CMSC 858G: Bandits, Experts and Games 10/24/16

Lecture 8: Adversarial Bandits

Instructor: Alex Slivkins Scribed by: Shuo Li, Katherine Scola

We study *adversarial bandits*: bandits with adversarially chosen costs. On the scale from "optimistic" to "pessimistic", IID assumption is super-optimistic, and adversarial is super-pessimistic. We focus on deterministic, oblivious adversary: that is, the costs for all arms and all rounds are chosen in advance.

We will proceed via a reduction to the full-feedback problem studied in the previous lecture. Further, we will actually solve adversarial bandits in a more general version that explicitly includes expert advice.

Setup and notation. There are K actions and a fixed time horizon T. Each action a has an associated cost $c_t(a)$ at round t. The total cost of each expert is the sum over all rounds: $\texttt{cost}(a) = \Sigma_{t=1}^{T} c_t(x)$. Regret is defined as the difference between the total cost of the algorithm and the minimum total cost of an action:

$$R(T) = \texttt{cost}(\texttt{ALG}) - \min_{\text{actions } a} \texttt{cost}(a). \tag{1}$$

(We do not take expectations in this definition; instead, we upper-bound the expected regret.)

Results. We will design an algorithm with regret bound

$$\mathbb{E}[R(T)] \leq O\left(\sqrt{KT \log K}\right)$$

Curiously, this upper bound not only matches the result for IID bandits, but in fact improves it a little bit, replacing the $\log(T)$ term with $\log(K)$.

Recall that we had a lower bound $\Omega(\sqrt{KT})$ (which holds even for IID bandits). Note the $\sqrt{\log K}$ gap between the upper and lower bound. It turns out the lower bound is in fact the right one: there is another algorithm with regret $O(\sqrt{KT})$.

1 Recap: best-expert problem

Last lecture, we studied bandits with *full feedback* (cost of each arm is revealed after every round) and adversarially chosen costs. A common interpretation of this problem is that each action corresponds to an "expert" that gives advice or makes predictions, and in each round the algorithm needs to choose which expert to follow. Hence, this problem is also known as the *best-expert problem*.

To facilitate exposition in this lecture, when we talk about the full-feedback problem, we will refer to actions as *experts* $x \in X$, where X is the set of all experts. The number of experts is $N = |X|$. Regret is defined as in (1).

We studied an algorithm for the best-expert problem called `Hedge`. In each round, this algorithm computes a distribution over experts, denoted p_t, and samples an expert from this distribution. We will use the following regret bounds proved in last lecture:

Theorem 1.1 (`Hedge`). *For the best-expert problem with adaptive adversary,* `Hedge` *satisfies*

$$\mathbb{E}[R(T)] \leq 2\sqrt{3} \cdot \sqrt{UT \log N},$$

where U is a number known to the algorithm such that

(a) $c_t(x) \leq U$ for all experts x and all rounds t, or

(b) $\mathbb{E}[G_t] \leq U$ for all rounds t, where $G_t = \sum_{x \in X} p_t(x)\, c_t^2(x)$.

This regret bound is essentially optimal. Specifically, we have the following lower bound:

Theorem 1.2 (lower bound). *Consider the best-expert problem with randomized, oblivious adversary and 0-1 costs. For each T, N there is a problem instance with N experts and time horizon T such that every algorithm suffers regret*

$$\mathbb{E}[R(T)] \geq \Omega\left(\sqrt{T \log N}\right). \tag{2}$$

The problem instance is very simple: each $c_t(x)$ is chosen independently and uniformly at random from $\{0, 1\}$. It is crucial for this theorem that the benchmark in regret is the "hindsight" benchmark $\min_x \texttt{cost}(x)$ rather than the "foresight" benchmark $\min_x \mathbb{E}[\texttt{cost}(x)]$. (Obviously, with respect to the latter benchmark any algorithm will have 0 regret for this problem instance.)

The theorem also implies that for each algorithm, there exists a deterministic problem instance with 0-1 costs such that this algorithm suffers regret (2) on this problem instance.

2 Reduction: from bandit feedback to full feedback

Our algorithm will be a reduction from bandit feedback to full feedback. Specifically, we take `Hedge`, an algorithm for the full-feedback setting, and use it as a subroutine.

