When to observe the state of a Markov process

Aditya Mahajan
McGill University

INFORMS Applied Probability Society Conference
11 July, 2017

Acknowledgments

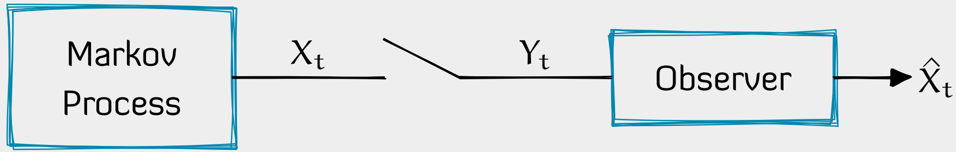
Part of the results presented are based on:

Shuman et al, "Measurement scheduling for soil moisture sensing: From physical models to optimal control," Proc IEEE 2010.

Acknowledgments

- David Schuman (Macalester College)
- Ashutosh Nayyar (University of Southern California)
- Mingyan Liu (University of Michigan)
- Demos Teneketzis (University of Michigan)

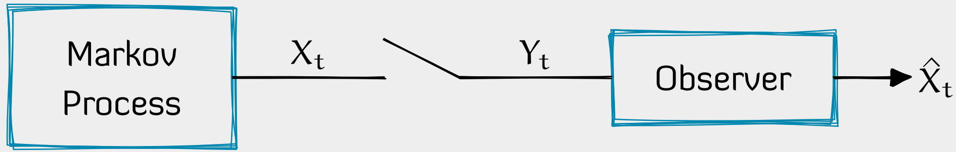
- Jalal Arabneydi (Concordia Univeristy)



Markov Process $\{X_t\}_{t\geqslant 0}$, where $X_t \in \mathcal{X}$ (finite). Transition probability matrix P.

Observer

- At the beginning of time slot t:
  - decides whether to take an observation ($U_t = 1$) or not ($U_t = 0$)
  - observation $Y_t = \begin{cases} X_t, & \text{if } U_t = 1 \\ \mathfrak{E}, & \text{if } U_t = 0 \end{cases}$
  - choosing $U_t = 1$ has a cost c
- At the end of time slot t:
  - decides an estimate $\hat{X}_t \in \mathcal{X}$
  - incurs an estimation error $d(X_t, \hat{X}_t)$



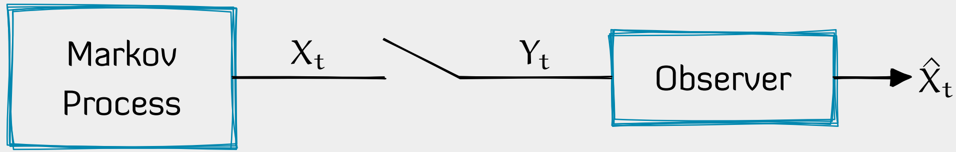
Optimization problem

Choose:

- Observation policy $f = (f_0, f_1, \dots)$, where $U_t = f_t(Y_{0:t-1}, U_{0:t-1})$
- Estimation policy $g = (g_0, g_1, \dots)$, where $\hat{X}_t = g_t(Y_{0:t}, U_{0:t})$

to minimize

$$\mathbb{E}\left[\sum_{t=0}^{\infty} \alpha^t \left[cU_t + d(X_t, \hat{X}_t)\right]\right]$$



Optimization problem

Choose:

- Observation policy $f = (f_0, f_1, \dots)$, where $U_t = f_t(Y_{0:t-1}, U_{0:t-1})$
- Estimation policy $g = (g_0, g_1, \dots)$, where $\hat{X}_t = g_t(Y_{0:t}, U_{0:t})$

to minimize

$$\mathbb{E}\left[\sum_{t=0}^{\infty} \alpha^t \left[c U_t + d(X_t, \hat{X}_t)\right]\right]$$

Remarks

- This is a POMDP (partially observable Markov decision process)
- How do we identify optimal strategies in a computationally tractable manner?

Outline

Belief state MDP

- Optimal estimation policy can be computed off-line without knowning the observation policy
- Optimal observation policy is characterized by a convex set (the set where the optimal action is to take an observation).

Outline

Belief state MDP

- Optimal estimation policy can be computed off-line without knowning the observation policy
- Optimal observation policy is characterized by a convex set (the set where the optimal action is to take an observation).

