HIDDEN MARKOV MODELS

Sequence Learning

Lê Hồng Phương

<*phuonglh@gmail.com*>
Vietnam National University of Hanoi
Hanoi University of Science

August 2017

Content

Content

Content

Content

Content

Content

Introduction

- Mô hình Markov ẩn (*Hidden Markov Models*–HMMs) là một trong những thành phần quan trọng của các mô hình thống kê trong các hệ thống xử lí tiếng nói và ngôn ngữ hiện đại.
- Mô hình Markov thực chất là một hàm xác suất của một quá trình Markov.
- Mô hình Markov được nghiên cứu đầu tiên bởi Andrei A. Markov với mục đích ban đầu là để mô hình hóa dãy chữ cái trong các tác phẩm văn học tiếng Nga.
- Sau đó các mô hình Markov được phát triển cho các công cụ thống kê tổng quát và có nhiều ứng dụng quan trọng trong các bài toán nhận dạng và phân loại thường gặp trong nhiều ngành của khoa học máy tính (ngành học tự động, xử lí tiếng nói và ngôn ngữ, tin sinh học...)

Content

Markov Models

- Ta thường phải xét một dãy (theo thời gian) các biến ngẫu nhiên không độc lập, mà giá trị của mỗi biến phụ thuộc vào các thành phần trước nó trong dãy.
- Tuy nhiên, trong nhiều hệ thống như vậy, để dự đoán các biến ngẫu nhiên trong tương lai, ta chỉ cần dựa vào giá trị của biến ngẫu nhiên hiện tại mà không cần xét các giá trị của các biến ngẫu nhiên trong quá khứ.
- Nghĩa là, các phần tử tương lai của dãy độc lập có điều kiện với các phần tử trong quá khứ, nếu biết phần tử hiện tại.

Markov Models

- Giả sử $\mathbf{x} = (x_1, x_2, \ldots, x_T)$ là một dãy biến ngẫu nhiên lấy giá trị trong một tập hữu hạn gồm n trạng thái $S = \{s_1, s_2, \ldots, s_n\}$.
- Dãy $\mathbf{x}$ được gọi là một mô hình Markov nếu nó thoả mãn hai *tính chất Markov* sau:

  Độc lập với quá khứ:

$$P(x_{t+1} = s_k | x_1, \ldots, x_t) = P(x_{t+1} = s_k | x_t) \quad (1)$$

  Bất biến theo thời gian:

$$P(x_{t+1} = s_k | x_t) = P(x_2 = s_k | x_1) \quad (2)$$

Markov Models

Mỗi mô hình Markov được xác định bởi:

- Một ma trận chuyển trạng thái ngẫu nhiên $A = (a_{ij})_{i,j=1,\ldots,n}$:

$$a_{ij} = P(x_{t+1} = s_j | x_t = s_i), \tag{3}$$

  trong đó $a_{ij} \geq 0, \forall i, j$ và $\sum_{j=1}^{n} a_{ij} = 1, \forall i$. Ma trận A không phụ thuộc vào t do tính chất bất biến theo thời gian của mô hình.

- Một phân phối xác suất $\pi = (\pi_1, \pi_2, \ldots, \pi_n)$ xác định trạng thái ban đầu của mô hình, π_i là xác suất để mô hình xuất phát từ trạng thái s_i:

$$\pi_i = P(x_1 = s_i), \tag{4}$$

Markov Models

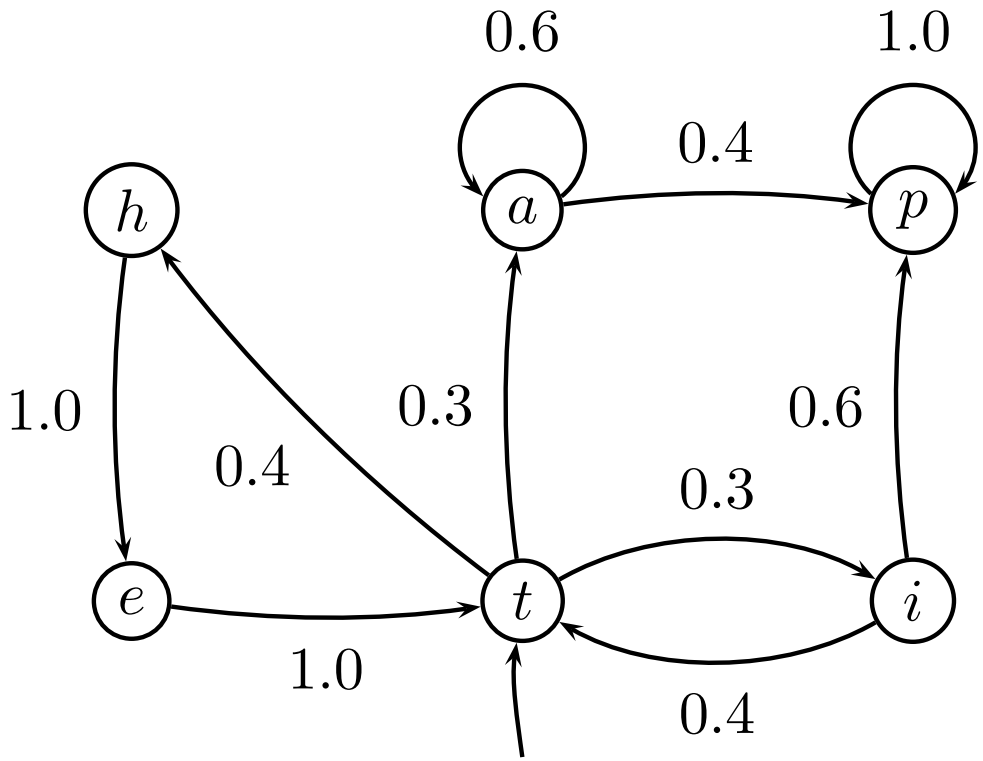
- Vì mô hình phải xuất phát từ một trong n trạng thái nên

