ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA KHOA HỌC - KỸ THUẬT MÁY TÍNH

CƠ SỞ TOÁN CHO KHOA HỌC MÁY TÍNH (CO5263)

Đề tài:

Xác suất và Thống kê

GVHD: TS. Nguyễn An Khương
TS. Trần Tuấn Anh
Nhóm: 7
Học viên: Ngô Minh Đại - 2470722
Nguyễn Xuân Hiền - 2470749
Trần Đăng Hùng - 2470750
Nguyễn Đình Nhật Minh - 2370736
Trần Hoài Tâm - 2470743
Vương Minh Toàn - 2491057

TP. Hồ Chí Minh, 06/2025

Mục lục

Danh sách hình vẽ

1 Mở đầu

Học máy vốn luôn gắn liền với sự bất định: chúng ta dùng những dữ liệu đã biết (features) để dự đoán điều chưa biết (target), đồng thời muốn hiểu rõ mức độ tin cậy của dự đoán đó. Chẳng hạn, trong học có giám sát, nếu đưa vào các chỉ số huyết áp, mỡ máu và tuổi tác của một bệnh nhân, ta không chỉ muốn dự đoán xem họ có khả năng bị đau tim hay không, mà còn muốn biết xác suất cụ thể – ví dụ 10% hay 30% – để bác sĩ có hướng can thiệp phù hợp. Tùy mục tiêu, ta có thể tối ưu hóa độ chính xác (chọn giá trị có xác suất cao nhất) hoặc giảm thiểu sai số trung bình so với giá trị thực.

Trong học không giám sát, bất định giúp chúng ta phát hiện "sai lệch" (anomaly). Ví dụ, một thiết bị đo nhiệt độ môi trường liên tục gửi về dãy số, việc biết được giá trị 45°C liệu có hiếm gặp trong quá khứ hay không sẽ quyết định xem ta có gắn cờ cảnh báo hay không. Còn trong học sâu, một robot dọn nhà sẽ phải cân nhắc: nếu quét sàn (hành động A) tốn 5 phút nhưng mang lại 10 điểm sạch sẽ, trong khi lau bụi (hành động B) tốn 3 phút nhưng chỉ được 4 điểm, robot sẽ chọn chuỗi hành động tối ưu sao cho tổng "phần thưởng" là lớn nhất.

Từ những ví dụ trên, có thể thấy rõ: để hiểu và áp dụng hiệu quả các mô hình học máy, chúng ta cần nắm vững nền tảng xác suất. Tài liệu này được thiết kế nhằm cung cấp cái nhìn hệ thống và dễ tiếp cận về các khái niệm cơ bản trong xác suất học – bao gồm không gian mẫu, biến cố, biến ngẫu nhiên rời rạc và liên tục, hàm phân phối xác suất, kỳ vọng, phương sai, và những định lý quan trọng như định lý Bayes hay độc lập có điều kiện.

Tài liệu này giới thiệu các khái niệm cơ bản trong xác suất, bao gồm không gian mẫu, biến cố, biến ngẫu nhiên, hàm phân phối, kỳ vọng, phương sai, và các định lý quan trọng như Bayes. Những kiến thức này không chỉ là lý thuyết khô khan mà còn là nền tảng để xây dựng các mô hình trong học máy, ra quyết định dưới rủi ro, và hiểu rõ bản chất ngẫu nhiên trong thế giới thực.

Giờ hãy bắt đầu với **Phần 2: Khái niệm cơ bản**, nơi chúng ta sẽ khám phá nền tảng của xác suất và thống kê.

2 Khái niệm cơ bản

2.1 Xác suất

Xác suất của một sự kiện (hay một biến cố, tình huống giả định) là khả năng xảy ra của sự kiện đó, được đánh giá dưới dạng một số thực nằm giữa 0 và 1.

Khi một sự kiện không thể xảy ra, thì xác suất của nó bằng 0. Ví dụ, xác suất của sự kiện "ném một đồng xu lên mà rơi lơ lửng giữa không trung mãi mãi" là 0.

Khi một sự kiện chắc chắn xảy ra, thì xác suất của nó bằng 1 (hay còn viết là 100%). Ví dụ, xác suất của sự kiện "mặt trời mọc ở hướng đông vào sáng mai" là 1.

Khi một sự kiện có thể xảy ra hoặc không xảy ra, và chúng ta không biết chắc chắn kết quả, thì xác suất của nó nằm trong khoảng từ lớn hơn 0 đến nhỏ hơn 1. Một sự kiện càng dễ xảy ra thì xác suất của nó càng gần 1; ngược lại, nếu càng khó xảy ra thì xác suất càng gần 0. Ví dụ, giả sử tôi tham gia một trò chơi quay số may mắn mà chỉ có 1 người thắng trong số 500 người chơi. Khi đó, xác suất tôi là người thắng là $\frac{1}{500} = 0.002 = 0.2\%$.

Không chỉ các sự kiện trong tương lai, mà cả các sự kiện trong quá khứ – nếu ta không có đủ thông tin để biết chắc chắn chúng đã xảy ra hay chưa – cũng có thể được gán một xác suất nào đó, thể hiện mức độ tin tưởng của chúng ta. Ví dụ, có một giả thuyết lịch sử cho rằng danh họa Van Gogh đã tự cắt tai mình vì khủng hoảng tâm lý. Dù không ai có thể biết chắc chắn, nhưng nhiều nhà nghiên cứu đánh giá giả thuyết này có xác suất cao dựa trên bằng chứng lịch sử và thư từ để lại.

2.1.1 Ba tiên đề cơ bản (Kolmogorov)

1. **Tiên đề 1: Giới hạn xác suất**

   Xác suất của một sự kiện luôn nằm trong đoạn từ 0 đến 1:

$$0 \leq P(A) \leq 1$$

   Ví dụ 2.1:

   - Xác suất sự kiện "trái đất quay quanh mặt trời" là 1.
   - Xác suất sự kiện "một con gà biết bay như chim đại bàng" là 0.

2. **Tiên đề 2: Sự kiện và phủ định sự kiện**

Xác suất của một sự kiện, xác suất của phần bù (phủ định sự kiện) luôn có tổng bằng 1:

$$P(A) + P(\text{not } A) = 1$$

Ví dụ 2.2:

- Nếu xác suất trời mưa hôm nay là 0.6, thì xác suất trời không mưa là $1 - 0.6 = 0.4$.
- Một đồng xu khi tung lên có thể ra sấp hoặc ngửa. Nếu xác suất ra mặt sấp là 0.5 thì xác suất ra mặt ngửa cũng là 0.5.

3. **Tiên đề 3: Xác suất của các sự kiện loại trừ nhau**

Nếu hai sự kiện A và B không thể cùng xảy ra (loại trừ nhau), thì:

$$P(A \cup B) = P(A) + P(B)$$

Nếu hai sự kiện có thể xảy ra đồng thời:

$$P(A \cup B) = P(A) + P(B) - P(A \cap B)$$

Ví dụ 2.3 (loại trừ nhau):

- Một học sinh có thể đạt điểm 7 hoặc điểm 9 trong bài kiểm tra, nhưng không thể có cả hai cùng lúc. Nếu $P(7) = 0.4$ và $P(9) = 0.3$ thì:

$$P(7 \text{ hoặc } 9) = 0.4 + 0.3 = 0.7$$

Ví dụ 2.4 (có giao nhau):

- Trong một lớp học, 60% học sinh học tiếng Anh, 50% học tiếng Pháp, và 20% học cả hai. Khi đó:

$$P(\text{Anh hoặc Pháp}) = 0.60 + 0.50 - 0.20 = 0.90$$

2.1.2 Định nghĩa biến cố

Trong xác suất, một biến cố (event) A là một tập con của toàn bộ kết quả có thể, gọi là không gian mẫu Ω. Nói cách khác, biến cố là một nhóm kết quả mà chúng ta quan tâm.

Ví dụ 2.5:

- Với một đồng xu công bằng, $\Omega = \{\text{Ngửa (H)}, \text{Sấp (T)}\}$. Biến cố "ra Sấp" là $A = \{T\}$.

- Với hai đồng xu, $\Omega = \{(H, H), (H, T), (T, H), (T, T)\}$. Biến cố "có đúng một mặt Sấp" là $A = \{(H, T), (T, H)\}$.

2.1.3 Ứng dụng trong Khoa học Máy tính

Xác suất là nền tảng cho nhiều kỹ thuật và thuật toán trong computer science:

1. **Thuật toán ngẫu nhiên (Randomized Algorithms)**
   - **Randomized QuickSort:** Chọn phần tử pivot ngẫu nhiên để giảm thiểu xác suất gặp trường hợp xấu.
   - **Bloom Filter:** Dùng nhiều hàm băm giả ngẫu nhiên để kiểm tra sự tồn tại phần tử với độ sai lệch có thể tính được xác suất.
2. **Machine Learning & Thống kê**
   - **Naive Bayes:** Dùng định lý Bayes $P(C|x) \propto P(x|C)P(C)$ để phân loại nhãn dựa trên xác suất.
   - **Bayesian Networks:** Mô hình biểu diễn quan hệ phụ thuộc giữa các biến ngẫu nhiên bằng đồ thị có hướng.
3. **Mô phỏng Monte Carlo**
   - Tính gần đúng tích phân hoặc thống kê của hệ thống phức tạp (tích phân đa chiều, mô phỏng vật lý hạt).
   - Ứng dụng trong đồ họa (path tracing) và tài chính (định giá quyền chọn).
4. **Lý thuyết thông tin & mã hóa**
   - **Entropy:** $H(X) = -\sum_x P(x) \log P(x)$ đo độ không chắc chắn của biến ngẫu nhiên X.
   - **Huffman Coding:** Xây cây mã ưu tiên các kết quả có xác suất cao để giảm chiều dài trung bình mã.

Với mọi ứng dụng trên, hiểu rõ biến cố và tính chất của xác suất (như không âm, tổng bằng 1, cộng cho biến cố tách biệt) giúp phân tích, thiết kế và đánh giá hiệu quả các thuật toán và mô hình.

2.2 Biến Ngẫu Nhiên

Giả sử ta xét một biến ngẫu nhiên X. Ta giả định rằng tồn tại nhiều tình huống (hay kịch bản) khác nhau có thể xảy ra, và trong mỗi tình huống đó, X sẽ nhận một giá trị cụ thể. Do đó, ta có thể mô hình hóa biến ngẫu nhiên này như một hàm số ánh xạ từ không gian các tình huống có thể xảy ra sang tập số thực:

$$X : \Omega \to \mathbb{R}$$

Trong đó, Ω là tập hợp đại diện cho toàn bộ các tình huống (hay còn gọi là không gian mẫu). Các tình huống riêng lẻ hoặc nhóm tình huống (tập con của Ω) được gọi là **sự kiện**.

Giả sử mỗi sự kiện đều được gán một xác suất để biểu thị khả năng xảy ra của nó. Khi đó, Ω kết hợp với một độ đo xác suất P sẽ tạo thành **không gian xác suất**, ký hiệu là (Ω, P).

Chúng ta cũng giả định rằng, với mọi cặp số thực $a < b$, xác suất $P(a < X < b)$ luôn tồn tại. Nói cách khác, tập hợp

$$\{\omega \in \Omega \mid a < X(\omega) < b\}$$

phải là một tập đo được trong không gian xác suất (Ω, P). Khi điều kiện này được thỏa mãn, hàm X được gọi là **hàm đo được**.

Từ đó, ta có thể phát biểu hai định nghĩa toán học như sau:

Định nghĩa 2.1. Một biến ngẫu nhiên (random variable) với giá trị thực là một hàm số đo được trên một không gian xác suất:

$$X : (\Omega, P) \to \mathbb{R} \tag{2.1}$$

Định nghĩa 2.2. Nếu ta có hai biến ngẫu nhiên X, Y (với cùng một mô hình không gian xác suất), thì ta sẽ nói rằng $X = Y$ *theo nghĩa xác suất*, hay $X = Y$ *hầu khắp mọi nơi*, nếu như sự kiện "$X = Y$" có xác suất bằng 1 (tức là tập hợp các trường hợp mà ở đó $X \neq Y$ có xác suất bằng 0, có thể bỏ qua).

Ví dụ 2.6. Một trò chơi gồm 4 lần tung một đồng xu không đối xứng, trong đó xác suất xuất hiện mặt sấp là 0.6 và mặt ngửa là 0.4. Mỗi lần tung được xem là một phép thử độc lập.

Gọi Ω là không gian mẫu gồm tất cả các chuỗi độ dài 4 chỉ gồm ký tự S (sấp) và N (ngửa). Khi đó, Ω có $2^4 = 16$ phần tử. Ví dụ, chuỗi $SSNN$ là một phần tử trong Ω.

Xác suất của một chuỗi cụ thể được tính bằng cách nhân các xác suất thành phần. Ví dụ:

$$P(SSNN) = 0.6 \times 0.6 \times 0.4 \times 0.4 = 0.0576$$

Ta định nghĩa một biến ngẫu nhiên $X : \Omega \to \{0, 1, 2, 3, 4\}$, trong đó $X(\omega)$ là số lần xuất hiện mặt sấp trong chuỗi ω. Đây là một biến ngẫu nhiên rời rạc, và có thể mô tả phân phối xác suất của nó dựa theo phân phối nhị thức có trọng số.

Ví dụ 2.7. Giả sử B là sự kiện "xuất hiện ít nhất 3 lần mặt sấp" trong trò chơi ở Ví dụ 2.6. Khi đó, hàm chỉ báo χ_B của sự kiện B được định nghĩa như sau:

$$\chi_B(\omega) = \begin{cases} 1 & \text{nếu } \omega \in B \\ 0 & \text{nếu } \omega \notin B \end{cases}$$

Hàm χ_B là một biến ngẫu nhiên nhận hai giá trị 0 và 1, trong đó:

- $\chi_B(\omega) = 1$ nếu chuỗi ω có ít nhất 3 ký tự là S;
- $\chi_B(\omega) = 0$ nếu không.

Ngược lại, nếu ta có một biến ngẫu nhiên $Y : \Omega \to \{0, 1\}$, thì tồn tại một sự kiện $A \subset \Omega$ sao cho $Y = \chi_A$, với $A = \{\omega \in \Omega \mid Y(\omega) = 1\}$.

2.3 Đa Biến Ngẫu Nhiên

Khi xét nhiều biến ngẫu nhiên cùng lúc, chúng ta quan tâm tới mối quan hệ giữa chúng.

2.3.1 Xác suất đồng thời (Joint Probability)

Cho hai biến ngẫu nhiên A và B trên cùng một không gian xác suất $(\Omega, \mathbb{P})$. Xác suất đồng thời cho biết xác suất để A và B cùng đạt hai giá trị cụ thể đồng thời.

2.3.2.1 Trường hợp rời rạc

Giả sử A chỉ nhận giá trị trong tập đếm được $\mathcal{A}$, và B chỉ nhận giá trị trong tập đếm được $\mathcal{B}$. Khi đó, hàm khối xác suất chung (joint probability mass function) của (A, B) được định nghĩa

$$p_{A,B}(a, b) \;=\; \mathbb{P}\big(A = a,\; B = b\big), \quad a \in \mathcal{A},\; b \in \mathcal{B}.$$

Hàm khối xác suất chung phải thỏa mãn không âm và tổng bằng 1:

$$p_{A,B}(a,b) \;\geq\; 0 \quad \text{với mọi } a,b, \qquad \sum_{a\in\mathcal{A}}\sum_{b\in\mathcal{B}} p_{A,B}(a,b) \;=\; 1.$$

Từ hàm khối xác suất chung, ta có thể rút ra hàm khối xác suất biên của A bằng cách tổng qua b:

$$p_A(a) \;=\; \sum_{b\in\mathcal{B}} p_{A,B}(a,b), \qquad a\in\mathcal{A}.$$

Tương tự, hàm khối xác suất biên của B là

$$p_B(b) \;=\; \sum_{a\in\mathcal{A}} p_{A,B}(a,b), \qquad b\in\mathcal{B}.$$

Vì $\{A=a,\ B=b\} \subseteq \{A=a\}$ nên luôn có

$$p_{A,B}(a,b) \;=\; \mathbb{P}(A=a,\ B=b) \;\leq\; \mathbb{P}(A=a) \;=\; p_A(a).$$

Tương tự, $p_{A,B}(a,b) \leq p_B(b)$.

2.3.2.2 Trường hợp liên tục

Giả sử A và B là hai biến ngẫu nhiên liên tục (định nghĩa trên $\mathbb{R}$). Khi đó, tồn tại hàm mật độ chung (joint probability density function) $f_{A,B}(a,b)$ sao cho với mọi miền $D\subset\mathbb{R}^2$,

$$\mathbb{P}\big((A,B)\in D\big) \;=\; \iint_D f_{A,B}(a,b)\,\mathrm{d}a\,\mathrm{d}b.$$

Hàm mật độ này phải không âm và tích phân trên toàn $\mathbb{R}^2$ bằng 1:

$$f_{A,B}(a,b) \;\geq\; 0 \quad \text{với mọi } (a,b)\in\mathbb{R}^2, \qquad \iint_{\mathbb{R}^2} f_{A,B}(a,b)\,\mathrm{d}a\,\mathrm{d}b = 1.$$

Từ hàm mật độ chung, ta lấy mật độ biên của A bằng cách tích phân qua biến b:

$$f_A(a) \;=\; \int_{-\infty}^{+\infty} f_{A,B}(a,b)\,\mathrm{d}b, \qquad a\in\mathbb{R},$$

và mật độ biên của B bằng cách tích phân qua biến a:

$$f_B(b) \;=\; \int_{-\infty}^{+\infty} f_{A,B}(a,b)\,\mathrm{d}a, \qquad b\in\mathbb{R}.$$

Mặc dù $f_{A,B}(a,b)$ không phải là xác suất trực tiếp mà là mật độ, ta vẫn có quan hệ cận trên: vì tích phân qua một biến phải bằng mật độ biên của biến kia, nên

$$\int_{-\infty}^{+\infty} f_{A,B}(a,b)\,\mathrm{d}b \;=\; f_A(a) \;\Longrightarrow\; f_{A,B}(a,b) \;\leq\; f_A(a), \quad f_{A,B}(a,b) \;\leq\; f_B(b).$$

2.3.2 Xác suất có điều kiện và Định lý Bayes

Xác suất có điều kiện:

$$P(B=b \mid A=a) = \frac{P(A=a, B=b)}{P(A=a)}.$$

Định lý Bayes:

$$P(A \mid B) = \frac{P(B \mid A)P(A)}{P(B)}.$$

Trong thống kê Bayes:

$$P(H \mid E) = \frac{P(E \mid H)P(H)}{P(E)},$$

với $P(H)$ là tiên nghiệm, $P(H \mid E)$ là hậu nghiệm.

Ví dụ:

Chọn hai quả bóng (không hoàn lại) từ hộp có 2 bóng đỏ, 1 bóng xanh. X = màu quả 1, Y = màu quả 2.

