Tuần 2 - Tổng hợp kiến thức Buổi học số 4

Time-Series Team

Ngày 16 tháng 7 năm 2025

Buổi học số 4 (Thứ 6, 11/07/2025) nhóm mình sẽ tổng hợp lại thành 4 nội dung chính:

- ***Phần 1: Bernoulli Naïve Bayes Classifier***
- ***Phần 2: Gaussian Function***
- ***Phần 3: Gaussian Naïve Bayes Classifier***
- ***Phần 4: Mở rộng: Word segmentation trong bài toán phân loại email spam***

Phần 1: Bernoulli Naïve Bayes Classifier

1. Review

1.1. Independent Events (Các biến cố độc lập)

Khái niệm

Hai biến cố A và B được gọi là **độc lập** nếu và chỉ nếu:

$$P(A \cap B) = P(A) \cdot P(B)$$

Nếu điều kiện này không thỏa mãn, tức là:

$$P(A \cap B) \neq P(A) \cdot P(B)$$

thì hai biến cố **phụ thuộc (dependent)**.

Ví dụ minh họa

Một con xúc xắc công bằng được tung một lần. Gọi:

$$A = \{4\}, \qquad B = \{2, 4, 6\}$$

Khi đó:

$$P(A) = \frac{1}{6}, \quad P(B) = \frac{3}{6} = \frac{1}{2}, \quad P(A \cap B) = P(\{4\}) = \frac{1}{6}$$

Kiểm tra:

$$P(A)P(B) = \frac{1}{6} \cdot \frac{1}{2} = \frac{1}{12} \neq P(A \cap B)$$

Do đó, A và B **không độc lập**.

1.2. Conditional Independence (Độc lập có điều kiện)

Khái niệm

Hai biến cố A và B được gọi là **độc lập có điều kiện** khi biết C nếu và chỉ nếu:

$$P(A \cap B \mid C) = P(A \mid C) \cdot P(B \mid C), \qquad P(C) > 0$$

Hiểu đơn giản: Khi **đã biết chắc** rằng C xảy ra, việc biết A xảy ra hay không **không làm thay đổi** xác suất xảy ra của B (và ngược lại).

Phân biệt với độc lập thông thường

- **Độc lập toàn cục (Global Independence):**

$$P(A \cap B) = P(A) \cdot P(B)$$

Hai biến cố A và B độc lập trong toàn bộ không gian mẫu.

- **Độc lập có điều kiện (Conditional Independence):** Độc lập chỉ xảy ra trong một không gian con C. Tức là **chỉ khi biết** C, A và B mới không ảnh hưởng lẫn nhau.

Điểm quan trọng:

1. Hai biến cố A và B có thể **độc lập toàn cục**, nhưng khi giới hạn trong C thì trở thành **phụ thuộc có điều kiện**.
2. Ngược lại, hai biến cố A và B có thể **phụ thuộc toàn cục**, nhưng trở thành **độc lập khi biết** C (vì C giải thích toàn bộ nguyên nhân phụ thuộc).

Ví dụ trực quan bằng sơ đồ Venn

Giả sử:

- A và B độc lập toàn cục, nên chúng **có phần giao nhau** (vì $P(A), P(B) > 0$).
- Thêm một biến cố C sao cho:

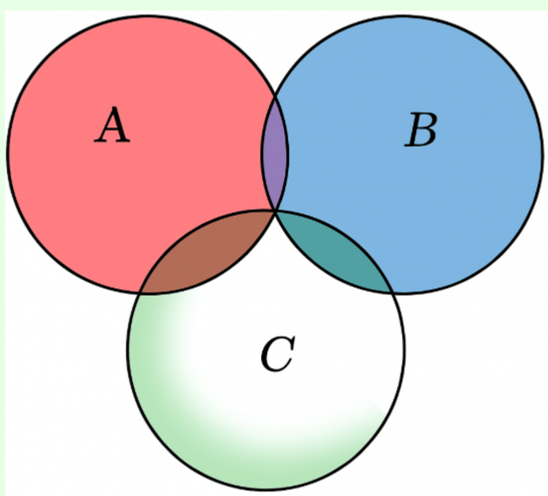
$$A \cap B \neq \emptyset, \quad A \cap C \neq \emptyset, \quad B \cap C \neq \emptyset, \quad (A \cap B) \cap C = \emptyset$$

Khi giới hạn trong C:

$$P(A \cap B \mid C) = 0 \neq P(A \mid C)P(B \mid C)$$

$\Rightarrow$ A và B **trở thành phụ thuộc có điều kiện**.

Sơ đồ Venn:



1.3. Total Probability Theorem (Định lý xác suất toàn phần)

Phát biểu

Giả sử $R_1, R_2, \ldots, R_n$ là **một hệ đầy đủ các biến cố** (pairwise disjoint, $\bigcup_{i=1}^{n} R_i = \Omega$). Với mọi biến cố S:

$$P(S) = \sum_{i=1}^{n} P(S \cap R_i) = \sum_{i=1}^{n} P(R_i)\, P(S \mid R_i)$$

Giải thích

- Ta chia không gian mẫu thành n "kịch bản" R_i **không chồng chéo**.
- Xác suất S xảy ra chính là tổng xác suất S xảy ra trong từng kịch bản R_i.

Ví dụ

Bài toán: Một nhà máy có 2 dây chuyền:

- Dây chuyền R_1: sản xuất 60% sản phẩm, xác suất lỗi $P(S \mid R_1) = 2\%$.
- Dây chuyền R_2: sản xuất 40% sản phẩm, xác suất lỗi $P(S \mid R_2) = 5\%$.

Tính: Xác suất một sản phẩm bất kỳ bị lỗi.

Lời giải:

$$P(S) = P(R_1)P(S \mid R_1) + P(R_2)P(S \mid R_2) = 0.6 \times 0.02 + 0.4 \times 0.05 = 0.032$$

$\Rightarrow$ Có 3.2% sản phẩm bị lỗi.

1.4. Bayes' Rule (Định lý Bayes)

Phát biểu

Giả sử $C_1, C_2, \ldots, C_n$ là **một hệ đầy đủ các biến cố** và X là biến cố bất kỳ với $P(X) \neq 0$:

$$P(C_i \mid X) = \frac{P(C_i)P(X \mid C_i)}{\sum_{j=1}^{n} P(C_j)P(X \mid C_j)}, \quad i = 1, 2, \ldots, n$$

Giải thích

- Ta muốn **cập nhật niềm tin** về việc C_i xảy ra khi đã quan sát X.
- Tử số: "Độ tin cậy ban đầu về C_i" $\times$ "Xác suất quan sát X nếu C_i đúng".
- Mẫu số: Tổng xác suất quan sát X trong tất cả các kịch bản.

Ví dụ

Bài toán: Tiếp nối ví dụ trên. Một sản phẩm bị lỗi. Hỏi nó đến từ dây chuyền nào?

Tính: Xác suất sản phẩm lỗi đến từ dây chuyền R_2:

$$P(R_2 \mid S) = \frac{P(R_2)P(S \mid R_2)}{P(R_1)P(S \mid R_1) + P(R_2)P(S \mid R_2)} = \frac{0.4 \times 0.05}{0.032} \approx 0.625$$

Kết luận: Nếu một sản phẩm bị lỗi, khả năng nó đến từ dây chuyền R_2 là 62.5%.

2. Giới thiệu về Naïve Bayes Classifier

2.1. Naïve Bayes là gì?

Khái niệm

Naïve Bayes Classifier (NBC) là một thuật toán phân loại dựa trên **Định lý Bayes**, với giả định **các feature độc lập có điều kiện** cho trước class.
Giả sử có:

- Một biến ngẫu nhiên C biểu diễn **class/label** (ví dụ: Pass/Fail, Spam/Not Spam).
- n feature $F_1, F_2, \ldots, F_n$ (các đặc trưng quan sát được).

