J Syst Sci Complex (2021) 34: 1–21

HIDDEN MARKOV MODELS: INVERSE FILTERING, BELIEF ESTIMATION AND PRIVACY PROTECTION*

LOURENÇO Inês · MATTILA Robert · ROJAS Cristian R. · HU Xiaoming · WAHLBERG Bo

DOI: 10.1007/s11424-021-1247-1
Received: 6 July 2021

Abstract A *hidden Markov model* (HMM) comprises a state with Markovian dynamics that can only be observed via noisy sensors. This paper considers three problems connected to HMMs, namely, inverse filtering, belief estimation from actions, and privacy enforcement in such a context. First, we discuss how HMM parameters and sensor measurements can be reconstructed from posterior distributions of an HMM filter. Next, we consider a rational decision-maker that forms a private belief (posterior distribution) on the state of the world by filtering private information. We show how to estimate such posterior distributions from observed optimal actions taken by the agent. In the setting of adversarial systems, we finally show how the decision-maker can protect its private belief by confusing the adversary using slightly sub-optimal actions. Applications range from financial portfolio investments to life science decision systems.

Keywords Hidden markov models, counter-adversarial systems, inverse filtering, belief estimation, inverse decision making.

1 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 satisfying performing controller. This is because the data has to supply the lack of prior knowledge on the process

LOURENÇO Inês · MATTILA Robert · ROJAS Cristian R. · WAHLBERG Bo
Division of Decision and Control Systems, School of Electrical Engineering and Computer Science, KTH Royal Institute of Technology, Stockholm, Sweden. Email: {ineslo, rmattila, crro, bo}@kth.se
HU Xiaming
Division of Optimizations and Systems Theory, School of Engineering Sciences, KTH Royal Institute of Technology, Stockholm, Sweden. Email: hu@kth.se
*This work was supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP), the Swedish Research Council and the Swedish Research Council Research Environment NewLEADS under contract 2016-06079.
◇ *This paper was recommended for publication by Editor XIE Liang-Liang.*

usually encoded in a model. Such models can in many control application be identified from optimally designed input and output data, [1, 2]. However, for more integrated intelligent systems, the behaviour of the systems can only be indirectly observed from alternative sources such as state observers, beliefs or control decisions.

Since the pioneering work of Kalman [3], there have been numerous applications of inverse optimal control. More recently [4, 5], the inverse problem for discrete-time Linear Quadratic (LQ) Regulators over finite-time horizon is considered, in which the identifiability of the corresponding model structure is fully studied. Specifically, the unknown parameters in the quadratic objective function are reconstructed using the given discrete-time linear system dynamics and its noisy measurement of output. In [6] continuous-time inverse quadratic optimal control problem over finite-time interval is studied, in which the first complete result on the necessary and sufficient condition for the existence of corresponding standard linear quadratic cost functions is obtained. The problem of inverse reinforcement learning for Markov decision processes concerns how to reveal a reward function from observed optimal behaviour, see [7]. An important application is to find the reward function of an expert for apprenticeship learning [8].

The corresponding problem of inverse filtering concerns how to recover the stochastic dynamics from observations of the optimal state estimator including its uncertainty. Inverse filtering for hidden Markov models was first studied in [9]. There is a family of inverse problems depending on what is unknown. A complete solution to this problem is presented in [10]. The inverse Kalman filtering problem for linear Gaussian systems has been studied in [11], and is related to the inverse LQ control problem.

The work to be presented in this paper builds on results for inverse filtering. We present tools for learning a model of a process from an alternative source: data from a filter acting on it or observed optimal actions. These algorithms will be described within the context of "counter-adversarial systems". Figure 1 shows the structure of the problems to be studied.

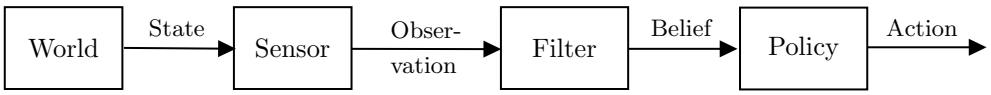

Figure 1: Problem description.

A sensor measures the current state of the world/system, and generates observations/data. This data is filtered to compute the posterior distribution of the state. This forms the belief of an agent, who then decides on an action according to an optimal policy. The objective of this paper is to study the corresponding inverse problems for systems described using hidden Markov models, from the corresponding estimation and optimization problems.

The main contribution of this paper is to describe solutions to the following three questions.

- *The first question is how to estimate the world and sensor model of an agent based on observed posteriors/beliefs?*
- *The second question is how can the agent's belief be estimated from observed actions*?

- *The third question is how to perform slightly sub-optimal decisions in order to make it difficult for a possible adversarial agent to estimate the private belief?*

The first problem has been previously studied in the context of inverse filtering [9, 10]. The presented solutions to the second and third problems build on the conference papers [12–14].

2 Background

In this section, we present the notation used throughout the paper and introduce the Markov models considered, namely Markov chains [15] and hidden Markov models [16], which are widely known and have been extensively studied in the literature for more than fifty years.

2.1 Notation

All vectors are column vectors unless transposed, $\cdot^T$. The vector of ones is represented as $\mathbb{1}$. The element at position i of a vector is denoted $[\cdot]_i$ and at row i and column j of a matrix as $[\cdot]_{ij}$. The operator $\text{diag}(\cdot) : \mathbb{R}^n \to \mathbb{R}^{n\times n}$ applied to a vector results in the placement of the vector on the diagonal of a matrix with all other elements zero. The indicator function $\mathrm{I}\{\cdot\}$ equals one if $\cdot$ is true and zero otherwise. We employ $z_{0:k}$ as shorthand for the sequence $z_0, \dots, z_k$, and define $z_{k+1:k} = \varnothing$. Inequalities $(\leq, \geq)$ between vectors are evaluated element-wise. The probability of event $\cdot$ is given by $\Pr[\cdot]$ and the probability density (or mass) function by $p(\cdot)$. The expected value of a random variable X is denoted as $\mathbb{E}\{X\}$. The set of positive real numbers is denoted as $\mathbb{R}^+$, and of positive real numbers as $\mathbb{R}_+$. The symbol $\sim$ means "distributed according to".

2.2 Markov Chains, Beliefs in HMMs, and Policies

In this section we explain in bold each of the blocks of the diagram in Figure 1.

A discrete-time **stochastic process** is a sequence of random variables $\{x_k,\ k = 0, 1, 2, \dots\}$. Each element $\{1, \dots, X\}$ of the state-space $\mathcal{X}$ of the stochastic process is called a **state** $x_k \in \mathcal{X}$ of the process where k represents its evolution over time. The *Markov property* asserts that the state x_k of the process depends only on the previous state, x_{k-1}. A stochastic process that satisfies the Markov property is called a *Markov chain*. The probabilities of transitioning from one state to another can be summarized in the $X \times X$ transition matrix, P:

$$[P]_{ij} = \Pr[\, x_{k+1} = j | x_k = i \,], \tag{1}$$

with $i, j \in \mathcal{X}$ and where $0 \leq [P_k]_{ij} \leq 1$ and $\sum_{j=1}^{X} [P]_{ij} = 1$, or, equivalently, $P\mathbb{1} = \mathbb{1}$, and where the dynamics are assumed to be time-invariant.