The reduction proceeds as follows. For each arm, we create an expert which always recommends this arm. We use `Hedge` with this set of experts. In each round, we use the expert chosen by `Hedge` to pick an arm, and define "fake costs" on all experts in order to provide `Hedge` with valid inputs.

Algorithm 1 Reduction from bandit feedback to full feedback

Given: set X of experts.
In each round t,

1. Call `Hedge`, receive the probability distribution p_t over X.
2. Draw an expert x_t independently from p_t.
3. Use x_t to pick arm a_t (TBD).
4. Observe the cost $c_t(a_t)$ of the chosen arm.
5. Define "fake costs" $\hat{c}_t(x)$ for all experts $x \in X$ (TBD).
6. Return the "fake costs" to `Hedge`.

The details of *how* to define a_t using x_t, and *how* to define fake costs, will be provided later.

3 Adversarial bandits with expert advice

The reduction defined above suggests a more general problem: what if experts can make arbitrary predictions? This problem is called *bandits with expert advice*. We will in fact solve this problem rather than the original adversarial bandits problem. We do it for three reasons: because it is a very interesting generalization, because we can solve it with very little extra work, and because it separating experts from actions makes the solution clearer.

Formally, the problem of bandits with expert advice is defined as follows. There are K arms and N experts, and a fixed time horizon T. In each round $t \in [T]$,

- adversary picks cost $c_t(a)$ for each arm a,
- each expert x recommends an arm $a_{t,x}$,
- algorithm picks arm a_t and receive the corresponding cost $c_t(a_t)$.

The total cost of each expert is defined as $\texttt{cost}(x) = \sum_{t\in[T]} c_t(a_{t,x})$. The goal is to minimize regret w.r.t. best expert (rather than w.r.t. best action):

$$R(T) = \texttt{cost}(\texttt{ALG}) - \min_{\text{experts } x} \texttt{cost}(x).$$

We focus on deterministic, oblivious adversary: the costs for all arms and all rounds are selected in advance. Further, we assume that the expert recommendations $a_{t,x}$ are also chosen in advance; in other words, experts cannot learn over time.

We will use the reduction in Algorithm 1 to solve this problem. In fact, we will *really* use the same reduction: the pseudocode applies as is! Our algorithm will have regret

$$\mathbb{E}[R(T)] \leq O\left(\sqrt{KT \log N}\right).$$

Note the logarithmic dependence on N – this regret bound allows to handle *lots* of experts.

In fact, there is a matching lower bound, for any given K, T, N. It can be proved by a simple reduction to the basic $\Omega(\sqrt{KT})$ lower bound for bandits (and we may see it on HW2).

4 Preliminary analysis: unbiased estimates

The cost of each expert x at time step t is a function of the arm $a_{t,x}$ they recommend at that time step. For brevity, denote this as $c_t(x) = c_t(a_{t,x})$. We will also define "fake costs", denoted $\hat{c}_t(x)$, which are fed to `Hedge`. The regret bound for `Hedge` will be in terms of these fake costs:

$$\widehat{R}_{\texttt{Hedge}}(T) = \widehat{\texttt{cost}}(\texttt{Hedge}) - \min_x \widehat{\texttt{cost}}(x).$$

We want these fake costs to be unbiased estimates of the true costs. This is because we will need to convert a bound on the "fake regret" $\widehat{R}_{\texttt{Hedge}}(T)$ into a statement about the true costs accumulated by `Hedge`. Formally, we will ensure that

$$\mathbb{E}[\hat{c}_t(x) \mid \vec{p}_t] = c_t(x) \quad \text{for all experts } x, \tag{3}$$

where $\vec{p}_t = (p_t(x) : \text{ all experts } x)$. We use it to show:

Claim 4.1. *Assuming* (3), $\mathbb{E}[R_{\texttt{Hedge}}(T)] \leq \mathbb{E}[\widehat{R}_{\texttt{Hedge}}(T)]$.