Mục tiêu: Dự đoán class C cho một tập feature quan sát được $f_1, f_2, \ldots, f_n$.

2.2. Công thức tổng quát

Theo định lý Bayes:

$$P(C \mid F_1, F_2, \ldots, F_n) = \frac{P(F_1, F_2, \ldots, F_n \mid C)\, P(C)}{P(F_1, F_2, \ldots, F_n)}$$

Vì $P(F_1, F_2, \ldots, F_n)$ là hằng số khi so sánh các class, ta chỉ cần tối đa hóa:

$$\hat{C} = \arg\max_C P(C)\, P(F_1, F_2, \ldots, F_n \mid C)$$

Giả định "Naïve":

$$P(F_1, F_2, \ldots, F_n \mid C) = \prod_{i=1}^{n} P(F_i \mid C)$$

$\Rightarrow$ Công thức rút gọn:

$$\hat{C} = \arg\max_C P(C) \prod_{i=1}^{n} P(F_i \mid C)$$

2.3. Các biến thể của Naïve Bayes

Tuỳ loại dữ liệu đầu vào, ta có các biến thể:

- **Bernoulli Naïve Bayes**: Áp dụng cho dữ liệu nhị phân (Yes/No, True/False, 0/1).
- **Multinomial Naïve Bayes**: Áp dụng cho dữ liệu đếm (số lần xuất hiện từ trong văn bản).
- **Gaussian Naïve Bayes**: Áp dụng cho dữ liệu liên tục (giả định phân phối chuẩn).

2.4. Ưu và nhược điểm

Ưu điểm:

- Đơn giản, dễ hiểu, dễ cài đặt.
- Hiệu quả ngay cả với tập dữ liệu nhỏ.
- Tính toán nhanh nhờ giả định độc lập.

Nhược điểm:

- Giả định độc lập có điều kiện thường không đúng trong thực tế.
- Độ chính xác không cao nếu các feature phụ thuộc mạnh lẫn nhau.

3. Bernoulli Naïve Bayes Classifier

3.1. Giới thiệu

Khái niệm

Bernoulli Naïve Bayes (BNB) là một biến thể của Naïve Bayes, chuyên dùng cho dữ liệu nhị phân (**binary/Boolean data**).
Đặc điểm:

- Mỗi feature chỉ có hai giá trị: **Yes/No**, **True/False** hoặc **0/1**.
- Phân phối xác suất của feature được mô hình hoá bằng **phân phối Bernoulli**.
- Phù hợp khi dữ liệu chỉ biểu diễn *sự có mặt hay vắng mặt* của một đặc trưng.

3.2. Công thức tổng quát

Bài toán: Dự đoán class C khi biết n feature nhị phân $F_1, F_2, \ldots, F_n$.
Theo Bayes:

$$P(C \mid F_1, F_2, \ldots, F_n) = \frac{P(F_1, F_2, \ldots, F_n \mid C)P(C)}{P(F_1, F_2, \ldots, F_n)}$$

Giả định Naïve (độc lập có điều kiện):

$$P(F_1, F_2, \ldots, F_n \mid C) = \prod_{i=1}^{n} P(F_i \mid C)$$

Ước lượng Bernoulli cho mỗi feature:

$$P(F_i \mid C) = \begin{cases} \theta_{i|C}, & \text{nếu feature xuất hiện (Yes/1)} \\ 1 - \theta_{i|C}, & \text{nếu feature không xuất hiện (No/0)} \end{cases}$$

với:

$$\theta_{i|C} = \frac{\text{số mẫu trong class } C \text{ có feature } i \text{ xuất hiện}}{\text{tổng số mẫu thuộc class } C}$$

Quy tắc quyết định:

$$\hat{C} = \arg\max_{C} P(C) \prod_{i=1}^{n} P(F_i \mid C)$$

3.3. One Feature

Phát biểu

Giả sử ta có hai lớp (**class/label**):

- $R = \text{Pass}$
- $R = \text{Fail}$

và một **feature nhị phân** (*Bernoulli variable*):

- $S = \text{Studied (Yes/No)}$

Theo định lý Bayes:

$$P(R \mid S) = \frac{P(S \mid R) P(R)}{P(S)}$$

Trong đó:

$$P(S) = P(S \mid R = \text{Pass}) P(R = \text{Pass}) + P(S \mid R = \text{Fail}) P(R = \text{Fail})$$

Ví dụ minh họa: One Feature

Xét 6 sinh viên với kết quả thi được ghi nhận như bảng sau:

Studied	**Result**
No	Fail
No	Pass
Yes	Fail
Yes	Pass
Yes	Pass
No	Fail

Gọi:

- R là biến ngẫu nhiên đại diện cho kết quả (**Result**):

$$R_1 = \text{Pass}, \quad R_2 = \text{Fail}$$

- S là biến ngẫu nhiên đại diện cho việc học bài (**Studied**):

$$S = \text{Yes hoặc No}$$

Các xác suất ban đầu:

$$P(R = \text{Pass}) = \frac{3}{6} = 0.5, \quad P(R = \text{Fail}) = \frac{3}{6} = 0.5$$

$$P(S = \text{Yes} \mid R = \text{Pass}) = \frac{2}{3}, \quad P(S = \text{Yes} \mid R = \text{Fail}) = \frac{1}{3}$$

Bước 1: Tính xác suất quan sát (Total Probability Theorem):

$$\begin{aligned} P(S = \text{Yes}) &= P(S = \text{Yes} \mid R = \text{Pass})P(R = \text{Pass}) \\ &\quad + P(S = \text{Yes} \mid R = \text{Fail})P(R = \text{Fail}) \\ &= \frac{2}{3} \cdot 0.5 + \frac{1}{3} \cdot 0.5 = 0.5 \end{aligned}$$

Bước 2: Áp dụng Bayes cho từng lớp:

$$\begin{aligned} P(R = \text{Pass} \mid S = \text{Yes}) &= \frac{P(S = \text{Yes} \mid R = \text{Pass})P(R = \text{Pass})}{P(S = \text{Yes})} \\ &= \frac{\frac{2}{3} \cdot 0.5}{0.5} = \frac{2}{3} \\ P(R = \text{Fail} \mid S = \text{Yes}) &= \frac{P(S = \text{Yes} \mid R = \text{Fail})P(R = \text{Fail})}{P(S = \text{Yes})} \\ &= \frac{\frac{1}{3} \cdot 0.5}{0.5} = \frac{1}{3} \end{aligned}$$

Bước 3: Kết luận phân loại:

$$\boxed{R = \text{Pass}} \text{ vì } P(R = \text{Pass} \mid S = \text{Yes}) > P(R = \text{Fail} \mid S = \text{Yes})$$

Giải thích: Nếu một sinh viên đã học bài, xác suất đỗ là 66.7%.

3.4. Multiple Features

Phát biểu tổng quát

Giả sử:

- Tồn tại K lớp (class) khác nhau: $C_1, C_2, \ldots, C_K$.
- Có n **feature nhị phân**: $F_1, F_2, \ldots, F_n$, mỗi feature nhận giá trị 1 (xuất hiện/Yes/True) hoặc 0 (không xuất hiện/No/False).

Giả định Naïve Bayes (độc lập có điều kiện):

$$P(F_1, F_2, \ldots, F_n \mid C_k) = \prod_{i=1}^{n} P(F_i \mid C_k)$$

Phân phối Bernoulli cho mỗi feature:

$$P(F_i \mid C_k) = \begin{cases} \theta_{i|C_k}, & \text{nếu } F_i = 1 \\ 1 - \theta_{i|C_k}, & \text{nếu } F_i = 0 \end{cases}$$

trong đó:

$$\theta_{i|C_k} = \frac{\text{số mẫu thuộc } C_k \text{ có feature } F_i = 1}{\text{tổng số mẫu thuộc } C_k}$$

Công thức Bayes tổng quát:

$$P(C_k \mid F_1, F_2, \ldots, F_n) = \frac{P(C_k) \prod_{i=1}^{n} P(F_i \mid C_k)}{\sum_{j=1}^{K} P(C_j) \prod_{i=1}^{n} P(F_i \mid C_j)}$$

Quy tắc phân loại (Maximum a Posteriori – MAP):

$$\hat{C} = \arg\max_{C_k} P(C_k) \prod_{i=1}^{n} P(F_i \mid C_k)$$

Ý nghĩa:

- Với mỗi class C_k, ta tính xác suất hậu nghiệm dựa trên tích các xác suất Bernoulli của từng feature.
- Chọn class có xác suất lớn nhất làm kết quả dự đoán.

