Multi-armed bandits and reinforcement learning

Lecture 6: Linear Bandits

Instructor: Shipra Agrawal Scribes contributed by: Mauro Escobar, Yuanjun Gao

In this lecture we will study the case where the number of arms is much bigger than the number of time periods, i.e., $N \gg T$. Intuitively, this seems a difficult problem without further assumptions, because every arm needs to be explored at least once. The conceptual idea behind handling large number of arms is to drop the assumption of unrelated arms, and take advantage of the relation between them – playing one arm will give information about "similar" arms, thus reducing the exploration required. The assumption of linear rewards in linear bandit model will impose one specific similarity structure between arms. There are many other models, for example, convex bandits, general metric similarity structures, spectral bandits.

1 Linear Bandits

Consider N arms, $N \gg T$. For arm every i, we are given a vector $x_i \in \mathbb{R}^d$. On pulling arm i at time t, we observe r_t such that

$$\mathsf{E}\left[r_t \,|\, I_t = i\right] \;=\; x_i^\top \omega, \qquad \text{where } \omega \in \mathbb{R}^d \text{ is fixed, but unknown.}$$

To see that this model imposes a similarity structure which can be taken advantage of, consider following example. Let

$$x_1 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \qquad x_2 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \qquad x_3 = \begin{pmatrix} 0 \\ 0.5 \end{pmatrix}, \qquad x_4 = \begin{pmatrix} 1 \\ 0 \end{pmatrix},$$

pulling arm 1 tells us: some information about pulling arm 2, everything about pulling arm 3, and nothing about pulling arm 4.

Definition 1 (Regret)**.** *For linear bandits, we define the regret as follows,*

$$R(T) \;=\; T \cdot \left(\max_{i=1,\ldots,N} x_i^\top \omega\right) - \sum_{t=1}^{T} r_t.$$

Since in this model, an arm is completely defined by the corresponding vector x_i, instead of considering N arms, we can index the arms as all vectors in a set $A \subset \mathbb{R}^d$. This way of formulating the problem removes the requirement of finite, or even countably many arms.

Then, at time t, the algorithm needs to pick a vector $x_t \in A$, and observe r_t such that $\mathsf{E}\left[r_t \,|\, x_t\right] = x_t^\top \omega$. In this case, the regret becomes

$$R(T) \;=\; T \cdot \left(\max_{x \in A} x^\top \omega\right) - \sum_{t=1}^{T} x_t^\top \omega.$$

We consider a generalization of this problem, where there is an arbitrary sequence of subsets $A_1, A_2, \ldots, A_T \subseteq A$, fixed in advance, but unknown to the decision making algorithm. At time t, the algorithm first observes A_t, and then it needs to pick some $x_t \in A_t$. And regret is defined as,

$$R(T) \;=\; \sum_{t=1}^{T} \left(\max_{x \in A_t} x^\top \omega\right) - \sum_{t=1}^{T} x_t^\top \omega.$$

2 Applications

2.1 Route optimization

Consider a graph G with n nodes and d edges. Each arm is a possible path in the graph, then, the number of arms could be exponentially large. We consider the following setup:

- $x \in \mathbb{R}^d$: is the incidence vector of a path ($x_e = 1$ if edge e belongs to the path, and $x_e = 0$ otherwise),
- $A \subset \mathbb{R}^d$: is the collection of all incidence vectors of paths in the graph, $|A|$ is the number of valid paths,
- $\omega \in \mathbb{R}^d$: is such that ω_e is the delay of using the edge e.

Then, the delay of a path P with incidence vector x is $\sum_{e \in P} \omega_e = x^\top \omega$. Observe that using generalization to a different set A_t, we can now model the problem where at every time step t, route between a different source-destination pair (s_t, d_t) needs to be picked.

2.2 Movie recommendations

We consider that vector represent movie features, such as cast, genre, studio, etc.

- $x \in \mathbb{R}^d$: movie features vector (d features),
- $A \subset \mathbb{R}^d$: set of all possible feature vectors for movies.

