Gradient-Based Optimization of Stochastic Strategies for Robotic Surveillance Problems

Connor Hughes

March 27th, 2022

Capture Probability Jacobian Computation:

See the paper "Stochastic Strategies for Robotic Surveillance as Stackelberg Games" [1] for the problem introduction and formulation. We have the following formulas for computing the capture probability matrix (denoted by F_{cap} below):

$$\begin{aligned} F_{cap} &= \sum_{k=1}^{\tau} F_k \\ F_1 &= P \\ F_{k+1} &= P(F_k - diag(F_k)), \;\; 1 \leq k \leq \tau - 1 \end{aligned}$$

Where P is the transition probability matrix associated with the Markov chain that describes the stochastic surveillance strategy, and τ is the intruder's attack duration. We also have the following constraints which ensure that the transition probability matrix is in agreement with the connectivity of the environment graph and that the transition probabilities at each node represent a valid probability distribution, respectively:

$$\begin{aligned} P\,\mathbb{1}_n &= \mathbb{1}_n \\ p_{ij} &\geq 0, \;\; \forall (i,j) \in \mathcal{E} \\ p_{ij} &= 0, \;\; \forall (i,j) \notin \mathcal{E} \end{aligned}$$

where n is the number of nodes in the environment graph. We will begin by computing the gradient of each element of the capture probability matrix F_{cap} with respect to a perturbation in the entries of P, (i.e., computing the Jacobian of the mapping f where $F_{cap} = f(P)$). We will then consider various methods for using these gradients to select a perturbation to the P matrix, apply this perturbation to get an updated P matrix, and project the result onto the set of admissible P matrices (i.e., those which do not violate the above constraints). In this way, we can implement gradient-based algorithms for optimizing P.

It is known that the mapping f is a polynomial in n^2 variables of order τ, which in general is non-convex, and thus we expect that gradient descent will likely converge to local minima. However, by applying the gradient descent algorithm repeatedly with random initial surveillance strategies, it is expected that multiple minima will be found, and this exploration process may provide insight into the "landscape of f" which may prove useful later on. Next, we will implement a stochastic gradient descent algorithm to identify effective (though perhaps suboptimal) strategies and compare their performance with known optimal strategies for the simple environment graph topologies studied in [1]. From here, we aim to extend this approach to larger graphs and parallelize the gradient-based methods to contend with the increasing computational cost. In addition, we will explore machine-learning based approaches for extracting information from the environment graph structure that will help accelerate the optimization and increase the quality of the identified strategies.

Consider a matrix of perturbations to every entry of the matrix P, denoted by $\tilde{P}$. We similarly denote the resulting perturbation to each first-hitting time probability matrix by $\tilde{F_k}$. Then, we can write:

$$\begin{aligned} F_1 &= P \\ F_1 + \tilde{F_1} &= P + \tilde{P} \\ \implies \tilde{F_1} &= \tilde{P} \end{aligned}$$

$$\begin{aligned}
F_{k+1} &= P(F_k - \operatorname{diag}(F_k)) \\
F_{k+1} + \tilde{F}_{k+1} &= (P + \tilde{P})(F_k + \tilde{F}_k - \operatorname{diag}(F_k + \tilde{F}_k)) \\
&= PF_k + P\tilde{F}_k - P\operatorname{diag}(F_k) - P\operatorname{diag}(\tilde{F}_k) + \tilde{P}F_k + \tilde{P}\tilde{F}_k - \tilde{P}\operatorname{diag}(F_k) - \tilde{P}\operatorname{diag}(\tilde{F}_k) \\
\implies \tilde{F}_{k+1} &= P\tilde{F}_k - P\operatorname{diag}(\tilde{F}_k) + \tilde{P}F_k + \tilde{P}\tilde{F}_k - \tilde{P}\operatorname{diag}(F_k) - \tilde{P}\operatorname{diag}(\tilde{F}_k)
\end{aligned}$$

Note that the above is a transformation to deviation variables. From here, considering an infinitesimal perturbation to each entry in P, we have:

$$\begin{aligned}
\partial\tilde{F}_1 &= \partial\tilde{P} \\
\partial\tilde{F}_{k+1} &= P\partial\tilde{F}_k - P\operatorname{diag}(\partial\tilde{F}_k) + \partial\tilde{P}F_k + \partial\tilde{P}\partial\tilde{F}_k - \partial\tilde{P}\operatorname{diag}(F_k) - \partial\tilde{P}\operatorname{diag}(\partial\tilde{F}_k) \\
&\approx P\partial\tilde{F}_k - P\operatorname{diag}(\partial\tilde{F}_k) + \partial\tilde{P}F_k - \partial\tilde{P}\operatorname{diag}(F_k)
\end{aligned}$$

