In Proc. IJCAI 2021, pp. 4946--4950, IJCAI, 2021.

Intelligent and Learning Agents: Four Investigations

Shivaram Kalyanakrishnan
Department of Computer Science and Engineering, Indian Institute of Technology Bombay
shivaram@cse.iitb.ac.in

Abstract

My research is driven by my curiosity about the nature of *intelligence*. Of the several aspects that characterise the behaviour of intelligent agents, I primarily study sequential decision making, learning, and exploration. My interests also extend to broader questions on the effects of AI on life and society. In this paper, I present four distinct investigations drawn from my recent work, which range from theoretical to applied, and which involve both analysis and design. I also share my outlook as an early-career researcher.

1 Introduction

This paper serves as my response to the conference's kind invitation to share my experiences as an early-career researcher. The context impels me to preface the technical portion (sections 2–5) with a few words on how the projects described in the paper came to into being. I especially hope this account will benefit young researchers.

It is tempting to somehow unify the various threads of one's work and present them as a well-knit, cohesive body (preferably with a short name), which is carried forward purposefully and surefootedly. In *reality*, my research is a loose federation of investigations, conceived under varying circumstances and constrained by time, expertise, and resources. If I had to pick *one* scientific theme that most influences my research, it would be the very broad question of how "intelligence" works. I became acquainted with a particular line of inquiry related to this question during my Ph.D., which was on the topic of reinforcement learning. The investigations I proceed to describe were all undertaken subsequently.

A conversation with a probing theoretician made me realise how little I knew about *computational* aspects involving MDPs, which are a standard abstraction used in reinforcement learning. Section 2 summarises a resulting line of work, which is entirely theoretical. The idea outlined in Section 3 is from the thesis of my student Arghya Roy Chaudhuri, who proposes a practical approach to tackle exploration problems with a very large (even infinite) number of choices. The projects described in sections 4 and 5 are both outcomes of my interest in the *application* of AI. Section 4 presents an online scheduling solution for India's large railway network. In Section 5, I summarise a paper that maps out unique opportunities and challenges for AI in India. I conclude in Section 6 by sharing my outlook as an early-career researcher.

2 Complexity of Policy Iteration

Markov Decision Problems (MDPs) are a well-studied, widely-used abstraction of sequential decision making in stochastic environments. A common definition takes an MDP as a tuple (S, A, R, T, γ), where S is a set of states in which an agent can be, and A the set of actions it can take. The reward function R assigns an immediate numeric reward for taking action $a \in A$ from state $s \in S$. The transition function T gives the probability of reaching state $s' \in S$, conditioned on (s, a). Consider an agent that interacts with the MDP over time $t \geq 0$, starting from state s^0 at $t = 0$. If the agent acts according to a *policy* $\pi : S \to A$, which specifies the action to take from each state, it traverses a *random* "state-action-reward" trajectory $s^0, a^0, r^0, s^1, a^1, r^1, s^2, \ldots$, wherein $a^t = \pi(s^t)$, $r^t = R(s^t, a^t)$, and $s^{t+1} \sim T(s^t, a^t)$ for $t \geq 0$. The essence of *sequential* decision making is to take actions that strike the right balance between immediate and future rewards. In one standard formulation, the *value* of each state $s \in S$ under policy π is given by

$$V^\pi(s) = \mathbb{E}_\pi[r^0 + \gamma r^1 + \gamma^2 r^2 + \ldots | s^0 = s],$$

where $\gamma \in [0, 1)$ is a discount factor. It is a well-known result that every MDP has an *optimal policy* $\pi^\star : S \to A$ such that for every policy π and state $s \in S$, $V^{\pi^\star}(s) \geq V^\pi(s)$. MDP planning is the problem of computing an optimal policy for a given MDP (S, A, R, T, γ).