	$Y =$ Đỏ	$Y =$ Xanh
$X =$ Đỏ	$\frac{1}{3}$	$\frac{1}{3}$
$X =$ Xanh	$\frac{1}{3}$	0

Tính:

$$P(Y = \text{Đỏ} \mid X = \text{Đỏ}) = \frac{P(X = \text{Đỏ}, Y = \text{Đỏ})}{P(X = \text{Đỏ})} = \frac{1/3}{2/3} = \frac{1}{2}.$$

2.3.3 Độc lập và Độc lập có điều kiện

1.3.4.1 Độc lập

Hai biến ngẫu nhiên A và B được gọi là **độc lập** nếu:

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

Ví dụ: Tung hai đồng xu. Gọi X là kết quả của đồng xu thứ nhất (1 nếu ngửa, 0 nếu sấp), Y là kết quả của đồng xu thứ hai. Vì hai đồng xu không ảnh hưởng nhau, ta có:

$$P(X = 1, Y = 0) = P(X = 1) \cdot P(Y = 0) = \frac{1}{2} \cdot \frac{1}{2} = \frac{1}{4}$$

1.3.4.2 Độc lập có điều kiện

Hai biến A và B được gọi là **độc lập có điều kiện theo** biến C nếu:

$$P(A, B \mid C) = P(A \mid C) \cdot P(B \mid C)$$

Ví dụ: Gọi Z là thời tiết hôm nay (ví dụ: trời mưa). Biết được Z, xác suất một người mang dù (X) và mặc áo mưa (Y) có thể được coi là độc lập, vì mỗi hành động chỉ phụ thuộc vào Z, không phụ thuộc lẫn nhau:

$$P(X, Y \mid Z) = P(X \mid Z) \cdot P(Y \mid Z)$$

2.4 Ví dụ

Chúng ta sẽ thử nghiệm một ví dụ với những nội dung đã được đề cập ở trên. Giả sử rằng một bác sĩ phụ trách xét nghiệm AIDS cho một bệnh nhân. Việc xét nghiệm này khá chính xác và nó chỉ thất bại với xác suất 1%, khi nó cho kết quả dương tính dù bệnh nhân khỏe mạnh. Hơn nữa, nó không bao giờ thất bại trong việc phát hiện HIV nếu bệnh nhân thực sự bị nhiễm bệnh. Ta sử dụng D_1 để biểu diễn kết quả chẩn đoán (1 nếu dương tính và 0 nếu âm tính) và H để biểu thị tình trạng nhiễm HIV (1 nếu dương tính và 0 nếu âm tính). Ta có liệt kê xác suất có điều kiện theo bảng dưới đây:

Bảng 1: Xác suất có điều kiện của $P(D_1 \mid H)$.

Xác suất có điều kiện	$H = 1$	$H = 0$
$P(D_1 = 1 \mid H)$	1	0.01
$P(D_1 = 0 \mid H)$	0	0.99

Ta có thể nhận thấy rằng tổng của từng cột đều bằng 1 (nhưng tổng từng hàng thì không), vì xác suất có điều kiện cần có tổng bằng 1. Hãy cùng tìm xác suất bệnh nhân bị AIDS nếu xét nghiệm trả về kết quả dương tính, tức là

$$P(H = 1 \mid D = 1)$$

Rõ ràng điều này sẽ phụ thuộc vào mức độ phổ biến của bệnh, bởi vì nó ảnh hưởng đến số lượng dương tính giả. Giả sử rằng dân số khá khỏe mạnh, ví dụ: $P(H = 1) = 0.0015$ Để áp dụng Định lý Bayes, chúng ta cần áp dụng phép biên hóa và quy tắc nhân để xác định

$$\begin{aligned} P(D_1 = 1) &= P(D_1 = 1, H = 0) + P(D_1 = 1, H = 1) \\ &= P(D_1 = 1 \mid H = 0)P(H = 0) + P(D_1 = 1 \mid H = 1)P(H = 1) \\ &= 0.01 \times 0.985 + 1 \times 0.015 \\ &= 0.011485. \end{aligned}$$

Do đó, ta có:

$$\begin{aligned} P(H = 1 \mid D_1 = 1) &= \frac{P(D_1 = 1 \mid H = 1)P(H = 1)}{P(D_1 = 1)} \\ &= \frac{1 \times 0.015}{0.011485} \\ &= 0.1306 \end{aligned}$$

Nói cách khác, chỉ có 13,06% khả năng bệnh nhân thực sự mắc bệnh AIDS, dù ta dùng một bài kiểm tra rất chính xác. Như ta có thể thấy, xác suất có thể trở nên khá phản trực giác. Một bệnh nhân phải làm gì nếu nhận được tin dữ như vậy? Nhiều khả năng họ sẽ yêu cầu bác sĩ thực hiện một xét nghiệm khác để làm rõ sự việc. Bài kiểm tra thứ hai có những đặc điểm khác và không tốt bằng bài thứ nhất, như ta có thể thấy như sau:

Bảng 2: Xác suất có điều kiện của $P(D_2 \mid H)$.

Xác suất có điều kiện	$H = 1$	$H = 0$
$P(D_2 = 1 \mid H)$	0.98	0.03
$P(D_2 = 0 \mid H)$	0.02	0.97

$$\begin{aligned} P(D_1 = 1, D_2 = 1 \mid H = 0) &= P(D_1 = 1 \mid H = 0)P(D_2 = 1 \mid H = 0) \\ &= 0.01 \times 0.03 \\ &= 0.0003, \end{aligned}$$

Không may thay, bài kiểm tra thứ hai cũng có kết quả dương tính. Hãy cùng tính các xác suất cần thiết để sử dụng định lý Bayes bằng cách giả định tính độc lập có điều kiện:

$$\begin{aligned} P(D_1 = 1, D_2 = 1 \mid H = 1) &= P(D_1 = 1 \mid H = 1)P(D_2 = 1 \mid H = 1) \\ &= 1 \times 0.98 \\ &= 0.98. \end{aligned}$$

Bây giờ chúng ta có thể áp dụng phép biến hóa và quy tắc nhân xác suất:

$$\begin{aligned} P(D_1 = 1, D_2 = 1) &= P(D_1 = 1, D_2 = 1, H = 0) + P(D_1 = 1, D_2 = 1, H = 1) \\ &= P(D_1 = 1, D_2 = 1 \mid H = 0)P(H = 0) + P(D_1 = 1, D_2 = 1 \mid H = 1)P(H = 1) \\ &= 0.0003 \times 0.985 + 0.98 \times 0.015 \\ &= 0.00176955 \end{aligned}$$

Cuối cùng xác suất bệnh nhân mắc bệnh AIDS qua hai lần dương tính là

$$\begin{aligned} P(H = 1 \mid D_1 = 1, D_2 = 1) &= \frac{P(D_1 = 1, D_2 = 1 \mid H = 1)P(H = 1)}{P(D_1 = 1, D_2 = 1)} \\ &= \frac{0.98 \times 0.015}{0.00176955} \\ &= 0.8307. \end{aligned}$$

Nhận xét: Thử nghiệm thứ hai mang lại độ tin cậy cao hơn rằng không phải mọi chuyện đều ổn. Mặc dù bài kiểm tra thứ hai kém chính xác hơn bài đầu tiên, nhưng nó vẫn cải thiện đáng kể dự đoán.

2.5 Kỳ vọng và phương sai

2.5.1 Kỳ Vọng (Expectations)

Thông thường, việc ra quyết định không chỉ yêu cầu xem xét các xác suất được gán cho từng sự kiện riêng lẻ mà còn cần tổng hợp chúng thành các đại lượng hữu ích có thể hướng dẫn ta. Ví dụ, khi các biến ngẫu nhiên nhận giá trị liên tục, chúng ta thường quan tâm đến việc biết được giá trị kỳ vọng trung bình là bao nhiêu. Đại lượng này được gọi một cách chính thức là *kỳ vọng* (expectation).

Giả sử chúng ta đang đầu tư, điều đầu tiên cần quan tâm có thể là lợi nhuận kỳ vọng – trung bình cộng tất cả các kết quả có thể xảy ra (và được cân nhắc theo xác suất tương ứng). Ví dụ, giả sử rằng với 50% xác suất, một khoản đầu tư có thể thất bại hoàn toàn, với 40% xác suất nó có thể mang lại lợi nhuận gấp 2 lần, và với 10% xác suất nó có thể mang lại lợi nhuận gấp 10 lần. Để tính lợi nhuận kỳ vọng, ta cộng tất cả các mức lợi nhuận lại, mỗi mức được nhân với xác suất xảy ra của nó:

$$0.5 \cdot 0 + 0.4 \cdot 2 + 0.1 \cdot 10 = 1.8$$

Vậy nên, lợi nhuận kỳ vọng là 1.8 lần.
Kỳ vọng của biến ngẫu nhiên rời rạc X được định nghĩa là:

$$E[X] = E_{x\sim P}[x] = \sum_x xP(X = x)$$

Giải thích: Đây là giá trị trung bình kỳ vọng của biến ngẫu nhiên rời rạc X. Mỗi giá trị có thể xảy ra của X được nhân với xác suất xảy ra của chính nó. Sau đó, các tích này được cộng lại để tính kỳ vọng. Ví dụ: Nếu một đồng xu có 50% ra sấp (giá trị 0) và 50% ra ngửa (giá trị 1), thì: $E[X] = 0 * 0.5 + 1 * 0.5 = 0.5 =>$ Trung bình kỳ vọng là 0.5. Tương tự, với biến ngẫu nhiên liên tục:

$$E[X] = \int x\, dp(x)$$

Giải thích: Khi biến ngẫu nhiên X có phân phối liên tục, ta không thể dùng tổng như trên. Thay vào đó, ta dùng tích phân của x nhân với hàm mật độ xác suất $p(x)$. Điều này tương tự như tính trung bình có trọng số, với trọng số là xác suất liên tục trên mỗi giá trị x. Khi quan tâm đến kỳ vọng của một hàm số $f(x)$:
Công thức rời rạc:

$$E_{x\sim P}[f(x)] = \sum_x f(x)P(x),$$

Công thức liên tục:

$$E_{x\sim P}[f(x)] = \int f(x)p(x)dx$$

Giải thích: Thay vì tính trung bình giá trị của X, ta tính trung bình của hàm số $f(X)$. Đây là cách đánh giá các biến đổi phi tuyến của X – ví dụ như log, bình phương, hàm lợi ích (utility), v.v. Cực kỳ quan trọng trong các bài toán như:

- Kỳ vọng lợi nhuận trong tài chính
- Kỳ vọng mất mát trong học máy

- Tính entropy, information gain trong học thống kê

Trở lại ví dụ, nếu độ hài lòng với mất trắng là -1, và các độ hài lòng tương ứng với các mức lợi nhuận 1, 2 và 10 lần là 1, 2 và 4 thì:

$$0.5 \cdot (-1) + 0.4 \cdot 2 + 0.1 \cdot 4 = 0.7$$

Nếu thực sự đây là hàm utility của bạn, bạn nên giữ tiền trong ngân hàng.

2.5.2 Phương Sai

Trong các quyết định tài chính, ta không chỉ quan tâm đến kỳ vọng mà còn đến mức độ *dao động* của các kết quả quanh kỳ vọng đó. Không thể lấy kỳ vọng của hiệu:

$$E[X - E[X]]$$

Vì:

$$E[X - E[X]] = E[X] - E[E[X]] = E[X] - E[X] = 0$$

Nên không thể dùng hiệu này để đo mức dao động; phải bình phương lên để loại bỏ dấu âm. Thay vào đó, ta xem xét kỳ vọng của bình phương hiệu:

$$\mathrm{Var}[X] = E[(X - E[X])^2] = E[X^2] - E[X]^2$$

Từ khai triển bình phương:

$$(X - E[X])^2 = X^2 - 2XE[X] + E[X]^2$$

Phương sai của hàm theo biến ngẫu nhiên:

$$\mathrm{Var}_{x\sim P}[f(x)] = E_{x\sim P}[f(x)^2] - E_{x\sim P}[f(x)]^2$$

Ví dụ: quay trở lại ví dụ về đầu tư ở trên, ta có thể tính phương sai khoản đầu tư như sau:

$$0.5 \cdot 0 + 0.4 \cdot 2^2 + 0.1 \cdot 10^2 - 1.8^2 = 8.36$$

Theo quy ước, kỳ vọng và phương sai được ký hiệu là μ và σ^2.

Giải thích: Phương sai đo mức độ dao động (phân tán) của một biến ngẫu nhiên X xung quanh kỳ vọng (trung bình) của nó Nếu phương sai nhỏ $\to$ các giá trị X hầu như gần trung bình $\to$ ít biến động Nếu phương sai lớn $\to$ các giá trị X có thể lệch xa trung bình $\to$ biến động nhiều, rủi ro cao.

2.5.3 Độ lệch chuẩn

Độ lệch chuẩn là căn bậc hai của phương sai, giúp diễn giải dễ hơn vì cùng đơn vị với biến gốc.

$$\sigma_X = \sqrt{\mathrm{Var}(X)}$$

Các tính chất của phương sai có thể được áp dụng lại cho độ lệch chuẩn:

- Với biến ngẫu nhiên X bất kỳ: $\sigma_X \geq 0$.
- Với biến ngẫu nhiên X và hằng số a, b bất kỳ: $\sigma_{aX+b} = |a|\sigma_X$
- Nếu hai biến ngẫu nhiên X và Y là độc lập: $\sigma_{X+Y} = \sqrt{\sigma_X^2 + \sigma_Y^2}$

2.5.4 Vector ngẫu nhiên và Ma trận Hiệp Phương Sai

Với biến ngẫu nhiên vector:

$$\mu = \mathbb{E}_{x\sim P}[x], \quad \mu_i = \mathbb{E}_{x\sim P}[x_i]$$

Ma trận hiệp phương sai:

$$\Sigma \stackrel{\text{def}}{=} \mathrm{Cov}_{x\sim P}[x] = \mathbb{E}_{x\sim P}[(x-\mu)(x-\mu)^\top]$$

Tác động của nó đối với một vector $\mathbf{v}$:

$$\mathbf{v}^\top \Sigma \mathbf{v} = \mathbb{E}_{x\sim P}[\mathbf{v}^\top (x-\mu)(x-\mu)^\top \mathbf{v}] = \mathrm{Var}_{x\sim P}[\mathbf{v}^\top x]$$

Các phần tử ngoài đường chéo của Σ thể hiện mức độ tương quan giữa các thành phần: giá trị 0 nghĩa là không có tương quan, còn giá trị dương lớn nghĩa là tương quan mạnh.

2.6 Thảo luận

Trong học máy, có rất nhiều điều mà chúng ta không chắc chắn. Chúng ta có thể không chắc chắn về giá trị của nhãn được gán cho một đầu vào. Chúng ta có thể không chắc chắn về giá trị ước lượng của một tham số. Chúng ta thậm chí có thể không chắc chắn liệu dữ liệu đến trong quá trình triển khai có đến từ cùng một phân phối như dữ liệu huấn luyện hay không. **Bằng sự bất định aleatoric**, chúng ta hiểu là sự không chắc chắn vốn có trong vấn đề, do tính ngẫu nhiên thật sự mà các biến quan sát được không thể giải thích hết. **Bằng sự bất định epistemic**, chúng ta hiểu là sự không chắc chắn về các tham số của mô hình – loại bất định mà ta có thể hy vọng sẽ giảm đi khi thu thập thêm dữ liệu. Chúng ta có thể có bất định epistemic liên quan đến xác suất đồng xu ra mặt ngửa, nhưng ngay cả khi đã biết xác suất đó, ta vẫn còn bất định aleatoric về kết quả của những lần tung tiếp theo. Dù chúng ta quan sát bao nhiêu, cũng không thể chắc chắn hơn hoặc kém 50% rằng lần tung tới sẽ ra mặt ngửa. Các thuật ngữ này xuất phát từ mô hình cơ học (xem ví dụ Kiureghian và Ditlevsen (2009) về khía cạnh của định lượng bất định). Cũng cần lưu ý rằng, về mặt ngôn ngữ triết học, mọi sự bất định đều là epistemic vì nó liên quan đến tri thức. Chúng ta thấy rằng việc lấy mẫu từ một phân phối xác suất không biết có thể giúp ước lượng các tham số của phân phối tạo dữ liệu. Tuy nhiên, tốc độ đạt được điều này có thể rất chậm. Trong ví dụ tung đồng xu, ta không thể làm gì hơn ngoài việc thiết kế các ước lượng hội tụ theo tốc độ:

$$\frac{1}{\sqrt{n}}$$

với n là kích thước mẫu (số lần tung). Nghĩa là khi tăng từ 10 lên 1000 lần quan sát (một việc hoàn toàn khả thi), ta giảm được độ bất định đi 10 lần, còn tăng thêm 1000 lần nữa chỉ giảm thêm được hệ số 1.41. Đây là một đặc điểm cố hữu của học máy: sau những cải thiện dễ dàng ban đầu, các bước tiến tiếp theo đòi hỏi rất nhiều dữ liệu và tính toán. Để thấy rõ điều này trong mô hình ngôn ngữ lớn, xem Revels et al. (2016). Chúng ta cũng làm rõ hơn về ngôn ngữ và công cụ mô hình thống kê. Trong quá trình đó, chúng ta học được về xác suất có điều kiện và một trong những phương trình quan trọng nhất trong thống kê – định lý Bayes. Đây là công cụ hiệu quả để tách biệt thông tin đến từ dữ liệu thông qua phân bố hậu nghiệm:

$$P(B \mid A)$$

phân bố này thể hiện mức độ dữ liệu B ủng hộ các tham số A thế nào, cùng với phân bố tiên nghiệm $P(A)$, vốn chi phối mức độ khả thi của A ban đầu. Đặc biệt, ta thấy quy tắc này có thể được dùng để gán xác suất cho các chẩn đoán dựa trên hiệu quả kiểm tra và độ phổ biến của căn bệnh (tức là phân bố tiên nghiệm).

2.6.1 Ứng dụng của bất đẳng thức Chebyshev trong XS&TK

Cuối cùng, nhóm đã giới thiệu một số câu hỏi liên quan đến một phân phối xác suất cụ thể – gồm kỳ vọng và phương sai. Dù còn nhiều thứ khác ngoài kỳ vọng tuyến tính và phương sai bậc hai, hai đại lượng này cũng cung cấp khá nhiều kiến thức. Ví dụ, **bất đẳng thức Chebyshev** phát biểu rằng:

$$P(|X-\mu| \geq k\sigma) \leq \frac{1}{k^2}$$

với μ là kỳ vọng, σ^2 là phương sai của phân phối, và $k > 1$ là tham số độ tin cậy ta chọn. Điều này cho biết các mẫu lấy từ phân phối đó sẽ nằm trong khoảng:

$$[-\sqrt{2}\sigma, \sqrt{2}\sigma]$$

với xác suất ít nhất 50%, tập trung xung quanh kỳ vọng. Bất đẳng thức Chebyshev cho phép xác định rằng hầu hết dữ liệu phải nằm gần trung bình, nếu phương sai nhỏ. Ví dụ ứng dụng trong kiểm soát chất lượng:

$$\bar{x} = 50, \quad s^2 = 4, \quad s = 2 \Rightarrow \text{Ít nhất 75\% sản phẩm nằm trong } (50 \pm 2s) = (46, 54)$$

Trong thực hành, khi không biết phân phối, có thể dùng Chebyshev để ước lượng sơ khởi khoảng tin cậy cho giá trị trung bình. Chebyshev cũng giúp đánh giá độ "phân tán"giữa các phân phối khác nhau, từ đó lựa chọn mô hình hợp lý khi làm phân tích thống kê mô tả hoặc phân tích dữ liệu.

3 Bài tập

Bài tập 1. Hãy cho một ví dụ về việc khi quan sát nhiều dữ liệu hơn có thể làm giảm lượng không chắc chắn về kết quả đầu ra xuống mức thấp một cách tùy ý. *(Nguồn: Bài tập 1 Phần 2.6.8)*

Bài giải 1.

