FINITE DIMENSIONAL ALGORITHMS FOR OPTIMAL SCHEDULING OF HIDDEN MARKOV MODEL SENSORS

Vikram Krishnamurthy

Department of Electrical Engineering
University of Melbourne,
Victoria 3010, Australia
vikram@ee.mu.oz.au

Bo Wahlberg

Department of S3
Royal Institute of Technology,
100 44 Stockholm Sweden
bo.wahlberg@s3.kth.se

ABSTRACT

Consider the Hidden Markov model estimation problem where the realization of a single Markov chain is observed by a number of noisy sensors. The sensor scheduling problem for the resulting Hidden Markov model is as follows: Design an optimal algorithm for selecting at each time instant, one of the many sensors to provide the next measurement. Each measurement has an associated measurement cost. The problem is to select an optimal measurement scheduling policy, so as to minimize a cost function of estimation errors and measurement costs. The problem of determining the optimal measurement policy is solved via stochastic dynamic programming. An optimal finite dimensional algorithm is presented along with numerical results.

1. INTRODUCTION

In many signal processing applications several types of sensors are available for measuring a given process. However physical and computational constraints often impose the requirement that at each time instant, one is able to use only one out of a possible total of L sensors. In such cases, one has to make the decision: *Which sensor (or mode of operation) should be chosen at each time instant to provide the next measurement.* Typically associated with each type of measurement is a per unit-of-time measurement cost, reflecting the fact that some measurements are more costly or difficult to make than others, although they may contain more useful or reliable information. The problem of optimally choosing which one of the L sensor observations to pick at each time instant is called the *sensor scheduling problem.* The resulting time sequence which at each instant specifies the best sensor to choose is termed the *sensor schedule sequence.*

Several papers have studied the sensor scheduling problem for systems with linear Gaussian dynamics where linear measurements in Gaussian noise are available at a number of sensors (see [1] for the continuous-time problem and [6] for the discrete-time problem). In this paper we study the discrete-time sensor scheduling problem for Hidden Markov Model (HMM) sensors. We assume that the underlying process is a finite state Markov chain. At each time instant, observations of the Markov chain in white noise are made at L different sensors. However, only one sensor observation can be chosen at each time instant. The aim is to devise an algorithm that optimally picks which single sensor to use at each time instant, in order to minimize a given cost function.

In our recent work [4], we formulated the HMM sensor scheduling problem and presented the dynamic programming functional recursion for determining the optimal sensor schedule. However, the dynamic programming equations in [4] do not directly translate into practical solution methodologies. The fundamental problem is that at each time instant of the dynamic programming recursion, one needs to compute the dynamic programming cost over an uncountably infinite set. Thus an approximate algorithm was then presented in [4] which was based on discretizing the dynamic programming recursion to a finite grid.

The main contribution of this paper is to present an optimal **finite dimensional** solution to the HMM sensor scheduling problem. Unlike the grid based approximate solution presented in [4], we determine a closed-form finite dimensional solution to the dynamic programming recursion. We show that the solution to the dynamic programming equation is piecewise linear and convex. An algorithm is given for computing these piecewise linear segments. The finite dimensional scheduling algorithms presented in this paper are similar to those recently used in the operations research (see [5] for a tutorial survey) and in robot navigation systems [3] for the optimal control of Partially observed Markov Decision Processes (POMDP). (We will use the terms Hidden Markov Model control and POMDP interchangeably).

There are numerous applications of the optimal scheduling of sensors. Some applications include [6], [4], [3]: finding the optimum channel allocation among various components of a measurement vector when they must be transmitted over a time-shared communication channel of limited bandwidth; finding the optimum timing of measurements when the number of possible measurements is limited because of energy constraints; and finding the optimum trade-off between measurement of range and range rate in radar systems with a given ambiguity function. There is also growing interest in flexible sensors such as multi-mode radar which can be configured to operate in one of many modes.

