MATH 829: Introduction to Data Mining and Analysis
Hidden Markov Models

Dominique Guillot

Departments of Mathematical Sciences
University of Delaware

May 11, 2016

Hidden Markov Models

Recall: a (discrete time homogeneous) Markov chain $(X_n)_{n\geq 0}$ is a process that satisfies:

$$P(X_{n+1} = j | X_0 = i_0, \ldots, X_{n-1} = i_{n-1}, X_n = i) = P(X_{n+1} = j | X_n = i)$$
$$= P(X_1 = j | X_0 = i)$$
$$=: p(i, j).$$

Hidden Markov Models

Recall: a (discrete time homogeneous) Markov chain $(X_n)_{n\geq 0}$ is a process that satisfies:

$$P(X_{n+1} = j | X_0 = i_0, \ldots, X_{n-1} = i_{n-1}, X_n = i) = P(X_{n+1} = j | X_n = i)$$
$$= P(X_1 = j | X_0 = i)$$
$$=: p(i,j).$$

A **Hidden Markov Model** has two components:

Hidden Markov Models

Recall: a (discrete time homogeneous) Markov chain $(X_n)_{n\geq 0}$ is a process that satisfies:

$$
\begin{aligned}
&P(X_{n+1} = j | X_0 = i_0, \ldots, X_{n-1} = i_{n-1}, X_n = i) = P(X_{n+1} = j | X_n = i) \\
&= P(X_1 = j | X_0 = i) \\
&=: p(i,j).
\end{aligned}
$$

A **Hidden Markov Model** has two components:

1. A Markov chain that describes the **state** of the system and is **unobserved**.

Hidden Markov Models

Recall: a (discrete time homogeneous) Markov chain $(X_n)_{n\geq 0}$ is a process that satisfies:

$$P(X_{n+1}=j|X_0=i_0,\ldots,X_{n-1}=i_{n-1},X_n=i)=P(X_{n+1}=j|X_n=i)$$
$$=P(X_1=j|X_0=i)$$
$$=:p(i,j).$$

A **Hidden Markov Model** has two components:

1. A Markov chain that describes the **state** of the system and is **unobserved**.
2. An **observed** process where each output depends on the state of the chain.

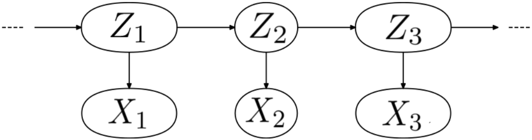
Hidden Markov Models

Recall: a (discrete time homogeneous) Markov chain $(X_n)_{n\geq 0}$ is a process that satisfies:

$$P(X_{n+1} = j | X_0 = i_0, \ldots, X_{n-1} = i_{n-1}, X_n = i) = P(X_{n+1} = j | X_n = i)$$
$$= P(X_1 = j | X_0 = i)$$
$$=: p(i, j).$$

A **Hidden Markov Model** has two components:

1. A Markov chain that describes the **state** of the system and is **unobserved**.
2. An **observed** process where each output depends on the state of the chain.



Hidden Markov Models (cont.)

More precisely, a **Hidden Markov Model** consists of:

Hidden Markov Models (cont.)

More precisely, a **Hidden Markov Model** consists of:

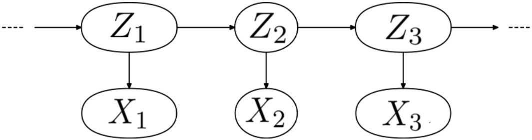
1. A Makov chain $(Z_t : t = 1, \ldots, T)$ with states $S := \{s_1, \ldots, s_{|S|}\}$, say:

$$P(Z_{t+1} = s_j | Z_t = s_i) = A_{ij}.$$

More precisely, a **Hidden Markov Model** consists of:

1. A Makov chain $(Z_t : t = 1, \dots, T)$ with states $S := \{s_1, \dots, s_{|S|}\}$, say:
$$P(Z_{t+1} = s_j | Z_t = s_i) = A_{ij}.$$

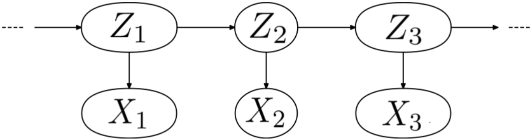
2. An observation process $(X_t : t = 1, \dots, T)$ taking values in $V := \{v_1, \dots, v_{|V|}\}$ such that
$$P(X_t = v_j | Z_t = s_i) = B_{ij}.$$



Hidden Markov Models (cont.)

