A graph Patrol Problem with a random attacker and an observable patroller

Thomas Lowbridge
School of Mathematical Sciences
University of Nottingham

January 23, 2018

Contents

1 Introduction to a random attacker patroller game with observation

The model has a graph, $Q = (N, E)$, with a set of nodes labeled 1 to n, $N = \{1, ..., n\}$, and a set of edges linking these nodes. The adjacency matrix $a = (a_{i,j})_{i,j \in N}$, has $a_{i,j} = 1$ if i and j are adjacent and $a_{i,j} = 0$ if they are not adjacent. By definition we will use $a_{i,i} = 1 \quad \forall i \in N$.

An attacker has some attack time for node i, called X_i and chooses to attack node i with some probability, p_i. The attackers arrive according to some Poisson process with rate Λ, so by Poisson thinning they arrive at node i according to a Poisson process with rate $\lambda_i = \Lambda p_i$.

The patroller, uses some walk (with possible waiting) to patrol the graph.We assume that a patrollers walk is able to capture all attacks that have already begun, but not completed. But unlike the 'normal' setting the past unit time, the attackers do not start their attacks and instead will wait for the patroller to leave. Each missed attack at node i inccures a cost of c_i to the patroller.

We can formulate the state space, as the delineation of separate nodes. $\Omega = \{(\boldsymbol{s}, \boldsymbol{o}) = \quad | \quad s_i = 1, 2, ..., o_i = 0, 1, 2, ... \quad \forall i \in N\}$. Where $\boldsymbol{s} = (s_1, ..., s_n)$ has each s_i represent the number of time periods since the last visit for that node i and $\boldsymbol{o} = (o_1, ..., o_n)$ has each o_i represent the number of attackers present in the last time period when the node i was last visited (i.e The number of attackers known to be beginning their attack s_i time ago at node i).

The s_i increment by 1 if the node is not visited upon each action, or if the node is visited reset to $s_i = 1$. The o_i do not change for nodes not visited, when a node is visited, the 0_i 'reset' according to the Poisson distribution $Po(\lambda_i \times 1) = Po(\lambda)$. Due to $s_i = 1$ if and only if the patroller is currently at this node, we will use $l(\boldsymbol{s}) = \arg\min_{i \in N} s_i$ to represent the current node.

As the future of the process is independent of its past, the process can be formulated as a Markov decision process(MDP), where at the end of the period, the patroller chooses which adjacent node to visit. Thus the action space is $\mathcal{A} = \{j \,|\, a_{l(\boldsymbol{s}),j} = 1\}$, with a deterministic, stationary policy, $\pi : \Omega \to \mathcal{A}$.

The transitions of the MDP aren't entirely deterministic, $\boldsymbol{s}$ is purely deterministic, but $\boldsymbol{o}$ is partially probabilistic. In state $(\boldsymbol{s}, \boldsymbol{o})$ with the decision to visit node $i \in \mathcal{A}$, then the state will transition to $(\widetilde{\boldsymbol{s}}, \widetilde{\boldsymbol{o}})$ where $\widetilde{s}_j = s_j + 1$ if $j \neq i$ and $\widetilde{s}_j = 1$ if $j = i$ and $\widetilde{o}_j = o_j$ if $j \neq i$ and $o_j \sim Po(\lambda)$ if $j = i$.

