Learning MDP with second-hand samples

Siddharthan Rajasekaran, Jie Fu
sperundurairajas@wpi.edu, jfu2@wpi.edu

1 Summary of Discussions

In this report, we are interested in reducing the sample complexity of PAC-learning in MDP using similarity between different states. Our approach is to integrate Expectation-Maximization with PAC-learning.

2 Problem statement

We consider an MDP (S, A, P, r) in which the transition probability function P is unknown. Moreover, the state space can be partitioned into different clusters $\bigcup_{i=1}^{m} \Pi_i = S$. The system dynamics in each cluster is homogeneuous. The class of each state is only observed through uncertain sensors. For example, in Mars rover exploration task, the system can only tell from images a probabilistic distribution of the types of the ground: At state s, the probability that the current state is in type "sand" is 0.7 and in type "gravel" is 0.2. This information provided by the sensor is considered *side information* for the system.

Problem: Given the side information, develop an algorithm that learns the underlying transitio function P from data collected through exploring the state space.

Notation: Let the set of classes be $\mathcal{C} = \{1, \ldots, m\}$. Given a state $s \in S$, if it is in class j, the transition function $P(s, a) = \Psi_j(s, a)$ where $\Psi_j : S \times A \times S \to [0, 1]$ is the transition function given that all states in the MDP is in class j.

Data set: The data set is a sequence of tuples $D = \{(s_i, a_i, s_{i+1}, p_i), i = 1, \ldots, N\}$ where (s_i, a_i, s_{i+1}) is an observed transition at step i and $p_i : \mathcal{C} \to [0, 1]$ is a probability distribution over the classes. $p_i(j)$ is the probability that the state s_i belongs to class j.

Assumption 1 *Using conditional probability, for any $s \in S$, we have*

$$P(s' \mid s, a) = \sum_{j=1}^{m} P(s' \mid s, a, c_j) p(c_j \mid s) = \sum_{j=1}^{m} \Psi_j(s' \mid s, a) p(c_j \mid s).$$

Problem 1: Assuming 1 Given the data D, compute the transition functions $\{\Psi_j \mid j = 1, \ldots, m\}$ as well as the bounds of error confidence intervals.

Problem 2: In the second case, we consider that the state space of the MDP is not clearly partitioned into m different classes. Rather, a state with 0.5 "sand" and 0.5 "gravel" could be a mixture of ground condition that is neither sand or gravel, but some type in between. Now, the conditional probability cannot be used directly and the relation between $P(s' \mid s, a)$ and $\Psi_j(s' \mid s, a)$ may no longer be linear.

2.1 When the assumption holds

We consider systems which evolve at discrete instances of time. The time index can be $1, 2, \ldots, N$ where N can be finite. Let $\mathcal{T}$ be an index set, which can be finite or countably infinite. Assume also that there is an underlying probability

space (Ω, S, P) with respect to which all random variables are defined. A family of random variables w_k, $k \in \mathcal{T}$ is a discrete time stochastic process which is the noise to the system. The stochastic system

$$x_{k+1} = f(x_k, u_k, w_k), \quad k = 0, 1, \ldots.$$

where $x_k \in X$ and $u_k \in U$ are state and input at time step k.

Now, consider the case with different classes, it is similar to the case that the system dynamics has a hidden parameter variable θ_k and evolve according to

$$x_{k+1} = f(x_k, u_k, \theta_k, w_k), \quad k = 0, 1, \ldots,$$

where θ_k is the parameter when the system and takes discrete values in $\{\theta_j, j = 1, \ldots, m\}$. As an example of θ_k, in Mars rover case, θ_k can be the friction coefficient with respect to the ground. The domain Θ of friction parameter can be continuous. In this case, the different classes may be considered as a finite subset of the parameter space Θ. Particular, if we assume Θ is a probability simplex, then θ_j can be picked to be affinely independent points that determine the simplex Θ.

