STATS 710 – Seq. Decision Making with mHealth Applications Oct 20, 2016

Lecture 13: VE algorithm and regret analysis

Instructors: Susan Murphy and Ambuj Tewari *Scribe: Lauren Steimle*

1 Recap from Previous Classes

Exp3

- Multi-armed bandit setting
- Non-stochastic setting with bounded losses. That is, the losses, $\ell_{i,t}$, are arbitrary numbers in $[0, 1]$.
- Regret is defined with respect to the best fixed action in hindsight
- Regret bound was $\sqrt{2KT \log K}$ where K is the number of actions.

Exp4

- Multi-armed bandit setting or contextual bandit setting.
- Non-stochastic with bounded losses.
- New definition of regret. That is, regret is defined with respect to the best fixed "expert" in hindsight where an expert is a source of a probability distribution over $\mathcal{A}$ at any given time. For example, experts could be selecting one action the entire time (i.e. Exp3) or policies in a contextual bandit
- Regret bound of $\sqrt{2KT \log N}$ where N is the number of experts

Epoch Greedy

- Contextual bandit algorithm for stochastic problems. That is, there is some distribution on the contexts and rewards, $(\mathcal{X}, R^a) \overset{iid}{\sim} \mathcal{D}$
- It can compete with any policy class, Π, with VC-dim$(\Pi) < \infty$.
- It achieves expected regret of $O(T^{2/3}(\text{VC-dim}(\Pi))^{1/3})$.
- It is computationally efficient provided the following optimization can be efficiently performed:

$$\operatorname*{argmax}_{\pi \in \Pi} \hat{V}(\pi), \text{ where}$$

$$\hat{V}(\pi) = \frac{1}{|D|} \sum_{(x,a,r) \in D} 2r \mathbb{1}[\pi(x) = a]$$

  where D is the current dataset collected during exploration rounds.
- Epoch greedy is a nice algorithm in the sense that it reduces the contextual bandit problem into a computational cost-sensitive classification problem.

2 VC-classes and Exp4 (VE) Algorithm [BLL+11]

Today, we are going to improve the regret rate obtained by Epoch Greedy. We'll start from a VC-class and consider the case where $K = 2$. A lot of this theory extends to general K actions, but then we would need to consider multi-class classification and VC-dimension is no longer the right object to measure the complexity.

Goal: To design an algorithm, called the VE algorithm, whose expected regret will scale as $O(\sqrt{Td\log(T/d)})$ where $d = \text{VC-dim}(\Pi)$.

Setting: Stochastic contextual bandits with bounded losses, $(X_t, (\ell_{a,t})_{a\in\mathcal{A}}) \overset{iid}{\sim} \mathcal{D}, (\ell_{a,t})_{a\in\mathcal{A}} \in [0,1])$.

2.1 Background on VC-dimensions

The growth function, GF, is a function of a class of classifiers (policies, Π) and a sample size n. It is a measure of how the complexity of the classifier grows with sample size.

$$GF(\Pi, n) = \max_{x_1, x_2, \ldots, x_n, \pi\in\Pi} |\{\pi(x_1), \pi(x_2), \ldots, \pi(x_n) : \pi \in \Pi\}| \tag{1}$$

It describes the maximum cardinality of the labels you can get on the n contexts as π ranges over the policy class, Π. If we are talking about binary classifiers, each element $\pi(x_i)$ takes on a 0 or 1. This implies that, for binary classifiers, we have a trivial bound of:

$$GF(\Pi, n) \leq 2^n.$$

VC-dimension describes the size of the dataset for which we can actually get all 2^n labeling on this data set using the classifiers in Π.

$$\text{VC-dim}(\Pi) = \max\{n : GF(\Pi, n) = 2^n\} \tag{2}$$

Intuitively, this means the larger the VC-dimension, the more complex the set of classifiers.

It can be shown that if $\text{VC-dim}(\Pi) = d < \infty$, then beyond d, the growth function behaves as a polynomial. That is, for any VC-class (i.e., a class of functions with finite VC-dimension) with VC-dimension $= d$:

$$GF(\Pi, n) \leq \left(\frac{en}{d}\right)^d \tag{3}$$

This result is known as the Sauer-Shelah-Vapnik-Chervonenkis lemma.