Partially helps in computing optimal strategies

Outline

Belief state MDP

- Optimal estimation policy can be computed off-line without knowning the observation policy
- Optimal observation policy is characterized by a convex set
  (the set where the optimal action is to take an observation).

Partially helps in computing optimal strategies

Reachability analysis and a countable state MDP

- The set of reachable belief states is countable.
- The countable state MDP can be approximated by a finite state MDP

Outline

Belief state MDP

- Optimal estimation policy can be computed off-line without knowning the observation policy
- Optimal observation policy is characterized by a convex set
  (the set where the optimal action is to take an observation).

Partially helps in computing optimal strategies

Reachability analysis and a countable state MDP

- The set of reachable belief states is countable.
- The countable state MDP can be approximated by a finite state MDP

Approximate dynamic program

- Approximate value iteration
- Approximate policy iteration

Belief state MDP

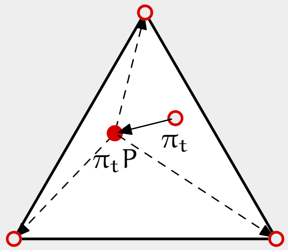
Belief state $\pi_t(x) = \mathbb{P}(X_t = x \mid Y_{0:t}, U_{0:t})$, for all $x \in \mathcal{X}$.

Belief state MDP

Belief state $\pi_t(x) = \mathbb{P}(X_t = x \mid Y_{0:t}, U_{0:t})$, for all $x \in \mathcal{X}$.

Evolution of the belief state

$$\pi_{t+1} = \begin{cases} \pi_t P, & \text{if } U_{t+1} = 0 \\ e_x, & \text{w.p. } [\pi_t P]_x \text{ if } U_{t+1} = 1 \end{cases}$$

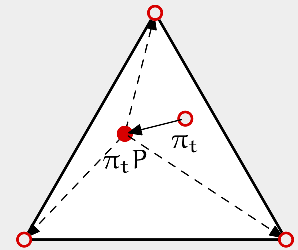


Belief state MDP

Belief state $\pi_t(x) = \mathbb{P}(X_t = x \mid Y_{0:t}, U_{0:t})$, for all $x \in \mathcal{X}$.

Evolution of the belief state

$$\pi_{t+1} = \begin{cases} \pi_t P, & \text{if } U_{t+1} = 0 \\ e_x, & \text{w.p. } [\pi_t P]_x \text{ if } U_{t+1} = 1 \end{cases}$$



Belief state is a sufficient statistic

There is no loss of optimality to restrict attention to strategies of the form

$$\hat{X}_t = g_t(\pi_t) \quad \text{and} \quad U_{t+1} = f_{t+1}(\pi_t).$$

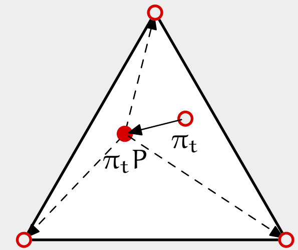
Belief state MDP

Belief state

$\pi_t(x) = \mathbb{P}(X_t = x \mid Y_{0:t}, U_{0:t})$, for all $x \in \mathcal{X}$.

Evolution of the belief state

$$\pi_{t+1} = \begin{cases} \pi_t P, & \text{if } U_{t+1} = 0 \\ e_x, & \text{w.p. } [\pi_t P]_x \text{ if } U_{t+1} = 1 \end{cases}$$



Belief state is a sufficient statistic

There is no loss of optimality to restrict attention to strategies of the form

$$\hat{X}_t = g_t(\pi_t) \quad \text{and} \quad U_{t+1} = f_{t+1}(\pi_t).$$

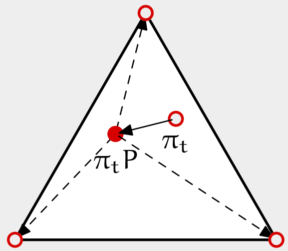
Structure of optimal estimation policy

Define $D(\pi, \hat{x}) = \sum_{x \in \mathcal{X}} \pi(x) d(x, \hat{x})$ and $D^*(\pi) = \min_{\hat{x} \in \mathcal{X}} D(\pi, \hat{x})$.

Then $g^*_t(\pi_t) = \arg\min_{\hat{x} \in \mathcal{X}} D(\pi_t, \hat{x})$.