Xét trường hợp tung một đồng xu gồm 2 mặt hình và chữ, ta không biết xác suất tung đồng xu cân bằng (nghĩa là một mặt bằng 0,5) hay bị chệch (bias). Gọi p là xác suất kết quả đầu ra (outcome) là mặt hình không chắc chắn, có giá trị trong đoạn (0,1).

Ta tiến hành quan sát dữ liệu bằng cách tung đồng xu và ghi lại kết quả mặt hình hoặc chữ. Mỗi lần tung là một phần dữ liệu.

* Tung 1 lần: Mặt hình, ước lượng p có thể là 1 (1/1) nhưng thông tin lại quá ít.

* Tung 1-10 lần: ta được 7 lần mặt hình trong 10 lần tung, ước lượng $p = 0{,}7 < 1$ (7/10). Tuy nhiên, sự không chắc chắn vẫn còn đáng kể.

* Tung 1-100 lần: ta được 58 lần mặt hình trong 100 lần tung, ước lượng $p = 0{,}58$ (58/100). Con số này có thể gần với p thực hơn là 0,7. Phạm vi các giá trị hợp lý cho p dựa trên dữ liệu này đang bắt đầu thu hẹp.

* Tung 1-1000 lần: ta được 523 lần mặt hình trong 1000 lần tung, ước lượng $p = 0{,}523$, gần với p thực hơn. Tung 1-10.000 lần: ta được 5098 lần mặt hình trong 10.000 lần tung, ước lượng $p = 0{,}5098$.

* Tung 1-1.000.000 lần: ta được 500.350 lần mặt hình trong số 1.000.000 lần tung, ước lượng $p = 0{,}50035$.

* Giảm độ không chắc chắn: Khi ta thu thập ngày càng nhiều dữ liệu (thực hiện nhiều lần tung hơn), tỷ lệ mẫu về mặt hình (số lần mặt hình chia cho tổng số lần tung) trở thành ước lượng ngày càng đáng tin cậy hơn về xác suất thực p. Nguyên tắc thống kê chính được áp dụng trường hợp này là Luật số lớn, Law of Large Numbers. Nguyên tắc này chỉ ra rằng khi số lần thử tăng lên, giá trị trung bình của các kết quả thu được từ một số lượng lớn các lần thử sẽ gần với giá trị mong đợi (hay trung bình) và sẽ có xu hướng gần hơn khi thực hiện nhiều lần thử hơn. Trong trường hợp này, tỷ lệ mẫu của mặt hình hội tụ về xác suất thực p.

* Mức thấp tùy ý: Sự không chắc chắn về p có thể được định lượng bằng các khái niệm như khoảng tin cậy. Khoảng tin cậy cho p đưa ra một phạm vi giá trị mà p thực có khả năng nằm

trong đó, dựa trên dữ liệu quan sát được. Khi số lần tung (n) tăng lên, độ rộng của khoảng tin cậy này giảm xuống. Về mặt lý thuyết, khi n tiến tới vô cực, độ rộng của khoảng tin cậy tiến tới 0. Điều này có nghĩa là bằng cách thu thập một lượng dữ liệu đủ lớn, ta có thể giảm sự không chắc chắn về xác suất thực p xuống mức thấp tùy ý như kỳ vọng.

Mã nguồn minh hoạ: Exercise1.ipynb

```python
# Bài tập 1 (Nguồn: Bài tập 1 Phần 2.6.8):
## Yêu cầu: Giảm sự bất định tuỳ ý trong ước lượng xác suất.
## Các giả định:
## Tung một đồng xu, xác suất thu được mặt Hình (head).
true_prob_heads = 0.55
print("Bài tập 1: [bold yellow]Xác suất tung đồng xu")
print(f"Xác suất thu được mặt Hình: {true_prob_heads}","\n")
## Các thử nghiệm với số lượng phép thử tung đồng xu khác nhau.
sample_sizes = [10, 100, 1000, 10000, 100000]
## n_flips - số lần tung đồng xu.
## Kết quả mặt Hình (head) là 1, mặt chữ là 0 (tail).
## Thêm thư viện cần thiết.
## Thư viện định dạng chuỗi.
from rich import print
## Thư viện xử lý dữ liệu và thống kê.
import numpy as np
## Thư viện thống kê.
from statsmodels.stats.proportion import proportion_confint
from scipy import stats
## Tiến hành thử nghiệm.
for n_flips in sample_sizes:
    ## Thực hiện tung đồng xu ngẫu nhiên.
    n_heads = np.random.binomial(n_flips, true_prob_heads, 1)[0]
    ## Tính xác suất thu được mặt Hình.
    sample_prob_heads = n_heads / n_flips
    ## Tính khoảng tin cậy cho xác suất thu được mặt Hình bằng phương pháp
    ↪ khoảng điểm Wilson.
```

```
    conf_interval = proportion_confint(n_heads, n_flips, alpha=0.05,
    ↪  method='wilson')
    ## Tính độ rộng của khoảng độ tin cậy.
    interval_width = conf_interval[1] - conf_interval[0]
    print(f"  Số lần tung đồng xu    : {n_flips}")
    print(f"  Số lần thu mặt Hình    : {n_heads}")
    print(f"  Ước lượng mẫu          : {sample_prob_heads:.4f}")
    print(f"  Khoảng tin cậy         : ({conf_interval[0]:.4f},
    ↪  {conf_interval[1]:.4f})")
    print(f"  Độ rộng khoảng tin cậy: {interval_width:.4f}","\n")

print("[bold yellow]Nhận xét:")
print("* Khi số lần thử tăng lên, độ rộng của khoảng tin cậy sẽ giảm xuống.")
print("* Sự không chắc chắn của xác suất thực thu được mặt Hình giảm và nhỏ")
print("dần tuỳ ý với dữ liệu vô hạn.")
print("Theo [bold cyan]Luật số lớn - The law of large numbers.")

Kết quả

Xác suất thu được mặt Hình: 0.55

  Số lần tung đồng xu    : 10
  Số lần thu mặt Hình    : 7
  Ước lượng mẫu          : 0.7000
  Khoảng tin cậy         : (0.3968, 0.8922)
  Độ rộng khoảng tin cậy: 0.4954

  Số lần tung đồng xu    : 100
  Số lần thu mặt Hình    : 46
  Ước lượng mẫu          : 0.4600
  Khoảng tin cậy         : (0.3656, 0.5574)
  Độ rộng khoảng tin cậy: 0.1917
```

```

  Số lần tung đồng xu    : 1000
  Số lần thu mặt Hình    : 535
  Ước lượng mẫu          : 0.5350
  Khoảng tin cậy         : (0.5040, 0.5657)
  Độ rộng khoảng tin cậy: 0.0617

  Số lần tung đồng xu    : 10000
  Số lần thu mặt Hình    : 5482
  Ước lượng mẫu          : 0.5482
  Khoảng tin cậy         : (0.5384, 0.5579)
  Độ rộng khoảng tin cậy: 0.0195

  Số lần tung đồng xu    : 100000
  Số lần thu mặt Hình    : 55080
  Ước lượng mẫu          : 0.5508
  Khoảng tin cậy         : (0.5477, 0.5539)
  Độ rộng khoảng tin cậy: 0.0062

Nhận xét:
* Khi số lần thử tăng lên, độ rộng của khoảng tin cậy sẽ giảm xuống.
* Sự không chắc chắn của xác suất thực thu được mặt Hình giảm và nhỏ dần tuỳ ý với dữ liệu vô hạn.
Theo Luật số lớn - The law of large numbers.
```

Bài tập 2. Hãy cho một ví dụ về việc giảm lượng bất định đến một điểm nào đó khi tăng quan sát nhiều dữ liệu hơn. Giải thích vì sao chúng ta muốn xác định được điểm kỳ vọng này. *(Nguồn: Bài tập 2 Phần 2.6.8)*

Bài giải 2.

Ví dụ: Giả sử ta mong muốn đo chính xác chiều dài thực tế của một cái bàn với thước kẻ chuẩn.

* "Kết quả"của sự không chắc chắn: là chiều dài thực, chính xác của chiếc bàn.

* Quan sát dữ liệu: ta đo chiều dài của chiếc bàn nhiều lần bằng thước nhựa chuẩn.

* Giảm sự không chắc chắn ban đầu (Thống kê):

- Phép đo đầu tiên có thể cho kết quả 150,3 cm. Những thay đổi nhỏ trong cách đọc vạch do cách đặt thước kẻ hay góc nhìn hoặc thậm chí là những biến động nhỏ về nhiệt độ gây ra sự giãn nở/co lại cho mỗi phép đo sẽ tạo một số lỗi ngẫu nhiên.

- Ta thực hiện tiếp phép đo thứ hai và thứ ba được kết quả lần lượt là 150,1 cm và 150,4 cm. Bằng cách thực hiện nhiều phép đo (thu thập thêm dữ liệu) và tính toán giá trị trung bình, khi đó tác động của những lỗi đo ngẫu nhiên sẽ giảm. Giá trị trung bình của một số phép đo thường là ước tính tốt hơn về chiều dài thực so với bất kỳ phép đo đơn lẻ nào. Càng đo nhiều, giá trị trung bình của các phép đo càng có khả năng gần với chiều dài thực, nếu chỉ có lỗi ngẫu nhiên. Điều này tương tự như ví dụ tung đồng xu là tính trung bình giúp giảm sự không chắc chắn do biến động ngẫu nhiên trong các quan sát gây ra. Sự không chắc chắn về mặt thống kê của phép đo trung bình giảm khi số lần đo nhiều hơn (tỷ lệ thuận với $1/\sqrt{n}$, trong đó n là số phép đo).

* Điểm cao (do giới hạn về phép đo): Thước nhựa tiêu chuẩn thường có những hạn chế như:

- Các vạch chia có thể chỉ chính xác đến 1,0 mm hay 0,1 cm. Ta không thể đọc chính xác với độ hoàn hảo thực tế ngoài độ phân giải này. Ngoài ra, có một lỗi hệ thống nhỏ trong chính thước đo (ví dụ: thước được sản xuất quá ngắn hoặc quá dài). Phương pháp đặt thước khi đo liên tục có thể tạo ra một khoảng cách nhỏ hoặc chồng chéo khi bắt đầu.

- Đây là những lỗi hệ thống hoặc lỗi dụng cụ, không phải là lỗi hoàn toàn ngẫu nhiên được tính trung bình qua nhiều lần thử. Việc tăng lượng phép thử cũng không thể khắc phục được những hạn chế cố hữu này.

- Nếu độ chính xác của thước đo chỉ đến mm gần nhất, thì việc thực hiện một triệu phép đo sẽ không mang lại cho bạn độ chính xác đến từng nm. Giá trị trung bình có thể hội tụ đến một giá trị như 150,28 cm, nhưng không thể chắc chắn về vị trí phần trăm hoặc phần nghìn vì công cụ và kỹ thuật không cung cấp mức độ chi tiết hoặc loại bỏ độ lệch hệ thống.

* Nguyên nhân xảy ra tình trạng ổn định và định vị:

Tình trạng ổn định trong quá trình giảm độ không chắc chắn xảy ra vì khi đạt đến điểm mà độ không chắc chắn còn lại bị chi phối bởi các hạn chế vốn có của công cụ và phương pháp đo lường (ví dụ: độ phân giải, lỗi hiệu chuẩn, lỗi ứng dụng hệ thống), thay vì sự thay đổi ngẫu nhiên khi thực hiện các phép đo riêng lẻ.

* Ban đầu, việc tăng lượng dữ liệu sẽ làm giảm độ không chắc chắn một cách hiệu quả bằng cách tính trung bình nhiễu ngẫu nhiên.

* "Điểm"mà tình trạng ổn định bắt đầu trở nên đáng chú ý là khi độ không chắc chắn về mặt thống kê (giảm dần theo nhiều dữ liệu hơn) trở nên nhỏ so với độ không chắc chắn tối thiểu không thể giảm được do độ chính xác của hệ thống đo lường và các sai lệch hệ thống tiềm ẩn gây ra.

Ngay cả với dữ liệu vô hạn, ước lượng về chiều dài thực của bàn sẽ chỉ đáng tin cậy đến giới hạn độ chính xác của thước đo (có thể là ±0,05 cm nếu đọc đến mm gần nhất) hoặc độ lớn của bất kỳ độ lệch hệ thống. Ta không thể sử dụng phương pháp này để xác định chiều dài với độ chính xác tùy ý (ví dụ: đến nm gần nhất), bất kể thực hiện bao nhiêu phép đo. Để giảm thêm sự không chắc chắn, ta sẽ cần một dụng cụ đo chính xác hơn (như máy đo khoảng cách tia X) hoặc một phương pháp chính xác hơn, bao gồm một tập hợp các giới hạn tiềm ẩn khác.

Mã nguồn minh hoạ: Exercise2.ipynb

```python
# Bài tập 2 (Nguồn: Bài tập 2 Phần 2.6.8):
## Yêu cầu: Các mức độ không chắc chắn trong phép đo chiều dài có sai số.
## Các giả định:
## Đo chiều dài của một cái bàn bằng thước đo bằng nhựa có độ phân giải mm.
## Thêm thư viện cần thiết.
## Thư viện định dạng chuỗi.
from rich import print
## Thư viện xử lý dữ liệu và thống kê.
import numpy as np
## Thư viện thống kê.
from statsmodels.stats.proportion import proportion_confint
from scipy import stats
## Định nghĩa chiều dài thực của một cái bàn (cm).
true_length = 150.285
## Định nghĩa các đặc tính của sai số trong đo đạc.
## Độ lệch chuẩn của biến động ngẫu nhiên (cm).
random_error_std_dev = 0.2
## Sai số hệ thống của thiết bị đo có vạch chia 0.1 cm.
```

```python
systematic_error = 0.1
## Giả định giới hạn độ phân giải cùng sai số thước đo 0.1 cm.
resolution = 0.1
print("Bài tập 2: [bold yellow]Đo chiều dài với sai số","\n")
print(f"Độ dài thực                        : {true_length} cm")
print(f"Độ lệch chuẩn của sai số ngẫu nhiên: {random_error_std_dev} cm")
print(f"Sai số hệ thống                    : {systematic_error} cm")
print(f"Độ phân giải đo lường              : {resolution} cm","\n")
## Các thử nghiệm với số lượng phép đo khác nhau.
num_measurements_list = [1, 10, 100, 1000, 10000, 100000]
## Tiến hành phép thử.
for n_measurements in num_measurements_list:
    random_errors = np.random.normal(0, random_error_std_dev, n_measurements)
    ## Mỗi lần đo = Chiều dài thực + sai số ngẫu nhiên + sai số hệ thống.
    raw_measurements = true_length + random_errors + systematic_error
    ## Làm tròn đến đơn vị độ phân giải gần nhất.
    simulated_measurements = np.round(raw_measurements / resolution) *
    ↪  resolution
    ## Tính giá trị trung bình của phép đo.
    average_measurement = np.mean(simulated_measurements)
    ## Tính độ lệch chuẩn của trung bình, standard error of the mean (SEM).
    ## Với độ lệch chuẩn mẫu là Sample_standard_deviation
    ## SEM = sample_standard_deviation / sqrt(n)
    ## SEM phản ánh độ biến động của trung bình mẫu bởi sai số ngẫu nhiên.
    sample_std_dev = np.std(simulated_measurements, ddof=1)
    sem = sample_std_dev / np.sqrt(n_measurements)
    ## Tính khoảng tin cậy cho trung bình thật của các giá trị đo được.
    ## Chú ý: Khoảng này ước lượng giá trị trung bình có được với số lần đo
    ↪  không giới hạn.
    ## Khoảng tin cậy tập trung quanh giá trị trung bình của lần đo.
    t_critical = stats.t.ppf(1 - 0.05/2, df=n_measurements-1) if
    ↪  n_measurements > 1 else np.inf
```

```python
    margin_of_error = t_critical * sem if n_measurements > 1 else np.inf
    conf_interval_mean = (average_measurement - margin_of_error, average_measurement + margin_of_error)
    ## Tính độ rộng của khoảng tin cậy.
    interval_width_mean = conf_interval_mean[1] - conf_interval_mean[0] if n_measurements > 1 else np.inf
    print(f"Số lần đo                          : {n_measurements}")
    print(f"Giá trị đo trung bình              : {average_measurement:.4f} cm")
    if n_measurements > 1:
        print(f"Phương sai mẫu                     : {sample_std_dev:.4f} cm")
        print(f"Sai số chuẩn của trung bình (SEM)  : {sem:.4f} cm")
        print(f"Khoảng tin cậy cho trung bình đo   : ({conf_interval_mean[0]:.4f}, {conf_interval_mean[1]:.4f})")
        print(f"Độ rộng khoảng tin cậy             : {interval_width_mean:.4f}","\n")
    else:
        print(f"Khoảng tin cậy cho trung bình đo   : Không tính cho một lần đo.")
        print(f"Độ rộng khoảng tin cậy             : Không tính.","\n")

print("[bold yellow]Nhận xét:")
print("Trước hết, độ rộng khoảng tin cậy cho trung bình phép đo giảm dần khi các lỗi ngẫu nhiên được tính trung bình.")
print("Tuy nhiên, khi số lượng phép đo trở nên rất lớn, độ rộng khoảng cách sẽ ổn định.")
print("Điều này là do lỗi hệ thống và giới hạn độ phân giải của thước đo trở thành những yếu tố chi phối.")
print("Phép đo trung bình hội tụ về phía chiều dài thực cộng với lỗi hệ thống, làm tròn theo độ phân giải.")
print("Không thể giảm sự không chắc chắn xuống dưới mức sai số do lỗi hệ thống và độ phân giải gây ra,")
```

```
print("ngay cả khi có dữ liệu vô hạn khi sử dụng phương pháp đo lường cụ thể
↪ này.")

* Kết quả:
Bài tập 2: Đo chiều dài với sai số

Độ dài thực                        : 150.285 cm
Độ lệch chuẩn của sai số ngẫu nhiên: 0.2 cm
Sai số hệ thống                    : 0.1 cm
Độ phân giải đo lường               : 0.1 cm

Số lần đo                          : 1
Giá trị đo trung bình              : 150.4000 cm
Khoảng tin cậy cho trung bình đo   : Không tính cho một lần đo.
Độ rộng khoảng tin cậy             : Không tính.

Số lần đo                          : 10
Giá trị đo trung bình              : 150.4000 cm
Phương sai mẫu                     : 0.1563 cm
Sai số chuẩn của trung bình (SEM)  : 0.0494 cm
Khoảng tin cậy cho trung bình đo   : (150.2882, 150.5118)
Độ rộng khoảng tin cậy             : 0.2237

Số lần đo                          : 100
Giá trị đo trung bình              : 150.3420 cm
Phương sai mẫu                     : 0.1950 cm
Sai số chuẩn của trung bình (SEM)  : 0.0195 cm
Khoảng tin cậy cho trung bình đo   : (150.3033, 150.3807)
Độ rộng khoảng tin cậy             : 0.0774

Số lần đo                          : 1000
Giá trị đo trung bình              : 150.3801 cm
```

```
Phương sai mẫu                      : 0.1965 cm
Sai số chuẩn của trung bình (SEM)   : 0.0062 cm
Khoảng tin cậy cho trung bình đo    : (150.3679, 150.3923)
Độ rộng khoảng tin cậy              : 0.0244

Số lần đo                           : 10000
Giá trị đo trung bình               : 150.3833 cm
Phương sai mẫu                      : 0.2039 cm
Sai số chuẩn của trung bình (SEM)   : 0.0020 cm
Khoảng tin cậy cho trung bình đo    : (150.3793, 150.3873)
Độ rộng khoảng tin cậy              : 0.0080

Số lần đo                           : 100000
Giá trị đo trung bình               : 150.3848 cm
Phương sai mẫu                      : 0.2018 cm
Sai số chuẩn của trung bình (SEM)   : 0.0006 cm
Khoảng tin cậy cho trung bình đo    : (150.3836, 150.3861)
Độ rộng khoảng tin cậy              : 0.0025

Nhận xét:
Trước hết, độ rộng khoảng tin cậy cho trung bình phép đo giảm dần khi các lỗi ngẫu nhiên được tính trung bình. Tuy nhiên, khi số lượng phép đo trở nên rất lớn, độ rộng khoảng cách sẽ ổn định. Điều này là do lỗi hệ thống và giới hạn độ phân giải của thước đo trở thành những yếu tố chi phối. Phép đo trung bình hội tụ về phía chiều dài thực cộng với lỗi hệ thống, làm tròn theo độ phân giải. Không thể giảm sự không chắc chắn xuống dưới mức sai số do lỗi hệ thống và độ phân giải gây ra, ngay cả khi có dữ liệu vô hạn khi sử dụng phương pháp đo lường cụ thể này.
```

Bài tập 3. Chứng minh bằng thực nghiệm sự hội tụ đến giá trị trung bình cho phép tung đồng xu. Tính toán phương sai của ước lượng xác suất nhìn thấy mặt hình sau khi tung n lần.