Consider any $\theta = \sum_j v_j \theta_j$ with $\sum_j v_j = 1$, for any state action pair (x, a) and a noise w, the next state is

$$x' = f(x, a, \theta, w) = f(x, a, \sum_j v_j \theta_j, w)$$

Claim 1 *In the case we can represent the vector field* $f(x, u, \theta, w) = g_1(x, u)^T \theta + g_2(x, u, w)$*, then*

$$x' = f(x, a, \theta, w) = \sum_j v_j f(x, a, \theta_j, w)$$

where $\theta = \sum_j v_j \theta_j$ *holds.*

Proof:

$$\begin{aligned} x_{k+1} =& g_1(x, u)^T \theta + g_2(x, u, w) \\ =& \sum_j v_j g_1(x, u)^T \theta_j + g_2(x, u, w) \\ =& \sum_j v_j g_1(x, u)^T \theta_j + \sum_j v_j g_2(x, u, w) \\ =& \sum_j v_j (g_1(x, u)^T \theta_j + g_2(x, u, w)) = \sum_j v_j f(x, a, \theta_j, w) \end{aligned}$$

since $\sum_j v_j = 1$.

It is as if we allow the system evolves independently in m different systems parameterized by different θ_j and the combine the output in these systems given the weight v_j.

3 Problem 1: Learning with side information

In the following sections, we will see two different ways of finding the transition probability for each terrain given a trajectory sampled from the real system with

time varying probability distribution over the terrains. In both the methods we will aggregate homeomorphic states and actions. For example, facing North and taking the action "East" is same as facing East and taking the action "South". Hence the state action pairs (North, East) = (East, South).

3.1 EM with PAC

In this section, we will first see why, given a sequence of events sampled from a probability distribution over several events, the probability of a particular event is given by $P(event) = \frac{\#event}{\#all\ events}$, where # denotes the total number of times an event occurred in the sample. This informs us about how the problem of interest can be formulated similarly.

Consider the case of learning $T(s'|s,a)$ in an MDP with just one terrain given a trajectory $\tau = [s_1, a_1, s_2, a_2, ..., s_T, a_T]$. Let the current estimate of $T(s'|s,a)$ be $\Psi(s'|s,a)$

The probability of sampling the above trajectory given Ψ and the control actions $\mathbf{a} = [a_1, a_2, ..., a_T]$ is given by

$$P(\tau|\Psi||a) = \prod_t \Psi(s_{t+1}|s_t, a_t) \tag{1}$$

where $||$ indicates conditioning over variables we do not have control over for our problem of estimating Ψ. The log likelihood of the trajectory is given by $\mathcal{L} = \ln\Big[\prod_t \Psi(s_{t+1}|s_t,a_t)\Big]$. The maximum likelihood solution over Ψ for the problem can be formulated as,

$$\Psi^* = \arg\max_{\Psi} \sum_t \ln \Psi(s_{t+1}|s_t,a_t) \tag{2}$$

$$\text{s.t.} \sum_{s'} \Psi(s'|s,a) = 1, \forall s, a \tag{3}$$

The Lagrangian is given by,

$$L = \sum_t \ln \Psi(s_{t+1}|s_t,a_t) + \sum_{s,a} \lambda_{s,a}\Big[\sum_{s'} \Psi(s'|s,a) - 1\Big] \tag{4}$$

$$\nabla_{\Psi(s'|s_{t_0},a_{t_0})} L = \sum_{t:s_{t+1}=s', s_t=s_{t_0}, a_t=a_{t_0}} \frac{1}{\Psi(s_{t+1}|s_t,a_t)} + \lambda_{s_{t_0},a_{t_0}} = 0 \tag{5}$$

$$(or) \quad \Psi(s_{t+1}|s_t,a_t) \propto \sum_{t:s_{t+1}=s', s_t=s_{t_0}, a_t=a_{t_0}} 1 \tag{6}$$

$$\implies \quad \Psi(s'|s_{t_0},a_{t_0}) = \frac{\#(s_{t_0},a_{t_0},s')}{\#(s_{t_0},a_{t_0})} \tag{7}$$