Dạng 2 features

Giả sử có hai lớp:

- $R =$ Pass
- $R =$ Fail

Ba **feature nhị phân**:

- $Co =$ Confident (Yes/No)
- $St =$ Studied (Yes/No)
- $Si =$ Sick (Yes/No)

Theo **giả định Naïve Bayes**: các feature **độc lập có điều kiện** cho trước class:

$$P(Co, St, Si \mid R) = P(Co \mid R)P(St \mid R)P(Si \mid R)$$

$$P(R \mid Co, St, Si) = \frac{P(Co, St, Si \mid R)P(R)}{P(Co, St, Si)}$$

Ví dụ minh họa: Multiple Features

Xét 7 sinh viên với kết quả thi được ghi nhận như bảng sau:

Confident	**Studied**	**Sick**	**Result**
Yes	No	Yes	Fail
Yes	No	No	Pass
Yes	Yes	Yes	Fail
No	No	Yes	Pass
No	Yes	No	Pass
No	No	No	Fail
Yes	Yes	Yes	??? (*cần dự đoán*)

Gọi:

- R là biến ngẫu nhiên đại diện cho kết quả (**Result**):

$$R_1 = \text{Pass}, \quad R_2 = \text{Fail}$$

- Các biến đặc trưng nhị phân (**feature**):

$$Co = \text{Confident}, \quad St = \text{Studied}, \quad Si = \text{Sick}$$

Bước 1: Tính các xác suất ban đầu (từ bảng dữ liệu)

$$P(R = \text{Pass}) = \frac{3}{6} = 0.5, \quad P(R = \text{Fail}) = \frac{3}{6} = 0.5$$

Tính các xác suất có điều kiện:

$$P(Co = \text{Yes} \mid R = \text{Pass}) = \frac{2}{3}, \quad P(Co = \text{Yes} \mid R = \text{Fail}) = \frac{1}{3}$$
$$P(St = \text{Yes} \mid R = \text{Pass}) = \frac{2}{3}, \quad P(St = \text{Yes} \mid R = \text{Fail}) = \frac{1}{3}$$
$$P(Si = \text{Yes} \mid R = \text{Pass}) = \frac{1}{3}, \quad P(Si = \text{Yes} \mid R = \text{Fail}) = \frac{2}{3}$$

Bước 2: Tính xác suất kết hợp theo giả định độc lập có điều kiện (Naïve Bayes)

$$\begin{aligned}
P(Co, St, Si \mid R = \text{Pass}) &= P(Co \mid R = \text{Pass}) \cdot P(St \mid R = \text{Pass}) \cdot P(Si \mid R = \text{Pass}) \\
&= \frac{2}{3} \cdot \frac{2}{3} \cdot \frac{1}{3} = \frac{4}{27} \\
P(Co, St, Si \mid R = \text{Fail}) &= P(Co \mid R = \text{Fail}) \cdot P(St \mid R = \text{Fail}) \cdot P(Si \mid R = \text{Fail}) \\
&= \frac{1}{3} \cdot \frac{1}{3} \cdot \frac{2}{3} = \frac{2}{27}
\end{aligned}$$

Bước 3: Tính xác suất quan sát $P(Co, St, Si)$

$$\begin{aligned}
P(Co, St, Si) &= P(Co, St, Si \mid R = \text{Pass})P(R = \text{Pass}) \\
&\quad + P(Co, St, Si \mid R = \text{Fail})P(R = \text{Fail}) \\
&= \frac{4}{27} \cdot 0.5 + \frac{2}{27} \cdot 0.5 = \frac{1}{9}
\end{aligned}$$

Bước 4: Áp dụng định lý Bayes

$$
\begin{aligned}
P(R = \text{Pass} \mid Co, St, Si) &= \frac{P(Co, St, Si \mid R = \text{Pass})P(R = \text{Pass})}{P(Co, St, Si)} \\
&= \frac{\frac{4}{27} \cdot 0.5}{1/9} = \frac{2}{3}
\end{aligned}
$$

$$
\begin{aligned}
P(R = \text{Fail} \mid Co, St, Si) &= \frac{P(Co, St, Si \mid R = \text{Fail})P(R = \text{Fail})}{P(Co, St, Si)} \\
&= \frac{\frac{2}{27} \cdot 0.5}{1/9} = \frac{1}{3}
\end{aligned}
$$

Bước 5: Kết luận phân loại

$$\boxed{R = \text{Pass}} \quad \text{vì } P(R = \text{Pass} \mid Co, St, Si) > P(R = \text{Fail} \mid Co, St, Si)$$

Dự đoán cho sinh viên thứ 7
Nếu sinh viên thứ 7 có thông tin:

$$Co = \text{Yes}, \quad St = \text{Yes}, \quad Si = \text{Yes}$$

thì kết quả dự đoán là:

$$\boxed{\text{Result} = \text{Pass}}$$

vì xác suất đỗ là 66.7%, cao hơn rớt (33.3%).

Phần 2: Gaussian Function

2. Gaussian Function

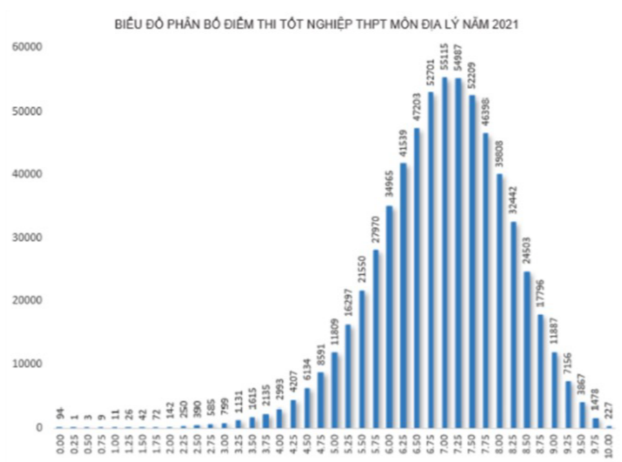
2.1. Definition

Phân phối chuẩn (Normal Distribution)

Khái niệm

- Một **phân phối** đơn giản là tập hợp các giá trị dữ liệu được sắp xếp theo thứ tự.
- **Phân phối xác suất** mô tả cách mà xác suất được phân bố trên các giá trị của một biến ngẫu nhiên.
- **Phân phối chuẩn (Gaussian Distribution)** là loại phân phối xác suất quan trọng nhất, thường xuất hiện tự nhiên trong nhiều hiện tượng thực tế (chiều cao, cân nặng, điểm thi...).

Ví dụ minh họa: Biểu đồ phân bố điểm thi THPT môn Địa lý năm 2021 tại Việt Nam có dạng xấp xỉ một đường cong chuẩn.



Biểu đồ phân bố điểm thi THPT môn Địa lý năm 2021

Mean (Kỳ vọng)

Công thức

Cho tập dữ liệu $X = \{x_1, x_2, \ldots, x_n\}$, **mean** được tính như sau:

$$\mu_X = \frac{1}{n}\sum_{i=1}^{n} x_i$$

Mean đặc trưng cho trung tâm phân phối xác suất của biến ngẫu nhiên.