1. Phương sai tỷ lệ với số lượng quan sát như thế nào?

2. Sử dụng bất đẳng thức Chebyshev để giới hạn độ lệch khỏi kỳ vọng.

3. Mối liên quan đến định lý giới hạn tập trung, *central limit theorem*?

(Nguồn: Bài tập 3 Phần 2.6.8)

Bài giải 3.

Phân tích phương sai của ước lượng xác suất nhìn thấy mặt hình khi tung đồng xu n lần.

Gọi X_i là biến ngẫu nhiên hiển thị kết quả lần tung thứ i:

$X_i = 1$ nếu lần tung thứ i là mặt hình.

$X_i = 0$ nếu lần tung thứ i là mặt chữ.

Giả định mỗi lần tung đồng xu là độc lập và phân phối như nhau.

Gọi p là xác suất thực để nhận được mặt hình cho một lần tung.

Giá trị kỳ vọng của mỗi lần tung là: $E[X_i] = 1.p + 0.(1-p) = p$ (1)

Phương sai của một lần tung (một biến ngẫu nhiên *Bernuoulli*) là: $Var(X_i) = p.(1-p)$ (2)

Sau n lần tung, tổng số lần xuất hiện mặt hình là: $S_n = \sum_{i=1}^{n} X_i$

Ước lượng về xác suất hiện mặt hình là tỷ lệ mẫu: $\hat{p}_n = \frac{S_n}{n} = \frac{1}{n}\sum_{i=1}^{n} X_i$

Áp dụng tính chất của phương sai đối với biến ngẫu nhiên độc lập: $Var(cY) = c^2 Var(Y)$ và $Var(\sum_{i=1}^{n} Y_i) = \sum_{i=1}^{n}[Var(Y_i)]$.

Tính phương sai của ước lượng này:

$Var(\hat{p}_n) = Var(\frac{1}{n}\sum_{i=1}^{n} X_i) = \frac{1}{n^2}\sum_{i=1}^{n}[Var(X_i)]$ (3)

Thay (2) vào (3) ta được:

$Var(\hat{p_n}) = \frac{1}{n^2}\sum_{i=1}^{n} .p.(1-p) = \frac{1}{n^2}.n.p.(1-p) = \frac{p.(1-p)}{n}$ (4)

1. Sự thay đổi của phương sai theo số lượng quan sát.

Từ phương trình (4) ta có phương sai tỷ lệ mẫu:

$Var(\hat{p_n}) = \frac{p.(1-p)}{n}$

Vì p.(1-p) là hằng số, được xác định bởi xác suất thực sự của đồng xu, nên phương sai ***tỷ lệ nghịch*** với số quan sát n. Nghĩa là khi tăng lượng quan sát lên thì phương sai sẽ nhỏ hay độ chính xác sẽ tăng lên.

2. Sử dụng bất đẳng thức Chebyshev để chặn độ lệch so với giá trị kỳ vọng.

Bất đẳng thức Chebyshev phát biểu rằng đối với bất kỳ biến ngẫu nhiên Y có giá trị kỳ vọng μ, phương sai σ^2 và $\epsilon > 0$ thì:

$P(|Y - \mu| \geq \epsilon) \leq \frac{\sigma^2}{\epsilon^2}$ (5)

Trường hợp biến ngẫu nhiên là $\hat{p}_n$ thì kỳ vọng là $E[\hat{p}_n] = p$ và phương sai là $Var(\hat{p_n}) = \frac{p.(1-p)}{n}$. Áp dụng bất đẳng thức Chebyshev, từ bất đẳng thức (5), ta được:

$P(|Y - \mu| \geq \epsilon) \leq \frac{\sigma^2}{\epsilon^2} = \frac{p.(1-p)}{n.\epsilon^2}$ (6)

Từ bất đẳng thức (6) và theo Định luật số lớn (*Law of Large Numbers*) cho thấy khi n lớn hay $n \to \infty$ thì $\hat{p}_n$ hội tụ dần về p.

3. Mối liên hệ với định lý giới hạn trung tâm.

Định lý Giới hạn tập trung, *Central Limit theorem*, đưa ra một kết quả vững chắc hơn bất đẳng thức Chebyshev đối với n lớn. Trong khi Chebyshev cung cấp một chặn tổng quát về xác suất lệch cho bất kỳ phân phối nào, định lý cho chúng ta biết về hình dạng của phân phối của giá trị trung bình mẫu (hoặc tỷ lệ) khi kích thước mẫu lớn.

Định lý Giới hạn tập trung phát biểu rằng tổng (hoặc trung bình) của một số lượng lớn các biến ngẫu nhiên, mỗi biến có giá trị kỳ vọng và phương sai hữu hạn, sẽ có phân phối xấp xỉ chuẩn.

Cụ thể, đối với tỷ lệ mẫu $\hat{p}_n$, khi n lớn, phân phối $\hat{p}_n$ xấp xỉ phân phối chuẩn với:

* Giá trị kỳ vọng: $E[\hat{p}_n] = p$

* Phương sai: $Var(\hat{p}_n) = \frac{p(1-p)}{n}$

* Độ lệch chuẩn (Sai số chuẩn), *Standard Deviation* (*Standard Error*):

$SE(\hat{p}_n) = \sqrt{\frac{p.(1-p)}{n}}$

Kết luận:

* Phương sai $\frac{p.(1-p)}{n}$ định lượng độ phân tán của tỷ lệ mẫu xung quanh xác suất thực sự.

* Phương sai này tỷ lệ nghịch theo n, nghĩa là sự không chắc chắn giảm khi có nhiều dữ liệu hơn.

* Bất đẳng thức Chebyshev sử dụng phương sai này để đưa ra một chặn trên được đảm bảo cho xác suất của các độ lệch lớn càng nhỏ khi n lớn.

Định lý Giới hạn tập trung phát biểu rằng với n lớn, phân phối mẫu của tỷ lệ mẫu xấp xỉ là phân phối chuẩn, với giá trị kỳ vọng p và phương sai $\frac{p.(1-p)}{n}$. Tính toán phương sai là một thành phần quan trọng được định lý này sử dụng để mô tả các đặc tính của phân phối mẫu.

Mã nguồn minh hoạ: Exercise3.ipynb

```python
# Bài tập 3 (Nguồn: Bài tập 3 Phần 2.6.8):
## Yêu cầu: Phân tích phương sai của phép ước lượng xác suất.
## Thêm thư viện cần thiết.
## Thư viện định dạng chuỗi.
from rich import print
## Thư viện xử lý dữ liệu và thống kê.
import numpy as np
## Thư viện trực quan hoá bằng đồ thị.
import matplotlib.pyplot as plt
import math
## Các tham số ban đầu cho bài toán tung đồng xu.
## Xác suất thực xuất hiện mặt Hình (head).
true_prob_heads = 0.6
## Các thử nghiệm với số lượng phép đo khác nhau.
sample_sizes = [10, 50, 100, 500, 1000, 5000]
## Số lần lặp lại thử nghiệm cho phương sai thực nghiệm.
num_simulations = 10000
## Đặt tham số epsilon cho bất đẳng thức Chebyshev (độ lệch khỏi trung bình).
epsilon = 0.05
## Tính toán thực nghiệm.
print(f"Xác suất thực của mặt Hình: {true_prob_heads}")
print("-" * 40)
## Phương sai mẫu thực nghiệm = p(1-p)/n.
print("Phương sai mẫu thực nghiệm:")
theoretical_variances = {}
for n in sample_sizes:
    theoretical_variance = (true_prob_heads * (1 - true_prob_heads)) / n
    theoretical_variances[n] = theoretical_variance
    print(f"  n = {n}: Variance = {theoretical_variance:.6f}")
print("-" * 40)
## Mô phỏng và tính toán thực nghiệm.
print("Phương sai mẫu thực nghiệm:")
```

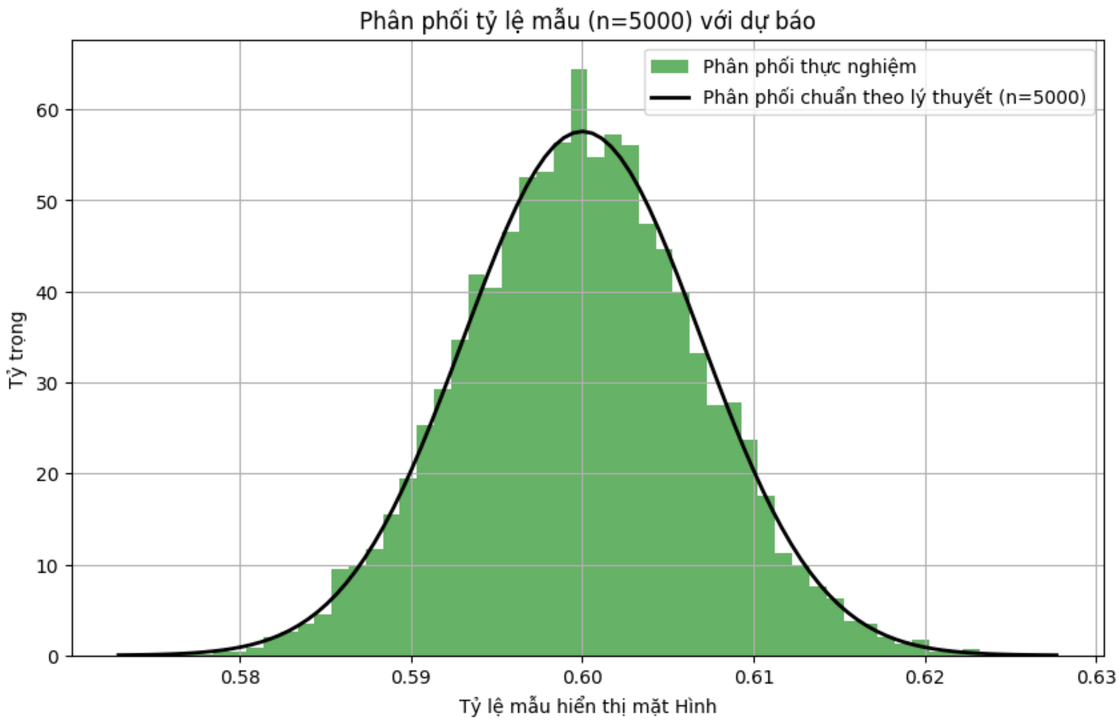
```python
sample_proportions_for_clt = {}
for n in sample_sizes:
    ## Chạy nhiều mô phỏng cho mỗi cỡ mẫu.
    sample_proportions = []
    deviations_greater_than_epsilon = 0
    for _ in range(num_simulations):
        ## Xét giá trị mặt Hình là 1, mặt chữ là 0.
        ## Tạo n số ngẫu nhiên thuộc (0.1).
        outcomes = (np.random.rand(n) < true_prob_heads).astype(int)
        ## Tính số lần hiện mặt Hình.
        n_heads = np.sum(outcomes)
        ## Tính tỷ lệ mẫu.
        sample_prop = n_heads / n
        ## Lưu tỷ lệ mẫu.
        sample_proportions.append(sample_prop)
        ## Kiểm tra độ lệch cho bất đẳng thức Chebyshev.
        if abs(sample_prop - true_prob_heads) >= epsilon:
            deviations_greater_than_epsilon += 1
        ## Tính phương sai từ xác suất mẫu.
    empirical_variance = np.var(sample_proportions, ddof=0)
    print(f"  n = {n}: Phương sai thực nghiệm = {empirical_variance:.6f}")
    ## Lưu các tỷ lệ mẫu đối với mẫu có kích thước lớn nhất để trực quan hoá.
    if n == sample_sizes[-1]:
        sample_proportions_for_clt[n] = sample_proportions
    ## Trình bày bất phương trình Chebyshev.
    ## Tính độ lệch xác suất thực nghiệm.
    empirical_prob_deviation = deviations_greater_than_epsilon /
    ↪ num_simulations
    ## Tính cận trên của bất đẳng thức Chebyshev.
    chebyshev_bound = theoretical_variances[n] / (epsilon**2)
    print(f"Bất đẳng thức Chebyshev (epsilon = {epsilon}):")
    ## Y là tỷ lệ mẫu.
```

```
    print(f"P(|Y - p| >= {epsilon}) <= {chebyshev_bound:.4f}")
    print(f"Độ lệch xác suất thực nghiệm: {empirical_prob_deviation:.4f}")
    ## Khẳng định rằng xác suất thực nghiệm nhỏ hơn hoặc bằng giới hạn (trong phạm vi nhiễu mô phỏng epsilon),
    ## mặt khác, sử dụng dung sai nhỏ vì mô phỏng không hoàn hảo.
    assert empirical_prob_deviation <= chebyshev_bound + 0.01, f"cận trên vi phạm với mẫu n={n}"
    print(f"(Xác suất thực nghiệm kỳ vọng nhỏ hơn hoặc bằng cận trên)","\n")
print("-" * 40)
## Mối liên hệ với Định lý giới hạn tập trung.
print("Trực quan hoá phân phối tỷ lệ mẫu cho trường hợp cỡ mẫu lớn nhất:")
## Chọn tỷ lệ mẫu cho cỡ mẫu lớn nhất.
largest_n = sample_sizes[-1]
sample_props_clt = sample_proportions_for_clt[largest_n]
## Vẽ đồ thị cho tỷ lệ mẫu.
plt.figure(figsize=(10, 6))
plt.hist(sample_props_clt, bins=50, density=True, alpha=0.6, color='g', label='Phân phối thực nghiệm')
## Đưa đồ thị phân phối chuẩn theo lý thuyết được dự đoán.
## Trung bình: Mean = true_prob_heads.
## Độ lệch chuẩn: Standard Deviation = sqrt(theoretical_variance for largest_n).
clt_mean = true_prob_heads
clt_std_dev = math.sqrt(theoretical_variances[largest_n])
## Tạo một tập mẫu (ảo) biểu diễn đường cong phân phối chuẩn đồ thị Create a range of x values for the normal distribution curve.
xmin, xmax = plt.xlim()
x = np.linspace(xmin, xmax, 100)
p_normal = (1 / (clt_std_dev * np.sqrt(2 * np.pi))) * np.exp(-((x - clt_mean)**2) / (2 * clt_std_dev**2))
## Biểu diễn đồ thị.
plt.plot(x, p_normal, 'k', linewidth=2, label=f'Phân phối chuẩn theo lý thuyết (n={largest_n})')
```

```
plt.title(f'Phân phối tỷ lệ mẫu (n={largest_n}) với dự báo')
plt.xlabel('Tỷ lệ mẫu hiển thị mặt Hình')
plt.ylabel('Tỷ trọng')
plt.legend()
plt.grid(True)
plt.show()
print("\n","[bold yellow]Nhận xét:")
print("Đồ thị tần suất tỷ lệ mẫu cho kích thước mẫu lớn (n = 5000) có hình
    ↪ dạng gần giống đồ thị phân phối chuẩn.")
print("Đỉnh của biểu đồ nằm xung quanh xác suất thực của mặt Hình (0.6).")
print("Độ phân tán của biểu đồ liên quan đến phương sai lý thuyết (và sai số
    ↪ chuẩn) cho quy mô mẫu đó.")
print("Đồ thị này minh họa trực quan Định lý giới hạn tập trung:")
print("Sự phân phối của các giá trị trung bình hay tỷ lệ mẫu tiến tới phân
    ↪ phối chuẩn khi quy mô mẫu tăng,")
print("với giá trị trung bình bằng giá trị trung bình thực và phương sai bằng
    ↪ giá trị phương sai thực chia cho quy mô mẫu.")

Kết quả

Xác suất thực của mặt Hình: 0.6
----------------------------------------
Phương sai mẫu thực nghiệm:
n = 10: Variance = 0.024000
n = 50: Variance = 0.004800
n = 100: Variance = 0.002400
n = 500: Variance = 0.000480
n = 1000: Variance = 0.000240
n = 5000: Variance = 0.000048
----------------------------------------
Phương sai mẫu thực nghiệm:
n = 10: Phương sai thực nghiệm = 0.024075
```

```
Bất đẳng thức Chebyshev (epsilon = 0.05):
P(|Y - p| >= 0.05) <= 9.6000
Độ lệch xác suất thực nghiệm: 0.7490
(Xác suất thực nghiệm kỳ vọng nhỏ hơn hoặc bằng cận trên)

n = 50: Phương sai thực nghiệm = 0.004875
Bất đẳng thức Chebyshev (epsilon = 0.05):
P(|Y - p| >= 0.05) <= 1.9200
Độ lệch xác suất thực nghiệm: 0.4788
(Xác suất thực nghiệm kỳ vọng nhỏ hơn hoặc bằng cận trên)

n = 100: Phương sai thực nghiệm = 0.002454
Bất đẳng thức Chebyshev (epsilon = 0.05):
P(|Y - p| >= 0.05) <= 0.9600
Độ lệch xác suất thực nghiệm: 0.3151
(Xác suất thực nghiệm kỳ vọng nhỏ hơn hoặc bằng cận trên)

n = 500: Phương sai thực nghiệm = 0.000488
Bất đẳng thức Chebyshev (epsilon = 0.05):
P(|Y - p| >= 0.05) <= 0.1920
Độ lệch xác suất thực nghiệm: 0.0236
(Xác suất thực nghiệm kỳ vọng nhỏ hơn hoặc bằng cận trên)

n = 1000: Phương sai thực nghiệm = 0.000244
Bất đẳng thức Chebyshev (epsilon = 0.05):
P(|Y - p| >= 0.05) <= 0.0960
Độ lệch xác suất thực nghiệm: 0.0017
(Xác suất thực nghiệm kỳ vọng nhỏ hơn hoặc bằng cận trên)

n = 5000: Phương sai thực nghiệm = 0.000048
Bất đẳng thức Chebyshev (epsilon = 0.05):
P(|Y - p| >= 0.05) <= 0.0192
```

```
Độ lệch xác suất thực nghiệm: 0.0000
(Xác suất thực nghiệm kỳ vọng nhỏ hơn hoặc bằng cận trên)
----------------------------------------
Trực quan hoá phân phối tỷ lệ mẫu cho trường hợp cỡ mẫu lớn nhất:
```



Bài tập 4. (phần 2.6.8) Assume that we draw m samples x_i from a probability distribution with zero mean and unit variance. Compute the averages $z_m \stackrel{\text{def}}{=} m^{-1} \sum_{i=1}^{m} x_i$. Can we apply Chebyshev's inequality for every z_m independently? Why not?

