Non-Asymptotic Analysis of Monte Carlo Tree Search

Non-Asymptotic Analysis of Monte Carlo Tree Search

Devavrat Shah
devavrat@mit.edu
LIDS, MIT

Qiaomin Xie
qiaomin.xie@cornell.edu
ORIE, Cornell University

Zhi Xu
zhixu@mit.edu
LIDS, MIT

ABSTRACT

In this work, we consider the popular tree-based search strategy within the framework of reinforcement learning, the Monte Carlo Tree Search (MCTS), in the context of infinite-horizon discounted cost Markov Decision Process (MDP) with deterministic transitions. While MCTS is believed to provide an approximate value function for a given state with enough simulations, cf. [5, 6], the claimed proof of this property is incomplete. This is due to the fact that the variant of MCTS, the Upper Confidence Bound for Trees (UCT), analyzed in prior works utilizes "logarithmic" bonus term for balancing exploration and exploitation within the tree-based search, following the insights from stochastic multi-arm bandit (MAB) literature, cf. [1, 3]. In effect, such an approach assumes that the regret of the underlying recursively dependent non-stationary MABs concentrates around their mean exponentially in the number of steps, which is unlikely to hold as pointed out in [2], even for stationary MABs.

As the key contribution of this work, we establish polynomial concentration property of regret for a class of *non-stationary* multi-arm bandits. This in turn establishes that the MCTS with appropriate *polynomial* rather than *logarithmic* bonus term in UCB has the claimed property of [5, 6]. Interestingly enough, empirically successful approaches (cf. [10]) utilize a similar polynomial form of MCTS as suggested by our result. Using this as a building block, we argue that MCTS, combined with nearest neighbor supervised learning, acts as a "policy improvement" operator, i.e., it iteratively improves value function approximation for *all* states, due to combining with supervised learning, despite evaluating at only finitely many states. In effect, we establish that to learn an ε-approximation of the value function for deterministic MDPs with respect to ℓ_∞ norm, MCTS combined with nearest neighbor requires a sample size scaling as $\widetilde{O}\big(\varepsilon^{-(d+4)}\big)$, where d is the dimension of the state space. This is nearly optimal due to a minimax lower bound of $\widetilde{\Omega}\big(\varepsilon^{-(d+2)}\big)$ [8] suggesting the strength of the variant of MCTS we propose here and our resulting analysis. [1]

ACM Reference Format:
Devavrat Shah, Qiaomin Xie, and Zhi Xu. 2020. Non-Asymptotic Analysis of Monte Carlo Tree Search. In *ACM SIGMETRICS / International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS '20 Abstracts), June 8–12, 2020, Boston, MA, USA.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3393691.3394202

[1] Extended Abstract. The full paper can be found at [9].

SIGMETRICS '20 Abstracts, June 8–12, 2020, Boston, MA, USA

ACM ISBN 978-1-4503-7985-4/20/06.
https://doi.org/10.1145/3393691.3394202

Introduction. Monte Carlo Tree Search (MCTS) is a search framework for finding optimal decisions, based on the search tree built by random sampling of the decision space [4]. Recently, MCTS has been combined with deep neural networks for reinforcement learning, achieving remarkable success for games of Go in AlphaGo Zero [10]. However, despite the wide application and empirical success of MCTS, there is only limited work on theoretical guarantees of MCTS and its variants. A notable exception is the work of [5] and [6], which propose running tree search by applying the Upper Confidence Bound algorithm — originally designed for stochastic multi-arm bandit (MAB) problems [1, 3] — to each node of the tree. This leads to the so-called UCT (Upper Confidence Bounds for Trees) algorithm, which is one of the popular forms of MCTS. In [5], certain asymptotic optimality property of UCT is claimed. The proof therein is, however, incomplete. More importantly, UCT as suggested in [5] requires exponential concentration of regret for the underlying non-stationary MAB. Such exponential concentration of regret, however, is unlikely to hold in general even for stationary MAB as pointed out in [2].