Proof. First, we connect true cost of `Hedge` with its fake cost: we prove that

$$\mathbb{E}[\widehat{\texttt{cost}}(\texttt{Hedge})] = \mathbb{E}[\texttt{cost}(\texttt{Hedge})]. \tag{4}$$

Indeed,

$$\begin{aligned}
\mathbb{E}[\hat{c}_t(x_t)|\vec{p}_t] &= \mathbb{E}[\sum_x p_t(x)\hat{c}_t(x)|\vec{p}_t] \qquad \text{(focus on round } t\text{)} \\
&= \sum_x p_t \, \mathbb{E}[\hat{c}_t(x)|\vec{p}_t] \qquad \text{(take } \vec{p}_t \text{ out of conditioning)} \\
&= \sum_x p_t(x) \; c_t(x) \qquad \text{(use (3))} \\
&= \mathbb{E}[c_t(x_t) \mid \vec{p}_t].
\end{aligned}$$

Taking expectation of both sides implies that $\mathbb{E}[\hat{c}_t(x_t)] = \mathbb{E}[c_t(x_t)]$. Summing over all rounds, we obtain (4).

Second, we deal with the benchmark:

$$\mathbb{E}[\min_x \widehat{\texttt{cost}}(x)] \leq \min_x \mathbb{E}[\widehat{\texttt{cost}}(x)] = \min_x \mathbb{E}[\texttt{cost}(x)] = \min_x \texttt{cost}(x).$$

The first equality holds because by (3), and the second equality holds because true costs are deterministic. Combining this with (4) implies the claim. □

Note that we used the "full power" of assumption (3): a weaker assumption $\mathbb{E}[\hat{c}_t(x)] = \mathbb{E}[c_t(x)]$ would not have sufficed to argue about true vs. fake costs of `Hedge`.

5 Algorithm `Exp4` and crude analysis

Let us complete the specification of Algorithm 1. To define fake costs, we will use a standard trick in statistics called Inverse Propensity Score (IPS). Whatever the way arm a_t is chosen in each round t given the probability distribution $\vec{p}_t$ returned by `Hedge`, this defines distribution q_t over arms a:

$$q_t(a) = \Pr[a_t = a \mid \vec{p}_t].$$

Using these probabilities, we define the fake costs on each arm as follows:

$$\hat{c}_t(a) = \begin{cases} \frac{c_t(a_t)}{q_t(a_t)} & a_t = a \\ 0 & \text{otherwise.} \end{cases}$$

This step is well-defined as long as the algorithm can compute probability $q_t(a_t)$, because the algorithm knows the true cost $c_t(a_t)$ for the chosen arm a_t. The fake cost on each expert x is defined as the fake cost of the arm chosen by this expert: $\hat{c}_t(x) = \hat{c}_t(a_{t,x})$.

This is indeed an unbiased estimator for true costs:

Claim 5.1. (3) *holds if* $q_t(a) > 0$ *for all arms* a.

Proof. Let us argue about each arm a separately:

$$\mathbb{E}[\hat{c}_t(a) \mid \vec{p}_t] = q_t(a) \cdot \frac{c_t(a_t)}{q_t(a)} + \Pr[a_t \neq a] \cdot 0 = c_t(a).$$

Now, for a given expert x plug in arm $a = a_{t,x}$, its choice in round t. □

Thus, we need to ensure that $q_t(a) > 0$ for all arms a, *i.e.,* that each arm is chosen with a strictly positive probability, regardless of which expert x_t is chosen by `Hedge`. On the other hand, we'd like to follow expert x_t with high probability so as to ensure low cost in this round.

Therefore, we complete the specification of our algorithm as follows:

- in each round t, with some low probability $\gamma \in (0, \frac{1}{2})$, we pick an arm independently and uniformly at random. Otherwise, follow the advice of expert x_t.

This algorithm is known as `Exp4`. It is parameterized by $\gamma \in (0, \frac{1}{2})$.

Note that $q_t(a) \geq \gamma/K > 0$ for each arm a, as needed. Combining this with Claim 5.1 and Claim 4.1, we obtain $\mathbb{E}[R_{\texttt{Hedge}}(T)] \leq \mathbb{E}[\widehat{R}_{\texttt{Hedge}}(T)]$.

In each round, our algorithm accumulates cost at most 1 from the low-probability exploration, and cost $c_t(x_t)$ from the chosen expert x_t. So the expected cost in this round is at most $\gamma + c_t(x_t)$, which implies that $\texttt{cost}(\texttt{Exp4}) \leq \texttt{cost}(\texttt{Hedge}) + \gamma T$.

