16.410/413
Principles of Autonomy and Decision Making
Lecture 20: Intro to Hidden Markov Models

Emilio Frazzoli

Aeronautics and Astronautics
Massachusetts Institute of Technology

November 22, 2010

Assignments

Readings

- Lecture notes
- [AIMA] Ch. 15.1-3, 20.3.
- Paper on Stellar: L. Rabiner, "A tutorial on Hidden Markov Models..."

Outline

Markov Chains

Definition (Markov Chain)

A Markov chain is a sequence of random variables X_1, X_2, X_3, ..., X_t, ..., such that the probability distribution of X_{t+1} depends only on t and x_t (Markov property), in other words:

$$\Pr[X_{t+1} = x | X_t = x_t, X_{t-1} = x_{t-1}, \ldots, X_1 = x_1] = \Pr[X_{t+1} = x | X_t = x_t]$$

- If each of the random variables $\{X_t : t \in \mathbb{N}\}$ can take values in a finite set $X = \{x_1, x_2, \ldots, x_N\}$, then—for each time step t—one can define a matrix of transition probabilities T^t (transition matrix), such that

$$T_{ij}^t = \Pr[X_{t+1} = x_j | X_t = x_i]$$

- If the probability distribution of X_{t+1} depends only on the preceding state x_t (and not on the time step t), then the Markov chain is stationary, and we can describe it with a single transition matrix T.

Graph models of Markov Chains

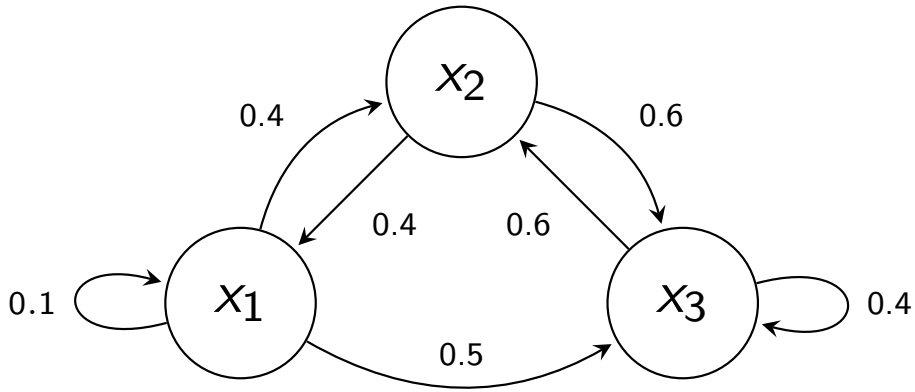
- The transition matrix has the following properties:
  - $T_{ij} \geq 0$, for all $i, j \in \{1, \ldots, N\}$.
  - $\sum_{j=1}^{N} T_{ij} = 1$, for all $i \in \{1, \ldots, N\}$ (the transition matrix is stochastic).
- A finite-state, stationary Markov Chain can be represented as a weighted graph $G = (V, E, w)$, such that
  - $V = X$
  - $E = \{(i, j) : T_{ij} > 0\}$
  - $w((i, j)) = T_{ij}$.

Whack-THE-mole

A mole has burrowed a network of underground tunnels, with N openings at ground level. We are interested in modeling the sequence of openings at which the mole will poke its head out of the ground. The probability distribution of the "next" opening only depends on the present location of the mole.

- Three holes: $X = \{x_1, x_2, x_3\}$.
- Transition probabilities:

$$T = \begin{bmatrix} 0.1 & 0.4 & 0.5 \\ 0.4 & 0 & 0.6 \\ 0 & 0.6 & 0.4 \end{bmatrix}$$



