A DIRECT ALGORITHM FOR JOINT OPTIMAL SENSOR SCHEDULING AND MAP STATE ESTIMATION FOR HIDDEN MARKOV MODELS

David Jun, David M. Cohen, Douglas L. Jones

University of Illinois at Urbana-Champaign
Department of Electrical and Computer Engineering

ABSTRACT

Sensing systems with multiple sensors and operating modes warrant active management techniques to balance estimation quality and measurement costs. Existing literature shows that in the joint sensor-scheduling and state-estimation problem for HMMs, estimator optimization can be done independently of the scheduler at each time step. We investigate the special case when a MAP estimator is used, and show how the joint problem can be converted to a standard Partially Observable Markov Decision Process (POMDP), which in turn enables us to use POMDP solvers. As this approach is highly redundant, we derive a direct solution, which exploits the separability property while still utilizing standard solvers. When compared to standard techniques, the direct algorithm provides savings by a factor of the state-space dimension. Numerical results are given for an example motivated by wildlife monitoring.

Index Terms— sensor management, POMDP, controlled HMM

1. INTRODUCTION

Sensing devices are becoming a ubiquitous part of life. Most commonly, these devices are used to estimate the state of the surrounding environment (e.g. there is a person in the room, an enemy plane is approaching, etc.). In many applications, practical limitations prevent sensors from running all the time. For example, battery-powered devices are limited by energy. Alternatively, radar systems may need to trade off reliability of measurement with the probability that the measurement is detected by an enemy. In these cases, resources must be appropriately managed. These are typical examples of the problems addressed in the field of *sensor management*, which has received growing interest in the past decade; see [1] for a recent survey.

The fundamental problem for sensor management is that sequential sensing actions are taken over time, where each action corresponds to choosing a sensor, which generates new observations that provide additional information [2]. The goal is to construct a causal policy that uses all of the information collected thus far to determine the next sensing action, which has some associated sensing cost. The temporal correlations of the state of the world are modeled as Markov, and each sensing action determines the quality of the observations, along with generally influencing the underlying Markov chain.

In this paper, we consider problems where the underlying Markov chain is not affected by any of the sensing actions, which can be interpreted as a hidden Markov model (HMM) with multiple sensing options. This assumption is motivated by monitoring applications with stationary, passive sensing platforms such as the acoustic wildlife monitoring application presented in [3]. The problem of joint optimal sensor scheduling and state estimation for HMMs was first formulated in [4] and is illustrated in Fig. 1. It was shown in [4] that

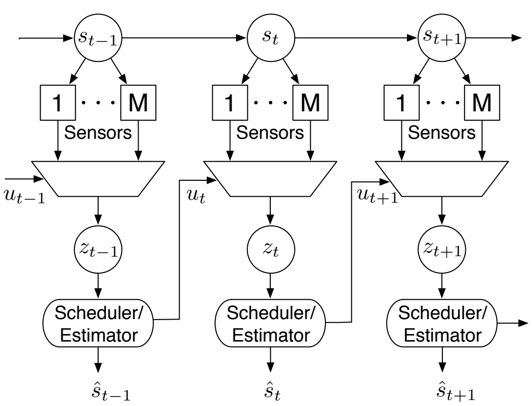

Fig. 1. A standard illustration of an HMM augmented with multiple sensors and a closed-loop sensor scheduler and state estimator. A standard HMM would have only the states s_t and observations z_t connected according to the typical HMM assumptions. This figure was adapted from [4].

the problem is separable; the estimation policy can be solved first, independent of the scheduler, and the scheduling policy is constructed using dynamic programming techniques.

Dynamic programming is a solution technique for solving Markov decision problems (MDP), which was first for-

The authors acknowledge the support of the Multiscale Systems Center, one of six research centers funded under the Focus Center Research Program (FCRP), a Semiconductor Research Corporation program.

mulated as an optimal control problem [5], where actions correspond more generally to physical actuation and control of a system. The MDP framework is utilized widely for operations research and robotic planning problems [6].

Partially observable Markov decision processes (POMDP) are MDPs where the state space is not fully observable (i.e., only noisy observations of the states are available). As even finite-horizon POMDPs are PSPACE-complete [7], the focus of current research is on developing solution methods that scale to very large problems [8]. Sensor management problems are a special type of POMDPs, where actions are limited to sensing actions.