$$\sum_{i=1}^{n} \pi_i = 1.$$

  Nếu s_j không thể là trạng thái khởi đầu thì $\pi_j = 0$.
- Ta có thể không cần xác định riêng phân phối π nếu
  - giả định rằng mô hình luôn xuất phát từ một *trạng thái ban đầu* s_0 nào đó;
  - sử dụng ma trận chuyển trạng thái để gắn các xác suất chuyển trạng thái từ s_0 sang các trạng thái khác. Khi đó ma trận chuyển trạng thái A có kích thước $(n+1) \times (n+1)$.)

Markov Models

Ta có thể biểu diễn mỗi mô hình Markov bằng một sơ đồ trạng thái hay ôtômát hữu hạn trạng thái (không đơn định) với các cung được gắn xác suất.



Markov Models

Dễ dàng tính xác suất của dãy trạng thái $z_1, \ldots, z_T$:

$$\begin{aligned} P(z_1, \ldots, z_T) &= P(z_1)P(z_2|z_1)P(z_3|z_1, z_2) \cdots P(z_T|z_1, \ldots, z_{T-1}) \\ &= P(z_1)P(z_2|z_1)P(z_3|z_2) \cdots P(z_T|z_{T-1}) \\ &= \pi_{z_1} \prod_{t=1}^{T-1} a_{z_t z_{t+1}} \end{aligned}$$

Với mô hình Markov trên, ta có

$$\begin{aligned} P(t, i, p) &= P(z_1 = t)P(z_2 = i|z_1 = t)P(z_3 = p|z_2 = i) \\ &= 1.0 \times 0.3 \times 0.6 \\ &= 0.18 \end{aligned}$$

Content

Hidden Markov Models

- Trong mô hình Markov ẩn, ta không biết dãy trạng thái mà mô hình đi qua, mà ta chỉ biết một hàm xác suất nào đó của nó thông qua các quan sát hay kí hiệu mà mô hình phát ra.
- Mô hình Markov ẩn là một bộ năm (S, Σ, π, A, B), trong đó
  1. S là tập trạng thái
  2. Σ là tập quan sát
  3. π là vector xác suất của các trạng thái khởi đầu
  4. A là ma trận xác suất chuyển trạng thái
  5. B là ma trận xác suất phát quan sát ứng với *các bước chuyển trạng thái*.

Hidden Markov Models

Một số ký hiệu dùng trong HMMs:.

Tập trạng thái	$S = \{s_1, \ldots, s_n\}$
Tập quan sát	Σ
Các x.s. trạng thái ban đầu	$\pi = (\pi_i)_{i=1,\ldots,n}$
Các x.s. chuyển trạng thái	$A = (a_{ij})_{i,j=1,\ldots,n}$
Xác suất phát quan sát	$B = (b_{ijk}), k = 1, 2, \ldots, \lvert\Sigma\rvert$
Dãy trạng thái	$\mathbf{z} = (z_1, \ldots, z_{T+1})$, $z_t : S \to \{1, \ldots, n\}$
Dãy quan sát	$\mathbf{x} = (x_1, \ldots, x_T)$, $x_t \in \Sigma$.

Hidden Markov Models

- Trong định nghĩa trên, B là ma trận ba chiều với kích thước $n \times n \times |\Sigma|$, b_{ijk} là xác suất để mô hình phát ra kí hiệu $x_k \in \Sigma$ khi chuyển từ trạng thái s_i sang trạng thái s_j.
- Như vậy, xác suất phát quan sát được gắn với các *bước chuyển trạng thái*.
- Do vậy, dãy quan sát gồm T quan sát tương ứng với dãy gồm $T + 1$ trạng thái.