Policy Iteration (PI) is one of the three main algorithmic approaches to MDP planning, the other two being value iteration (VI) and linear programming (LP) [Littman *et al.*, 1995]. Given an input MDP, PI is initialised with an arbitrary policy π_0. For illustration consider an MDP with $S = \{s_1, s_2, s_3\}$, and $A = \{1, 2, 3, 4\}$. Suppose $\pi_0(s_1) = 2, \pi_0(s_2) = 1, \pi_0(s_3) = 3$, as depicted in Figure 1. The first step in PI, called *policy evaluation*, is the computation of V^{π_0}. The second step, *policy improvement*, thereafter identifies a set of "improving" actions for each state. If π_0 is an optimal policy, it is guaranteed to have no improving actions for any state; otherwise at least one state is sure to have a non-empty set of improving actions. In our illustration, s_1 has two improving actions, s_2 none, and s_3 one. Now, consider any

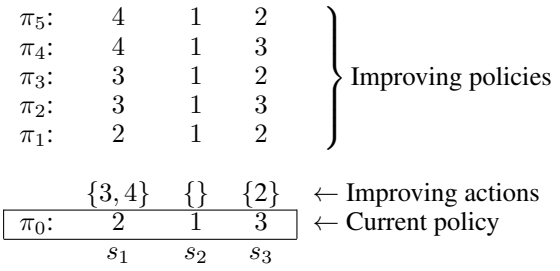

Figure 1: Illustration of the policy improvement step in PI on an MDP with three states and four actions. Explanations are in the text.

arbitrary policy $\pi \neq \pi_0$ that differs from π_0 only on states that have improving actions, and moreover, on the states it does differ, π takes some improving action of π_0. An important result called the policy improvement theorem establishes that for any such policy π, $V^{\pi}(s) \geq V^{\pi_0}(s)$ for $s \in S$, and there is at least one state $\bar{s} \in S$ such that $V^{\pi}(\bar{s}) > V^{\pi_0}(\bar{s})$. In essence, π strictly dominates, or *improves*, π_0. In our example from Figure 1, π_0 has five improving policies, π_1–π_5.

The policy evaluation and improvement steps to go from π_0 to an improving policy π can now be repeated from π to obtain an improving policy π'—and this iterative process continued until an optimal policy is found. On an MDP with $n \geq 2$ states and $k \geq 2$ actions, each iteration can be completed using poly(n, k) arithmetic operations. The total number of iterations is trivially upper-bounded by k^n, which is the number of policies. Does PI, which is typically very efficient in practice, enjoy *tighter* upper bounds in terms of n and k?

Algorithms from the PI family vary in their *switching rule*, which determines the improving policy selected when a choice is available. The most common variant of PI is Howard's (or greedy) PI [Howard, 1960], in which every state with improving actions is switched (the choice made among improving actions is arbitrary). In our illustration, Howard's PI would proceed from π_0 to either π_3 or π_5. Mansour and Singh [1999] show that Howard's PI takes at most $O(k^n/n)$ iterations to terminate. They also devise a randomised variant of PI that is shown to take at most $O((k/2)^n)$ expected iterations for large k, and $O(1.7172^n)$ expected iterations for $k = 2$. In their algorithm, the iterate following π_0 is one among π_1–π_5, picked at random.

Our investigation into the complexity of PI has thrown more light on the structure of the policy space in MDPs, which we exploit to furnish tighter running-time bounds. For instance, we describe a deterministic variant of PI that takes at most $O(k^{0.7019n})$ iterations for $k \geq 2$ [Kalyanakrishnan *et al.*, 2016a; Taraviya and Kalyanakrishnan, 2019], as well as separate randomised variants that take $O(2 + \ln(k-1))^n$ expected iterations for $k > 2$ [Kalyanakrishnan *et al.*, 2016b] and $O(1.6001^n)$ expected iterations for $k = 2$ [Taraviya and Kalyanakrishnan, 2019]. We also show a bound of $(O(\sqrt{k \log k})^n)$ expected iterations for a randomised variant of Howard's PI [Taraviya and Kalyanakrishnan, 2019].