Indeed, rigorous analysis of MCTS is subtle, even though its asymptotic convergence may seem natural. A key challenge is that the tree policy (e.g., UCT) for selecting actions typically needs to balance exploration and exploitation, so the action selection process at each node is non-stationary (non-uniform) across multiple simulations. A more severe difficulty arises due to the hierarchical/iterative structure of tree search, which induces complicated probabilistic dependency between a node and the nodes within its sub-tree. Specifically, as part of simulation within MCTS, at each intermediate node (or state), the action is chosen based on the outcomes of the past simulation steps within the sub-tree of the node in consideration. Such strong dependencies across time (i.e., depending on the history) and space (i.e., depending on the sub-trees downstream) among nodes makes the analysis non-trivial. The goal of this paper is to address this challenge and provide a rigorous theoretical foundation for MCTS. In particular, we are interested in the following:

- What is the appropriate form of MCTS for which the asymptotic convergence property claimed in the literature (cf. [5, 6]) holds?
- Can we rigorously establish the "strong policy improvement" property of MCTS when combined with supervised learning as observed in the literature (e.g., in [10])? If yes, what is the quantitative form of it?
- Does supervised learning combined with MCTS lead to the optimal policy, asymptotically? If so, what is its finite-sample (non-asymptotic) performance?

As the main contribution of this work, we provide affirmative answers to all of the above questions.

Non-stationary MAB and recursive polynomial concentration. In stochastic Multi Arm Bandit (MAB), the goal is to discover the action (arm) with the best average reward while choosing as few non-optimal actions as possible in the process. The rewards for any given action is assumed to be i.i.d., leading to the UCB algorithm: at any time $t \geq 1$, an action with maximal index is chosen where the index of an action is the empirical average reward observed for the action plus a *logarithmic* bonus term $B_{t,s}$ that scales as $\sqrt{\log t/s}$, for an action that has been picked $s \leq t$ times. As mentioned, Monte Carlo Tree Search (MCTS) has a similar goal where reward depends on the future actions. To take future actions into consideration, MCTS effectively expands all possible future actions recursively in the form of (decision-like) tree. As such, determining the optimal future path corresponding to maximal reward starting at the root node of the MCTS tree requires solving multiple MABs, one per each intermediate node within the tree. Apart from the MABs associated with the leaf layer of the tree, all the MABs associated with the intermediate nodes turn out to have rewards that are generated by MAB algorithms for nodes downstream. This creates complicated, hierarchically inter-dependent MABs.

To determine the appropriate, UCB-like algorithm for MAB corresponding to each node of the MCTS tree, it is essential to understand the concentration property of rewards, i.e., concentration of regret for MABs associated with the nodes downstream. While the rewards at leaf level may enjoy exponential concentration due to independence, the regret of any algorithm even for such an MAB is unlikely to have exponential concentration in general, cf. [2, 7]. Further, the MAB of our interest has non-stationary rewards due to strong dependence across hierarchy. Indeed, an oversight of this complication led [5, 6] to suggest UCT inspired by the standard UCB algorithm for MABs with stationary, independent rewards.

As an important contribution of this work, we formulate an appropriate form of non-stationary MAB which correctly models the MAB at each node within the tree. In particular, assuming that the rewards, though non-stationary, satisfy certain *polynomial* concentration. Then, we establish that under the UCB algorithm that chooses the arm with highest index, where index is defined as the empirical reward plus an appropriate *polynomial* (and *not logarithmic*) bonus term, a similar form of *polynomial* concentration holds for the induced regret. In particular, let $\bar{X}_t$ denote the empirical average of the rewards collected at a given node over t visits of the node. Then, under the UCB algorithm with a bonus term scaling as $t^{\eta(1-\eta)}/s^{1-\eta}$, where $1/2 \leq \eta < 1$, we establish that (a) $\bar{X}_t$ converges to the optimal mean reward obtained by choosing the right action, and (b) $\bar{X}_t$ satisfy a polynomial concentration inequality around the optimal mean reward, i.e., the convergence rate is polynomial.

Corrected UCT for MCTS and non-asymptotic analysis. As desired, the non-stationary MAB enjoys a recursive polynomial concentration: starting from polynomially concentrated arm rewards, the proper UCB algorithm leads to a polynomially concentrated empirical reward. Hence, we immediately obtain that we can recursively define the UCB algorithm at each level in MCTS, starting from the leaf level, with appropriately chosen polynomial bonus terms $B_{t,s}$. In effect, setting $\eta = 1/2$, we obtain modified UCT where $B_{t,s}$ scales as $t^{1/4}/s^{1/2}$. This is in contrast to the $\sqrt{\log t/s}$ scaling in the standard UCB as well as UCT suggested in the literature [5, 6].