Markov chains assume that the state of the system (interpreted as the state of the world, or the agent's external environment later) is fully observable. However, in most settings this is not the case. *Hidden Markov models* (HMMs) are finite-state Markov Chains measured through a **sensor** via a noisy observation process, which means that the state of the system evolves stochastically with time, but is only partially rather than fully observable. The unknown states are denoted *hidden* states, $x_k \in \mathcal{X}$, and the **observations** $y_k \in \mathcal{Y}$. HMMs are therefore parametrized by transition probabilities P just as Markov chains, but also by *observation likelihoods* B, that determine the probability (or probability density, if $\mathcal{Y}$ is a continuum) of an

observation y_k being obtained in state x_k. The Markov property still applies to the hidden states, as in (1). In general, the observation likelihoods are conditional probability densities, but for a finite-dimensional observation process $Y = \{1, \ldots, Y\}$, they can be summarized in an $X \times Y$ observation matrix with elements:

$$[B]_{xy} = \Pr[\, y_k = y \mid x_k = x \,], \tag{2}$$

Here, $x \in \mathcal{X}$ and $y \in \mathcal{Y}$. Moreover, $B \geq 0$ and $\sum_{y=1}^{Y} [B]_{xy} = 1$ or, equivalently, $B\mathbb{1} = \mathbb{1}$.

Given an HMM and a set of observations $\mathcal{O}_k = \{y_1, \ldots, y_k\}$, the problem of estimating the hidden state of the system at a certain time given observations up (and including) that point in time is called **filtering**. In Bayesian state estimation, optimal filtering techniques can be used by the agent to solve the filtering problem. Examples are Kalman filters, if the observations are generated by a linear and Gaussian process [17], and HMM filters, if the state-space is discrete [16]. The filtering process can, however, depend on more subjective and abstract information such as the beliefs and opinions of an investor listening to rumors or gossip.

The HMM filter is a recursive algorithm for computing the posterior distribution $\pi_k \in \mathbb{R}^X$, also referred to as the **belief**:

$$[\pi_k]_i = \Pr[\, x_k = i \mid y_{1:k} \,], \tag{3}$$

where $\pi \geq 0$ and $\mathbb{1}^T \pi = 1$. Given knowledge of the model parameters P and B, the iterative update of the belief is given by the function $T(\pi_k - 1, y_k)$ as

$$\pi_k = T(\pi_{k-1}, y_k; P, B) = \frac{B_{y_k} P^T \pi_{k-1}}{\mathbb{1}^T B_{y_k} P^T \pi_{k-1}}, \tag{4}$$

where $B_y = \mathrm{diag}(Be_y) \in \mathbb{R}^{X \times X}$ is a diagonal matrix of the yth column containing the observation matrix B. More details can be found in [16].

Based on the beliefs or preferences, decision-making is the process of performing **decisions** by choosing one amongst several alternatives. Therefore, the diagram from Figure 1 is extended with a component designated as **policy**, G, where $u_k \in \mathcal{U}$ is the action performed based on the current belief π_k and $\mathcal{U}$ is the decision set. Assuming $\mathcal{U}$ to be a finite set of actions, G is in full generality a probabilistic policy: given a belief π, it assigns a probability to each action $\Pr[u_k = u | \pi_k = \pi]$. In many applications, the policy reduces to a deterministic policy (where the distribution G is a degenerate probability mass function).

To summarize, the dynamics of filtering and decision-making in HMMs that are schematically represented in Figure 1 are mathematically given by:

$$\begin{aligned} \text{world:} \quad & x_k \sim P, \;\; x_0 \sim \pi_0 \\ \text{sensor:} \quad & y_k \sim B, \\ \text{filter:} \quad & \pi_k = T(\pi_{k-1}, y_k; P, B), \\ \text{policy:} \quad & u_k \sim G. \end{aligned} \tag{5}$$

3 Inverse Filtering

In Section 2.2 we have stated the *filtering* problem as: *Given observations about a system, what can be said about its state?* This problem can be solved in a recursive manner using an HMM filter, as described by (4).

The *inverse filtering* problem, on the other hand, consists of estimating information about the system from the belief vector of a filter applied to the system. This can be formulated as:

Problem 3.1 (General Inverse Filtering) Given $\pi_1, \ldots, \pi_N$, what can be concluded regarding:

- the parameters P_{filter} and B_{filter} of the HMM filter*?
- the observations $y_1, \ldots, y_N$?
- the true transition and sensor matrices P and B of the HMM?

This problem is schematically represented in Figure 2.

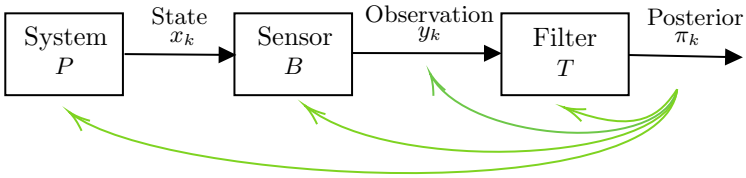

Figure 2: A sensor measures the current world state, x_k, and obtains information y_k. By filtering this information, using (4), an agent obtains a private belief π_k. The inverse filtering problem consists of estimating characteristics of the HMM based on knowledge about the posterior distribution.

Before presenting our main result on inverse filtering, the following remarks are in order:

- If only the posteriors $\pi_1, \ldots, \pi_N$ are available, but no prior knowledge on the structure of P and B is given, it is not possible to recover the actual observations $y_1, \ldots, y_N$, even if the set $\mathcal{Y}$ were known, since exchanging, say, the values 1 and 2 in an observation sequence is equivalent to exchanging the first and second columns of B. Thus, in the most optimistic scenario, we can only hope to recover the observation sequence *up to relabelling*.

- If the observation sequence could be exactly recovered (up to relabelling), one could run an HMM parameter estimation algorithm such as EM (Baum-Welch) [18] or spectral learning [19] on such sequence in order to estimate P and B. Thus, the important question that remains is whether one can recover P_{filter}, B_{filter} and $y_1, \ldots, y_N$.

*In this section, to allow the HMM filter to be designed based on a *wrong* model of the HMM whose state is being estimated, we will denote by P_{filter} and B_{filter} the respective transition and sensor matrices used by the HMM filter, instead of the P and B matrices that appear in (4), respectively.

The following result provides a positive answer to the question in the previous paragraph [10]:

Result 1 Assume that P, B are positive matrices. Then, P_{filter}, B_{filter} and $y_1, \ldots, y_N$ can be recovered from a finite number of measurements k using a *nullspace clustering algorithm*[†].

Note that the theorem does *not* require that $P_{\text{filter}} = P$ or $B_{\text{filter}} = B$, *i.e.*, the HMM filter does not need to be perfectly matched to the HMM it aims to estimate!

We now provide a rough idea of the derivation of Result 1, and a sketch of the nullspace clustering method that can solve the inverse filtering problem. The HMM filter update equation (4) can be written as

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

where $\otimes$ denotes the Kronecker product, and vec is the (column) vectorization operator. According to this equation, the matrix $(\pi_{k-1}^T \otimes [\pi_k \mathbb{1}^T - I])$ is known for each k, while $\text{vec}(\text{diag}(b_{y_k}^{\text{filter}}) P_{\text{filter}}^T)$ lies in its nullspace. One can recover $\text{vec}(\text{diag}(b_{y_k}^{\text{filter}}) P_{\text{filter}}^T)$ by "clustering" matrices $(\pi_{k-1}^T \otimes [\pi_k \mathbb{1}^T - I])$ into groups such that the intersection of the nullspaces of the matrices in each group has dimension equal to 1. Once a basis vector for each of these common nullspaces has been determined, $\text{vec}(\text{diag}(b_{y_k}^{\text{filter}}) P_{\text{filter}}^T)$ can be found by normalization (see [10] for details), and then P_{filter} and B_{filter} can be computed by noticing that $\sum_{i=1}^{Y} \text{vec}(\text{diag}(b_i^{\text{filter}}) P_{\text{filter}}^T) = \text{vec}(P_{\text{filter}}^T)$, which gives P, and from this one can determine the columns of B.