In this paper, we focus on the maximum *a posteriori* (MAP) estimator, showing that by enumerating all possible sensor/estimation pairs, the scheduling problem can be cast as a standard POMDP. While this enables the use of standard existing solvers, it is highly redundant due to the separation property. Our contribution is to derive a direct solution, enabling us to leverage standard solvers while achieving speed-ups that are typically on the order of the dimension of the state space. Our contribution relies on exact solution methods which do not scale with problem size; nonetheless, useful practical speedups are achieved for our motivating applications, which exhibit tens of states and sensors.

2. SENSOR SELECTION AND STATE ESTIMATION

2.1. Problem Formulation

We assume there is a single discrete-time Markov chain with a finite state-space, denoted by $\mathcal{S}$, that is observed by M noisy sensors which give measurements from a finite observation space $\mathcal{Z}$. Let $\mathcal{T}$ denote the transition kernel, such that $\tau(s, s')$ is the one-step probability of transitioning from state s to s'. At any given time, only a single sensor, $u \in \mathcal{M} \triangleq \{1, \dots, M\}$, can be used to gain information about the underlying state s, where $\mathcal{O}$ is an observation function and $o(s, u, z)$ is the probability of observing $z \in \mathcal{Z}$, given that the true state of nature is s and sensor u is used.

The belief state, $b(s)$, is an $|S|$-dimensional vector in a probability simplex $\mathcal{B}$ defined over $\mathcal{S}$, and is updated according to Bayes' rule after observing z using sensor u:

$$b_u^z(s') = \frac{o(s', u, z)}{p(z|u, b)} \cdot \sum_{s \in \mathcal{S}} \tau(s, s') \cdot b(s) \tag{1}$$

where $p(z|u, b) = \sum_{s'} o(s', u, z) \sum_s \tau(s, u, s') b(s)$ is the probability of observing z. For this scheduling/estimation problem, it was shown in [4] that b is a sufficient statistic for constructing optimal scheduling and estimation policies. In particular, assume that at time t sensor u_t is used, z_t is observed, and b is updated to $b_{u_t}^{z_t}$. Then,

$$\hat{s}_t = \epsilon(b_{u_t}^{z_t}) \tag{2}$$

$$u_{t+1} = \mu(b_{u_t}^{z_t}) \tag{3}$$

where $\epsilon : \mathcal{B} \to \mathcal{S}$ and $\mu : \mathcal{B} \to \mathcal{M}$ are the estimation and scheduling policies, respectively.

The class of cost functions considered in [4] consist of an estimation error term weighted by a state-dependent sensor usage cost. For problems with this structure, it was shown that the estimation and scheduling problems are separable; this follows because the state estimate $\hat{s}_t$ does not affect the future evolution of $b_{u_t}^{z_t}$. This observation allows us to freely choose any state estimator.

In this paper, we consider the special case of using the MAP estimator. A joint policy (ϵ, μ) is evaluated by a value function $V^{(\epsilon,\mu)} : \mathcal{B} \to \mathbb{R}$, which we define to be the expected discounted reward[1] for implementing the joint policy (ϵ, μ), given initial belief b:

$$V^{(\epsilon,\mu)}(b) = \mathbb{E}_{(\epsilon,\mu)} \left\{ \sum_{t=0}^{\infty} \gamma^t \cdot \sum_{s \in \mathcal{S}} \left[\mathbb{1}_{\{s-\epsilon(b_t)\}} - c\left(s, \mu(b_t)\right) \right] \cdot b_t(s) \middle| b_0 = b \right\} \tag{4}$$

where $0 \leq \gamma < 1$ is a discount factor, and $\mathbb{1}$ is the indicator function.