We will use this in the following way. In the VE algorithm, we draw T_0 samples uniformly at random. We know that the total number of labelings of $x_1, \ldots, x_{T_0}$ is finite. Further, we will define an equivalence class between classifiers in which two classifiers are equivalent if they agree on the observed samples. This is how we can reduce an infinite policy problem to a finite policy problem.

2.2 The VE Algorithm

Algorithm 1 VE

```
for $t = 1$ to $T_0$ do
    Receive context $x_t$
    Select $A_t$ uniformly at random
    Receive loss $\ell_{\mathcal{A}_t,t}$
end for
build a cover $\Pi' \subseteq \Pi$ such that $\forall \pi \in \Pi$, there is exactly one $\pi' \in \Pi'$ such that $\pi(x_t) = \pi'(x_t)\ \ \forall 1 \leq t \leq T_0$
for $t = T_0 + 1$ to $T$ do
    Select action $A_t$ by running Exp4 with set of "experts" = $\Pi'$ (and thus, $N = |\Pi'| \leq (\frac{eT_0}{d})^d$ )
end for
```

The cover is built using $x_1, x_2, \ldots, x_{T_0}$. The bound on the number of experts, $N = |\Pi'| \leq (\frac{eT_0}{d})^d$, follows from the Sauer-Shelah-V-C Lemma.

2.3 Regret Analysis for the VE Algorithm

Since we did Exp4 analysis using losses, we will continue to use losses, instead of rewards. Define the expected loss corresponding to policy π as:

$$L(\pi) = \mathbb{E}\left[\ell_{\pi(x)}\right] \tag{4}$$

where $(x, \ell_a) \sim \mathcal{D}$.

$$\text{Expected Regret } := \mathbb{E}\left[\sum_{t=1}^{T} \ell_{\mathcal{A}_t,t}\right] - T \cdot L(\pi^*) \tag{5}$$

$$\pi^* = \operatorname*{argmin}_{\pi \in \Pi} L(\pi)$$

Let π' be the member of Π' such that $\pi^*(x_t) = \pi'(x_t) \quad \forall 1 \leq t \leq T_0$.

To study the regret, we use the following reasoning (with explanations below; note that $K = 2$):

$$\begin{aligned}
\mathbb{E}\left[\sum_{t=1}^{T} \ell_{A_t,t}\right] &\leq T_0 + \mathbb{E}\left[\sum_{t=T_0+1}^{T} \ell_{A_t,t}\right] && (6)\\
&\leq T_0 + \mathbb{E}\left[\sum_{t=T_0+1}^{T} \ell_{\pi'(x_t),t}\right] + \sqrt{2K(T-T_0)\log N} && (7)\\
&\leq T_0 + \mathbb{E}\left[\sum_{t=T_0+1}^{T} \ell_{\pi'(x_t),t}\right] + \sqrt{2KT\log N} && (8)\\
&\leq T_0 + \mathbb{E}\left[\sum_{t=T_0+1}^{T} \ell_{\pi^*(x_t),t}\right] + \mathbb{E}\left[\sum_{t=T_0+1}^{T} \mathbb{1}(\pi'(x_t) \neq \pi^*(x_t))\right] + \sqrt{2KT\log N} && (9)
\end{aligned}$$

On the right-hand side of (6), the first term follows because we bound the losses from stages 1 to T_0 by T_0. For the remainder, we focus on the second term which describes the second phase of the algorithm that runs according to Exp4. In the analysis of Exp4, we developed a regret bound that compares the algorithm's performance to any expert, and so we focus on the particular expert used in the algorithm, π', which is in our expert set. So (7) follows from Exp4's regret bound and the fact that $\pi' \in \Pi'$ (the set of experts we were comparing to). (8) follows because $T - T_0$ is upper bounded by T.

To get to (9), we would like to bound the loss between the performance of π^* and π'. Even though we don't have π^*, we know that π' is related to π^*. π^* a fixed policy, but π' is a random policy that depends on the data in the first T_0 stages. To bound the performance difference between these two policies, we want to think about how often they disagree in the second stage. (By construction, they will agree in the first stage). Note that:

$$\ell_{\pi'(x_t),t} \leq \ell_{\pi^*(x_t),t} + \mathbb{1}(\pi'(x_t) \neq \pi^*(x_t)) \tag{10}$$

Using the fact in (10) and linearity of expectation, the right-hand side of (9) follows from (8).