To cluster matrices $(\pi_{k-1}^T \otimes [\pi_k \mathbb{1}^T - I])$, one can solve the following convex optimization program, which is a convex relaxation of the nullspace clustering problem we aim to solve:

$$\begin{aligned} \min_{\{w_k\}_{k=1}^N} \quad & \sum_{i=1}^{N} \sum_{j>i}^{N} \|w_i - w_j\|_\infty \\ \text{s.t.} \quad & (\pi_{k-1}^T \otimes [\pi_k \mathbb{1}^T - I]) w_k = 0, \quad \text{for } k = 1, \ldots, N, \\ & w_k \geq \mathbb{1}, \qquad\qquad\qquad\qquad \text{for } k = 1, \ldots, N. \end{aligned}$$

The solutions $w_1, \ldots, w_N$ of this problem correspond to vectors in the nullspace of each matrix $(\pi_{k-1}^T \otimes [\pi_k \mathbb{1}^T - I])$ satisfying the condition stated above.

Note that computing P_{filter}, B_{filter} and the observations $y_1, \ldots, y_N$ is not an estimation problem, in the sense that there is no noise to filter out, thus these quantities can be obtained exactly with an (almost surely) finite number of samples! A necessary condition for this is that the number of samples must be large enough so that the (a priori unknown) observation sequence $y_1, \ldots, y_N$ includes all possible values in $\mathcal{Y}$.

In case the posteriors π_k are contaminated with noise, it is possible to modify the clustering algorithm above to estimate the P_{filter}, B_{filter} matrices, using a technique such as *spherical K-means* [20]. See [9] for a version of such algorithm to recover B_{filter} when P_{filter} is known.

The problem of inverse filtering has been extended by the authors in [11] to the case of linear Gaussian systems, where the HMM filter is replaced by a Kalman filter.

[†]The validity of this result depends on the exactness of the convex relaxation used in the nullspace clustering algorithm.

3.1 Example: Sleep Tracking

To illustrate an application of inverse filtering, we provide in this section an example related to sleep tracking; see [9] for further details. Doctors have classified the sleep stages of humans into five categories: *wake*, *S1*, *S2*, *slow wave sleep* (*SWS*) and *rapid eye movement* (*REM*). These stages are not directly measurable, but commercially available sleep trackers can estimate them through measurements of hearth rate, wrist movements, electroencephalograms (EEG), or other means, by implementing an HMM filter where the underlying state is the current sleep stage of the user.

Inverse filtering can be used in this setup to diagnose possible malfunctioning of a sleep tracker. For simplicity, we will assume that the transition matrix P is known, since manually labeled data is publicly available from which P can be estimated.

By suitably discretizing EEG data of patients from which sleep stages have been manually tagged by doctors (which will be assumed as the true underlying state of an HMM), an HMM filter has been fitted to the data, which resembles a sleep tracker. Figure 3 shows the sleep pattern of a patient, as well as the mean of the posterior distribution provided by our HMM filter. The results of applying inverse filtering to this data, as a function of the standard deviation of the noise present in the posterior of the HMM filter, are presented in Figure 4. As this figure shows, inverse filtering can be highly successful in recovering the observation sequence as well as the sensor matrix when the noise level is reasonably small.

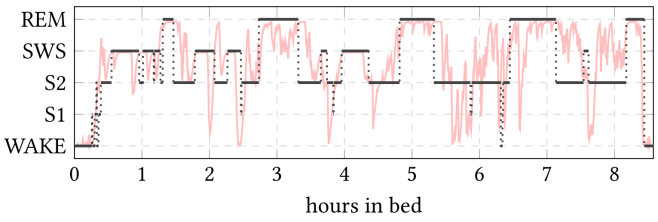

Figure 3: Sleep pattern of a patient over one night: Current sleep stage (in black), and mean of the belief distribution delivered by HMM filter (in red).

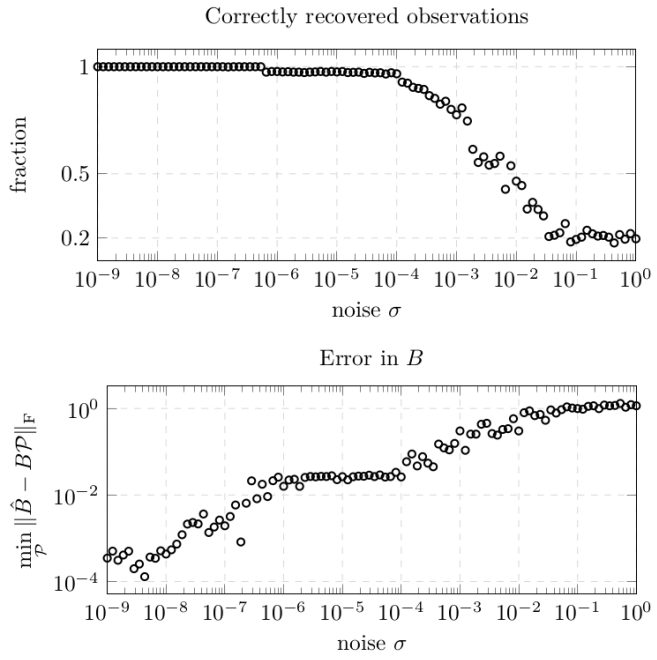

Figure 4: Top plot: Fraction of correctly recovered observations as a function of the standard deviation of the noise contaminating the belief, σ. Bottom plot: Error in the estimation of the sensor matrix B, as a function of σ.

4 Belief Estimation from Decisions

The inverse filtering problem presented in Section 3 showed that from the knowledge of the private belief of the agent, different characteristics of the model can be reconstructed. However, access to the agent's private belief is not always guaranteed. Hence, the following question arises:

Problem 4.1 How can the private belief of an agent be estimated by observing its decisions?

The problem of reconstructing the beliefs of an agent establishes a basis for solving inverse filtering problems, such as the ones from Section 3, and from there questions such as how accurate are the adversary's sensors and how should we design our state sequence (transition kernel) to as accurately as possible estimate the adversary's sensors or confuse it. It is also necessary for, in a realistic setup, analyzing its behaviour as well as predicting its future actions. This problem has practical implications in, not only electronic warfare (where predicting the future actions is central to establishing counter-measures) and cyber-physical security, but also in, e.g., radar calibration and interactive learning [21].

Based on some cost function (or via other means), the adversary acts according to a policy G (which can be either stochastic or deterministic, as discussed in Section 2.2). We present answers

to Problem 4.1 in two different settings. In Section 4.1, we study an approach for estimating the private beliefs given knowledge of the observed actions and of the state sequence. For this case, we present the optimal smoother for inverse filtering in adversarial systems, obtaining a full probability distribution over the possible beliefs. In Section 4.2, we present an alternative approach for estimating the private belief of the decision-maker based on its actions, this time without any assumptions on the belief generating process. We derive a set of private beliefs that are consistent with the observed actions, and present results specialized in a case-study on regime-switching portfolio allocation.

Figure 5 represents the general scheme considered in both parts of this section where the green arrow represents the question posed in Problem 4.1.

