Network connectivity protection under the risk of adversarial model

薛冯晨517021910928

2020 年 6 月 21 日

Abstract

In the article, we talk about a network under the risk of adversarial model. The attacker and defender both select some edges to destroy or protect pair connectivity in a time slot. And we propose an algorithm for defender to protect network, which borrows the idea of EXP3 algorithm[1] for multi-arm bandit(MAB) problem. Then some analyses show that my algorithm can get an optimal result as time approaching infinity.

ketwords

network connectivity, reinforcement learning, multi-arm bandit

1 Introduction

As security is critical to network performance, it is vulnerable to a wide variety of attacks. Attacks can be from two categories: ”Passive” when a network intruder intercepts data traveling through the network, and ”Active” in which an intruder initiates commands to disrupt the network’s normal operation or to conduct reconnaissance and lateral movements to find and gain access to assets available via the network. Under the condition of ”active” attack, the network defender should confront with the intruder. So this can also be called ”adversarial”.

For example, a malicious attacker may perform a denial of service (DoS) attack by jamming in a selected area of links or creating routing worm to cause severe congestions over the network. Or more directly, the attacker may use some electromagnetic interference to obstruct the wireless network connectivity in the war. So an algorithm for defender to protect the network in the adversarial condition has strong practical meaning.

2 Modeling

The realistic problem is always manifold and difficult to solve so we only abstract the necessary information and model it as multi-arm bandit problem.

2.1 Attacker and denfender

Though the attacker is always a real person under the adversarial condition, we don't care about what exactly causes the edge failure. Actually we model it as a sequence of edges which are going to fail for each time slot. In such case, the types of attack and how it works are all neglected and we just abstract the most important information that which edges fail.

As for defender, the basic mechanism can also be ignored and what we need to know is which edges need to be protected. How to protect them specifically will not be considered. Another critical thing is the capacity of defender, that is how many edges it can protect. This is an important parameter during the modeling.

Attacker and defender are respectively the ”environment” and ”agent” in figure 2.1 about reinforcement learning. The defender do action, i.e. selecting edges to protect and then it get observations, i.e. the connectivity of network. What we need to do is finding the optimal action based on all the previous observations and rewards.

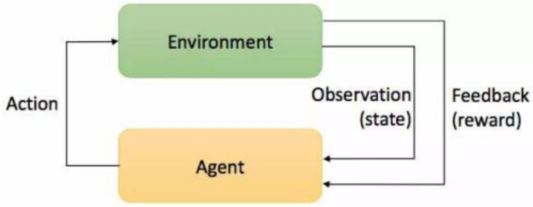

Figure 1: Reinforcement learning

2.2 Graph

In realistic situation the network may be manifold and difficult to manage. Here we abstract the structure of network into a graph $G(V, E)$ such as the one shown in figure 2.2. It has two terminal nodes s and d which respectively refer to start node and destination node. Attacker uses some method to continuously attack some edges in order to destroy the connection between s and d while defender do exactly the opposite.

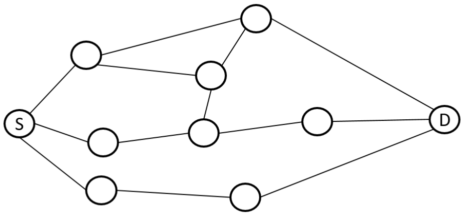

Figure 2: A network with two terminal nodes s and d

2.3 Reward

Defender is the manager of network, so it can send messages from node s and receive them in node d. Therefore whether the network is connected or not is easy to detect for defender, but to know which edges fail cost more. So we assume it can only get the connectivity information.

After selecting some edges to protect, these edges will get a reward defined by the connection. For example the most naive reward function of edge e as time t is:

$$R(e,t) = \begin{cases} 0 & \text{if connection fails,} \\ 1 & \text{if connection succeeds.} \end{cases}$$

If considering the cost $c(e)$ of protecting each edges, we can define the reward to reduce the weights of edges with higher $c(e)$ as

$$R(e,t) = \begin{cases} 0 & \text{if connection fails,} \\ \frac{\lambda_e}{c(e)} & \text{if connection succeeds.} \end{cases}$$

where λ_e is a hyper parameter for each edge.

