Cornell University

Surveillance Evasion Through Bayesian Reinforcement Learning

Dongping Qi, David Bindel, Alex Vladimirsky

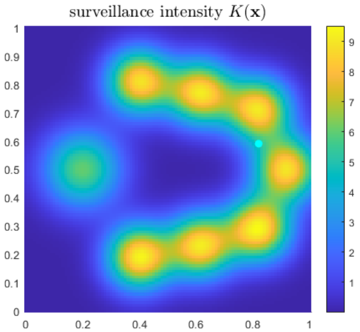
Center for Applied Mathematics
Cornell University

March 27, 2023

Outline

Cornell University

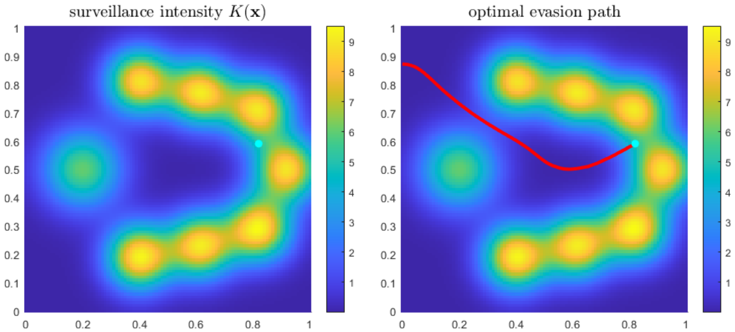
Evasion Under Known Surveillance



$$\mathbb{P}(\text{E is captured before } t) = 1 - \exp\left(-\int_0^t K(\boldsymbol{y}(s))ds\right).$$

Evasion Under Known Surveillance

Cornell University

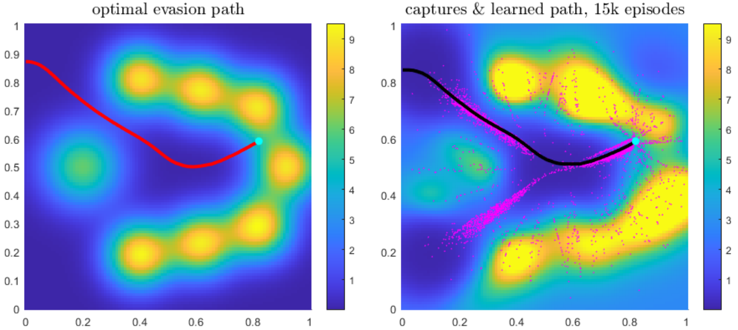


$$\mathbb{P}(\text{E is captured before } t) = 1 - \exp\left(-\int_0^t K(\boldsymbol{y}(s))ds\right).$$

Evasion Under Unknown Surveillance

Cornell University

Suppose $K(\boldsymbol{x})$ is unknown to the Evader:



- A good continuous model to reconstruct $K(\boldsymbol{x})$.
- Strategically learn $K(\boldsymbol{x})$ & find the true optimal path eventually.

Performance Metric

Cornell University

Define a capture indicator

$$\Delta_i = \begin{cases} 1, & \text{if E is captured in ith episode;} \\ 0, & \text{otherwise.} \end{cases}$$

Experimentally observed excess rate of captures (regret):

$$\mathfrak{S}_j = \frac{1}{j}\sum_{i=1}^{j}\Delta_i - W_*, \quad j = 1, \cdots, T$$

where W_* = capture probability along the optimal path.

Outline

Cornell University

Continuously Modeled Algorithms

Alg-PC: piecewise-constant model

Initialize model & parameters;
for $t = 1 : T$ **do**

$$\hat{K}(\boldsymbol{x}) = \exp\left(\tilde{Z}(\boldsymbol{x}) - \sqrt{\ln(T|\mathcal{G}|/\gamma)}\tilde{\sigma}_Z(\boldsymbol{x})\right);$$

Planning according to $\hat{K}(\boldsymbol{x})$;
Simulate $\hat{K}$-optimal path;
Update statistics $(\tilde{Z}, \tilde{\sigma}_Z)$.

