Aided Optimal Search: Data-Driven Target Pursuit from On-Demand Delayed Binary Observations

Luca Carlone, Allan Axelrod, Sertac Karaman, Girish Chowdhary

2016-08-10
Optimal Search of Moving Target
**Aided** Optimal Search of Moving Target

**DDDAS paradigm:**
“the model suggests where to sample data, the data improves the model” [F. Darema]
Outline

Problem statement
Aided Optimal Search

Target Trajectory Estimation
Sparse Gaussian Mixture Model

Target Pursuit
Mixed-Integer Convex Programming

Experiments
Outline

Problem statement
Aided Optimal Search

Target Trajectory Estimation
Sparse Gaussian Mixture Model

Target Pursuit
Mixed-Integer Convex Programming

Experiments
Aided Optimal Search in UGS field

- Target is detected by sensor when it passes within sensing radius
- Sensors record binary detections and time of detection
- Searcher interrogates sensors in its neighborhood (on-demand data)
Related work

- **optimal search**: stochastic target
  - dynamic programming [Eagle ’84], branch & bound [Eagle & Yee ’90, Lau ‘06], Bayesian approach [Bourgault et al. ’06]

- **pursuit-evasion**: adversarial target
  - **on graphs**: full visibility [Cops & robbers], local visibility [Hunter & rabbit]
  - **continuous space**: full/partial visibility in a quadrant [Lion & Man game], in polygons, differential games [homicidal chauffeur]

- **intruder isolation over graphs**:
  - delayed observations [Chen et al. ’14], Manhattan grid [Kalyanam et al. ’13]

- **Related problems**: Sensor allocation [Blasch et al. ‘10], POMDP […], UGS-based localization [Niu et al. ‘06]

**Novelty**: UGS, delayed/on-demand observations, continuous state space
Aided Optimal Search in UGS field

**Searcher**

\[ x_{t+1} = A_x x_t + B_x u_t \]

- \( x_t \) searcher position & velocity
- \( u_t \) controllable input

**Target**

\[ y_{t+1} = A_y y_t + B_y w_t \]

- \( y_t \) target position & velocity
- \( w_t \) unknown stochastic input

**Sensors (UGS)**

\( s_i \in \mathbb{R}^2 \), known positions
Aided Optimal Search in UGS field

**Searcher**

\[ x_{t+1} = A_x x_t + B_x u_t \]

- \( x_t \) searcher position & velocity
- \( u_t \) controllable input

**Target**

\[ y_{t+1} = A_y y_t + B_y w_t \]

- \( y_t \) target position & velocity
- \( w_t \) unknown stochastic input

**Sensors (UGS)**

\[ s_i \in \mathbb{R}^2, \text{ known positions} \]
Outline

Problem statement
Aided Optimal Search

Target Trajectory Estimation
Sparse Gaussian Mixture Model

Target Pursuit
Mixed-Integer Convex Programming

Experiments
Estimation of Target trajectory

• **Challenges:**
  
  • **delayed observations**
    • require estimating the entire trajectory of target (large state space)
  
  • **binary measurements**
    • make continuous-space state estimation challenging
Estimation of Target trajectory

Related works discretize the scenario:

- 250,000 states to model trajectory after 100 steps
- 2500 states to model target position

Resolution 10m
Estimation of Target trajectory

• **Challenges:**
  
  • **delayed observations**
    • require estimating the entire trajectory of target (large state space)
  
  • **binary measurements**
    • make continuous-space state estimation challenging
Exact Bayesian Smoothing

**Insight 1:** with clever choice of distributions, Bayesian smoothing can be performed in closed form

- Assume initial prior distributed as Gaussian Mixture model (GMM):
  \[ P(y_1) = \mathcal{M}_P(\{\mu_{1,j}, P_{1,j}, \alpha_{1,j}\}_{i=1}^m) \]

- **Prediction** phase of Bayes smoother produces a GMM

- **Update** phase of Bayes smoother produces a GMM if measurement likelihoods are chosen wisely

Likelihood of detection

Likelihood of no detection
Exact Bayesian Smoothing

**Insight 2:** parametrizing the GMM in information (inverse covariance form) enables fast computation

\[ \mathcal{M}_P(\{\mu_{t,j}, P_{t,j}, \alpha_{t,j}\}_{i=1}^m) \]

\[ \mathcal{M}(\{\eta_{t,j}, \Omega_{t,j}, \alpha_{t,j}\}_{i=1}^m) \]

Covariance is dense (quadratic complexity)

Information matrix is sparse (tridiagonal); linear complexity

Bayesian smoothing is inexpensive in information form:
GMM inference is relatively fast even for large number of mixture components.