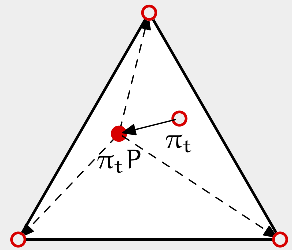
Belief state MDP: Dynamic program

$$V(\pi) = D^*(\pi) + \alpha \cdot \min \left\{ c + \sum_{x \in \mathcal{X}} [\pi P]_x \cdot V(e_x), \quad V(\pi P) \right\}$$



Belief state MDP: Dynamic program

$$V(\pi) = D^*(\pi) + \alpha \cdot \min \left\{ c + \sum_{x \in \mathcal{X}} [\pi P]_x \cdot V(e_x), \quad V(\pi P) \right\}$$



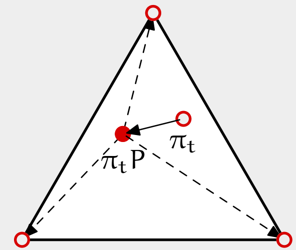
Theorem

1. The value function $V(\pi)$ is concave in π.

2. Let $\mathcal{S} = \{\pi : f^*(\pi) = 1\}$. Then $\mathcal{S}$ is convex.

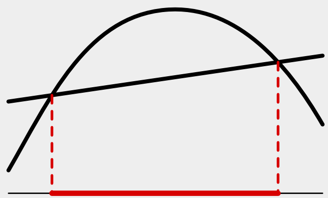
Belief state MDP: Dynamic program

$$V(\pi) = D^*(\pi) + \alpha \cdot \min\left\{ c + \sum_{x \in \mathcal{X}} [\pi P]_x \cdot V(e_x), \quad V(\pi P) \right\}$$



Theorem

1. The value function $V(\pi)$ is concave in π.
2. Let $\mathcal{S} = \{\pi : f^*(\pi) = 1\}$. Then $\mathcal{S}$ is convex.



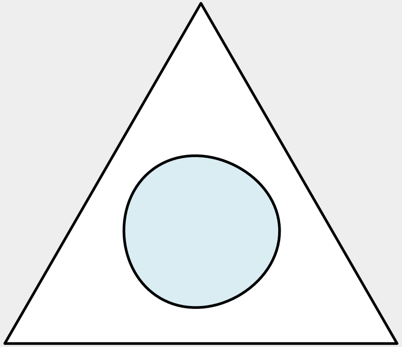
Proof outline

1. Is a standard result for POMDPs.
2. The first term $c + \sum_{x \in \mathcal{X}} [\pi P]_x V(e_x)$ is linear in π;
   The second term $V(\pi P)$ is concave in π.
   The set where a linear function lies below a concave function is convex.

Implication of the structural result

The optimal strategy is easy to implement

- Compute the optimal communication set $\mathcal{S}$ off line.
- The sensor simply needs to check if $\pi_t \in \mathcal{S}$.

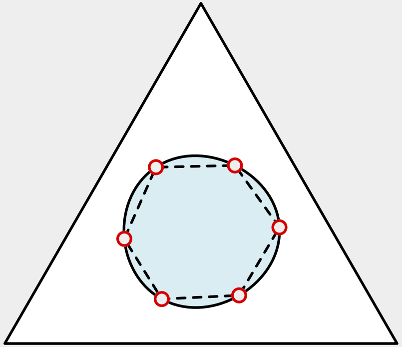


Implication of the structural result

The optimal strategy is easy to implement

- Compute the optimal communication set $\mathcal{S}$ off line.
- The sensor simply needs to check if $\pi_t \in \mathcal{S}$.

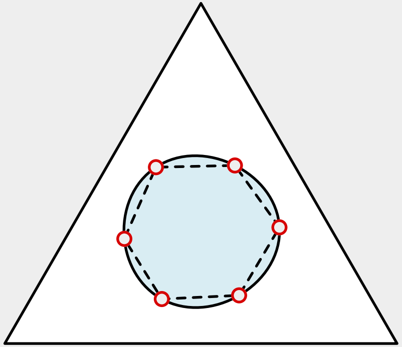
- If we approximate the convex set $\mathcal{S}$ by a polytope, then we may use LP to check if $\pi_t \in \mathcal{S}$.



Implication of the structural result

The optimal strategy is easy to implement

- Compute the optimal communication set $\mathcal{S}$ off line.
- The sensor simply needs to check if $\pi_t \in \mathcal{S}$.
- If we approximate the convex set $\mathcal{S}$ by a polytope, then we may use LP to check if $\pi_t \in \mathcal{S}$.