Whack-the-mole 2/3

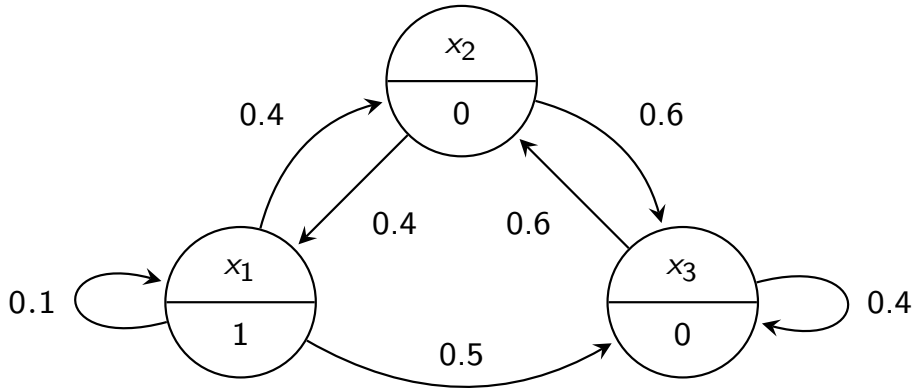
Let us assume that we know, e.g., with certainty, that the mole was at hole x_1 at time step 1 (i.e., $\Pr[X_1 = x_1] = 1$). It takes d time units to go get the mallet. Where should I wait for the mole if I want to maximize the probability of whacking it the next time it surfaces?

- Let the vector $p^t = (p_1^t, p_2^t, p_3^t)$ give the probability distribution for the location of the mole at time step t, i.e., $\Pr[X_t = x_i] = p_i^t$.
- Clearly, $p^1 = \pi = (1, 0, 0)$, and $\sum_{i=1}^{N} p_i^t = 1$, for all $t \in \mathbb{N}$.
- We have $p^{t+1} = T' p^t = T'^2 p^{t-1} = \ldots = T'^t \pi$. [1]

[1] To avoid confusion with too many "T"'s, I will use the Matlab notation M' to indicate the transpose of a matrix M.

Whack-the-mole 3/3

- Doing the calculations:
  - $p^1 = (1, 0, 0)$;
  - $p^2 = T' p^1 = (0.1, 0.4, 0.5)$;
  - $p^3 = T' p^2 = (0.17, 0.34, 0.49)$;
  - $p^4 = T' p^3 = (0.153, 0.362, 0.485)$;
  - $\ldots$
  - $p^\infty = \lim_{t \to \infty} T'^t p^1$
    $= (0.1579, 0.3553, 0.4868)$.



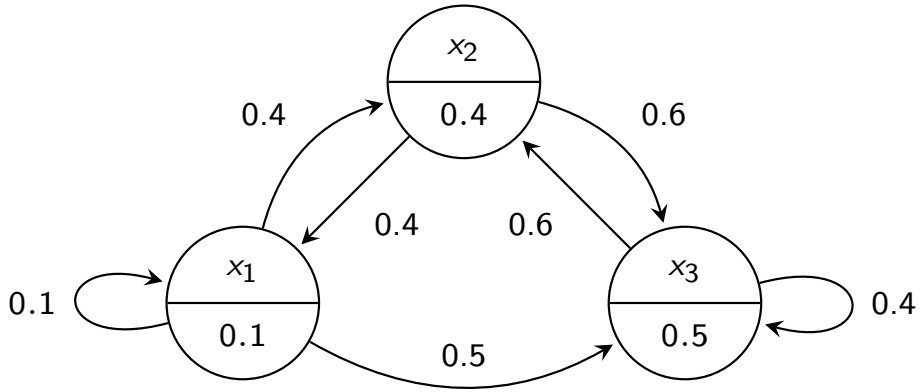
- Under some technical conditions, the state distribution of a Markov chain converge to a stationary distribution p^∞, such that

$$p^\infty = T' p^\infty.$$

- The stationary distribution can be computed as the eigenvector of the matrix T' associated with the unit eigenvalue (how do we know that T has a unit eigenvalue?), normalized so that the sum of its components is equal to one.

Whack-the-mole 3/3

- Doing the calculations:
  - $p^1 = (1, 0, 0)$;
  - $p^2 = T' p^1 = (0.1, 0.4, 0.5)$;
  - $p^3 = T' p^2 = (0.17, 0.34, 0.49)$;
  - $p^4 = T' p^3 = (0.153, 0.362, 0.485)$;
  - ...
  - $p^\infty = \lim_{t \to \infty} T'^t p^1$
    $= (0.1579, 0.3553, 0.4868)$.