Hidden Markov Models

- Thay vì gắn việc phát quan sát với các bước chuyển trạng thái như trên, ta có thể gắn nó với các trạng thái, dẫn tới định nghĩa khác về B.
- Khi đó, ta xác định ma trận hai chiều $B = (b_{ik})$, trong đó b_{ik} là xác suất để mô hình phát ra quan sát x_k khi đang ở trạng thái s_i.
- Trong trường hợp này, dãy quan sát gồm T quan sát tương ứng với dãy gồm T trạng thái.
- Chú ý rằng các ma trận A và B đều độc lập với $t \in \{1, 2, \ldots, T\}$.

Applications of HMMs

- HMMs can be used as black-box density models on sequences.
- They have the advantage over Markov models:
  - They can represent long-range dependencies between observations, mediated via the hidden variables.
- HMMs are widely used in time series prediction.

Applications of HMMs

It is more common to associate the hidden states with some meaning, and then try to estimate the hidden states from the observations:

- Online scenario:

$$p(z_t|x_1, x_2, \ldots, x_t) = ?$$

- Offline scenario:

$$p(z_t|x_1, x_2, \ldots, x_T) = ?$$

This task is called **decoding**.

Applications of HMMs

Some examples of applications:

- **Automatic speech recognition:**
  - $\mathbf{x}_t$ represents features extracted from the speech signal
  - z_t represents the word that is being spoken
  - $p(z_{t+1}|z_t)$ represents the language model, $p(\mathbf{x}_t\,|z_t)$ represents the acoustic model
- **Part-of-speech tagging:**
  - x_t represents a word
  - z_t represents its part-of-speech (POS) like noun, verb, adjective, *etc.*

Applications of HMMs

- **Gene finding:**
  - x_t represents a DNA nucleotides (A, C, G, T)
  - z_t represents whether we are inside a gene-coding region or not.
- **Activity recognition:**
  - $\mathbf{x}_t$ reprensents features extracted from a video frame
  - z_t is the class of activity that the person is engaged in (running, walking, sitting, *etc.*

Content

Three Basic Problems of HMMs

Có ba bài toán cơ bản đối với HMMs:

1. Biết mô hình $\theta = (A, B, \pi)$, tính xác suất của một dãy quan sát, tức tính $P(\mathbf{x}\,|\theta)$.
2. Biết một dãy quan sát $\mathbf{x}$ và mô hình θ, tìm dãy trạng thái $\mathbf{z}$ thích hợp nhất ứng với $\mathbf{x}$.
3. Biết một dãy quan sát $\mathbf{x}$ và một không gian các mô hình $\theta = (A, B, \pi)$, tìm mô hình thích hợp nhất đối với $\mathbf{x}$.

Three Basic Problems of HMMs

- Bài toán 1 được dùng để so sánh các mô hình, xem mô hình nào là tốt nhất.
- Bài toán 2 ứng dụng trong các bài toán phân loại để tìm dãy các trạng thái ẩn giải thích hợp lí nhất các thông tin quan sát được.
- Bài toán 3 là bài toán xây dựng mô hình từ dữ liệu huấn luyện.

Content

Evaluation Problem

- Cho dãy quan sát $\mathbf{x} = (x_1, \ldots, x_T)$ và mô hình $\theta = (A, B, \pi)$, tính $P(\mathbf{x}\,|\theta)$.
- Dễ thấy, xác suất để mô hình phát ra $\mathbf{x}$ là xác suất để mô hình ở trạng thái $\mathbf{z}$ và ở trạng thái đó mô hình phát ra $\mathbf{x}$, với mọi $\mathbf{z}$.

$$P(\mathbf{x}) = \sum_{\mathbf{z}} P(\mathbf{z}, \mathbf{x}) = \sum_{\mathbf{z}} P(\mathbf{z}) P(\mathbf{x}\,|\,\mathbf{z}). \tag{5}$$

Evaluation Problem

Xác suất để mô hình ở trạng thái $\mathbf{z} = (z_1, \ldots, z_{T+1})$ là

$$P(\mathbf{z}) = \pi_{z_1} \prod_{t=1}^{T} a_{z_t z_{t+1}}. \tag{6}$$

Khi mô hình ở trạng thái $\mathbf{z}$, xác suất để nó phát ra quan sát $\mathbf{x}$ là

$$P(\mathbf{x} \,|\, \mathbf{z}) = \prod_{t=1}^{T} P(x_t | z_t, z_{t+1}) = \prod_{t=1}^{T} b_{z_t z_{t+1} x_t}. \tag{7}$$

Evaluation Problem

- Thay (6) và (7) vào (5), ta có

$$P(\mathbf{x}) = \sum_{z_1 \ldots z_{T+1}} \pi_{z_1} \prod_{t=1}^{T} a_{z_t z_{t+1}} b_{z_t z_{t+1} x_t}. \tag{8}$$

- Công thức (8) tuy đơn giản nhưng không hiệu quả, thậm chí không tính được đối với mô hình kích thước lớn.[1]
- Trong thực tế, ta cần sử dụng kỹ thuật quy hoạch động để giảm số phép tính cần thực hiện.

