

عنوان: مدل‌های پنهان مارکوف

مقدمه‌ای بر مدل‌های احتمالی در داده‌های زمانی

کاربردها در یادگیری ماشین و پردازش زبان

ارائه دهنده: دکتر عاطفه پاکزاد

اردیبهشت ۱۴۰۵

انگیزه مسئله

در بسیاری از مسائل واقعی:

- حالت واقعی سیستم قابل مشاهده نیست.
- فقط نشانه‌هایی از آن را مشاهده می‌کنیم.

نمونه‌ها:

- تشخیص گفتار
- تحلیل احساسات
- تحلیل بازار مالی
- بیوانفورماتیک

سؤال کلیدی:

چگونه می‌توان وضعیت پنهان سیستم را از روی مشاهدات تخمین زد؟

مثال شهودی (هوا و چتر)

فرض کنید هوا را مستقیم نمی‌بینیم.

اما مشاهده می‌کنیم:

مردم چتر دارند یا ندارند.

پس:

حالت واقعی (Hidden State)

Rainy / Sunny

مشاهده (Observation)

Umbrella / No Umbrella

هدف:

حدس زدن وضعیت هوا از روی چتر مردم

مدل‌های زمانی

در بسیاری از داده‌ها ترتیب زمانی مهم است.

نمونه‌ها:

- گفتار
- متن
- قیمت سهام
- سیگنال‌های زیستی

مدل‌های زمانی باید بتوانند:

- وابستگی زمانی را مدل کنند.
- تغییر حالت در طول زمان را توضیح دهند.

زنجیره مارکوف

یک زنجیره مارکوف مدلی برای تغییر حالت‌ها در زمان است.

خاصیت مارکوف:

احتمال حالت آینده فقط به حالت فعلی وابسته است.

فرمول:

$$P(X_t | X_{t-1}, X_{t-2}, \dots, X_1) = P(X_t | X_{t-1})$$

یعنی گذشته دور اهمیتی ندارد.

فرض کنید می‌خواهیم وضعیت هوا در طول روزها را مدل کنیم.

دو حالت داریم:

- Sunny (آفتابی)
- Rainy (بارانی)

در هر روز، هوا یکی از این دو حالت است.

ویژگی مهم مارکوف:

وضعیت فردا فقط به وضعیت امروز بستگی دارد، نه به کل گذشته.

از نظر ریاضی:

$$P(X_{t+1} | X_t, X_{t-1}, \dots, X_1) = P(X_{t+1} | X_t)$$

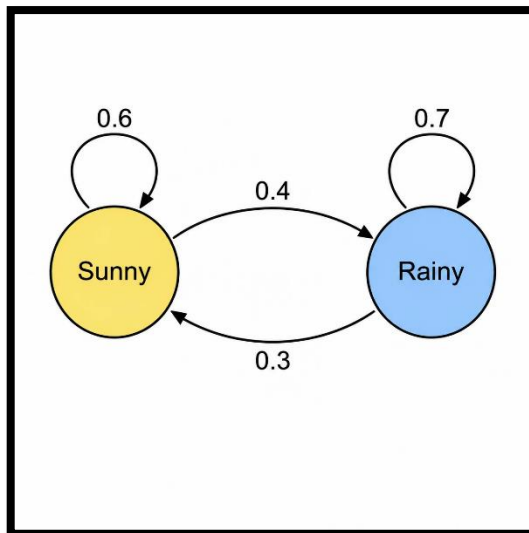
یعنی اگر بدانیم امروز چه وضعیتی است، دانستن روزهای قبل اطلاعات جدیدی نمی‌دهد.

نمایش گرافی (Graph Representation)

زنجیره مارکوف را می‌توان به صورت یک گراف جهت‌دار نشان داد.

در این گراف:

- هر گره (node) یک حالت است
- هر یال (edge) یک انتقال بین حالت‌هاست
- روی هر یال یک احتمال انتقال نوشته می‌شود.