2.3.1 Controlling the number of disagreements

So "all" we have to do is control the disagreement term:

$$\mathbb{E}\left[\sum_{t=T_0+1}^{T} \mathbb{1}(\pi'(x_t) \neq \pi^*(x_t))\right] \tag{11}$$

To bound this expectation, we bound the tail by analyzing the following probability:

$$\mathbb{P}\left(\sum_{t=T_0+1}^{T} \mathbb{1}(\pi'(x_t) \neq \pi^*(x_t)) > k\right) \tag{12}$$

where k is a free parameter representing the number of disagreements. The number of disagreements can range from 0 to $T - T_0$ so $k \in \{0, 1, \ldots, T - T_0\}$. In this probability, we are asking the question "Given two members of the same equivalence class (with the equivalence class being generated from the prefix of data), how much can they disagree on future data?" The probability in (12) is upper bounded by the probability that there exist two policies π_1 and π_2 that agree on the first T_0 data points but disagree more than k times on the remainder of the data:

$$(12) \leq \mathbb{P}\left(\exists \pi_1, \pi_2 \in \Pi'' : \pi_1(x_t) = \pi_2(x_t) \forall 1 \leq t \leq T_0 \text{ and } \sum_{t=T_0+1}^{T} \mathbb{1}(\pi_1(x_t) \neq \pi_2(x_t)) > k\right) \tag{13}$$

where Π'' is a cover of Π using $(x_1, x_2, \ldots, x_T)$. (Note that Π'' only shows up in the proof of the regret bound and not in the actual algorithm itself.) The "slick" part of the proof is the following: Π'' only depends on the data itself and does not depend on the order of the data. We argue that the probability in (13) is invariant under permutations (iid implied exchangeability).

$$= \mathbb{P}\left(\exists \pi_1, \pi_2 \in \Pi'' : \pi_1(x_{\sigma(t)}) = \pi_2(x_{\sigma(t)}) \forall 1 \leq t \leq T_0 \text{ and } \sum_{t=T_0+1} \mathbb{1}(\pi_1(x_{\sigma(t)}) \neq \pi_2(x_{\sigma(t)})) > k\right) \tag{14}$$

for every permutation $\sigma : \{1, \dots, T\} \to \{1, \dots, T\}$. Because all of the probabilities for the $T!$ permutations are equal, we can add up the $T!$ probabilities and divide by $T!$ to get the same probability as in (14). Now, we think of σ as a random permutation, and (14) as a probability on an augmented space where there are random variables $x_1, \dots, x_T$ and a random variable σ. Now there are two sources of randomness: the randomness in the sample and the randomness in σ. We condition on the data ($x_{1:T} := x_1, \dots, x_T$), but σ is still random:

$$\mathbb{P}\left(\exists \pi_1, \pi_2 \in \Pi'' : \pi_1(x_{\sigma(t)}) = \pi_2(x_{\sigma(t)}) \forall 1 \leq t \leq T_0 \text{ and } \sum_{t=T_0+1}^{T} \mathbb{1}(\pi_1(x_{\sigma(t)}) \neq \pi_2(x_{\sigma(t)})) > k \;\middle|\; x_{1:T}\right) \tag{15}$$

$$\leq |\Pi''|^2 \max_{\pi_1, \pi_2} \mathbb{P}\left(\pi_1(x_{\sigma(t)}) = \pi_2(x_{\sigma(t)}) \forall 1 \leq t \leq T_0 \text{ and } \sum_{t=T_0+1}^{T} \mathbb{1}(\pi_1(x_{\sigma(t)}) \neq \pi_2(x_{\sigma(t)})) > k \;\middle|\; x_{1:T}\right) \tag{16}$$

where (16) follows from the union bound.