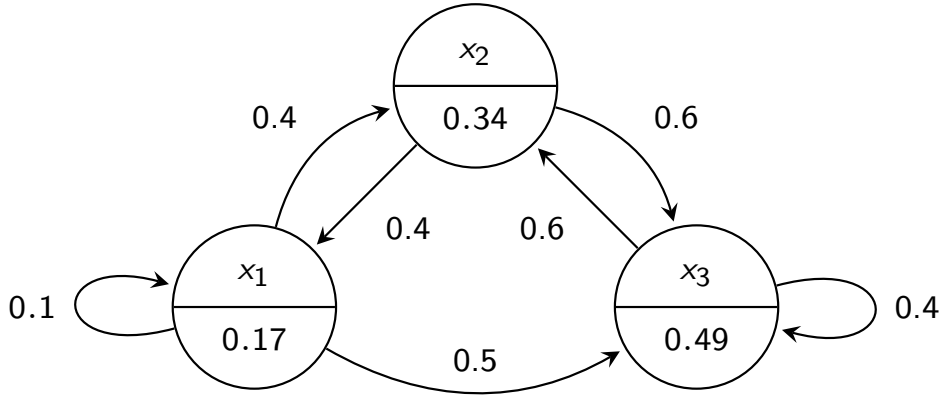
- Under some technical conditions, the state distribution of a Markov chain converge to a stationary distribution p^∞, such that

$$p^\infty = T' p^\infty.$$

- The stationary distribution can be computed as the eigenvector of the matrix T' associated with the unit eigenvalue (how do we know that T has a unit eigenvalue?), normalized so that the sum of its components is equal to one.

Whack-the-mole 3/3

- Doing the calculations:
  - $p^1 = (1, 0, 0)$;
  - $p^2 = T' p^1 = (0.1, 0.4, 0.5)$;
  - $p^3 = T' p^2 = (0.17, 0.34, 0.49)$;
  - $p^4 = T' p^3 = (0.153, 0.362, 0.485)$;
  - ...
  - $p^\infty = \lim_{t \to \infty} T'^t p^1$
    $= (0.1579, 0.3553, 0.4868)$.



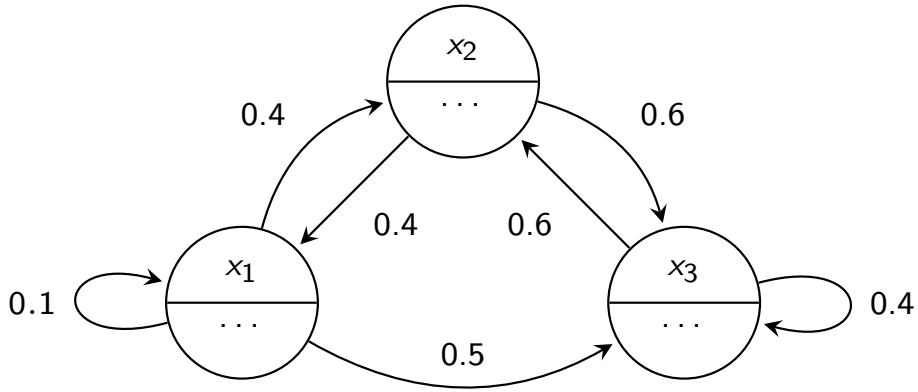
- Under some technical conditions, the state distribution of a Markov chain converge to a stationary distribution p^∞, such that

$$p^\infty = T' p^\infty.$$

- The stationary distribution can be computed as the eigenvector of the matrix T' associated with the unit eigenvalue (how do we know that T has a unit eigenvalue?), normalized so that the sum of its components is equal to one.

Whack-the-mole 3/3

- Doing the calculations:
  - $p^1 = (1, 0, 0)$;
  - $p^2 = T' p^1 = (0.1, 0.4, 0.5)$;
  - $p^3 = T' p^2 = (0.17, 0.34, 0.49)$;
  - $p^4 = T' p^3 = (0.153, 0.362, 0.485)$;
  - $\ldots$
  - $p^\infty = \lim_{t \to \infty} T'^t p^1$
    $= (0.1579, 0.3553, 0.4868)$.



