Inverse filtering and other problems on Markov decision processes

Cristian R. Rojas

Division of Decision and Control Systems
KTH Royal Institute of Technology
Stockholm, Sweden

Joint work with Robert Mattila, Inês Lourenço, Bo Wahlberg and Vikram Krishnamurthy

French System Identification Group
May 27, 2021

Outline

Introduction

Background

Inverse filtering

Belief estimation

Conclusions

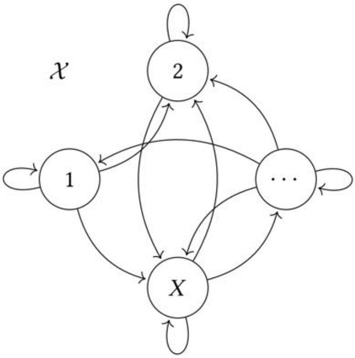
Introduction

- Nowadays, model-free techniques such as reinforcement learning aim to learn a controller/policy directly from data of a process to be controlled.
- These techniques may require an unreasonably large number of interactions with the process to determine a reasonably performing controller. This is because the data has to supply the lack of prior knowledge on the process (usually encoded in a model).
- In this talk, we develop preliminary tools for learning a model of a process from an alternative source: data from an existing *controller* or *filter* acting on it.

  These tools will be described within the context of "counter-adversarial systems".

Markov chains

A simple model of a dynamic sytem



- Time: k
- State: x_k
- Discrete state-space:

$$\mathscr{X} = \{1, \ldots, X\}$$

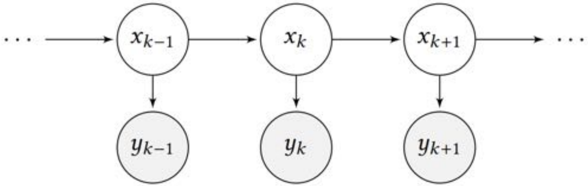
- Transition matrix:

$$[P]_{ij} = \mathrm{P}[x_{k+1} = j \mid x_k = i]$$

Note: Depends only on current state

Hidden Markov models (HMMs)

- A Markov chain observed via an uncertain sensor

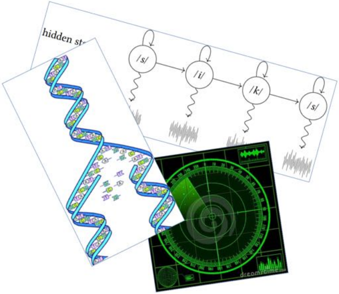


- Observation: y_k
- Discrete observation space: $\mathscr{Y} = \{1, \ldots, Y\}$
- Observation matrix: $[B]_{ij} = \mathrm{P}[y_k = j \mid x_k = i]$

Hidden Markov models (HMMs) (cont.)

Applications:

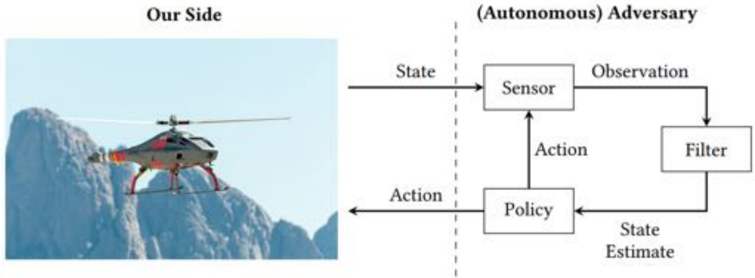
Social networks, speech recognition, target tracking, intent modeling, acoustics, computational biology, climatology, finance and econometrics, handwriting and text recognition, image processing, computer vision, time-series analysis, medicine, etc.



Generalizations:

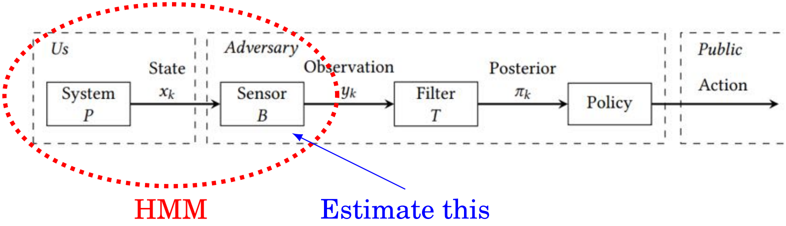
- Control: *(partially observed) Markov decision processes*
- General state/observation spaces: *Linear state-space model, ...*
- ...

Counter-adversarial autonomous systems



Counter-adversarial autonomous systems (cont.)

