Dynamic Programming Lecture #5

Outline:

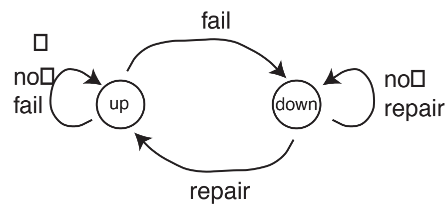
- Finite state Markov chains
- Viterbi Algorithm

Finite State Markov Chains

- Restrict attention to finite state systems:

$$x_k \in X = \{1, 2, \ldots, N\}$$

- Simple example: Machine up/down dynamics:



- Define $p_{ij} =$ probability to jump from state i to state j
- Back to example: Set $x = \begin{cases} 1 & \text{up} \\ 2 & \text{down} \end{cases}.$

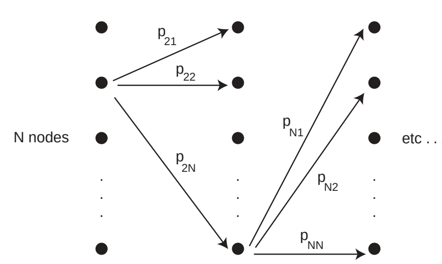
$$p_{12} = \text{probability of failure given up}$$

$$1 - p_{12} = \text{probability of no failure given up}$$

$$p_{21} = \text{probability of repair given down}$$

$$p_{22} = \text{probability of no repair given down}$$

- Staged evolution: Random jumps



Stochastic Matrices

- Define "stochastic matrix":

$$P = [p_{ij}] = \begin{pmatrix} p_{11} & p_{12} \\ p_{21} & p_{22} \end{pmatrix}$$

Note: Each row sums to 1.

- Q: What is $Pr(x_k = j | x_0 = i)$?
- A: $(\,P^k\,)_{ij}$, i.e.,

$$(\underbrace{P \cdot P \ldots P}_{k \text{ times}})_{ij} \text{ element}$$

- Proof: Index over immediate predecessor (total probability):

$$\begin{aligned} Pr(x_2 = j | x_0 = i) &= \sum_k Pr(x_2 = j | x_0 = i \;\&\; x_1 = k) Pr(x_1 = k | x_0 = i) \\ &= \sum_k p_{kj} p_{ik} \\ &= (\,P_{i^{\text{th}} \text{ row}}\,)\,(\,P_{j^{\text{th}} \text{ column}}\,) \end{aligned}$$

$$Pr(x_3 = j | x_0 = i) = \sum_k Pr(x_3 = j | x_0 = i \;\&\; x_2 = k) Pr(x_2 = k | x_0 = i) \quad \text{etc}$$

Probability Density Propagation

- Suppose we don't know x_0 but have a probability distribution:

$$x_0 \sim (q_0^1, q_0^2, \ldots, q_0^N) \quad \text{(row vector)}$$
$$q_0^i = Pr(x_0 = i)$$

- Q: What is $q_1^j = Pr(x_1 = j)$? (via total probability)

$$\begin{aligned}
Pr(x_1 = j) &= \sum_i Pr(x_1 = j | x_0 = i) Pr(x_0 = i) \\
&= \sum_i p_{ij} q_0^i \\
&= \begin{pmatrix} q_0^1 & \ldots & q_0^N \end{pmatrix} \underbrace{\begin{pmatrix} p_{1j} \\ \vdots \\ p_{Nj} \end{pmatrix}}_{j^{\text{th}} \text{ column of } P}
\end{aligned}$$

$$\Rightarrow q_1 = q_0 P \Rightarrow q_k = q_0 P^k \text{ or } q^+ = qP$$

Hidden Markov models

- Now suppose at each state we get an observation.
- Set $r(z; i, j) =$ probability of observing value z given a transition $i \to j$.
- Let $q^i =$ probability of being at state i (belief propagation).
- Derive $q^+ = \Phi(q, z)$.