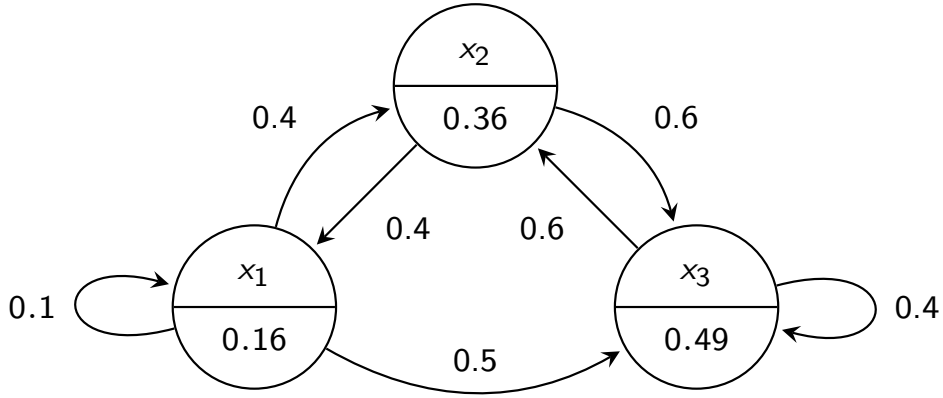
- Under some technical conditions, the state distribution of a Markov chain converge to a stationary distribution p^∞, such that

$$p^\infty = T' p^\infty.$$

- The stationary distribution can be computed as the eigenvector of the matrix T' associated with the unit eigenvalue (how do we know that T has a unit eigenvalue?), normalized so that the sum of its components is equal to one.

Whack-the-mole 3/3

- Doing the calculations:
  - $p^1 = (1, 0, 0)$;
  - $p^2 = T' p^1 = (0.1, 0.4, 0.5)$;
  - $p^3 = T' p^2 = (0.17, 0.34, 0.49)$;
  - $p^4 = T' p^3 = (0.153, 0.362, 0.485)$;
  - ...
  - $p^\infty = \lim_{t \to \infty} T'^t p^1$
    $= (0.1579, 0.3553, 0.4868)$.



- Under some technical conditions, the state distribution of a Markov chain converge to a stationary distribution p^∞, such that

$$p^\infty = T' p^\infty.$$

- The stationary distribution can be computed as the eigenvector of the matrix T' associated with the unit eigenvalue (how do we know that T has a unit eigenvalue?), normalized so that the sum of its components is equal to one.

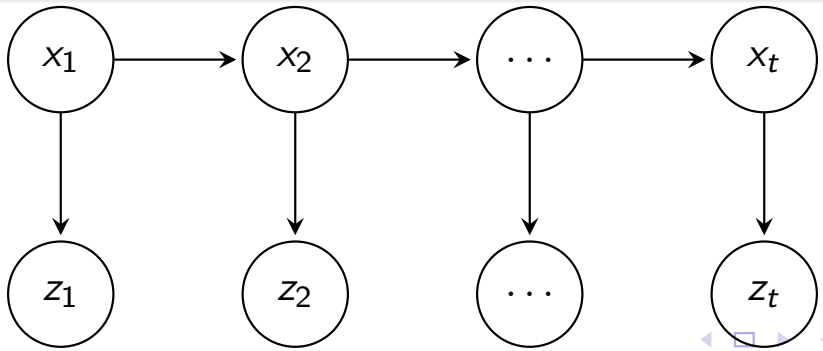
Outline

Hidden Markov Model

- In a Markov chain, we reason directly in terms of the sequence of states.
- In many applications, the state is not known, but can be (possibly partially) observed, e.g., with sensors.
- These sensors are typically noisy, i.e., the observations are random variables, whose distribution depends on the actual (unknown) state.