Ví dụ: Với $X = \{2, 8, 5, 4, 1, 4\}$, $n = 6$:

$$\mu_X = \frac{1}{6}(2 + 8 + 5 + 4 + 1 + 4) = \frac{24}{6} = 4$$

Variance (Phương sai)

Công thức

Phương sai đo lường mức độ phân tán của dữ liệu quanh giá trị trung bình:

$$var(X) = \frac{1}{n}\sum_{i=1}^{n}(x_i - \mu)^2$$

Ví dụ: Với $X = \{5, 3, 6, 7, 4\}$, $n = 5$:

$$\mu = \frac{1}{5}(5 + 3 + 6 + 7 + 4) = 5$$

$$var(X) = \frac{1}{5}[(5-5)^2 + (3-5)^2 + (6-5)^2 + (7-5)^2 + (4-5)^2] = 2$$

Gaussian Function

Công thức chuẩn

Hàm mật độ xác suất (PDF) của phân phối chuẩn:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}}e^{-\frac{(x-\mu)^2}{2\sigma^2}}, \quad -\infty < x < \infty$$

- μ: mean (trung bình)
- σ^2: variance (phương sai)

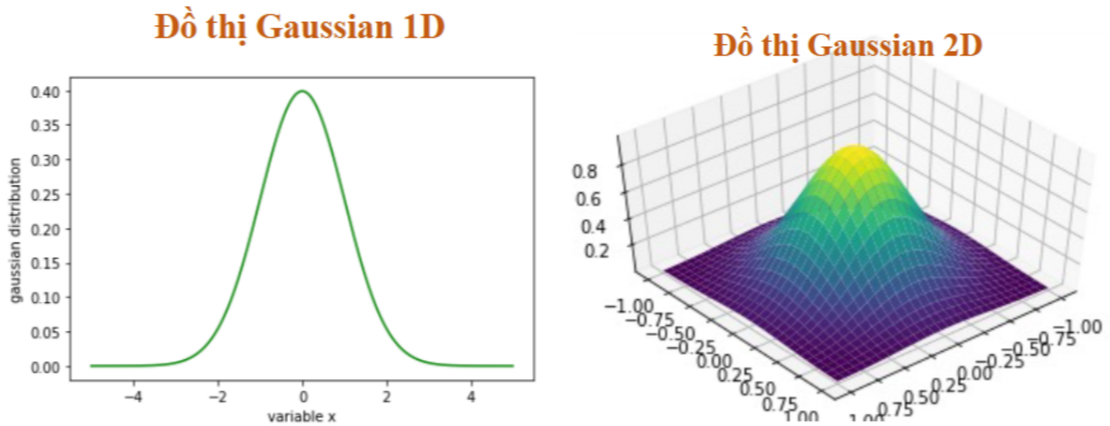
Ví dụ: Cho $\mu = 0, \sigma^2 = 1$, tính $f(x)$ với $x \in [-4, -3, -2, -1, 0, 1, 2, 3]$:

$$f(x = -4) = \frac{1}{\sqrt{2\pi}}e^{-\frac{(-4-0)^2}{2}} = \frac{e^{-8}}{\sqrt{2\pi}} \approx 1.3 \times 10^{-4}$$

Bảng giá trị:

x	-4	-3	-2	-1	0	1	2	3
$f(x)$	1.3e-04	4.4e-03	5.3e-02	2.4e-01	3.9e-01	2.4e-01	5.3e-02	4.4e-03

Đồ thị minh họa:



(Trái) Đồ thị Gaussian 1D, (Phải) Đồ thị Gaussian 2D

2.2. Central Limit Theorem (Định lý giới hạn trung tâm)

Phát biểu

Định lý giới hạn trung tâm (Central Limit Theorem – CLT) khẳng định rằng:

- Khi ta lấy **tổng** hoặc **giá trị trung bình** của một số lượng lớn các **biến ngẫu nhiên độc lập**, không nhất thiết phải có cùng phân phối,
- Sau khi được **chuẩn hoá đúng cách** (chuẩn hóa z-score), phân phối của tổng hoặc giá trị trung bình này sẽ **tiệm cận phân phối chuẩn** (**normal distribution**),
- Ngay cả khi các biến gốc **không tuân theo phân phối chuẩn**.

Cách chuẩn hóa: Giả sử ta có $X_1, X_2, \ldots, X_n$ là các biến ngẫu nhiên độc lập với:

$$\mu = E[X_i], \quad \sigma^2 = Var(X_i)$$

Gọi:

$$\bar{X} = \frac{1}{n}\sum_{i=1}^{n} X_i$$

thì khi $n \to \infty$:

$$Z = \frac{\bar{X} - \mu}{\sigma/\sqrt{n}} \to \mathcal{N}(0, 1)$$

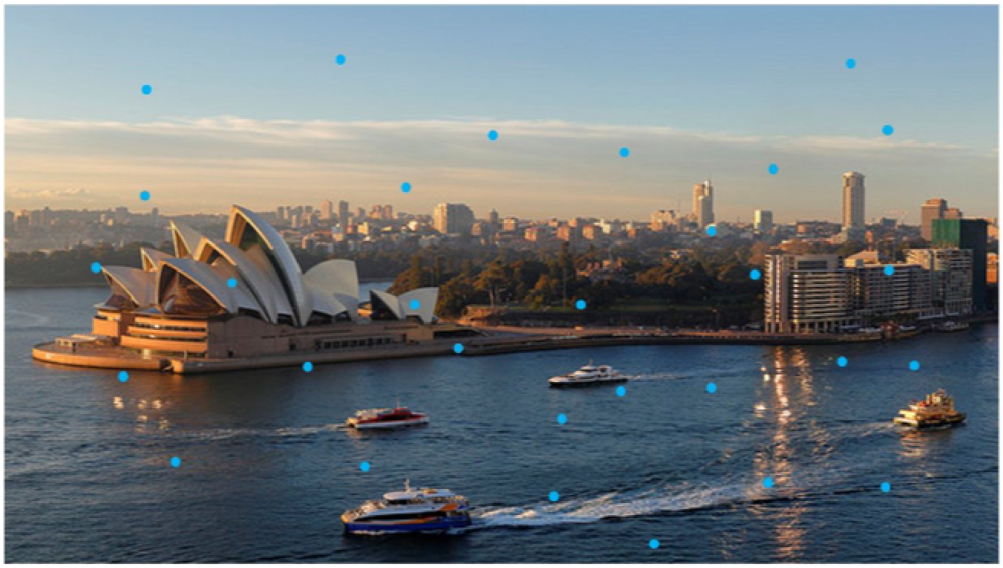
(về phân phối chuẩn tắc).

Ý nghĩa:

- Dù dữ liệu gốc có phân phối **lệch** hay **kỳ lạ** (uniform, exponential, v.v.), nhưng khi tính trung bình nhiều mẫu độc lập, kết quả sẽ gần giống **hình chuông chuẩn**.
- CLT là nền tảng cho các phương pháp suy luận thống kê, vì nó cho phép ước lượng tham số bằng phân phối chuẩn ngay cả khi phân phối gốc không chuẩn.

Ví dụ minh họa:

- Lấy **ngẫu nhiên 30 pixel** trong một bức ảnh, tính **trung bình giá trị màu**.
- Lặp lại **10.000 lần**, vẽ biểu đồ histogram của 10.000 giá trị trung bình thu được.
- Kết quả: Histogram này xấp xỉ **phân phối chuẩn**, mặc dù giá trị màu gốc của từng pixel không hề chuẩn.



Ví dụ minh họa định lý giới hạn trung tâm trên ảnh chụp thực tế

Phần 3: Gaussian Naïve Bayes Classifier

3.1. Giới thiệu

Khái niệm

Gaussian Naïve Bayes (GNB) là một biến thể của Naïve Bayes, chuyên dùng cho dữ liệu **liên tục**.
Đặc điểm:

- Các feature được giả định tuân theo **phân phối chuẩn (Gaussian Distribution)**.
- Thay vì đếm tần suất xuất hiện như Bernoulli hoặc Multinomial NB, ta ước lượng các tham số **mean (μ)** và **variance (σ^2)** cho từng feature theo từng class.
- Được sử dụng rộng rãi trong phân loại khi dữ liệu đầu vào là các giá trị thực (chiều dài, trọng lượng, điểm số...).