- Domain decomposition $\mathcal{G}$;
- Data are used locally for estimation;
- $\tilde{Z}, \tilde{\sigma}_Z$ are piecewise-constant.
- **Ignores the correlations between K values in different cells.**

Alg-GP: GP-regression model

Initialize model & parameters;
for $t = 1 : T$ **do**

$$\hat{K}(\boldsymbol{x}) = \exp\left(M(\boldsymbol{x}) - \sqrt{\ln(T|\mathcal{G}|/\gamma)}\rho(\boldsymbol{x})\right);$$

Planning according to $\hat{K}(\boldsymbol{x})$;
Simulate $\hat{K}$-optimal path;
Update statistics $(\tilde{Z}, \tilde{\sigma}_Z)$;
Update posterior $(M(\boldsymbol{x}), \rho(\boldsymbol{x}))$;
Hyperparameter tuning every 1000 episodes.

- $\tilde{Z}, \tilde{\sigma}_Z$ values are inputs of GP at cell centers;

Kernels of GP Regression

Cornell University

- Squared exponential kernel:

$$\Sigma(\boldsymbol{x}, \boldsymbol{x}') = \alpha \exp\left(-\frac{|\boldsymbol{x} - \boldsymbol{x}'|^2}{\beta^2}\right).$$

- Matérn kernel (ν controls differentiability of GP):

$$\Sigma(\boldsymbol{x}, \boldsymbol{x}') = \alpha \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\sqrt{2\nu}|\boldsymbol{x} - \boldsymbol{x}'|/\beta\right)^{\nu} B_{\nu}\left(\sqrt{2\nu}|\boldsymbol{x} - \boldsymbol{x}'|/\beta\right).$$

(α, β) are hyperparameters which need tuning.

Exploration v.s. Exploitation

Confidence bound encouraged intensity/planning cost(Alg-PC):

$$\hat{K}(\boldsymbol{x}) = \exp\left(\underbrace{\tilde{Z}(\boldsymbol{x})}_{\text{exploitation motive}} - \underbrace{\sqrt{\ln(T|\mathcal{G}|/\gamma)}\tilde{\sigma}_Z(\boldsymbol{x})}_{\text{exploration bonus}} \right).$$

Similarly for Alg-GP:

$$\hat{K}(\boldsymbol{x}) = \exp\left(M(\boldsymbol{x}) - \sqrt{\ln(T|\mathcal{G}|/\gamma)}\rho(\boldsymbol{x}) \right).$$

The constant term $\sqrt{\ln(T|\mathcal{G}|/\gamma)}$ is inspired by a discrete algorithm Alg-D (with a proven regret bound).

Outline

Cornell University

Alg-D: a Model-based algorithm on Graph

Cornell University

A graph version of SE:

- Assume an "edge capture probability" Ψ_e.
- Shortest path problem with edge cost $-\log(1-\Psi_e)$.

Alg-D (inspired by [AOM17]):

- A confidence bound modification:

$$\hat{\Psi}_e = -\log(1-\tilde{\Psi}_e) - \sqrt{\frac{\ln(T|\mathcal{E}|/\gamma)}{\max(N_e, 1)}}$$

  and truncate $\hat{\Psi}_e$ to be positive if needed.
- **A regret bound of order $\mathcal{O}(1/\sqrt{T})$ can be proven**.
- Degrees of nodes have to grow to obtain all directions of motion in the continuous setting.

UCT: a Model-free Search Algorithm

Cornell University

An MDP version of SE

- Adding a "captured state".
- A capture induces a unit cost.

Upper Confidence Bounds on Trees[KS06]:

- Model-free, directly attempts to learn state-action value Q_e.
- Select actions according to

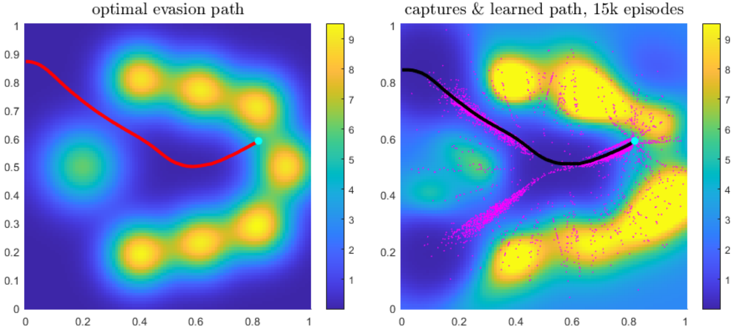
$$\hat{e} = \arg\min_{e \in \mathcal{E}(v)} \tilde{Q}_e - \lambda \sqrt{\frac{\ln(N_v)}{\max(N_e, 1)}}.$$

- Inefficient data usage, slow convergence.