Definition (Hidden Markov Model)

A Hidden Markov Model (HMM) is a sequence of random variables, $Z_1, Z_2, \ldots, Z_t, \ldots$ such that the distribution of Z_t depends only on the (hidden) state x_t of an associated Markov chain.



HMM with finite observations

Definition (Hidden Markov Model)

A Hidden Markov Model (HMM) is composed of the following:

- $\mathcal{X}$: a finite set of states.
- $\mathcal{Z}$: a finite set of observations.
- $T : \mathcal{X} \times \mathcal{X} \to \mathbb{R}_+$, i.e., transition probabilities
- $M : \mathcal{X} \times \mathcal{Z} \to \mathbb{R}_+$, i.e., observation probabilities
- $\pi : \mathcal{X} \to \mathbb{R}_+$, i.e., prior probability distribution on the initial state.

If the random variables $\{Z_t : t \in \mathbb{N}\}$ take value in a finite set, we can represent a HMM using a matrix notation. For simplicity, also map $\mathcal{X}$ and $\mathcal{Z}$ to consecutive integers, starting at 1.

- T is the transition matrix.
- M is the observation (measurement) matrix: $M_{ik} = \Pr[Z_t = z_k | X_t = x_i]$.
- π is a vector.

Clearly, T, M, and π have non-negative entries, and must satisfy some normalization constraint ($\sum_j T_{ij} = \sum_k M_{ik} = \sum_l \pi_l = 1$).

Outline

Problem 1: Evaluation

The evaluation problem

Given a HMM λ, and observation history $Z = (z_1, z_2, \ldots, z_t)$, compute $\Pr[Z|\lambda]$.

- That is, given a certain HMM λ, how well does it match the observation sequence Z?
- Key difficulty is the fact that we do not know the state history $X = (x_1, x_2, \ldots, x_t)$.

The naïve approach

- In principle, we can write:

$$\Pr[Z|\lambda] = \sum_{\text{all } X} \Pr[Z|X,\lambda] \Pr[X|\lambda]$$

  where

  - $\Pr[Z|X,\lambda] = \prod_{i=1}^{t} \Pr[Z_i = z_i | X_i = x_i] = M_{x_1 z_1} M_{x_2 z_2} \dots M_{x_t z_t}$
  - $\Pr[X|\lambda] = \Pr[X_1 = x_1] \cdot \prod_{i=2}^{t} \Pr[X_i = x_i | X_{i-1} = x_{i-1}]$
    $= \pi_{x_1} T_{x_1 x_2} T_{x_2 x_3} \dots T_{x_{t-1} x_t}$

- For each possible state history X, $2t$ multiplications are needed.
- Since there are $\mathrm{card}(\mathcal{X})^t$ possible state histories, such approach would require time proportional to $t \cdot \mathrm{card}(\mathcal{X})^t$, i.e., exponential time.

The forward algorithm

- The naïve approach makes many redundant calculations. A more efficient approach exploits the Markov property of HMMs.
- Let $\alpha_k(s) = \Pr[Z_{1:k}, X_k = s|\lambda]$, where $Z_{1:k}$ is the partial sequence of observations, up to time step k.
- We can compute the vectors α_k iteratively, as follows:
  1. Initialize $\alpha_1(s) = \pi_s M_{s,z_1}$ (for all $s \in \mathcal{X}$)
  2. Repeat, for $k = 1$ to $k = t - 1$, and for all s,

$$\alpha_{k+1}(s) = M_{s,z_{k+1}} \sum_{q \in \mathcal{X}} \alpha_k(q) T_{q,s}$$

  3. Summarize the results as

$$\Pr[Z|\lambda] = \sum_{s \in \mathcal{X}} \alpha_t(s)$$

- This procedure requires time proportional to $t \cdot \mathrm{card}(\mathcal{X})^2$.