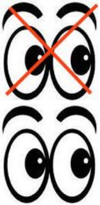
Abstraction:

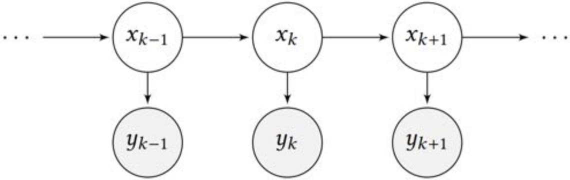


Goal of first part of the talk:
How to estimate the components of an adversary based on different information sets (*e.g.*, x_k, π_k, or action)

(Inverse) filtering

Usually interested in the state of an HMM, which is hidden:



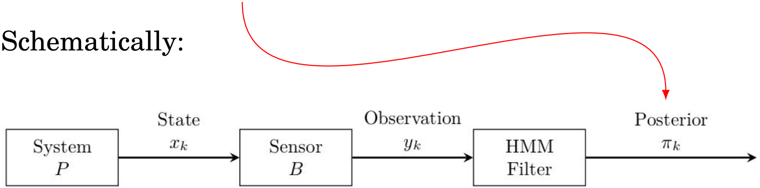


(Inverse) filtering (cont.)

Given observations $y_1,\ldots,y_k$, an **HMM filter** computes the probability of the system being in each state at time k:

$$[\pi_k]_i = \mathrm{P}[x_k = i \mid y_1,\ldots,y_k]$$

Schematically:



(Inverse) filtering (cont.)

Given observations $y_1, \ldots, y_k$, an **HMM filter** computes the probability of the system being in each state at time k:

$$[\pi_k]_i = \mathrm{P}[x_k = i \mid y_1, \ldots, y_k]$$

Formally,

$$\pi_k = \frac{\mathrm{diag}(b_{y_k}) P^T \pi_{k-1}}{b_{y_k}^T P^T \pi_{k-1}} \qquad (b_{y_k} := B_{:,y_k})$$

Inverse filtering (cont.)

Question:

Given $\pi_1, \ldots, \pi_k$, what can be said about

- the parameters P and B?
- the observations $y_1, \ldots, y_k$?

Inverse filtering: Naïve solution

Assume P is known
Rewrite the HMM filter

$$\pi_k = \frac{\text{diag}(b_{y_k}) P^T \pi_{k-1}}{b_{y_k}^T P^T \pi_{k-1}}$$

as

$$(b_{y_k}^T P^T \pi_{k-1})\pi_k = \text{diag}(b_{y_k}) P^T \pi_{k-1}$$

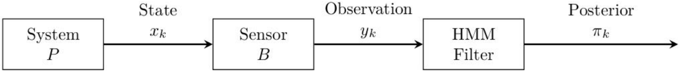
This equation holds for every update of π_k

Idea:

Can we find parameters consistent with data from an optimization problem?

Inverse filtering: Naïve solution (cont.)

Assume P is known and the HMM filter matches P, B:



Naïve solution: Optimization (feasibility) problem:

$$\begin{array}{ll}\min\limits_{\{y_k\}_{k=1}^N,\{b_i\}_{i=1}^Y} & \sum\limits_{k=1}^{N} \left\| (b_{y_k}^T P^T \pi_{k-1})\pi_k - \operatorname{diag}(b_{y_k}) P^T \pi_{k-1} \right\|_\infty \\ \text{s.t.} & y_k \in \{1,\dots,Y\}, \qquad k = 1,\dots,N \\ & b_i \geqslant 0, \qquad i = 1,\dots,Y \\ & [b_1 \cdots b_Y]\mathbb{1} = \mathbb{1}\end{array}$$

Can be written as a mixed-integer linear program (MILP)

Question: Can we exploit structure to solve it efficiently?

Inverse filtering: Efficient solution

Lemma

The HMM filter update equation

$$\pi_k = \frac{B_{y_k} P^T \pi_{k-1}}{\mathbb{1}^T B_{y_k} P^T \pi_{k-1}}$$

What we want

can be equivalently written as

$$\left(\pi_k [P^T \pi_{k-1}]^T - \text{diag}[P^T \pi_{k-1}]\right) b_{y_k} = 0$$

Lemma

If P and B are positive matrices, then the nullspace of

$$\pi_k [P^T \pi_{k-1}]^T - \text{diag}[P^T \pi_{k-1}]$$

has dimension 1.

Inverse filtering: Efficient solution (cont.)

