Title: Multi-armed bandit

URL: https://en.wikipedia.org/wiki/Multi-armed_bandit

PageID: 2854828

Categories: Category:Machine learning, Category:Metaphors referring to body parts, Category:Science and technology during World War II, Category:Sequential experiments, Category:Sequential methods, Category:Stochastic optimization

---

In probability theory and machine learning , the multi-armed bandit problem (sometimes called the K - [1] or N -armed bandit problem [2]) is named from imagining a gambler at a row of slot machines (sometimes known as " one-armed bandits "), who has to decide which machines to play, how many times to play each machine and in which order to play them, and whether to continue with the current machine or try a different machine. [3]

More generally, it is a problem in which a decision maker iteratively selects one of multiple fixed choices (i.e., arms or actions) when the properties of each choice are only partially known at the time of allocation, and may become better understood as time passes. A fundamental aspect of bandit problems is that choosing an arm does not affect the properties of the arm or other arms. [4]

Instances of the multi-armed bandit problem include the task of iteratively allocating a fixed, limited set of resources between competing (alternative) choices in a way that minimizes the regret . [5] [6] A notable alternative setup for the multi-armed bandit problem includes the " best arm identification (BAI) " problem where the goal is instead to identify the best choice by the end of a finite number of rounds. [7]

The multi-armed bandit problem is a classic reinforcement learning problem that exemplifies the exploration–exploitation tradeoff dilemma . In contrast to general reinforcement learning, the selected actions in bandit problems do not affect the reward distribution of the arms.

The multi-armed bandit problem also falls into the broad category of stochastic scheduling .

In the problem, each machine provides a random reward from a probability distribution specific to that machine, that is not known a priori . The objective of the gambler is to maximize the sum of rewards earned through a sequence of lever pulls. [5] [6] The crucial tradeoff the gambler faces at each trial is between "exploitation" of the machine that has the highest expected payoff and "exploration" to get more information about the expected payoffs of the other machines. The trade-off between exploration and exploitation is also faced in machine learning. In practice, multi-armed bandits have been used to model problems such as managing research projects in a large organization, like a science foundation or a pharmaceutical company . [5] [6] In early versions of the problem, the gambler begins with no initial knowledge about the machines.

Herbert Robbins in 1952, realizing the importance of the problem, constructed convergent population selection strategies in "some aspects of the sequential design of experiments". [8] A theorem, the Gittins index , first published by John C. Gittins , gives an optimal policy for maximizing the expected discounted reward. [9]

Empirical motivation

The multi-armed bandit problem models an agent that simultaneously attempts to acquire new knowledge (called "exploration") and optimize their decisions based on existing knowledge (called "exploitation"). The agent attempts to balance these competing tasks in order to maximize their total value over the period of time considered. There are many practical applications of the bandit model, for example:

clinical trials investigating the effects of different experimental treatments while minimizing patient losses, [5] [6] [10] [11]

adaptive routing efforts for minimizing delays in a network,

financial portfolio design [12] [13]

In these practical examples, the problem requires balancing reward maximization based on the knowledge already acquired with attempting new actions to further increase knowledge. This is known as the exploitation vs. exploration tradeoff in machine learning .

The model has also been used to control dynamic allocation of resources to different projects, answering the question of which project to work on, given uncertainty about the difficulty and payoff of each possibility. [14]

Originally considered by Allied scientists in World War II , it proved so intractable that, according to Peter Whittle , the problem was proposed to be dropped over Germany so that German scientists could also waste their time on it. [15]

The version of the problem now commonly analyzed was formulated by Herbert Robbins in 1952.

The multi-armed bandit model

The multi-armed bandit (short: bandit or MAB) can be seen as a set of real distributions B = { R 1 , … , R K } {\displaystyle B=\{R_{1},\dots ,R_{K}\}} , each distribution being associated with the rewards delivered by one of the K ∈ N + {\displaystyle K\in \mathbb {N} ^{+}} levers. Let μ 1 , … , μ K {\displaystyle \mu _{1},\dots ,\mu _{K}} be the mean values associated with these reward distributions. The gambler iteratively plays one lever per round and observes the associated reward. The objective is to maximize the sum of the collected rewards. The horizon H {\displaystyle H} is the number of rounds that remain to be played. The bandit problem is formally equivalent to a one-state Markov decision process . The regret ρ {\displaystyle \rho } after T {\displaystyle T} rounds is defined as the expected difference between the reward sum associated with an optimal strategy and the sum of the collected rewards:

ρ = T μ ∗ − ∑ t = 1 T r ^ t {\displaystyle \rho =T\mu ^{*}-\sum _{t=1}^{T}{\widehat {r}}_{t}} ,

where μ ∗ {\displaystyle \mu ^{*}} is the maximal reward mean, μ ∗ = max k { μ k } {\displaystyle \mu ^{*}=\max _{k}\{\mu _{k}\}} , and r ^ t {\displaystyle {\widehat {r}}_{t}} is the reward in round t .