2. SIGNAL MODEL AND PROBLEM FORMULATION

Let $k = 0, 1, \dots$ denote discrete time. Assume X_k is an S-state Markov chain with state space $\{e_1, \dots, e_S\}$. Here e_i denotes the S-dimensional unit vector with 1 in the i-th position and zeros elsewhere. This choice of using unit vectors to represent the state space considerably simplifies our subsequent notation. Define the $S \times S$ transition probability matrix $A = [a_{ji}]_{S\times S}$ where $a_{ji} =$

This work was partially supported by the Center for Expertise in Networked Decision Systems

$P(X_k = e_i | X_{k-1} = e_j)$. Denote the initial probability vector π_0 of the Markov chain as

$$\pi_0 = [\pi_0(i)]_{S\times 1} \text{ where } \pi_0(i) = P(X_0 = i) \qquad (1)$$

2.1. Sensor Scheduling Problem

Assume there are L noisy sensors available which can be used to give measurements of X_k. At each time instant k, we are allowed to pick only one of the L possible sensor measurements. Motivated by the physical and computational constraints alluded to in the introduction, we assume that having picked this sensor, we are not allowed to look at any of the other $L-1$ observations at time k.

Let $u_k \in \{1, \dots L\}$ denote the sensor picked at time k. The observation measured by this sensor is denoted as $y_k(u_k)$. Suppose $u_k = l$, where $l \in \{1, \dots, L\}$. We assume that the measurement $y_k(l)$ of the l-th sensor belongs to a known finite set of symbols $O_1(l), O_2(l), \dots, O_{M_l}(l)$. For $u_k \in \{1, \dots, L\}$, denote the symbol probabilities as $b_i(u_k = l, y_k(u_k) = O_m(l)) = P(y_k(u_k) = O_m(u_k) | X_k = e_i, u_k = l)$, $i = 1, 2, \dots, S$. These represent the probability that an output $O_m(l)$ is obtained given that the state of the Markov chain is e_i and that the lth sensor is chosen. The symbol probabilities are assumed known. Finally define the symbol probability matrix

$$B(u_k, O_m(u_k)) = \text{diag}\begin{bmatrix} b_1(u_k, O_m(u_k)) \\ \vdots \\ b_S(u_k, O_m(u_k)) \end{bmatrix}$$

Let $Y_k = \{u_1, u_2, \dots, u_k, y_1(u_1), y_2(u_2), \dots, y_k(u_k)\}$ denote information available at time k upon which to base estimates and sensor scheduling decisions. The sensor scheduling and estimation problem proceeds in three stages for each $k = 0, 1, \dots, N-1$, where N is a fixed positive integer

1) Scheduling: Based on Y_k we generate $u_{k+1} = \mu_{k+1}(Y_k)$ which determines which sensor is to be used at the next time step.

2) Observation: We then observe $y_{k+1}(u_{k+1})$ where u_{k+1} is the sensor selected in the previous stage.

3) Estimation: After observing $y_{k+1}(u_{k+1})$ we generate our best estimate $\hat{X}_{k+1}$ of the state of the Markov chain X_{k+1} as $\hat{X}_{k+1} = \mathbb{E}\{X_{k+1} | Y_{k+1}\}$. Here $\hat{X}_{k+1}$ denotes the Hidden Markov Model filtered state estimate at time $k+1$ defined as

$$\hat{X}_{k+1} = \mathbb{E}\{X_{k+1} \mid Y_{k+1}\} = \sum_{i=1}^{S} e_i \, P(X_{k+1} = e_i \mid Y_{k+1}), \qquad \hat{X}_0 = \pi_0 \qquad (2)$$

where π_0 is the initial apriori state estimate at time $k = 0$.

Define the sensor scheduling sequence $\mu = \{\mu_1, \mu_2, \dots, \mu_N\}$ and say that the scheduling sequences are admissible if μ_{k+1} maps Y_k to $\{1, \dots, M\}$. Note that μ is a sequence of functions.