Algorithm:

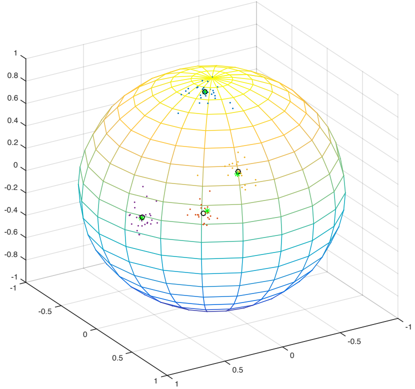
1. For each k, compute a basis (vector) for the nullspace of

$$\pi_k[P^T \pi_{k-1}]^T - \mathrm{diag}[P^T \pi_{k-1}] \qquad (*)$$

2. Collect the different basis vectors into the columns of matrix B, and normalize it so its rows sum to 1
3. For each k, check which column of B is contained in the nullspace of $(*)$ $\Rightarrow$ this yields y_k (up to relabeling)

Inverse filtering: Efficient solution (cont.)

Noisy case:
If the π_k's are contaminated by noise, estimating B yields a **clustering problem** (*e.g.*, spherical K-means)



Every nullspace is a noisy estimate of one column of B.

Inverse filtering: Example

Sleep tracking

- 5 sleep stages: Wake, S1, S2, SWS, REM
- Wearables (Fitbit, Apple Watch, ...) employ *automatic sleep stagers*
- An HMM:
  - *unobserved*: sleep stage
  - *observed*: heart rate, movement, ...

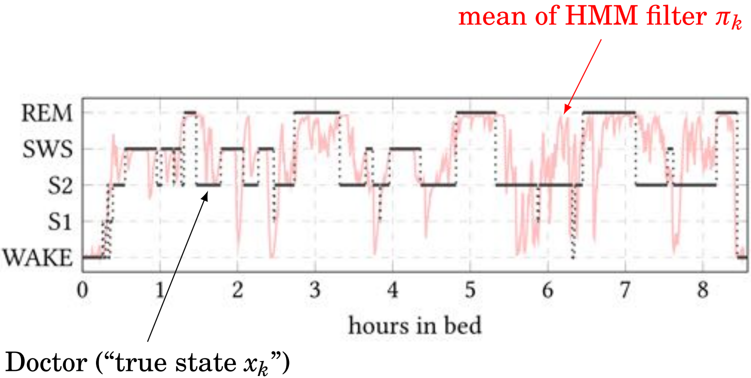
Inverse filtering:

- Can a competitor's sensor system be *reverse engineered*?
- Medical equipment → *fault detection* / *cyber-security*?

Result: We can reconstruct measurements and sensor!

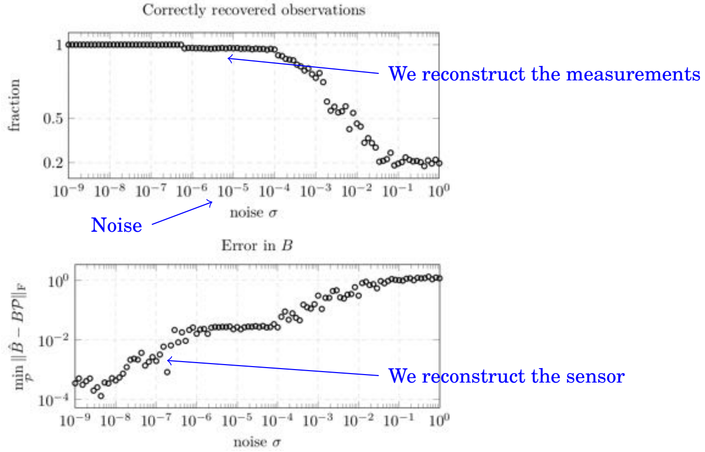
Inverse filtering: Example (cont.)

Sleep stages:

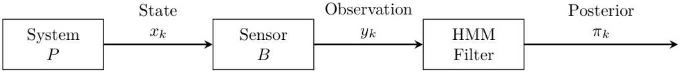


Inverse filtering: Example (cont.)

Results:



Inverse filtering: Extensions



- Extended to *linear (Gaussian) dynamical systems*
- So far, we have solved the inverse filtering problem for HMMs **assuming that P is known**

- If only the posteriors $\pi_1, \ldots, \pi_k$'s are known (*but not P!*), we can still solve the problem!

  Rough idea: HMM filter updates can be written as

$$(\pi_{k-1}^T \otimes [\pi_k \mathbb{1}^T - I])\, \mathrm{vec}(\mathrm{diag}(b_{y_k})P^T) = 0$$

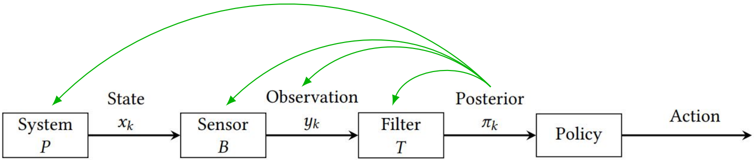
  $\mathrm{vec}(\mathrm{diag}(b_{y_k})P^T)$ can be estimated by "clustering" the nullspaces of matrices $\pi_{k-1}^T \otimes [\pi_k \mathbb{1}^T - I]$, using convex optimization!