برای مثال ما:

گره‌ها:

- Sunny •
- Rainy •

یال‌ها:

- Sunny → Sunny •
- Sunny → Rainy •
- Rainy → Rainy •
- Rainy → Sunny •

به صورت مفهومی:

Sunny → Sunny

Sunny → Rainy

Rainy → Rainy

Rainy → Sunny

یعنی از هر حالت می‌توان به حالت‌های دیگر (یا خودش) رفت.

۳. معنی احتمال انتقال

هر یال یک احتمال دارد.

مثلاً:

اگر امروز Sunny باشد:

- احتمال Sunny شدن فردا = ۰.۶
- احتمال Rainy شدن فردا = ۰.۴

اگر امروز Rainy باشد:

- احتمال Rainy بودن فردا = ۰.۷
- احتمال Sunny شدن فردا = ۰.۳

این اعداد از داده‌های واقعی یا تخمین آماری به دست می‌آیند.

نکته مهم:

مجموع احتمال‌های خروجی از هر حالت باید ۱ باشد.

مثلاً برای Sunny:

$$0.6 + 0.4 = 1$$

۴. ماتریس انتقال (Transition Matrix)

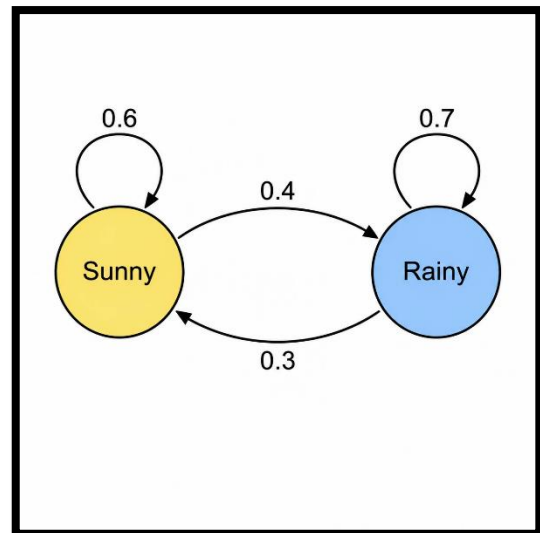
برای اینکه این احتمالات را منظم بنویسیم از **ماتریس انتقال** استفاده می‌کنیم.

اگر دو حالت داشته باشیم:

Sunny

Rainy

ماتریس انتقال به شکل زیر نوشته می‌شود:



$$A = \begin{bmatrix} P(S \rightarrow S) & P(S \rightarrow R) \\ P(R \rightarrow S) & P(R \rightarrow R) \end{bmatrix}$$

مثلاً:

$$A = \begin{bmatrix} 0.6 & 0.4 \\ 0.3 & 0.7 \end{bmatrix}$$

تفسیر سطرها:

سطر اول: اگر امروز Sunny باشد

سطر دوم: اگر امروز Rainy باشد

تفسیر ستونها:

ستون اول: حالت بعدی Sunny

ستون دوم: حالت بعدی Rainy

پس:

- Sunny $\rightarrow$ Sunny یعنی 0.6
- Sunny $\rightarrow$ Rainy یعنی 0.4
- Rainy $\rightarrow$ Sunny یعنی 0.3
- Rainy $\rightarrow$ Rainy یعنی 0.7

۵. مثال از تغییرات در زمان

فرض کنید امروز Sunny است.

می‌خواهیم احتمال هوا در روز بعد را بدانیم.

از ماتریس:

$$A = \begin{bmatrix} 0.6 & 0.4 \\ 0.3 & 0.7 \end{bmatrix}$$

$$\text{Sunny} \rightarrow \text{Sunny} = 0.6$$

$$\text{Sunny} \rightarrow \text{Rainy} = 0.4$$

پس:

• احتمال فردا Sunny = 0.6

• احتمال فردا Rainy = 0.4

اگر فردا Rainy شد، برای روز بعد:

$$\text{Rainy} \rightarrow \text{Rainy} = 0.7$$

$$\text{Rainy} \rightarrow \text{Sunny} = 0.3$$

بنابراین می‌توانیم تکامل سیستم در زمان را مدل کنیم.