A zero-regret strategy is a strategy whose average regret per round ρ / T {\displaystyle \rho /T} tends to zero with probability 1 when the number of played rounds tends to infinity. [16] Intuitively, zero-regret strategies are guaranteed to converge to a (not necessarily unique) optimal strategy if enough rounds are played.

Variations

A common formulation is the Binary multi-armed bandit or Bernoulli multi-armed bandit, which issues a reward of one with probability p {\displaystyle p} , and otherwise a reward of zero.

Another formulation of the multi-armed bandit has each arm representing an independent Markov machine. Each time a particular arm is played, the state of that machine advances to a new one, chosen according to the Markov state evolution probabilities. There is a reward depending on the current state of the machine. In a generalization called the "restless bandit problem", the states of non-played arms can also evolve over time. [17] There has also been discussion of systems where the number of choices (about which arm to play) increases over time. [18]

Computer science researchers have studied multi-armed bandits under worst-case assumptions, obtaining algorithms to minimize regret in both finite and infinite (asymptotic) time horizons for both stochastic [1] and non-stochastic [19] arm payoffs.

Best arm identification

An important variation of the classical regret minimization problem in multi-armed bandits is best arm identification (BAI), [20] also known as pure exploration . This problem is crucial in various applications, including clinical trials, adaptive routing, recommendation systems, and A/B testing.

In BAI, the objective is to identify the arm having the highest expected reward. An algorithm in this setting is characterized by a sampling rule , a decision rule, and a stopping rule , described as follows:

Sampling rule : (a t) t ≥ 1 {\displaystyle (a_{t})_{t\geq 1}} is a sequence of actions at each time step

Stopping rule : τ {\displaystyle \tau } is a (random) stopping time which suggests when to stop collecting samples

Decision rule : a ^ τ {\displaystyle {\hat {a}}_{\tau }} is a guess on the best arm based on the data collected up to time τ {\displaystyle \tau }

There are two predominant settings in BAI:

Fixed budget setting: Given a time horizon T ≥ 1 {\displaystyle T\geq 1} , the objective is to identify the arm with the highest expected reward a ■ ∈ arg ■ max k μ k {\displaystyle a^{\star }\in \arg \max _{k}\mu _{k}} minimizing probability of error δ {\displaystyle \delta } .

Fixed confidence setting: Given a confidence level δ ∈ (0 , 1) {\displaystyle \delta \in (0,1)} , the objective is to identify the arm with the highest expected reward a ■ ∈ arg ■ max k μ k {\displaystyle a^{\star }\in \arg \max _{k}\mu _{k}} with the least possible amount of trials and with probability of error P (a ^ τ ≠ a ■) ≤ δ {\displaystyle \mathbb {P} ({\hat {a}}_{\tau }\neq a^{\star })\leq \delta } .

For example using a decision rule , we could use m 1 {\displaystyle m_{1}} where m {\displaystyle m} is the machine no.1 (you can use a different variable respectively) and 1 {\displaystyle 1} is the amount for each time an attempt is made at pulling the lever, where ∫ ∑ m 1 , m 2 , (. . .) = M {\displaystyle \int \sum m_{1},m_{2},(...)=M} , identify M {\displaystyle M} as the sum of each attempts m 1 + m 2 {\displaystyle m_{1}+m_{2}} , (...) as needed, and from there you can get a ratio, sum or mean as quantitative probability and sample your formulation for each slots.