By recursively applying the convergence and concentration property of the non-stationary MAB for the resulting algorithm for MCTS, we establish that for any query state s of a MDP with deterministic transitions, using a total of n simulations of the MCTS, we can obtain a value function estimation within error $\delta\varepsilon_0 + O\big(n^{-1/2}\big)$ for some $\delta < 1$ (independent of n but dependent on the depth of MCTS tree), if we start with a value function estimation for all the leaf nodes within error ε_0 . That is, MCTS is indeed asymptotically correct as was conjectured in the prior literature.

MCTS with supervised learning, strong policy improvement, and near optimality. The result stated above for MCTS implies its "bootstrapping" property – if we start with a value function estimation for *all* state within error ε, then MCTS can produce estimation of value function for a *given query* state within error less than ε with enough simulations. By coupling such improved estimations for a number of query states, combined with expressive enough supervised learning, one can hope to generalize such improved estimations of value function for *all* states. That is, MCTS coupled with supervised learning can be "strong policy improvement operator".

Indeed, this is precisely what we establish by utilizing nearest neighbor supervised learning. Specifically, we establish that with total of $\widetilde{O}\big(\frac{1}{\varepsilon^{4+d}}\big)$ number of samples, MCTS with nearest neighbor finds an ε-approximation of the optimal value function for deterministic MDPs with respect to ℓ_∞-norm; here d is the dimension of the state space. This is nearly optimal in view of a minimax lower bound of $\widetilde{\Omega}\big(\frac{1}{\varepsilon^{2+d}}\big)$ [8].

An Implication. As mentioned earlier, the modified UCT policy per our result suggests using bonus term $B_{t,s}$ that scales as $t^{1/4}/s^{1/2}$ at each node within the MCTS. Interestingly enough, the empirical results of AlphaGo Zero [10] are obtained by utilizing $B_{t,s}$ that scales as $t^{1/2}/s$. This is qualitatively similar to what our result suggests and in contrast to the classical UCT.

REFERENCES

[1] Rajeev Agrawal. 1995. Sample mean based index policies by o (log n) regret for the multi-armed bandit problem. *Advances in Applied Probability* 27, 4 (1995), 1054–1078.

[2] Jean-Yves Audibert, Rémi Munos, and Csaba Szepesvári. 2009. Exploration–exploitation tradeoff using variance estimates in multi-armed bandits. *Theoretical Computer Science* 410, 19 (2009), 1876–1902.

[3] Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer. 2002. Finite-time analysis of the multiarmed bandit problem. *Machine learning* 47, 2-3 (2002), 235–256.

[4] Cameron B. Browne, Edward Powley, Daniel Whitehouse, Simon M. Lucas, Peter I. Cowling, Philipp Rohlfshagen, Stephen Tavener, Diego Perez, Spyridon Samothrakis, and Simon Colton. 2012. A survey of monte carlo tree search methods. *IEEE Transactions on Computational Intelligence and AI in games* 4, 1 (2012), 1–43.

[5] Levente Kocsis and Csaba Szepesvári. 2006. Bandit based monte-carlo planning. In *European conference on machine learning*. Springer, 282–293.

[6] Levente Kocsis, Csaba Szepesvári, and Jan Willemson. 2006. Improved monte-carlo search. *Univ. Tartu, Estonia, Tech. Rep* (2006).

[7] Antoine Salomon and Jean-Yves Audibert. 2011. Deviations of stochastic bandit regret. In *International Conference on Algorithmic Learning Theory*. Springer, 159–173.

[8] Devavrat Shah and Qiaomin Xie. 2018. Q-learning with Nearest Neighbors. In *Advances in Neural Information Processing Systems 31*. 3115–3125.

[9] Devavrat Shah, Qiaomin Xie, and Zhi Xu. 2020. Non-Asymptotic Analysis of Monte Carlo Tree Search. *preprint* (2020). https://sites.coecis.cornell.edu/qiaominxie/files/2020/01/sxz_sigm20.pdf

[10] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. 2017. Mastering the game of go without human knowledge. *Nature* 550, 7676 (2017), 354.