۶. ارتباط گراف و ماتریس انتقال

در واقع:

- گراف نمایش بصری مدل است.
- ماتریس انتقال نمایش ریاضی همان گراف است.

هر یال در گراف $\leftrightarrow$ یک مقدار در ماتریس.

مثلاً:

$$\text{Sunny} \rightarrow \text{Rainy}$$

در ماتریس همان مقدار:

$$a_{SR}$$

است.

۷. فرض مهم در زنجیره مارکوف ساده

در این مدل فرض می‌کنیم:

حالت‌ها قابل مشاهده هستند.

یعنی:

ما مستقیم می‌دانیم:

روز ۱ $\rightarrow$ Sunny

روز ۲ $\rightarrow$ Rainy

روز ۳ $\rightarrow$ Sunny

پس توالی حالت‌ها را می‌بینیم.

این تفاوت اصلی با **Hidden Markov Model** است.

در HMM:

- حالت‌ها پنهان هستند
- فقط نشانه‌هایی از آن‌ها را می‌بینیم

مثال:

حالت واقعی: Weather

مشاهده: Umbrella

۸. خلاصه شهودی

در زنجیره مارکوف:

- گره‌ها = حالت‌ها
- یال‌ها = انتقال بین حالت‌ها
- وزن یال = احتمال انتقال
- ماتریس انتقال = نمایش ریاضی گراف

و فرض کلیدی:

سیستم فقط به حالت فعلی وابسته است.

یک مثال کاملاً عددی و ساده بسازیم:

۱. تعریف مدل زنجیره مارکوف هوا

دو حالت داریم:

- S : Sunny (آفتابی)
- R : Rainy (بارانی)

ماتریس انتقال را این طور فرض می کنیم:

$$A = \begin{bmatrix} 0.8 & 0.2 \\ 0.3 & 0.7 \end{bmatrix}$$

که ترتیب حالت ها را این طور در نظر می گیریم: $[S, R]$

بنابراین:

- $P(S \rightarrow S) = 0.8$
- $P(S \rightarrow R) = 0.2$
- $P(R \rightarrow S) = 0.3$
- $P(R \rightarrow R) = 0.7$

سطر = «امروز»، ستون = «فردا».

۲. وضعیت اولیه (روز صفر)

فرض کن امروز قطعاً هوا آفتابی است.

این را به صورت یک بردار احتمال می نویسیم:

$$\pi^{(0)} = [1 \quad 0]$$

یعنی:

- احتمال $Sunny = 1$
- احتمال $Rainy = 0$

۳. احتمال هوا در روز بعد (۱ روز بعد)

برای به دست آوردن توزیع احتمال هوا در روز ۱، کافی است بردار اولیه را در ماتریس انتقال ضرب کنیم:

$$\pi^{(1)} = \pi^{(0)} A$$

حساب کنیم:

$$\pi^{(1)} = [1 \quad 0] \begin{bmatrix} 0.8 & 0.2 \\ 0.3 & 0.7 \end{bmatrix} = [1 \cdot 0.8 + 0 \cdot 0.3, \quad 1 \cdot 0.2 + 0 \cdot 0.7] = [0.8 \quad 0.2]$$

تفسیر:

- احتمال آفتابی بودن فردا = ۰.۸
- احتمال بارانی بودن فردا = ۰.۲

این دقیقاً همان چیزی است که در ماتریس سطر Sunny نوشته بودیم (طبیعی است، چون امروز حتماً Sunny بوده).

۴. احتمال هوا در دو روز بعد

حالا می‌خواهیم ببینیم دو روز بعد (روز ۲) هوا با چه احتمالی آفتابی یا بارانی است.