To summarize, we have made sure that:

Lemma 5.2. $\mathbb{E}[R_{\texttt{Exp4}}(T)] \leq \mathbb{E}[\widehat{R}_{\texttt{Hedge}}(T)] + \gamma T$.

For a crude regret bound, observe that $c_t(a) \leq 1/q_t(a) \leq K/\gamma$. So we can immediately upper-bound $\mathbb{E}[\widehat{R}_{\texttt{Hedge}}(T)]$ using the regret bound for `Hedge` with $U = K/\gamma$. Then:

$$\mathbb{E}[R_{\texttt{Exp4}}(T)] \leq O(\sqrt{(K/\gamma)\, T \log N} + \gamma T).$$

To minimize regret, chose γ to (approximately equalize the two summands:

Theorem 5.3. *Consider adversarial bandits with expert advice. Algorithm* `Exp4` *with parameter* $\gamma = T^{-1/3}\, (K \log N)^{1/3}$ *achieves regret*

$$\mathbb{E}[R(T)] = O(T^{2/3}\, (K \log N)^{1/3}).$$

6 Improved analysis

We can obtain a better regret bound by analyzing the quantity

$$\widehat{G}_t := \sum_x p_t(x)\, \hat{c}_t^2(x).$$

We will prove an upper bound on $\mathbb{E}[G_t]$, and use the corresponding regret bound for `Hedge`.

Lemma 6.1. *Fix parameter* $\gamma \in (0, \frac{1}{2})$ *and round* t. *Then* $\mathbb{E}[G_t] \leq \frac{K}{1-\gamma}$.

Proof. For each action a, let $X_a = \{x \in X : a_{t,x} = a\}$ be the set of all experts that recommended that action. Let

$$p_t(a) := \sum_{x \in X_a} p_t(x)$$

be the probability that the expert chosen by `Hedge` recommends action a. Then

$$q_t(a) = p_t(a)(1-\gamma) + \frac{\gamma}{K} \geq (1-\gamma)\, p_t(a).$$

For each expert x, letting $a = a_{t,x}$ be the recommended action, we have:

$$\hat{c}_t(x) = \hat{c}_t(a) \leq \frac{c_t(a)}{q_t(a)} \leq \frac{1}{q_t(a)} \leq \frac{1}{(1-\gamma)\, p_t(a)}.$$

Each realization of G_t satisfies:

$$\begin{aligned}
\widehat{G}_t &= \sum_a \sum_{x \in X_a} p_t(a) \cdot \hat{c}_t(x) \cdot \hat{c}_t(x) && \text{(re-write as a sum over arms)} \\
&\leq \sum_a \sum_{x \in X_a} \frac{p_t(x)}{(1-\gamma)\, p_t(a)}\, \hat{c}_t(a) && \text{(replace one } \hat{c}_t(a) \text{ with an upper bound)} \\
&= \frac{1}{1-\gamma} \sum_a \frac{\hat{c}_t(a)}{p_t(a)} \sum_{x \in X_a} p_t(x) && \text{(move ``constant terms'' out of the inner sum)} \\
&= \frac{1}{1-\gamma} \sum_a \hat{c}_t(a) && \text{(the inner sum is just } p_t(a)\text{)}
\end{aligned}$$

To complete the proof, take expectations over both sides and recall that $\mathbb{E}[\hat{c}_t(a)] = c_t(a) \leq 1$. □

Thus, we can use the regret bound for `Hedge` with $U = K/(1-\gamma)$. This is a big improvement over $U = K/\gamma$ that we used in the proof of Theorem 5.3. Let us complete the analysis, being a little careful so as to derive a better constant:

$$\begin{aligned}
\mathbb{E}[\widehat{R}_{\texttt{Hedge}}(T)] &\leq 2\sqrt{3/(1-\gamma)} \cdot \sqrt{TK \log N} \\
\mathbb{E}[R_{\texttt{Exp4}}(T)] &\leq 2\sqrt{3/(1-\gamma)} \cdot \sqrt{TK \log N} + \gamma T && \text{(by Lemma 5.2)} \\
&\leq 2\sqrt{3} \cdot \sqrt{TK \log N} + 2\gamma T && \text{(since } \sqrt{1/(1-\gamma)} \leq 1+\gamma\text{)} \qquad (5)
\end{aligned}$$

(To derive (5), we assumed w.l.o.g. that $2\sqrt{3} \cdot \sqrt{TK \log N} \leq T$.)