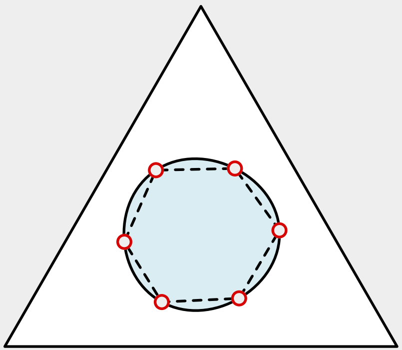
However, how to we compute $\mathcal{S}$ off line?

- Value iteration: The structural results do not help with value iteration algorithms. We still need to use a point-based method to find the optimal policy.

Implication of the structural result

The optimal strategy is easy to implement

- Compute the optimal communication set $\mathcal{S}$ off line.
- The sensor simply needs to check if $\pi_t \in \mathcal{S}$.
- If we approximate the convex set $\mathcal{S}$ by a polytope, then we may use LP to check if $\pi_t \in \mathcal{S}$.



However, how to we compute $\mathcal{S}$ off line?

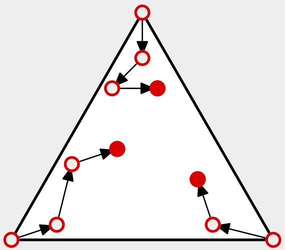
- Value iteration: The structural results do not help with value iteration algorithms. We still need to use a point-based method to find the optimal policy.
- Policy iteration: Note that the process $\{\pi_t\}$ is a Markov renewal process. Given a policy f, we can evaluate its performance in terms of the first passage cost and first passage time when this process hits the set $\mathcal{S}$. These can be approximated by simulation.

  If we approximate $\mathcal{S}$ by a polytope with k vertices, then we can find the best such representation using stochastic gradient descent.

An alternative characterization

Reachable set of belief state

Reachable set $\mathcal{R} = \{e_x P^n : x \in \mathcal{X} \text{ and } n \in \mathbb{N}\}$. This is a countable set.

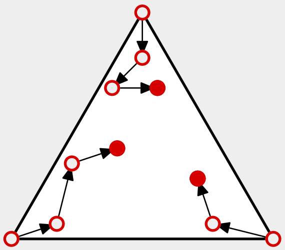


Reachable set of belief state

Reachable set $\mathcal{R} = \{e_x P^n : x \in \mathcal{X} \text{ and } n \in \mathbb{N}\}$. This is a countable set.

An equivalent information state $\pi_t \equiv (x, n)$, where

- x is the last measurement;
- n is the time since the last measurement.



Reachable set of belief state

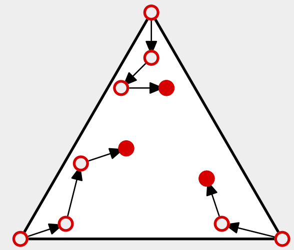
Reachable set

$\mathcal{R} = \{e_x P^n : x \in \mathcal{X} \text{ and } n \in \mathbb{N}\}$. This is a countable set.

An equivalent information state

$\pi_t \equiv (x, n)$, where

- x is the last measurement;
- n is the time since the last measurement.



Dynamic program

$$V(x,n) = D^*(e_x P^n) + \alpha \cdot \min\left\{ c + \sum_{y \in \mathcal{X}} P^{n+1}_{xy} \cdot V(y,1), \quad V(x,n+1) \right\}$$

Countable state MDP!

Finite dimensional approximate value iteration

Finite state approximation

Restricted policies

For any $m \in \mathbb{Z}_{>0}$, let $\mathcal{F}_m \coloneqq \{f \in \mathcal{F} : f(x, m) = 1\}$ denote the set of policies in which **the time between two measurements is always less than or equal to m**.

Finite state approximation

Restricted policies

For any $m \in \mathbb{Z}_{>0}$, let $\mathcal{F}_m \coloneqq \{f \in \mathcal{F} : f(x, m) = 1\}$ denote the set of policies in which the time between two measurements is always less than or equal to m.

