Notes on Markov Models for 16.410 and 16.413

1 Markov Chains

A Markov chain is described by the following:

- a set of states $\mathcal{S} = \{s_1, s_2, \ldots s_n\}$
- a set of transition probabilities $T(s_i, s_j) = p(s_j|s_i)$
- an initial state $s_0 \in \mathcal{S}$

The Markov Assumption

The state at time t, s_t, depends only on the previous state s_{t-1} and not the previous history. That is,

$$p(s_t|s_{t-1}, s_{t-2}, s_{t-3}, s_0) = p(s_t|s_{t-1}) \tag{1}$$

Things you might want to know about Markov chains:

- Probability of being in state s_i at time t
- Stationary distribution

2 Markov Decision Processes

The extension of Markov chains to decision making

A Markov decision process (MDP) is a model for deciding how to act in "an accessible, stochastic environment with a known transition model" (Russell & Norvig, pg 500.). A Markov decision process is described by the following:

- a set of states $\mathcal{S} = \{s_1, s_2, \ldots s_n\}$
- a set of actions $\mathcal{A} = \{a_1, a_2, \ldots, a_m\}$
- a set of transition probabilities $T(s_i, a, s_j) = p(s_j|s_i, a)$
- a set of rewards $R : \mathcal{S} \times \mathcal{A} \mapsto \Re$
- a discount factor $\gamma \in [0, 1]$
- an initial state $s_0 \in \mathcal{S}$

Things you might want to know about MDPs:

- The optimal policy

One way to compute the optimal policy:

Define the optimal value function $V(s_i)$ by the Bellman equation:

$$V(s_i) = \max_a \left(R(s_i, a) + \gamma \sum_{j=1}^{|\mathcal{S}|} p(s_j|s_i, a) \cdot V(s_j) \right) \tag{2}$$

The value iteration algorithm using Bellman's equation:

1. t = 0
2. $\forall s_i \in \mathcal{S} : V^0(s_i) = 0$
3. do
4. change = 0
5. $\forall s_i \in \mathcal{S}$:
6. $V^t(s_i) = \max_a \left(R(s_i, a) + \gamma \sum_{j=1}^{|\mathcal{S}|} p(s_j|s_i, a) \cdot V^{t-1}(s_j) \right)$
7. change = change + $V^t(s_i) - V^{t-1}(s_i)$
8. while not converged

3 Hidden Markov Models

The extension of Markov chains to partially observable worlds

A Hidden Markov Model is described by the following:

- a set of states $\mathcal{S} = \{s_1, s_2, \ldots s_n\}$
- a set of observations $\mathcal{Z} = \{z_1, z_2, \ldots z_n\}$
- a set of transition probabilities $T(s_i, s_j) = p(s_j|s_i)$
- a set of emission probabilities $O(z_i, s_j) = p(z_i|s_j)$
- an initial state distribution $p_0(s)$

We never know the true state of the system. Even at the start, we are given only an initial distribution over states, $p_0(s)$. At each point in time, we get some observation z. We can infer the posterior distribution

$$\begin{aligned} p(s_t|z_t, z_{t-1}, z_{t-2}, z_{t-3}, \ldots, z_0) &= \alpha p(z_t|s_t) p(s_t|z_{t-1}, z_{t-2}, z_{t-3}, \ldots, z_0) & (3) \\ &= \alpha p(z_t|s_t) \sum_{s_{t-1}} p(s_t|s_{t-1}) p(s_{t-1}|z_{t-1}, z_{t-2}, z_{t-3}, \ldots, z_0) & (4) \end{aligned}$$

Equation 4 is known as the "Bayes' filter", and can be computed recursively.

This is exactly how the Kalman filter works.

Things you might want to know about HMMs (Rabiner's famous 3 questions):

- Given the observation sequence $Z = z_1 z_2 \ldots z_t$, and a model $\lambda = (T, O, p_0)$, how do we efficiently compute $p(Z|\lambda)$, the probability of the observation sequence given the model?
- Given the observation sequence $Z = z_1 z_2 \ldots z_t$, and a model $\lambda = (T, O, p_0)$, how do we choose a corresponding state sequence $Q = s_1, s_2, \ldots, s_t$ which is optimal in some meaninful sense (i.e., best "explains" the observations)?
- How do we adjust the model parameters $\lambda = (T, O, p_0)$ to maximize $p(Z|\lambda)$?

