. For example, the reliance on the sub-Gaussian noise assumption for the reward and safety constraint functions is crucial in our previous discussion. But in real-world applications, noise distributions may not strictly adhere to this assumption, potentially affecting the algorithm's regret guarantees. Besides, the algorithm's efficiency decreases with the dimensionality of the action space, as the computation of the confidence regions and the safe set becomes more complex. Also, the algorithm's performance hinges on the accurate estimation of the unknown parameter θ* and the construction of the safe set, which can be challenging in complex, high-dimensional environments.

Future research on Safe-LTS should delve into enhancing the algorithm's adaptability to dynamic environments where safety constraints may evolve over time. This includes developing methods that allow for the algorithm to efficiently respond to changes in the constraint landscape. Besides, there is a need to extend the theoretical framework to accommodate non-sub-Gaussian noise characteristics, which are more reflective of real-world scenarios. Investigating the scalability of Safe-LTS to high-dimensional action spaces is crucial, potentially through the incorporation of dimensionality reduction techniques or optimized confidence region computations.

7. Combining Technologies with Practical Applications

The convergence of diverse TS variants has opened new avenues for tackling complex decision-making challenges across various domains, as is shown in **Table 1**. Each variant, with its distinctive features, brings a unique set of advantages to real-time applications.

Table 1. A comparison between some of the innovative algorithms.

	NeuralTS	Meta-TS	FGTS	SW-TS	Safe-LTS
Core Concept	Deep learning integration	Meta-learning adaptation	Aggressive exploration strategy	Sliding window adaptation	Safety constraint integration
Application	Recommendation systems, personalized medicine	Multi-task learning, rapid adaptation to new problems	Exploration in pessimistic scenarios	Financial trading, online advertising	Autonomous driving, robotics
Advantages	Approximation of high-dimensional reward functions	Learning and adaptation across tasks	Encourages exploration of high-reward actions	Responsive to recent changes in the environment	Ensures decisions adhere to safety margins
Theoretical Guarantees	Cumulative regret of $O(T^{1/2})$	Regret analysis based on meta-learning	Theoretical regret with optimistic bias	Regret analysis in non-stationary environments	Principled safety decision-making
Computational Efficiency	High, but requires optimization	Depends on meta-model complexity	High, due to simple exploration strategy	Moderate, maintains sliding window	Low, calculates confidence regions
Empirical Performance	Competitive with state-of-the-art benchmarks	Quick adaptation to new tasks	Superior to standard TS in specific scenarios	Robust to reward signal delays	Reliable performance in safety-critical applications
Future Directions	Enhance computational	Extend to more	Fine-tune balance	Explore sophisticated	Develop efficient

Table 1. (continued).

	efficiency of deep learning components	complex bandit problems	between exploration and exploitation	windowing techniques	methods for updating confidence regions
Remarks	Requires extensive data for training neural networks	Needs sufficient task diversity	Parameter tuning is crucial for maximizing benefits	Sensitive to window size and shape	Sensitive to complexity of safety constraints

When deep learning is seamlessly integrated with Thompson Sampling, NeuralTS is proposed. Its ability to approximate complex reward landscapes with neural networks provides a powerful tool for both exploration and exploitation, making it a strong candidate for scenarios where the reward function is intricate and not easily modelled with traditional approaches. That's why NeuralTS excels at handling high-dimensional reward functions that are prevalent in modern applications such as recommendation systems and personalized medicine. Future research can involve enhancing the computational efficiency of its deep learning components and broadening its applicability to a wider range of problems.

When it comes to learning from a series of tasks and adapting its strategy to new, unseen problems, Meta-TS stands out for its meta-learning approach. This is particularly useful in multi-task environments where the ability to quickly adapt to new situations can significantly improve performance. However, the meta-learning paradigm of Meta-TS could still benefit from a more comprehensive theoretical analysis that could extend its reach to more complex bandit problems.

For pessimistic scenarios where traditional TS might be too conservative, FGTS introduces an aggressive exploration strategy. By incorporating an optimistic bias, FGTS encourages exploration of high-reward actions that could potentially break out of suboptimal local maxima. Future research could focus on fine-tuning the balance between exploration and exploitation to maximize the benefits of its aggressive strategy. Applying sliding-window technique, SW-TS is enabled to stay responsive to recent changes while still leveraging historical data, demonstrating adaptability to non-stationary environments, making it an excellent choice for applications where the underlying reward distributions may change over time, such as in financial trading or online advertising. It might benefit from further integration with other MAB algorithms, potentially leading to hybrid methods that can tackle an even broader spectrum of non-stationary environments.