To write down the cost function, which is dependent on the state $(\boldsymbol{s}, \boldsymbol{o})$ and the action to visit node i chosen, we will look at the expected cost of incurred at all nodes and sum these costs for the next time period.

$$C_j(\boldsymbol{s}, \boldsymbol{o}, i) = \begin{cases} c_j \lambda_j \int_0^{s_j} P(t - 1 < X_j \le t)dt + o_j P(0 < X_j \le s_j) \text{ for } i \ne j \\ c_j \lambda_j \int_0^{s_j - 1} P(t - 1 < X_j \le t)dt + o_j P(0 < X_j \le s_j) \text{ for } i = j \end{cases}$$

$$= \begin{cases} c_j \lambda_j \int_{s_j - 1}^{s_j} P(X_j \le t)dt + o_j P(X_j \le s_j) \text{ for } i \ne j \\ c_j \lambda_j \int_{s_j - 2}^{s_j - 1} P(X_j \le t)dt + o_j P(X_j \le s_j) \text{ for } i = j \end{cases} \quad (1)$$

With $C(\boldsymbol{s}, \boldsymbol{0}, i) = \sum_{j=1}^{n} C_j(\boldsymbol{s}, \boldsymbol{o}, i)$ being the cost function for the MDP.

We will now make the assumptions that X_j is bounded by B_j and that instead of using $Po(\lambda)$ for the observation transition and placing a bound on this Poisson distribution, named b_j, so we are now drawing from a truncated Poisson distribtion, henceforth called $TPo(\lambda, b_j)$. Then we can immediately say that the $o_j \le b_j$ state is finite and the state s_j has the same cost function for $s_j \ge B_j + 2$ and hence we will restrict our space to this. So our modified transition is $\widetilde{s_j} = \min(s_j + 1, B_j + 2)$ if $j \ne i$ and $\widetilde{s_j} = 1$ if $i = j$. $\widetilde{o_j} = o_j$ if $i \ne j$ and $o_j \sim TPo(\lambda, b_j)$ if $i = j$.

Further reduction is possible as if $X_j \le B_j$ then any observations o_j which started s_j time units ago is bound to have finished if $s_j \ge B_j + 1$. So our new state space is further reduced to having only $(\lfloor B_j \rfloor + 1, 0)$ when $s_j = \lfloor B_j \rfloor + 1$.

So $\Omega = \{(\boldsymbol{s}, \boldsymbol{0}) | s_i = 1, 2, .., \lfloor B_i \rfloor + 1, o_i = 1, ..., b_i \, \forall i \in N\} \cup \{(\lfloor B_j \rfloor + 2, 0)\}$.

With further modified transitions that if $s_j = \lfloor B_j \rfloor + 1$ then $\widetilde{o_j} = 0$ for $i \ne j$.

Now our state space and action space are finite we need only consider deterministic, stationary policies. Applying such a policy generates a sequence of states under a given policy π, namely $\{\psi_\pi^k(\boldsymbol{s_0}, \boldsymbol{o_0}), k = 0, 1, 2, ...\}$. However we are not guaranteed to every have a regenerating process when the same node is visited due to the unpredictable nature of $o_i \sim TPo(\lambda, b_i)$. Unless $b_i = 0 \quad \forall i \in N$ then we have removed the probabilistic nature of o_i's transition. We will not focus on the special case of $b_i = 0 \quad \forall i \in N$ but it is shown how to develop a index for the single node problem in Appendix A.

2 Single node problem

Focusing on the problem of a single nodes and stripping off the index, i, for the nodes. This problem has a visiting cost, $\omega > 0$ and we are looking to minimize the long run cost of the system.

2.1 Deterministic Attack time

Consider the case where $X = x$, where x is a constant (So $B = x$). Then we can further reduce the state space, as we choosing to visit later rather than

earlier (as long as its not too late) allows us to possibly catch more (as we know when the attacks can start to finish). So we limit the state space with non-zero observed attackers to only have $s_j = \lfloor B \rfloor + 1$, as visiting at then gets any attacks caught when visiting at any $s_j < \lfloor B \rfloor + 1$.

So in the deterministic case $\Omega = \{(\lfloor B \rfloor + 1, o) \mid o = 0, 1, ..., b\} \cup \{(\lfloor B \rfloor + 2, 0)\}$.