The backward algorithm

- The forward algorithm is enough to solve the evaluation problem.
- However, a similar approach working backward in time is useful for other problems.
- Let $\beta_k(s) = \Pr\left[Z_{(k+1):t} | X_k = s, \lambda\right]$, where $Z_{(k+1):t}$ is a partial observation sequence, from time step $k+1$ to the final time step t.
- We can compute the vectors β_k iteratively, as follows:
  1. Initialize $\beta_t(s) = 1$ (for all $s \in \mathcal{X}$)
  2. Repeat, for $k = t-1$ to $k = 1$, and for all s,

$$\beta_k(s) = \sum_{q \in \mathcal{X}} \beta_{k+1}(q) T_{s,q} M_{q,z_{k+1}}$$

  3. Summarize the results as

$$\Pr[Z|\lambda] = \sum_{s \in \mathcal{X}} \beta_1(s) \pi(s) M_{s,z_1}$$

- This procedure requires time proportional to $t \cdot \mathrm{card}(\mathcal{X})^2$.

Outline

Problem 2: Explanation

The explanation problem

Given a HMM λ, and an observation history $Z = (z_1, z_2, \ldots, z_t)$, find a sequence of states that best explains the observations.

We will consider slightly different versions of this problem:

- **Filtering**: given measurements up to time k, compute the distribution of X_k.
- **Smoothing**: given measurements up to time k, compute the distribution of X_j, $j < k$.
- **Prediction**: given measurements up to time k, compute the distribution of X_j, $j > k$.
- **Decoding**: Find the most likely state history X given the observation history Z.

Filtering

- We need to compute, for each $s \in \mathcal{X}$, $\Pr[X_k = s | Z_{1:k}]$.

- We have that

$$\Pr[X_k = s | Z_{1:k}, \lambda] = \frac{\Pr[X_k = s, Z_{1:k} | \lambda]}{\Pr[Z_{1:k} | \lambda]} = \eta \, \alpha_k(s)$$

where $\eta = 1/\Pr[Z_{1:k} | \lambda]$ is a normalization factor that can be computed as

$$\eta = \left(\sum_{s \in \mathcal{X}} \alpha_k(s) \right)^{-1}.$$

Smoothing

- We need to compute, for each $s \in \mathcal{X}$, $\Pr[X_k = s|Z, \lambda]$ $(k < t)$ (i.e., we use the whole observation history from time 1 to time t to update the probability distribution at time $k < t$.)
- We have that

$$\Pr[X_k = s|Z, \lambda] = \frac{\Pr[X_k = s, Z|\lambda]}{\Pr[Z|\lambda]}$$

$$= \frac{\Pr\left[X_k = s, Z_{1:k}, Z_{(k+1):t}|\lambda\right]}{\Pr[Z|\lambda]}$$

$$= \frac{\Pr\left[Z_{(k+1):t}|X_k = s, Z_{1:k}, \lambda\right] \cdot \Pr[X_k = s, Z_{1:k}|\lambda]}{\Pr[Z|\lambda]}$$

$$= \frac{\Pr\left[Z_{(k+1):t}|X_k = s, \lambda\right] \cdot \Pr[X_k = s, Z_{1:k}|\lambda]}{\Pr[Z|\lambda]}$$

$$= \frac{\beta_k(s)\alpha_k(s)}{\Pr[Z|\lambda]}.$$

Whack-the-mole example

- Let us assume that that every time the mole surfaces, we can hear it, but not see it (it's dark outside)
- our hearing is not very precise, and has the following measurement probabilities:

$$M = \begin{bmatrix} 0.6 & 0.2 & 0.2 \\ 0.2 & 0.6 & 0.2 \\ 0.2 & 0.2 & 0.6 \end{bmatrix}$$

- Let us assume that over three times the mole surfaces, we make the following measurements: (1, 3, 3)
- Compute the distribution of the states of the mole, as well as its most likely state trajectory.

Whack-the-mole example

- Forward-backward
  - $\alpha_1 = (0.6, 0, 0)$ $\beta_1 = (0.1512, 0.1616, 0.1392)$
  - $\alpha_2 = (0.012, 0.048, 0.18)$ $\beta_2 = (0.4, 0.44, 0.36)$.
  - $\alpha_3 = (0.0041, 0.0226, 0.0641)$ $\beta_3 = (1, 1, 1)$.
- Filtering/smoothing
  - $\pi_1^f = (1, 0, 0)$, $\pi_1^s = (1, 0, 0)$
  - $\pi_2^f = (0.05, 0.2, 0.75)$, $\pi_2^s = (0.0529, 0.2328, 0.7143)$
  - $\pi_3^f = (0.0450, 0.2487, 0.7063)$, $\pi_3^s = (0.0450, 0.2487, 0.7063)$.