To ensure safety in critical applications such as autonomous driving and robotics, Safe-LTS emerges as a result. By constructing confidence regions that encapsulate the uncertainty in parameter estimates, Safe-LTS provides a principled way to make decisions that adhere to safety margins. Future work could involve designing more efficient methods for updating confidence regions and exploring its applicability to problems with complex safety constraints.

The collective integration of these TS variants has already demonstrated improved performance in balancing the delicate trade-off between exploration and exploitation, effectively dealing with delayed feedback, and adapting to the ever-changing dynamics of real-world environments.

In summary, the fusion of TS variants with cutting-edge technologies such as deep learning, meta-learning, and safety constraints has propelled the field of sequential decision-making to new heights. As we continue to push the boundaries of what these algorithms can achieve, we pave the way for innovative solutions to some of the most pressing challenges in artificial intelligence, machine learning, and beyond. Future research in this domain should focus on enhancing the universality, efficiency, stability, and scalability of these algorithms, ensuring they are well-prepared to handle the diverse and intricate decision-making challenges of the real world.

8. Conclusion

This review paper has provided an extensive survey of TS and its variants, highlighting their evolution and application. It has systematically reviewed the literature, examined the theoretical foundations, algorithmic frameworks, and innovations of these TS variants, and offered insights into future research directions. In particular, it discusses NeuralTS, Meta-TS, FGTS, and Safe-LTS. These variants have been tailored to address specific challenges, including high-dimensional reward functions, unknown prior distributions, pessimistic scenarios, and safety constraints. Those algorithms all have unique innovative aspects. More specifically, NeuralTS integrates deep learning to handle complex reward landscapes, Meta-TS leverages meta-learning for rapid adaptation to new tasks, FGTS applies an aggressive exploration strategy to overcome suboptimality in pessimistic situations, and Safe-LTS ensures decisions adhere to safety margins, particularly in critical applications. These algorithms have demonstrated effectiveness in both theory and practice, with empirical validations showcasing their potential across various problem domains. Despite they all have promising performance in specific contexts, this survey talks about their limitations that warrant future research. For instance, NeuralTS faces high computational complexity, Meta-TS relies on conservative assumptions, FGTS requires careful parameter tuning, and Safe-LTS may be less efficient in high-dimensional action spaces. Addressing these limitations will enhance the universality, efficiency, and stability of these algorithms. All in all, this review paper has provided a comprehensive perspective on TS and its variants and outlined clear directions for future research. Hopefully, these outcomes will advance the field of decision theory and contribute significantly to practical applications.

References

[1] Phan M, Abbasi-Yadkori Y, Domke J, 2019 Thompson sampling and approximate inference (in Advances in Neural Information Processing Systems) vol 32.

[2] Zhang W, Zhou D, Li L, Gu Q, 2020 Neural thompson sampling preprint 2010.00827.

[3] Trovo F, Paladino S, Restelli M, Gatti N, 2020 Sliding-window thompson sampling for non-stationary settings (Journal of Artificial Intelligence Research) vol 68 pp 311–364.

[4] Moradipari A, Alizadeh M, Thrampoulidis C, 2020 Linear thompson sampling under unknown linear constraints (in 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) IEEE) pp 3392–3396.

[5] Moradipari A, Amani S, Alizadeh M, Thrampoulidis C, 2021 Safe linear thompson sampling with side information (IEEE Transactions on Signal Processing) vol 69 pp 3755–3767.

[6] Vernade C, Carpentier A, Lattimore T, Zappella G, Ermis B, Brueckner M, 2020 Linear bandits with stochastic delayed feedback (in International Conference on Machine Learning PMLR) pp 9712–9721.

[7] Hamidi N, Bayati M, 2020 On frequentist regret of linear thompson sampling preprint 2006.06790.

[8] Kveton B, Konobeev M, Zaheer M, Hsu C W, Mladenov M, Boutilier C, Szepesvari C, 2021 Meta-thompson sampling (in International Conference on Machine Learning PMLR) pp 5884–5893.

[9] Zhang T, 2022 Feel-good thompson sampling for contextual bandits and reinforcement learning (SIAM Journal on Mathematics of Data Science) vol 4 no 2 pp 834–857.

[10] Gigli M, 2024 Thompson sampling for Performance Marketing and delayed conversions.

[11] Zheng H, Deng W, Moya C, Lin G, 2024 Accelerating approximate thompson sampling with underdamped langevin monte carlo (in International Conference on Artificial Intelligence and Statistics PMLR) pp 2611–2619.