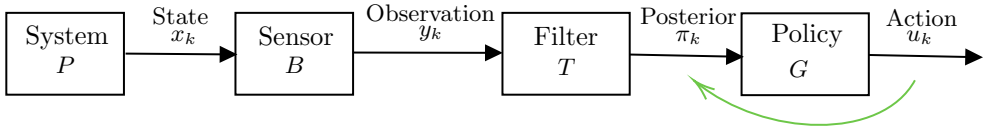

Figure 5: Acting rationally according to a certain policy and its private belief π_k, the agent optimizes a cost function according to its policy and performs an action u_k. The inverse problem here consists of estimating the private belief based on the action performed.

4.1 Estimating Private Beliefs Using a Bayesian Approach

This section considers a way of estimating an agent's private beliefs in a Bayesian framework, where the generative model of the agent's beliefs is known, and the policy can be of a general structure. More specifically, we consider fixed-interval smoothing problems for counter-adversarial autonomous systems, where an agent *i)* measures our current state via a noisy sensor, *ii)* employs an autonomous filtering and control system that computes a posterior estimate (belief) of our state and *iii)* takes an action that we can observe based on its control policy [21, 22]. Mathematically, this corresponds to a game between two players (*us* and the *adversary*), where the knowledge of each player is the following. We know x_k (our true state), the adversary knows y_k and π_k (its measurement and state estimate), and *both* can see the action u_k selected by the adversary. The dynamics of the game are given by the model presented in (5) and as described throughout Section 2.

Based on observed actions u_k and knowledge of our states x_k, our goal is to estimate the past and present beliefs of the adversary. Formally, the central question here is posed as a reformulation of Problem 4.1 as:

Problem 4.2 Suppose the probability distributions P, G and B are known, as well as the initial belief π_0. Given knowledge of our state sequence $x_{0:N}$ and recorded actions of the adversary $u_{1:N}$, what can be said about the corresponding (for us, unobserved) past and present beliefs π_k of the adversary?

4.1.1 Optimal Smoother for Estimating Beliefs

Given measurements up to time $N \geq k$, the goal of Problem 4.2 is to determine the conditional distribution of the belief at time k. This is a well-studied problem for partially observed dynamical models, generally referred to as the *smoothing* problem. [16, 18]. It corresponds to computing the (fixed-interval) smoothing distribution

$$\alpha_{k|N}(\pi) \overset{\text{def.}}{=} p(\pi_k = \pi | u_{1:N}, x_{0:N}), \tag{6}$$

which computes the posteriors over *all the past beliefs*, where N is fixed and $1 \leq k \leq N$. Note that $\alpha_{k|N}(\cdot)$ is a density over Π, where Π_k is the recursive sequence of belief sets:

$$\Pi_k \overset{\text{def.}}{=} \{T(\pi, y) : y \in \mathcal{Y}, \pi \in \Pi_{k-1}\}, \tag{7}$$

initiated with $\Pi_0 = \{\pi_0\}$. Our approach builds on [21] and relies on the optimal inverse filter (which computes the posterior over the *current* belief given a state-sequence and actions). The optimal smoother we compute includes more information than the filter, yielding therefore more accurate estimates (in a mean-squared error sense) [17].

Theorem 4.3 *For a discrete adversarial system, the smoother $\alpha_{k|N}(\pi)$ can be evaluated via*

$$\alpha_{k|N}(\pi) = \frac{\beta_{k|N}(\pi)\alpha_k(\pi)}{\sum_{z \in \Pi_k} \beta_{k|N}(z)\alpha_k(z)}, \tag{8}$$

for $\pi \in \Pi_k$, where $\alpha_k(\pi)$ is the optimal inverse filter $\alpha_k(\pi) \overset{def.}{=} p(\pi_k = \pi | u_{1:k}, x_{0:k})$ that was presented in [21] for the general case and in [14] for discrete systems, and which is also denoted the forward variable. The backward variable, $\beta_{k|N}(\pi)$, can be computed recursively via

$$\beta_{k|N}(\pi) = \sum_{z \in \Pi_{k+1}} G_{z,u_{k+1}}[P]_{x_k,x_{k+1}}[B]_{x_k,y_{\pi,z}}\beta_{k+1|N}(z), \tag{9}$$

for $\pi \in \Pi_k$, initialized by $\beta_{N|N}(\pi) = 1$ for all $\pi \in \Pi_N$.

In summary, to evaluate the smoothing distribution $\alpha_{k|N}(\pi)$ for a discrete system, we:

i) compute the optimal inverse filter $\alpha_k(\pi)$ from [14];
ii) compute the backward variables $\beta_{k|N}(\pi)$ via the recursion (9);
iii) combine the filter $\alpha_k(\pi)$ and $\beta_{k|N}(\pi)$ using (8).

4.1.2 Numerical Results

We consider a three-state system so that $\pi_k \in \mathbb{R}^3$, and the filter $\alpha_k(\pi)$ and smoother $\alpha_{k|N}(\pi)$ yield probability mass functions (pmfs) over the 2-dimensional unit simplex.

In particular, we consider the following randomized adversarial system:

$$P = \begin{bmatrix} 0.7 & 0.2 & 0.1 \\ 0.1 & 0.4 & 0.5 \\ 0.1 & 0.1 & 0.8 \end{bmatrix}, \quad B = \begin{bmatrix} 0.3 & 0.3 & 0.4 \\ 0.1 & 0.8 & 0.1 \\ 0.1 & 0.4 & 0.5 \end{bmatrix}, \tag{10}$$

with $\mathcal{U} = \{1, 2\}$ and a G that yields the first action if $[\pi_k]_1 \geq 0.5$, and the second action otherwise.

Figure 6 illustrates the smoother $\alpha_{3|6}(\pi) = p(\pi_3 = \pi | a_{1:6}, x_{0:6})$ computed via (8). Its CME is marked with a brown circle and the adversary's actual belief as a black star. The smoother, having access to additional data than the optimal inverse filter, (i.e., the actions $u_{4:6}$ and states $x_{4:6}$), rules out one of the potential beliefs of the adversary. Consequently, its CME is closer to the actual belief of the adversary. A more detailed comparison of both can be found in [14].

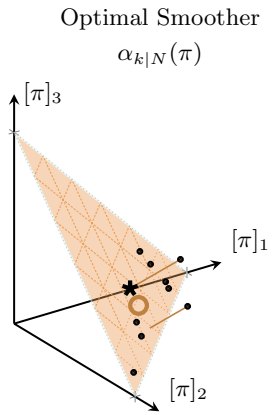

Figure 6: The figure shows the smoother $\alpha_{k|N}(\pi)$ at time $k = 3$ and $N = 6$. The bars display the probability mass function (some beliefs have zero probability). The actual belief of the adversary is marked with a black star (★), and the *conditional mean estimate* (CME) by a circle (○). It should be noted that the smoother's CME lies close to the actual belief.

4.2 Estimating Private Beliefs from Observed Decisions

As discussed in the introduction of this section, papers [9, 11] do not consider how to obtain the private beliefs if only actions based on them are observed. An alternative approach to the Bayesian one outlined above is the following that is based on *inverse optimization*. We consider sequential stochastic decision problems and determine the set of beliefs that a rational adversary could have held given an observed action. To elaborate, above we considered only the resulting policy of the adversary G. In this section, we put structural assumptions on *how G is computed*.