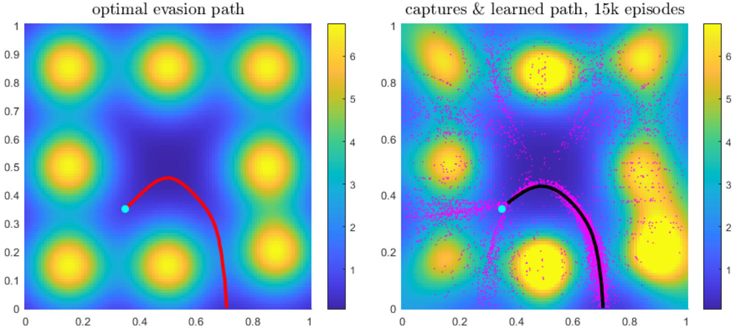
Outline

Cornell University

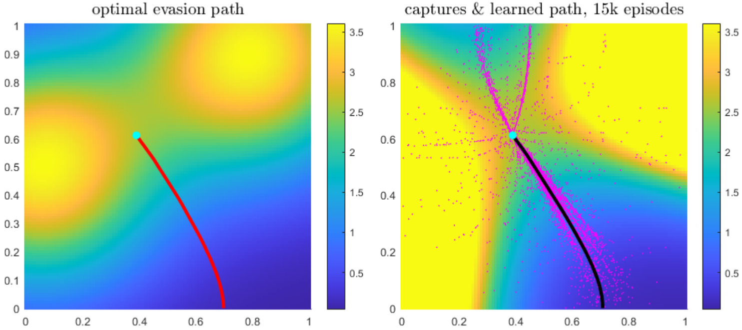
Learning Results of Alg-GP



Learning Results of Alg-GP

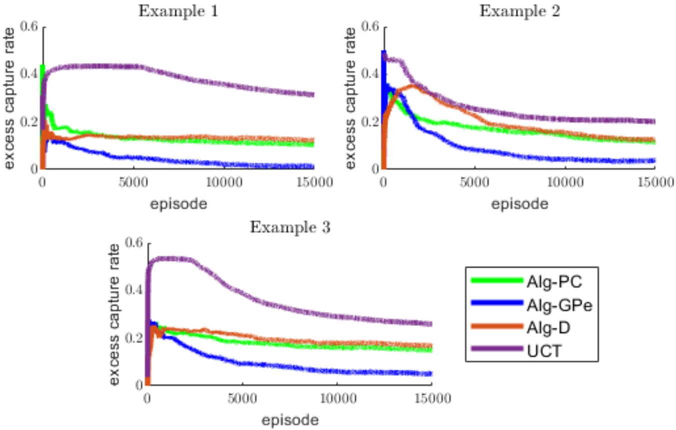


Learning Results of Alg-GP



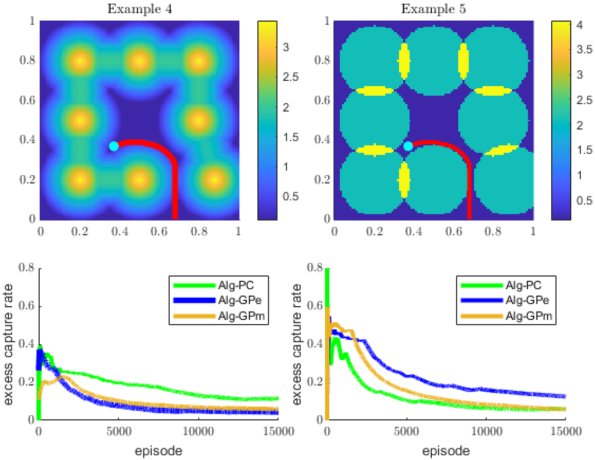
Performance Metric Results

Cornell University



Examples with Non-smooth Intensity

Choose Matérn kernel with $\nu = 5/2$:



Conclusions

- We consider a continuous path planning problem with unknown surveillance intensity.
- Our proposed algorithms apply confidence bound techniques to tackle the exploration-exploitation dilemma.
- Alg-GP takes advantage of the spatial correlations in $K(\boldsymbol{x})$ and results in faster learning.