[1] Nếu tính trực tiếp công thức này trong trường hợp tổng quát, với mô hình có n trạng thái thì ta cần tính $(2T+1)n^{T+1}$ phép nhân.

Evaluation Problem

- Kỹ thuật quy hoạch động trong HMMs thường được mô tả bằng thuật ngữ *lưới* (trellis, lattice).
- Ta sử dụng một ma trận để ghi các đại lượng xác suất theo trạng thái và thời gian.
- Mỗi phần tử (s, t) của ma trận là xác suất của mọi đường trong HMM từ trạng thái khởi đầu tới trạng thái s ở thời điểm t.
- Có hai phương pháp tính toán: *tính tiến* hoặc *tính lùi*.

Forward Procedure

Đặt

$$\alpha_{s_j}(t) = P(x_1 x_2 \dots x_{t-1}, z_t = s_j). \tag{9}$$

Tức $\alpha_{s_j}(t)$ là xác suất để mô hình ở trạng thái s_j vào thời điểm t *và* đã phát quan sát bộ phận $(x_1, x_2, \dots, x_{t-1})$.

Forward Procedure

1. Khởi tạo, $\forall j = 1, \ldots, n$:

$$\alpha_{s_j}(1) = \pi_{s_j}.$$

2. Quy nạp, $\forall j = 1, \ldots, n$; $\forall t = 1, \ldots, T$:

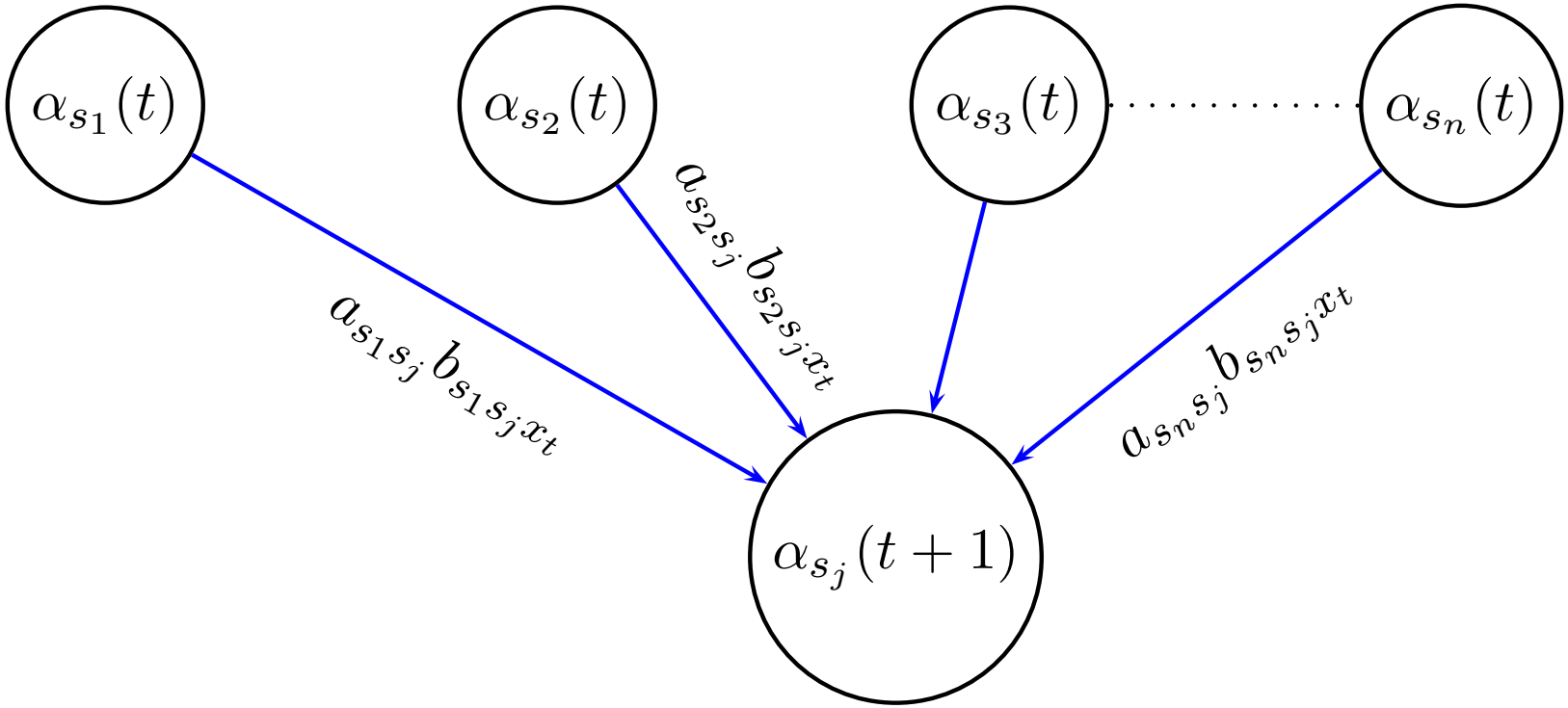
$$\alpha_{s_j}(t+1) = \sum_{i=1}^{n} \alpha_{s_i}(t) a_{s_i s_j} b_{s_i s_j x_t}.$$