Problem 1 - Forward Algorithm

Probability of sequence Z given λ is the probability of Z over all state sequences Q

$$
\begin{aligned}
p(Z|\lambda) &= \sum_Q p(Z|Q,\lambda)p(Q|\lambda) && (5)\\
&= \sum_{s_1,s_2,s_3,\ldots} p_0(s_1)p(z_1|s_1)p(s_2|s_1)p(z_2|s_2)p(s_3|s_2)\ldots && (6)
\end{aligned}
$$

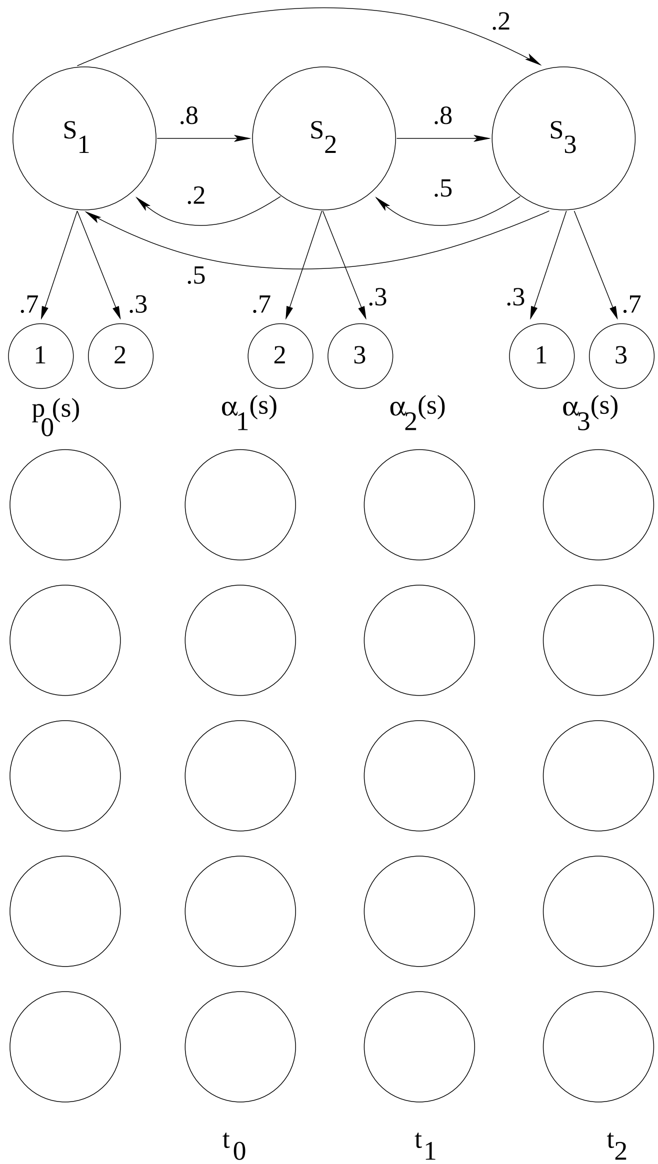
- Problem: Summing over all state sequences is $2t|S|^t$.
- Instead, build lattice of states forward in time, computing probabilities of each possible trajectory as lattice is built
- Forward algorithm is $|S|^2 t$

Algorithm:

1. Initialize: $\alpha_1(s_i) = p_0(s_i)p(z_1|s_i)$
2. Induction: Repeat for $\tau = 1 : t$

$$\alpha_{\tau+1}(s_i) = \left[\sum_{j=1}^{|S|} \alpha_\tau(s_j)p(s_i|s_j)\right] p(z_{\tau+1}|s_i) \tag{7}$$

3. Termination: $p(Z|\lambda) = \sum_{j=1}^{|S|} \alpha_t(s_j)$



.2
s_1
.8
s_2
.8
s_3
.2
.5
.5
.7
.3
.7
.3
.3
.7
1
2
2
3
1
3
$p_0(s)$
$\alpha_1(s)$
$\alpha_2(s)$
$\alpha_3(s)$
t_0
t_1
t_2