3.2. Công thức tổng quát

Bài toán

Mục tiêu: Dự đoán class C khi biết n feature liên tục $F_1, F_2, \ldots, F_n$.

Theo Bayes:

$$P(C \mid F_1, F_2, \ldots, F_n) = \frac{P(F_1, F_2, \ldots, F_n \mid C)\, P(C)}{P(F_1, F_2, \ldots, F_n)}$$

Vì $P(F_1, F_2, \ldots, F_n)$ là hằng số khi so sánh các class:

$$\hat{C} = \arg\max_{C} P(C)\, P(F_1, F_2, \ldots, F_n \mid C)$$

Giả định Naïve (Độc lập có điều kiện):

$$P(F_1, F_2, \ldots, F_n \mid C) = \prod_{i=1}^{n} P(F_i \mid C)$$

Ước lượng Gaussian cho mỗi feature liên tục:
Với mỗi feature F_i (giả sử tuân theo phân phối chuẩn trong từng class C):

$$P(F_i = x \mid C) \approx f_{F_i|C}(x) = \frac{1}{\sigma_{i|C}\sqrt{2\pi}} \exp\left(-\frac{(x - \mu_{i|C})^2}{2\sigma_{i|C}^2}\right)$$

trong đó:

$$\mu_{i|C} = \frac{1}{N_C}\sum_{j=1}^{N_C} x_{ij}, \quad \sigma_{i|C}^2 = \frac{1}{N_C}\sum_{j=1}^{N_C}(x_{ij} - \mu_{i|C})^2$$

Vì sao dùng PDF thay vì xác suất đúng nghĩa?

- Trong xác suất liên tục, **xác suất tại một điểm đúng nghĩa bằng 0**:

$$P(F_i = x \mid C) = 0$$

- Thay vào đó, ta quan tâm tới **mật độ xác suất (PDF)**:

$$f_{F_i|C}(x) = \lim_{\Delta x \to 0} \frac{P(x \leq F_i \leq x + \Delta x \mid C)}{\Delta x}$$

- Trong Gaussian Naïve Bayes, khi so sánh giữa các class, ta chỉ cần so sánh **tỉ lệ mật độ tại điểm** x vì:

$$P(x \leq F_i \leq x + \Delta x \mid C) \propto f_{F_i|C}(x)$$

  và Δx triệt tiêu trong quá trình so sánh.

- Do đó, công thức Bayes vẫn hoạt động đúng khi thay thế $P(F_i \mid C)$ bằng $f_{F_i|C}(F_i)$.

Quy tắc phân loại (Maximum A Posteriori – MAP):

$$\hat{C} = \arg\max_C \; P(C \mid F_1, F_2, \ldots, F_n) = \arg\max_C \; P(C) \prod_{i=1}^{n} f_{F_i|C}(F_i)$$

Giải thích về MAP (Maximum A Posteriori)

MAP là gì? MAP (Maximum A Posteriori) là quy tắc ra quyết định chọn **class có xác suất hậu nghiệm lớn nhất**:

$$\hat{C} = \arg\max_C P(C \mid L)$$

Tại sao MAP tối ưu trong Gaussian NB?

- Trong Gaussian Naïve Bayes, mỗi class được mô hình bằng phân phối chuẩn với tham số ước lượng từ dữ liệu.
- Bayes đảm bảo:

$$P(C \mid L) \propto P(L \mid C)P(C)$$

  vì $P(L)$ là hằng số với mọi class.

- Chọn class có posterior lớn nhất tương đương với việc tối thiểu hoá xác suất phân loại sai (Bayes Optimal Classifier).
- MAP còn hợp lý khi ta có prior khác nhau giữa các class ($P(C)$ không đồng đều).

Tóm lại: MAP là quy tắc ra quyết định tối ưu nhất trong ý nghĩa xác suất: nó giảm thiểu rủi ro phân loại sai trung bình.

3.3. One Feature

Phát biểu

Giả sử ta có:

- Hai lớp (**class/label**):
$$C = 0 \quad \text{và} \quad C = 1$$

- Một feature liên tục:
$$L = \text{Length (continuous)}$$

Mục tiêu: Dự đoán class khi biết $L = 3.0$.
Theo định lý Bayes:
$$P(C \mid L) = \frac{P(L \mid C)P(C)}{P(L)}$$

với:
$$P(L) = P(L \mid C = 0)P(C = 0) + P(L \mid C = 1)P(C = 1)$$

Ví dụ minh họa: One Feature

Xét 10 mẫu dữ liệu với đặc trưng chiều dài (**Length**) và kết quả phân loại (**Category**) được ghi nhận như sau:

Length	Category
1.4	0
1.0	0
1.3	0
1.9	0
2.0	0
3.8	1
4.1	1
3.9	1
4.2	1
3.4	1

Gọi:

- C: biến ngẫu nhiên đại diện cho **Category**, với:
$$C = 0 \quad (\text{class } 0) \quad \text{và} \quad C = 1 \quad (\text{class } 1)$$

- L: feature liên tục (**Length**).

Bước 1: Tính xác suất tiên nghiệm (Prior Probability)
Vì mỗi class có 5 mẫu:
$$P(C = 0) = P(C = 1) = 0.5$$

Bước 2: Ước lượng tham số Gaussian cho từng class

Sử dụng công thức:

$$\mu_C = \frac{1}{N_C}\sum_{j=1}^{N_C} L_j, \quad \sigma_C^2 = \frac{1}{N_C}\sum_{j=1}^{N_C}(L_j - \mu_C)^2$$

Kết quả:

$$\mu_0 = 1.52, \quad \sigma_0^2 = 0.1416$$
$$\mu_1 = 3.88, \quad \sigma_1^2 = 0.0776$$

Bước 3: Tính mật độ Gaussian tại $L = 3.0$
Áp dụng công thức phân phối chuẩn:

$$P(L \mid C) = \frac{1}{\sigma_C\sqrt{2\pi}} \exp\left(-\frac{(L-\mu_C)^2}{2\sigma_C^2}\right)$$

Cụ thể:

$$P(L = 3.0 \mid C = 0) = 0.0004638$$
$$P(L = 3.0 \mid C = 1) = 0.0097495$$

Bước 4: Tính xác suất quan sát (Total Probability)

$$P(L = 3.0) = P(L = 3.0 \mid C = 0)P(C = 0) + P(L = 3.0 \mid C = 1)P(C = 1)$$
$$P(L = 3.0) = 0.0004638 \times 0.5 + 0.0097495 \times 0.5 = 0.005106$$

Bước 5: Áp dụng định lý Bayes (Quy tắc MAP)

$$P(C = 0 \mid L = 3.0) = \frac{P(L = 3.0 \mid C = 0)P(C = 0)}{P(L = 3.0)} = \frac{0.0004638 \times 0.5}{0.005106} \approx 0.0455$$

$$P(C = 1 \mid L = 3.0) = \frac{P(L = 3.0 \mid C = 1)P(C = 1)}{P(L = 3.0)} = \frac{0.0097495 \times 0.5}{0.005106} \approx 0.9545$$

Bước 6: Kết luận phân loại

$$\boxed{C = 1} \quad \text{vì } P(C = 1 \mid L = 3.0) > P(C = 0 \mid L = 3.0)$$

Giải thích: Khi $Length = 3.0$, xác suất hậu nghiệm thuộc class 1 là khoảng 95.45%, cao hơn rất nhiều so với class 0 (4.55%).