3 LinUCB Algorithm

Recall the UCB Algorithm:

Algorithm 1 UCB Algorithm

for $t = 1, 2, \ldots, T$ **do**

1. For each arm i, build estimates $\hat{\mu}_{i,t-1} = \frac{1}{n_{i,t}} \sum_{s \le t-1 \,:\, T_s = i} r_s$,
2. For each arm i, build confidence intervals, such that
$$\mu_i \in \left[\hat{\mu}_{i,t-1} - \sqrt{\frac{\log t}{n_{i,t-1}}}, \hat{\mu}_{i,t} + \sqrt{\frac{\log t}{n_{i,t-1}}} \right] \qquad \text{w.p. } 1 - \frac{2}{T^2},$$
3. For each arm i, pick the optimistic estimate $\text{UCB}_{i,t-1} := \hat{\mu}_{i,t-1} + \sqrt{\frac{\log t}{n_{i,t-1}}}$,
4. Play arm $I_t = \underset{i=1,\ldots,N}{\text{argmax}}\ \text{UCB}_{i,t-1}$.

end for

We will adequately modify this algorithm to get LinUCB algorithm for linear bandits.

LinUCB:

Step 1: Given the history up to time τ: $(r_1, x_1), (r_2, x_2), \ldots, (r_\tau, x_\tau)$, we want to solve

$$\hat{\omega}_\tau \;=\; \underset{z \in \mathbb{R}^d}{\text{argmax}} \left\{ \sum_{t=1}^{\tau} (r_t - x_t^\top z)^2 + \|z\|^2 \right\},$$

which solution is

$$\hat{\omega}_\tau \;=\; M_\tau^{-1} y_\tau,$$

where $M_\tau = \mathbf{I}_{d\times d} + \sum_{t=1}^{\tau} x_t x_t^\top$ and $y_\tau = \sum_{t=1}^{\tau} r_t x_t$.

As a sanity check: consider the N-armed bandit problem. It can be modeled as linear bandit with $x_t = \mathrm{e}_{I_t}$ (the I_t-th canonical vector) for all t, then,

$$M_\tau = \mathbf{I} + \sum_{t=1}^{\tau} x_t x_t^\top = \begin{bmatrix} n_{1,\tau}+1 & & \\ & \ddots & \\ & & n_{d,\tau}+1 \end{bmatrix} \quad \text{and} \quad y_{\tau,i} = \sum_{s\le\tau\,:\,I_s=i} r_s, \qquad \text{therefore,} \quad \hat{\omega}_\tau = \begin{pmatrix} \hat{\mu}_{1,\tau} \\ \vdots \\ \hat{\mu}_{d,\tau} \end{pmatrix}.$$

Step 2: Using exponential inequality for ratios and martingales, the following theorem can be proved.

Theorem 2 (Rusmevichientong, Tsitsiklis, 2010. Abbasi-Yadkori et al., 2011)**.** *If $\|x_t\|_2 \le \sqrt{Ld}$, $\|\omega\|_2 \le \sqrt{d}$ and $|r_t| \le 1$. Then, with probability at least $1-\delta$, the vector ω lies on the set*

$$C_t = \left\{ z \in \mathbb{R}^d \,:\, \|z - \hat{\omega}_t\|_{M_t} \le \sqrt{d \log\left(\frac{TdL}{\delta} + 1\right) + \sqrt{d}} \right\}^{1}.$$

Check that this bound will recover the UCB confidence interval within $\sqrt{d}$ in the special case of N-armed bandit problem modeled as linear bandit.

Step 3: For every $x \in \mathbb{R}^d$, we want to find UCB(x) such that UCB$(x) \ge x^\top \omega$. Define

$$\mathrm{UCB}(x) \;:=\; \operatorname*{argmax}_{z\in C_t} x^\top z.$$

Step 4: At time t, pick

$$\operatorname*{argmax}_{x\in A_t} \max_{z\in C_t} z^\top x.$$

Solving the double maximization problem of step 4 is difficult when number of arms is large (NP-hard even when sets A_t are convex).

We will show that this algorithm achieves an $\tilde{O}(d\sqrt{T})$ regret bound. [1] shows a modification to get an efficient algorithm with regret bound of $\tilde{O}(d^{3/2}\sqrt{T})$.

[1] Matrix norm: $\|x\|_M = \sqrt{x^\top M x}$.