Problem 2 - Viterbi Decoding

Same principle as forward algorithm, with an extra term

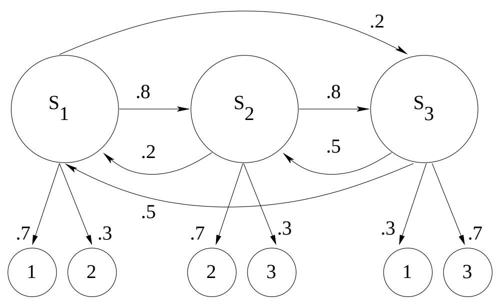
Algorithm:

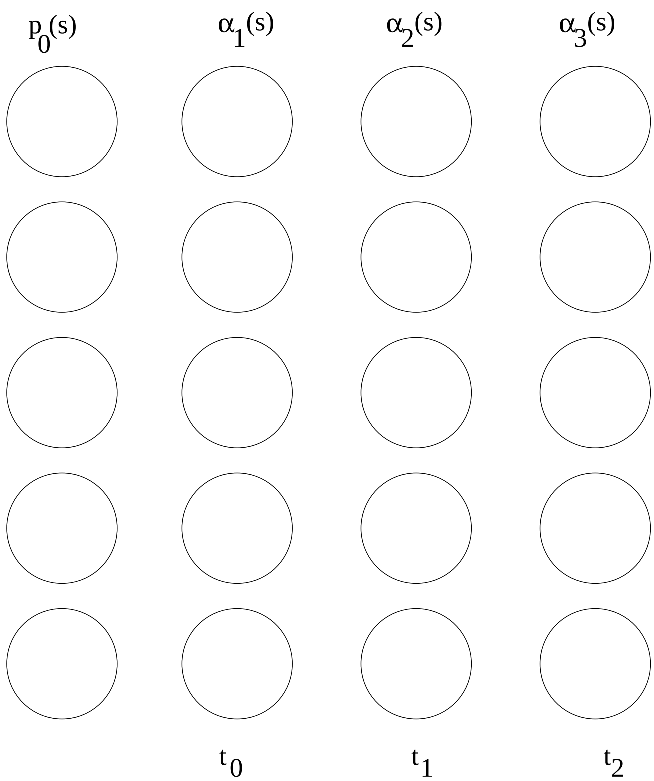
1. Initialize:
   $\alpha_1(s_i) = p_0(s_i)p(z_1|s_i)\ \psi_1(s_i) = 0$
2. Induction: Repeat for $\tau = 1 : t$

$$\alpha_{\tau+1}(s_i) = \left[\max_{s_j} \alpha_\tau(s_j)p(s_i|s_j)\right] p(z_{\tau+1}|s_i) \tag{8}$$

$$\psi_{\tau+1}(s_i) = \left[\max_{s_j} \alpha_\tau(s_j)p(s_i|s_j)\right] \tag{9}$$

3. Termination: $p(Z|\lambda) = \left[\max_{s_j} \alpha_\tau(s_j)p(s_i|s_j)\right]$
   $s^*_\tau = \psi_{\tau+1}(s^*_{\tau+1})$





Problem 3 - Improving the Model Parameters

This is a topic for another class.

4 Partially Observable Markov Decision Processes

The extension of HMMs to decision making

A POMDP is described by the following:

- a set of states $\mathcal{S} = \{s_1, s_2, \ldots s_n\}$
- a set of actions $\mathcal{A} = \{a_1, a_2, \ldots, a_m\}$
- a set of transition probabilities $T(s_i, a, s_j) = p(s_j|s_i, a)$
- a set of observations $\mathcal{Z} = \{z_1, z_2, \ldots, z_l\}$
- a set of observation probabilities $O(s_i, a_j, z_k) = p(z \mid s, a)$
- an initial distribution over states, $p_0(s)$
- a set of rewards $R : \mathcal{S} \times \mathcal{A} \times \mathcal{Z} \mapsto \Re$

We won't talk in this class about how to solve these.

5 Things you should know about (hidden) Markov (decision) processes

- Each kind of Markov model
- How to frame a problem as a (possible hidden) Markov process
- Value iteration
- The Bayes' Filter equation
- The forward algorithm
- The Viterbi algorithm