Inverse filtering: Extensions (cont.)

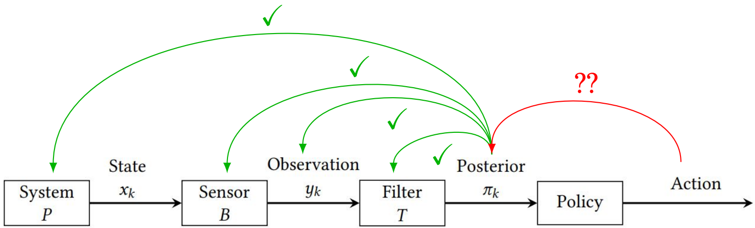
Note: Inverse filtering does **not** require HMM filter to be based on the *true* P, B matrices of the system and sensor, *i.e.*, there can be *model mismatch*!

I.e., given posteriors $\pi_1, \ldots, \pi_N$, one can determine:

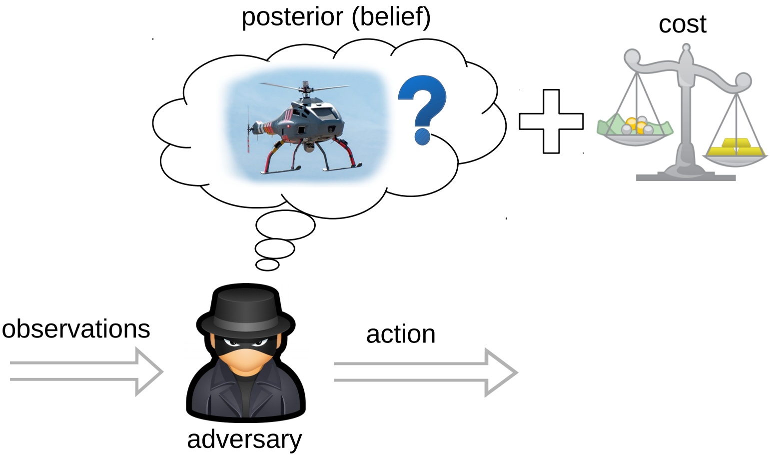
- P_{filter}, B_{filter} matrices of the HMM filter, and measurements $y_1, \ldots, y_N$
- true system and sensor matrices P_{true},B_{true}, using EM (Baum-Welch) algorithm, or spectral learning



Next subproblem



Belief estimation in counter-adversarial setting



Belief estimation in portfolio selection

Model

1. Adversary makes observation y_k
2. Adversary computes posterior

$$[\pi_k]_i = \mathrm{P}[x_k = i \mid y_1, \ldots, y_k]$$

   using the HMM filter
3. Adversary selects an action by minimizing its expected cost:

$$\begin{aligned} \min_{u_k} \quad & \mathrm{E}\{c(x_k, u_k) \mid y_1, \ldots, y_k\} = \sum_{i=1}^{X} [\pi_k]_i c(i, u_k) \\ \text{s.t.} \quad & u_k \in \mathscr{C} \end{aligned}$$

4. We observe the chosen action u_k^*

Belief estimation

Question: Given u_k^*, how can the posterior π_k be estimated?

Idea:

- Use inverse optimization:
  - Write down optimality (KKT) conditions
  - Find which value of π_k makes u_k^* optimal

Belief estimation: Solution

Theorem

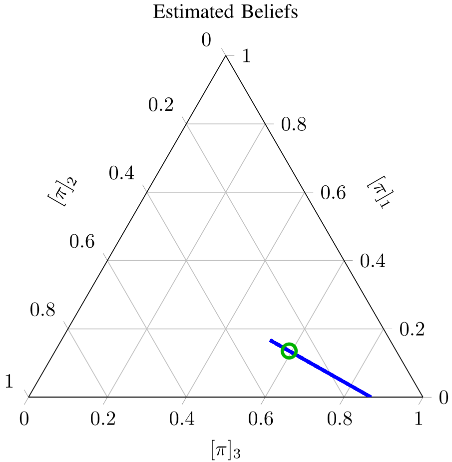
Assume that for each fixed x, c(x,u) is convex and differentiable in u, and that the constraint set $\mathscr{C}$ is affine:

$$\mathscr{C} = \{u \in \mathbb{R}^U : Au = b,\ u \geqslant 0\}, \qquad A \in \mathbb{R}^{N\times U},\ b \in \mathbb{R}^N.$$