You can also do ∫ ∑ k ∝ i N − (n j) {\displaystyle \int \sum _{k\propto _{i}}^{N}-(n_{j})} where m 1 + m 2 {\displaystyle m1+m2} equal to each a unique machine slot, x , y {\displaystyle x,y} is the amount each time the lever is triggered, N {\displaystyle N} is the sum of (m 1 x , y) + (m 2 x , y) (. . .) {\displaystyle (m1_{x},_{y})+(m2_{x},_{y})(...)} , k {\displaystyle k} would be the total available amount in your possession, k {\displaystyle k} is relative to N {\displaystyle N} where N = n (n a , b) , (n 1 a , b) , (n 2 a , b) {\displaystyle N=n(n_{a},b),(n1_{a},b),(n2_{a},b)} reduced n j {\displaystyle n_{j}} as the sum of each gain or loss from a , b {\displaystyle a,b} (for example, suppose you have 100$ that is defined as n {\displaystyle n} , and a {\displaystyle a} would be a gain, b {\displaystyle b} is equal to a loss. From there you get your results either positive or negative to add for N {\displaystyle N} with your own specific rule) and i {\displaystyle i} as the maximum you are willing to spend.

It is possible to express this construction using a combination of multiple algebraic formulation, as mentioned above where you can limit with T {\displaystyle T} for, or in time and so on.

Bandit strategies

A major breakthrough was the construction of optimal population selection strategies, or policies (that possess uniformly maximum convergence rate to the population with highest mean) in the work described below.

Optimal solutions

In the paper "Asymptotically efficient adaptive allocation rules", Lai and Robbins [21] (following papers of Robbins and his co-workers going back to Robbins in the year 1952) constructed convergent population selection policies that possess the fastest rate of convergence (to the population with highest mean) for the case that the population reward distributions are the one-parameter exponential family. Then, in Katehakis and Robbins [22] simplifications of the policy and the main proof were given for the case of normal populations with known variances. The next notable progress was obtained by Burnetas and Katehakis in the paper "Optimal adaptive policies for sequential allocation problems", [23] where index based policies with uniformly maximum convergence rate were constructed, under more general conditions that include the case in which the distributions of outcomes from each population depend on a vector of unknown parameters. Burnetas and Katehakis (1996) also provided an explicit solution for the important case in which the distributions of outcomes follow arbitrary (i.e., non-parametric) discrete, univariate distributions.

Later in "Optimal adaptive policies for Markov decision processes" [24] Burnetas and Katehakis studied the much larger model of Markov Decision Processes under partial information, where the transition law and/or the expected one period rewards may depend on unknown parameters. In this work, the authors constructed an explicit form for a class of adaptive policies with uniformly maximum convergence rate properties for the total expected finite horizon reward under sufficient assumptions of finite state-action spaces and irreducibility of the transition law. A main feature of these policies is that the choice of actions, at each state and time period, is based on indices that are inflations of the right-hand side of the estimated average reward optimality equations. These inflations have recently been called the optimistic approach in the work of Tewari and Bartlett, [25] Ortner [26] Filippi, Cappé, and Garivier, [27] and Honda and Takemura. [28]

For Bernoulli multi-armed bandits, Pilarski et al. [29] studied computation methods of deriving fully optimal solutions (not just asymptotically) using dynamic programming in the paper "Optimal Policy for Bernoulli Bandits: Computation and Algorithm Gauge." [29] Via indexing schemes, lookup tables, and other techniques, this work provided practically applicable optimal solutions for Bernoulli bandits provided that time horizons and numbers of arms did not become excessively large. Pilarski et al. [30] later extended this work in "Delayed Reward Bernoulli Bandits: Optimal Policy and Predictive Meta-Algorithm PARDI" [30] to create a method of determining the optimal policy for Bernoulli bandits when rewards may not be immediately revealed following a decision and may be delayed. This method relies upon calculating expected values of reward outcomes which have not yet been revealed and updating posterior probabilities when rewards are revealed.

When optimal solutions to multi-arm bandit tasks [31] are used to derive the value of animals' choices, the activity of neurons in the amygdala and ventral striatum encodes the values derived from these policies, and can be used to decode when the animals make exploratory versus exploitative choices. Moreover, optimal policies better predict animals' choice behavior than alternative strategies (described below). This suggests that the optimal solutions to multi-arm bandit problems are biologically plausible, despite being computationally demanding. [32]

Approximate solutions

Many strategies exist which provide an approximate solution to the bandit problem, and can be put into the four broad categories detailed below.

Semi-uniform strategies

Semi-uniform strategies were the earliest (and simplest) strategies discovered to approximately solve the bandit problem. All those strategies have in common a greedy behavior where the best lever (based on previous observations) is always pulled except when a (uniformly) random action is taken.