Tóm tắt:

- Cho m mẫu $x_1, x_2, \ldots, x_m$ được lấy từ một phân phối xác suất có:
  - Kỳ vọng $E[x_i] = 0$ (trung bình bằng 0).
  - Phương sai $\mathrm{Var}(x_i) = 1$ (phương sai đơn vị).

– Trung bình mẫu:

$$z_m = \frac{1}{m}\sum_{i=1}^{m} x_i$$

- Question: Có thể áp dụng bất đẳng thức Chebyshev cho từng z_m một cách độc lập không? Tại sao?

Giải bài tập 4:

1. Ta cần tính kỳ vọng và phương sai của z_m.

Vì các x_i có kỳ vọng $E[x_i] = 0$ và phương sai $\text{Var}(x_i) = 1$, và các mẫu là độc lập, ta có:

- Kỳ vọng:

Chúng ta áp dụng tính chất tuyến tính của kỳ vọng, đó là:

$$\mathbb{E}[aX + bY] = a\mathbb{E}[X] + b\mathbb{E}[Y]$$

Cho nên:

$$\mathbb{E}[z_m] = \mathbb{E}\left[\sum_{i=1}^{m} x_i\right] = \sum_{i=1}^{m} \mathbb{E}[x_i] = \frac{1}{m}\cdot(0 + 0 + \cdots + 0) = 0$$

- Phương sai:

Phương sai tổng của hai biến ngẫu nhiên:

$$\text{Var}(X + Y) = \text{Var}(X) + \text{Var}(Y) + 2\cdot\text{Cov}(X, Y)$$

Nếu X và Y là độc lập, thì $\text{Cov}(X, Y) = 0$, và công thức rút gọn thành:

$$\text{Var}(X + Y) = \text{Var}(X) + \text{Var}(Y)$$

Áp dụng:

$$\text{Nếu } Z = aX, \text{ thì } \text{Var}(Z) = a^2\cdot\text{Var}(X)$$

$$\text{Var}(z_m) = \text{Var}\left(\frac{1}{m}\sum_{i=1}^{m} x_i\right) = \left(\frac{1}{m}\right)^2\cdot\text{Var}\left(\sum_{i=1}^{m} x_i\right) = \frac{1}{m^2}\cdot m = \frac{1}{m}$$

2. Có thể áp dụng bất đẳng thức Chebyshev cho từng z_m không?

Có vì:

Với mỗi giá trị cụ thể của z_m, ta có thể áp dụng công thức Chebyshev:

$$\mathbb{P}(|z_m| \geq \varepsilon) \leq \frac{1}{m\varepsilon^2}$$

Nếu $m = 100, \varepsilon = 0.1,$ thì: $\mathbb{P}(|z_{100}| \geq 0.1) \leq \frac{1}{100 \cdot 0.01} = 1$

Nếu $m = 1000, \varepsilon = 0.1,$ thì: $\mathbb{P}(|z_{1000}| \geq 0.1) \leq \frac{1}{1000 \cdot 0.01} = 0.1$

```python
# Mã nguồn code minh hoạ
import numpy as np

def simulate_chebyshev(m_values, epsilon, num_trials=10000):
    results = {}
    for m in m_values:
        # Sinh num_trials mẫu trung bình từ m biến ngẫu nhiên phân phối chuẩn (mean=0, var=1)
        samples = np.random.normal(loc=0, scale=1, size=(num_trials, m))
        z_m = samples.mean(axis=1)

        # Xác suất thực nghiệm: tỷ lệ |z_m| >= epsilon
        empirical_prob = np.mean(np.abs(z_m) >= epsilon)

        # Giới hạn từ bất đẳng thức Chebyshev
        chebyshev_bound = 1 / (m * epsilon**2)

        results[m] = {
            "Empirical Probability": empirical_prob,
            "Chebyshev Bound": chebyshev_bound
        }

    return results

# Thử nghiệm với các giá trị m khác nhau
m_values = [1, 5, 10, 50, 100, 500, 50000]
epsilon = 0.5
```

```

results = simulate_chebyshev(m_values, epsilon)

for m, res in results.items():
    print(f"m = {m}")
    print(f"  Xác suất thực nghiệm   = {res['Empirical Probability']:.4f}")
    print(f"  Giới hạn Chebyshev     = {res['Chebyshev Bound']:.4f}")
    print()

Output:
m = 1
  Xác suất thực nghiệm   = 0.6094
  Giới hạn Chebyshev     = 4.0000

m = 5
  Xác suất thực nghiệm   = 0.2670
  Giới hạn Chebyshev     = 0.8000

m = 10
  Xác suất thực nghiệm   = 0.1189
  Giới hạn Chebyshev     = 0.4000

m = 50
  Xác suất thực nghiệm   = 0.0003
  Giới hạn Chebyshev     = 0.0800

m = 100
  Xác suất thực nghiệm   = 0.0000
  Giới hạn Chebyshev     = 0.0400

m = 500
  Xác suất thực nghiệm   = 0.0000
  Giới hạn Chebyshev     = 0.0080
```

Bài tập 5. (phần 2.6.8) Given two events with probability $P(\mathcal{A})$ and $P(\mathcal{A})$, compute upper and lower bounds on $P(\mathcal{A} \cup \mathcal{B})$ and $P(\mathcal{A} \cap \mathcal{B})$. Hint: graph the situation using a Venn diagram.

Tóm tắt:

- Cho hai biến cố A và B với xác suất $P(A)$ và $P(B)$. Hãy tìm cận trên và cận dưới cho:
  - $P(A \cup B)$
  - $P(A \cap B)$.

Giải bài tập 5:

1. Giải thích bằng sơ đồ Venn

- Vẽ hai hình tròn giao nhau, một đại diện cho A, một cho B. Diện tích mỗi hình tròn tương ứng với xác suất của biến cố đó.

- Phần giao nhau thể hiện $P(A \cap B)$

- Toàn bộ phần nằm trong cả hai hình tròn (không tính phần chồng 2 lần) thể hiện $P(A \cup B)$.

2. Chúng ta sẽ tìm cận trên và cận dưới của $P(A \cup B)$

Ta có công thức tổng quát:

$$P(A \cup B) = P(A) + P(B) - P(A \cap B)$$

- Cận dưới:

Để $P(A \cup B)$ nhỏ nhất, thì $P(A \cap B)$ phải lớn nhất có thể, tức là $P(A \cap B) = min(P(A), P(B))$

Khi đó:

$$P(A \cup B)_{\min} = P(A) + P(B) - min(P(A), P(B)) = \max(P(A), P(B))$$

- Cận trên:

Để $P(A \cup B)$ lớn nhất, thì $P(A \cap B)$ phải nhỏ nhất có thể, tức là $P(A \cap B) = 0$ (hai biến cố rời nhau)

Khi đó:

$$P(A \cup B)_{\max} = P(A) + P(B)$$

Mà tổng này không được vượt quá 1. Vậy:

$$P(A \cup B)_{\max} = min(1, P(A) + P(B))$$

3. Chúng ta sẽ tìm cận trên và cận dưới của $P(A \cap B)$

- Cận trên:

Giao của A và B không thể lớn hơn biến cố nhỏ hơn, nên:

$$P(A \cap B)_{\max} = min(P(A), P(B))$$

- Cận dưới:

Từ công thức ở trên, để $P(A \cap B)$ nhỏ nhất, thì $P(A \cup B)$ phải lớn nhất, tức là:

$$P(A \cap B)_{\min} = P(A) + P(B) - min(1, P(A) + P(B)) = max(0, P(A) + P(B) - 1)$$

```python
# mã nguồn code minh hoạ
def bounds_of_union_and_intersection(p_a, p_b):
    # Kiểm tra đầu vào hợp lệ
    if not (0 <= p_a <= 1 and 0 <= p_b <= 1):
        raise ValueError("Xác suất phải nằm trong khoảng từ 0 đến 1")

    # P(A ∪ B)
    union_lower = max(p_a, p_b)
    union_upper = min(1, p_a + p_b)

    # P(A ∪ B)
    intersection_lower = max(0, p_a + p_b - 1)
    intersection_upper = min(p_a, p_b)

    return {
        "P(A ∪ B)": {
            "lower_bound": union_lower,
```

```
            "upper_bound": union_upper
        },
        "P(A ∪ B)": {
            "lower_bound": intersection_lower,
            "upper_bound": intersection_upper
        }
    }

# Ví dụ
p_a = 0.4
p_b = 0.7
result = bounds_of_union_and_intersection(p_a, p_b)

for event, bounds in result.items():
    print(f"{event}:")
    print(f"  Cận dưới: {bounds['lower_bound']}")
    print(f"  Cận trên: {bounds['upper_bound']}")

Output:
P(A ∪ B):
  Cận dưới: 0.7
  Cận trên: 1
P(A ∪ B):
  Cận dưới: 0.10000000000000009
  Cận trên: 0.4
```

Bài tập 6. (phần 2.6.8) Assume that we have a sequence of random variables, say A, B, and C, where B only depends on A, and C only depends on B, can you simplify the joint $P(A, B, C)$ probability? Hint: this is a Markov chain.

Tạm dịch: Cho A, B, C là các biến ngẫu nhiên. B chỉ phụ thuộc vào A, C chỉ phụ thuộc vào B. Đơn giản hóa $P(A, B, C)$ thế nào?

Giải bài tập 6:

C chỉ phụ thuộc vào B ⇒ C độc lập có điều kiện với A khi đã biết B, ký hiệu là:

$$P(C|A,B) = P = (C|B)$$

Theo quy tắc chuỗi (chain rule) trong xác suất:

$$P(A,B,C) = P(A).P(B|A).P(C|A,B)$$

Ta được công thức rút gọn:

$$P(A,B,C) = P(A).P(B|A).P(C|B)$$

```python
#### Mã nguồn code minh hoạ
# Define the probabilities
P_A = {
    'a1': 0.5,
    'a2': 0.5,
}

P_B_given_A = {
    'a1': {'b1': 0.4, 'b2': 0.6},
    'a2': {'b1': 0.7, 'b2': 0.3},
}

P_C_given_B = {
    'b1': {'c1': 0.8, 'c2': 0.2},
    'b2': {'c1': 0.1, 'c2': 0.9},
}

# Calculate joint probability for a specific path: A='a1', B='b1', C='c1'
a = 'a1'
b = 'b1'
```

```
c = 'c1'

joint_prob = P_A[a] * P_B_given_A[a][b] * P_C_given_B[b][c]

print(f"P(A={a}, B={b}, C={c}) = {joint_prob}")
# P(A=a1, B=b1, C=c1) = 0.16000000000000003
```

Bài tập 7. (phần 2.6.8) In Section 2.6.5, assume that the outcomes of the two tests are not independent. In particular assume that either test on its own has a false positive rate of 10% and a false negative rate of 1%. That is, assume that $P(D = 1 \mid H = 0) = 0.1$ and that $P(D = 0 \mid H = 1) = 0.01$. Moreover, assume that for $H = 1$ (infected) the test outcomes are conditionally independent, i.e., that $P(D_1, D_2 \mid H = 1) = P(D_1 \mid H = 1)P(D_2 \mid H = 1)$ but that for healthy patients the outcomes are coupled via $P(D_1 = D_2 = 1 \mid H = 0) = 0.02$.

1. Work out the joint probability table for D_1 and D_2, given $H = 0$ based on the information you have so far.
2. Derive the probability that the patient is diseased ($H = 1$) after one test returns positive. You can assume the same baseline probability $P(H = 1) = 0.0015$ as before.
3. Derive the probability that the patient is diseased ($H = 1$) after both tests return positive.

Tóm tắt bài toán
Biến cố:

- $H = 1$: Bệnh nhân mắc bệnh (nhiễm bệnh).
- $H = 0$: Bệnh nhân khỏe mạnh.
- D_1, D_2: Kết quả của hai xét nghiệm ($D_i = 1$ nếu dương tính, $D_i = 0$ nếu âm tính).

Giả định:

- Xác suất:
  - Xác suất tiên nghiệm: $P(H = 1) = 0.0015$
    $\Rightarrow P(H = 1) = 0.9985$.

- Xét nghiệm dương tính giả (False Positive): $P(D_i = 1 \mid H = 0) = 0.1$.
  $\Rightarrow$ Xét nghiệm âm tính thật (True Positive): $P(D_i = 0 \mid H = 0) = 0.9$.
- Xét nghiệm âm tính giả (False Negative): $P(D_i = 0 \mid H = 1) = 0.01$.
  $\Rightarrow$ Xét nghiệm dương tính thật (True positive): $P(D_i = 1 \mid H = 1) = 0.99$.

• Tính độc lập có điều kiện:
  - Nếu $H = 1$: D_1 và D_2 độc lập, tức:

$$P(D_1, D_2 \mid H = 1) = P(D_1 \mid H = 1)P(D_2 \mid H = 1).$$

  - Nếu $H = 0$: D_1 và D_2 không độc lập, với:

$$P(D_1 = D_2 = 1 \mid H = 0) = 0.02.$$

Câu hỏi:

1. Tính bảng xác suất chung cho D_1 và D_2 khi đã biết $H = 0$ dựa trên các thông tin trên.
2. Suy ra xác suất bệnh nhân thực sự bị bệnh ($H = 1$) sau khi một xét nghiệm cho kết quả dương tính. (Giả sử xác suất ban đầu $P(H = 1) = 0.0015$ như trước.)
3. Suy ra xác suất bệnh nhân bị bệnh ($H = 1$) sau khi cả hai xét nghiệm đều cho kết quả dương tính.

Giải bài tập 7:

7.1. Bảng xác suất đồng thời cho $P(D_1, D_2 \mid H = 0)$.

Ta có: Với $H = 0$, hai xét nghiệm không độc lập nhưng có thể mô hình hóa thông qua xác suất đồng thời. Ta cần điền các giá trị còn lại của bảng:

$D_1 \backslash D_2$	$D_2 = 0$	$D_2 = 1$	Tổng
$D_1 = 0$	a	b	0.9
$D_1 = 1$	c	0.02	0.1
Tổng	0.9	0.1	1

Bảng 3: Bảng xác suất đồng thời

Tính toán:

- $c = P(D_1 = 1 \mid H = 0) - P(D_1 = D_2 = 1 \mid H = 0) = 0.1 - 0.02 = 0.08.$
- $b = P(D_2 = 1 \mid H = 0) - P(D_1 = D_2 = 1 \mid H = 0) = 0.1 - 0.02 = 0.08.$
- $a = 0.9 - b = 0.9 - 0.08 = 0.82.$

Kết quả:

$D_1 \backslash D_2$	$D_2 = 0$	$D_2 = 1$	Tổng
$D_1 = 0$	0.82	0.08	0.9
$D_1 = 1$	0.08	0.02	0.1
Tổng	0.9	0.1	1

Bảng 4: Bảng xác suất đồng thời

7.2. Xác suất bệnh sau một xét nghiệm dương tính ($P(H = 1 \mid D_1 = 1)$) Công thức Bayes:

$$P(H = 1 \mid D_1 = 1) = \frac{P(D_1 = 1 \mid H = 1)P(H = 1)}{P(D_1 = 1)}$$

Tính các thành phần:

- $P(D_1 = 1 \mid H = 1) = 1 - P(D_1 = 0 \mid H = 1) = 1 - 0.01 = 0.99.$
- $P(H = 1) = 0.0015$ (đề bài cho) $\Rightarrow P(H = 0) = 0.9985$.
- $P(D_1 = 1) = P(D_1 = 1 \mid H = 1)P(H = 1) + P(D_1 = 1 \mid H = 0)P(H = 0)$
- $P(D_1 = 1) = 0.99 \times 0.0015 + 0.1 \times 0.9985 \approx 0.101485.$

Kết quả:

$$P(H = 1 \mid D_1 = 1) = \frac{0.99 \times 0.0015}{0.101485} \approx 0.0146 \quad (1.46\%).$$

7.3. Xác suất bệnh sau hai xét nghiệm dương tính ($P(H = 1 \mid D_1 = D_2 = 1)$)

Ta có công thức Bayes:

$$P(H = 1 \mid D_1 = D_2 = 1) = \frac{P(D_1 = D_2 = 1 \mid H = 1)P(H = 1)}{P(D_1 = D_2 = 1)}.$$

Do D_1, D_2 độc lập khi $H = 1$, ta có:

$$P(D_1 = D_2 = 1 \mid H = 1) = P(D_1 = 1 \mid H = 1)^2 = 0.99^2 = 0.9801.$$

- Tử số: $P(D\_1 = D\_2 = 1 \mid H = 1)P(H = 1) = 0.9801 \times 0.0015 \approx 0.00147$.
- Mẫu số:

$$P(D_1 = D_2 = 1) = P(D_1 = D_2 = 1 \mid H = 1)P(H = 1) + P(D_1 = D_2 = 1 \mid H = 0)P(H = 0)$$

$$P(D_1 = D_2 = 1) = 0.9801 \times 0.0015 + 0.02 \times (1 - 0.0015) \approx 0.02147.$$

- Kết quả:

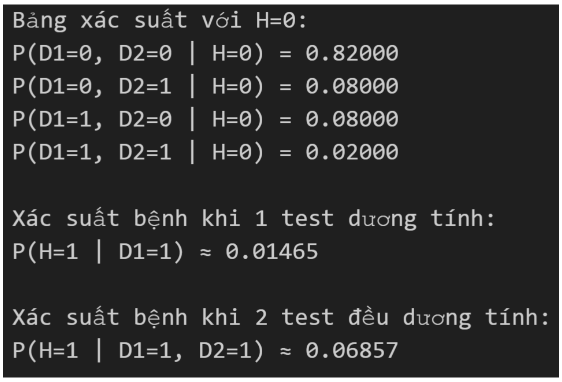
$$P(H = 1 \mid D_1 = D_2 = 1) = \frac{0.9801 \times 0.0015}{0.9801 \times 0.0015 + 0.02 \times (1 - 0.0015)} \approx 0.0685 \quad (6.85\%).$$