Suppose now we are in the state $(\lfloor B \rfloor + 1, o)$ for some $o = 1, 2, ..., b$ then our decision is either to

- Visit now

- Visit at the next time step

- Visit k time steps later, $k \geq 2$

Note. The logic of these is simply that the patroller can decide when to visit, but if they reach $(\lfloor B \rfloor + 2, 0)$ and decide to wait for one time period, they will not transition and therefore the same decide will be made again.

Now we write down the long-run average costs of following such a strategy

- Visit now:
$$\frac{c\lambda \int_0^{\lfloor B \rfloor} P(X \leq t)dt + \omega}{\lfloor B \rfloor + 1} = \frac{\omega}{\lfloor B \rfloor + 1} \tag{2}$$

- Visit at the next time step:
$$\frac{oc + c\lambda \int_0^{\lfloor B \rfloor + 1} P(X \leq t)dt + \omega}{\lfloor B \rfloor + 2} = \frac{oc + c\lambda(\lfloor B \rfloor + 1 - B) + \omega}{\lfloor B \rfloor + 2} \tag{3}$$

- Visit k time steps later, $k \geq 2$:
$$\frac{oc + c\lambda \int_0^{\lfloor B \rfloor + k} P(X \leq t)dt + \omega}{\lfloor B \rfloor + k + 1} = \frac{oc + c\lambda(\lfloor B \rfloor + k - B) + \omega}{\lfloor B \rfloor + k + 1} \tag{4}$$

Our first decision is if we should Visit now, this depends on if Equation 2 is less than equation 3 and 4.

Well for this to be true we get that $\omega < c(\lfloor B \rfloor + 1)(\lambda(\lfloor B \rfloor - B + 1) + o)$ and $\omega < c(\lfloor B \rfloor + 1)(\lambda(\frac{\lfloor B \rfloor - B}{k} + 1) + \frac{o}{k})$. Hence as the second inequality ($\forall k = 2, 3, ...$) is guaranteed by the first inequality we get.

Visit now if:
$$\omega < c(\lfloor B \rfloor + 1)(\lambda(\lfloor B \rfloor - B + 1) + o) \tag{5}$$

Similarly for visiting at the next time step, we get $\omega > c(\lfloor B \rfloor + 1)(\lambda(\lfloor B \rfloor - B + 1) + o)$ and $\omega < c(\lambda(B + 1) - \frac{o}{k-1})$ for all $k = 2, ...$ so $\omega < c(\lambda(B + 1) - o)$.

Visit next time step if:
$$c(\lfloor B \rfloor + 1)(\lambda(\lfloor B \rfloor - B + 1) + o) < \omega < c(\lambda(B + 1) - o) \tag{6}$$

Note. The question is whether it is possible to get an empty region here. Well if $c(\lfloor B \rfloor + 1)(\lambda(\lfloor B \rfloor - B + 1) + o) > c(\lambda(B+1) - o)$ is empty it will enforce the never visit decision immediately

We will never visit, if we at a point were we will choose $k+1$ over k for all $k = 2,...$

This happens when $\omega > c(\lambda(B+1) - o)$.

Never visit if:
$$\omega > c(\lambda(B+1) - o) \tag{7}$$

So to conclude, we either end up with 3 or 2 regions

- If $c(\lfloor B \rfloor + 1)(\lambda(\lfloor B \rfloor - B + 1) + o) \leq c(\lambda(B+1) - o)$
  - Visit immediately if $\omega \leq c(\lfloor B \rfloor + 1)(\lambda(\lfloor B \rfloor - B + 1) + o)$
  - Visit next time step if $c(\lfloor B \rfloor + 1)(\lambda(\lfloor B \rfloor - B + 1) + o) \leq \omega \leq c(\lambda(B+1) - o)$
  - Never Visit if $\omega \geq c(\lambda(B+1) - o)$
- If $c(\lfloor B \rfloor + 1)(\lambda(\lfloor B \rfloor - B + 1) + o) > c(\lambda(B+1) - o)$
  - Visit immediately if $\omega \leq c(\lfloor B \rfloor + 1)(\lambda(\lfloor B \rfloor - B + 1) + o)$
  - Never visit if $\omega \geq c(\lfloor B \rfloor + 1)(\lambda(\lfloor B \rfloor - B + 1)$