4.4. Multiple Features

Phát biểu tổng quát

Bài toán: Dự đoán class C khi biết n **feature liên tục** $F_1, F_2, \ldots, F_n$.
Theo định lý Bayes:

$$P(C \mid F_1, F_2, \ldots, F_n) = \frac{P(F_1, F_2, \ldots, F_n \mid C)\, P(C)}{P(F_1, F_2, \ldots, F_n)}$$

Giả định Naïve (độc lập có điều kiện):

$$P(F_1, F_2, \dots, F_n \mid C) = \prod_{i=1}^{n} P(F_i \mid C)$$

Ước lượng Gaussian cho mỗi feature:

$$P(F_i = x \mid C) = \frac{1}{\sigma_{i|C}\sqrt{2\pi}} \exp\left(-\frac{(x-\mu_{i|C})^2}{2\sigma_{i|C}^2}\right)$$

Quy tắc phân loại (Maximum A Posteriori – MAP):

$$\hat{C} = \arg\max_{C} P(C) \prod_{i=1}^{n} P(F_i \mid C)$$

Tham số cần ước lượng:

$$\mu_{i|C} = \frac{1}{N_C}\sum_{j=1}^{N_C} x_{ij}, \quad \sigma_{i|C}^2 = \frac{1}{N_C}\sum_{j=1}^{N_C}(x_{ij} - \mu_{i|C})^2$$

Dạng 2 features

Giả sử có hai lớp:

- $C = 0$
- $C = 1$

Hai **feature liên tục**:

- L = Length (continuous)
- W = Width (continuous)

Theo **giả định Naïve Bayes**: các feature **độc lập có điều kiện** cho trước class:

$$P(L, W \mid C) = P(L \mid C) \cdot P(W \mid C)$$

$$P(C \mid L, W) = \frac{P(L, W \mid C) \cdot P(C)}{P(L, W)}$$

Ví dụ minh họa: Two Features

Dữ liệu được ghi nhận như sau:

Length	Width	Category
4.9	3.0	0
4.7	3.2	0
4.6	3.1	0
5.0	3.6	0
5.4	3.9	0
4.6	3.4	0
6.4	3.2	1
6.9	3.1	1
5.5	2.3	1
6.5	2.8	1
5.7	2.8	1
6.3	3.3	1

Mục tiêu: Dự đoán Category khi Length = 4.1 và Width = 2.9.
Gọi:

- C: class (**Category**) với $C = 0$ hoặc $C = 1$.
- L: feature liên tục (**Length**).
- W: feature liên tục (**Width**).

Bước 1: Tính xác suất tiên nghiệm (Prior Probability)
Vì mỗi class có 6 mẫu:

$$P(C = 0) = P(C = 1) = 0.5$$

Bước 2: Tính tham số Gaussian cho từng class

$$\mu_{0L} = 4.867, \quad \sigma^2_{0L} = 0.078$$
$$\mu_{0W} = 3.367, \quad \sigma^2_{0W} = 0.095$$

$$\mu_{1L} = 6.216, \quad \sigma^2_{1L} = 0.228$$
$$\mu_{1W} = 2.916, \quad \sigma^2_{1W} = 0.111$$

Bước 3: Tính xác suất có điều kiện $P(L, W \mid C)$
Áp dụng công thức Gaussian:

$$P(x \mid C) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

Ta có:

$$P(L = 4.1 \mid C = 0) = 0.0342, \quad P(W = 2.9 \mid C = 0) = 0.4129$$
$$P(L = 4.1 \mid C = 1) = 0.00004, \quad P(W = 2.9 \mid C = 1) = 1.1938$$

Do độc lập có điều kiện:

$$P(L = 4.1, W = 2.9 \mid C = 0) = 0.0342 \times 0.4129 = 0.01412$$

$$P(L = 4.1, W = 2.9 \mid C = 1) = 0.00004 \times 1.1938 = 0.000047$$

Bước 4: Tính xác suất quan sát $P(L, W)$

$$P(L = 4.1, W = 2.9) = P(L, W \mid C = 0)P(C = 0) + P(L, W \mid C = 1)P(C = 1)$$
$$P(L = 4.1, W = 2.9) = 0.01412 \cdot 0.5 + 0.000047 \cdot 0.5 = 0.00708$$

Bước 5: Áp dụng Bayes để tính Posterior

$$P(C = 0 \mid L = 4.1, W = 2.9) = \frac{P(L, W \mid C = 0)P(C = 0)}{P(L, W)} = \frac{0.01412 \cdot 0.5}{0.00708} \approx 0.9962$$

$$P(C = 1 \mid L = 4.1, W = 2.9) = \frac{P(L, W \mid C = 1)P(C = 1)}{P(L, W)} = \frac{0.000047 \cdot 0.5}{0.00708} \approx 0.0038$$

Bước 6: Kết luận phân loại

$$\boxed{C = 0} \quad \text{vì } P(C = 0 \mid L = 4.1, W = 2.9) > P(C = 1 \mid L = 4.1, W = 2.9)$$

Giải thích: Khi $Length = 4.1$ và $Width = 2.9$, xác suất thuộc class 0 (99.62%) cao hơn rất nhiều so với class 1 (0.38%).

Phần 4: Mở rộng: Word segmentation trong bài toán phân loại email spam

Trong buổi trước, chúng ta đã tìm hiểu về mô hình **Unigram** – nơi mỗi email được biểu diễn như một túi các từ độc lập, và xác suất của toàn văn bản được tính bằng tích xác suất từng từ.

Trong buổi này, khi tiếp cận mô hình **Naive Bayes Classifier (NBC)** cho bài toán phân loại email spam, chúng ta chính thức đưa mô hình Unigram vào hành động — như một phần quan trọng trong quá trình ước lượng xác suất có điều kiện $P(w \mid \text{label})$.

Tuy nhiên, để mô hình NBC + Unigram có thể làm việc, có một bước tiền xử lý bắt buộc phải có: **Word Segmentation** — tách đoạn văn bản liền mạch thành các từ rời rạc để tính xác suất.

Trong phần này, chúng ta sẽ tìm hiểu vì sao việc tách từ lại quan trọng đến vậy, và làm thế nào để máy tính có thể 'đọc' được các từ từ một chuỗi ký tự không có khoảng trắng.

4.1. Word Segmentation — Tách từ trong văn bản tiếng Anh tự do

*"Nếu Unigram là cách để dạy máy tính hiểu từng từ, thì Word Segmentation là cách giúp máy **nhìn ra** những từ đó từ đầu."*

Đặt vấn đề

Hãy xem xét đoạn văn sau:

```
congratulationsyouhavewonafreeiphoneclicknow
```

Bạn có nhận ra câu này?

`Congratulations! You have won a free iPhone. Click now.`

Với người, đây là chuyện hiển nhiên. Nhưng với máy tính, nếu không có khoảng trắng ngăn cách, nó chỉ thấy một chuỗi ký tự không rõ đâu là từ.

→ **Đây chính là lúc cần Word Segmentation.**

Tại sao lại cần Word Segmentation?

Trong rất nhiều hệ thống xử lý ngôn ngữ tự nhiên, dữ liệu đầu vào không hề được chuẩn hóa. Đặc biệt là:

- Spam thường cố tình viết liền để đánh lừa hệ thống.
- Một số ngôn ngữ như tiếng Trung, Nhật, không dùng khoảng trắng phân tách từ.
- Dữ liệu chat, bình luận, tin nhắn thường không có chuẩn mực chính tả.

Vì vậy, việc tách chuỗi ký tự thành từ là bước **cực kỳ quan trọng trước khi áp dụng Unigram hay bất kỳ mô hình phân loại nào**.

Ví dụ

Hãy xét lại chuỗi:

```
clickheretowinfreeiphone
```

Nếu không tách đúng, máy tính có thể hiểu nhầm thành các từ vô nghĩa như:
`cli | ckhe | reto | winf | reeip | hone`

→ Lúc này, toàn bộ mô hình Unigram phía sau sẽ "rác" vì đầu vào đã sai.

4.2. Mô hình xác suất cho Word Segmentation (Bayes-based)

Word Segmentation là gì?