Code Python cho bài tập 7

```python
def bayes_single_test(P_H1, false_pos, false_neg):
    Tính xác suất P(H=1 | D=1) sau một test dương tính.
    P_H0 = 1 - P_H1
    P_D_pos_given_H1 = 1 - false_neg
    P_D_pos_given_H0 = false_pos
    numerator = P_D_pos_given_H1 * P_H1
    denominator = numerator + P_D_pos_given_H0 * P_H0
    posterior = numerator / denominator
    return posterior
def bayes_double_test(P_H1, false_pos, false_neg, P_D1_D2_1_H0):
    # Tính xác suất P(H=1 | D1=1, D2=1) sau hai test dương tính.
    P_H0 = 1 - P_H1
    P_D1_1_given_H1 = 1 - false_neg
    P_D2_1_given_H1 = 1 - false_neg
    # Vì D1 và D2 độc lập có điều kiện H=1
    P_D1_D2_1_H1 = P_D1_1_given_H1 * P_D2_1_given_H1
    numerator = P_D1_D2_1_H1 * P_H1
    denominator = numerator + P_D1_D2_1_H0 * P_H0
    posterior = numerator / denominator
    return posterior
```

```python
def compute_joint_prob_H0(false_pos, P_D1_eq_D2_1_given_H0):
    #Tự động suy ra bảng xác suất joint P(D1, D2 | H=0) từ false positive rate
    #và xác suất đồng thời D1=D2=1
    p11 = P_D1_eq_D2_1_given_H0
    p10 = false_pos - p11
    p01 = false_pos - p11
    p00 = 1 - (p11 + p10 + p01)
    return {
        (0, 0): p00,
        (0, 1): p01,
        (1, 0): p10,
        (1, 1): p11
    }
# ======= INPUT TỪ ĐỀ BÀI BÀI TOÁN 7 =======
P_H1 = 0.0015  # Xác suất bệnh
false_positive = 0.1
false_negative = 0.01
P_D1_eq_D2_eq_1_given_H0 = 0.02
# ======= LỜI GIẢI CHO BÀI TOÁN 7  =======
# Câu 1: joint probability table
joint_table = compute_joint_prob_H0(false_positive, P_D1_eq_D2_eq_1_given_H0)
print("Joint probability table for D1, D2 given H=0:")
for (d1, d2), prob in joint_table.items():
    print(f"P(D1={d1}, D2={d2} | H=0) = {prob:.4f}")
# Câu 2: P(H=1 | D=1)
posterior_1 = bayes_single_test(P_H1, false_positive, false_negative)
print(f"\nP(H=1 | D=1) = {posterior_1:.4f} (~{posterior_1 * 100:.2f}%)")
# Câu 3: P(H=1 | D1=1, D2=1)
posterior_2 = bayes_double_test(P_H1, false_positive, false_negative,
P_D1_eq_D2_eq_1_given_H0)
print(f"P(H=1 | D1=1, D2=1) = {posterior_2:.4f} (~{posterior_2 * 100:.2f}%)")
```

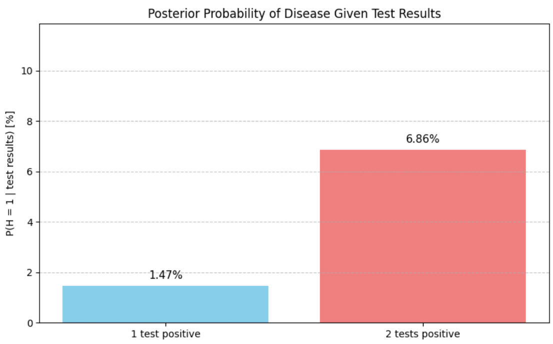
Kết quả của chạy Code Python cho bài tập 7



```
Bảng xác suất với H=0:
P(D1=0, D2=0 | H=0) = 0.82000
P(D1=0, D2=1 | H=0) = 0.08000
P(D1=1, D2=0 | H=0) = 0.08000
P(D1=1, D2=1 | H=0) = 0.02000

Xác suất bệnh khi 1 test dương tính:
P(H=1 | D1=1) ≈ 0.01465

Xác suất bệnh khi 2 test đều dương tính:
P(H=1 | D1=1, D2=1) ≈ 0.06857
```

Hình 1: Kết quả bài tập 7(Python)

Trực quan hóa hai xác suất hậu nghiệm $P(H = 1 \mid D = 1)$ và $P(H = 1 \mid D_1 = 1, D_2 = 1)$ bằng biểu đồ cột



Hình 2: Biểu đồ trực quan hóa hai xác suất hậu nghiệm

Mô phỏng bài toán bằng thử nghiệm ngẫu nhiên (Monte Carlo)
Cho kết quả tương đồng với lời giải ở trên

$$\text{Monte Carlo estimate of } P(H = 1 | D1 = 1, D2 = 1) : 0.0681 (\approx 6.81\%)$$

Python code mô phỏng bài toán bằng thử nghiệm ngẫu nhiên (Monte Carlo)

```python
import numpy as np
def monte_carlo_simulation(n_trials, P_H1, false_pos, false_neg, P_D1_D2_1_H0):
    # Mô phỏng Monte Carlo để ước lượng P(H=1 | D1=1, D2=1)
    # Tạo mảng ngẫu nhiên xác định người bệnh (H=1) hay khỏe (H=0)
```

```python
    H = np.random.rand(n_trials) < P_H1
    # Khởi tạo mảng lưu kết quả test D1 và D2
    D1 = np.zeros(n_trials, dtype=int)
    D2 = np.zeros(n_trials, dtype=int)
    # Với H = 1 (bệnh): D1, D2 độc lập, xác suất đúng dương = 1 - false_negative
    idx_H1 = np.where(H)[0]
    D1[idx_H1] = np.random.rand(len(idx_H1)) < (1 - false_neg)
    D2[idx_H1] = np.random.rand(len(idx_H1)) < (1 - false_neg)
    # Với H = 0 (khỏe): tạo D1, D2 có phụ thuộc để giữ đúng P(D1=1, D2=1 | H=0) = 0.02
    idx_H0 = np.where(~H)[0]
    for i in idx_H0:
        r = np.random.rand()
        if r < P_D1_D2_1_H0:
            D1[i] = 1
            D2[i] = 1
        elif r < P_D1_D2_1_H0 + (false_pos - P_D1_D2_1_H0):  # D1=1, D2=0
            D1[i] = 1
            D2[i] = 0
        elif r < P_D1_D2_1_H0 + 2 * (false_pos - P_D1_D2_1_H0):  # D1=0, D2=1
            D1[i] = 0
            D2[i] = 1
        else:
            D1[i] = 0
            D2[i] = 0
    # Đếm số trường hợp D1=1 và D2=1
    both_positive = (D1 == 1) & (D2 == 1)
    count_both_positive = np.sum(both_positive)
    count_H1_given_both_positive = np.sum(H[both_positive])
    # Tính xác suất hậu nghiệm
    posterior_estimate = count_H1_given_both_positive / count_both_positive
    return posterior_estimate
# ====== THÔNG SỐ ĐẦU VÀO ======
```

```
P_H1 = 0.0015
false_pos = 0.1
false_neg = 0.01
P_D1_D2_1_H0 = 0.02
n_simulations = 10**7  # 10 triệu thử nghiệm
# ====== CHẠY MÔ PHỎNG VÀ XUẤT KẾT QUẢ ======
posterior_mc = monte_carlo_simulation(n_simulations, P_H1, false_pos, false_neg,
P_D1_D2_1_H0)
print(f"\nMonte Carlo estimate of P(H=1 | D1=1, D2=1): {posterior_mc:.4f}
(~{posterior_mc * 100:.2f}%)")
```

Bài tập 8. (phần 2.6.8) Assume that you are an asset manager for an investment bank and you have a choice of stocks s_i to invest in. Your portfolio needs to add up to 1 with weights α_i for each stock. The stocks have an average return $\boldsymbol{\mu} = E_{\mathbf{s}\sim P}[\mathbf{s}]$ and covariance $\boldsymbol{\Sigma} = \mathrm{Cov}_{\mathbf{s}\sim P}[\mathbf{s}]$.

1. Compute the expected return for a given portfolio $\boldsymbol{\alpha}$.
2. If you wanted to maximize the return of the portfolio, how should you choose your investment?
3. Compute the variance of the portfolio.
4. Formulate an optimization problem of maximizing the return while keeping the variance constrained to an upper bound. This is the Nobel-Prize winning Markovitz portfolio (Mangram, 2013). To solve it you will need a quadratic programming solver, something way beyond the scope of this book.

Giải bài tập 8:

Ta có:

- Danh mục đầu tư: Gồm các cổ phiếu s_i với trọng số α_i (thỏa mãn $\sum \alpha_i = 1$).
- Lợi nhuận trung bình của cổ phiếu: $\mu = E_{s\sim P}[s]$.
- Ma trận hiệp phương sai: $\Sigma = \mathrm{Cov}_{s\sim P}[s]$.

Tạm dịch: Giả sử bạn là một quản lý tài sản tại một ngân hàng đầu tư và bạn có nhiều lựa chọn cổ phiếu s_i để đầu tư. Danh mục đầu tư của bạn cần có tổng trọng số bằng 1, với trọng số α_i cho mỗi cổ phiếu. Các cổ phiếu có mức lợi nhuận trung bình

$$\mu = E_{s\sim P}[s]$$

và hiệp phương sai

$$\Sigma = \mathrm{Cov}_{s \sim P}[s].$$

1. Tính toán lợi nhuận kỳ vọng cho một danh mục đầu tư **s** đã cho.
2. Nếu muốn tối đa hóa lợi nhuận của danh mục đầu tư, nên phân bổ đầu tư như thế nào?
3. Tính toán phương sai của danh mục đầu tư.
4. Xây dựng bài toán tối ưu để tối đa hóa lợi nhuận trong khi giữ phương sai ở một ngưỡng nhất định. Đây chính là danh mục đầu tư Markowitz đã đoạt giải Nobel (Mangram, 2013). Để giải bài toán này, sẽ cần sử dụng một trình giải tối ưu bậc hai (quadratic programming solver), một công cụ vượt xa phạm vi của cuốn sách này.

Ta có:
- Danh mục đầu tư $\mathbf{s} = \{s_1, s_2, ..., s_n\}$: Gồm các cổ phiếu s_i với trọng số tương ứng α_i, $\alpha = \{\alpha_1, \alpha_2, ..., \alpha_n\}$, với ràng buộc:

$$\sum_{i=1}^{n} \alpha_i = 1$$

- Lợi nhuận trung bình của cổ phiếu: $\mu = E_{s \sim P}[\mathbf{s}]$.
- Ma trân hiệp phương sai: $\Sigma = \mathrm{Cov}_{s \sim P}[\mathbf{s}]$, với vector ngẫu nhiên $\mathbf{s} = [s_1, s_2, \ldots, s_n]^T$, ma trận Σ có dạng:

$$\Sigma = \begin{bmatrix} \mathrm{Var}(s_1) & \mathrm{Cov}(s_1, s_2) & \ldots & \mathrm{Cov}(s_1, s_n) \\ \mathrm{Cov}(s_2, s_1) & \mathrm{Var}(s_2) & \ldots & \mathrm{Cov}(s_2, s_n) \\ \vdots & \vdots & \ddots & \vdots \\ \mathrm{Cov}(s_n, s_1) & \mathrm{Cov}(s_n, s_2) & \ldots & \mathrm{Var}(s_n) \end{bmatrix}$$

8.1. Tính lợi nhuận kỳ vọng của danh mục:
- Lợi nhuận kỳ vọng của danh mục là trung bình trọng số của lợi nhuận các cổ phiếu: - Mỗi cổ phiếu s_i đóng góp tỷ suất sinh lợi trung bình là μ_i, đầu tư trọng số α_i vào nó, nên lợi nhuận kỳ vọng của danh mục là

$$E[\mathbf{s}] = \sum_{i=1}^{n} \alpha_i \mu_i = \alpha^\top \mu$$

- Ví dụ: Nếu danh mục gồm 2 cổ phiếu với $\alpha = \{0.6, 0.4\}$ và $\mu = \{0.1, 0.05\}$, thì:

$$E[\alpha] = 0.6 \times 0.1 + 0.4 \times 0.05 = 0.08 \quad (8\%).$$

8.2. Tối đa hóa lợi nhuận:

- Để tối đa hóa lợi nhuận, Đầu tư toàn bộ (100%) vào cổ phiếu có lợi nhuận kỳ vọng μ_i cao nhất:

$$\alpha_j = \begin{cases} 1 & \text{nếu } j = \arg\max_i \mu_i, \\ 0 & i \neq j \end{cases}$$

8.3. Tính phương sai của danh mục đầu tư: Phương sai danh mục phản ánh độ biến động:

$$\mathrm{Var}(\alpha^T \mathbf{s}) = \alpha^T \Sigma \alpha = \sum_{i,j} \alpha_i \alpha_j \Sigma_{i,j}.$$

Ví dụ: Nếu $\Sigma = \begin{bmatrix} 0.04 & 0.01 \\ 0.01 & 0.02 \end{bmatrix}$ và $\alpha = [0.6, 0.4]^T$:

$$\mathrm{Var}(\alpha^T \mathbf{s}) = 0.6^2 \times 0.04 + 2 \times 0.6 \times 0.4 \times 0.01 + 0.4^2 \times 0.02 = 0.0208.$$

8.4. Bài toán tối ưu danh mục Markovitz:

Tối đa hóa lợi nhuận với ràng buộc rủi ro tối đa $\sigma^2_{\max}$:

$$\begin{aligned} \max_{\alpha} \quad & \alpha^T \mu \\ \text{subject to} \quad & \alpha^T \Sigma \alpha \leq \sigma^2_{\max}, \\ & \sum \alpha_i = 1, \\ & \alpha_i \geq 0 \quad (\text{nếu không cho phép bán khống}). \end{aligned}$$

Để mô phỏng cho bài tập 8: Thực hiện lấy dữ liệu từ nguồn thị trường chính khoán Việt Nam (import vnstock as vn). Đặc tả dữ liệu:

- Chạy thực nghiệm trên 5 mã chứng khoán: VCB, FPT, MWG, VNM, HPG.
- Thời điểm lấy dữ liệu từ ngày 01/01/2023 đến ngày 01/01/2024.
- Điểm dữ liệu: là giá trị trung bình giao dịch trong 1 ngày của tất cả các phiên khớp lệnh trong ngày đó. Có 250 ngày giao dịch $\Rightarrow$ có 250 điểm dữ liệu.

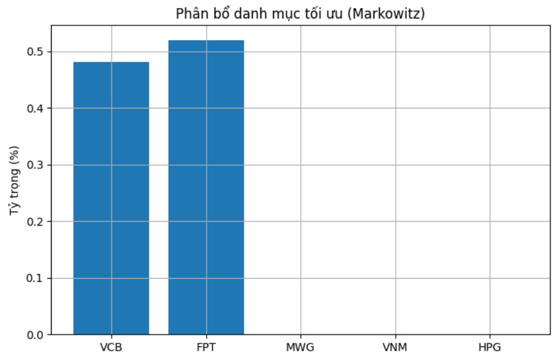
Kết quả chạy thực nghiệm cho bài tập 8:

1. (8.1) Lợi nhuận kỳ vọng hàng ngày và hàng năm của danh mục

- Lợi suất kỳ vọng hàng ngày của danh mục: 0.1898%
- Lợi suất kỳ vọng hàng năm của danh mục: 47.8326%

2. (8.2) Tối đa hóa lợi nhuận danh mục đầu tư
   - Tối đa hóa lợi nhuận danh mục:
     - VCB: 0.00%
     - FPT: 100.00%
     - MWG: 0.00%
     - VNM: 0.00%
     - HPG: 0.00%
   - Lợi suất kỳ vọng danh mục tối đa lợi nhuận: 0.19%
   - Rủi ro tương ứng (độ lệch chuẩn): 1.96%
3. (8.3) Tính toán phương sai của danh mục đầu tư
   - Phương sai danh mục Markowitz: 0.000210
4. (8.4) Xây dựng bài toán tối ưu để tối đa hóa lợi nhuận trong khi giữ phương sai ở một ngưỡng nhất định (Tối đa hóa lợi nhuận với ràng buộc phương sai).
   - Danh mục tối ưu với ràng buộc phương sai:
     - VCB: 0.00%
     - FPT: 100.00%
     - MWG: 0.00%
     - VNM: 0.00%
     - HPG: 0.00%
   - Lợi suất kỳ vọng: 0.19%
   - Phương sai danh mục: 0.000383
   - Rủi ro tương ứng (độ lệch chuẩn): 1.96%

5. (8.5) Trực quan hóa danh mục đầu tư tối ưu bằng Bar Chart



Hình 3: Danh mục đầu tư tối ưu

Code Python cho bài tập 8

```
## Chọn 5 mã từ danh mục
stock = vn.Vnstock().stock(source='TCBS')
stock.listing.all_symbols()
import vnstock as vn
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import cvxpy as cp
# Đọc dữ liệu từ file CSV
df = pd.read_csv('vietnam_stock_prices_2023_2024_5ma.csv', index_col=0,
parse_dates=True)
# Tính lợi suất hàng ngày (daily returns)
returns = df.pct_change().dropna()
# Tính kỳ vọng lợi suất trung bình và ma trận hiệp phương sai
mu = returns.mean().values
Sigma = returns.cov().values
# Số lượng tài sản
n = len(mu)
# Khởi tạo biến trọng số danh mục
w = cp.Variable(n)
```

```
# Bài toán Markowitz: tối ưu hóa tỷ lệ Sharpe (không có rủi ro)
risk_aversion = 1  # hệ số điều chỉnh giữa lợi nhuận và rủi ro
# Hàm mục tiêu: cực tiểu hóa rủi ro - tối đa hóa lợi nhuận
objective = cp.Maximize(mu @ w - risk_aversion * cp.quad_form(w, Sigma))
# Ràng buộc: tổng trọng số = 1 và mỗi trọng số không âm
constraints = [cp.sum(w) == 1, w >= 0]
# Giải bài toán tối ưu
prob = cp.Problem(objective, constraints)
prob.solve()
# Kết quả
optimal_weights = w.value
portfolio_return = mu @ optimal_weights
portfolio_risk = np.sqrt(optimal_weights.T @ Sigma @ optimal_weights)
# --- Tính lợi nhuận kỳ vọng của danh mục ---
# Lợi nhuận kỳ vọng hàng ngày đã có:
expected_return_daily = np.dot(mu, optimal_weights)
# Giả định có 252 ngày giao dịch/năm
expected_return_annual = expected_return_daily * 252
print(f"\nLợi suất kỳ vọng hàng ngày của danh mục: {expected_return_daily:.4%}")
print(f"Lợi suất kỳ vọng hàng năm của danh mục: {expected_return_annual:.2%}")
# --- Tối đa hóa lợi nhuận danh mục đầu tư ---
# Khởi tạo biến trọng số mới
w_max_return = cp.Variable(n)
# Hàm mục tiêu: tối đa hóa lợi nhuận kỳ vọng
objective_max_return = cp.Maximize(mu @ w_max_return)
# Ràng buộc: tổng trọng số = 1, không bán khống
constraints_max_return = [cp.sum(w_max_return) == 1, w_max_return >= 0]
# Giải bài toán
prob_max_return = cp.Problem(objective_max_return, constraints_max_return)
prob_max_return.solve()
# Kết quả
weights_max_return = w_max_return.value
```

```
portfolio_return_max = mu @ weights_max_return
portfolio_risk_max = np.sqrt(weights_max_return.T @ Sigma @ weights_max_return)
print("\n Tối đa hóa lợi nhuận danh mục:")
for ticker, weight in zip(df.columns, weights_max_return):
    print(f"{ticker}: {weight:.2%}")
print(f"\n Lợi suất kỳ vọng danh mục tối đa lợi nhuận: {portfolio_return_max:.2%}")
print(f" Rủi ro tương ứng (độ lệch chuẩn): {portfolio_risk_max:.2%}")
# Tính phương sai danh mục Markowitz
portfolio_variance = optimal_weights.T @ Sigma @ optimal_weights
print(f"\nPhương sai danh mục Markowitz: {portfolio_variance:.6f}")
# Cài đặt ngưỡng phương sai cho phép: 0.0004 tương ứng độ lệch chuẩn ≈ 2%
target_variance = 0.0004  # Bạn có thể thay đổi giá trị này
# Biến trọng số
w_risk_constrained = cp.Variable(n)
# Hàm mục tiêu: tối đa hóa lợi nhuận kỳ vọng
objective_risk_constrained = cp.Maximize(mu @ w_risk_constrained)
# Ràng buộc:
constraints_risk_constrained = [
    cp.sum(w_risk_constrained) == 1,           # Tổng trọng số bằng 1
    w_risk_constrained >= 0,                   # Không bán khống
    cp.quad_form(w_risk_constrained, Sigma) <= target_variance
    # Phương sai không vượt ngưỡng
]
# Giải bài toán
prob_risk_constrained = cp.Problem(objective_risk_constrained,
constraints_risk_constrained) prob_risk_constrained.solve()
# Kết quả
weights_risk_constrained = w_risk_constrained.value
portfolio_return_risk_constrained = mu @ weights_risk_constrained
portfolio_variance_risk_constrained = weights_risk_constrained.T @ Sigma @
weights_risk_constrained
print("\nDanh mục tối ưu với ràng buộc phương sai:")
```

```
for ticker, weight in zip(df.columns, weights_risk_constrained):
    print(f"{ticker}: {weight:.2%}")
print(f"\nLợi suất kỳ vọng: {portfolio_return_risk_constrained:.2%}")
print(f"Phương sai danh mục: {portfolio_variance_risk_constrained:.6f}")
print(f"Độ lệch chuẩn: {np.sqrt(portfolio_variance_risk_constrained):.2%}")
# Vẽ biểu đồ phân bổ danh mục
plt.figure(figsize=(8, 5))
plt.bar(df.columns, optimal_weights)
plt.title("Phân bổ danh mục tối ưu (Markowitz)")
plt.ylabel("Tỷ trọng (%)")
plt.grid(True)
plt.show()
```

4 Bài toán nâng cao

Bài tập nâng cao 1. Bài toán nâng cao dựa trên bài tập 5 (phần 2.6.8). Bài toán mô tả như sau:
Cho ba biến cố A, B, C với các xác suất đã biết:

- $P(A) = a$
- $P(B) = b$
- $P(C) = c$

Yêu cầu bài toán: Hãy tìm giới hạn trên và giới hạn dưới có thể có của $P(A \cup B \cup C)$ và $P(A \cap B \cap C)$.