Note. If $o = 0$ then we never fall into the second region as $(\lfloor B \rfloor + 1)(\lfloor B \rfloor - B + 1) = (\lfloor B \rfloor + 1)(1 - R) \leq \lfloor B \rfloor + 1 \leq B + 1$ where the remainder upon flooring is defined by $R = B - \lfloor B \rfloor$ and $0 \leq R < 1$.

Also of interest is if $X = B$ has B very close to an interger, eg.$B = 3.01$ then the condition to fall into the second region is that $o < \frac{\lambda}{100}$, requiring a very high rate to fall into the category of never deciding to wait one time step.

2.2 Bernoulli Attack time

Consider the case where $X = \begin{cases} x_1 \text{ with probability } p \\ x_2 \text{ with probability } 1 - p \end{cases}$ then we apply the same logic to attempt to get a decision dependent on the visiting cost. We will assume without loss of generality that $x_2 > x_1$, then $B = x_2$. We will get some reduction of the state space as before, but it will not be as drastic, by applying the same logic there may be a gap between some states we will never choose to visit.

We will limit the state space with non-zero observed attackers to have either $s_j = \lfloor x_1 \rfloor + 1$ or $s_j = \lfloor x_2 \rfloor + 1$, due to the first one catching all the attacks caught for any $s_j < \lfloor x_1 \rfloor + 1$ and the second one catching all attacks caught for any $\lfloor x_1 \rfloor < s_j < \lfloor x_2 \rfloor + 1$.

So in the Bernoulli case $\Omega\{(\lfloor x_1 \rfloor + 1, o) \mid o = 0, 1, ..., b\} \cup \{(\lfloor x_1 \rfloor + 1, o) \mid o = 0, 1, ..., b\} \cup \{(\lfloor x_1 \rfloor + 2, o)\}$.

We will be assuming that $\lfloor x_1 \rfloor \neq \lfloor x_2 \rfloor$ as otherwise it follows that only one s_j survives.

Suppose we are in the state $(\lfloor x_1 \rfloor + 1, o)$ for some $o = 1, 2, ..., b$ then our decision is either to

- Visit now

- Wait till we are in state $(\lfloor x_2 \rfloor + 1, o)$ and then visit

- Wait till we are in state $(\lfloor x_2 \rfloor + 2, o)$ and then visit

- Wait till we are in state $(\lfloor x_2 \rfloor + 2, o)$ and wait

Now we write down the long-run average costs of following such a strategy

- Visit now:
$$\frac{c\lambda \int_0^{\lfloor x_1 \rfloor} P(X \leq t)dt + \omega}{\lfloor x_1 \rfloor + 1} = \frac{\omega}{\lfloor x_1 \rfloor + 1} \tag{8}$$

- Wait till state $(\lfloor x_2 \rfloor + 1, o)$ and then visit:
$$\frac{c\lambda \int_0^{\lfloor x_2 \rfloor} P(X \leq t)dt + coP(X \leq \lfloor x_2 \rfloor)) + \omega}{\lfloor x_2 \rfloor + 1}$$
$$= \frac{c\lambda(\lfloor x_2 \rfloor - x_1)p + cop + \omega}{\lfloor x_2 \rfloor + 1} \tag{9}$$

- Wait till state $(\lfloor x_2 \rfloor + 2, 0)$ and then visit:
$$\frac{c\lambda \int_0^{\lfloor x_2 \rfloor + 1} P(X \leq t)dt + coP(X \leq \lfloor x_2 \rfloor + 1) + \omega}{\lfloor x_2 + 2 \rfloor}$$
$$= \frac{c\lambda((x_2 - x_1)p + (\lfloor x_2 \rfloor + 1 - x_2)) + co + \omega}{\lfloor x_2 \rfloor + 2} \tag{10}$$