Epsilon-greedy strategy : [33] The best lever is selected for a proportion 1 – ■ {\displaystyle 1-\epsilon } of the trials, and a lever is selected at random (with uniform probability) for a proportion ■ {\displaystyle \epsilon } . A typical parameter value might be ■ = 0.1 {\displaystyle \epsilon =0.1} , but this can vary widely depending on circumstances and predilections.

Epsilon-first strategy [citation needed] : A pure exploration phase is followed by a pure exploitation phase. For N {\displaystyle N} trials in total, the exploration phase occupies ■ N {\displaystyle \epsilon N} trials and the exploitation phase (1 – ■) N {\displaystyle (1-\epsilon)N} trials. During the exploration phase, a lever is randomly selected (with uniform probability); during the exploitation phase, the best lever is always selected.

Epsilon-decreasing strategy [citation needed] : Similar to the epsilon-greedy strategy, except that the value of ■ {\displaystyle \epsilon } decreases as the experiment progresses, resulting in highly explorative behaviour at the start and highly exploitative behaviour at the finish.

Adaptive epsilon-greedy strategy based on value differences (VDBE) : Similar to the epsilon-decreasing strategy, except that epsilon is reduced on basis of the learning progress instead of manual tuning (Tokic, 2010). [34] High fluctuations in the value estimates lead to a high epsilon (high exploration, low exploitation); low fluctuations to a low epsilon (low exploration, high exploitation). Further improvements can be achieved by a softmax -weighted action selection in

case of exploratory actions (Tokic & Palm, 2011). [35]

Adaptive epsilon-greedy strategy based on Bayesian ensembles (Epsilon-BMC) : An adaptive epsilon adaptation strategy for reinforcement learning similar to VBDE, with monotone convergence guarantees. In this framework, the epsilon parameter is viewed as the expectation of a posterior distribution weighting a greedy agent (that fully trusts the learned reward) and uniform learning agent (that distrusts the learned reward). This posterior is approximated using a suitable Beta distribution under the assumption of normality of observed rewards. In order to address the possible risk of decreasing epsilon too quickly, uncertainty in the variance of the learned reward is also modeled and updated using a normal-gamma model. (Gimelfarb et al., 2019). [36]

Probability matching strategies

Probability matching strategies reflect the idea that the number of pulls for a given lever should match its actual probability of being the optimal lever. Probability matching strategies are also known as Thompson sampling or Bayesian Bandits, [37] [38] and are surprisingly easy to implement if you can sample from the posterior for the mean value of each alternative.

Probability matching strategies also admit solutions to so-called contextual bandit problems. [37]

Pricing strategies

Pricing strategies establish a price for each lever. For example, as illustrated with the POKER algorithm, [16] the price can be the sum of the expected reward plus an estimation of extra future rewards that will gain through the additional knowledge. The lever of highest price is always pulled.

Contextual bandit

A useful generalization of the multi-armed bandit is the contextual multi-armed bandit. At each iteration an agent still has to choose between arms, but they also see a d-dimensional feature vector, the context vector they can use together with the rewards of the arms played in the past to make the choice of the arm to play. Over time, the learner's aim is to collect enough information about how the context vectors and rewards relate to each other, so that it can predict the next best arm to play by looking at the feature vectors. [39]

Approximate solutions for contextual bandit

Many strategies exist that provide an approximate solution to the contextual bandit problem, and can be put into two broad categories detailed below.

Online linear bandits

LinUCB (Upper Confidence Bound) algorithm : the authors assume a linear dependency between the expected reward of an action and its context and model the representation space using a set of linear predictors. [40] [41]

LinRel (Linear Associative Reinforcement Learning) algorithm : Similar to LinUCB, but utilizes singular value decomposition rather than ridge regression to obtain an estimate of confidence. [42] [43]

Online non-linear bandits

UCBogram algorithm : The nonlinear reward functions are estimated using a piecewise constant estimator called a regressogram in nonparametric regression . Then, UCB is employed on each constant piece. Successive refinements of the partition of the context space are scheduled or chosen adaptively. [44] [45] [46]

Generalized linear algorithms : The reward distribution follows a generalized linear model, an extension to linear bandits. [47] [48] [49] [50]

KernelUCB algorithm : a kernelized non-linear version of LinUCB, with efficient implementation and finite-time analysis. [51]

Bandit Forest algorithm : a random forest is built and analyzed w.r.t the random forest built knowing the joint distribution of contexts and rewards. [52]

Oracle-based algorithm : The algorithm reduces the contextual bandit problem into a series of supervised learning problem, and does not rely on typical realizability assumption on the reward function. [53]

Constrained contextual bandit

In practice, there is usually a cost associated with the resource consumed by each action and the total cost is limited by a budget in many applications such as crowdsourcing and clinical trials. Constrained contextual bandit (CCB) is such a model that considers both the time and budget constraints in a multi-armed bandit setting.