3.1 Regret analysis (sketch)

For the regular multi-armed bandit setting, we provide a sketch for a simple analysis for the order of the regret.

$$
\begin{aligned}
R(T) =& \sum_{t=1}^{T} (\mu_t^* - \mu_{I_t}) && (1)\\
\leq& \sum_{t=1}^{T} \mathrm{UCB}_{I_t^*,t-1} - \mu_{I_t} && (2)\\
\leq& \sum_{t=1}^{T} \mathrm{UCB}_{I_t,t-1} - \mu_{I_t} && (3)\\
=& \sum_{i=1}^{N} \sum_{t:I_t=i} \sqrt{\frac{\log T}{n_{i,t-1}}} && (4)\\
=& \sum_{i=1}^{N} \sum_{k=1}^{N_{i,T}} \sqrt{\frac{\log T}{k}} && (5)\\
=& \sqrt{\log T} \sum_{i} \sqrt{n_{i,T}} && (6)\\
\leq& \sqrt{\log T}\sqrt{NT} && (7)
\end{aligned}
$$

(1) comes from definition. (2) hold with high probability since $\mathrm{UCB}_{I_t^*,t-1} > \mu_t^*$ with high probability. (3) holds by definition of the UCB algorithm (i.e. we pick the bandit with the highest UCB). (4) holds because $UCB - \mu$ are bounded by $\sqrt{\frac{\log T}{n_{I_t,t-1}}}$. (5) is a rearrangement of (4) by noting that each time arm i has the highest UCB, it will be pulled one more time, so $n_{i,t}$ increases by 1. (6) uses $\sum_{i=1}^{n} \frac{1}{\sqrt{i}} = O(\sqrt{n})$. (7) holds because $n_{i,T} = \frac{T}{n}$ gives the worst case.

We adapt this idea to the linear bandit case by noting

$$
\begin{aligned}
R(T) \leq& \sum_{t=1}^{T} w^\top x_t^* - w^\top x_t \\
=& \sum_{t=1}^{T} \mathrm{UCB}_{x_t^*,t-1} - w^\top x_t \\
\leq& \sum_{t=1}^{T} \mathrm{UCB}_{x_t,t-1} - w^\top x_t && (8)
\end{aligned}
$$

Here we have $\mathrm{UCB}_{x_t,t} = z^\top{}_{t-1} x_t$ for some $z_{t-1} \in C_t$, where $\|z_{t-1} - w\|_{M_t} \leq 2\sqrt{d \log(dT/\delta)}$ with probability $1-\delta$. We proceed by

$$
\begin{aligned}
(8) =& \sum_{t=1}^{T} z'_{t-1} x_t - w' x_t \\
\leq& \sum_{t=1}^{T} \|z_{t-1} - w\|_{M_{t-1}} \|x_t\|_{M_{t-1}^{-1}} && (9)\\
\leq& 2\sqrt{d \log(dT/\delta)} \sum_{t=1}^{T} \|x_t\|_{M_{t-1}^{-1}} && (10)
\end{aligned}
$$

Here (9) comes from Cauchy-Schwarz inequality ($|x^\top w| \leq \|x\|_{M^{-1}} \|w\|_M$). (10) is because, as mentioned above,

$\|z_{t-1} - w\|_{M_t} \leq 2\sqrt{d\log(Td/\delta)}$ holds with probability $1-\delta$.

Now we want to get something similar to (6) to bound the summation $\sum_{t=1}^{T} \|x_t\|_{M_{t-1}^{-1}} = \sum_{t=1}^{T} \sqrt{x_t' M_{t-1}^{-1} x_t}$. The tricky thing is that although M_t keeps increasing, there are many directions in $M_t \in \mathbb{R}^{d\times d}$, so even for large t, if x_t is in the direction of a eigenvector of M_{t-1} with a small eigenvalue, $\|x_t\|_{M_{t-1}^{-1}}$ can still be large. Fortunately, we have the following lemma

Lemma 3. *(Lemma 11 of [3], or, Lemma 2 of [4]) Denote $\lambda_{j,t-1}$ as the j^{th} largest eigenvalue of M_{t-1}, then eigenvalues of M_t can be arranged so that $\lambda_{j,t} \geq \lambda_{j,t-1}$, and we have*

$$\|x_t\|^2_{M_{t-1}^{-1}} \leq 10 \sum_{j=1}^{d} \frac{\lambda_{j,t} - \lambda_{j,t-1}}{\lambda_{j,t-1}}$$