The above formula Eq. 7 is the result of maximum likelihood estimate of the underlying probability distribution Ψ. In our problem, we also have another variable namely the type of environment. This can be treated as a hidden class variable in case of clustering and we can use Expectation (over the type environment) - Maximization (similar to the maximization in Eq. 4). Since we know the distribution over class, $P(c|t)$, at time t from the observations, we

marginalize out time to get $P(c_j|s_t, a_t, s_{t+1})$. The maximization problem can be written as,

$$\Psi^* = \arg\max_{\Psi_{c_1}...\Psi_{c_m}} \sum_t \ln P(s_{t+1}|s_t, a_t, \Psi_{c_1}...\Psi_{c_m}) \tag{8}$$

$$= \arg\max_{\Psi_{c_1}...\Psi_{c_m}} \sum_{t=1}^{T}\sum_{j=1}^{m} P(c_j|s_t, a_t, s_{t+1}, \Psi_{c_1}...\Psi_{c_m}) \ln P(s_{t+1}, c_j|s_t, a_t, \Psi_{c_1}...\Psi_{c_m}) \tag{9}$$

$$= \arg\max_{\Psi_{c_1}...\Psi_{c_m}} \sum_{t=1}^{T}\sum_{j=1}^{m} \Big[\sum_{t'} \big[P(c_j|t')P(t'|s_t, a_t, s_{t+1})\big] \ln P(s_{t+1}, c_j|s_t, a_t, \Psi_{c_1}...\Psi_{c_m})\Big] \tag{10}$$

$$= \arg\max_{\Psi_{c_1}...\Psi_{c_m}} \sum_{t=1}^{T}\sum_{j=1}^{m} \Big[\underbrace{\sum_{t':(s_{t'},a_{t'},s_{t'+1})=(s_t,a_t,s_{t+1})} \left[\frac{P(c_j|t')}{\#(s_t, a_t, s_{t+1})}\right]}_{\beta_{tj}} \ln P(s_{t+1}, c_j|s_t, a_t, \Psi_{c_1}...\Psi_{c_m})\Big] \tag{11}$$

where we get Eq. 9 from EM. We know the value of $P(c_j|t)$ from the sensor, hence we replace the inner sum by β_{tj} which denotes the probability of class c_j given the tuple (s_t, a_t, s_{t+1}). That is,

$$\Psi^* = \arg\max_{\Psi} \sum_{t=1}^{T}\sum_{j=1}^{m} \beta_{tj} \ln P(s_{t+1}, c_j|s_t, a_t, \Psi_{c_1}...\Psi_{c_m}) \tag{12}$$

$$= \arg\max_{\Psi} \sum_{t=1}^{T}\sum_{j=1}^{m} \beta_{tj} \ln \big[P(s_{t+1}|c_j, s_t, a_t, \Psi_{c_1}...\Psi_{c_m})P(c_j|s_t, a_t)\big] \tag{13}$$

$$= \arg\max_{\Psi} \sum_{t=1}^{T}\sum_{j=1}^{m} \beta_{tj} \ln \Psi_{c_j}(s_{t+1}|s_t, a_t) + \underbrace{\ln P(c_j|s_t, a_t)}_{\text{const.}} \tag{14}$$

$$\Psi^* = \sum_{j=1}^{m} \arg\max_{\Psi_{c_j}} \sum_{t=1}^{T} \beta_{tj} \ln \Psi_{c_j}(s_{t+1}|s_t, a_t) \tag{15}$$

$$\Psi^*_{c_j} = \arg\max_{\Psi_{c_j}} \sum_{t=1}^{T} \beta_{tj} \ln \Psi_{c_j}(s_{t+1}|s_t, a_t) \tag{16}$$

$$\text{s.t.} \sum_{s'} \Psi_{c_j}(s'|s, a) = 1, \forall s, a \tag{17}$$

We get Eq. 14 from the fact that the class variable c_j just picks the corresponding probability distribution Ψ_{c_j}. We get Eq. 16 since each term in the summation is independent of other terms in the maximization problem. The objective in Eq. 16 is similar to Eq. 2 except that each term is multiplied by β_{tj}. Hence, by following the same steps we get (similar to Eq. 6),

$$\Psi_{c_j}(s'|s_{t_0}, a_{t_0}) \propto \sum_{t:(s_t,a_t,s_{t+1})=(s_{t_0},a_{t_0},s')} \beta_{tj} \tag{18}$$

$$\propto \beta_{tj} \sum_{t:(s_t,a_t,s_{t+1})=(s_{t_0},a_{t_0},s')} 1 \tag{19}$$

$$\propto \frac{\sum_{t':(s_{t'},a_{t'},s_{t'+1})=(s_{t_0},a_{t_0},s')} P(c_j|t')}{\cancel{\#(s_t, a_t, s_{t+1})}} \cdot \cancel{\#(s_t, a_t, s_{t+1})} \tag{20}$$

$$\Psi_{c_j}(s'|s_{t_0}, a_{t_0}) = \frac{\sum_{t':(s_{t'},a_{t'},s_{t'+1})=(s_{t_0},a_{t_0},s')} P(c_j|t')}{\sum_{t':(s_{t'},a_{t'})=(s_{t_0},a_{t_0})} P(c_j|t')} \tag{21}$$

$$= \frac{\text{effective}\#(s_{t_0}, a_{t_0}, s')}{\text{effective}\#(s_{t_0}, a_{t_0})} \tag{22}$$