3. Tính tổng cuối:

$$P(\mathbf{x}) = \sum_{j=1}^{n} \alpha_{s_j}(T+1).$$

Thuật toán này chỉ cần $2n^2T$ phép nhân.

Forward Procedure



Backward Procedure

Thủ tục lùi tính xác suất của phần còn lại của quan sát thay vì tính xác suất của phần đầu của quan sát như trong thủ tục tiến. Đặt

$$\beta_{s_j}(t) = P(x_t \cdots x_T | z_t = s_j). \tag{10}$$

Tức $\beta_{s_j}(t)$ là xác suất của chuỗi quan sát kể từ thời điểm t *biết trước* trạng thái ở thời điểm t là s_j.

Backward Procedure

1. Khởi tạo, $\forall j = 1, \ldots, n$:

$$\beta_{s_j}(T+1) = 1.$$

2. Quy nạp, $\forall j = 1, \ldots, n$; $\forall t = T, T-1, \ldots, 1$:

$$\beta_{s_j}(t) = \sum_{i=1}^{n} a_{s_j s_i} b_{s_j s_i x_t} \beta_{s_i}(t+1).$$

3. Tính tổng cuối:

$$P(\mathbf{x}) = \sum_{j=1}^{n} \pi_{s_j} \beta_{s_j}(1).$$

Forward-Backward Procedure

Ta có thể kết hợp cả hai thủ tục tiến và lùi để tính xác suất của dãy quan sát. Do

$$\begin{aligned} P(\mathbf{x}, z_t = s_j) &= P(x_1 \cdots x_t, z_t = s_j, x_{t+1} \cdots x_T) \\ &= P(x_1 \cdots x_t, z_t = s_j) \times P(x_{t+1} \cdots x_T | x_1 \cdots x_t, z_t = s_j) \\ &= P(x_1 \cdots x_t, z_t = s_j) \times P(x_{t+1} \cdots x_T | z_t = s_j) \\ &= \alpha_{s_j}(t)\beta_{s_j}(t), \end{aligned}$$

nên

$$P(\mathbf{x}) = \sum_{j=1}^{n} \alpha_{s_j}(t)\beta_{s_j}(t), \quad \forall t. \tag{11}$$

Dễ thấy các phương trình (9) và (10) là các trường hợp đặc biệt của phương trình (11).

Content

Decoding Problem

- Tìm dãy trạng thái hợp lí nhất ứng với dãy quan sát $\mathbf{x}$ đã cho, tức tìm

$$\arg\max_{\mathbf{z}} P(\mathbf{z} \,|\, \mathbf{x}, \theta)$$

- Ta có

$$P(\mathbf{z} \,|\, \mathbf{x}) = \frac{P(\mathbf{z}, \mathbf{x})}{P(\mathbf{x})}.$$

- Vì $\mathbf{x}$ cố định nên ta chỉ cần tìm $\mathbf{z} = (z_1, z_2, \ldots, z_{T+1})$ sao cho:

$$P(\mathbf{z}, \mathbf{x}) \to \max.$$

- Dễ thấy phương pháp tìm kiếm $\mathbf{z}$ bằng cách vét cạn là không khả thi, vì số dãy $\mathbf{z}$ có thể là n^{T+1}, một hàm mũ của n và T.

Decoding Problem – Viterbi Algorithm

Ta sử dụng thuật toán Viterbi như sau. Đặt

$$\pi[t, s_j] = \max_{z_1 \cdots z_{t-1}} P(z_1 \cdots z_{t-1}, x_1 \cdots x_{t-1}, z_t = s_j).$$

Tức $\pi[t, s_j]$ là xác suất của đường đi hợp lí nhất từ trạng thái khởi đầu tới trạng thái s_j tại thời điểm t.