Let $\hat{V}_m \colon \mathcal{X} \times \{1, \dots, m\} \to \mathbb{R}$ denote the corresponding value function, i.e.,

$$\hat{V}_m(x, n) = D^*(e_x P^n) + \alpha \cdot \min \left\{ c + \sum_{y \in \mathcal{X}} P_{xy}^{n+1} \cdot \hat{V}_m(y, 1), \quad \hat{W}_m(x, n+1) \right\}$$

$$\text{where } \hat{W}_m(x, n) = \begin{cases} \hat{V}_m(x, n), & \text{if } n \leqslant m \\ \infty, & \text{otherwise} \end{cases}$$

and let f_m^* denote the corresponding optimal policy.

Size of state space

$m \cdot |\mathcal{X}|$

Finite state approximation (cont.)

Theorem The restricted model constructed above is an approximating sequence for the original model (see [Sennott 1999]). Therefore,

1. $\lim_{m\to\infty} \hat{V}_m = V$
2. Any limit point of $\{f^*_m\}_{m=1}^{\infty}$ is optimal for the original model.

▷ Sennott, "Stochastic dynamic programming and the control of queueing systems," Wiley, 1999.

Finite state approximation (cont.)

Theorem

The restricted model constructed above is an approximating sequence for the original model (see [Sennott 1999]). Therefore,

1. $\lim_{m\to\infty} \hat{V}_m = V$
2. Any limit point of $\{f^*_m\}_{m=1}^{\infty}$ is optimal for the original model.

Approximation bound

Given a $x \in \mathcal{X}$ and $m \in \mathbb{Z}_{>0}$, let $\tau_m(x)$ denote the stopping time when the system leaves the set $\mathcal{X} \times \{1, \dots, m\}$. Then,

$$\left| V(x, 1) - \hat{V}_m(x, 1) \right| \leqslant \frac{2\,\mathbb{E}[\alpha^{\tau_m(x)}]}{1-\alpha} c \leqslant \frac{2\alpha^m}{1-\alpha} c$$

▷ Sennott, "Stochastic dynamic programming and the control of queueing systems," Wiley, 1999.

Finite state approximation (cont.)

Theorem

The restricted model constructed above is an approximating sequence for the original model (see [Sennott 1999]). Therefore,

1. $\lim_{m\to\infty} \hat{V}_m = V$
2. Any limit point of $\{f^*_m\}_{m=1}^{\infty}$ is optimal for the original model.

Approximation bound

Given a $x \in \mathcal{X}$ and $m \in \mathbb{Z}_{>0}$, let $\tau_m(x)$ denote the stopping time when the system leaves the set $\mathcal{X} \times \{1, \dots, m\}$. Then,

$$\left| V(x,1) - \hat{V}_m(x,1) \right| \leqslant \frac{2\,\mathbb{E}[\alpha^{\tau_m(x)}]}{1-\alpha} c \leqslant \frac{2\alpha^m}{1-\alpha} c$$

Practical method

Pick a large m. Check that $\max_{x\in\mathcal{X}} \min\{n : f^*_m(x,n) = 1\}$ is sufficiently smaller than m.

▷ Sennott, "Stochastic dynamic programming and the control of queueing systems," Wiley, 1999.

Finite dimensional approximate policy iteration

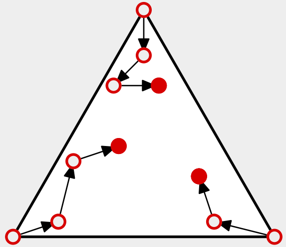
An efficient policy evaluation

Policy parameterization

Given a policy f and state $x \in \mathcal{X}$, define

$$k_x = \inf\{n : f(x, n) = 1\}$$

Then, under policy f, the states $\mathcal{R}_f \coloneqq \{(x, n) : x \in \mathcal{X}, n \leqslant k_x\}$ are ergodic. All other states are transient.



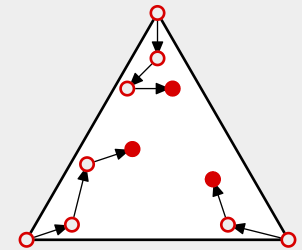
An efficient policy evaluation

Policy parameterization

Given a policy f and state $x \in \mathcal{X}$, define

$$k_x = \inf\{n : f(x, n) = 1\}$$

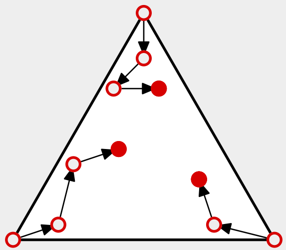
Then, under policy f, the states $\mathcal{R}_f \coloneqq \{(x, n) : x \in \mathcal{X}, n \leqslant k_x\}$ are ergodic. All other states are transient.