Although the upper bounds given above for PI are the tightest ones known currently in their respective categories, they remain substantially separated from known *lower* bounds. In fact, a significant outcome of our body of work is the many *open problems* it has unearthed. Most significant among these would be to make any progress on tightening the upper bound for Howard's PI when $k = 2$. The current best upper bound remains the $O(2^n/n)$ iterations shown by Mansour and Singh [1999], whereas the tightest known lower bound is only $\Omega(n)$ iterations [Hansen and Zwick, 2010]. Based on trends observed in small MDPs, we conjecture that $O(\phi^n)$ is a valid upper bound, where $\phi = \frac{1+\sqrt{5}}{2} < 1.6181$ [Kalyanakrishnan *et al.*, 2016a]. From a lower-bounding perspective, it remains unknown whether there exist MDPs with $k > 2$ actions in which some PI variant can visit all k^n policies. The answer is affirmative for $k = 2$ [Melekopoglou and Condon, 1994]. Our lower bound of $\Omega(\sqrt{k}^n)$ iterations [Ashutosh *et al.*, 2020] is the tightest lower bound known yet for $k \geq 3$.

3 Quantile-regret Minimisation in Bandits

Suppose I have $K \geq 2$ coins with me, each having some fixed, unknown *bias* (the probability of showing up heads when tossed). Imagine that I invite you to pick and toss coins from this collection T times, $T \geq 1$. At every step you can choose any of the K coins, using the outcomes of the previous tosses to inform your choice. Now, if I promise to pay you as many Rupees as the number of heads you obtain from your T tosses, how would you go about your task?

Since your knowledge about the biases can only come from tossing the coins, you will have to *explore* sufficiently to gain confidence in your estimates. At the same time, since your payment depends on the overall number of heads, you must also *exploit* seemingly optimal coins, identified based on their empirical performance. This need to balance between exploration and exploitation is usually formalised in the language of stochastic multi-armed bandits [Robbins, 1952], wherein each arm corresponds to a coin, and a *pull* of the arm reveals a probabilistic reward (in our case a Rupee for a head and none for a tail). Applications include sequential drug testing (arm $\equiv$ drug, mean reward $\equiv$ efficacy), on-line advertising (arm $\equiv$ ad, mean reward $\equiv$ click-through rate), and many others.

The most widely-used objective for bandit problems is that of minimising *regret*. If you somehow knew the mean rewards of each arm *a priori*, it is clear that you will maximise your expected payment by always pulling an arm with the largest mean reward. If this quantity is $p^{\star}$, then the expected reward from T pulls is $Tp^{\star}$, something no strategy can better. In reality, since the mean rewards are unknown, some amount of payoff is sacrificed on account of having to explore, and the expected aggregate reward will be $f(T) \leq Tp^{\star}$. The *regret* of a strategy is defined to be the difference $Tp^{\star} - f(T)$. It is well known that any reasonable sampling algorithm must incur at least $\Omega(K \log T)$ regret [Lai and Robbins, 1985]. Many algorithms indeed achieve $O(K \log T)$ regret [Thompson, 1933; Auer *et al.*, 2002].

Unfortunately, even optimal regret-minimisation algorithms can be impractical if the number of arms K is itself large. In applications such as on-line advertising and molecular drug design, K can easily be a few millions—so large that most arms cannot be pulled even once! The most common workaround is to look for some additional information

about the set of arms. Some authors assume that the arms can be embedded in a metric space, in which the mean rewards vary continuously [Kleinberg, 2005]. Others associate feature vectors with arms, assuming that mean rewards come from a linear combination [Dani *et al.*, 2008]. These approaches tend to fail when associated hyperparameters (say related to Lipschitz constants) are not tuned well, or when underlying assumptions (such as the linearity of rewards) do not hold.

We propose a conceptually simple alternative to regret minimisation in problems where K is large [Roy Chaudhuri and Kalyanakrishnan, 2018]. Rather than compare an algorithm's performance with that of always pulling the *best* arm, we compare it with obtaining the $(1-\rho)$-th quantile of the distribution of the arm's mean rewards, for given $\rho \in (0,1)$. For example, if $\rho = 0.1$, we desire an algorithm that can perform on par with the arm at the 90^{th} percentile. This concession is justified in many practical applications, wherein the mean reward distribution is not especially skewed at the higher quantiles. Significantly, the revised objective, called quantile-regret minimisation or ρ-regret minimisation, opens the door to a new algorithmic strategy.