- In state $(\lfloor x_2 \rfloor + 2, 0)$ waiting $k \geq 1$ then visiting:
$$\frac{c\lambda \int_0^{\lfloor x_2 \rfloor + 1 + k} P(X \leq t)dt + coP(X \leq \lfloor x_2 \rfloor + 1 + k) + \omega}{\lfloor x_2 + 2 + k \rfloor}$$
$$= \frac{c\lambda((x_2 - x_1)p + (\lfloor x_2 \rfloor + 1 + k - x_2)) + co + \omega}{\lfloor x_2 \rfloor + 2 + k} \tag{11}$$

Very similar to before we will look at when certain costs are better. Starting with Visit now beating all others if $\omega < \frac{cp(\lfloor x_1 \rfloor + 1)(\lambda(\lfloor x_2 \rfloor - x_1) + o)}{\lfloor x_2 \rfloor - \lfloor x_1 \rfloor}$ and $\omega < \frac{c(\lfloor x_1 \rfloor + 1)(\lambda(x_2 - x_1)p + (\lfloor x_2 \rfloor - x_2 + 1) + o)}{\lfloor x_2 \rfloor - \lfloor x_1 \rfloor + 1}$ and $\omega < \frac{c(\lfloor x_1 \rfloor + 1)(\lambda(x_2 - x_1)p + (\lfloor x_2 \rfloor - x_2 + 1 + k) + o)}{\lfloor x_2 \rfloor - \lfloor x_1 \rfloor + 1 + k}$ for all $k \geq 1$. The first inequality guarantees the other two so we get

Visit now if:

$$\omega < \frac{cp(\lfloor x_1 \rfloor + 1)(\lambda(\lfloor x_2 \rfloor - x_1) + o)}{\lfloor x_2 \rfloor - \lfloor x_1 \rfloor} \tag{12}$$

We can similarly find the visit in state $(\lfloor x_2 \rfloor + 1, o)$ by requiring that $\omega > \frac{cp(\lfloor x_1 \rfloor + 1)(\lambda(\lfloor x_2 \rfloor - x_1) + o)}{\lfloor x_2 \rfloor - \lfloor x_1 \rfloor}$ and $\omega < c(\lambda(p((x_2 - x_1)(\lfloor x_2 \rfloor + 1) - (\lfloor x_2 \rfloor - x_1)(\lfloor x_2 \rfloor + 2)) + (\lfloor x_2 \rfloor - x_2 + 1)(\lfloor x_2 \rfloor + 1) + o((1 - p)(\lfloor x_2 \rfloor + 1) - p))$ and $\omega < \frac{c}{k+1}(\lambda(p((x_2 - x_1)(\lfloor x_2 \rfloor + 1) - (\lfloor x_2 \rfloor - x_1)(\lfloor x_2 \rfloor + 2 + k)) + (\lfloor x_2 \rfloor - x_2 + k + 1)(\lfloor x_2 \rfloor + 1) + o((1 - p)(\lfloor x_2 \rfloor + 1) - p(k + 1)))$. Note the second inequality implies the third one, so

Visit in state $(\lfloor x_2 \rfloor + 1, o)$ if:

$$\frac{cp(\lfloor x_1 \rfloor + 1)(\lambda(\lfloor x_2 \rfloor - x_1) + o)}{\lfloor x_2 \rfloor - \lfloor x_1 \rfloor} < \omega <$$
$$c(\lambda(p((x_2 - x_1)(\lfloor x_2 \rfloor + 1) - (\lfloor x_2 \rfloor - x_1)(\lfloor x_2 \rfloor + 2)) + (\lfloor x_2 \rfloor - x_2 + 1)(\lfloor x_2 \rfloor + 1) + o((1 - p)(\lfloor x_2 \rfloor + 1) - p))$$
$$= c(\lambda(1 - p)((\lfloor x_2 \rfloor - x_2)(\lfloor x_2 \rfloor + 1) + \lfloor x_2 \rfloor) + 1 + px_1 + o((1 - p)(\lfloor x_2 \rfloor + 1) - p)) \tag{13}$$

Again we could possible not have this region if the left hand side overlaps the right.