3.2 Marginalization with PAC

Consider a sample trajectory τ generated with an arbitrary distribution over classes. The maximum likelihood probability of observing a tuple (s_{t_0}, a_{t_0}, s') under the time varying class distribution is given by

$$P(s'|s_{t_0}, a_{t_0}) = \frac{\#(s_{t_0}, a_{t_0}, s')}{\#(s_{t_0}, a_{t_0})}$$

We can rewrite $P(s'|s_{t_0}, a_{t_0})$ by marginalizing the class variable c_j as

$$\begin{aligned}
P(s'|s_{t_0}, a_{t_0}) &= \sum_{j=0}^{m} P(s', c_j|s_{t_0}, a_{t_0}) \\
&= \sum_{j=0}^{m} P(s'|c_j, s_{t_0}, a_{t_0}) P(c_j|s_{t_0}, a_{t_0}) \\
&= \sum_{j=0}^{m} P(s'|c_j, s_{t_0}, a_{t_0}) \sum_{t} P(c_j|t) P(t|s_{t_0}, a_{t_0}) \\
&= \sum_{j=0}^{m} P(s'|c_j, s_{t_0}, a_{t_0}) \sum_{t:(s_t,a_t)=(s_{t_0},a_{t_0})} \frac{P(c_j|t)}{\#(s_{t_0}, a_{t_0})} \\
&= \sum_{j=0}^{m} \Psi^*_{c_j}(s'|s_{t_0}, a_{t_0}) \sum_{t:(s_t,a_t)=(s_{t_0},a_{t_0})} \frac{P(c_j|t)}{\#(s_{t_0}, a_{t_0})} \\
\text{Let } \alpha_{tj} &= \sum_{t':(s_{t'},a_{t'})=(s_t,a_t)} \frac{P(c_j|t')}{\#(s_t, a_t)} \\
\therefore \frac{\#(s_{t_0}, a_{t_0}, s')}{\#(s_{t_0}, a_{t_0})} &= \sum_{j=0}^{m} \Psi^*_{c_j}(s'|s_{t_0}, a_{t_0}) \alpha_{tj} \\
&= [\alpha_{t1}, \alpha_{t2} \dots \alpha_{tm}] \begin{bmatrix} \Psi^*_{c_1} \\ \Psi^*_{c_2} \\ \vdots \\ \Psi^*_{c_m} \end{bmatrix} \\
&= \boldsymbol{\alpha}_t^T \begin{bmatrix} \Psi^*_{c_1} \\ \Psi^*_{c_2} \\ \vdots \\ \Psi^*_{c_m} \end{bmatrix}
\end{aligned}$$

The above equation has m unknowns $(\Psi^*_{c_j}, \forall j)$. We can however, solve the above equation by segmenting the existing trajectory into m pieces. We will denote the piece index in the super script like $\boldsymbol{\alpha}_t^{T(j)}$.

$$\text{Let } A = \begin{bmatrix} \boldsymbol{\alpha}_t^{T(1)} \\ \boldsymbol{\alpha}_t^{T(2)} \\ \vdots \\ \boldsymbol{\alpha}_t^{T(m)} \end{bmatrix}$$

$$\begin{bmatrix} P^{(1)} \\ P^{(2)} \\ \vdots \\ P^{(m)} \end{bmatrix} = A \begin{bmatrix} \Psi^*_{c_1} \\ \Psi^*_{c_2} \\ \vdots \\ \Psi^*_{c_m} \end{bmatrix}$$

$$\therefore \begin{bmatrix} \Psi^*_{c_1} \\ \Psi^*_{c_2} \\ \vdots \\ \Psi^*_{c_m} \end{bmatrix} = A^{-1} \begin{bmatrix} P^{(1)} \\ P^{(2)} \\ \vdots \\ P^{(m)} \end{bmatrix}$$

where $P^{(j)}$ is $P^{(j)}(s'|s_{t_0}, a_{t_0})$ and Ψ_{c_j} is $\Psi_{c_j}(s'|s_{t_0}, a_{t_0})$. We solve the above system of equations for all possible combinations of (s_{t_0}, a_{t_0}, s') that occur in the trajectory τ.