Whereas (regular) regret-minimisation requires each arm to be pulled at least once, for ρ-regret minimisation, it suffices to first select a smaller number of arms at *random*, and to restrict pulls to this subset. The principle behind the approach is that for any $\delta \in (0,1)$, a subset of size $\Theta((1/\rho)\log(1/\delta))$ selected uniformly at random will contain an arm from the top ρ fraction with probability $1-\delta$, *independent* of the total number of arms K. Thus, one can achieve the same ρ-regret growth rate across all K (which can even be infinite). We provide algorithms and upper bounds on the ρ-regret that are sub-linear in T [Roy Chaudhuri and Kalyanakrishnan, 2018], in fact even when the randomly-chosen subset can only be constant-sized [Roy Chaudhuri and Kalyanakrishnan, 2020]. Empirical evaluations demonstrate both qualitative advantages and performance gains over regret-minimisation strategies when K is large.

4 Policy Search for Railway Rescheduling

The Indian railway network annually carries the most passengers in the world, in addition to over a billion tonnes of freight. However, its "track-length per passenger" is only a fraction of those of networks in the United States and China. The large carrying capacity coupled with limited infrastructure results in significant challenges in *scheduling*.

We propose an approach based on policy search to the railway "rescheduling" problem [Prasad *et al.*, 2020]. The two inputs to the rescheduling problem are the infrastructure of the network and a reference timetable. The infrastructure comprises a list of the stations, intermediate segments, their connectivity, track lengths, and so on. A reference timetable has an entry for each train, giving its desired arrival and departure times at each station, travel times between stations, and such. The main catch in implementing the reference timetable is that the infrastructure might not support it. For illustration, consider a station from which the reference timetable requires two trains to depart at the same time, in the same direction. If there is only a single track leading out in the intended direction, at least one of the trains must necessarily wait. The objective of rescheduling is to generate a revised timetable that (1) respects all the infrastructural constraints, while (2) minimising the aggregate, priority-weighted departure delay, abbreviated PWDD, which is calculated relative to the reference timetable. Since rescheduling algorithms are run whenever there are unexpected delays in operations, they must be able to compute schedules within a few minutes.

Whereas the rescheduling task can be formulated accurately as an MDP, its extremely large state space (exponential in the number of resources) rules out exact solution techniques. In recent work, Khadilkar [2019] demonstrates reasonable success with an approximate form of Q-learning, which outperforms several heuristic approaches common in the transportation literature [Khadilkar, 2017]. We propose a policy search method as an alternative to Q-learning, and show significant gains in PWDD [Prasad *et al.*, 2020].

Demands on computation and memory constrain solutions to use only localised state representations, which view each train in isolation, rather than describe the full cross product. Unfortunately, value function-based reinforcement learning methods such as Q-learning are dependent on the Markov property, which is only weakly satisfied by such representations. By contrast, policy search approaches (such as hill climbing and evolutionary algorithms) bypass value function learning altogether, instead aiming to directly optimise the desired objective function (here PWDD for the entire network). An additional advantage of policy search methods is that they can readily accommodate domain knowledge. For example, we embed a manually-designed rule for deadlock-avoidance in our scheduler, and only optimise the control that follows downstream. The component we optimise takes as input the local features of a given train, and determines whether the train must be moved or not. This mapping is implemented using a neural network, whose few hundreds of weights are the policy parameters optimised.

Table 1 compares the PWDD obtained by our policy search method (using CMA-ES [Hansen, 2016]) with that of the Q-learning approach of Khadilkar [2019]. Results are averages from 100 "randomly perturbed" timetables [Prasad *et al.*, 2020]. Observe the consistent gains in each of two synthetic and three real, dense railway lines. Encouraged by these positive results, we continue to refine our solution for scalability and performance. One effort currently underway is to generalise our algorithm to handle branching in networks (the results shown in Table 1 are all on linear topologies).