2.2. Equivalence to a Standard POMDP

Although not explicitly stated in [4], choosing the MAP estimator results in a standard POMDP problem [9]. To see this, define $\pi \triangleq (\epsilon, \mu)$ as the policy, with the associated action space:

$$\mathcal{A} = \{(e, u) : e \in S, u \in \mathcal{M}\} \tag{5}$$

which is just the enumeration of all estimation, sensor pairs; note that $|\mathcal{A}| = M \cdot |\mathcal{S}|$. Next, define $\mathcal{R} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ to be the reward function, where

$$r(s, a) = \mathbb{1}_{\{s-e\}} - c\left(s, u\right), \quad \text{for } a = (e, u) \tag{6}$$

which is just the term in the square bracket in (4). Thus, the scheduling/estimation problem can be formulated as a standard POMDP, given as the tuple $\langle \mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \mathcal{Z}, \mathcal{O}, \gamma \rangle$.

With this notation, the optimal value function, V^*, is the unique solution to Bellman's equation [5]:

$$V^*(b) = \max_{a \in \mathcal{A}} \left\{ \sum_{s \in \mathcal{S}} r(s, a) b(s) + \gamma \mathbb{E}\left[V^*(b_a^z) \mid a, b \right] \right\} \tag{7}$$

3. A DIRECT SOLUTION

The advantage of interpreting the scheduling/estimation problem as a standard POMDP is that we can use standard solvers

[1] From now on, we consider reward, which is the negative of cost.

(such as the enumeration algorithm, incremental pruning, or the Witness algorithm, for example); see [6] for a detailed exposition. Although this is convenient, it is redundant given that we know that the scheduling and estimation problems are separable. In this section, we derive a more direct solution.

With slight abuse in notation, let c_u be an $|\mathcal{S}|$-dimensional vector, with $c_u(s) = c(u, s)$. Then, writing the value iteration equation starting from (4) for iteration n:

$$\begin{aligned} V_n(b) &= \max_{(e,u)\in A} \sum_{s\in\mathcal{S}} \mathbb{1}_{\{s-e\}} b(s) - c_u^T b + \gamma \mathbb{E}[V_{n-1}(b_u^z) \mid u, b] \\ &= \max_{e\in\mathcal{S}} \{b(e)\} + \max_{u\in\mathcal{M}} \left\{-c_u^T b + \gamma \mathbb{E}[V_{n-1}(b_u^z) \mid u, b]\right\} \end{aligned} \tag{8}$$

where the second line shows why the scheduling/estimation problem is separable; the first max defines the MAP estimator, and the second max represents the value iteration step if one were to consider the following *auxiliary* POMDP: $\langle \mathcal{S}, \mathcal{M}, \mathcal{T}, -c, \mathcal{Z}, \mathcal{O}, \gamma \rangle$.

Thus, we can use any POMDP solver and make one value iteration on the auxiliary problem, generating some minimum set of alpha vectors, denoted as $\Gamma_n^\star$. Again abusing notation, let g_e be an $|\mathcal{S}|$-dimensional unit vector with a one in the e^{th} dimension; define $\Gamma_\mathcal{S} = \{g_e : e \in \mathcal{S}\}$. Continuing from (8),

$$V_n(b) = \max_{\phi\in\Gamma_\mathcal{S}} \phi^T b + \max_{\alpha\in\Gamma_n^\star} \alpha^T b \tag{9}$$

$$= \max_{\phi\in\mathcal{S},\alpha\in\Gamma_n^\star} (\phi + \alpha)^T b \tag{10}$$

$$= \max_{\beta\in\bar{\Gamma}_n} \beta^T b \tag{11}$$

where

$$\bar{\Gamma}_n = \Gamma_\mathcal{S} \oplus \Gamma_n^\star \triangleq \{\phi + \alpha : \phi \in \Gamma_\mathcal{S}, \alpha \in \Gamma_n^\star\} \tag{12}$$

and $\oplus$ is known as the cross-sum operator. This method of *enumerating* all of the possible alpha vectors to represent the value function at iteration n is part of what is known as the enumeration algorithm. The second step of the enumeration algorithm is a pruning step, denoted as $\Gamma_n = \text{Prune}(\bar{\Gamma}_n)$, which reduces the set of alpha vectors to its minimum set. In its simplest form, Prune systematically checks each vector in $\bar{\Gamma}_n$ to see if it dominates at some belief state, which serves as a witness that the vector is in the minimum set. The existence of such a point can be determined by solving a linear program (LP):

$$\begin{aligned} &\max_{b\in\mathcal{B}} \quad \delta \\ \text{s.t.} \quad &\hat{\gamma}^T b \geq \gamma^T b + \delta \quad \forall\, \gamma \neq \hat{\gamma} \in \bar{\Gamma}_n \\ &\sum_{s=1}^{\mathcal{S}} b(s) = 1 \\ &b(s) \geq 0 \quad \forall\, s \in \mathcal{S} \end{aligned}$$