Policy evaluation

$$\begin{aligned}
\text{Note that } V(x,1) &= D(e_x P) &&+ \alpha V(x,2) \\
V(x,2) &= D(e_x P^2) &&+ \alpha V(x,3) \\
\vdots &= \quad \cdots &&+ \quad \cdots \\
V(x,k_x) &= D(e_x P^{k_x}) &&+ \alpha\Big[c + \sum_{y \in \mathcal{X}} P^{k_x}_{xy} V(y,1)\Big]
\end{aligned}$$

An efficient policy evaluation



Policy parameterization

Given a policy f and state $x \in \mathcal{X}$, define

$$k_x = \inf\{n : f(x, n) = 1\}$$

Then, under policy f, the states $\mathcal{R}_f \coloneqq \{(x, n) : x \in \mathcal{X}, n \leqslant k_x\}$ are ergodic. All other states are transient.

Policy evaluation

$$\begin{aligned} \text{Note that } V(x,1) &= D(e_x P) &&+ \alpha V(x,2) \\ V(x,2) &= D(e_x P^2) &&+ \alpha V(x,3) \\ \vdots \quad &= \quad \cdots &&+ \quad \cdots \\ V(x,k_x) &= D(e_x P^{k_x}) &&+ \alpha \Big[c + \sum_{y \in \mathcal{X}} P^{k_x}_{xy} V(y,1)\Big] \end{aligned}$$

Thus,

$$\boxed{V(x,1) = \sum_{n=1}^{k_x} \alpha^{n-1} D(e_x P^n) + \alpha^{k_x} \Big[c + \sum_{y \in \mathcal{X}} P^{k_x}_{xy} V(y,1)\Big]}$$

An efficient policy evaluation (cont.)

Notation For any $x \in \mathcal{X}$, define:

$$D^*_x(k) = \sum_{t=0}^{k-1} \alpha^t D^*(e_x P^t) \quad \text{and} \quad v_x = V(x, 1).$$

An efficient policy evaluation (cont.)

Notation For any $x \in \mathcal{X}$, define:

$$D^*_x(k) = \sum_{t=0}^{k-1} \alpha^t D^*(e_x P^t) \quad \text{and} \quad v_x = V(x, 1).$$

For a policy f parameterized by $(k_x)_{x \in \mathcal{X}}$. Define: $Q_{xy} = P^{k_x}_{xy}$

An efficient policy evaluation (cont.)

Notation

For any $x \in \mathcal{X}$, define:

$$D^*_x(k) = \sum_{t=0}^{k-1} \alpha^t D^*(e_x P^t) \quad \text{and} \quad v_x = V(x, 1).$$

For a policy f parameterized by $(k_x)_{x \in \mathcal{X}}$. Define: $Q_{xy} = P^{k_x}_{xy}$

Policy evaluation

$$\boxed{v_x = D^*_x(k_x) + \alpha^{k_x}\Big[c + \sum_{y \in \mathcal{X}} Q_{xy} v_y\Big]}$$

An efficient policy evaluation (cont.)

Notation

For any $x \in \mathcal{X}$, define:

$$D^*_x(k) = \sum_{t=0}^{k-1} \alpha^t D^*(e_x P^t) \quad \text{and} \quad v_x = V(x, 1).$$

For a policy f parameterized by $(k_x)_{x \in \mathcal{X}}$. Define: $Q_{xy} = P^{k_x}_{xy}$

Policy evaluation

$$\boxed{v_x = D^*_x(k_x) + \alpha^{k_x} \Big[c + \sum_{y \in \mathcal{X}} Q_{xy} v_y \Big]}$$

or, more compactly

$$\boxed{v_f = D^*_f + \alpha_f \odot [c + Qv] \implies v_f = [I - \alpha_f \odot Q]^{-1} [D^*_f + c\alpha_f]}$$

Effective state space: $\mathcal{X}$.

Approximate policy improvement

Policy improvement

Fix approximation level $m \in \mathbb{Z}_{>0}$.

Given the vector $(v_x)_{x \in \mathcal{X}}$, an improved policy parameterized by $(k_x)_{x \in \mathcal{X}}$ is:

$$\boxed{k_x = \arg \min_{k \in \{1, \dots, m\}} \Big\{ D^*_x(k) + \alpha^k \Big[c + \sum_{y \in \mathcal{X}} P^k_{xy} v_y \Big] \Big\}}$$

Approximate policy improvement

Policy improvement

Fix approximation level $m \in \mathbb{Z}_{>0}$.