Prediction

- We need to compute, for each $s \in \mathcal{X}$, $\Pr[X_k = s | Z, \lambda]$ $(k > t)$
- Since for all times $> t$ we have no measurements, we can only propagate in the future "blindly," without applying measurements.
- Use filtering to compute $\pi = \Pr[X_t = s | Z, \lambda]$.
- Use π as an initial condition for a Markov Chain (no measurements), propagate for $k - t$ steps.

Decoding

- Filtering and smoothing produce distributions of states at each time step.
- Maximum likelihood estimation chooses the state with the highest probability at the "best" estimate at each time step.
- However, these are pointwise best estimate: the sequence of maximum likelihood estimates is not necessarily a good (or feasible) trajectory for the HMM!
- How do we find the most likely state history, or state trajectory? (As opposed to the sequence of point-wise most likely states?)

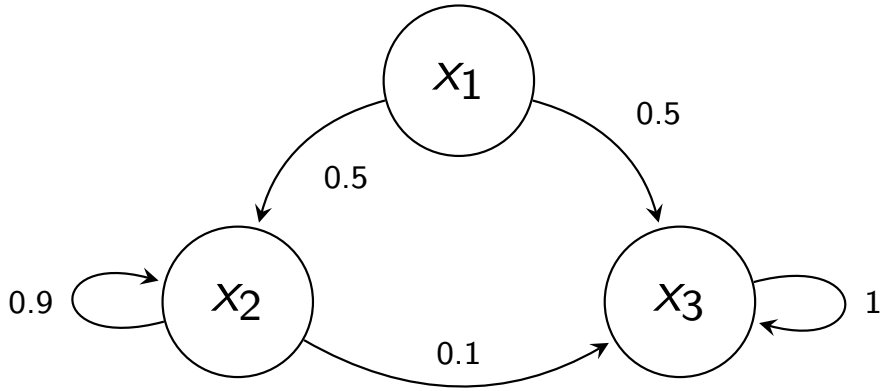
Example: filtering/smoothing vs. decoding 1/4

- Three states:
  $\mathcal{X} = \{x_1, x_2, x_3\}$.
- Three possible observations:
  $\mathcal{Z} = \{2, 3\}$.
- Initial distribution:
  $\pi = (1, 0, 0)$.
- Transition probabilities:

$$T = \begin{bmatrix} 0 & 0.5 & 0.5 \\ 0 & 0.9 & 0.1 \\ 0 & 0.1 & 0.9 \end{bmatrix}$$

- Observation probabilities:

$$M = \begin{bmatrix} 0.5 & 0.5 \\ 0.9 & 0.1 \\ 0.1 & 0.9 \end{bmatrix}$$



Observation sequence:

$$Z = (2, 3, 3, 2, 2, 2, 3, 2, 3).$$

- Using filtering:

t	x_1	x_2	x_3
1	**1.0000**	0	0
2	0	0.1000	**0.9000**
3	0	0.0109	**0.9891**
4	0	0.0817	**0.9183**
5	0	0.4165	**0.5835**
6	0	**0.8437**	0.1563
7	0	0.2595	**0.7405**
8	0	**0.7328**	0.2672
9	0	0.1771	**0.8229**

- The sequence of *point-wise* most likely states is:

$$(1, 3, 3, 3, 3, 2, 3, 2, 3).$$

- The above sequence is not feasible for the HMM model!