More precisely, a **Hidden Markov Model** consists of:

1. A Makov chain $(Z_t : t = 1, \ldots, T)$ with states $S := \{s_1, \ldots, s_{|S|}\}$, say:
$$P(Z_{t+1} = s_j | Z_t = s_i) = A_{ij}.$$
2. An observation process $(X_t : t = 1, \ldots, T)$ taking values in $V := \{v_1, \ldots, v_{|V|}\}$ such that
$$P(X_t = v_j | Z_t = s_i) = B_{ij}.$$

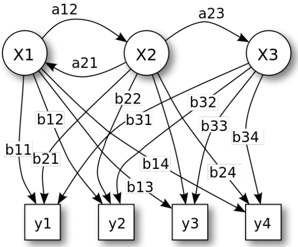


Remarks:

1. The observed variable X_t depends **only** on Z_t, the state of the Markov chain at time t.
2. The output is a **random** function of the current state.

Examples

A HMM with states $S = \{x_1, x_2, x_3\}$ and possible observations $V = \{y_1, y_2, y_3, y_4\}$.



Source: Wikipedia.

- a's are the state transition probabilities.
- b's are the output probabilities.

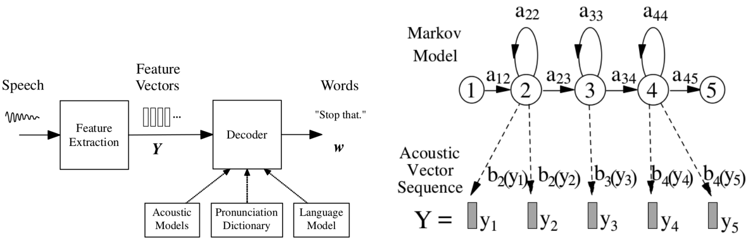
Examples (cont.)

Examples of applications:

- Recognizing human facial expression from sequences of images (see e.g. Schmidt et al, 2010).

Examples of applications:

- Recognizing human facial expression from sequences of images (see e.g. Schmidt et al, 2010).
- Speech recognition systems (see e.g. Gales and Young, 2007)

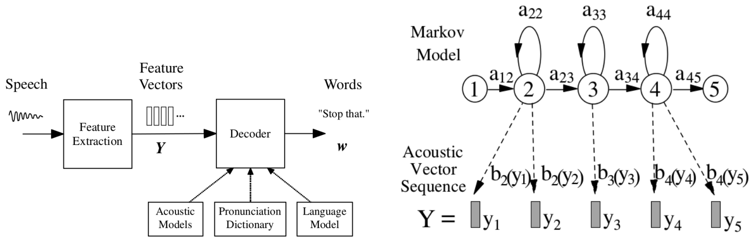


Gales and Young, 2007.

Examples (cont.)

Examples of applications:

- Recognizing human facial expression from sequences of images (see e.g. Schmidt et al, 2010).
- Speech recognition systems (see e.g. Gales and Young, 2007)



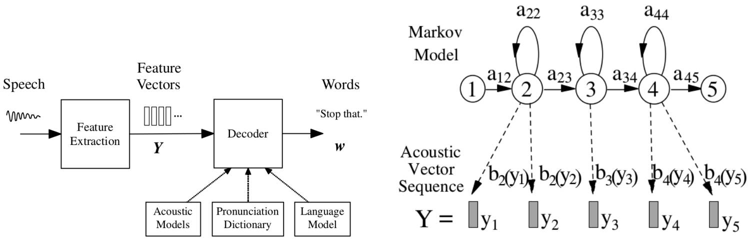
Gales and Young, 2007.