A. Badanidiyuru et al. [54] first studied contextual bandits with budget constraints, also referred to as Resourceful Contextual Bandits, and show that a O (T) {\displaystyle O({\sqrt {T}})} regret is achievable. However, their work focuses on a finite set of policies, and the algorithm is computationally inefficient.

A simple algorithm with logarithmic regret is proposed in: [55]

UCB-ALP algorithm : The framework of UCB-ALP is shown in the right figure. UCB-ALP is a simple algorithm that combines the UCB method with an Adaptive Linear Programming (ALP) algorithm, and can be easily deployed in practical systems. It is the first work that show how to achieve logarithmic regret in constrained contextual bandits. Although [55] is devoted to a special case with single budget constraint and fixed cost, the results shed light on the design and analysis of algorithms for more general CCB problems.

Adversarial bandit

Another variant of the multi-armed bandit problem is called the adversarial bandit, first introduced by Auer and Cesa-Bianchi (1998). In this variant, at each iteration, an agent chooses an arm and an adversary simultaneously chooses the payoff structure for each arm. This is one of the strongest generalizations of the bandit problem [56] as it removes all assumptions of the distribution and a solution to the adversarial bandit problem is a generalized solution to the more specific bandit problems.

Example: Iterated prisoner's dilemma

An example often considered for adversarial bandits is the iterated prisoner's dilemma . In this example, each adversary has two arms to pull. They can either Deny or Confess. Standard stochastic bandit algorithms don't work very well with these iterations. For example, if the opponent cooperates in the first 100 rounds, defects for the next 200, then cooperate in the following 300, etc. then algorithms such as UCB won't be able to react very quickly to these changes. This is because after a certain point sub-optimal arms are rarely pulled to limit exploration and focus on exploitation. When the environment changes the algorithm is unable to adapt or may not even detect the change.

Approximate solutions

Exp3

Source: [57]

EXP3 is a popular algorithm for adversarial multiarmed bandits, suggested and analyzed in this setting by Auer et al. [2002b].

Recently there was an increased interest in the performance of this algorithm in the stochastic setting, due to its new applications to stochastic multi-armed bandits with side information [Seldin et al., 2011] and to multi-armed bandits in the mixed stochastic-adversarial setting [Bubeck and Slivkins, 2012].

The paper presented an empirical evaluation and improved analysis of the performance of the EXP3 algorithm in the stochastic setting, as well as a modification of the EXP3 algorithm capable of achieving "logarithmic" regret in stochastic environment.

Exp3 chooses an arm at random with probability (1 − γ) {\displaystyle (1-\gamma)} it prefers arms with higher weights (exploit), it chooses with probability γ {\displaystyle \gamma } to uniformly

randomly explore. After receiving the rewards the weights are updated. The exponential growth significantly increases the weight of good arms.

The (external) regret of the Exp3 algorithm is at most O (K T l o g (K)) {\displaystyle O({\sqrt {KTlog(K)}})}

Follow the perturbed leader (FPL) algorithm

We follow the arm that we think has the best performance so far adding exponential noise to it to provide exploration. [58]

Exp3 vs FPL

Infinite-armed bandit

In the original specification and in the above variants, the bandit problem is specified with a discrete and finite number of arms, often indicated by the variable K {\displaystyle K} . In the infinite armed case, introduced by Agrawal (1995), [59] the "arms" are a continuous variable in K {\displaystyle K} dimensions.