Given the vector $(v_x)_{x \in \mathcal{X}}$, an improved policy parameterized by $(k_x)_{x \in \mathcal{X}}$ is:

$$\boxed{k_x = \arg \min_{k \in \{1, \dots, m\}} \Big\{ D^*_x(k) + \alpha^k \Big[c + \sum_{y \in \mathcal{X}} P^k_{xy} v_y \Big] \Big\}}$$

Very similar to Markov-Renewal Programming

▷ Jewell, "Markov-Renewal Programming. I: Formulation, Finite Return Models," Operations Research, 1963.

Approximate policy improvement

Policy improvement

Fix approximation level $m \in \mathbb{Z}_{>0}$.

Given the vector $(v_x)_{x \in \mathcal{X}}$, an improved policy parameterized by $(k_x)_{x \in \mathcal{X}}$ is:

$$\boxed{k_x = \arg\min_{k \in \{1,\dots,m\}} \Big\{ D^*_x(k) + \alpha^k \Big[c + \sum_{y \in \mathcal{X}} P^k_{xy} v_y \Big] \Big\}}$$

Very similar to Markov-Renewal Programming

> This is the optimal policy for the truncated model, so the previous approximation bound still holds

▷ Jewell, "Markov-Renewal Programming. I: Formulation, Finite Return Models," Operations Research, 1963.

Example

Dynamics $\mathcal{X} = \{\text{Low}, \text{Mid}, \text{High}\}$

$$P = \begin{bmatrix} 1-p & p & 0 \\ p & 1-p & p \\ 0 & p & 1-p \end{bmatrix}, \quad p = 0.2$$

Observation Cost $c = 0.75$

Distortion $d(x, \hat{x}) = |x - \hat{x}|$

Dynamics $\mathcal{X} = \{\text{Low}, \text{Mid}, \text{High}\}$

$$P = \begin{bmatrix} 1-p & p & 0 \\ p & 1-p & p \\ 0 & p & 1-p \end{bmatrix}, \quad p = 0.2$$

Observation Cost $c = 0.75$

Distortion $d(x, \hat{x}) = |x - \hat{x}|$

Optimal policy

$$k = \begin{bmatrix} 4 \\ 5 \\ 4 \end{bmatrix} \quad v = \begin{bmatrix} 5.69 \\ 6.22 \\ 5.69 \end{bmatrix}$$

Dynamics $\mathcal{X} = \{\text{Low}, \text{Mid}, \text{High}\}$

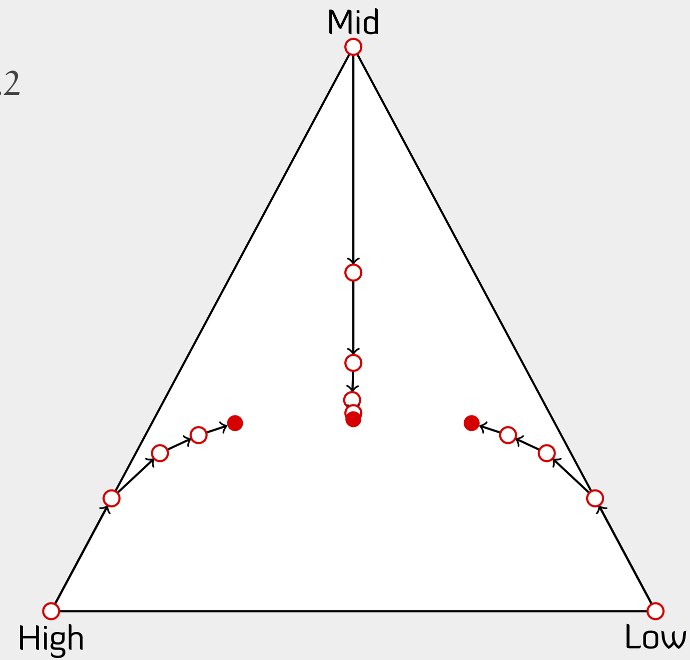
$$P = \begin{bmatrix} 1-p & p & 0 \\ p & 1-p & p \\ 0 & p & 1-p \end{bmatrix}, \quad p = 0.2$$