- Longitudinal comparisons in medical studies (see e.g. Scott et al. 2005).

Examples (cont.)

Examples of applications:

- Recognizing human facial expression from sequences of images (see e.g. Schmidt et al, 2010).
- Speech recognition systems (see e.g. Gales and Young, 2007)



Gales and Young, 2007.

- Longitudinal comparisons in medical studies (see e.g. Scott et al. 2005).
- Many applications in finance (e.g. pricing options, valuation of life insurance policies, credit risk modeling, etc.).
- etc..

Three problems

Three (closely related) important problems naturally arise when working with HMM:

Three problems

Three (closely related) important problems naturally arise when working with HMM:

1. What is the probability of a given observed sequence?

Three problems

Three (closely related) important problems naturally arise when working with HMM:

1. What is the probability of a given observed sequence?
2. What is the most likely series of states that generated a given observed sequence?

Three problems

Three (closely related) important problems naturally arise when working with HMM:

1. What is the probability of a given observed sequence?
2. What is the most likely series of states that generated a given observed sequence?
3. What are the state transition probabilities and the observation probabilities of the model (i.e., how can we estimate the parameters of the model)?

Probability of an observed sequence

- Suppose the parameters of the model are known.

Probability of an observed sequence

- Suppose the parameters of the model are known.
- Let $x = (x_1, \ldots, x_T) \in V^T$ be a given observed sequence.

Probability of an observed sequence

- Suppose the parameters of the model are known.
- Let $x = (x_1, \ldots, x_T) \in V^T$ be a given observed sequence.
- What is $P(x; A, B)$?

Probability of an observed sequence

- Suppose the parameters of the model are known.
- Let $x = (x_1, \ldots, x_T) \in V^T$ be a given observed sequence.
- What is $P(x; A, B)$?

Conditioning on the hidden states, we obtain:

Probability of an observed sequence

- Suppose the parameters of the model are known.
- Let $x = (x_1, \ldots, x_T) \in V^T$ be a given observed sequence.
- What is $P(x; A, B)$?

Conditioning on the hidden states, we obtain:

$$P(x; A, B) = \sum_{z \in S^T} P(x|z; A, B) P(z; A, B)$$

- Suppose the parameters of the model are known.
- Let $x = (x_1, \ldots, x_T) \in V^T$ be a given observed sequence.
- What is $P(x; A, B)$?

Conditioning on the hidden states, we obtain:

$$\begin{aligned} P(x; A, B) &= \sum_{z \in S^T} P(x|z; A, B) P(z; A, B) \\ &= \sum_{z \in S^T} \prod_{i=1}^{T} P(x_i|z_i; B) \cdot \prod_{i=1}^{T} P(z_i|z_{i-1}; A) \end{aligned}$$

Probability of an observed sequence

- Suppose the parameters of the model are known.
- Let $x = (x_1, \ldots, x_T) \in V^T$ be a given observed sequence.
- What is $P(x; A, B)$?

Conditioning on the hidden states, we obtain:

$$
\begin{aligned}
P(x; A, B) &= \sum_{z \in S^T} P(x|z; A, B) P(z; A, B) \\
&= \sum_{z \in S^T} \prod_{i=1}^{T} P(x_i | z_i; B) \cdot \prod_{i=1}^{T} P(z_i | z_{i-1}; A) \\
&= \sum_{z \in S^T} \prod_{i=1}^{T} B_{z_i, x_i} \cdot \prod_{i=1}^{T} A_{z_{i-1}, z_i}.
\end{aligned}
$$

Probability of an observed sequence

- Suppose the parameters of the model are known.
- Let $x = (x_1, \ldots, x_T) \in V^T$ be a given observed sequence.
- What is $P(x; A, B)$?