If the program does not return $\delta \geq 0$, $\hat{\gamma}$ is removed. Once this procedure is completed for every vector in $\bar{\Gamma}_n$, Γ_n will be the minimum set. At worst, Prune($\bar{\Gamma}_n$) must solve $|\bar{\Gamma}_n|$ LPs.

Solving these LPs is by far the most computationally expensive part of any POMDP algorithm. As such, it is common to measure the complexity of an algorithm by the number of LPs it must solve. It is well known (see [6]) that the maximum number[2] of LPs any POMDP algorithm must solve at iteration n is $|\mathcal{A}| \cdot |\Gamma_{n-1}|^{|\mathcal{Z}|}$.

Claim: The maximum number of LPs that the direct algorithm will have to solve at iteration n is $|\mathcal{M}| \cdot |\Gamma_{n-1}|^{|\mathcal{Z}|} + |\Gamma_n| \cdot |\mathcal{S}|$.

Proof

1. The *auxiliary* POMDP defined above requires at most $|\mathcal{M}| \cdot |\Gamma_{n-1}|^{|\mathcal{Z}|}$ LPs.
2. $|\text{Prune}(\Gamma_n^\star \oplus \Gamma_\mathcal{S})| \triangleq |\Gamma_n|$.
3. For two minimum sets Γ_1 and Γ_2, $|\Gamma_1 \oplus \Gamma_2| \geq \max\{|\Gamma_1|, |\Gamma_2|\}$.
4. Therefore, $|\Gamma_n^\star| \leq |\Gamma_n|$, and $|\Gamma_n^\star \oplus \Gamma_\mathcal{S}| \leq |\Gamma_n| \cdot |\mathcal{S}|$.
5. Prune($\Gamma_n^\star \oplus \Gamma_\mathcal{S}$) requires at most $|\Gamma_n| \cdot |\mathcal{S}|$ NPs.
6. Both prune operations combined require no more than $|\mathcal{M}| \cdot |\Gamma_{n-1}|^{|\mathcal{Z}|} + |\Gamma_n| \cdot |\mathcal{S}|$ LPs. ∎

Compare this with the indirect method that solves the converted POMDP problem in Section 2.2 using standard algorithms. This method solves at most $|\mathcal{S}| \cdot |\mathcal{M}| \cdot |\Gamma_{n-1}|^{|\mathcal{Z}|}$ LPs. In situations where $|\Gamma_{n-1}| \approx |\Gamma_n|$, the complexity is dominated by the $|\Gamma_{n-1}|^{|\mathcal{Z}|}$ term. In these cases, the direct method provides a complexity reduction of almost a factor of $|\mathcal{S}|$.

Finally, note that the space of estimator decisions does not have to be the same as the state space. It is straightforward to adopt these results for any decision space. For example, if the decision space is $\mathcal{D}$, substitute $|\mathcal{D}|$ for $|\mathcal{S}|$ above. An example in which the decision space is different than the state space is given in the next section.

4. RESULTS

Practical savings of the direct algorithm are shown on a sample problem. For purposes of experimentation, model parameters are representative, but were not learned experimentally. This problem is motivated by an acoustic bird-monitoring application. The bird can exist in one of three states: absent, present and calling, or present and resting. The sensing options are sleep or sense. The sensor's observation is either silent, call A, or call B. The absent and resting states are indistinguishable to the sensor. Sleeping comes at a cost of 1

[2] In the case of the batch enumeration algorithm, this is the exact number of LPs solved at each iteration