This holds for any $\gamma > 0$. Therefore:

Theorem 6.2. *Consider adversarial bandits with expert advice. Algorithm* `Exp4` *with parameter* $\gamma \in (0, 1/2T)$ *achieves regret*

$$\mathbb{E}[R(T)] \leq 2\sqrt{3} \cdot \sqrt{TK \log N} + 1.$$

7 Remarks

`Exp4` stands for EXPloration, EXPloitation, EXPonentiation, and EXPerts. The version for adversarial bandits without expert advice (*i.e.,* with experts that correspond to arms) is called `Exp3`. Both algorithms were introduced (and named) in the seminal paper (Auer et al., 2002b), along with several extensions and the $\Omega(\sqrt{KT})$ lower bound.

The running time for `Exp3` is very nice because in each round, we only need to do a small amount of computation to update the weights. However, in `Exp4`, the running time is $O(N + K)$ per round, which can become very slow when N, the number of experts, is very large.

For several important special cases it is possible to obtain good regret *and* good running time, via different algorithms. One way to accomplish that is to reduce to a best-expert algorithm other than `Hedge` that obtains small regret and is computationally efficient where `Hedge` is not. In fact, the reduction described above, and the analysis leading up to Theorem 5.3, still applies if `Hedge` is replaced with an arbitrary best-experts algorithm for which we have a regret bound in terms of (K, T, U) (where U is a known upper bound on the costs). We will follow this idea next class: we will apply the same reduction to a different best-experts algorithm to obtain good regret and fast running time for an interesting special case.

`Exp4` is a really powerful algorithm, and can be applied in many different settings:

- *contextual bandits:* we will see this application later in this course.

- *shifting regret:* In adversarial bandits, rather than compete with the best fixed arm, we can compete with "policies" that can change the arm from one round to another, but not too often. More formally, an *S-shifting policy* is sequence of arms $\pi = (a_t : t \in [T])$ with at most S "shifts": rounds t such that $a_t \neq a_{t+1}$. *S-shifting regret* is defined as the algorithm's total cost minus the total cost of the best S-shifting policy:

$$R_S(T) = \texttt{cost}(\texttt{ALG}) - \min_{S\text{-shifting policies } \pi} \texttt{cost}(\pi).$$

  Consider this as a bandit problem with expert advice, where each S-shifting policy is an expert. The number of experts $N \leq (KT)^S$; while it may be a large number, $\log(N)$ is not too bad! Using `Exp4` and plugging $N \leq (KT)^S$ into Theorem 6.2, we obtain

$$\mathbb{E}[R_S(T)] = O(\sqrt{KST \cdot \log(KT)}). \tag{6}$$

- *Slowly changing costs:* Consider a randomized oblivious adversary such that the expected cost of each arm can change by at most ϵ in each round. Rather than compete with the best fixed arm, we'd like to compete with the (cost of) the best *current* arm: $c_t^* = \min_a c_t(a)$. More formally, we'd like to minimize *dynamic regret*, defined as

$$R^*(T) = \min(\texttt{ALG}) - \sum_{t=1}^{T} c_t^*.$$

  (Note that dynamic regret is the same as T-shifting regret.) One way to solve this problem is via S-shifting regret, for an appropriately chosen value of S.

8 Extensions of adversarial bandits

Much research has been done on various extensions of adversarial bandits. Some of these extensions are briefly discussed below.