Line	Stations	Trains	Span	PWDD /minutes	
				PS	QL
SYN-1	11	60	4 hours	4.28	4.78
SYN-2	11	120	7 hours	15.5	18.54
KRCL	59	85	3 days	42.34	43.04
Kanpur	27	190	3 days	3.92	4.65
Ajmer	52	444	7 days	1.54	1.66

Table 1: Benchmark railway lines, and a comparison of PWDD values obtained by policy search (PS) and Q-learning (QL).

5 AI in India

The last decade has witnessed the dramatic surge AI and its entry into our daily lives. What effects has AI had on life and society; what is its future trajectory? We argue that societies must not view AI as an independent, universal force in whose path they happen to lie, but rather, actively direct its course and engage with it on their own terms [Kalyanakrishnan *et al.*, 2018]. As a concrete example, take self-driving vehicles, which have been the flagship of the advent of AI in the United States [Stone *et al.*, 2016], and are popularly perceived as *synonymous* with AI. On the other hand, countries such as India have the choice of ignoring self-driving vehicles altogether as they plan their technological growth. They could instead invest in integrated public transportation systems, which, incidentally, can also benefit from data-driven AI techniques for planning and scheduling. In other words, although the common view of AI is often defined by its end products, a more useful interpretation of AI is as a set of powerful, general-purpose *tools*, which can be used to create a variety of products and services.

The blossoming of AI opens up many opportunities specific to India. Modern techniques in speech recognition and natural language processing can be applied in many ways to preserve and enhance the country's linguistic diversity. The success of these methods depends critically on the availability of large data sets [Banko and Brill, 2001]. Hence, it is essential to bring more local-language data into an accessible digital format. In fact, across a wide variety of areas, AI-driven development can be promoted by the creation of structured and linked public data sets [Wood *et al.*, 2014]. Take the example of the healthcare sector, where the country faces significant shortages of infrastructure and personnel. While AI cannot substitute for these fundamental elements, it can inform and augment existing processes. Even relatively basic data analysis performed at a large scale can yield valuable inputs to policy making [Salvi *et al.*, 2015]. Interestingly, the process of digitising and structuring data can itself be supported by AI techniques such as OCR and text mining

An investment in AI to realise its opportunities needs to by accompanied by policy-making to address accompanying challenges. Job losses are an immediate concern as AI replaces human skill at the workplace. Also worrying are possible long-term effects caused by existing divisions based on caste, gender, and economic power. For instance, much optimism about AI stems from the fact that smart phones—a convenient vehicle to carry data and services—have now reached the far corners of the country. Yet, significantly fewer women have access to mobile phones than men: the gender gap could actually *widen* as more services migrate to smart platforms. It is another matter that the "tech" industry is dominated by men, and this imbalance leads to a strong gender bias in the products and services created [Truckenbrod, 1993].

We emphasise the need for rigorous academic scholarship on the effects of AI on Indian society [Kalyanakrishnan *et al.*, 2018]. It is essential that technological development through the efforts of computer scientists and engineers is complemented by independent assessments and contributions from social scientists, lawyers, and policy-makers.

6 Outlook

As an early-career researcher working in the field of AI, my own (constantly evolving) goals are set by my commitments to science, to engineering, and ultimately to society.

As evidenced by sections 2–4, there is no shortage of interesting scientific problems and applications to keep me busy. From a technical standpoint, a long-term target would be to understand how faculties such as sensing, language-processing, control, learning, and search—which are usually studied in isolation—interact with each other and integrate into complex, intelligent systems. It is also becoming evident that in practice, agents need to be designed for co-existence with humans, whose trust and confidence they must evoke. I hope to work closely with industry partners to get a clearer "end-to-end" picture of deployed agents. Also of interest are "grand challenge" demonstrations [Silver *et al.*, 2016] that can excite young minds to take up research in AI.