Conditioning on the hidden states, we obtain:

$$\begin{aligned} P(x; A, B) &= \sum_{z \in S^T} P(x|z; A, B) P(z; A, B) \\ &= \sum_{z \in S^T} \prod_{i=1}^{T} P(x_i|z_i; B) \cdot \prod_{i=1}^{T} P(z_i|z_{i-1}; A) \\ &= \sum_{z \in S^T} \prod_{i=1}^{T} B_{z_i, x_i} \cdot \prod_{i=1}^{T} A_{z_{i-1}, z_i}. \end{aligned}$$

Problem: Although the previous expression is simple, it involves summing over a set of size $|S|^T$, which is generally too computationally intensive.

Probability of an observed sequence (cont.)

- We can compute $P(x; A, B)$ efficiently using *dynamic programming*.

Probability of an observed sequence (cont.)

- We can compute $P(x; A, B)$ efficiently using *dynamic programming*.
- Idea: avoid computing the same quantities multiple times!

Probability of an observed sequence (cont.)

- We can compute $P(x; A, B)$ efficiently using *dynamic programming*.
- Idea: avoid computing the same quantities multiple times!
- Let $\alpha_i(t) := P(x_1, x_2, \ldots, x_t, z_t = s_i; A, B)$.

Probability of an observed sequence (cont.)

- We can compute $P(x; A, B)$ efficiently using *dynamic programming*.
- Idea: avoid computing the same quantities multiple times!
- Let $\alpha_i(t) := P(x_1, x_2, \ldots, x_t, z_t = s_i; A, B)$.

The Forward Procedure for computing $\alpha_i(t)$

---

1. Initialize $\alpha_i(0) := A_{0,i},\ i = 1, \ldots, |S|$.
2. Recursion: $\alpha_j(t) := \sum_{i=1}^{|S|} \alpha_i(t-1) A_{ij} B_{j,x_t},\ j = 1, \ldots, |S|,$ $t = 1, \ldots, T$.

Probability of an observed sequence (cont.)

- We can compute $P(x; A, B)$ efficiently using *dynamic programming*.
- Idea: avoid computing the same quantities multiple times!
- Let $\alpha_i(t) := P(x_1, x_2, \ldots, x_t, z_t = s_i; A, B)$.

The Forward Procedure for computing $\alpha_i(t)$

---

1. Initialize $\alpha_i(0) := A_{0,i}$, $i = 1, \ldots, |S|$.
2. Recursion: $\alpha_j(t) := \sum_{i=1}^{|S|} \alpha_i(t-1) A_{ij} B_{j,x_t}$, $j = 1, \ldots, |S|$, $t = 1, \ldots, T$.

Now,

$$
\begin{aligned}
P(x; A, B) &= P(x_1, \ldots, x_T; A, B) \\
&= \sum_{i=1}^{|S|} P(x_1, \ldots, x_T, z_T = s_i; A, B) \\
&= \sum_{i=1}^{|S|} \alpha_i(T).
\end{aligned}
$$

Probability of an observed sequence (cont.)

- We can compute $P(x; A, B)$ efficiently using *dynamic programming*.
- Idea: avoid computing the same quantities multiple times!
- Let $\alpha_i(t) := P(x_1, x_2, \dots, x_t, z_t = s_i; A, B)$.

The Forward Procedure for computing $\alpha_i(t)$

---

1. Initialize $\alpha_i(0) := A_{0,i},\ i = 1, \dots, |S|$.
2. Recursion: $\alpha_j(t) := \sum_{i=1}^{|S|} \alpha_i(t-1) A_{ij} B_{j,x_t},\ j = 1, \dots, |S|$, $t = 1, \dots, T$.

Now,

$$\begin{aligned} P(x; A, B) &= P(x_1, \dots, x_T; A, B) \\ &= \sum_{i=1}^{|S|} P(x_1, \dots, x_T, z_T = s_i; A, B) \\ &= \sum_{i=1}^{|S|} \alpha_i(T). \end{aligned}$$

Complexity is now $O(|S| \cdot T)$ instead of $O(|S|^T)$!