3 Algorithm

3.1 EXP3 algoritm for MAB problem

The main concerns of MAB problem is to trade off ”exploration” against ”exploitation”. Exploration means try to select the edges never protected and exploitation strategy prefers to always choose the best known edges. In EXP3 algorithm[1], the edge to protect is selected by its probability p, which is calculated by weights w of each edge as:

$$p_i(t) = (1-\gamma)\frac{w_i(t)}{\sum_{j=1}^{|E|} w_j(t)} + \frac{\gamma}{|E|} \qquad i = 1, 2, \ldots, |E|$$

The exploration parameter γ for each path and edge controls the minimum of that probability, so that the better edges appears more likely while others will also be selected soon or later.

After defender competing with attacker and getting a reward r at time t, we should update the weight of protected edge e_j as:

$$\hat{r}_j(t) = \begin{cases} r/p_j(t) & if\ e_j\ is\ protected \\ 0 & otherwise. \end{cases}$$
$$w_j(t+1) = w_j(t)e^{\gamma_1 \hat{r}_j(t)/|s_i|}$$

Actually any way to calculate $w(t+1)$ is reasonable, but this formation can make the following analyses more easily.

3.2 Our algorithm

Hyper parameter k denotes the number of edges that defender can protect. Even though only one path from node s to d, the connectivity is protected successfully. So in the optimal algorithm, these edges prefer to be distributed along the same path. Otherwise under some extreme conditions, the algorithm performance will not be optimal. And if the attacker is more gentle, the connectivity can also be protected greatly.

According to this insights, the defender should get a cover $S = \{s_1, s_2, \ldots, s_n\}$ of $G(V, E)$ using DFS(Depth First Search) whose elements are all path from node s to d. Then in our algorithm 1, one path is firstly selected according to the modeled probability from the cover S. Then it should choose k edges to protect

Algorithm 1: Connectivity Protection Algorithm

```
Input: Graph $G(V, E)$, exploration parameter $\gamma \in (0, 1]$, $k$,
Initialize all values with 1
Get a cover $S = \{s_1, s_2, \ldots, s_n\}$ of $E$ whose element $s_i (i = 1, \ldots, |S|)$ is a path from node $s$ to $d$ using DFS
for $t = 1, 2, \ldots$ do
    Set
        $p_i(t) = (1 - \gamma_1) \frac{w_i(t)}{\sum_{j=1}^{|S|} w_j(t)} + \frac{\gamma_1}{|S|} \qquad i = 1, 2, \ldots, n$
    Select $s_k$ randomly according to the probabilities
    Set
        $p_{kj}(t) = (1 - \gamma_{kj}) \frac{w_{kj}(t)}{\sum_{j=1}^{|s_k|} w_{kj}(t)} + \frac{\gamma_{kj}}{|s_j|} \qquad i = 1, 2, \ldots, |s_j|$
    Select $E' = \{e_{i_1}, \ldots, e_{i_n}\}$ to protect randomly according to the probabilities
    Get rewards $r = 1$ or $0$
    for $j = 1, 2, \ldots, |s_i|$ do
        $\hat{r}_j(t) = \begin{cases} r/p_j(t) & if\ e_j \in E' \\ 0 & otherwise. \end{cases}$
        $w_{kj}(t+1) = w_{kj}(t) e^{\gamma_{kj} \hat{r}_j(t)/|s_i|}$
    end
    for $j = 1, 2, \ldots, |S|$ do
        $\hat{r}_j(t) = \begin{cases} r/p_j(t) & if\ j = k \\ 0 & otherwise. \end{cases}$
        $w_j(t+1) = w_j(t) e^{\gamma_1 \hat{r}_j(t)/|S|}$
    end
end
```

specifically in this path, where k is a hyper parameter denoting the number of edges defender can protect. So actually the problem is modeled as one MAB problem nesting in the other.