Although the opportunities offered by AI are plentiful, the incentives for pursuing them are heavily skewed in favour of those with commercial prospects. For illustration, consider a project to develop a translation engine for a language that is spoken only by a small community, of a few thousands of people. Such a project is unlikely to catch the attention of any large company that is answerable to its shareholders. On the other hand, with available open source tools and cloud-based computing resources, it is reasonable to expect that at a basic technical solution can be put together by a small group of dedicated enthusiasts, including some from the community itself. The main stumbling block would be the group's lack of training and exposure to modern AI techniques. One can easily imagine many other applications that potential beneficiaries might not have the technical skills and knowledge to develop. As an educator, one of my main goals is to promote the "democratisation" of AI—to train and enable students to develop AI for their own needs. I believe this approach is necessary to achieve equitable socioeconomic development.

A career in AI is bound to encounter *ethical* challenges in the coming years. I am committed to making responsible choices, even if they are inconvenient.

Acknowledgements

The author was partially supported by SERB grant ECR/2017/002479.

References

[Ashutosh *et al.*, 2020] Kumar Ashutosh, Sarthak Consul, Bhishma Dedhia, Parthasarathi Khirwadkar, Sahil Shah, and Shivaram Kalyanakrishnan. Lower bounds for policy iteration on multi-action MDPs. In *Proc. CDC 2020*, pages 1744–1749. IEEE, 2020.

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

[Banko and Brill, 2001] Michele Banko and Eric Brill. Scaling to very very large corpora for natural language disambiguation. In *Proc. ACL 2001*, pages 26–33. Association for Computational Linguistics, 2001.

[Dani *et al.*, 2008] Varsha Dani, Thomas P. Hayes, and Sham M. Kakade. Stochastic linear optimization under bandit feedback. In *Proc. COLT 2008*, pages 355–366. Omnipress, 2008.

[Hansen and Zwick, 2010] Thomas Dueholm Hansen and Uri Zwick. Lower bounds for Howard's algorithm for finding minimum mean-cost cycles. In *Proc. ISAAC 2011*, pages 425–426. Springer, 2010.

[Hansen, 2016] Nikolaus Hansen. The CMA evolution strategy: A tutorial. *CoRR*, abs/1604.00772, 2016.

[Howard, 1960] Ronald A. Howard. *Dynamic programming and Markov processes*. MIT Press, 1960.

[Kalyanakrishnan *et al.*, 2016a] Shivaram Kalyanakrishnan, Utkarsh Mall, and Ritish Goyal. Batch-switching policy iteration. In *Proc. IJCAI 2016*, pages 3147–3153. AAAI Press, 2016.

[Kalyanakrishnan *et al.*, 2016b] Shivaram Kalyanakrishnan, Neeldhara Misra, and Aditya Gopalan. Randomised procedures for initialising and switching actions in policy iteration. In *Proc. AAAI 2016*, pages 3145–3151. AAAI Press, 2016.

[Kalyanakrishnan *et al.*, 2018] Shivaram Kalyanakrishnan, Rahul Alex Panicker, Sarayu Natarajan, and Shreya Rao. Opportunities and challenges for artificial intelligence in India. In *Proc. AIES 2018*, pages 164–170. ACM, 2018.

[Khadilkar, 2017] Harshad Khadilkar. Scheduling of vehicle movement in resource-constrained transportation networks using a capacity-aware heuristic. In *Proc. Amer. Control Conf.*, pages 5617–5622. IEEE, 2017.

[Khadilkar, 2019] Harshad Khadilkar. A scalable reinforcement learning algorithm for scheduling railway lines. *IEEE Trans. on ITS*, 20(2):727–736, 2019.

[Kleinberg, 2005] Robert Kleinberg. Nearly tight bounds for the continuum-armed bandit problem. In *Adv. NIPS 17*, pages 697–704. MIT Press, 2005.