$$
\begin{aligned}
(q^i)^+ &= \frac{Pr(x^+ = i \ \& \ \text{ observe } z)}{Pr(\text{observe } z)} \\
&= \frac{\sum_j Pr(x^+ = i \ \& \ \text{ observe } z | x = j) Pr(x = j)}{\sum_s \sum_j Pr(x^+ = s \ \& \ \text{ observe } z | x = j) Pr(x = j)} \\
&= \frac{\sum_j p_{ji} q^j r(z; j, i)}{\sum_s \sum_j p_{js} q^j r(z; j, s)}
\end{aligned}
$$

- Deterministic evolution!
- Akin to Kalman filter

Viterbi Algorithm

- Given an observation sequence:

$$\mathbf{Z} = \{z_1, z_2, ..., z_N\}$$

what is the "most likely" state sequence?

$$\mathbf{X} = \{x_0, x_1, ..., x_N\}$$

i.e.,

$$\begin{aligned}\hat{\mathbf{X}} &= \arg\max Pr(\mathbf{X}|\mathbf{Z}) \\ &= \arg\max \frac{Pr(\mathbf{X} \cap \mathbf{Z})}{Pr(\mathbf{Z})}\end{aligned}$$

- Since denominator is a fixed value, we maximize

$$\hat{\mathbf{X}} = \arg\max Pr(\mathbf{X} \cap \mathbf{Z})$$

- Notation:
  - Interval of sequence: $\mathbf{X}_{[i,j]} = \{x_i, ..., x_j\}$
  - Initial condition probability: π_{x_0}
- Rewrite:

$$\begin{aligned}Pr(\mathbf{X} \cap \mathbf{Z}) &= \pi_{x_0} Pr(\mathbf{X}_{[1,N]} \cap \mathbf{Z}|x_0) \\ &= \pi_{x_0} Pr(x_1, z_1|x_0) Pr(\mathbf{X}_{[2,N]} \cap \mathbf{Z}_{[2,N]}|\mathbf{X}_{[0,1]}, z_1) \\ &= \pi_{x_0} p_{x_0 x_1} r(z_1; x_0, x_1) Pr(\mathbf{X}_{[2,N]} \cap \mathbf{Z}_{[2,N]}|\mathbf{X}_{[0,1]}, z_1)\end{aligned}$$

- Likewise:

$$\begin{aligned}Pr(\mathbf{X}_{[2,N]} \cap \mathbf{Z}_{[2,N]}|\mathbf{X}_{[0,1]}, z_1) &= Pr(x_2, z_2|\mathbf{X}_{[0,1]}, z_1) Pr(\mathbf{X}_{[3,N]} \cap \mathbf{Z}_{[3,N]}|\mathbf{X}_{[0,2]}, \mathbf{Z}_{[1,2]}) \\ &= p_{x_1 x_2} r(z_2; x_1, x_2) Pr(\mathbf{X}_{[3,N]} \cap \mathbf{Z}_{[3,N]}|\mathbf{X}_{[0,2]}, \mathbf{Z}_{[1,2]})\end{aligned}$$

Viterbi Algorithm, cont

- Finally...

$$Pr(\mathbf{X} \cap \mathbf{Z}) = \pi_{x_0} \Pi_{k=1}^{N} p_{x_{k-1}x_k} r(z_k; x_{k-1}, x_k)$$

- Equivalent optimization: Maximize $\log(\cdot)$ or minimize $-\log(\cdot)$, i.e.,

$$-\log(\pi_{x_0}) - \sum_{k=1}^{N} \log(p_{x_{k-1}x_k} r(z_k; x_{k-1}, x_k))$$

  over

$$\{x_0, x_1, ..., x_N\}$$

- This looks like (deterministic!) shortest path problem:

$$\lambda_{si}^{0} = -\log(\pi_i)$$

$$\lambda_{ij}^{k} = -\log(p_{ij} r(z_k; i, j))$$

- Deterministic shortest path can be solved *forward*. Let $D_k(i)$ denote minimum arrival distance:

$$D_{k+1}(i) = \min_{j} \left\{ D_k(j) + \lambda_{ji}^{k} \right\}$$