4 Analysis

The measure of performance for our algorithm is the regret which measures how many times the defender win by following algorithm A than choosing the best action continuously. Because there are two EXP3 process in algorithm 1, the regret is also consisting of two parts respectively referring to selecting path and edges. Given any time horizon T, the regret is defined as:

$$\begin{aligned} R_A(T) &= R_{path}(T) + \sum_{i=1}^{|S|} R^i_{edge}(T) \\ &= [G_{max}(T) - \mathbb{E}(G_A(T))] + \sum_{i=1}^{|S|} [G^i_{max}(T) - \mathbb{E}(G^i_A(T))] \end{aligned}$$

where

$$G_{max}(T) \overset{def}{=} \max_{s_i \in S} \sum_{t=1}^{T} r_i(t)$$

$$G^i_{max}(T) \overset{def}{=} \max_{e_j \in E'} \sum_{t=1}^{T} r_j(t)$$

are the returns of single globally best action.

Then talk about the performance of our algorithm. The paper [2] gives a upper bound as theorem 1.

Theorem 1. *For any* $\gamma \in (0, 1]$*,*

$$R^i_{edge}(T) = G_{max} - \mathbb{E}(G_A) \leq (e-1)\gamma G_{max} + \frac{|s_i| \ln |s_i|}{\gamma}$$

γ is a hyper parameter of our algorithm, so we can assign an appropriate value as theorem 2

Theorem 2. *For any* $t > 0$*, assume that* $g_i \geq G^i_{max}$ *and our algorithm is run with parameter*

$$\gamma = \min\{1, \sqrt{\frac{|s_i| \ln |s_i|}{(e-1) g_i}}\}$$

Then

$$R^i_{edge}(T) \leq 2\sqrt{e-1}\sqrt{g_i|s_i|\ln|s_i|} = O(\sqrt{g_i|s_i|\ln|s_i|})$$

Proof: If $g \leq \frac{|s_i|\ln|s_i|}{e-1}$, then the bound is trivial since the expected regret cannot be more than g. Otherwise, by Theorem 1, the expected regret is at most

$$(e-1)\gamma G_{max} + \frac{|s_i|\ln|s_i|}{\gamma} = 2\sqrt{e-1}\sqrt{g_i|s_i|\ln|s_i|}$$

□

So now we have an upper bound of the regret $R^i_{edge}(T)$, and similarly we can get $R_{path}(T) \leq O(g|S|\ln|S|)$. Given that $g \leq T$ for the definition, the whole regret can be represented as

$$\begin{aligned}
R_A(T) &= R_{path}(T) + \sum_{i=1}^{|S|} R^i_{edge}(T) \\
&= O(\sqrt{g|S|\ln|S|}) + \sum_{i=1}^{|S|} O(\sqrt{g_i|s_i|\ln|s_i|}) \\
&\approx O(\sqrt{g|S|\ln|S|}) + O(|S|\sqrt{\bar{g}|\bar{s}|\ln|\bar{s}|}) \\
&= O(\sqrt{g|S|\ln|S|}) + O(\sqrt{g|E|\ln|\bar{s}|}) \\
&= O(\sqrt{T|S|\ln|S|}) + O(\sqrt{T|E|\ln|\bar{s}|})
\end{aligned}$$

Ana because usually $|E| \gg |S|$, the regret $R_A(T)$ is $O(\sqrt{T|E|\ln|\bar{s}|})$.

If talking about the average regret, we can find that $R_A(T)/T = O(\sqrt{\frac{|E|\ln|\bar{s}|}{T}})$, which approaches zeros as t going to infinity. So our algorithm can give an optimal result at this condition.

5 Conclusion

In this report, we introduce the researching problem called network connectivity protection and model it under adversarial regime. Then we transfer it into an edge selecting problem and propose an algorithm based on EXP3. The analyses show that it is asymptotically optimal with regret $O(\sqrt{T|E|\ln|\bar{s}|})$ regardless of the reward assignments.

References

[1] Gergely Neu. Explore no more: Improved high-probability regret bounds for non-stochastic bandits. In *Advances in Neural Information Processing Systems*, pages 3168–3176, 2015.

[2] Sébastien Bubeck and Nicolo Cesa-Bianchi. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *arXiv preprint arXiv:1204.5721*, 2012.