We assume the following cost structure. If based on the observation at time k, the decision is made to choose $u_{k+1} = l$ (i.e. to choose the l-th sensor at time $k+1$ where $l \in \{1, \dots, L\}$), then the instantaneous cost incurred at time k is

$$\alpha_k(l) \, \|X_k - \hat{X}_k\|_D + c_k(X_k, l). \qquad (3)$$

Here $\alpha_k(l)$, $l = 1, 2 \dots, L$ are known positive scalar weights. The "distance" function D is assumed to be a convex function with $D : \mathbb{R}^S \to \mathbb{R}$. $\|X_k - \hat{X}_k\|_D$ denotes the state estimation error (with respect to the distance function D) at time k due to choosing the sensor schedule $u_1, \dots, u_k$. For example if D is the l_2 norm, then $\|X_k - \hat{X}_k\|_D$ denotes the Euclidean distance between X_k and its estimate $\hat{X}_k$. In such a case the instantaneous cost is the mean square error in the state estimate when using the sensor u_{k+1}. Finally, $c_k(X_k, u_k)$ denotes the instantaneous cost of using the sensor u_{k+1} when the state of the Markov chain is X_k.

Our aim is to find the optimal sensor schedule to minimize the total accumulated cost J_μ from time 1 to N over the set of admissible control laws:

$$J_\mu = \mathbb{E}\Big\{ \sum_{k=0}^{N-1} \alpha_k(u_{k+1}) \|X_k - \hat{X}_k\|_D + \sum_{k=0}^{N-1} c_k(X_k, u_{k+1})\} + \alpha_N \|X_N - \hat{X}_N\|_D \Big\} \qquad (4)$$

where $u_{k+1} = \mu_{k+1}(Y_k)$.

2.2. Information State Formulation

As is standard with such stochastic control problems – in this section, we convert the partially observed stochastic control problem to a fully observed stochastic control problem defined in terms of the *information state*.

Let π_k denote information state at time k. with elements $\pi_k(i) = P(X_k = e_i | Y_k)$, $i = 1, 2, \dots, S$. Also because we have assumed that X_k is a unit vector $\in \{e_1, \dots, e_N\}$, we straightforwardly have $\pi_k = \hat{X}_k$. The information state π_k is a sufficient statistic to describe the current state of a HMM (see [2]). The information state update is computed straightforwardly by the HMM state filter (also known as the "forward algorithm" [4]):

$$\pi_{k+1} = \frac{B(u_{k+1}, y_{k+1}(u_{k+1})) \, A' \, \pi_k}{\mathbf{1}'_S B(u_{k+1}, y_{k+1}(u_{k+1})) \, A' \, \pi_k} \qquad (5)$$

where $\mathbf{1}_S$ represents an S-dimensional vector of ones.

Let $\mathcal{P}$ denote the set of all information states π. That is,

$$\mathcal{P} = \Big\{ \pi \in \mathbb{R}^S : \mathbf{1}'_S \pi = 1, \quad 0 \le \pi(i) \le 1 \text{ for all } i \in \{1 \dots, S\} \Big\} \qquad (6)$$

Consider the cost functional (4). Define

$$c_k(u_{k+1}) = \begin{bmatrix} c_k(e_1, u_{k+1}) & \cdots & c_k(e_S, u_{k+1}) \end{bmatrix}'$$

Using the smoothing property of conditional expectation, the cost functional of (4) can be rewritten in the form (see [2, Chapter 5] for details)

$$J_\mu = \mathbb{E}\{C_N(\pi_N) + \sum_{k=0}^{N-1} C_k(\pi_k, u_{k+1})\} \qquad (7)$$

where $u_{k+1} = \mu_{k+1}(\pi_k)$,

$$C_N(\pi_N) = \alpha_N g(\pi_N)' \pi_N$$
$$C_k(\pi_k, u_{k+1}) = \alpha_k(u_{k+1}) g(\pi_k)' \pi_k + c'_k(u_{k+1}) \pi_k \qquad (8)$$

Here $g(\pi_k)$ denotes the S dimensional "distance" vector

$$g(\pi_k) = \begin{bmatrix} \|e_1 - \pi_k\|_D & \|e_2 - \pi_k\|_D & \cdots & \|e_S - \pi_k\|_D \end{bmatrix}' \qquad (9)$$

We now have a fully observed control problem in terms of the information state π: Find an admissible control law, μ, which minimizes the cost functional of (7), subject to the state evolution equation of (5).

3. STOCHASTIC DYNAMIC PROGRAMMING

We use stochastic dynamic framework to optimally solve the HMM sensor scheduling problem. In this section we first present the backward dynamic programming recursion. The main result is to give an explicit solution to this dynamic programming recursion.