Intuitively, this lemma shows that if x_t is in the direction of a eigenvector of M_{t-1} with a small eigenvalue, then, it will sufficiently increase that eigenvalue, which would benefit that direction in the next time step. Therefore, in any direction we will get decreasing terms in the summation. More precisely, we have

$$(10) \leq 2\sqrt{d\log(Td/\delta)} \sum_{t=1}^{T} \sqrt{\sum_j \left(\frac{\lambda_{j,t}}{\lambda_{j,t-1}} - 1\right)} \tag{11}$$

The remaining analysis involves considering the worst possible value (to maximize above expression) of $\lambda_{j,t}, j, t$ under the constraint $\sum_j \Pi_{t=1}^{T} \frac{\lambda_{j,t}}{\lambda_{j,t-1}} = \sum_j \lambda_{j,T} \leq T$, and $\frac{\lambda_{j,t}}{\lambda_{j,t-1}} \geq 1$. It can be shown (refer to [4]: Lemma 3 in Section 5) that at maximizer $h_{tj} := \frac{\lambda_{j,t}}{\lambda_{j,t-1}}$ are equal for all t, j and $\sum_{t=1}^{T} \sqrt{\sum_j \left(\frac{\lambda_{j,t}}{\lambda_{j,t-1}} - 1\right)} \leq O(\sqrt{dT\ln(T)})$, so that assuming $d \leq T$

$$(10) \leq O(\sqrt{d\log(Td/\delta)}\sqrt{dT\ln(T)}) = O(d\sqrt{T\log^2(T/\delta)}) \tag{12}$$

This proves that regret of this UCB algorithm for linear bandits is

$$R(T) \leq O(d\sqrt{T\log^2(T/\delta)})$$

with probability $1-\delta$.

4 Gradient descent based algorithm

We demonstrate a nicer gradient descent based algorithm. This algorithm will work even when after picking x_t, the reward is $r_t = x_t \top w_t$ for some arbitrary unknown w_t at time t. However, it only works for linear bandits and not contextual bandits, i.e., set of contexts are not allowed to change over time. Also, the regret guarantees obtained will be slightly weaker.

More precisely, here we want to pick $x_t \in A$ each time to maximize the reward (here A is not time-varying, and we assume it to behave well. For example, we assume that it is a convex set), and we assume that at each time our expected reward is $x_t^\top w_t$, where the weight w_t can change across time (and have no pattern) In this case we compare our strategy with the best strategy that keeps pulling the best arm in A. So we define regret as

$$R(T) = \left(\max_{x\in A} \sum_t x^\top w_t\right) - \sum_{t=1}^{T} x_t^\top w_t$$

4.1 Full information setting

Under the full information setting, we observe w_t after picking x_t. We can use the simple idea of online linear optimization by gradient ascent. Notice that the reward is $r_t(x) = x^\top w_t$, so the gradient is simply $\frac{dr_t(x)}{dx} = w_t$. Therefore we want our x_t go in the direction of w_t a little. Therefore we update our choice by

$$x_t = \Pi_A(x_{t-1} + \eta w_{t-1})$$

where Π_A is the projection operator and η is a constant step-size. We have

Algorithm 2 Gradient Ascent Algorithm for Full Information Linear Bandit under Adversarial Case

Input: $\eta > 0$

for $t = 1, 2, ...$ **do**

1. Set $x_t = \Pi_A(x_{t-1} + \eta w_{t-1})$
2. Play arm x_t, observe reward r_t and w_t

end for

Theorem 4. *Under the full information setting, assuming $\|w_t\| \leq \sqrt{d}$, $\forall x \in A, \|x\| \leq \sqrt{d}$, then using the gradient ascent algorithm with $\eta = \frac{1}{\sqrt{T}}$, we have*

$$R(T) \leq d\sqrt{T}$$

More generally, suppose $\|w_t\| \leq D$, $\forall x \in A, \|x\| \leq G$, then we have

$$R(T) \leq DG\sqrt{T}$$

Note that in the adversarial N-armed bandit setting, we have a $\sqrt{T \log N}$ bound.