Word Segmentation là quá trình tách một chuỗi ký tự liền mạch (không có dấu cách) thành một chuỗi các từ riêng biệt sao cho hợp lý và có ý nghĩa.

Mục tiêu: Cho một chuỗi ký tự $C = c_1c_2\ldots c_n$, ta cần tìm ra cách chia chuỗi đó thành một dãy từ $w_1, w_2, \ldots, w_k$ sao cho:

$$\boxed{(w_1, w_2, \ldots, w_k)^* = \arg \max_{(w_1,\ldots,w_k)} P(w_1, w_2, \ldots, w_k)}$$

Tức là: chọn cách phân đoạn sao cho xác suất xảy ra của toàn bộ dãy từ là cao nhất.

Áp dụng mô hình Unigram:
Giả định đơn giản là các từ độc lập với nhau về mặt xác suất. Khi đó:

$$P(w_1, w_2, \ldots, w_k) = \prod_{i=1}^{k} P(w_i)$$

→ Bài toán tách từ trở thành bài toán **tối ưu hóa xác suất**: Tìm cách chia chuỗi sao cho tích xác suất của các từ là lớn nhất.

Mô hình bài toán

- Gọi $C = c_1c_2\ldots c_n$ là chuỗi ký tự cần tách.
- Gọi $W = w_1w_2\ldots w_k$ là một cách tách chuỗi thành các từ.
- Ta chọn W^* sao cho:

$$W^* = \arg\max_W P(w_1, w_2, \ldots, w_k) = \arg\max_W \prod_{i=1}^{k} P(w_i)$$

- $P(w_i)$ được ước lượng từ tập văn bản lớn (văn phạm email thực tế).

Ví dụ

Giả sử chuỗi cần tách là:

```
clickheretowinfreeiphone
```

Ta xét 2 cách phân tách chuỗi thành từ:

Phương án A (tách đúng):

$$W_A = \texttt{click} \mid \texttt{here} \mid \texttt{to} \mid \texttt{win} \mid \texttt{free} \mid \texttt{iphone}$$

Giả sử xác suất Unigram (ước lượng từ tập email spam) là:

$$\begin{array}{rr}
P(\texttt{click}) = 0.20 & \log P = -0.70 \\
P(\texttt{here}) = 0.10 & \log P = -1.00 \\
P(\texttt{to}) = 0.25 & \log P = -0.60 \\
P(\texttt{win}) = 0.05 & \log P = -1.30 \\
P(\texttt{free}) = 0.15 & \log P = -0.82 \\
P(\texttt{iphone}) = 0.05 & \log P = -1.30 \\
\hline
\textbf{Tổng log xác suất} & = \boxed{-5.72}
\end{array}$$

Phương án B (tách sai):

$$W_B = \texttt{cli} \mid \texttt{ckhe} \mid \texttt{retowin} \mid \texttt{freeip} \mid \texttt{hone}$$

Giả sử các "từ" này không tồn tại trong từ điển, ta áp dụng Laplace smoothing (ví dụ mỗi từ được gán xác suất nhỏ $P = 0.001$):

$$\log P(w_i) = \log 0.001 \approx -6.91 \quad \Rightarrow \quad \text{Tổng log xác suất} = 5 \times (-6.91) = \boxed{-34.55}$$

So sánh:

$$\log P(W_A) = -5.72 \quad \gg \quad \log P(W_B) = -34.55$$

$\rightarrow$ Phương án A được ưu tiên vì có log-xác suất cao hơn rất nhiều.

Nhận xét:

- Word Segmentation là **bước đầu tiên và bắt buộc** nếu đầu vào chưa có dấu cách.
- Nếu tách từ sai $\rightarrow$ toàn bộ pipeline Naive Bayes phía sau (ước lượng xác suất, dự đoán nhãn) sẽ hỏng.
- Mô hình Word Segmentation có thể dùng lại chính giả định Unigram — giả định quen thuộc mà ta đã dùng trong NBC.

4.3. Làm thế nào để tách từ ?

Có nhiều thuật toán Word Segmentation, nhưng cách phổ biến nhất là sử dụng một từ điển thống kê $P(w)$, ước lượng từ một tập văn bản lớn. Thuật toán cơ bản gồm 2 bước chính:

Bước 1: Ước lượng xác suất $P(w)$ từ dữ liệu huấn luyện

Để áp dụng mô hình Word Segmentation dựa trên Unigram, ta cần một hàm xác suất $P(w)$ — xác suất của mỗi từ xuất hiện trong văn bản.

Cách tính xác suất Unigram $P(w)$

Giả sử ta có tập huấn luyện gồm N từ (có thể lặp lại), trong đó từ w xuất hiện $f(w)$ lần. Khi đó:

$$P(w) = \frac{f(w)}{N}$$

- $f(w)$: số lần từ w xuất hiện trong tập văn bản huấn luyện (email spam hoặc cả spam + ham).
- N: tổng số từ (có lặp lại) trong tập đó.

$\rightarrow$ Từ càng phổ biến, xác suất càng cao.

Ví dụ:
Nếu trong 10000 từ huấn luyện, từ `free` xuất hiện 150 lần thì:

$$P(\texttt{free}) = \frac{150}{10000} = 0.015 \quad \Rightarrow \quad \log P(\texttt{free}) \approx -4.20$$

Chú ý: Ta thường dùng $\log P(w)$ để cộng thay vì nhân, tránh underflow khi xử lý chuỗi dài.

Bước 2: Tìm cách tách tối ưu bằng quy hoạch động (Dynamic Programming)

Để tìm ra cách chia chuỗi ký tự $C = c_1c_2...c_n$ thành dãy từ $W = w_1, ..., w_k$ sao cho $\prod_i P(w_i)$ lớn nhất, ta sử dụng kỹ thuật **quy hoạch động**:

Ý tưởng

- Gọi $F(i)$ là **log-xác suất tốt nhất** để tách chuỗi con $C[0:i]$.
- Với mỗi vị trí i, ta thử các cách cắt từ $j \rightarrow i$ (giới hạn độ dài từ).
- Nếu chuỗi con $C[j:i]$ là một từ hợp lệ, ta tính:

$$F(i) = \max_{j<i} \left[F(j) + \log P(C[j:i])\right]$$

- Cuối cùng, ta truy vết (backtrace) để lấy ra chuỗi từ tương ứng.

Ví dụ

Bài toán: Tách chuỗi ký tự sau thành các từ sao cho tổng **log-xác suất** là lớn nhất:

```
getfreeiphone
```

Giả sử tập xác suất $P(w)$ như sau:

$$
\begin{aligned}
P(\texttt{get}) &= 0.2 & \Rightarrow \quad \log P &= -0.7\\
P(\texttt{free}) &= 0.1 & \Rightarrow \quad \log P &= -1.0\\
P(\texttt{iphone}) &= 0.05 & \Rightarrow \quad \log P &= -1.3\\
P(\texttt{getfree}) &= 0.005 & \Rightarrow \quad \log P &= -5.3\\
P(\texttt{freeiphone}) &= 0.001 & \Rightarrow \quad \log P &= -6.9\\
(\text{Các từ khác}) &\Rightarrow \text{Xác suất thấp hoặc không có}
\end{aligned}
$$

Duyệt từng vị trí bằng DP:
Gọi $F(i)$ là log-xác suất tốt nhất để tách $C[0:i]$

- $F(3) = \texttt{get} \rightarrow F(3) = \log P(\texttt{get}) = -0.7$
- $F(7) = \texttt{get} + \texttt{free} \rightarrow F(7) = F(3) + \log P(\texttt{free}) = -0.7 - 1.0 = -1.7$
- $F(13) = \texttt{get} + \texttt{free} + \texttt{iphone} \rightarrow$

$$F(13) = F(7) + \log P(\texttt{iphone}) = -1.7 - 1.3 = \boxed{-3.0}$$

- Các phương án khác (như $\texttt{getfree} + \texttt{iphone}$) có tổng log-xác suất thấp hơn: $-5.3 - 1.3 = -6.6$

Kết quả:

$$\boxed{\texttt{get} \mid \texttt{free} \mid \texttt{iphone} \quad \text{có tổng log-xác suất cao nhất: } -3.0}$$

→ Đây là cách tách tối ưu được thuật toán lựa chọn.