- a stronger version of the same results: *e.g.,* extend to adaptive adversaries, obtain regret bounds that hold with high probability (rather than merely in expectation), and improve the constants. (All of the above has been done in the original paper Auer et al. (2002b)).
- an algorithm with $O(\sqrt{KT})$ regret – shaving off that $\log K$ factor and matching the lower bound up to constant factors – has been achieved in Audibert and Bubeck (2010).
- improved results for shifting regret: while applying `Exp4` is computationally inefficient, Auer et al. (2002b) obtain the same regret bound (6) via a modification of `Exp3` (and a more involved analysis), with essentially the same running time as `Exp3`. Further, a version of that algorithm achieves the same regret bound multiplied by $\sqrt{S}$ which holds *for all S at once*.
- While we have only considered a finite number of experts and made no assumptions about what these experts are, similar results can be obtained for infinite classes of experts with some special structure. In particular, borrowing the tools from statistical learning theory, it is possible to handle classes of experts with a small VC-dimension.
- The notion of "best fixed arm" is not entirely satisfying for adaptive adversaries. An important line of research on adversarial bandits (*e.g.,* Dekel et al., 2012) considers notions of regret in which the benchmark is "counterfactual": it refers not to the costs realized in a given run of the algorithm, but to the costs that would have been realized had the algorithm do something different.
- For adversarial bandits with slowly changing costs, there are algorithms with better regret bounds (than those from an application of `Exp4`), and fast running times (Slivkins and Upfal, 2008; Slivkins, 2014). Also, they handle more general versions of slowly changing costs, *e.g.,* allow the expected cost of each arm to evolve over time as an independent random walk on a bounded interval.
- Lastly, *data-dependent* regret bounds are desirable: not only optimal in the worst case, but also better for some "nice" special cases. In particular, Hazan and Kale (2009) obtain an improved regret bound when the realized cost of each arm a does not change too much compared to its average $\texttt{cost}(a)/T$: namely, the regret bound is of the form $\tilde{O}(\sqrt{V})$, where $V = \sum_{t,a} \left(c_t(a) - \frac{\texttt{cost}(a)}{T}\right)^2$ is the *total variation* of the costs. Note, however, that IID 0-1 costs is essentially the worst case, as far as this particular regret bound is concerned.

  Bubeck and Slivkins (2012); Seldin and Slivkins (2014) pursue a different direction: they design algorithms that (essentially) match the regret of `Exp3` in the worst case, and achieve logarithmic regret (as `UCB1`) if the costs are actually IID.

9 Bibliographic notes

The material in this lecture is presented in the original paper Auer et al. (2002a), as well as in various books and courses on online learning (e.g., Cesa-Bianchi and Lugosi, 2006; Bubeck and

Cesa-Bianchi, 2012). Among these sources, my presentation was most influenced by (Kleinberg, Spring 2007), but I made the "reduction to `Hedge` " even more explicit.

References

J.Y. Audibert and S. Bubeck. Regret Bounds and Minimax Policies under Partial Monitoring. *J. of Machine Learning Research (JMLR)*, 11:2785–2836, 2010.

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

Peter Auer, Nicolò Cesa-Bianchi, Yoav Freund, and Robert E. Schapire. The nonstochastic multiarmed bandit problem. *SIAM J. Comput.*, 32(1):48–77, 2002b. Preliminary version in *36th IEEE FOCS*, 1995.

Sébastien Bubeck and Nicolo Cesa-Bianchi. Regret Analysis of Stochastic and Nonstochastic Multi-armed Bandit Problems. *Foundations and Trends in Machine Learning*, 5(1), 2012.

Sébastien Bubeck and Aleksandrs Slivkins. The best of both worlds: stochastic and adversarial bandits. In *25th Conf. on Learning Theory (COLT)*, 2012.

Nicolò Cesa-Bianchi and Gábor Lugosi. *Prediction, learning, and games*. Cambridge Univ. Press, 2006.

Ofer Dekel, Ambuj Tewari, and Raman Arora. Online bandit learning against an adaptive adversary: from regret to policy regret. In *29th Intl. Conf. on Machine Learning (ICML)*, 2012.

Elad Hazan and Satyen Kale. Better algorithms for benign bandits. In *20th ACM-SIAM Symp. on Discrete Algorithms (SODA)*, pages 38–47, 2009.

Robert Kleinberg. Lecture notes: *CS683: Learning, Games, and Electronic Markets* (week 8), Spring 2007. Available at `http://www.cs.cornell.edu/courses/cs683/2007sp/lecnotes/week8.pdf`.

Yevgeny Seldin and Aleksandrs Slivkins. One practical algorithm for both stochastic and adversarial bandits. In *31th Intl. Conf. on Machine Learning (ICML)*, 2014.

Aleksandrs Slivkins. Contextual bandits with similarity information. *J. of Machine Learning Research (JMLR)*, 15(1):2533–2568, 2014. Preliminary version in *COLT 2011*.

Aleksandrs Slivkins and Eli Upfal. Adapting to a changing environment: the Brownian restless bandits. In *21st Conf. on Learning Theory (COLT)*, pages 343–354, 2008.