3.1. Dynamic Programming Formulation

Define the "value-to-go" function as

$$J_N(\pi) = \mathbb{E}\{C_N(\pi_N)|\pi_n = \pi\}$$

$$J_k(\pi) = \min_{\mu_{k+1},\cdots,\mu_N} \mathbb{E}\left\{C_N(\pi_N) + \sum_{t=k}^{N-1} C_t(\pi_t, \mu_{t+1}(\pi_t)) \mid \pi_k = \pi\right\}$$

Then the dynamic programming recursion proceeds backwards in time from $k = N$ to $k = 0$ as follows:

$$J_N(\pi_N) = C_N(\pi_N)$$

and for $k = N-1, N-2, \ldots, 0$

$$J_k(\pi_k) = \min_{u_{k+1}\in\{1,\ldots,L\}} [\, C_k(\pi_k, u_{k+1}) + \sum_{m=1}^{M_{u_{k+1}}} J_{k+1}\left(\frac{B(u_{k+1}, O_m(u_{k+1}))\, A'\, \pi_k}{\mathbf{1}' B(u_{k+1}, O_m(u_{k+1}))\, A'\, \pi_k}\right) \mathbf{1}' B(u_{k+1}, O_m(u_{k+1})) A' \pi_k \,], \qquad \text{for all } \pi_k \in \mathcal{P} \quad (10)$$

Finally, the optimal cost starting from the initial condition π_0 is given by $J_0(\pi_0)$ and if $u^*_{k+1} = \mu^*_{k+1}(\pi_k)$ minimises the right hand side of (10) for each k and each π_k, the optimal scheduling policy is given by $\mu^* = \{\mu^*_1, \mu^*_2, \ldots, \mu^*_N\}$.

As it stands the above dynamic programming equations (10) has a major problem: The information state π is continuous valued. Hence the above dynamic programming equations (10) do not directly translate into practical solution methodologies since $J_k(\pi)$ needs to be evaluated at each $\pi \in \mathcal{P}$, an uncountably infinite set. In [4], a suboptimal algorithm was obtained by discretising the information state space $\mathcal{P}$ to a finite grid. The aim of this section is to show that this approximation is un-necessary. Indeed the above dynamic programming can be explicitly solved – we show that $J_k(\pi)$ is convex and piecewise linear in π.

3.2. Finite Dimensional Optimal Scheduler

In this subsection we consider the case where the distance vector $g(\pi)$ defined in (9) is a piecewise linear and continuous function of π. Examples of such piecewise linear cost functions in the information state include the case when D is a discrete norm. Also piecewise linear costs can be used to approximate nonlinear cost functions of the information state uniformly and arbitrarily closely.

Let $\mathcal{P}_1, \ldots, \mathcal{P}_R$, denote the partition of the information state space simplex $\mathcal{P}$ over which $g(\pi)$ is piece-wise linear. The piecewise linearity implies that there exist R vectors $\bar{g}_1, \bar{g}_2, \ldots, \bar{g}_R$ such that $g(\pi)$ can be represented as

$$g(\pi) = \sum_{r=1}^{R} \bar{g}_r I(\pi \in \mathcal{P}_r)$$

where $I(\cdot)$ denotes the indicator function

$$I(\pi \in \mathcal{P}_r) = \begin{cases} 1 & \text{if } \pi \in \mathcal{P}_r \\ 0 & \text{otherwise} \end{cases}$$

Due to the convexity and piecewise linearity an equivalent and more convenient representation for our purposes is

$$g'(\pi)\pi = \min_{r\in 1,2\ldots,R} \bar{g}'_r \pi$$

The following theorem shows that the solution to the DP recursion is convex piecewise linear and thus completely characterized at each time k by a finite set of vectors (piecewise linear segments).

Theorem 3.1 *At each time instant k, the value to go function $J_k(\pi)$ is convex and piece-wise linear. $J_k(\pi)$ has the explicit representation*

$$J_k(\pi) = \min_{i\in\Gamma_k} \gamma'_{i,k}\pi \text{ for all } \pi \in \mathcal{P} \quad (11)$$

where Γ_k is a finite set of S-dimensional vectors.