Inferring the hidden states

- One of the most natural question one can ask about a HMM is: *what are the mostly likely states that generated the observations?*

Inferring the hidden states

- One of the most natural question one can ask about a HMM is: *what are the mostly likely states that generated the observations?*
- In other words, we would like to compute:

$$\operatorname*{argmax}_{z \in S^T} P(z|x; A, B).$$

Inferring the hidden states

- One of the most natural question one can ask about a HMM is: *what are the mostly likely states that generated the observations?*
- In other words, we would like to compute:

$$\operatorname*{argmax}_{z \in S^T} P(z|x; A, B).$$

- Using Bayes' theorem:

Inferring the hidden states

- One of the most natural question one can ask about a HMM is: *what are the mostly likely states that generated the observations?*
- In other words, we would like to compute:

$$\underset{z \in S^T}{\operatorname{argmax}}\, P(z|x; A, B).$$

- Using Bayes' theorem:

$$\underset{z \in S^T}{\operatorname{argmax}}\, P(z|x; A, B) = \underset{z \in S^T}{\operatorname{argmax}} \frac{P(x|z; A, B)P(z; A)}{P(x; A, B)}$$

Inferring the hidden states

- One of the most natural question one can ask about a HMM is: *what are the mostly likely states that generated the observations?*
- In other words, we would like to compute:

$$\operatorname*{argmax}_{z \in S^T} P(z|x; A, B).$$

- Using Bayes' theorem:

$$\begin{aligned}
\operatorname*{argmax}_{z \in S^T} P(z|x; A, B) &= \operatorname*{argmax}_{z \in S^T} \frac{P(x|z; A, B) P(z; A)}{P(x; A, B)} \\
&= \operatorname*{argmax}_{z \in S^T} P(x|z; A, B) P(z; A)
\end{aligned}$$

since the denominator does not depend on z.

Inferring the hidden states

- One of the most natural question one can ask about a HMM is: *what are the mostly likely states that generated the observations?*
- In other words, we would like to compute:

$$\operatorname*{argmax}_{z \in S^T} P(z|x; A, B).$$

- Using Bayes' theorem:

$$\begin{aligned}
\operatorname*{argmax}_{z \in S^T} P(z|x; A, B) &= \operatorname*{argmax}_{z \in S^T} \frac{P(x|z; A, B)P(z; A)}{P(x; A, B)} \\
&= \operatorname*{argmax}_{z \in S^T} P(x|z; A, B)P(z; A)
\end{aligned}$$

since the denominator does not depend on z.
- Note: There are $|S|^T$ possibilities for z so there is no hope of examining all of them to pick the optimal one in practice.

The Viterbi algorithm

- The **Viterbi algorithm** is a dynamic programming algorithm that can be used to efficiently compute the most likely path for the states, given a sequence of observations $x \in V^T$.

The Viterbi algorithm

- The **Viterbi algorithm** is a dynamic programming algorithm that can be used to efficiently compute the most likely path for the states, given a sequence of observations $x \in V^T$.
- Let $v_i(t)$ denote the most probable path that ends in state s_i at time t:

$$v_i(t) := \max_{z_t, \ldots, z_{t-1}} P(z_1, \ldots, z_{t-1}, z_t = s_i, x_1, \ldots, x_t; A, B).$$

The Viterbi algorithm

- The **Viterbi algorithm** is a dynamic programming algorithm that can be used to efficiently compute the most likely path for the states, given a sequence of observations $x \in V^T$.
- Let $v_i(t)$ denote the most probable path that ends in state s_i at time t:

$$v_i(t) := \max_{z_t,\ldots,z_{t-1}} P(z_1, \ldots, z_{t-1}, z_t = s_i, x_1, \ldots, x_t; A, B).$$

Key observation: We have

$$v_j(t) = \max_{1 \le i \le |S|} v_i(t-1) A_{ij} B_{j,x_t}.$$

The Viterbi algorithm

- The **Viterbi algorithm** is a dynamic programming algorithm that can be used to efficiently compute the most likely path for the states, given a sequence of observations $x \in V^T$.
- Let $v_i(t)$ denote the most probable path that ends in state s_i at time t:

$$v_i(t) := \max_{z_t, \ldots, z_{t-1}} P(z_1, \ldots, z_{t-1}, z_t = s_i, x_1, \ldots, x_t; A, B).$$