Non-stationary bandit

This framework refers to the multi-armed bandit problem in a non-stationary setting (i.e., in presence of concept drift). In the non-stationary setting, it is assumed that the expected reward for an arm k {\displaystyle k} can change at every time step t ∈ T {\displaystyle t\in {\mathcal {T}}} : μ t – 1 k ≠ μ t k {\displaystyle \mu _{t-1}^{k}\neq \mu _{t}^{k}} . Thus, μ t k {\displaystyle \mu _{t}^{k}} no longer represents the whole sequence of expected (stationary) rewards for arm k {\displaystyle k} . Instead, μ k {\displaystyle \mu ^{k}} denotes the sequence of expected rewards for arm k {\displaystyle k} , defined as μ k = { μ t k } t = 1 T {\displaystyle \mu ^{k}=\{\mu _{t}^{k}\}_{t=1}^{T}} . [60]

A dynamic oracle represents the optimal policy to be compared with other policies in the non-stationary setting. The dynamic oracle optimises the expected reward at each step t ∈ T {\displaystyle t\in {\mathcal {T}}} by always selecting the best arm, with expected reward of μ t ∗ {\displaystyle \mu _{t}^{*}} . Thus, the cumulative expected reward D (T) {\displaystyle {\mathcal {D}}(T)} for the dynamic oracle at final time step T {\displaystyle T} is defined as:

D (T) = ∑ t = 1 T μ t ∗ . {\displaystyle {\mathcal {D}}(T)=\sum _{t=1}^{T}{\mu _{t}^{*}}.}

Hence, the regret ρ π (T) {\displaystyle \rho ^{\pi }(T)} for policy π {\displaystyle \pi } is computed as the difference between D (T) {\displaystyle {\mathcal {D}}(T)} and the cumulative expected reward at step T {\displaystyle T} for policy π {\displaystyle \pi } :

ρ π (T) = ∑ t = 1 T μ t ∗ – E π μ [∑ t = 1 T r t] = D (T) – E π μ [∑ t = 1 T r t] . {\displaystyle \rho ^{\pi }(T)=\sum _{t=1}^{T}{\mu _{t}^{*}}-\mathbb {E} _{\pi }^{\mu }\left[\sum _{t=1}^{T}{r_{t}}\right]={\mathcal {D}}(T)-\mathbb {E} _{\pi }^{\mu }\left[\sum _{t=1}^{T}{r_{t}}\right].}

Garivier and Moulines derive some of the first results with respect to bandit problems where the underlying model can change during play. A number of algorithms were presented to deal with this case, including Discounted UCB [61] and Sliding-Window UCB. [62] A similar approach based on Thompson Sampling algorithm is the f-Discounted-Sliding-Window Thompson Sampling (f-dsw TS) [63] proposed by Cavenaghi et al. The f-dsw TS algorithm exploits a discount factor on the reward history and an arm-related sliding window to contrast concept drift in non-stationary environments. Another work by Burtini et al. introduces a weighted least squares Thompson sampling approach (WLS-TS), which proves beneficial in both the known and unknown non-stationary cases. [64]

Other variants

Many variants of the problem have been proposed in recent years.

Dueling bandit

The dueling bandit variant was introduced by Yue et al. (2012) [65] to model the exploration-versus-exploitation tradeoff for relative feedback.

In this variant the gambler is allowed to pull two levers at the same time, but they only get a binary feedback telling which lever provided the best reward. The difficulty of this problem stems from the fact that the gambler has no way of directly observing the reward of their actions.

The earliest algorithms for this problem were InterleaveFiltering [65] and Beat-The-Mean. [66] The relative feedback of dueling bandits can also lead to voting paradoxes . A solution is to take the Condorcet winner as a reference. [67]

More recently, researchers have generalized algorithms from traditional MAB to dueling bandits: Relative Upper Confidence Bounds (RUCB), [68] Relative EXponential weighing (REX3), [69] Copeland Confidence Bounds (CCB), [70] Relative Minimum Empirical Divergence (RMED), [71] and Double Thompson Sampling (DTS). [72]

Collaborative bandit

Approaches using multiple bandits that cooperate sharing knowledge in order to better optimize their performance started in 2013 with "A Gang of Bandits", [73] an algorithm relying on a similarity graph between the different bandit problems to share knowledge. The need of a similarity graph was removed in 2014 by the work on the CLUB algorithm. [74] Following this work, several other researchers created algorithms to learn multiple models at the same time under bandit feedback.

For example, COFIBA was introduced by Li and Karatzoglou and Gentile (SIGIR 2016), [75] where the classical collaborative filtering, and content-based filtering methods try to learn a static recommendation model given training data.

Combinatorial bandit

The Combinatorial Multiarmed Bandit (CMAB) problem [76] [77] [78] arises when instead of a single discrete variable to choose from, an agent needs to choose values for a set of variables. Assuming each variable is discrete, the number of possible choices per iteration is exponential in the number of variables. Several CMAB settings have been studied in the literature, from settings where the variables are binary [77] to more general setting where each variable can take an arbitrary set of values. [78]

See also

Gittins index – a powerful, general strategy for analyzing bandit problems.