باز هم ضرب ماتریسی:

$$\pi^{(2)} = \pi^{(1)} A$$

جایگذاری:

$$\pi^{(2)} = [0.8 \quad 0.2] \begin{bmatrix} 0.8 & 0.2 \\ 0.3 & 0.7 \end{bmatrix}$$

الان باید سطر در ستون ضرب کنیم:

مؤلفه‌ی اول (احتمال Sunny در روز ۲):

$$P(X_2 = S) = 0.8 \cdot 0.8 + 0.2 \cdot 0.3 = 0.64 + 0.06 = 0.70$$

مؤلفه‌ی دوم (احتمال Rainy در روز ۲):

$$P(X_2 = R) = 0.8 \cdot 0.2 + 0.2 \cdot 0.7 = 0.16 + 0.14 = 0.30$$

پس:

$$\pi^{(2)} = [0.70 \quad 0.30]$$

یعنی:

- احتمال آفتابی بودن دو روز بعد = ۰.۷۰
- احتمال بارانی بودن دو روز بعد = ۰.۳۰

نکته‌ی آموزشی خوب:

«جمع این دو تا چقدر شد؟»

جواب: $0.70 + 0.30 = 1 \rightarrow$ تأکید روی اینکه هنوز با یک توزیع احتمال طرفیم.

۵. احتمال هوا در سه روز بعد

همین روال را یک بار دیگر ادامه می‌دهیم:

$$\pi^{(3)} = \pi^{(2)} A$$

جایگذاری:

$$\pi^{(3)} = [0.70 \quad 0.30] \begin{bmatrix} 0.8 & 0.2 \\ 0.3 & 0.7 \end{bmatrix}$$

مؤلفه‌ی اول (Sunny) در روز ۳:

$$P(X_3 = S) = 0.70 \cdot 0.8 + 0.30 \cdot 0.3 = 0.56 + 0.09 = 0.65$$

مؤلفه‌ی دوم (Rainy) در روز ۳:

$$P(X_3 = R) = 0.70 \cdot 0.2 + 0.30 \cdot 0.7 = 0.14 + 0.21 = 0.35$$

پس:

$$\pi^{(3)} = [0.65 \quad 0.35]$$

تفسیر:

- اگر امروز حتماً آفتابی باشد، سه روز بعد:
 - با احتمال ۰.۶۵ هوا آفتابی است
 - با احتمال ۰.۳۵ هوا بارانی است

باز هم جمع ۱ است:

$$0.65 + 0.35 = 1$$

.....

محدودیت زنجیره مارکوف

در بسیاری از مسائل واقعی:

حالت‌ها قابل مشاهده نیستند.

مثال:

در تشخیص گفتار

حالت واقعی = کلمه

اما مشاهده = سیگنال صوتی

پس نیاز داریم به:

مدلی که حالت‌ها پنهان باشند.

معرفی Hidden Markov Model

Hidden Markov Model مدلی است که:

- حالت‌های واقعی پنهان هستند
- فقط خروجی‌های سیستم را مشاهده می‌کنیم
- حالت‌ها با خاصیت مارکوف تغییر می‌کنند

ساختار HMM

دو لایه در مدل وجود دارد:

لایه بالا: حالت‌های پنهان

$$X_1 \rightarrow X_2 \rightarrow X_3 \rightarrow \dots \rightarrow X_T$$

لایه پایین: مشاهدات

$$O_1 O_2 O_3 \dots O_T$$

هر مشاهده فقط به حالت همان زمان وابسته است.

مدل پنهان مارکوف (Hidden Markov Model)

مدل HMM یک مدل آماری برای داده‌های دنباله‌ای (sequence data) است که در آن فرض می‌کنیم:

- یک فرآیند پنهان وجود دارد که در طول زمان بین حالت‌ها جابه‌جا می‌شود.
- ما خودِ حالت‌ها را نمی‌بینیم.
- فقط مشاهده‌هایی را می‌بینیم که توسط آن حالت‌ها تولید شده‌اند.