4.2 Bandit setting

Suppose that instead of observing w, we only get to observe $r_t = w_t^\top x_t$ after picking x_t, we can adapt the gradient ascent algorithm for the full information setting by using an unbiased estimator for w_{t-1}. Here, instead of pulling x_t, we perturb it a little by a random walk. Specifically, we generate a random vector $u \in \mathbb{R}^d$, where each element u_i is generated independently and equals 1 or -1 with probability $1/2$. Then we pull arm $x_t + \delta u$ to get the reward. Interestingly, this random perturbation gives us an unbiased estimator for w

Claim 5. *The $\hat{w}_t$ defined below is an unbiased estimator of w_t*

$$\hat{w}_t = w_t^\top \frac{(x_t + \delta u)u}{\delta}$$

Proof.

$$\begin{aligned} E[\hat{w}_t] =& E[\frac{w_t^\top x_t u}{\delta}] + E[uu^\top w_t] \\ =& 0 + E[I_d w_t] \\ =& 0 + w_t \end{aligned}$$

Here we use the fact that $E(u) = 0$ (since it is a random walk) and $E[uu^\top] = I_d$ (since u_i are independent and $u_i^2 = 1$ with probability 1) □

So in sum, in the bandit setting, we use the following update rule

$$x_t = \Pi_A(x_{t-1} + \eta \hat{w}_{t-1})$$

Algorithm 3 Gradient Ascent Algorithm for Linear Bandit Setting under Adversarial Case

Input $\eta > 0$, $\delta > 0$

for t = 1,2,... **do**

1. $x_t = \Pi_A(x_{t-1} + \eta \hat{w}_{t-1})$
2. Play $x_t + \delta u_t$, where u_t is a random vector
3. Observe r_t
4. Define $\hat{w}_t = w_t^\top \frac{(x_t+\delta u)u}{\delta}$

end for

But each time we actually pull $x_t + \delta u_t$ for a random vector u_t.

By using an unbiased estimator instead of the true w_t, we sacrifice in the following two ways.

First the $\hat{w}_t$ can be big, so we have to increase the D in theorem 4. Actually now we have $D = \frac{\sqrt{d}}{\delta} \geq \|\hat{w}_t\|$, which increase the bound by $\frac{1}{\delta}$ fold. Giving us $\frac{d\sqrt{T}}{\delta}$

Secondly, instead of pulling x_t, we have the random perturbation δu, which adds an extra regret

$$\sum_t \delta u_t^\top w_t \leq \delta dT$$

This is because we have

$$w_t^\top (x_t + \delta u) \leq w_t^\top x_t - \delta |w_t^\top u|$$

but $|w_t^\top u| \leq d$ because $\|u\| = \sqrt{d}$ and we assume $\|w_t\| \leq \sqrt{d}$.

Combining the above two point together, we get a bound of the form

$$\frac{d\sqrt{T}}{\delta} + \delta dT$$

By setting $\delta = \frac{1}{T^{1/4}}$, we get a lower bound of

$$R(T) = O(dT^{3/4})$$

The optimal lower bound has been proved to be $\Omega(d\sqrt{T})$, which cannot be achieved by the algorithm stated above. The algorithm with the optimal rate involves a much more complicated algorithm.

[5] provides analysis of online gradient ascent algorithm for full information setting, [6] extends it to bandit setting. [7] provides efficient algorithm which achieves a regret upper bound with optimal dependence of $\sqrt{T}$ on time horizon T.

References

[1] Stochastic Linear Optimization under Bandit Feedback, Varsha Dani, Thomas P. Hayes, Sham M. Kakade, COLT 2008.

[2] Improved Algorithms for Linear Stochastic Bandits, Yasin Abbasi-yadkori, Dvid Pl, Csaba Szepesvri, NIPS 2011.

[3] Using Confidence Bounds for Exploitation-Exploration Trade-offs, Peter Auer. JMLR 3(Nov):397-422, 2002.

[4] Contextual Bandits with Linear Payoff Functions. Wei Chu. Lihong Li. Lev Reyzin. Robert E. Schapire. AISTATS 2011.

[5] Online Convex Programming and Generalized Infinitesimal Gradient Ascent, Martin Zinkevich. ICML 2003.

[6] Online convex optimization in the bandit setting: gradient descent without a gradient, Abraham D. Flaxman, Adam Tauman Kalai, H. Brendan McMahan. SODA 2005.

[7] Competing in the Dark: An Efficient Algorithm for Bandit Linear Optimization by Jacob Abernethy , Elad Hazan , Alexander Rakhlin, COLT 2008.