Here, we think about what the probability in (16) means. Imagine you have a collection of T numbers (some 0's and some 1's) with at least k 1's. Here, the 0's represent the number of agreements between π_1 and π_2, and 1's represent the number of disagreements. The stream of 0's and 1's come in a random order. We bound the probability that you only see 0's in the first T_0 draws from this collection of T numbers. The probability the first number is a zero is:

$$(1 - \frac{\text{\# of 1's}}{T}) \leq (1 - \frac{k}{T}) \tag{17}$$

The probability the second number is a zero conditional on the first number being zero is:

$$(1 - \frac{\text{\# of 1's}}{T-1}) \leq (1 - \frac{k}{T-1}) \leq (1 - \frac{k}{T}) \tag{18}$$

We continue to determine the probability that the current draw is zero conditional on the previous draws all being zero, until we get to draw number T_0. The probability of the T_0th draw is a zero conditional on the first $T_0 - 1$ draws being zero is less than $(1 - \frac{k}{T})$. Multiplying these probabilities together we get:

$$\text{Probability of seeing all 0's in first } T_0 \text{ draws} \leq (1 - \frac{k}{T})^{T_0} \tag{19}$$

$$\leq e^{-\frac{kT_0}{T}} \tag{20}$$

where (20) follows by the fact that $1 - x \leq e^{-x}$. Using this bound and fact that $|\Pi''|^2 \leq (\frac{eT}{d})^{2d}$ (by Sauer-Shelah-V-C Lemma), we get

$$\mathbb{P}\left(\sum_{t=T_0+1}^{T} \mathbb{1}(\pi'(x_t) \neq \pi^*(x_t)) > k\right) \leq \left(\frac{eT}{d}\right)^{2d} e^{-\frac{kT_0}{T}}. \tag{21}$$

2.3.2 From tail bound to expectation upper bound

Suppose, we have a non-negative integer-valued random variable Z for which we have a tail bound

$$\mathbb{P}\left(Z > k\right) \leq \left(\frac{eT}{d}\right)^{2d} e^{-\frac{kT_0}{T}}.$$

Then, we have, for any k_0,

$$\begin{aligned}\mathbb{E}\left[Z\right] = \sum_{k=0}^{\infty} \mathbb{P}\left(Z > k\right) &\leq k_0 + \sum_{k=k_0+1}^{\infty} \left(\frac{eT}{d}\right)^{2d} e^{-\frac{kT_0}{T}} \\ &= k_0 + e^{-k_0 T_0/T} \left(\frac{eT}{d}\right)^{2d} \left(\frac{e^{T_0/T}}{e^{T_0/T}-1}\right) = *\end{aligned}$$

Choose $k_0 = \left\lceil \frac{T}{T_0} \log\left[\left(\frac{eT}{d}\right)^{2d} \frac{e^{T_0/T}}{e^{T_0/T}-1}\right]\right\rceil$, then

$$\begin{aligned} * &\leq \frac{T}{T_0}\log\left[\left(\frac{eT}{d}\right)^{2d} \frac{e^{T_0/T}}{e^{T_0/T}-1}\right] + 2 \\ &= \frac{T}{T_0}\log\left(\frac{eT}{d}\right)^{2d} - \frac{T}{T_0}\log(e^{T_0/T}-1) + 3 \\ &\leq \frac{2dT}{T_0}\log\left(\frac{eT}{d}\right) + \frac{T}{T_0}\log(\frac{T}{T_0}) + 3 \qquad \text{-because } \log(e^x - 1) \geq log(x)\end{aligned}$$

Therefore, using (21), we get

$$\mathbb{E}\left[\sum_{t=T_0+1}^{T} \mathbb{1}(\pi'(x_t) \neq \pi^*(x_t))\right] \leq \frac{2dT}{T_0}\log\left(\frac{eT}{d}\right) + \frac{T}{T_0}\log(\frac{T}{T_0}) + 3. \tag{22}$$

2.3.3 Putting everything together

Plugging (22) into (9) gives us the following bound on the expected regret:

$$T_0 + \frac{2dT}{T_0}\log\left(\frac{eT}{d}\right) + \frac{T}{T_0}\log(\frac{T}{T_0}) + 3 + \sqrt{4Td\log\frac{eT}{d}}$$

Now, we can set $T_0 = \sqrt{4Td\log\frac{eT}{d}}$, to make this equal to

$$5\sqrt{Td\log\frac{eT}{d}} + \frac{\sqrt{T}}{2\sqrt{d\log\frac{eT}{d}}}\log\left(\frac{\sqrt{T}}{2\sqrt{d\log\frac{eT}{d}}}\right) + 3.$$

References

[BLL$^+$11] Alina Beygelzimer, John Langford, Lihong Li, Lev Reyzin, and Robert E Schapire. Contextual bandit algorithms with supervised learning guarantees. In *AISTATS*, pages 19–26, 2011.