بنابراین در HMM دو لایه وجود دارد:

۱. لایه حالت‌های پنهان (Hidden States)

۲. لایه مشاهده‌ها (Observations)

در هر زمان t :

- سیستم در یک حالت پنهان q_t قرار دارد.
- از آن حالت یک مشاهده O_t تولید می‌شود.

اجزای اصلی مدل HMM

- یک HMM با سه مجموعه پارامتر تعریف می‌شود:

- $\lambda = (A, B, \pi)$

ماتریس انتقال حالت‌ها (Transition Matrix)

$$A = [a_{ij}]$$

که در آن:

$$a_{ij} = P(q_{t+1} = j \mid q_t = i)$$

یعنی:

احتمال اینکه اگر در زمان t در حالت i باشیم، در زمان بعدی به حالت j برویم.

ویژگی‌ها:

- هر سطر مجموعه‌ش ۱ است.
- این همان قانون مارکوف را پیاده می‌کند.

مثال:

$$A = \begin{bmatrix} 0.7 & 0.3 \\ 0.4 & 0.6 \end{bmatrix}$$

ماتریس احتمال مشاهده (Emission Matrix)

$$B = [b_j(k)]$$

که در آن:

$$b_j(k) = P(o_t = v_k \mid q_t = j)$$

یعنی:

اگر سیستم در حالت j باشد، احتمال اینکه مشاهده v_k تولید شود چقدر است.

به عبارت دیگر:

هر حالت پنهان یک توزیع احتمال روی مشاهده‌ها دارد.

توزیع اولیه حالت‌ها

$$\pi = [\pi_i]$$

که در آن:

$$\pi_i = P(q_1 = i)$$

یعنی:

احتمال اینکه سیستم در ابتدای دنباله در حالت i شروع شود.

مثلاً:

$$\pi = [0.6 \quad 0.4]$$

ساختار گراف HMM

در HMM دو نوع یال داریم:

□ انتقال حالت‌ها

بین حالت‌ها:

$$q_t \rightarrow q_{t+1}$$

با احتمال‌های ماتریس A

□ تولید مشاهده

از حالت به مشاهده:

$$q_t \rightarrow o_t$$

با احتمال‌های ماتریس B

فرض مارکوف در HMM

دو فرض مهم وجود دارد:

فرض اول

$$P(q_{t+1} | q_t, q_{t-1}, \dots) = P(q_{t+1} | q_t)$$

یعنی حالت بعدی فقط به حالت فعلی بستگی دارد.

فرض دوم

$$P(o_t | q_t, q_{t-1}, \dots) = P(o_t | q_t)$$

یعنی مشاهده فقط به حالت فعلی بستگی دارد.

مثال ساده HMM

یک مثال کلاسیک برای تدریس: تشخیص آب و هوا از روی چتر

فرض کنید:

ما هوا را نمی بینیم، اما می بینیم که یک نفر چتر دارد یا ندارد.

پس:

حالت های پنهان

- Sunny
- Rainy

مشاهده ها

- Umbrella
- No Umbrella

۱. ماتریس انتقال

$$A = \begin{bmatrix} 0.8 & 0.2 \\ 0.3 & 0.7 \end{bmatrix}$$

تفسیر:

اگر امروز Sunny باشد:

- فردا Sunny با احتمال ۰.۸
- فردا Rainy با احتمال ۰.۲

اگر امروز Rainy باشد:

- فردا Sunny با احتمال ۰.۳
- فردا Rainy با احتمال ۰.۷

ماتریس مشاهده

$$B = \begin{bmatrix} 0.9 & 0.1 \\ 0.2 & 0.8 \end{bmatrix}$$

ترتیب مشاهده‌ها:

(No Umbrella , Umbrella)

تفسیر:

اگر هوا Sunny باشد:

- احتمال دیدن چتر = ۰.۱
- احتمال بدون چتر = ۰.۹

اگر هوا Rainy باشد:

- احتمال دیدن چتر = ۰.۸
- احتمال بدون چتر = ۰.۲

توزیع اولیه

$$\pi = [0.6 \quad 0.4]$$

یعنی:

- احتمال شروع با Sunny = 0.6
- احتمال شروع با Rainy = 0.4

مثال یک دنباله مشاهده

فرض کنید مشاهده‌های ما این باشد:

Umbrella , Umbrella , No Umbrella

یعنی در سه روز:

روز ۱ → چتر

روز ۲ → چتر

روز ۳ → بدون چتر

سوال مهم:

حالت‌های واقعی هوا چه بوده‌اند؟

مثلاً ممکن است:

Rainy Rainy Sunny

یا

Sunny Rainy Sunny

و غیره.