Then, the exact set of private beliefs $\pi_k \in \mathbb{R}^X$ of the agent who made decision u_k^* at time k is

$$\Pi_k = \left\{ \pi \in \mathbb{R}^X : \begin{array}{l} \textit{there exist } \lambda \in \mathbb{R}^U,\ \nu \in \mathbb{R}^N \textit{ such that} \\ \pi^T \mathbb{1} = 1,\ \pi \geqslant 0,\ \lambda \geqslant 0, \\ [\lambda]_j = 0 \textit{ if } [u_k^*]_i \neq 0 \textit{ for } j = 1,\dots,U, \\ \sum_{i=1}^X [\pi]_i \nabla_u c(i, u_k^*) - \lambda + A^T \nu = 0 \end{array} \right\}$$

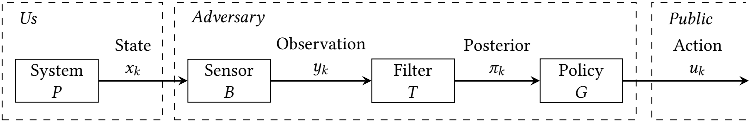
Belief estimation: Example



∘ True private belief π_k

– Set of consistent beliefs Π_k

Belief estimation: Bayesian approach



If the action u_k and the state x_k are known, as well as P, B, T and G, one can estimate the belief π_k using a Bayesian approach (*i.e.*, as a distribution on the simplex)

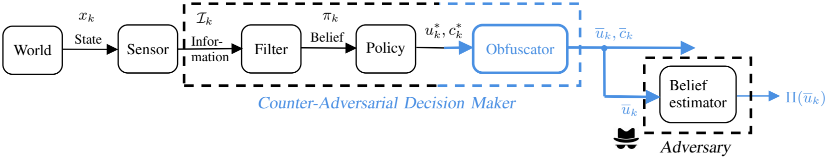
Idea: Estimate π_k using a *particle filter* / *smoother*!
(this can handle more general cases, *e.g.*, discrete actions, randomized policies, *etc.*)

More details in:

R. Mattila, I. Lourenço, C.R.R., V. Krishnamurthy, and B. Wahlberg. "Estimating private beliefs of Bayesian agents based on observed decisions". *IEEE L-CSS*, 3(3):523-528, 2019.

Belief estimation: Privacy protection

Question: How can we protect ourselves against an adversary is attempting to reconstruct own belief?



Using an obfuscator!

Since the set Π_k of beliefs of the adversary can be computed, we can *perturb* the optimal action u_k^* so that $\pi_k \notin \Pi_k$

More details in:

I. Lourenço, R. Mattila, C.R.R., and B. Wahlberg. "How to protect your privacy? A framework for counter-adversarial decision making". *CDC*, 2020.

Conclusions

- Introduced several inverse problems on HMMs and MDPs, including:
  - Inverse filtering for HMMs
  - Belief estimation
- These problems are very relevant in machine learning, as their solution allows to extract prior knowledge from agents for use in reinforcement learning and control
- Next steps:
  - Full problem: from actions + measurements to model!
    (Identifiability issues, quantization of belief space, . . .)
  - Applications to healthcare (reverse-engineering medical practitioners)

References

[1] R. Mattila, C.R.R., V. Krishnamurthy, and B. Wahlberg. “Inverse Filtering for Hidden Markov Models”. *NIPS*, 2017.

[2] R. Mattila, C.R.R., V. Krishnamurthy, and B. Wahlberg. “Inverse filtering for linear gaussian state-space models”. *CDC*, 2018.

[3] R. Mattila, I. Lourenço, C.R.R., V. Krishnamurthy, and B. Wahlberg. “Estimating private beliefs of Bayesian agents based on observed decisions”. *IEEE L-CSS*, 3(3):523-528, 2019.

[4] R. Mattila, I. Lourenço, V. Krishnamurthy, C.R.R., and B. Wahlberg. “What did your adversary believe? Optimal smoothing in counter-autonomous systems”. *ICASSP*, 2020.

[5] R. Mattila, C.R.R., V. Krishnamurthy, and B. Wahlberg. “Inverse filtering for hidden Markov models with applications to counter-adversarial autonomous systems”. *IEEE TSP*, 68:4987-5002, 2020.

[6] I. Lourenço, R. Mattila, C.R.R., and B. Wahlberg. “How to protect your privacy? A framework for counter-adversarial decision making”. *CDC*, 2020.

Thank you for your attention.

Questions?