[Lai and Robbins, 1985] T.L. Lai and Herbert Robbins. Asymptotically efficient adaptive allocation rules. *Adv. in Applied Mathematics*, 6(1):4 – 22, 1985.

[Littman *et al.*, 1995] Michael L. Littman, Thomas L. Dean, and Leslie Pack Kaelbling. On the complexity of solving Markov decision problems. In *Proc. UAI 1995*, pages 394–402. Morgan Kaufmann, 1995.

[Mansour and Singh, 1999] Yishay Mansour and Satinder Singh. On the complexity of policy iteration. In *Proc. UAI 1999*, pages 401–408. Morgan Kaufmann, 1999.

[Melekopoglou and Condon, 1994] Mary Melekopoglou and Anne Condon. On the complexity of the policy improvement algorithm for Markov decision processes. *INFORMS Journal on Computing*, 6(2):188–192, 1994.

[Prasad *et al.*, 2020] Rohit Prasad, Harshad Khadilkar, and Shivaram Kalyanakrishnan. Optimising a real-time scheduler for railway lines using policy search. In *Proc. Adaptive and Learning Agents (ALA) workshop at AAMAS 2020*, 2020. https://ala2020.vub.ac.be/papers/ALA2020_paper_29.pdf.

[Robbins, 1952] Herbert Robbins. Some aspects of the sequential design of experiments. *Bulletin of the AMS*, 58(5):527–535, 1952.

[Roy Chaudhuri and Kalyanakrishnan, 2018] Arghya Roy Chaudhuri and Shivaram Kalyanakrishnan. Quantile-regret minimisation in infinitely many-armed bandits. In *Proc. UAI 2018*, pages 425–434. AUAI Press, 2018.

[Roy Chaudhuri and Kalyanakrishnan, 2020] Arghya Roy Chaudhuri and Shivaram Kalyanakrishnan. Regret minimisation in multi-armed bandits using bounded arm memory. In *Proc. AAAI 2020*, pages 10085–10092. AAAI Press, 2020.

[Salvi *et al.*, 2015] Sundeep Salvi, Komalkirti Apte, Sapna Madas, Monica Barne, Sushmeeta Chhowala, Tavpritesh Sethi, Kunal Aggarwal, Anurag Agrawal, and Jaideep Gogtay. Symptoms and medical conditions in 204,912 patients visiting primary health-care practitioners in India: a 1-day point prevalence study (the POSEIDON study). *The Lancet Global Health*, 3(12):e776 – e784, 2015.

[Silver *et al.*, 2016] David Silver, Aja Huang, Chris J. Maddison, Arthur Guez, Laurent Sifre, George van den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Vedavyas Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy P. Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel, and Demis Hassabis. Mastering the game of Go with deep neural networks and tree search. *Nature*, 529(7587):484–489, 2016.

[Stone *et al.*, 2016] Peter Stone, Rodney Brooks, Erik Brynjolfsson, Ryan Calo, Oren Etzioni, Greg Hager, Julia Hirschberg, Shivaram Kalyanakrishnan, Ece Kamar, Sarit Kraus, Kevin Leyton-Brown, David Parkes, William Press, AnnaLee Saxenian, Julie Shah, Milind Tambe, and Astro Teller. Artificial intelligence and life in 2030. One hundred year study on artificial intelligence: Report of the 2015-2016 study panel. Technical report, Stanford University, 2016. http://ai100.stanford.edu/2016-report.

[Taraviya and Kalyanakrishnan, 2019] Meet Taraviya and Shivaram Kalyanakrishnan. A tighter analysis of randomised policy iteration. In *Proc. UAI 2019*. AUAI Press, 2019. ID 174.

[Thompson, 1933] William R. Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25:285–294, 1933.

[Truckenbrod, 1993] Joan Truckenbrod. Women and the social construction of the computing culture: Evolving new forms of computing. *AI & Society*, 7(4):345–357, 1993.

[Wood *et al.*, 2014] David Wood, Marsha Zaidman, Luke Ruth, and Michael Hausenblas. *Linked Data: Structured Data on the Web*. Manning Publications, 2014.