الگوریتم‌هایی مثل:

- **Forward algorithm** → محاسبه احتمال دنباله مشاهده
- **Viterbi algorithm** → پیدا کردن محتمل‌ترین توالی حالت‌ها
- **Baum–Welch** → آموزش پارامترها

برای حل این مسائل استفاده می‌شوند.

کد عملی Forward Algorithm

این الگوریتم احتمال دنباله مشاهده را حساب می‌کند.



```
import numpy as np

# States: 0 = Sunny, 1 = Rainy
# Observations: 0 = Umbrella, 1 = No Umbrella

states = ["Sunny", "Rainy"]
observations = ["Umbrella", "No Umbrella"]

A = np.array([
    [0.8, 0.2],
    [0.3, 0.7]
])

B = np.array([
    [0.1, 0.9],
    [0.8, 0.2]
])

pi = np.array([0.6, 0.4])

# Observation sequence: Umbrella, Umbrella, No Umbrella
O = [0, 0, 1]
```

```
def forward_algorithm(A, B, pi, O):
    N = A.shape[0]      # number of states
    T = len(O)          # length of observation sequence

    alpha = np.zeros((T, N))

    # Initialization
    for i in range(N):
        alpha[0, i] = pi[i] * B[i, O[0]]

    # Recursion
    for t in range(1, T):
        for j in range(N):
            alpha[t, j] = np.sum(alpha[t-1, :] * A[:, j]) * B[j, O[t]]

    # Termination
    prob = np.sum(alpha[T-1, :])
    return alpha, prob

alpha, prob = forward_algorithm(A, B, pi, O)

print("Forward table (alpha):")
print(alpha)
print("\nProbability of observation sequence:", prob)
```

```
*** Forward table (alpha):
[[0.06    0.32   ]
 [0.0144  0.1888 ]
 [0.061344 0.027008]]

Probability of observation sequence: 0.08835200000000001
```

Forward Algorithm

هدف این الگوریتم این است که بفهمیم:

$$P(O \mid \lambda)$$

یعنی احتمال اینکه این دنباله مشاهده تولید شود چقدر است.

مثال ما:

Umbrella, Umbrella, No Umbrella

```
import numpy as np
```

کتابخانه NumPy برای کار با آرایه‌ها و ماتریس‌ها استفاده می‌شود.

چون HMM پر از ماتریس است، تقریباً همیشه از NumPy استفاده می‌کنیم.

```
# States: Sunny=0, Rainy=1
# Observations: Umbrella=0, No Umbrella=1
```

کامپیوتر با متن کار نمی‌کند، بنابراین ما حالت‌ها را عددگذاری می‌کنیم.

حالت‌ها:

```
Sunny = 0
Rainy = 1
```

مشاهده‌ها:

```
Umbrella = 0
No Umbrella = 1
```

```
states = ["Sunny", "Rainy"]
observations = ["Umbrella", "No Umbrella"]
```

این فقط برای خوانایی خروجی است تا بعداً بدانیم هر عدد به چه حالتی مربوط است.

```
A = np.array([
    [0.8, 0.2],
    [0.3, 0.7]
])
```

توضیح:

این ماتریس انتقال حالت‌ها (Transition Matrix) است.

معنی آن:

Sunny → Sunny = 0.8

Sunny → Rainy = 0.2

Rainy → Sunny = 0.3

Rainy → Rainy = 0.7

یعنی اگر امروز Sunny باشد:

80% احتمال دارد فردا Sunny باشد.