- Using smoothing:

t	x_1	x_2	x_3
1	**1.0000**	0	0
2	0	**0.6297**	0.3703
3	0	**0.6255**	0.3745
4	0	**0.6251**	0.3749
5	0	**0.6218**	0.3782
6	0	**0.5948**	0.4052
7	0	0.3761	**0.6239**
8	0	0.3543	**0.6457**
9	0	0.1771	**0.8229**

- The sequence of *point-wise* most likely states is:

$$(1, 2, 2, 2, 2, 2, 3, 3, 3).$$

Viterbi's algorithm

- As before, let us use the Markov property of the HMM.
- Define

$$\delta_k(s) = \max_{X_{1:(k-1)}} \Pr\left[X_{1:k} = (X_{1:(k-1)}, s), Z_{1:k} | \lambda\right]$$

(i.e., $\delta_k(s)$ is the joint probability of the most likely path that ends at state s at time k, generating observations $Z_{1:k}$.)

- Clearly,

$$\delta_{k+1}(s) = \max_q \left(\delta_k(q) T_{q,s}\right) M_{s,z_{k+1}}$$

- This can be iterated to find the probability of the most likely path that ends at each possible state s at the final time. Among these, the highest probability path is the desired solution.
- We need to keep track of the path...

Viterbi's algorithm 2/3

- Initialization, for all $s \in \mathcal{X}$:
  - $\delta_1(s) = \pi_s M_{s,z_1}$
  - $\mathrm{Pre}_1(s) = \texttt{null}$.
- Repeat, for $k = 1, \ldots, t-1$, and for all $s \in \mathcal{X}$:
  - $\delta_{k+1}(s) = \max_q \left(\delta_k(q) T_{q,s}\right) M_{s,z_{k+1}}$
  - $\mathrm{Pre}_{k+1}(s) = \arg\max_q \left(\delta_k(q) T_{q,s}\right)$
- Select most likely terminal state: $s_t^* = \arg\max_s \delta_t(s)$
- Backtrack to find most likely path. For $k = t-1, \ldots, 1$
  - $q_k^* = \mathrm{Pre}_{k+1}(q_{k+1}^*)$
- The joint probability of the most likely path + observations is found as $p^* = \delta_t(s_t^*)$.

Whack-the-mole example

- Viterbi's algorithm
  - $\delta_1 = (0.6, 0, 0)$ $\quad \mathrm{Pre}_1 = (\texttt{null}, \texttt{null}, \texttt{null})$
  - $\delta_2 = (0.012, 0.048, 0.18)$ $\quad \mathrm{Pre}_2 = (1, 1, 1).$
  - $\delta_3 = (0.0038, 0.0216, 0.0432)$ $\quad \mathrm{Pre}_3 = (2, 3, 3).$
- Joint probability of the most likely path + observations: 0.0432
- End state of the most likely path: 3
- Most likely path: $3 \leftarrow 3 \leftarrow 1$.

- Using Viterbi's algorithm:

t	x_1	x_2	x_3
1	0.5/0	0	0
2	0/1	0.025/1	0.225/1
3	0/1	0.00225/2	0.2025/3
4	0/1	0.0018225/2	0.02025/3
5	0/1	0.0014762/2	0.002025/3
6	0/1	0.0011957/2	0.0002025/3
7	0/1	0.00010762/2	0.00018225/3
8	0/1	8.717e-05/2	1.8225e-05/3
9	0/1	7.8453e-06/2	1.6403e-05/3

- The most likely sequence is:

$$(1, 3, 3, 3, 3, 3, 3, 3, 3).$$

- *Note: Based on the first 8 observations, the most likely sequence would have been*

$$(1, 2, 2, 2, 2, 2, 2, 2)!$$

Viterbi's algorithm 3/3

- Viterbi's algorithm is similar to the forward algorithm, with the difference that the summation over the states at time step k becomes a maximization.
- The time complexity is, as for the forward algorithm, linear in t (and quadratic in $\mathrm{card}(\mathcal{X})$).
- The space complexity is also linear in t (unlike the forward algorithm), since we need to keep track of the "pointers" Pre_k.
- Viterbi's algorithm is used in most communication devices (e.g., cell phones, wireless network cards, etc.) to decode messages in noisy channels; it also has widespread applications in speech recognition.

16.410 / 16.413 Principles of Autonomy and Decision Making

Fall 2010