Lời giải bài toán nâng cao 1

1. Giới hạn trên và dưới của $P(A \cup B \cup C)$
   Ta có công thức xác suất:

$$P(A \cup B \cup C) = P(A) + P(B) + P(C) - P(A \cap B) - P(A \cap C) - P(B \cap C) + P(A \cap B \cap C)$$

   Hợp ba biến cố xảy ra khi có khả năng ít nhất một trong ba biến cố xảy ra.

   (a) Giới hạn trên của $P(A \cup B \cup C)$

   - Giá trị lớn nhất xảy ra khi các tập càng ít giao nhau càng tốt – tức là các phần tử trong A, B, C càng khác biệt.
   - Trường hợp cực đại:
     – Nếu $A \cap B = \emptyset$, $A \cap C = \emptyset$, $B \cap C = \emptyset$, tức ba biến cố rời nhau hoàn toàn, thì:
$$P(A \cup B \cup C) = P(A) + P(B) + P(C) = a + b + c$$
   - Tuy nhiên:
     – Tổng xác suất trong không gian mẫu không thể vượt quá 1.
     – Nếu $a + b + c > 1$ thì hợp không thể vượt quá 1.
   - Do đó:
$$P(A \cup B \cup C) \leq \min(1, a + b + c)$$

(b) Giới hạn dưới của $P(A \cup B \cup C)$

- Giá trị nhỏ nhất sẽ xảy ra khi:
  - Các phần giao (chồng lặp) giữa các tập là **cực đại** (tức các tập trùng nhau nhiều nhất).
  - Khi đó, phần không bị trừ đi là lớn nhất $\implies$ tổng hợp nhỏ nhất.
- Trường hợp cực đoan:
  - Nếu $A \subseteq B \subseteq C$, thì mọi phần tử trong A cũng thuộc B và C.
  - Lúc này, $A \cup B \cup C$ gần bằng C, tức chỉ bằng xác suất lớn nhất trong ba tập.
  - Do đó, một giới hạn dưới hợp lý là:

$$P(A \cup B \cup C) \geq \max(a, b, c)$$

- Nhưng vẫn chưa chặt chẽ nhất. Ta xét điều kiện cần:
  - Xác suất không thể âm: $P(A \cup B \cup C) \geq 0$
  - Tổng xác suất của ba tập là $a + b + c$, nhưng do các tập chồng nhau, một phần của xác suất được tính nhiều lần và sẽ bị loại trừ.
  - Trong trường hợp tối đa trùng nhau, phần bị trừ đi nhiều nhất là 2.
  - Vì vậy, ta có giới hạn dưới chặt hơn:

$$P(A \cup B \cup C) \geq a + b + c - 2$$

  - Nhưng nếu $a + b + c - 2 < \max(a, b, c)$, thì giới hạn dưới hợp lý hơn là:

$$P(A \cup B \cup C) \geq \max(a + b + c - 2, \max(a, b, c))$$

2. Giới hạn trên và dưới của $P(A \cap B \cap C)$
   Giao ba biến cố là vùng mà cả A, B, C cùng xảy ra.

(a) Giới hạn trên của $P(A \cap B \cap C)$

- Giá trị lớn nhất xảy ra khi ba tập có phần trùng nhau càng nhiều càng tốt.

- Nếu $A \subseteq B \subseteq C$, thì toàn bộ xác suất của A sẽ nằm trong B và C. Khi đó:

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

- Tương tự, nếu $C \subseteq B \subseteq A$, thì:

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

- Vậy giá trị lớn nhất của giao là nhỏ nhất trong ba xác suất.
- Do đó:

$$P(A \cap B \cap C) \leq \min(a, b, c)$$

(b) Giới hạn dưới của $P(A \cap B \cap C)$

- Giá trị nhỏ nhất là khi giao ba tập gần như trống rỗng, tức ba tập càng rời nhau càng tốt.
- Trường hợp cực đoan:
  - Nếu ba tập hoàn toàn rời nhau (disjoint), thì $P(A \cap B \cap C) = 0$
  - Đây là giới hạn thấp nhất có thể về mặt xác suất $\implies$ luôn đúng:

$$P(A \cap B \cap C) \geq 0$$

- Nhưng nếu tổng xác suất của ba biến cố vượt quá 2 (tức $a + b + c > 2$), thì:
  - Không thể rời nhau hoàn toàn.
  - Vì toàn bộ không gian mẫu chỉ có tổng xác suất bằng 1, nên phần chồng lặp bắt buộc phải xảy ra.
  - Khi đó, phần giao ba tập phải mang một phần xác suất dương.
- Vậy trong trường hợp tổng lớn hơn 2, thì phần giao ít nhất phải là:

$$P(A \cap B \cap C) \geq a + b + c - 2$$

- Và như mọi xác suất, giá trị này không thể âm:

$$P(A \cap B \cap C) \geq \max(0, a + b + c - 2)$$

Chạy demo bài toán nâng cao 1

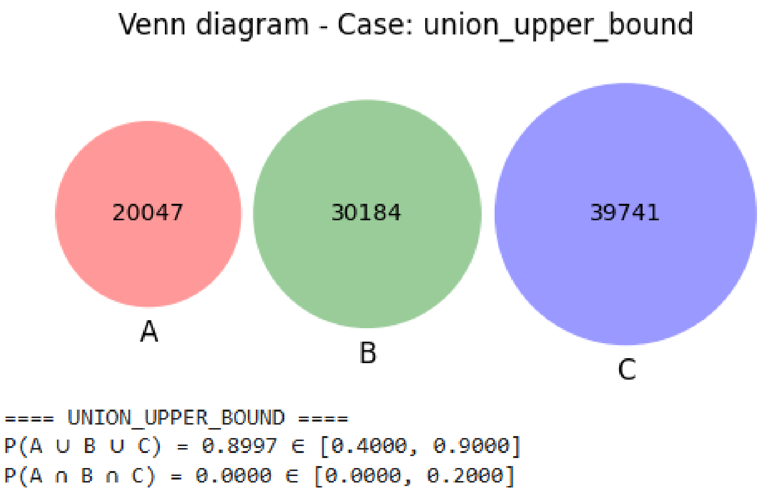
Code Python chạy demo cho bài toán

```python
!pip install matplotlib matplotlib-venn
import random
import matplotlib.pyplot as plt
from matplotlib_venn import venn3

def simulate_and_plot(P_A, P_B, P_C, case_type, N=100000):
    A, B, C = [], [], []

    for _ in range(N):
        r = random.random()

        if case_type == "union_upper_bound":  # TH1: rời nhau
            A.append(r < P_A)
            B.append(P_A <= r < P_A + P_B)
            C.append(P_A + P_B <= r < P_A + P_B + P_C)

        # TH2: C là tập con của B, B là tập con của A
        elif case_type == "union_lower_bound":
            a = r < P_A
            b = r < P_B if a else False
            c = r < P_C if b else False
            A.append(a)
            B.append(b)
            C.append(c)

        # TH3: C là tập con của B, B là tập con của A
        elif case_type == "inter_upper_bound":
            c = r < P_C
            b = True if c else (r < P_B)
            a = True if b else (r < P_A)
```

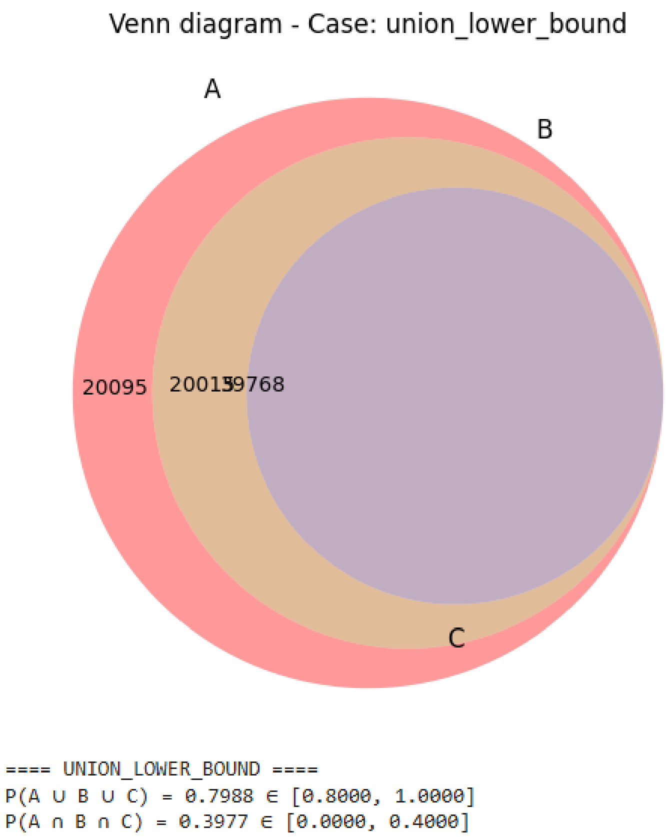
```
            A.append(a)
            B.append(b)
            C.append(c)

        elif case_type == "inter_lower_bound":  # TH4: rời nhau
            A.append(r < P_A)
            B.append(P_A <= r < P_A + P_B)
            C.append(P_A + P_B <= r < P_A + P_B + P_C)

    # Đếm từng vùng trong biểu đồ Venn
    only_A = sum(a and not b and not c for a, b, c in zip(A, B, C))
    only_B = sum(b and not a and not c for a, b, c in zip(A, B, C))
    only_C = sum(c and not a and not b for a, b, c in zip(A, B, C))
    A_B = sum(a and b and not c for a, b, c in zip(A, B, C))
    A_C = sum(a and c and not b for a, b, c in zip(A, B, C))
    B_C = sum(b and c and not a for a, b, c in zip(A, B, C))
    ABC = sum(a and b and c for a, b, c in zip(A, B, C))

    # Biểu đồ Venn
    plt.figure(figsize=(6,6))
    venn3(subsets = (only_A, only_B, A_B, only_C, A_C, B_C, ABC),
          set_labels = ('A', 'B', 'C'))
    plt.title(f"Venn diagram - Case: {case_type}")
    plt.show()

    # Tính xác suất thực nghiệm
    P_union = sum(a or b or c for a, b, c in zip(A, B, C)) / N
    P_inter = sum(a and b and c for a, b, c in zip(A, B, C)) / N

    # Cận lý thuyết
    upper_union = min(1, P_A + P_B + P_C)
    lower_union = max(P_A, P_B, P_C, P_A + P_B + P_C - 2)
```

```
    upper_inter = min(P_A, P_B, P_C)
    lower_inter = max(0, P_A + P_B + P_C - 2)

    # In kết quả
    print(f"==== {case_type.upper()} ====")
    print(f"P(A ∪ B ∪ C) = {P_union:.4f} ∈ [{lower_union:.4f}, {upper_union:.4f}]")
    print(f"P(A ∪ B ∪ C) = {P_inter:.4f} ∈ [{lower_inter:.4f}, {upper_inter:.4f}]\n")

# TH1: dấu bằng tại cận trên của hợp
simulate_and_plot(0.2, 0.3, 0.4, "union_upper_bound")

# TH2: dấu bằng tại cận dưới của hợp
simulate_and_plot(0.8, 0.6, 0.4, "union_lower_bound")

# TH3: dấu bằng tại cận trên của giao
simulate_and_plot(0.9, 0.7, 0.5, "inter_upper_bound")

# TH4: dấu bằng tại cận dưới của giao
simulate_and_plot(0.1, 0.3, 0.4, "inter_lower_bound")
```

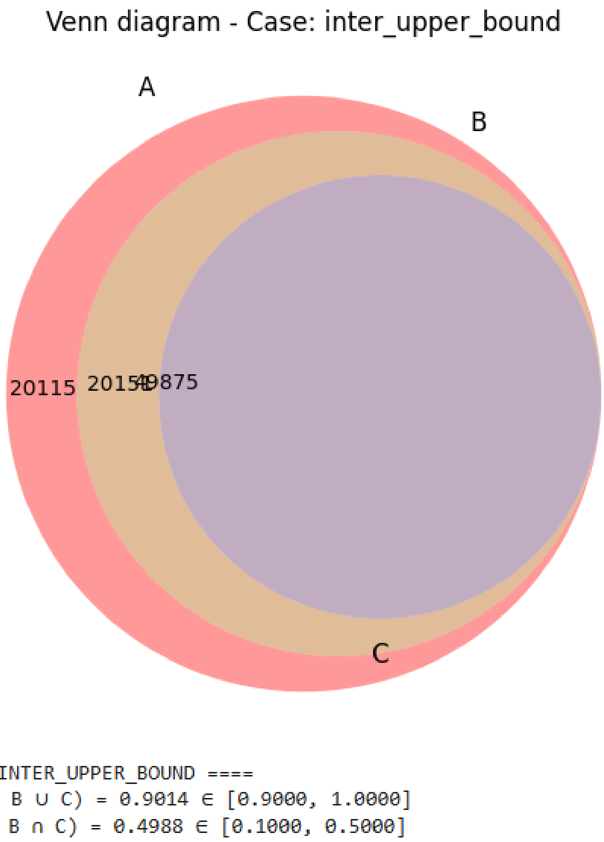
Kết quả xuất ra màn hình:



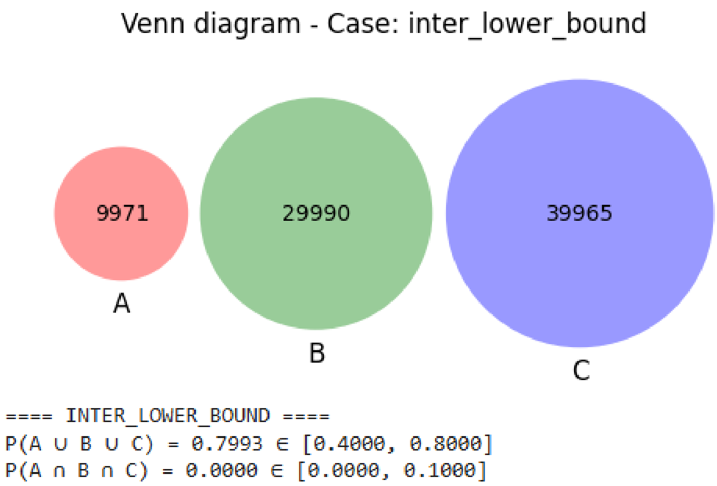
Hình 4: Mô phỏng trường hợp xảy ra dấu bằng tại giới hạn trên của $P(A \cup B \cup C)$



Hình 5: Mô phỏng trường hợp xảy ra dấu bằng tại giới hạn dưới của $P(A \cup B \cup C)$



Hình 6: Mô phỏng trường hợp xảy ra dấu bằng tại giới hạn trên của $P(A \cap B \cap C)$



Hình 7: Mô phỏng trường hợp xảy ra dấu bằng tại giới hạn dưới của $P(A \cap B \cap C)$

Bài tập nâng cao 2. Bài toán nâng cao dựa trên bài tập 6 (phần 2.6.8). Bài toán mô tả như sau:

Bài toán: Phát hiện gian lận qua mô hình Bayes

Một công ty tài chính đang sử dụng hệ thống phát hiện gian lận giao dịch dựa trên các biến sau:

- F: Biến nhị phân cho biết giao dịch có gian lận hay không (1 = gian lận, 0 = bình thường).
- L: Biến nhị phân cho biết liệu giao dịch được thực hiện từ vị trí lạ không (1 = vị trí lạ, 0 = vị trí quen thuộc).
- T: Biến nhị phân cho biết thời điểm giao dịch có phải vào giờ bất thường không (1 = bất thường, 0 = bình thường).

Giả sử mô hình thỏa mãn mối quan hệ:

- L và T độc lập có điều kiện khi biết F.
- Sơ đồ phụ thuộc có thể được biểu diễn như:

$$F \to L$$

$$F \to T$$

Yêu cầu:

1. Viết biểu thức xác suất đồng thời $P(F, L, T)$ dựa trên cấu trúc phụ thuộc nêu trên.
2. Sử dụng định lý Bayes để viết công thức tính $P(F = 1|L = 1, T = 1)$.
3. Giả sử có các xác suất sau:
   - $P(F = 1) = 0.01$
   - $P(L = 1|F = 1) = 0.9$, $P(L = 1|F = 0) = 0.1$
   - $P(T = 1|F = 1) = 0.8$, $P(T = 1|F = 0) = 0.2$

   Tính giá trị cụ thể của $P(F = 1|L = 1, T = 1)$.

Lời giải bài toán nâng cao 2

1. Viết biểu thức xác suất đồng thời $P(F, L, T)$

   Biểu thức xác suất đồng thời:

$$P(F, L, T) = P(F) \cdot P(L|F) \cdot P(T|F)$$

2. Sử dụng định lý Bayes để tính $P(F = 1|L = 1, T = 1)$

   Theo định lý Bayes:

$$P(F = 1|L = 1, T = 1) = \frac{P(F = 1) \cdot P(L = 1|F = 1) \cdot P(T = 1|F = 1)}{P(L = 1, T = 1)}$$

   Mẫu số $P(L = 1, T = 1)$ được tính bằng cách tổng trên tất cả giá trị của $F \in \{0, 1\}$:

$$P(L = 1, T = 1) = \sum_{f \in \{0,1\}} P(F = f) \cdot P(L = 1|F = f) \cdot P(T = 1|F = f)$$

3. Tính giá trị cụ thể của $P(F = 1|L = 1, T = 1)$ với các giá trị xác suất đã cho

   Tính mẫu số:

$$P(L = 1, T = 1) = 0.01 \cdot 0.9 \cdot 0.8 + 0.99 \cdot 0.1 \cdot 0.2 = 0.0072 + 0.0198 = 0.027$$

   Tính tử số:

$$\text{Tử số} = 0.01 \cdot 0.9 \cdot 0.8 = 0.0072$$

   Suy ra:

$$P(F = 1|L = 1, T = 1) = \frac{0.0072}{0.027} \approx 0.2667$$

   Ta nhận thấy mặc dù khả năng gian lận gốc chỉ là 1%, nhưng khi thấy **vị trí lạ + thời điểm bất thường**, xác suất gian lận tăng lên $\approx 26.67\%$!