PROOF. The proof is by induction. At time N, from (10), $J_k(N) = \min_r \bar{g}'_r \pi_N$ which is of the form (11).

Assume at time $k+1$ that $J_{k+1}(\pi)$ has the form $\min_{i\in\Gamma_{k+1}} \gamma'_{i,k+1}\pi$. Then it is easily shown that the value function at time k is piecewise linear. □

3.3. Optimal Algorithm

Theorem 3.1 shows that the solution to the DP recursion (10) is convex piecewise linear and completely characterized at each time k by the finite set of vectors Γ_k defined in (11). Thus we need to devise an algorithm for computing the set Γ_k at each time k. There are numerous linear programming based algorithms in the POMDP literature such as Sondik's algorithm [7] Cheng's algorithm [5], and the Witness algorithm [3] that can be used to compute the finite set of vectors Γ_k, see
http://www.cs.brown.edu/research/ai/pomdp/index.html for a tutorial exposition with graphics.

Note that the entire dynamic programming algorithm and hence the computing the value-to-go function vectors Γ_k are off-line and independent of the data. The vector set Γ_k is used during real-time simulations to give the optimal sensor schedule depending on the computed information state π_k

4. NUMERICAL EXAMPLE

The scenario involves an incoming aircraft.
State Space: The state space for this problem comprises of how far the aircraft is currently from the base station discretized into 3 distinct distances $d_1 = 10$, $d_2 = 5$, $d_3 = 1$. The transition probability matrix of X_k is

$$A = \begin{bmatrix} 0.8 & 0.2 & 0 \\ 0.1 & 0.8 & 0.1 \\ 0 & 0.2 & 0.8 \end{bmatrix} \quad (12)$$

Sensors: Assume that two sensors are available, i.e., $L = 2$.
1. `active`: This active sensor (e.g. radar) yields accurate measurements of the distance of the aircraft but renders the base more visible to the aircraft. Thus the `active` sensor is merely a HMM

state filter.
2. `predict`: Employ no sensor – predict the state of the aircraft. Thus the `predict` sensor is merely a HMM state predictor.
At any time instant k, $u_k \in \{\texttt{active}, \texttt{predict}\}$ denotes which one of the above two sensors is used.
Observation Symbols: When using the `active` sensor, the observation symbols at each time k consists of distance measurements to the base station d_1, d_2, d_3. In addition there is an additional observation symbol; `nothing` which results when the `predict` sensor is used (i.e. no observation is made). In terms of the notation of Sec.2, the number of possible observation symbols is $M = 4$ and $O_1 = d_1, O_2 = d_2, O_3 = d_3, O_4 = \texttt{nothing}$.

We assume that the distance the `active` sensor reports is never more than 1 discrete location away from the true distance. The `active` sensor detects the true distance with probability p. The `predict` sensor only records the observation `nothing` with probability 1. In particular defining the $N \times M$ matrix of symbol probabilities $\bar{B}(u_k) = [\bar{B}_{ij}(u_k)] = P\{y_k(u_k) = O_j | X_k = e_i\}$, we assign

$$\bar{B}(u_k = \texttt{active}) = \begin{bmatrix} p & 1-p & 0 & 0 \\ 1-p/2 & p & 1-p/2 & 0 \\ 0 & 1-p & p & 0 \end{bmatrix}$$

$$\bar{B}(u_k = \texttt{predict}) = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Costs: Our cost function comprises two components.
1. Sensor Costs:

$$c(X_k = e_i, u_{k+1} = \texttt{active}) = \frac{r^{\texttt{active}}}{d_i} + \rho^{\texttt{active}}$$
$$c(X_k = e_i, u_{k+1} = \texttt{predict}) = \frac{r^{\texttt{predict}}}{d_i} + \rho^{\texttt{predict}}$$

We assume $\rho^{\texttt{active}} = 8$, $\rho^{\texttt{predict}} = 5$ meaning that the operating cost of using the `active` sensor is higher than the `predict` sensor. Also the cost incurred is inversely proportional to the distance of the aircraft. This reflects the fact that when the aircraft is close to the base-station the threat is greater. We chose the gains $r^{\texttt{active}} = 2$ and $r^{\texttt{predict}} = 5$.
2.State Estimation Error Cost: For the state estimation error cost component, we consider the l_2 cost $\alpha_k(1 - \hat{X}'\hat{X})$. We chose $\alpha_k = 10$.
Results: With the above setup, we used the POMDP program available from the website
http://www.cs.brown.edu/research/ai/pomdp/index.html to optimally solve the HMM sensor scheduling problem.