Appendices

A Observations are always zero

On a single node we are limited to the state space of $\Omega = \{(1, 0), ..., (\lfloor B \rfloor + 2, 0)\}$, then on this state space we can implement a policy which returns every k time units, this will gives an average long run cost of

$$f(k) = \frac{c\lambda \int_0^{k-1} P(X \le t)dt + \omega}{k} \tag{14}$$

So to find out when the patroller would be indifferent from choosing to return every k or every $k + 1$, solve $f(k + 1) - f(k) = 0$ giving

$$\frac{1}{k(k+1)}(c\lambda(k \int_{k-1}^{k} P(X \le t)dt - \int_0^{k-1} P(X \le t)dt) - \omega) = 0$$

Prompting an index of

$$W(k) = c\lambda(k \int_{k-1}^{k} P(X \le t)dt - \int_0^{k-1} P(X \le t)dt)$$

We note that $W(0) = 0$ and for $k \geq B+1$ $W(k) = c\lambda(k - int_0^{k+1} P(X \leq t)dt = c\lambda(1 + \int_0^{k-1} P(X > t)dt) = c\lambda(1 + E[X])$. We will now show that:

- $W(k)$ is non-decreasing

- The optimal policy when $\omega \in [W(k-1), W(k)]$ is to visit every k time units

- If $w \geq c\lambda(1 + E[X])$ then it is optimal to never visit

Proof. First

$$W(k+1) - W(k) = c\lambda((k+1)\int_k^{k+1} P(X \leq t)dt - \int_0^k P(X \leq t)dt$$

$$- (k\int_{k-1}^k P(X \leq t)dt - \int_0^{k-1} P(X \leq t)dt))$$

$$= c\lambda((k+1)\int_k^{k+1} P(X \leq t)dt - k\int_{k-1}^k P(X \leq t)dt - \int_k^{k-1} P(X \leq t)dt)$$

$$= c\lambda(k+1)(\int_k^{k+1} P(X \leq t)dt - \int_{k-1}^k P(X \leq t)dt) \geq 0$$

As $P(X \leq t)$ is non-decreasing.

Second if $\omega \in [W(k-1), W(k)]$ then we will show that $f(m)$ is non-increasing for $m \leq k$ and non-decreasing for $m \geq k$.

For $m \leq k$

$$f(m) - f(m-1) = \frac{1}{m(m-1)}(c\lambda(m-1)\int_0^{m-1} P(X \leq t)dt - c\lambda m \int_0^{m-2} P(X \leq t)dt - \omega)$$

$$= \frac{1}{m(m-1)}(W(m-1) - \omega) \leq \frac{1}{m(m-1)}(W(m-1) - W(k-1)) \leq 0$$

Similarly for $m \geq k$

$$f(m+1) - f(m) = \frac{1}{m(m+1)}(W(m) - \omega) \geq \frac{1}{m(m+1)}(W(m) - W(k)) \geq 0$$

Hence choosing to visit every k time units is optimal.

Third and finally our upper limit of $c\lambda(1+E[X])$ (Which is $W(k)$ for $k \geq B+1$) means we are indifferent from picking k and $k+1$ for $k \geq B+1$ so we will never visit. I.e $f(k+1) \leq f(k) \iff w \geq W(k)$ so not optimal if $f(k+1) \geq f(k) \forall k \iff w \geq \sup_{k=1,2,\ldots} W(k) = \lim_{k \to \infty} W(k) = c\lambda(1 + E[X])$ $\square$