To do this we first need to specify the decision-making process of the agent. The classical model for decision-making under uncertainty assumes that a Bayesian agent chooses that which provides it with the highest expected utility [23–25]. Associated with each state $x_k \in \mathcal{X}$ is a cost function $c(x_k, u_k)$, where $u_k \in \mathcal{U} \subset \mathbb{R}^U$ is the decision variable and $\mathcal{U}$ is the decision set which in this section we assume is a subset of Euclidean space. The rational agent makes its decision $u_k^* \in \mathcal{U}$, based on optimizing its expected cost (conditional on its private information) [23–25]. This expectation depends on the observations O_k collected by the agent, which can include observations directly measured from the system according to the HMM model but in fact, here, we allow for the observations to be arbitrarily generated by the system – such information can

include abstract measurements such as gossip, rumors or other information. The sequential decision-making process is as follows.

1. New information is made available and the private information is updated to $\mathcal{O}_k$.
2. The agent uses its private information to update its private belief, π_k, as in (3).
3. The agent solves the optimization problem:

$$\begin{aligned} \min_{u_k \in \mathcal{U}} \quad & \mathbb{E}_{x_k}\{\, c(x_k, u_k) \mid \mathcal{O}_k \,\} \\ \text{s.t.} \quad & u_k \in \mathcal{C}. \end{aligned} \tag{11}$$

   where $\mathcal{C} \subset \mathcal{U}$ is the feasible set. The conditional expectation is computed with respect to the agent's private information (sigma-algebra) $\mathcal{O}$. Note that with a discrete state-space $\mathcal{X}$, problem (11) becomes

$$\begin{aligned} \min_{u_k \in \mathbb{R}^U} \quad & \sum_{i=1}^{X} [\pi_k]_i c(i, u_k) \\ \text{s.t.} \quad & u_k \in \mathcal{C}, \end{aligned} \tag{12}$$

   when written out explicitly, where $\pi_k \in [0,1]^X$ such that $\mathbb{1}^T \pi_k = 1$.
4. An optimal decision u_k^* – from the set of minimizers of (11) or (12) – is made and publicly announced.
5. Time k is increased by one, and we return to step 1.

Problem 4.1 can be written more specifically in this context as:

Problem 4.4 Decisions are made according to the procedure specified above. The state-space $\mathcal{X} = \{1, \ldots, X\}$ is discrete and the decision set $\mathcal{U} \subset \mathbb{R}^U$ is continuous. The cost functions $\{c(x,u)\}_{x \in \mathcal{X}}$, constraints $\mathcal{C}$ and a decision u_k^* of an agent are known. Determine the set Π_k of private beliefs pi_k that are consistent with the public data.

The assumption that the agent's cost function $c(x,u)$, as well as its constraints $\mathcal{C}$, are known, is not unreasonable: for example, the costs might related to utilities in a game with public rules, or they might be reconstructed (using, e.g., revealed preferences [26, 27]) or estimated (see the case-study in Section 4.2.2). We thus refer to the costs and constraints, together with the announced actions, as *public data*. The *private* information $\mathcal{O}_k$ and the corresponding *private* belief π_k are, however, typically not known – *nor are they supposed to be.*

4.2.1 Algorithm to Estimate Private Beliefs

Problem 4.4 is solved by leveraging results from inverse optimization (e.g., [28–31]). The key idea is that the *Karush–Kuhn–Tucker* (KKT) conditions (e.g., [32, 33]) for decision problem (12) are necessary *and* sufficient under the follwing assumption:

Assumption 1 For fixed x, the function $c(x,u)$ is convex and differentiable in u. The constraints $\mathcal{C}$ are affine $\mathcal{C} = \{u \in \mathbb{R}^U : Au = d, u \geq 0\}$, for some $A \in \mathbb{R}^{N \times U}$ and $d \in \mathbb{R}^N$.

Under Assumption 1, we have the following:

Theorem 4.5 (Solution to Problem 4.4) *Consider the setup in Problem 4.4 under Assumption 1. The agent that made decision u_k^* at time k could have had a private belief $\pi_k \in \mathbb{R}^X$ if and only if this π_k lies in the affine set Π_k, specified by:*

$$\Pi_k = \left\{ \pi \in \mathbb{R}^X : \begin{array}{l} \exists\, \lambda \in \mathbb{R}^U,\ \nu \in \mathbb{R}^N \ \textit{s.t.} \\ \pi^T \mathbb{1} = 1,\ \pi \geq 0,\ \lambda \geq 0, \\ [\lambda]_i = 0 \ \textit{if}\ [u_k^*]_i \neq 0 \quad \textit{for}\ i = 1, \ldots, U, \\ \sum_{i=1}^{X} [\pi]_i \nabla_u c(i, u_k^*) - \lambda + A^T \nu = 0 \end{array} \right\}. \tag{13}$$

The proof of this theorem consists of deriving the KKT conditions and considering not the decision variable u_k but instead the private belief π_k as an unknown variable. Since the cost function is convex and the constraints in problem (12) are defined by affine functions (under Assumption 1), constraint qualification (e.g., [32, 33]) guarantees that these equations are in fact also sufficient for optimality. Hence, a candidate private belief in the simplex $\{\pi \in \mathbb{R}^X : \pi \geq 0, \pi^T \mathbb{1} = 1\}$ would make the observed decision u_k^* optimal in (12) if and only if corresponding ν and λ exist. This means that the adversary observes the action u_k^* performed by the decision-maker and, using the inverse optimization relation (13), reconstructs a set of beliefs, $\Pi(u_k^*)$ that includes the private belief of the decision-maker. The extent to which its privacy is compromised is discussed in [13].

In [12] are provided bounds $\overline{\pi}_k \in \mathbb{R}^X$ and $\underline{\pi}_k \in \mathbb{R}^X$ on the private belief, as well as the solution to the problem of finding the closest private belief $\widehat{\pi}_k$ that is consistent with the public data given a prior estimate $\pi_k^0 \in \mathbb{R}^X$ of the private belief π_k.

4.2.2 Case-Study: Estimating the Investor's Belief from their Portfolio Allocation

In this section, we apply the result presented above in a financially themed case-study. More specifically, we estimate the private belief of a risk-averse investor based on observing his portfolio allocations.

In investment sciences, there have been substantial advances in *regime-switching market models* (e.g., [34–36]) that take into account that market conditions, also known as trends, (randomly) switch between different states, $x_k \in \mathcal{X}$. The investor is faced with the question: *Given U risky assets, how should a fixed amount of capital be invested so as to maximize the risk-adjusted return under switching market conditions?*

Denote the portfolio allocation vector by $u_k \in \mathbb{R}^U$, where a fraction $[u_k]_i$ of the total capital will be invested in asset i. Usually, one requires that $\mathbb{1}^T u_k = 1$ (that the full capital is exposed to the market), that $u_k \geq 0$ (it is only allowed to buy assets, not sell them short), and that investments are held for one full time-period. Each market state x_k results in a different mean vector $\mu_{x_k} \in \mathbb{R}^U$, and a corresponding covariance matrix $\Sigma_{x_k} \in \mathbb{R}^{U \times U}$, for the different assets. For a given risk aversion parameter $\gamma \in \mathbb{R}_+$, which quantifies how the investor trades potential

return against risk, a regime-switching mean-variance portfolio allocation problem is of the form:

$$\begin{aligned} \min_{u_k \in \mathbb{R}^U} \quad & \mathbb{E}_{x_k} \left\{ \gamma u_k^T \Sigma_{x_k} u_k - \mu_{x_k}^T u_k \mid \mathcal{O}_k \right\} \\ \text{s.t.} \quad & \mathbb{1}^T u_k = 1, \; u_k \geq 0, \end{aligned} \tag{14}$$

whose solution provides the investor with the portfolio giving the optimal risk-adjusted return for period k. Here, $\mathcal{O}_k$ is the investor's *private* information that is employed to compute the posterior distribution of the current market state x_k.