20% احتمال دارد Rainy شود.

```
B = np.array([
    [0.1, 0.9],
```

```
[0.8, 0.2]
])
```

این ماتریس مشاهده (Emission Matrix) است.

اگر حالت **Sunny** باشد:

```
Umbrella = 0.1
No Umbrella = 0.9
```

اگر حالت **Rainy** باشد:

```
Umbrella = 0.8
No Umbrella = 0.2
```

یعنی وقتی باران می بارد احتمال دیدن چتر بیشتر است.

```
pi = np.array([0.6, 0.4])
```

این توزیع اولیه حالت ها است.

در روز اول:

```
P(Sunny) = 0.6
P(Rainy) = 0.4
```

```
O = [0, 0, 1]
```

دنباله مشاهده ما.

```
0 → Umbrella
0 → Umbrella
1 → No Umbrella
```

پس دنباله:

```
Umbrella, Umbrella, No Umbrella
```

تابع Forward

```
def forward_algorithm(A, B, pi, O):
```

یک تابع تعریف می کنیم که الگوریتم forward را اجرا کند.

ورودی ها:

• $A \rightarrow$ ماتریس انتقال

- $B \rightarrow$ ماتریس مشاهده
- $\pi \rightarrow$ احتمال اولیه
- $O \rightarrow$ دنباله مشاهده

```
N = A.shape[0]
```

تعداد **حالت‌ها** را می‌گیریم.

در مثال ما:

```
N = 2
```

```
T = len(O)
```

طول دنباله مشاهده.

در مثال ما:

```
T = 3
```

```
alpha = np.zeros((T, N))
```

ماتریس **alpha** ساخته می‌شود.

این ماتریس هسته الگوریتم **forward** است.

ساختار آن:

زمان $\times$ حالت

در مثال:

3×2

یعنی:

t	Sunny	Rainy
۱		
۲		
۳		

مرحله Initialization

```
for i in range(N):
    alpha[0, i] = pi[i] * B[i, O[0]]
```

برای زمان اول فرمول:

$$\alpha_1(i) = \pi_i \times b_i(O_1)$$

یعنی:

احتمال شروع در حالت i
 $\times$
احتمال تولید مشاهده اول

مثلاً برای Sunny:

$$0.6 \times 0.1$$

مرحله Recursion

```
for t in range(1, T):
```

برای هر زمان بعدی (روز دوم و سوم).

```
for j in range(N):
```

برای هر حالت ممکن در زمان t .

```
alpha[t, j] = np.sum(alpha[t-1, :] * A[:, j]) * B[j, O[t]]
```

این مهم‌ترین خط الگوریتم است.

فرمول آن:

$$\alpha_t(j) = \left(\sum_i \alpha_{t-1}(i) a_{ij} \right) \times b_j(O_t)$$

- از همه حالت‌های قبلی می‌توانیم به j برسیم.
- احتمال آن‌ها را جمع می‌کنیم.
- احتمال تولید مشاهده را ضرب می‌کنیم.

مرحله Termination

```
prob = np.sum(alpha[T-1, :])
```

در آخرین زمان همه احتمال ها جمع می شوند:

$$P(O) = \sum_i \alpha_T(i)$$

این می شود احتمال کل دنباله مشاهده

```
return alpha, prob
```

خروجی:

- جدول alpha
- احتمال نهایی

اجرای الگوریتم

```
alpha, prob = forward_algorithm(A, B, pi, O)
```

تابع اجرا می شود.

```
print("Forward table (alpha):")
print(alpha)
```

جدول alpha چاپ می شود.

```
print("\nProbability of observation sequence:", prob)
```

احتمال کل دنباله چاپ می شود.

نتیجه

تقریباً می شود:

$P(O) \approx 0.03628$

یعنی:

احتمال دیدن Umbrella, Umbrella, No Umbrella حدود ۳.۶ درصد است.