Greedy algorithm

Optimal stopping

Search theory

Stochastic scheduling

References

Further reading

Guha, S.; Munagala, K.; Shi, P. (2010), "Approximation algorithms for restless bandit problems", Journal of the ACM , 58 : 1– 50, arXiv : 0711.3861 , doi : 10.1145/1870103.1870106 , S2CID 1654066

Dayanik, S.; Powell, W.; Yamazaki, K. (2008), "Index policies for discounted bandit problems with availability constraints", Advances in Applied Probability , 40 (2): 377– 400, doi : 10.1239/aap/1214950209 .

Powell, Warren B. (2007), "Chapter 10", Approximate Dynamic Programming: Solving the Curses of Dimensionality , New York: John Wiley and Sons, ISBN 978-0-470-17155-4 .

Robbins, H. (1952), "Some aspects of the sequential design of experiments", Bulletin of the American Mathematical Society , 58 (5): 527– 535, doi : 10.1090/S0002-9904-1952-09620-8 .

Sutton, Richard; Barto, Andrew (1998), Reinforcement Learning , MIT Press, ISBN 978-0-262-19398-6 , archived from the original on 2013-12-11 .

Allesiardo, Robin (2014), "A Neural Networks Committee for the Contextual Bandit Problem", Neural Information Processing – 21st International Conference, ICONIP 2014, Malaisia, November 03-06,2014, Proceedings , Lecture Notes in Computer Science, vol. 8834, Springer, pp. 374– 381, arXiv : 1409.8191 , doi : 10.1007/978-3-319-12637-1_47 , ISBN 978-3-319-12636-4 , S2CID 14155718 .

Weber, Richard (1992), "On the Gittins index for multiarmed bandits", Annals of Applied Probability , 2 (4): 1024– 1033, doi : 10.1214/aoap/1177005588 , JSTOR 2959678 .

Katehakis, M. ; C. Derman (1986), "Computing optimal sequential allocation rules in clinical trials", Adaptive statistical procedures and related topics , Institute of Mathematical Statistics Lecture Notes - Monograph Series, vol. 8, pp. 29– 39, doi : 10.1214/lnms/1215540286 , ISBN 978-0-940600-09-6 , JSTOR 4355518 .

Katehakis, Michael N. ; Veinott, Jr., Arthur F. (1987), "The multi-armed bandit problem: decomposition and computation", Mathematics of Operations Research , 12 (2): 262– 268, doi : 10.1287/moor.12.2.262 , JSTOR 3689689 , S2CID 656323

External links

MABWiser , open-source Python implementation of bandit strategies that supports context-free, parametric and non-parametric contextual policies with built-in parallelization and simulation capability.

PyMaBandits , open-source implementation of bandit strategies in Python and Matlab.

Contextual , open-source R package facilitating the simulation and evaluation of both context-free and contextual Multi-Armed Bandit policies.

bandit.sourceforge.net Bandit project , open-source implementation of bandit strategies.

Banditlib , open-source implementation of bandit strategies in C++.

Leslie Pack Kaelbling and Michael L. Littman (1996). Exploitation versus Exploration: The Single-State Case .

Tutorial: Introduction to Bandits: Algorithms and Theory. Part1 . Part2 .

Feynman's restaurant problem , a classic example (with known answer) of the exploitation vs. exploration tradeoff.

Bandit algorithms vs. A-B testing .

S. Bubeck and N. Cesa-Bianchi A Survey on Bandits .

A Survey on Contextual Multi-armed Bandits , a survey/tutorial for Contextual Bandits.

Blog post on multi-armed bandit strategies, with Python code .

Animated, interactive plots illustrating Epsilon-greedy, Thompson sampling , and Upper Confidence Bound exploration/exploitation balancing strategies.

v

t

e

Differentiable programming

Information geometry

Statistical manifold

Automatic differentiation

Neuromorphic computing

Pattern recognition

Ricci calculus

Computational learning theory

Inductive bias

IPU

TPU

VPU

Memristor

SpiNNaker

TensorFlow

PyTorch

Keras

scikit-learn

Theano

JAX

Flux.jl

MindSpore

Portals Computer programming Technology

Computer programming

Technology