4.6. Code Python minh họa Word Segmentation trong bài toán phân loại email spam

Trước khi đi vào chi tiết mã nguồn, hãy cùng nhắc lại toàn bộ bối cảnh bài toán:

1. Bối cảnh bài toán phân loại email spam

Mục tiêu của chúng ta là xây dựng một hệ thống có thể phân loại một email thành:

- **Spam** (thư rác)
- **Ham** (thư bình thường)

Cách tiếp cận chính:

1. Áp dụng mô hình **Naive Bayes + Unigram** để tính xác suất:

$$P(\text{Spam} \mid \text{email}) \propto P(\text{Spam}) \times \prod_{i=1}^{n} P(w_i \mid \text{Spam})$$

2. So sánh với $P(\text{Ham} \mid \text{email})$ → Chọn nhãn có xác suất lớn hơn.

Nhưng để làm được điều này, mô hình cần biết **chính xác các từ trong email**. Nếu email được viết liền, như:

```
getyourfreeiphoneclicknow
```

→ Chúng ta phải tách được thành:

`get` | `your` | `free` | `iphone` | `click` | `now`

Tại sao? Vì các từ như `free`, `click`, `iphone` thường có xác suất rất cao trong lớp Spam. Nếu không tách đúng → mô hình Naive Bayes sẽ không thể tính được các xác suất này → dự đoán sai.

2. Cách tiếp cận tách từ (Word Segmentation)

Như đã trình bày ở các phần trước:

- Ta sử dụng một từ điển thống kê $P(w)$, ước lượng từ tập văn bản lớn (thường là tập email spam + ham).
- Bài toán trở thành: tìm cách chia chuỗi thành các từ sao cho **tích xác suất** (hoặc tổng log-xác suất) của các từ là lớn nhất.

$$W^* = \arg\max_W \prod_{i=1}^{k} P(w_i) \quad \iff \quad W^* = \arg\max_W \sum_{i=1}^{k} \log P(w_i)$$

- Kỹ thuật giải: **Quy hoạch động (Dynamic Programming)**.

3. Mã nguồn Python minh họa

Dưới đây là một đoạn mã Python minh họa cách tách từ bằng kỹ thuật Dynamic Programming.

Mã nguồn minh họa Word Segmentation

```python
import math
from functools import lru_cache

# --- 1. Unigram probability dictionary ---
corpus = {
    'free': 5000,
    'click': 4500,
    'here': 3000,
    'win': 3500,
    'money': 4000,
```

```
    'iphone': 4200,
    'you': 10000,
    'your': 6000,
    'have': 7000,
    'won': 1500,
    'a': 12000,
    'now': 2500,
    'get': 4000
}

total = sum(corpus.values())

def P(w):
    """Unigram probability with strong penalty for unknown words"""
    return corpus.get(w, 1e-9) / total

# --- 2. Optimal segmentation using Dynamic Programming ---
@lru_cache(maxsize=None)
def segment(text):
    if not text:
        return (0.0, [])
    candidates = []
    for i in range(1, min(len(text) + 1, 20)):
        word = text[:i]
        log_prob = math.log(P(word))
        remaining_log_prob, remaining_words = segment(text[i:])
        candidates.append((log_prob + remaining_log_prob, [word] +
    remaining_words))
    return max(candidates, key=lambda x: x[0])

# --- 3. Function to compute log-prob for a given fixed segmentation ---
def log_prob_for_sequence(words):
    return sum(math.log(P(w)) for w in words)

# --- 4. Test ---
text = "getyourfreeiphoneclicknow"

# Optimal segmentation
opt_log, opt_words = segment(text)

# Wrong segmentation (manually forced)
wrong_words = ["gety", "ourf", "reemo", "neycli", "ckhere"]
wrong_log = log_prob_for_sequence(wrong_words)

# --- 5. Print comparison ---
print("Optimal Segmentation:")
print(f"Words: {' | '.join(opt_words)}")
print(f"Log-Probability: {opt_log:.2f}\n")

print("Wrong Segmentation:")
print(f"Words: {' | '.join(wrong_words)}")
print(f"Log-Probability: {wrong_log:.2f}\n")

print("Difference:", f"{opt_log - wrong_log:.2f} (log scale)")
```

Output:

```
Optimal Segmentation:
Words: get | your | free | iphone | click | now
Log-Probability: -16.60

Wrong Segmentation:
Words: gety | ourf | reemo | neycli | ckhere
Log-Probability: -159.19

Difference: 142.59 (log scale)
```

5. Giải thích kết quả

- **Danh sách từ tối ưu:** `['get', 'your', 'free', 'iphone', 'click', 'now']` → Đây chính là cách tách tối ưu vì tổng log-xác suất của chuỗi này **cao hơn rất nhiều** (ít âm hơn) so với mọi cách tách sai.
- **Tổng log-xác suất (tách đúng):**

$$\log P(W_{\text{đúng}}) = -16.60$$

  (dù vẫn âm, nhưng **ít âm hơn** so với cách tách sai → xác suất cao hơn đáng kể).

- **So sánh với tách sai:**

So sánh log-xác suất

Phân đoạn	Cách tách	Tổng log-xác suất
Đúng	get | your | free | iphone | click | now	-16.60
Sai	gety | ourf | reemo | neycli | ckhere	-159.19

→ **Cách tách đúng có log-xác suất cao hơn gần** 142.6 **đơn vị log**. Trên thang log (log-likelihood), chênh lệch này tương đương với:

$$\frac{P(W_{\text{đúng}})}{P(W_{\text{sai}})} \approx e^{142.6} \text{ (rất lớn!)}$$

→ Mô hình gần như chắc chắn chọn cách tách đúng.

- **Ý nghĩa với Naive Bayes:**

  Khi email được tách đúng thành các token có nghĩa (`get`, `free`, `click`,...), Naive Bayes có thể tra chính xác xác suất $P(w_i \mid \text{Spam})$ và tính:

$$P(\text{Spam} \mid \text{email}) \propto P(\text{Spam}) \times \prod_i P(w_i \mid \text{Spam})$$

  Các từ "nhạy cảm" như `free`, `click`, `iphone` có xác suất cao trong nhóm spam, nên chúng **kéo mạnh xác suất Spam lên**. Nếu tách sai thành các "từ rác" (`gety`, `ourf`,...), hầu hết $P(w_i \mid \text{Spam}) \approx 0$ (sau smoothing) → mô hình mất tác dụng.

6. Tóm lại

Tại sao bước này quan trọng?

- Word Segmentation là bước **tiền xử lý bắt buộc** khi email được viết liền.
- Nếu tách đúng → Naive Bayes hoạt động chính xác, phát hiện spam hiệu quả.
- Nếu tách sai → toàn bộ pipeline bị vô hiệu hóa vì các xác suất $P(w_i \mid \text{label})$ gần như bằng 0.

4.7. Kết hợp với pipeline mô hình phân loại

Quy trình đầy đủ:

1. Dùng mô hình **Word Segmentation** để tách chuỗi email thành các từ.
2. Dùng mô hình **Unigram + Naive Bayes** để tính xác suất mỗi nhãn (Spam/Ham).
3. Dự đoán nhãn có xác suất cao hơn.

4.8. Kết luận

- Nếu Unigram là bước để học xác suất từ, thì Word Segmentation là bước làm sạch và chuẩn hóa đầu vào để mọi thứ có thể vận hành đúng.
- Một email bị viết liền không thể phân loại đúng nếu không tách từ chính xác.
- Trong thực tế, Word Segmentation được xem là bước tiền xử lý "không thể thiếu" trong bất kỳ pipeline NLP nào.