Public stock data allows everyone to form estimates of the expected returns and covariances under different market conditions, meaning that, in practice, the cost functions of a Markowitz-type investor can be approximated. However, clearly, the success of an investor is closely related to how well he or she can estimate the current market conditions. This estimation depends on *private information*; for example, rumors or privileged information. Reconstructing an investor's private belief could allow for, e.g., change detection; which could indicate insider trading, and/or reverse engineering trading strategies.

4.2.3 Numerical Results

In order to visualize the results, we consider synthetic three-regime portfolio allocation problems, as in (14), with $X = 3$ and $U = 3$. This means that both action and belief spaces are represented by two-dimensional unit simplices.

Figure 7 exemplifies on the left a case where the set of consistent private beliefs Π_k is not a singleton. We computed this set using Theorem 4.5, and it is depicted as the green region (line). As expected, the actual private belief π_k (marked with a black star) lies inside this set.

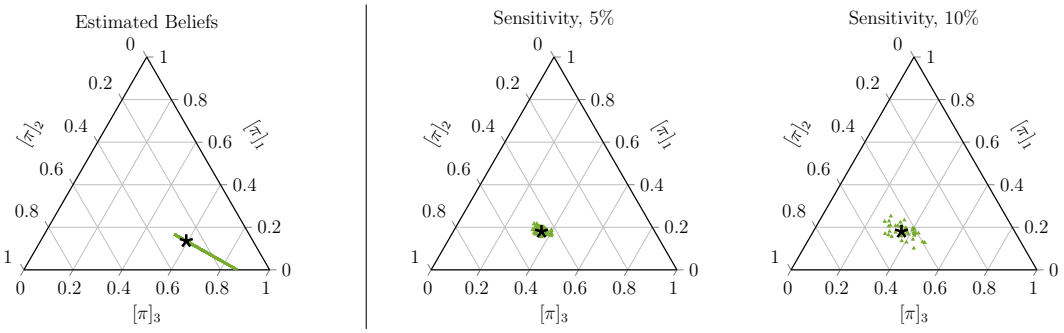

Figure 7: The actual private belief π_k is marked with a black star ($\star$) in all figures. On the left, the darker green region ($\blacktriangle$) corresponds to Π_k – the set of beliefs consistent with public data (defined in Theorem 4.5). On the right, the green triangles ($\blacktriangle$) mark the beliefs that are estimated when perturbed cost functions are used. Note that a different example (one in which the set Π_k is singleton) is used compared to that in the left figure.

Moreover, the cost functions $c(x, u)$ incurred by an agent are not always known with certainty. To explore the sensitivity of our results, we added random perturbations to the cost

functions when estimating the private belief. To illustrate the results, we consider, on the right, an example where the set Π_k is singleton (i.e., it contains only the actual private belief π_k). The results of 40 simulations are displayed. Random zero-mean Gaussian elementwise perturbations of standard deviations 5% and 10% were added to both the means μ_x and covariances Σ_x. Clearly, the results are robust to small perturbations since all the estimated private beliefs lie close to the actual private belief. A more elaborate description together with additional insights can be found in [12].

5 Belief Protection: Counter-Adversarial Decision-Making

Algorithms for estimating the private belief of an agent, like those presented in Section 4, are useful since they form the foundation for predicting its future actions. However, they also enable a number of attack vectors for a malicious actor. Assume that the actions of the Bayesian agent (the *agent*, or the *decision-maker*) are seen by an adversarial agent (the *adversary*), that maliciously aims to estimate its private belief. In this section we study the counter-adversarial problem of how the decision-maker can protect its private belief from the adversary, while, at the same time, limiting its increase in cost:

Problem 5.1 How should an agent modify its optimal decision in order to not expose its private belief, while limiting its cost increase due to taking a suboptimal decision?

This counter-adversarial decision-making problem has a vast number of applications, ranging from security of cyber-physical systems to protection of investment strategies, passing by analyzing how social and economic herding occurs. In social learning, privacy constraints prevent the estimation and disclosure of an agent's private belief to the other agents. Another potential area of application is the portfolio allocation setup presented in Section 4.2.2. Consider that a competing investor wants to make an informed investment decision as the one of a successful investor, but does not have its expertise and knowledge. By observing the actions of the main investor, the competitor wants to infer the private belief of this investor.

In summary, the setup considered is as follows. At each time step, the agent described in Section 4.2 – henceforth referred to as the *original decision-maker* (ODM) – collects information regarding its environment and performs an action. Assuming it is aware of an adverse threat, it must consider the question: *if I publicly announce the decision u_k^*, what is the set of beliefs $\Pi(u_k^*)$ consistent with my decision that the adversary can determine?* As was demonstrated in Section 4.2, the set includes the actual private belief, π_k, and, therefore, the privacy of the ODM is compromised. We propose a *Counter-adversarial Decision-Maker* (CDM), that adds to its decision-making process the *obfuscator* block shown in blue in Figure 8 to conceal its private belief from the adversary. While the ODM performs the optimal action u_k^* with cost c_k^*, the CDM performs a suboptimal action $\widetilde{u}_k$ with cost $\widetilde{c}_k$.

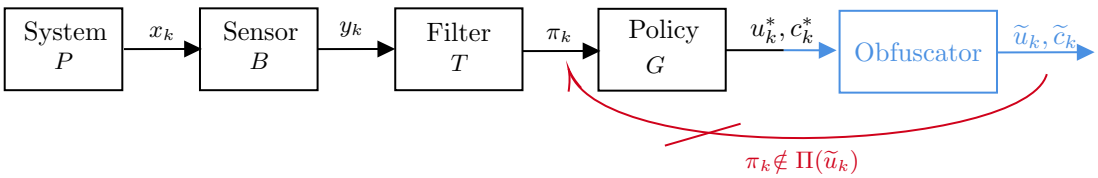

Figure 8: A *Counter-adversarial Decision-Maker* (CDM) uses an *obfuscator* block to transform its decision u_k^* into a suboptimal decision $\widetilde{u}_k$ with cost $\widetilde{c}_k$, by optimizing a privacy measure $\Psi(\widetilde{u}_k)$. The adversary observes the decision (now $\widetilde{u}_k$) and, again, reconstructs a set of beliefs (now $\Pi(\widetilde{u}_k)$). The new decision $\widetilde{u}_k$ is chosen such that, unlike for the ODM's in Figure 5, the CDM's privacy is not compromised.

Performing a suboptimal action $\widetilde{u}_k$ entails an increase in cost ($\widetilde{c}_k \geq c_k^*$). Thus, protecting its privacy comprises a trade-off between how much the decision-maker is able to obscure its private belief *versus* how much it is willing to pay for doing so. Problem 5.1 can be formulated as:

$$\begin{aligned} \max_{\widetilde{u}_k \in \mathcal{U}} \quad & \Psi(\widetilde{u}_k) \\ \text{s.t.} \quad & \widetilde{u}_k \in \mathcal{C}, \\ & \mathbb{E}_{x_k}\{\, c(x_k, \widetilde{u}_k)\} \leq c_k^*(1+b), \end{aligned} \tag{15}$$

which should be interpreted as follows. Obfuscating the decision-maker's private belief consists in making a suboptimal decision $\widetilde{u}_k$, such that the set of private beliefs reconstructed by the adversary, $\Pi(\widetilde{u}_k)$, from (13), maximizes a certain *privacy measure* $\Psi(\widetilde{u}_k)$. The last constraint represents how much (measured by the *obfuscation cost budget* $b \in \mathbb{R}_+$) the agent allocates to obfuscating its private belief.