All our simulations were run on a Pentium-2 400 MHz personal computer. using the "Incremental Pruning" algorithm developed by Cassandra, et.al. in 1997 (This is currently the fastest known algorithm for solving POMDPs, see [3]).

We ran the POMDP program for the above parameters over a horizon of $N = 7$ for different values of p (probability of detection). In all cases, no apriori assumption was made on the initial distance (state) of the target – thus we chose the information state (filtered density) at time 0 as $\pi_0 = [1/3,\ 1/3,\ 1/3]'$, see (1). We approximated the l_2 cost function by a uniform piecewise linear triangular interpolation consisting of 9 triangular patches.

Figure 1 shows the costs incurred for the optimal sensor schedule versus p. Also shown are the cost incurred if only the `active` or `predict` sensor is used alone. It can be inferred from Fig.1 that when the probability of detection is low ($0.5 < p < 0.65$) using the `predict` sensor alone does better than using the `active` sensor. The reason is that for low probability of detection, the `active` sensor (HMM filter) yields inaccurate state estimates. This together with the fact that the `active` sensor has a higher operating cost than the `predict` sensor, means that for low p it costs less to using a HMM predictor and not incur any cost obtaining extremely inaccurate measurements. Note that in all cases, the optimal sensor schedule incurs the smallest cost.

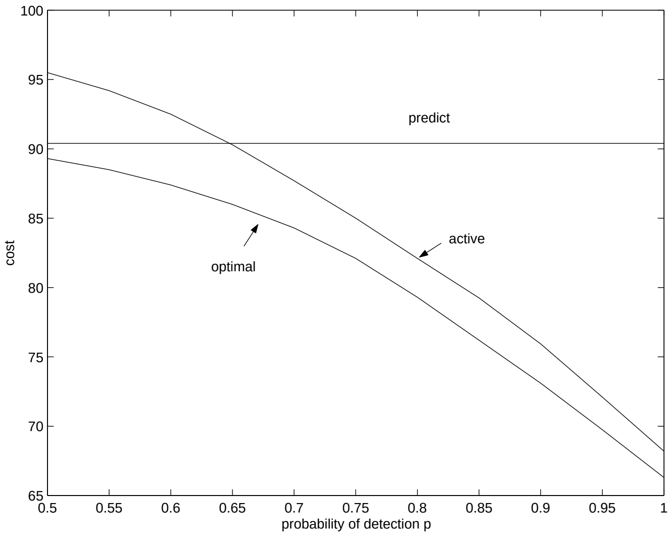

Fig. 1. Finite Horizon Costs

5. REFERENCES

[1] M. Athans. On the determination of optimal costly measurement strategies for linear stochastic systems. *Automatica*, 8:397–412, 1972.

[2] D.P. Bertsekas. *Dynamic Programming and Optimal Control*, volume 1 and 2. Athena Scientific, Belmont, Massachusetts, 1995.

[3] A. R. Cassandra. *Exact and Approximate Algorithms for Partially Observed Markov Decision Process*. PhD thesis, Brown University, 1998.

[4] J.S. Evans and V. Krishnamurthy. Optimal sensor scheduling for hidden Markov models. In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP98*, pages 2161–2164, Seattle, May 1998.

[5] W.S. Lovejoy. A survey of algorithmic methods for partially observed Markov decision processes. *Annals of Operations Research*, 28:47–66, 1991.

[6] L. Meier, J. Perschon, and R.M. Dressler. Optimal control of measurement systems. *IEEE Transactions on Automatic Control*, 12(5):528–536, October 1967.

[7] R.D. Smallwood and E.J. Sondik. Optimal control of partially observable Markov processes over a finite horizon. *Operations Research*, 21:1071–1088, 1973.