Chạy demo bài toán nâng cao 2

Code Python chạy demo cho bài toán

```python
# Xác suất đã cho
P_F1 = 0.01
P_F0 = 1 - P_F1
```

```python

P_L1_given_F1 = 0.9
P_L1_given_F0 = 0.1

P_T1_given_F1 = 0.8
P_T1_given_F0 = 0.2

# Tính tử số: P(F=1, L=1, T=1)
numerator = P_F1 * P_L1_given_F1 * P_T1_given_F1

# Tính mẫu số: P(L=1, T=1)
P_L1_T1 = (
    P_F1 * P_L1_given_F1 * P_T1_given_F1 +
    P_F0 * P_L1_given_F0 * P_T1_given_F0
)

# Bayes: P(F=1 | L=1, T=1)
P_F1_given_L1_T1 = numerator / P_L1_T1

print(f"Xác suất giao dịch là gian lận (P(F=1 | L=1, T=1)):
{P_F1_given_L1_T1:.4f}")

```

Kết quả xuất ra màn hình:

```
Xác suất giao dịch là gian lận (P(F=1 | L=1, T=1)): 0.2667
```

Hình 8: Kết quả thực hiện demo tính toán

Bài tập nâng cao 3. Bài nâng cao dựa trên bài toán xác suất xét nghiệm y tế ở EXCERCISE 7, kết hợp thêm các yếu tố thực tế như chi phí xét nghiệm, tối ưu quyết định, và mở rộng thành bài toán ra quyết định dựa trên Bayesian decision theory.

Tóm tắt bài toán - INPUT bài toán

1. $H = 1$: Bệnh nhân mắc bệnh (nhiễm bệnh).
2. $H = 0$: Bệnh nhân khỏe mạnh.
3. D_i: Kết quả của xét nghiệm thứ i ($D_i = 1$ nếu dương tính, $D_i = 0$ nếu âm tính).
4. Xác suất:
   - Xác suất tiên nghiệm: $P(H = 1) = 0.0015$ ($\Rightarrow P(H = 0) = 0.9985$).
   - Xét nghiệm dương tính giả (False Positive): $P(D_i = 1 \mid H = 0) = 0.1$.
   - $\Rightarrow$ Xét nghiệm âm tính thật (True Positive): $P(D_i = 0 \mid H = 0) = 0.9$.
   - Xét nghiệm âm tính giả (False Negative): $P(D_i = 0 \mid H = 1) = 0.01$.
   - $\Rightarrow$ Xét nghiệm dương tính thật (True positive): $P(D_i = 1 \mid H = 1) = 0.99$.
5. Xét nghiệm độc lập theo H:
   - $P(D_1, D_2, ...D_k \mid H = 1) = P(D_1 \mid H = 1)P(D_2 \mid H = 1)...P(D_k \mid H = 1)$
   - $P(D_1, D_2, ...D_k \mid H = 0) = P(D_1 \mid H = 0)P(D_2 \mid H = 0)...P(D_k \mid H = 0)$
6. Chi phí:
   - Điều trị đúng chi phí: 0$
   - Điều trị nhầm người không mắc bệnh, bệnh viện phải tốn chi phí: 500$
   - Bỏ sót điều trị người mắc bệnh, bệnh viện phải tốn chi phí: 10,000$
   - Một lần xét nghiệm bệnh nhân, bệnh viện phải tốn chi phí: 50$
7. Ràng buộc bài toán: Được phép xét nghiệm tối đa 3 lần cho mỗi bệnh nhân, với các lần xét nghiệm độc lập theo H. Sau mỗi lần xét nghiệm, có thể quyết định:
   - (A): Dừng lại và điều trị.
   - (B): Dừng lại và không điều trị.

- (C): Tiếp tục xét nghiệm lần 3 (lần cuối).

Yêu cầu - OUPUT của bài toán:

1. Tính xác suất hậu nghiệm $P(H = 1|D_1, D_2, ...D_k)$ sau mỗi bước.
2. Tính chi phí kỳ vọng cho mỗi hành động A, B, C.
3. Mô phỏng thuật toán giúp chọn hành động tối ưu ở mỗi bước xét nghiệm nhằm giảm thiểu chi phí kỳ vọng.
4. Sử dụng mô phỏng Monte Carlo để kiểm nghiệm chiến lược trên với 1 triệu bệnh nhân ngẫu nhiên.

Lời giải bài toán nâng cao 3

Phần 1: Tính xác suất hậu nghiệm $P(H = 1|D_1, D_2, ...D_k)$

- Mô hình hóa bài toán:
  - $H = 1$: Có bệnh, xác suất ban đầu $P(H = 1) = 0.0015$
  - $D_i \in \{0, 1\}$: Kết quả test thứ i (1 = dương tính, 0 = âm tính)
  - Các test độc lập có điều kiện theo H
- Công thức Bayes: Sau k lần xét nghiệm, với s lần dương tính:

$$P(H = 1|D_1, ..., D_k) = \frac{P(D_1, ..., D_k|H = 1) \cdot P(H = 1)}{P(D_1, ..., D_k)}$$

$$P(H = 1|D_1, ..., D_k) = \frac{P(D_1 = ... = D_s = 1|H = 1).P(D_{s+1} = ... = D_k = 0|H = 1) \cdot P(H = 1)}{P(D_1, ..., D_k|H = 1) + P(D_1, ..., D_k|H = 0)}$$

$$(D_1 = ... = D_s = 1, D_{s+1} = ... = D_k = 0)$$

Vì các test độc lập có điều kiện theo H, ta có:

$$\begin{aligned} P(D_1, ..., D_k|H = 1) &= P(D_1 = ... = D_s = 1|H = 1).P(D_{s+1} = ... = D_k = 0|H = 1) \\ &= (P(D_1 = 1|H = 1)^s.(P(D_k = 0|H = 1)^{k-s} \\ &= (0.99)^s \cdot (0.01)^{k-s} \end{aligned}$$

$$\begin{aligned} P(D_1, ..., D_k|H = 0) &= P(D_1 = ... = D_s = 1|H = 0)P(D_{s+1} = ... = D_k = 0|H = 0) \\ &= (P(D_1 = 1|H = 0))^s.(P(D_k = 0|H = 0))^{k-s} \\ &= (0.10)^s \cdot (0.90)^{k-s} \end{aligned}$$

$$\Rightarrow P(H = 1|D_1, ..., D_k) = \frac{(0.99)^s(0.01)^{k-s} \cdot 0.0015}{(0.99)^s(0.01)^{k-s} \cdot 0.0015 + (0.10)^s(0.90)^{k-s} \cdot 0.9985}$$

$$bayes\_posterior(s, k) = P(H = 1|D_1, ..., D_k) \tag{1}$$

Phần 2: Tính chi phí kỳ vọng của các hành động

1. Giả định của mô hình bài toán:

   - Điều trị đúng: 0$ chi phí
   - Điều trị nhầm người không mắc bệnh, bệnh viện phải tốn chi phí: 500$
   - Bỏ sót điều trị người mắc bệnh, bệnh viện phải tốn chi phí: 10,000$
   - Một lần xét nghiệm bệnh nhân, bệnh viện phải tốn chi phí: 50$

2. Tính chi phí kỳ vọng:

   - Giả sử tại bước k, số dương là s, gọi hàm: $p = bayes\_posterior(s, k)$
     - Nếu hành động (A): Điều trị thì gọi hàm tính chi phí: $cost\_A = (1 - p) * 500 + k * 50$;
     - Nếu hành động (B): Không điều trị thì gọi hàm tính chi phí: $cost\_B = p * 10000 + k * 50$;
     - Nếu hành động (C) tiếp tục test→ xây dựng cây quyết định đệ quy ở Phần 3.

Phần 3: Xây dựng thuật toán chọn hành động tối ưu

Pseudocode tính chi phí kỳ vọng tối ưu: Sử dụng 'đệ quy' để duyệt theo cây quyết định:

```
HÀM EXPECTED_COST(s, k):
    NẾU k đạt giới hạn test:
        p = BAYES_POSTERIOR(s, k)
        cost_A = điều trị ngay
```

```
        cost_B = bỏ qua
        TRẢ VỀ chi phí nhỏ nhất giữa A và B
    p = BAYES_POSTERIOR(s, k)
    cost_A = điều trị ngay
    cost_B = bỏ qua
    # Xét thêm 1 test
    p_pos = xác suất test dương kế tiếp
    p_neg = 1 - p_pos
    cost_pos = EXPECTED_COST(s + 1, k + 1)
    cost_neg = EXPECTED_COST(s,     k + 1)
    cost_test_thêm = chi phí test + kỳ vọng tương lai
    TRẢ VỀ chi phí nhỏ nhất giữa A, B và test thêm
```

Giải thích

- Hàm expected_cost(s, k, max_tests=3)
  - s số test dương tính đã thu được
  - k: số test đã làm
  - max_tests: giới hạn số test tối đa (ở đây là 3)
  - Mục tiêu là trả về chi phí thấp nhất có thể nếu đang ở trạng thái (s, k)
- Khi đã làm đủ test: $if \quad k == max\_tests$ thì không thể test thêm nữa mà phải ra quyết định lựa chọn giữa:
  - A: điều trị
  - B: không điều trị
- Trường hợp chọn điều trị ngay sau 1 test:
  - Điều trị nhầm thì chi phí bệnh viện phải trả: $cost\_A = (1 - p) * 500 + k * 50$
  - Không điều trị người bệnh thì chi phí bệnh viện phải trả: $cost\_B = p * 10000 + k * 50$
- Trường hợp làm thêm 1 test. Vì chưa test đủ, chọn hành động:TEST thêm lần nữa

  - Xác suất test tiếp theo dương hoặc âm
    * Dương tính $p\_next\_pos = p * 0.99 + (1 - p) * 0.10$
    * Âm tính $p\_next\_neg = 1 - p\_next\_pos$
  - Chi phí kỳ vọng tương ứng cho hai khả năng
    * Nếu kết quả dương tính $cost\_next\_pos = expected\_cost(s + 1, k + 1)$
    * Nếu âm tính: $cost\_next\_neg = expected\_cost(s, k + 1)$
  - Tổng chi phí kỳ vọng khi làm test thêm:

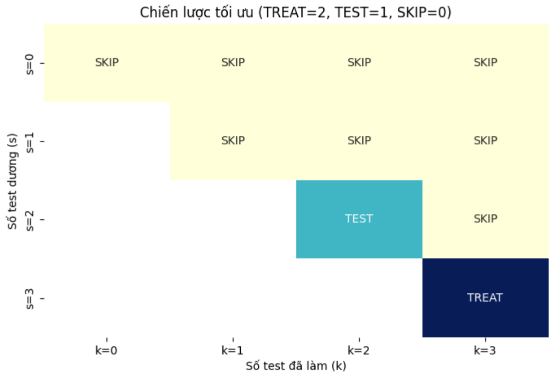
$$cost\_C = k*50+50+(p\_next\_pos*cost\_next\_pos+p\_next\_neg*cost\_next\_neg)$$

- So sánh và chọn hành động tối ưu (chi phí thấp nhất)

$$return min(cost\_A, cost\_B, cost\_C)$$

Code Python cho bước tính chi phí kỳ vọng (cây quyết định)

```python
from functools import lru_cache
@lru_cache(maxsize=None)
def expected_cost(s, k, max_tests=3):
    if k == max_tests:
        # Đã test đủ, chọn giữa A và B
        p = bayes_posterior(s, k)
        cost_A = (1 - p) * 500 + k * 50
        cost_B = p * 10000 + k * 50
        return min(cost_A, cost_B)
    # Nếu điều trị ngay
    p = bayes_posterior(s, k)
    cost_A = (1 - p) * 500 + k * 50
    cost_B = p * 10000 + k * 50
    # Nếu test thêm 1 lần
    # Kỳ vọng theo kết quả test tiếp theo (dương hoặc âm)
```



Hình 9: Mô phỏng, trực quan hóa cây quyết định

```
    p_next_pos = p * 0.99 + (1 - p) * 0.10
    p_next_neg = 1 - p_next_pos
    cost_next_pos = expected_cost(s + 1, k + 1)
    cost_next_neg = expected_cost(s, k + 1)
    cost_C = k * 50 + 50 + (p_next_pos * cost_next_pos + p_next_neg * cost_next_neg)
    return min(cost_A, cost_B, cost_C)
```

Mô phỏng, trực quan hóa cây quyết định để tìm chiến lược điều trị tối ưu

Phần 4: Mô phỏng Monte Carlo chi phí trung bình mỗi bệnh nhân mà bệnh viện phải chi trả

Ý tưởng

- Sinh $H \sim \text{Bernoulli}(P\_H1)$
- Mỗi bước test sinh ngẫu nhiên dựa trên H
- Áp dụng phương thức $expected\_cost()$ để ra quyết định sau mỗi test
- Tính tổng chi phí thật tế trên 1 triệu bệnh nhân

Pseudocode cho mô phỏng Monte Carlo:

```
INPUT: strategy_matrix[s][k]
CONSTANTS: COST_TEST, COST_TREAT_HEALTHY, COST_SKIP_SICK, P_H1, TPR, FPR,
NUM_PATIENTS
FUNCTION simulate_patient():
```

```
    is_sick ← random() < P_H1
    s, k, cost ← 0, 0, 0
    LOOP:
        action ← strategy_matrix[s][k]
        IF action == "TREAT":
            cost += COST_TREAT_HEALTHY if not is_sick
            RETURN cost + k × COST_TEST
        IF action == "SKIP":
            cost += COST_SKIP_SICK if is_sick
            RETURN cost + k × COST_TEST
        IF action == "TEST":
            test_result ← random() < (TPR if is_sick else FPR)
            k ← k + 1
            s ← s + 1 if test_result
            cost += COST_TEST
FUNCTION main():
    total_cost ← 0
    REPEAT NUM_PATIENTS TIMES:
        total_cost += simulate_patient()
    PRINT "Chi phí trung bình mỗi bệnh nhân mà bệnh viện phải chi trả:",
    total_cost / NUM_PATIENTS, "$"
```

Code Python chạy demo cho mô phỏng Monte Carlo

```python
import numpy as np
# Tham số
NUM_PATIENTS = 1_000_000
np.random.seed(42)  # để kết quả ổn định
# Lấy ma trận chiến lược đã tính ở bước trước: strategy_matrix[s, k]
# Đảm bảo bạn đã chạy đoạn mã trước đó để có biến `strategy_matrix`
# Hàm mô phỏng chi phí cho một bệnh nhân
def simulate_patient(strategy_matrix, max_tests=3):
    is_sick = np.random.rand() < P_H1
```

```
    k = 0  # số lần test đã làm
    s = 0  # số test ra dương tính
    total_cost = 0
    while True:
        action = strategy_matrix[s, k]
        if action == 'TREAT':
            total_cost += COST_TREAT_HEALTHY if not is_sick else 0
            total_cost += k * COST_TEST
            return total_cost
        elif action == 'SKIP':
            total_cost += COST_SKIP_SICK if is_sick else 0
            total_cost += k * COST_TEST
            return total_cost
        elif action == 'TEST':
            test_result = None
            if is_sick:
                test_result = np.random.rand() < TPR
            else:
                test_result = np.random.rand() < FPR
            k += 1
            if test_result:
                s += 1
            total_cost += COST_TEST
        else:
            raise ValueError(f"Unknown action: {action}")
# Mô phỏng nhiều bệnh nhân
total_cost = 0
for _ in range(NUM_PATIENTS):
    total_cost += simulate_patient(strategy_matrix)
average_cost = total_cost / NUM_PATIENTS
print(f"Chi phí trung bình cho mỗi bệnh nhân mà bệnh viện phải chi trả:
{average_cost:.2f}$")
```

Kết quả xuất ra màn hình là:

Chi phí trung bình cho mỗi bệnh nhân mà bệnh viện phải chi trả: 15.07$

5 Kết luận

Tiểu luận đã trình bày ngắn gọn các kiến thức nền tảng của xác suất thống kê như xác suất, kỳ vọng, phương sai, độ lệch chuẩn, ... và vai trò ứng dụng của xác suất trong học máy nói riêng và trong ngành Khoa học máy tính nói chung. Chẳng hạn, trong các ví dụ và bài tập nhóm đã đề cập đến vấn đề bất định trong Machine Learning. Bất định có 2 dạng là:

1. Bất định aleatoric: Là sự bất định vốn có trong dữ liệu, do tính ngẫu nhiên không thể loại bỏ, ví dụ như kết quả của một lần tung đồng xu.

2. Bất định epistemic: Là sự bất định do thiếu hiểu biết hoặc dữ liệu không đầy đủ, có thể giảm bớt bằng cách thu thập thêm dữ liệu.

Để giảm sự bất định epistemic trong machine learning thì việc thu thập thêm dữ liệu giúp giảm bất định là cần thiết, nhưng tốc độ giảm thường chậm, theo tỉ lệ giảm là $\frac{1}{\sqrt{n}}$, với n là số lượng mẫu. Tức là, nếu tăng gấp đôi dữ liệu thì chỉ giảm bất định một cách tương đối nhỏ. Thông qua thực nghiệm trong phần bài tập 1, 2 nhóm rút ra kết luận rằng sự bất định sẽ giảm ở tỷ lệ nhất định khi dataset tăng lên.

Tiểu luận cũng trình bày định lý Bayes như là một công cụ quan trọng trong thống kê, giúp cập nhật niềm tin dựa trên dữ liệu mới. Xác suất tiên nghiệm và xác suất có điều kiện đóng vai trò quan trọng và hữu ích trong các ứng dụng như chẩn đoán y khoa. Để minh họa cho điều này, nhóm đã chạy thực nghiệm được một số bài toán thực tế như bài toán ra quyết định hành động tiếp theo dựa trên định lý Bayes sau khi thực hiện k lần xét nghiệm HIV.

Cuối cùng, tiểu luận đã ứng dụng của kỳ vọng và xác suất vào trong các bài toán thực tế để xây dựng mô hình hóa xác suất, như mô hình thực nghiệm danh mục đầu tư chứng khoán tối ưu trong bài tập 8. Từ các bài toán thực nghiệm này đã nêu bật tầm quan trọng của kỳ vọng và phương sai trong việc hiểu và mô hình hóa phân phối xác suất, cung cấp cái nhìn sâu sắc về hành vi của các biến ngẫu nhiên.

Tài liệu

[1] https://d2l.ai/chapter_preliminaries/probability.html

[2] Probability and Statistics for Computer Science , David Forsyth, (2018).

[3] Alayrac, J.-B., Donahue, J., Luc, P., Miech, A., Barr, I., Hasson, Y., ... et al. (2022). Flamingo: a visual language model for few-shot learning. ArXiv:2204.14198.

[4] Alsallakh, B., Kokhlikyan, N., Miglani, V., Yuan, J., & Reblitz-Richardson, O. (2020). Mind the PAD – CNNs can develop blind spots. ArXiv:2010.02178.

[5] Anil, R., Dai, A. M., Firat, O., Johnson, M., Lepikhin, D., Passos, A., ... et al. (2023). PaLM 2 Technical Report. ArXiv:2305.10403.