Problem (15) is generally computationally intractable to solve exactly since the privacy measures $\Psi(\widetilde{u}_k)$ are typically not concave. We propose a probabilistic framework based on a similar concept to that of randomized actions in Markov decision processes, covered in [37]. This idea originated from the introduction of "mixed strategies" in the field of game theory [38]. For approximate methods to convexify the problem, we use Monte Carlo integration [39]. This method is particularly useful for integration in high-dimensional spaces, since it has been shown to have an accuracy in terms of the standard deviation of the error independent of the number of dimensions.

We assume that the distribution $p_{\widetilde{u}_k}(\cdot)$ is a probability mass function over a finite set of points $\{\widetilde{u}_k^{(l)}\}_{l=1}^M$. Then, the problem solved by the CDM becomes the following:

Theorem 5.2 (Obfuscation of the Private Belief on Average) *Optimization problem* (15)

can, assuming a policy concentrated in M actions, be written as:

$$\begin{aligned} \max_{p\in\mathbb{R}^M} \quad & \sum_{l=1}^{M} [p]_l \Psi(\widetilde{u}_k^{(l)}) \\ s.t. \quad & [p]_l = 0 \ \textit{if}\ \widetilde{u}_k^{(l)} \notin \mathcal{C}, \\ & \sum_{l=1}^{M} [p]_l \left\{ \sum_{i=1}^{X} \pi_i c(i, \widetilde{u}_k^{(l)}) \right\} \leq c^*(1+b), \\ & [p]_l \geq 0, \quad l = 1, \ldots, M, \quad \sum_{l=1}^{M} [p]_l = 1, \end{aligned} \tag{16}$$

which is a finite-dimensional linear program and, therefore, computationally efficient to solve using existing solvers.

5.1 Numerical Results

In Section 4.2.3, we saw that a rival investor (i.e., an adversary) that has access to less (or worse) private information than a main investor from Section 4.2.2, is able to estimate a set of private beliefs consistent with the private belief of the investor solving (14). In this section we show how the investor can utilize (16) to protect its privacy.

Suppose there are three risky assets and three market states (i.e., $U = X = 3$). The investor allocates 10% budget to preserve its privacy ($b = 0.1$) and aims to do so by using a maximal obfuscation measure:

$$\Psi(\widetilde{u}_k) = \operatorname{dist}(\pi_k, \Pi(\widetilde{u}_k)) = \min_{y\in\Pi(\widetilde{u}_k)} \|\pi_k - y\|_2 . \tag{17}$$

In other words, this measure states that the agent wants the reconstructed set to be as distant as possible from the actual private belief. Other privacy measures are discussed in [13].

The left plot of Figure 9 shows the actions chosen by the decision-makers at a certain timestep k. The ODM solves problem (12) and selects action u_k^*, which is the optimal action if there is no adversary. The CDM solves problem (16) and takes a random action between those in the set $\{\widetilde{u}_k^{(l)}\}_{l=1}^M$ (defined in (16)) marked as (∘), where each has a probability given by the vector p, here represented as the bar on top of each action (−). In this case, two actions had a positive probability of being chosen and the chosen action is denoted as $\widetilde{u}_k$.

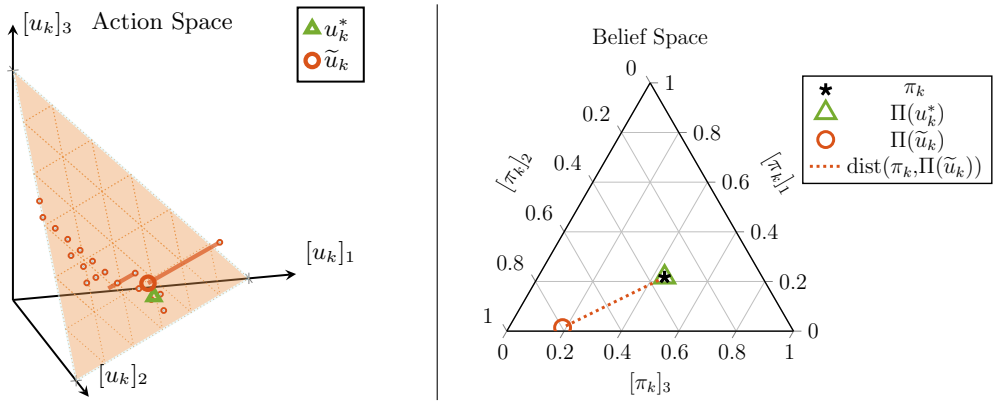

Figure 9: The left plot represents the action space with the actions chosen by the two decision-makers (△, ○), and the right plot the belief space with the respective sets of beliefs reconstructed by the adversary according to the actions chosen; at a certain time step.

The sets of beliefs that the adversary can reconstruct from each of the agent's actions at this timestep (described in Section 4.2) are illustrated on the right plot of Figure 9. In this example, all the sets have a single element. The actual private belief π_k of the decision-makers is shown in black ($\star$). The privacy of the ODM is compromised, since its private belief belongs to the set of beliefs reconstructed by the adversary ($\pi_k \in \Pi(u_k^*) \Leftrightarrow dist(\pi_k, \Pi(u_k^*)) = 0$). On the other hand, the suboptimal action performed by the CDM has allowed it to successfully obfuscate its private belief ($\pi_k \notin \Pi(\widetilde{u}_k)$). Nevertheless, according to the *maximal obfuscation* criterion chosen, the level of privacy depends on the distance between π_k and $\Pi(\widetilde{u}_k)$, shown in dashed lines and computed by (17).

Over fifty time steps, while the ODM's privacy is always compromised (the distance is zero), the CDM managed to increase its privacy to an average of 0.8 (where the maximum distance between two points in the simplex is approximately 1.4). The cost increased by 9%, which is still less than the allocated budget of 10%. More details can be found in [13].

6 Conclusions

In this work we explored the inverse filtering, belief estimation and privacy protection problems on HMMs. For the first we showed that HMM parameters and sensor measurements can be reconstructed from posterior distributions of an HMM filter by using a nullspace clustering algorithm, which was exemplified with a sleep tracking example.

Since the posterior distribution (also called belief) is often not known, we next considered the problem of estimating the private belief of the agent from its actions, in two different setups. In one setup, taking the full generative model into account to compute the optimal smoother for the private beliefs, we obtain a full probabilistic characterization of how likely different beliefs are. In another setup, given only measurements of the decision-maker's decisions and known preferences, we reconstruct a set of beliefs that either includes or is equal to the true belief.

The problem of estimating agents' private beliefs has implications in, for example, social learning and portfolio allocation, where it raises important questions of privacy. We thus next showed how an agent can modify its optimal decision in order to not expose its private belief, while limiting its cost increase due to taking a suboptimal decision.

All in all, this paper provides (counter-)adversarial frameworks for estimating (protecting) private beliefs of agents, and from there other characteristics of interest of their decision-making process. In the future, it would be interesting to analyse the computational complexity resulting from the estimation of growing sets of potential beliefs. Another extension would be to study the case of mismatched systems (e.g., where the adversary does not have perfect knowledge of the transition kernel P). We would also like to formulate the counter-adversarial problem in a game-theoretical framework, where the adversary is aware of the obfuscation mechanism used by the decision-maker. It would also be interesting to apply counter-adversarial decision-making to the smoother.