μA, while sensing costs 2.6 mA, which represent average current draw on a low-power microcontroller. We consider two variants of the problem: one in which we want to decide if the bird is absent or present, and one in which we want to decide if the bird is absent, calling, or resting. The state-transition matrix is

$$A = \begin{pmatrix} 0.9 & 0.07 & 0.03 \\ 0.0 & 0.8 & 0.2 \\ 0.5 & 0.15 & 0.8 \end{pmatrix},$$

where $A_{ij} = \tau(i, j)$. Not sensing results in a uniform observation matrix, and the observation matrix for sensing is

$$B(\text{sense}) = \begin{pmatrix} 0.98 & 0.01 & 0.01 \\ 0.31 & 0.48 & 0.21 \\ 0.98 & 0.01 & 0.01 \end{pmatrix},$$

where $B(\text{sense})_{ij} = o(i, \text{sense}, j)$. Finally, the discount factor is $\gamma = 0.3$.

Table 1 shows a comparison between the indirect and direct algorithms on the problem variations described above. Performance is evaluated with both the total number of LPs solved and total execution time. The solution with two decisions contains 6 vectors, and the solution with three decisions contains 24 vectors.

Here we have a case where $|\Gamma_{n-1}| \approx |\Gamma_n|$. As a result, we see LP reductions close to a factor of (# decisions) as expected. This reduction corresponds very closely to the decrease in total execution time, justifying the use of LPs as a complexity metric. For the two-decision problem, LPs were reduced by a factor of 1.69, and the algorithm completed 1.64 times faster. For three decisions, LPs were reduced by a factor of 2.78 and the algorithm completed 2.88 times faster.

# decisions	algorithm	# LPs	time (s)
2	Indirect	2,305	3.4
	Direct	1,358	2.1
3	Indirect	21,784	44.4
	Direct	7,826	15.4

Table 1. Complexity of indirect and direct algorithms

5. CONCLUSIONS

Sensor management for state estimation in HMMs is a well-studied problem. When a MAP estimator is chosen, we have shown that the problem can be converted into a POMDP. As such, standard POMDP solvers can be utilized. While a straightforward approach, standard solvers ignore a lot of the structure of the sensor management problem. Namely, the state decision can be considered independently from the sensor scheduler, whereas standard methods would treat them jointly. Guided by this insight, we showed that the problem can be separated into a MAP estimator and an auxiliary POMDP. Using a direct algorithm, complexity can be reduced by almost a factor of the state-space dimension. Further, the direct algorithm has no preference as to how the auxiliary POMDP is solved; any algorithm will work. This paper utilized batch enumeration, the simplest method for combining the MAP and POMDP portions. However, any number of tricks in the spirit of traditional POMDP algorithms could be applied to further speed up this step.

Finally, the wildlife monitoring applications motivating this work consist of problems that are small to medium-sized, relative to the size of the problems being addressed by current POMDP research; our proposed approach provides useful speedups for simple yet practical sensor management problems.

6. REFERENCES

[1] A.O Hero and D. Cochran, "Sensor management: Past, present, and future," *IEEE Sensors Journal*, vol. 11, no. 12, Dec. 2011.

[2] A.O. Hero, *Foundations and Applications of Sensor Management*, Springer, 2008.

[3] D. Jun, D.L. Jones, T.P. Coleman, W.J. Leonard, and R. Ratnam, "Practical sensor management for an energy-limited detection system," in *IEEE Conference on Acoustics, Speech and Signal Processing*, Mar. 2012.

[4] J. Evans and V. Krishnamurthy, "Optimal sensor scheduling for hidden Markov model state estimation," *International Journal of Control*, vol. 74, no. 18, 2001.

[5] D.P. Bertsekas, *Dynamic Programming and Optimal Control*, vol. 1, Athena Scientific Belmont, MA, 1995.

[6] A.R. Cassandra, *Exact and Approximate Algorithms for Partially Observable Markov Decision Processes*, Ph.D. thesis, Brown University, 1998.

[7] C.H Papadimitriou and J.N Tsitsiklis, "The complexity of Markov decision processes," *Mathematics of Operations Research*, vol. 12, no. 3, 1987.

[8] G. Shani, J. Pineau, and R. Kaplow, "A survey of point-based POMDP solvers," *Autonomous Agents and Multi-Agent Systems*, 2012.

[9] R.D. Smallwood and E.J. Sondik, "The optimal control of partially observable Markov processes over a finite horizon," *Operations Research*, vol. 21, no. 5, 1973.