Number of mixture components doubles every time we get a “no detection” (GMM reduction mitigates the problem).
Greedy search: visit maximum likelihood position
Problem statement
Aided Optimal Search

Target Trajectory Estimation
Sparse Gaussian Mixture Model

Target Pursuit
Mixed-Integer Convex Programming

Experiments
Target Pursuit

- **General formulation**: plan a finite-horizon strategy that allows to minimize the uncertainty on the position of the target:

  \[
  \min_{\mathcal{P}_{1:t+L}, \mathcal{P}_{1:t+L}, u_t, \ldots, u_{t+L}} f(\mathcal{P}_{1:t+L})
  \]

  subject to:

  - (initial searcher state)
  - (searcher dynamics)
  - (max searcher speed)
  - (max searcher acc)

  - (initial target posterior)
  - (target posterior evolution)
  - (measurements)

  minimize target uncertainty (trace of the covariance at the end of the horizon)

  subject to:

  we satisfy searcher motion constraints

  the posterior of the target trajectory is consistent with the Bayesian smoother
Target Pursuit

- General formulation is hard to solve, mainly because of the (non-convex) expressions of the Bayesian smoothing equations

**Idea 1**: approximate the posterior with weighted set of particles

\[ P(y_{1:t}|Z_{1:t}) \approx \sum_{k=1}^{K} \omega_t^{(k)} \delta(y_{1:t} - y_{1:t}^{(k)}) \]

\[ P_{1:\tau+1} = B(P_{1:\tau}, z_{\tau}^0) \quad \rightarrow \quad \begin{cases} y_{1:\tau+1}^{(k)} = Ay_{1:\tau}^{(k)} + \bar{w}_{\tau} \\ \omega_{\tau+1}^{(k)} = \omega_{\tau}^{(k)} P(z_{\tau}^0 | y_{1:\tau}^{(k)}) \end{cases} \]

Bayesian smoothing can be phrased as:
- particle trajectory prediction (can be precomputed)
- weight update (linear constraint)
Target Pursuit

• Depending on searcher motion we collect measurements or not

weight update:

\[
\begin{align*}
\omega_{\tau+1}^{(k)} &= \omega_{\tau}^{(k)} P(z_0^{(k)} \mid y_{1:\tau}^{(k)}) & \text{if measurement is acquired} \\
\omega_{\tau+1}^{(k)} &= \omega_{\tau}^{(k)} & \text{otherwise}
\end{align*}
\]

Idea 2: use binary variables to decide whether measurements are acquired or not

\[
\begin{align*}
\log(\omega_{\tau+1}^{(k)}) &= \log(\omega_{\tau}^{(k)}) + b_{i\tau} \log(\mathcal{P}(z_0^{(k)} \mid y_{1:\tau}^{(k)})) & b_{i\tau} \in \{0; 1\} \\
\|P x_\tau - s_i\| &\leq r_c + (1 - b_{i\tau}) \mathcal{M}
\end{align*}
\]

• The finite-horizon planning problem becomes a Mixed-Integer Convex Program
sample-based approximation of trajectory posterior

visiting a sensor reduces weight of samples crossing that sensor

cost minimizes trace of sample covariance

which in turns reduces the sample covariance
Outline

Problem statement
Aided Optimal Search

Target Trajectory Estimation
Sparse Gaussian Mixture Model

Target Pursuit
Mixed-Integer Convex Programming

Experiments
Aided Optimal Search

Implementation in IBM ILOG CPLEX Optimization Studio

- planning horizon: $L = 40$
- #Sensors: 100
- #Samples = 500
- planned trajectory in yellow

- finite-horizon planning:
  - plan for $L$ look-ahead steps
  - then execute the plan
  - re-plan when needed
**scenario**: high target speed
(proposed approach)
scenario: high target agility
(proposed approach)
**scenario:** multimodal initial target distribution (proposed approach)

50 Monte Carlo runs - simulation ends when: pursuer finds target (**Loc**), target escapes square region (**Esc**), or max time elapses (**Max**)

![Typical planning time (in seconds)](image)

(c) initial target uncertainty
Conclusion

• **Aided Optimal Search**: search of stochastic target in UGS field
  • on-demand, delayed observations
  • continuous state space

• **Target trajectory estimation**: exact Bayesian smoothing in information form
  ✓ fast, exact
  ⚠ nr. of components grows over time (model reduction mitigates the problem)

• **Target pursuit**: mixed-integer convex programming (MIP)
  ✓ grounded, produces meaningful plans
  ⚠ MIP is NP-hard (but timing is acceptable for relatively large problem size/horizon)
Acknowledgements
Partial support from:
Air Force Office of Scientific Research
DDDAS Program
(Dr. Federica Darema)

Thank you!