References

[1] Wahlberg B, Hjalmarsson H, and Annergren M, On optimal input design in system identification for control, In *49th IEEE Conference on Decision and Control (CDC)*, pages 5548–5553, 2010, doi: 10.1109/CDC.2010.5717863.

[2] Annergren M, Larsson C A, Hjalmarsson H, Bombois X, and Wahlberg B, Application-Oriented Input Design in System Identification: Optimal Input Design for Control [Applications of Control], *IEEE Control Systems Magazine*, 2017, **37**(2): 31–56, doi: 10.1109/MCS.2016.2643243.

[3] Kalman R E, When is a linear control system optimal?, *Journal of Basic Engineering*, 1964, **86** (1): 51–60.

[4] Zhang H, Umenberger J, and Hu X, Inverse optimal control for discrete-time finite-horizon Linear Quadratic Regulators, *Automatica*, 2019, **110**: 108593.

[5] Zhang H, Li Y, and Hu X, Discrete-Time Inverse Linear Quadratic Optimal Control over Finite Time-horizon Under Noisy Output Measurements, *Control Theory and Technology*, 2021.

[6] Li Y, Yao Y, and Hu X, Continuous-time inverse quadratic optimal control problem, *Automatica*, 2020, **117**: 108977.

[7] Ng A Y, Russell S J, and others , Algorithms for inverse reinforcement learning., In *Proccedings of the International Conference on Machine Learning(ICML)*, volume 1, page 2, 2000.

[8] Abbeel P and Ng A Y, Apprenticeship learning via inverse reinforcement learning, In *Proceedings of the twenty-first International Conference on Machine Learning*, page 1, 2004.

[9] Mattila R, Rojas C, Krishnamurthy V, and Wahlberg B, Inverse filtering for hidden Markov models, *Advances in Neural Information Processing Systems (NIPS) 2017*, 2017, **30**.

[10] Mattila R, Rojas C R, Krishnamurthy V, and Wahlberg B, Inverse Filtering for Hidden Markov Models with Applications to Counter-Adversarial Autonomous Systems, 2020.

[11] Mattila R, Rojas C R, Krishnamurthy V, and Wahlberg B, Inverse filtering for linear Gaussian

state-space models, In *2018 IEEE Conference on Decision and Control (CDC)*, pages 5556–5561. IEEE, 2018.

[12] Mattila R, Lourenço I, Rojas C R, Krishnamurthy V, and Wahlberg B, Estimating Private Beliefs of Bayesian Agents Based on Observed Decisions, *IEEE Control Systems Letters*, 2019, **3**(3): 523–528, ISSN 2475-1456, doi: 10.1109/LCSYS.2019.2911802.

[13] Lourenço I, Mattila R, Rojas C R, and Wahlberg B, How to Protect Your Privacy? A Framework for Counter-Adversarial Decision Making, *Proceedings of the 59th IEEE Conference in Decision and Control (CDC)*, 2020, pages 1785–1791.

[14] Mattila R, Lourenço I, Krishnamurthy V, Rojas C R, and Wahlberg B, What Did Your Adversary Believe? Optimal Filtering and Smoothing in Counter-Adversarial Autonomous Systems, In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5495–5499, 2020.

[15] Norris J R, *Markov Chains*, Cambridge university press, 1998.

[16] Krishnamurthy V, *Partially Observed Markov Decision Processes: From Filtering to Controlled Sensing*, Cambridge University Press, 2016.

[17] Anderson B D O and Moore J B, *Optimal Filtering*, Prentice-Hall, 1979.

[18] Cappé O, Moulines E, and Rydén T, *Inference in Hidden Markov Models*, Springer, 2005, ISBN 0387402640.

[19] Hsu D, Kakade S M, and Zhang T, A spectral algorithm for learning Hidden Markov Models, *Journal of Computer and System Sciences*, 2012, **78**(5): 1460–1480, ISSN 00220000, doi: 10.1016/j.jcss.2011.12.025.

[20] Buchta C, Kober M, Feinerer I, and Hornik K, Spherical k-means clustering, *Journal of Statistical Software*, 2012, **50**(10): 1–22.

[21] Krishnamurthy V and Rangaswamy M, How to Calibrate Your Adversary's Capabilities? Inverse Filtering for Counter-Autonomous Systems, *IEEE Transactions on Signal Processing*, Dec 2019, **67**(24): 6511–6525, ISSN 1941-0476, doi: 10.1109/TSP.2019.2956676.

[22] Kuptel A, Counter Unmanned Autonomous Systems (CUAxS): Priorities. Policy. Future Capabilities, *Multinational Capability Development Campaign (MCDC)*, 2017, pages 15–16.

[23] Mas-Colell A, Whinston M D, and Green J R, *Microeconomic theory*, volume 1, Oxford university press New York, 1995.

[24] Luenberger D G, *Microeconomic theory*, Mcgraw-Hill College, 1995.

[25] Machina M J, Choice under uncertainty: Problems solved and unsolved, *Journal of Economic Perspectives*, 1987, **1**(1): 121–154.

[26] Varian H R, Revealed preference, *Samuelsonian economics and the twenty-first century*, 2006, pages 99–115.

[27] Varian H R, *Microeconomic analysis*, volume 3, Norton, New York, 1992.

[28] Ahuja R K and Orlin J B, Inverse optimization, *Operations Research*, 2001, **49**(5): 771–783.

[29] Iyengar G and Kang W, Inverse conic programming with applications, *Operations Research Letters*, 2005, **33**: 319 – 330, ISSN 0167-6377, doi: https://doi.org/10.1016/j.orl.2004.04.007, URL `http://www.sciencedirect.com/science/article/pii/S016763770400063X`.

[30] Zhang J and Xu C, Inverse optimization for linearly constrained convex separable programming problems, *European Journal of Operational Research*, 2010, **200**(3): 671–679.

[31] Keshavarz A, Wang Y, and Boyd S, Imputing a convex objective function, In *IEEE International Symposium on Intelligent Control*, pages 613–619, Sep. 2011, doi: 10.1109/ISIC.2011.6045410.

[32] Boyd S and Vandenberghe L, *Convex Optimization*, Cambridge University Press, 2004, ISBN 0521833787.

[33] Rockafellar R T, *Convex Analysis*, Princeton University Press, 1970.

[34] Yin G G and Zhou X Y, Markowitz's mean-variance portfolio selection with regime switching: from discrete-time models to their continuous-time limits, *IEEE Transactions on Automatic Control*, 2004, **49**(3): 349–360.

[35] Elliott R J, Siu T K, and Badescu A, On mean-variance portfolio selection under a hidden Markovian regime-switching model, *Economic Modelling*, 2010, **27**(3): 678–686.

[36] Nystrup P, Madsen H, and Lindström E, Dynamic portfolio optimization across hidden market regimes, *Quantitative Finance*, January 2018, **18**(1): 83–95, ISSN 1469-7688, 1469-7696, doi: 10.1080/14697688.2017.1342857.

[37] Puterman M L, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*, John Wiley & Sons, Inc., 1994.

[38] McKinsey J C C, *Introduction to the Theory of Games*, Courier Corporation, 2003.

[39] Davis P J and Rabinowitz P, *Methods of numerical integration*, Courier Corporation, 2007.