where in the last line above we have dropped the terms which are second-order in the infinitesimal perturbations. Now, since our ultimate aim is to compute the Jacobian of the mapping f as described above, and $f : \mathbb{R}^{nxn} \to \mathbb{R}^{nxn}$, we know that the Jacobian J will be $n^2 x n^2$. One way to express this cleanly is by vectorizing both F_{cap} and P, and their corresponding perturbations. With this approach, we can write

$$\operatorname{vec}(\partial\tilde{F}_{cap}) = J\operatorname{vec}(\partial\tilde{P}) = \sum_{k=1}^{\tau} J_k \operatorname{vec}(\partial\tilde{P})$$

where J_k represents the Jacobian of the mapping f_k, where $F_k = f_k(P)$. With this notation established, we can rewrite the above recursive formulas as follows:

$$\begin{aligned}
\operatorname{vec}(\partial\tilde{F}_1) &= I_{n^2}\operatorname{vec}(\partial\tilde{P}) \\
\implies J_1 &= I_{n^2} \\
\operatorname{vec}(\partial\tilde{F}_{k+1}) &\approx (F_k^\top \otimes I_n - \operatorname{diag}(F_k) \otimes I_n)\operatorname{vec}(\partial\tilde{P}) + (I_n \otimes P - \underline{\mathrm{P}})\operatorname{vec}(\partial\tilde{F}_k)
\end{aligned}$$

where in the above expression, $\otimes$ denotes the Kronecker product, and we have used the property that $\operatorname{vec}(AB) = (I \otimes A)\operatorname{vec}(B) = (B^\top \otimes I)\operatorname{vec}(A)$. Additionally, we have defined the matrix $\underline{\mathrm{P}}$ to represent the following:

$$\underline{\mathrm{P}} = \begin{bmatrix}
P\,\mathbb{e}_1\,\mathbb{e}_1^\top & 0 & 0 & \dots & 0 \\
0 & P\,\mathbb{e}_2\,\mathbb{e}_2^\top & 0 & \dots & \vdots \\
0 & 0 & \ddots & \ddots & \vdots \\
\vdots & \vdots & \ddots & P\,\mathbb{e}_{n-1}\,\mathbb{e}_{n-1}^\top & 0 \\
0 & 0 & \dots & 0 & P\,\mathbb{e}_n\,\mathbb{e}_n^\top
\end{bmatrix}$$

In the recursive formula above, we can apply the chain rule in order to compute $J_k, \forall k \in \{1, ...\tau\}$, which in turn allows us to compute J. Once we have computed J, we can interpret each element $J(i,j)$ to be the first-order approximation of $\frac{\partial \operatorname{vec}(F)[i]}{\partial \operatorname{vec}(P)[j]}$.

Updating the Transition Probability Matrix:

We can then use the Jacobian J to implement a variety of gradient-based methods aimed at increasing the minimum capture probability. For example, we could consider the gradient of the current minimum capture probability (by extracting the appropriate row of J) and generate a perturbation P which corresponds to a step in this direction, so as to maximally increase this minimum capture probability. However, what if the minimum capture probability is not unique? We could consider steps corresponding to each of the gradients of the minimum capture probabilities, then compare which one yields the highest new minimum capture probability, which is analogous to applying Danskin's Lemma to the setting of discrete updates. Alternatively, we could average the gradients corresponding to the minimum capture probabilities and take a step in this direction. Even more generally, we could consider some number k of the lowest capture probabilities, regardless of whether they are minima, and average their corresponding gradients to determine the change to make to the P matrix.

In any such method, we will in some manner use the Jacobian J to select a "step" or perturbation $\tilde{P}$ which will be used to iteratively update the transition probability matrix P (let us denote the updated transition probability matrix by $P' = P + \tilde{P}$. However, after adding the perturbation $\tilde{P}$, we must then ensure that P' satisfies the applicable constraints. That is, we need:

(i) $P' \mathbb{1}_n = \mathbb{1}_n$

(ii) $0 \leq P'(i,j) \leq 1, \ \forall (i,j)$

(iii) $P'(i,j) = 0, \ \forall (i,j) \notin \mathcal{E}$

In order to ensure these constraints are met, we need the appropriate entries in $\tilde{P}$ equal to 0 to satisfy (iii). Since we are able to generate an initial transition probability matrix P_0 which has the appropriate entries equal to 0, doing so will ensure that P' has the same entries equal to zero. This can be accomplished by simply setting the appropriate columns of J equal to zero before computing the perturbation $\tilde{P}$. Then, after adding $\tilde{P}$ to P to get P', we can project the rows of P' onto the nearest point on the probability $n-$simplex, using the algorithm described here: https://arxiv.org/abs/1309.1541. This will ensure that constraints (i) and (ii) are satisfied.

References

[1] X. Duan, D. Paccagnan, and F. Bullo, "Stochastic strategies for robotic surveillance as stackelberg games," *IEEE Transactions on Control of Network Systems*, vol. 8, no. 2, pp. 769–780, 2021.