Key observation: We have

$$v_j(t) = \max_{1 \le i \le |S|} v_i(t-1) A_{ij} B_{j,x_t}.$$

In other words:

$$\begin{aligned} &\text{Best Path at } t \text{ that end at } j \\ &= \max_{1 \le i \le |S|} (\text{Best Path at } t-1 \text{ that end at } i) \\ &\times (\text{Go from } i \text{ to } j) \\ &\times (\text{Observe } x_t \text{ in state } s_j). \end{aligned}$$

The **Viterbi algorithm**:

1. Initialize $v_i(1) := \pi_i B_{i,x_1}$, $i = 1, \ldots, |S|$, where π_i is the *initial* distribution of the Markov chain.

The **Viterbi algorithm**:

1. Initialize $v_i(1) := \pi_i B_{i,x_1}$, $i = 1, \ldots, |S|$, where π_i is the *initial* distribution of the Markov chain.
2. Compute $v_i(t)$ recursively for $i = 1, \ldots, S$ and $t = 1, \ldots, T$.

The Viterbi algorithm

The **Viterbi algorithm**:

1. Initialize $v_i(1) := \pi_i B_{i,x_1}$, $i = 1, \ldots, |S|$, where π_i is the *initial* distribution of the Markov chain.
2. Compute $v_i(t)$ recursively for $i = 1, \ldots, S$ and $t = 1, \ldots, T$.
3. Finally, the most probable path is the path corresponding to

$$\max_{1 \leq i \leq |S|} v_i(T).$$

Estimating A, B, and π

- So far, we assumed the parameters A, B, and π of the HMM were known.

Estimating A, B, and π

- So far, we assumed the parameters A, B, and π of the HMM were known.
- We now turn to the estimation of these parameters.

Estimating A, B, and π

- So far, we assumed the parameters A, B, and π of the HMM were known.
- We now turn to the estimation of these parameters.
- Let $\theta := (A, B, \pi)$.

Estimating A, B, and π

- So far, we assumed the parameters A, B, and π of the HMM were known.
- We now turn to the estimation of these parameters.
- Let $\theta := (A, B, \pi)$.
- We know how to compute:
  1. $P(x|\theta)$ Forward algorithm.
  2. $P(z|x;\theta)$ Viterbi algorithm.

Estimating A, B, and π

- So far, we assumed the parameters A, B, and π of the HMM were known.
- We now turn to the estimation of these parameters.
- Let $\theta := (A, B, \pi)$.
- We know how to compute:
  1. $P(x|\theta)$ Forward algorithm.
  2. $P(z|x;\theta)$ Viterbi algorithm.
- We now want

$$\operatorname*{argmax}_{\theta} P(x|\theta),$$

i.e., the set of parameters for which the observed values are most likely to be obtained.

Estimating A, B, and π

- So far, we assumed the parameters A, B, and π of the HMM were known.
- We now turn to the estimation of these parameters.
- Let $\theta := (A, B, \pi)$.
- We know how to compute:
  1. $P(x|\theta)$ Forward algorithm.
  2. $P(z|x;\theta)$ Viterbi algorithm.
- We now want

$$\operatorname*{argmax}_{\theta} P(x|\theta),$$

i.e., the set of parameters for which the observed values are most likely to be obtained.

- Note: if we could observe z, then we could easily compute A, B, π.

Estimating A, B, and π

- So far, we assumed the parameters A, B, and π of the HMM were known.
- We now turn to the estimation of these parameters.
- Let $\theta := (A, B, \pi)$.
- We know how to compute:
  1. $P(x|\theta)$ Forward algorithm.
  2. $P(z|x;\theta)$ Viterbi algorithm.
- We now want

$$\operatorname*{argmax}_{\theta} P(x|\theta),$$

i.e., the set of parameters for which the observed values are most likely to be obtained.
- Note: if we could observe z, then we could easily compute A, B, π.
- We solve the problem using the **EM algorithm**.