Decoding Problem – Viterbi Algorithm

1. Khởi tạo, $\forall j = 1, \ldots, n$:

$$\pi[1, s_j] = P(s_j)$$
$$\psi[1, s_j] = 0.$$

2. Quy nạp, $\forall t = 1, 2, \ldots, T; \quad \forall j = 1, \ldots, n$:

$$\begin{aligned}
\pi[t+1, s_j] &= \max_{i=1\cdots n} \{\pi[t, s_i] \times P(s_j|s_i) \times P(x_t|s_i, s_j)\} \\
&= \max_{i=1\cdots n} \left\{\pi[t, s_i] \times a_{s_i s_j} \times b_{s_i s_j x_t}\right\} \\
\psi[t+1, s_j] &= \arg\max_{i=1\cdots n} \left\{\pi[t, s_i] \times a_{s_i s_j} \times b_{s_i s_j x_t}\right\}
\end{aligned}$$

3. Tìm đường đi:

$$\begin{aligned}
\widehat{z}_{T+1} &= \arg\max_{i=1\cdots n} \pi[T+1, s_i] \\
\widehat{z}_t &= \psi[t+1, \widehat{z}_{t+1}], \forall t = T, T-1, \ldots, 1 \\
P(\widehat{\mathbf{z}}) &= \max_{i=1\cdots n} \pi[T+1, s_i].
\end{aligned}$$

Content

Learning Problem

- Trong thực tế, mô hình $\theta = (A, B, \pi)$ thường không biết trước, mà ta cần xây dựng mô hình từ một dãy quan sát đã biết nào đó.
- Đầu vào của bài toán huấn luyện là dãy quan sát $\mathbf{x}$ và một tập trạng thái của mô hình.
- Ta cần ước lượng các tham số (A, B, π) của mô hình phù hợp với $\mathbf{x}$.

Learning Problem

- Sử dụng phương pháp ước lượng hợp lí cực đại, ta cần tìm θ cực đại hoá $P(\mathbf{x}\,|\theta)$ trên dữ liệu huấn luyện:

$$\arg\max_{\theta} P(\mathbf{x}\,|\theta).$$

- Không có phương pháp giải tích nào để chọn θ cực đại hóa $P(\mathbf{x}\,|\theta)$.
- Tuy nhiên, ta có thể tìm θ bằng các phương pháp học tự động phi giám sát

Learning Problem – Baum-Welch Algorithm

- Thuật toán Baum-Welch là một trường hợp riêng của thuật toán cực đại hoá kỳ vọng (EM) ước lượng các tham số của mô hình biến ẩn.
- Ý tưởng của thuật toán:
  - Mô hình chưa biết nhưng có thể khởi tạo các tham số bất kì (giả sử ngẫu nhiên) cho mô hình
  - Sử dụng các xác suất hậu nghiệm của các quan sát đã biết để làm tốt dần tham số của mô hình.

Learning Algorithm – Baum-Welch Algorithm

- Nếu ta có thể quan sát dãy trạng thái trong một giai đoạn dài theo thời gian và đếm số lần xuất hiện của một trạng thái s nào đó thì tần số xuất hiện này có thể là xấp xỉ của $P(z_t = s)$.
- Nếu giả định này là đúng thì ta sẽ có π_s, trong đó với một giai đoạn thời gian đủ lớn thì π sẽ là phân phối dừng của ma trận chuyển (tức π là vector riêng của ma trận chuyển ứng với giá trị riêng 1: $A\pi = \pi$).
- Nếu dãy quan sát $\mathbf{x}$ có xác suất cao trong mô hình thì tần số của các trạng thái ứng với các quan sát này cũng có thể là xấp xỉ của π_s:

$$\pi_s \approx \frac{1}{T} \sum_{t=1}^{T} P(z_t = s | \mathbf{x}).$$

Learning Algorithm – Baum-Welch Algorithm

- Tương tự, nếu ta có thể quan sát dãy trạng thái theo một giai đoạn dài của thời gian và đếm số lần trạng thái s_i và sau đó là trạng thái s_j xuất hiện thì tần số này là xấp xỉ của $P(z_t = s_i, z_{t+1} = s_j)$.
- Nếu giả định là đúng thì ta có $\pi_{s_i} a_{s_i s_j}$.
- Xấp xỉ này cũng có thể suy từ tần số kỳ vọng của các bước chuyển trạng thái ứng với các quan sát:

$$\pi_{s_i} a_{s_i s_j} \approx \frac{1}{T} \sum_{t=1}^{T} P(z_t = s_i, z_{t+1} = s_j | \mathbf{x}).$$

Learning Algorithm – Baum-Welch Algorithm

- Cuối cùng, nếu ta có thể quan sát dãy trạng thái và dãy quan sát và đếm số lần $z_t = s_i, z_{t+1} = s_j$ và x_t thì tần số này là xấp xỉ của $b_{s_i s_j x_t}$.
- Tức là

$$b_{s_i s_j x_t} \approx \frac{1}{T} \sum_{t=1}^{T} P(z_t = s_i, z_{t+1} = s_j, x_t | x_{t-1}).$$