Observation Cost $c = 0.75$

Distortion $d(x, \hat{x}) = |x - \hat{x}|$

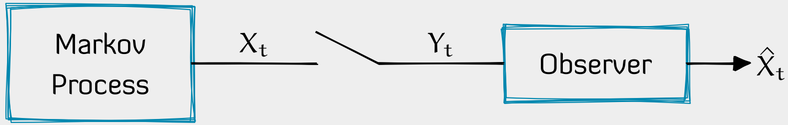
Optimal policy

$$k = \begin{bmatrix} 4 \\ 5 \\ 4 \end{bmatrix} \quad v = \begin{bmatrix} 5.69 \\ 6.22 \\ 5.69 \end{bmatrix}$$



Summary

Summary



Optimization problem

Choose:

- ▷ Observation policy $f = (f_0, f_1, \dots)$, where $U_t = f_t(Y_{0:t-1}, U_{0:t-1})$
- ▷ Estimation policy $g = (g_0, g_1, \dots)$, where $\hat{X}_t = g_t(Y_{0:t}, U_{0:t})$

to minimize

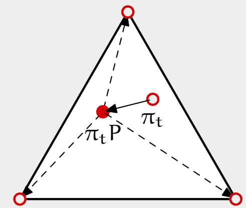
$$\mathbb{E}\left[\sum_{t=0}^{\infty} \alpha^t \left[cU_t + d(X_t, \hat{X}_t)\right]\right]$$

When to observe a Markov process–(Mahajan)

Summary

Belief state MDP: Dynamic program

$$V(\pi) = D^*(\pi) + \alpha \cdot \min \left\{ c + \sum_{x \in \mathcal{X}} [\pi P]_x \cdot V(e_x), \quad V(\pi P) \right\}$$



Theorem

1. The value function $V(\pi)$ is concave in π.
2. Let $\mathcal{S} = \{\pi : f^*(\pi) = 1\}$. Then $\mathcal{S}$ is convex.

Summary

Approximate value iteration

Fix approximation level m. Size of state space: $m|\mathcal{X}|$.

$$\hat{V}_m(x, n) = D^*(e_x P^n) + \alpha \cdot \min \left\{ c + \sum_{y \in \mathcal{X}} P^{n+1}_{xy} \cdot \hat{V}_m(y, 1), \quad \hat{W}_m(x, n+1) \right\}$$

where $\hat{W}_m(x, n) = \begin{cases} \hat{V}_m(x, n), & \text{if } n \leqslant m \\ \infty, & \text{otherwise} \end{cases}$

When to observe a Markov process–(Mahajan)

Summary

Approximate policy iteration

Fix approximation level m. Size of state space: $|\mathcal{X}|$.

Policy evaluation

$$\boxed{v_f = [I - \alpha_f \odot Q]^{-1}[D_f^* + c\alpha_f]}$$

Policy Improvement

$$\boxed{k_x = \arg\min_{k \in \{1,\dots,m\}} \Big\{ D_x^*(k) + \alpha^k \Big[c + \sum_{y \in \mathcal{X}} P_{xy}^k v_y \Big] \Big\}}$$

Concluding Remarks

Key Ideas

- When a POMDP has a **no-or-perfect observation property**, the reachable set has a nice structure.
- Using this structure, the belief space MDP can be transfored into a countable state MDP.
- In practice, under the optimal policy, only a finite set of states is reached. Therefore, finite state approximations work well.

Concluding Remarks

Key Ideas

- When a POMDP has a **no-or-perfect observation property**, the reachable set has a nice structure.
- Using this structure, the belief space MDP can be transfored into a countable state MDP.
- In practice, under the optimal policy, only a finite set of states is reached. Therefore, finite state approximations work well.

Other applications

- Machine maintenance
- Scheduling communication over time-varying networks
- . . .

Concluding Remarks

Key Ideas

- When a POMDP has a **no-or-perfect observation property**, the reachable set has a nice structure.
- Using this structure, the belief space MDP can be transfored into a countable state MDP.
- In practice, under the optimal policy, only a finite set of states is reached. Therefore, finite state approximations work well.

Other applications

- Machine maintenance
- Scheduling communication over time-varying networks
- . . .

Reinforcement Learning

- In general, it is difficult to come up with reinforcement learning algorithms for POMDPs (because the belief state depends on the model parameters).
- The countable state representation does not depend on the model parameters.
- One can run standard RL algorithms for finite state MDPs and use them for the finite state approximations presented here.