Most important future extension:

- Regret bound for Alg-PC and Alg-GP?

References I

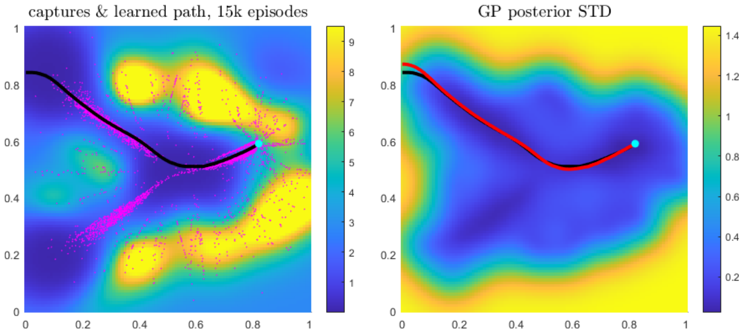
Cornell University

Mohammad Gheshlaghi Azar, Ian Osband, and Rémi Munos.
Minimax regret bounds for reinforcement learning.
In *International Conference on Machine Learning*, pages 263–272. PMLR, 2017.

Levente Kocsis and Csaba Szepesvári.
Bandit based monte-carlo planning.
In *Machine Learning: ECML 2006: 17th European Conference on Machine Learning Berlin, Germany, September 18-22, 2006 Proceedings 17*, pages 282–293. Springer, 2006.

Appendix: GP Posterior STD

Cornell University



Appendix: GP Posterior Update

Cornell University

- Denote the cells satisfying **Criteria*** as $\mathcal{G}_{\text{ob}}$. $\mathcal{G}_{\text{ob}}$'s centers are X_{ob}.
- Let $\tilde{Z}_{\text{ob}}, \tilde{\sigma}_{\text{ob}}$ be $\tilde{Z}, \tilde{\sigma}_Z$ values at X_{ob} reshaped as vectors.
- Use $\tilde{\Sigma}$ as an abbreviation of $[\Sigma_{\text{ob}} + \text{diag}(\tilde{\sigma}_{\text{ob}})]$.

GP update

Posterior mean update

$$M(\boldsymbol{x}) = m(\boldsymbol{x}) + \Sigma(\boldsymbol{x}, X_{\text{ob}})\tilde{\Sigma}^{-1}\left[\tilde{Z}_{\text{ob}} - m(X_{\text{ob}})\right].$$

Posterior covariance update

$$\rho^2(\boldsymbol{x}) = \Sigma(\boldsymbol{x}, \boldsymbol{x}) - \Sigma(\boldsymbol{x}, X_{\text{ob}})\tilde{\Sigma}^{-1}\Sigma(X_{\text{ob}}, \boldsymbol{x}).$$

Appendix: GP Hyperparameter Tuning

Cornell University

Hyperparameter Tuning

Let $\boldsymbol{z}_{\mathsf{ob,c}} = \tilde{Z}_{\mathsf{ob}} - m(X_{\mathsf{ob}})$ be the vector of centered observations.

$$\max_{\alpha,\beta>0} \; -\frac{1}{2}\boldsymbol{z}_{\mathsf{ob,c}}^{\mathsf{T}}\tilde{\Sigma}^{-1}\boldsymbol{z}_{\mathsf{ob,c}} - \frac{1}{2}\log|\tilde{\Sigma}| - \frac{n}{2}\log 2\pi,$$

Appendix: Approximation Power of GP

Cornell University

- Take a 10×10 decomposition $\mathcal{G}$.
- Find the average of $K(\boldsymbol{x})$ in each cell and treat it as the K value at the cell center.
- Use these center values as inputs of GP regression to interpolate K.