Learning Algorithm – Baum-Welch Algorithm

- Đặt $\gamma_t(s_i, s_j)$ với $1 \leq t \leq T, 1 \leq i, j \leq n$ là xác suất mô hình đi từ trạng thái s_i tới trạng thái s_j tại thời điểm t và phát ra dãy quan sát $\mathbf{x}$.
- Ta có

$$\begin{aligned}
\gamma_t(s_i, s_j) &= P(z_t = s_i, z_{t+1} = s_j | \mathbf{x}, \theta) \\
&= \frac{P(z_t = s_i, z_{t+1} = s_j, \mathbf{x} | \theta)}{P(\mathbf{x} | \theta)} \\
&= \frac{\alpha_{s_i}(t) a_{s_i s_j} b_{s_i s_j x_t} \beta_{s_j}(t+1)}{P(\mathbf{x} | \theta)}
\end{aligned}$$

Learning Algorithm – Baum-Welch Algorithm

Đặt $\gamma_t(s_i)$ là xác suất mô hình ở trạng thái s_i tại thời điểm t, tức là

$$\begin{aligned}\gamma_t(s_i) &= P(z_t = s_i | \mathbf{x}, \theta) \\ &= \frac{P(z_t = s_i, \mathbf{x} | \theta)}{P(\mathbf{x} | \theta)} \\ &= \frac{\alpha_{s_i}(t)\beta_{s_i}(t)}{P(\mathbf{x} | \theta)}.\end{aligned}$$

Learning Algorithm – Baum-Welch Algorithm

Dễ thấy rằng

$$\gamma_t(s_i) = \sum_{j=1}^{n} \gamma_t(s_i, s_j)$$

Nếu lấy tổng theo thời gian thì ta được các giá trị kỳ vọng:

- $\sum_{t=1}^{T} \gamma_t(s_i)$ = số bước chuyển từ trạng thái s_i trong dãy quan sát $\mathbf{x}$.
- $\sum_{t=1}^{T} \gamma_t(s_i, s_j)$ = số bước chuyển từ trạng thái s_i tới trạng thái s_j trong dãy quan sát $\mathbf{x}$.

Learning Algorithm – Baum-Welch Algorithm

- Do vậy, trước tiên ta chọn một mô hình θ nào đó (chọn chủ định hay chọn ngẫu nhiên).
- Sau đó ta chạy dãy quan sát $\mathbf{x}$ qua mô hình này để ước lượng các giá trị kỳ vọng cho mỗi tham số của mô hình và thay đổi các tham số để cực đại hoá các đường đi được sử dụng nhiều.
- Lặp lại quá trình này cho tới khi đạt được giá trị tối ưu cho các tham số của θ.

Learning Algorithm – Baum-Welch Algorithm

Các công thức ước lượng như sau:

$$
\begin{aligned}
\widehat{\pi}_{s_i} &= \text{tần số mô hình ở trạng thái } s_i \text{ tại thời điểm } t = 1 \\
&= \gamma_1(s_i) \\
\widehat{a}_{s_i s_j} &= \frac{\text{số bước chuyển từ } s_i \text{ tới } s_j}{\text{số bước chuyển từ } s_i} \\
&= \frac{\sum_{t=1}^{T} \gamma_t(s_i, s_j)}{\sum_{t=1}^{T} \gamma_t(s_i)} \\
\widehat{b}_{s_i s_j x_k} &= \frac{\text{số bước chuyển từ } s_i \text{ tới } s_j \text{ với quan sát } x_k}{\text{số bước chuyển từ } s_i \text{ tới } s_j} \\
&= \frac{\sum_{\{t: x_t = x_k, 1 \le t \le T\}} \gamma_t(s_i, s_j)}{\sum_{t=1}^{T} \gamma_t(s_i, s_j)}
\end{aligned}
$$

Learning Algorithm – Baum-Welch Algorithm

- Từ mô hình $\theta = (A, B, \pi)$ ta có mô hình mới $\widehat{\theta} = (\widehat{A}, \widehat{B}, \widehat{\pi})$.
- (Baum & Welch, 1970) đã chứng minh rằng

$$P(\mathbf{x}\,|\widehat{\theta}) \geq P(\mathbf{x}\,|\theta).$$

- Do đó mô hình $\widehat{\theta}$ tốt hơn mô hình θ. Đây là một tính chất tổng quát của thuật toán cực đại hoá kỳ vọng.
- Quá trình lặp để tìm mô hình tốt hơn dừng lại khi sai số giữa các mô hình là đủ bé.