Note that we need to be exposed to atleast m different distributions over the terrains for the matrix A to be non singular. The same is actually necessary even in case of the EM formulation to meaningfully estimate the underlying distributions. In case we have a trajectory with only one distribution over the terrain for all t, the EM sets $\Psi^*_{c_1} = \Psi^*_{c_2} ... = \Psi^*_{c_m}$ (which is in fact the maximum likelihood solution).

3.3 Results

We tested the marginalization based method on a toy example with four states and two classes. Each state has two actions 0 and 1. In the first class c_1, given the current state s_t, the action 0 results in a uniform distribution over the next possible states s_{t+1} while that of action 1 results in a Gaussian centered at the current state. The second class just reverses the effects of actions 0 and 1. That is, action 1 results in a uniform distribution and, action 0, in a Gaussian.

We generate a trajectory using the above conditions and use it to learn the underlying transition matrix of each class. We assume that the probability of class c_1 changes from 0.05 to 0.95 linearly along the trajectory and that of c_2 is just $1 - P(c_1)$. We split the trajectory into two halves since the first and the second has a different average class distributions.

The following graph shows how the norm of the error matrix (difference of learned and the underlying transition matrices) falls with increasing length of the trajectory.

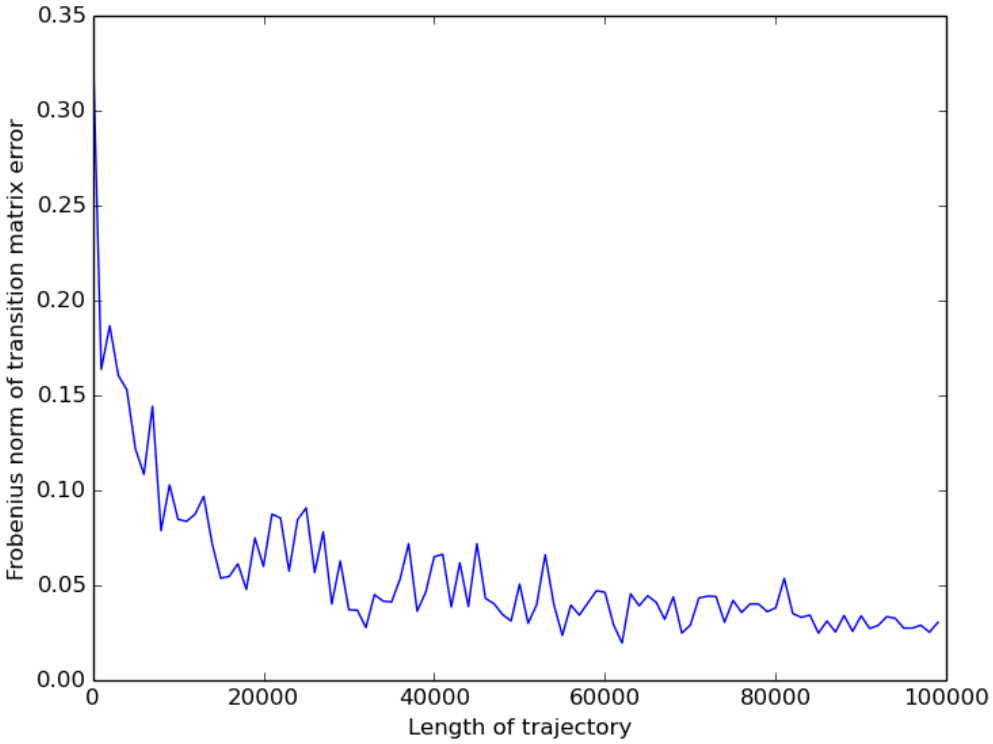

Figure 1: Norm of error between the learned transition matrix and the ground truth vs the length of the trajectory.