Learning Algorithm – Baum-Welch Algorithm

1. Khởi tạo ngẫu nhiên π, A, B sao cho

$$\sum_{s_i} \pi_{s_i} = 1$$

$$\sum_{s_j} a_{s_i s_j} = 1$$

$$\sum_{x_k} b_{s_i}(x_k) = 1, \forall s_i$$

2. Lặp cho tới khi hội tụ:
   - E-step
   - M-step
3. Trả về π, A, B.

Learning Algorithm – Baum-Welch Algorithm

E-step: $\forall s_i, \forall s_j$:

$$P(\mathbf{x}\,|\theta) = \sum_{j=1}^{n} \alpha_{s_j}(T)$$

$$\gamma_t(s_i, s_j) = \frac{\alpha_{s_i}(t) a_{s_i s_j} b_{s_j}(x_{t+1}) \beta_{s_j}(t+1)}{P(\mathbf{x}\,|\theta)}, \quad \forall t = 1, \ldots, T-1$$

$$\gamma_t(s_i) = \frac{\alpha_{s_i}(t)\beta_{s_i}(t)}{P(\mathbf{x}\,|\theta)}, \quad \forall t = 1, \ldots, T.$$

Learning Algorithm – Baum-Welch Algorithm

M-step: $\forall s_i, \forall s_j, \forall x_k \in \Sigma$:

$$\pi_{s_i} = \gamma_1(s_i)$$

$$\widehat{a}_{s_i s_j} = \frac{\sum_{t=1}^{T-1} \gamma_t(s_i, s_j)}{\sum_{t=1}^{T-1} \gamma_t(s_i)}$$

$$\widehat{b}_{s_i}(x_k) = \frac{\sum_{\{t=1,\ldots,T: x_t = x_k\}} \gamma_t(s_i)}{\sum_{t=1}^{T} \gamma_t(s_i)}$$

Content

Exercises

1. Implement the algorithms of HMMs (estimation, decoding, learning)
2. Devise the algorithms of HMMs when the observations are associated to states instead of to transitions
3. Application: Part-of-speech tagging (*to be continued...*)

Observation Model

When observations are associated with states rather than with transitions:

- Hidden states: $z_t \in \{1, 2, \ldots, K\}$.
- Observation model: $p(\mathbf{x}_t \,|z_t)$. The observations in an HMM can be discrete or continuous.
  - If they are discrete, we have an observation matrix $B = (b_{ik})$, where

$$p(x_t = i|z_t = k, \theta) = b_{ki}$$

  - If they are continuous, it is common for the observation model to be a conditional Gaussian:

$$p(x_t|z_t = k, \theta) = \mathcal{N}(x_t|\mu_k, \Sigma_k).$$

Denote the sequence $z_1, z_2, \ldots, z_T$ by $\mathbf{z}_{1:T}$. The corresponding joint distribution is

$$\begin{aligned} p(\mathbf{z}_{1:T}, \mathbf{x}_{1:T}) &= p(\mathbf{z}_{1:T}) p(\mathbf{x}_{1:T} \mid \mathbf{z}_{1:T}) \\ &= \left[p(z_1) \prod_{t=2}^{T} p(z_t | z_{t-1}) \right] \left[\prod_{t=1}^{T} p(x_t | z_t) \right] \end{aligned}$$

Forward Algorithm

Need to compute $p(z_t|\,\mathbf{x}_{1:t})$. We have

$$p(z_t = j|\,\mathbf{x}_{1:t-1}) = \sum_i p(z_t = j|z_{t-1} = i)p(z_{t-1} = i|\,\mathbf{x}_{1:t-1}).$$

Let $\alpha_t(i) = p(z_t = j|\,\mathbf{x}_{1:t})$, we have

$$\begin{aligned}\alpha_t(i) &= p(z_t = j|x_t, \mathbf{x}_{1:t-1}) \\ &= \frac{1}{Z_t} p(x_t|z_t = j, \mathbf{x}_{1:t-1}) p(z_t = j|\,\mathbf{x}_{1:t-1}) \\ &= \frac{1}{Z_t} p(x_t|z_t = j) p(z_t = j|\,\mathbf{x}_{1:t-1})\end{aligned}$$

where the normalization constant is given by

$$Z_t = p(x_t|\,\mathbf{x}_{1:t-1}) = \sum_j p(z_t = j|\,\mathbf{x}_{1:t-1}) p(x_t|z_t = j).$$

Forward Algorithm

In summary:

$$\alpha_t(i) \propto p(x_t|z_t = j) \left[\sum_i p(z_t = j|z_{t-1} = i) p(z_{t-1} = i|\, \mathbf{x}_{1:t-1}) \right]$$

$$\propto b_{jx_t} \left[\sum_i a_{ij} \alpha_{t-1}(i) \right].$$

The distribution $p(z_t|\, \mathbf{x}_{1:t})$ is called the **belief state** at time t, and is a vector of K numbers:

$$\alpha_t = \begin{pmatrix} \alpha_t(1) \\ \alpha_t(2) \\ \vdots \\ \alpha_t(K) \end{pmatrix}$$

Forward Algorithm

In matrix notation:

$$\alpha_t \propto \psi_t \bullet (A^T \alpha_{t-1}),$$

where $\psi_t(j) = p(x_t|z_t = j)$ is the local evidence at time t.