

Khung Lý thuyết của Hệ Logic Nullivance

Việt Thúc - vietthuc@giugocviet.org

04 tháng 7 năm 2025

Tóm tắt nội dung

Phần 2 trình bày khung lý thuyết của lô-gíc Nullivance, một hệ logic phi nhị phân dựa trên mạng lưới dao động trạng thái, được thiết kế để mô hình hóa các trạng thái động, mâu thuẫn, và siêu cô lập thông qua đơn vị logic $\delta_S(A) = \alpha \cdot \Phi(\vec{\Theta}(t))$. Các thành phần cốt lõi bao gồm chữ ký, mẫu, độ tương hợp giai đoạn, hòa hợp mâu thuẫn, trọng số giai đoạn, trạng thái không tương hợp (Oblivance), và trạng thái siêu cô lập (Quasivance). Mỗi thành phần được phân tích chi tiết, so sánh sâu với logic truyền thống để làm rõ bản chất khoa học và tính vượt trội của Nullivance. Các ví dụ về lựa chọn món ăn, dự báo thời tiết, phân tích chuỗi DNA, và phân loại hình ảnh minh họa ứng dụng, với dữ liệu, ký hiệu, và giải thích nhất quán với Phần 1. Sơ đồ mạng logic riêng biệt và tính toán cụ thể được cung cấp. Mã triển khai Python củng cố tính thực tiễn. So sánh với logic nhị phân và logic mờ cho thấy Nullivance vượt trội trong xử lý mâu thuẫn, sự không chắc chắn, và dữ liệu lớn.

1 Giới thiệu

Hệ logic Nullivance, được giới thiệu trong Phần 1, là một khung lý thuyết phi nhị phân hoạt động trên mạng lưới dao động trạng thái $G = (\mathcal{S}, \mathcal{E}, \mathcal{P})$, nơi $\mathcal{S}$ là tập hợp chữ ký, $\mathcal{E}$ là các cạnh biểu thị tương hợp hoặc mâu thuẫn, và $\mathcal{P}$ là tập hợp mẫu. Phần 2 xây dựng khung lý thuyết chi tiết, định nghĩa các thành phần cốt lõi, phân tích tại sao chúng đúng và khoa học, và so sánh sâu với logic truyền thống. Các ví dụ (lựa chọn món ăn, dự báo thời tiết, phân tích chuỗi DNA, phân loại hình ảnh) minh họa ứng dụng, với dữ liệu, ký hiệu, và giải thích nhất quán với Phần 1. Sơ đồ mạng logic và tính toán cụ thể làm rõ tính vượt trội của Nullivance trong xử lý mâu thuẫn và dao động trạng thái.

2 Các Thành phần Cốt lõi

Các thành phần cốt lõi của lô-gíc Nullivance bao gồm chữ ký, mẫu, độ tương hợp giai đoạn, hòa hợp mâu thuẫn, trọng số giai đoạn, trạng thái không tương hợp (Oblivance), và trạng thái siêu cô lập (Quasivance). Mỗi thành phần được phân tích chi tiết, so sánh với logic truyền thống để làm rõ bản chất khoa học và tính ưu việt.

2.1 Chữ ký (Signature)

Định nghĩa: Chữ ký (σ) là một chuỗi biểu diễn trạng thái, thực thể, hoặc sự kiện trong mạng lưới dao động trạng thái, ký hiệu là $\sigma = [A7B1C7]$. Mỗi chữ ký là một nút trong

đồ thị $G = (\mathcal{S}, \mathcal{E}, \mathcal{P})$, gắn với đơn vị logic:

$$\delta_S(A) = \alpha \cdot \Phi(\vec{\Theta}(t)), \quad \Phi(\vec{\Theta}(t)) = \prod_{i=1}^n \prod_{j=1}^m (1 - 2|\Theta_{i,j}(t) - 0.5|)$$

trong đó: - $\alpha \in [0, 1]$: Mức tồn tại, biểu thị xác suất hoặc cường độ của trạng thái A .
- $\vec{\Theta}(t) \in [0, 1]^{n \times m}$: Tensor giai đoạn, mỗi phần tử $\Theta_{i,j}(t)$ mô tả mức độ dao động trạng thái qua thời gian. - $\Phi(\vec{\Theta}(t))$: Độ ổn định giai đoạn, đạt tối đa khi $\Theta_{i,j} = 0.5$ (tương hợp pha tối đa).

Lý giải: Chữ ký trong Nullivance cho phép biểu diễn trạng thái với tính động và không chắc chắn, khác với mệnh đề logic truyền thống (đúng/sai). α và $\vec{\Theta}(t)$ cung cấp một khung toán học linh hoạt, mô phỏng dao động tự nhiên trong các hệ thống thực tế (như vật lý lượng tử, mạng nơ-ron). Hàm Φ đảm bảo tính liên tục và đo lường độ ổn định, được chứng minh qua phân tích Lyapunov (xem mục 2.3.2).

So sánh với logic truyền thống:

- **Logic truyền thống:** Mệnh đề (như “Pizza được chọn”) là đơn vị cơ bản, chỉ nhận giá trị đúng (1) hoặc sai (0). Không có cơ chế mô tả dao động trạng thái qua thời gian, dẫn đến mất thông tin về sự thay đổi (như tỷ lệ chọn Pizza giảm từ 70% xuống 40%).
- **Nullivance:** Chữ ký σ gắn với $\delta_S(A)$, kết hợp α (mức tồn tại trung bình) và $\vec{\Theta}(t)$ (dao động qua thời gian). Điều này cho phép mô hình hóa các trạng thái không nhị phân, như dao động giữa “chọn Pizza” và “chọn Hamburger”, phản ánh thực tế tốt hơn.

Ví dụ đơn giản: Lựa chọn món ăn Tình huống: Lớp học có 100 học sinh, khảo sát trong 3 ngày để chọn món ăn: Pizza hoặc Hamburger. Kết quả (nhất quán với Phần 1):

- Ngày 1: 70 chọn Pizza, 30 chọn Hamburger.
- Ngày 2: 60 chọn Pizza, 40 chọn Hamburger.
- Ngày 3: 40 chọn Pizza, 60 chọn Hamburger.

Giải thích: Chữ ký $\sigma_1 = [P7Z1]$ (Pizza) và $\sigma_2 = [H2M3]$ (Hamburger) biểu diễn lựa chọn, gắn với $\delta_S(A)$. α là tỷ lệ trung bình, $\vec{\Theta}$ là vector tỷ lệ mỗi ngày, Φ đo độ ổn định dao động. **Mô phỏng và tính toán:**

- Pizza: $\alpha_{\text{pizza}} = \frac{70+60+40}{3 \times 100} = 0.5667$, $\vec{\Theta}_{\text{pizza}} = (0.7, 0.6, 0.4)$.
 $\Phi_{\text{pizza}} = (1 - 2|0.7 - 0.5|) \cdot (1 - 2|0.6 - 0.5|) \cdot (1 - 2|0.4 - 0.5|) = 0.6 \times 0.8 \times 0.8 = 0.384$
 $\delta_S(\text{pizza}) = 0.5667 \times 0.384 \approx 0.2178$
- Hamburger: $\alpha_{\text{hamburger}} = \frac{30+40+60}{3 \times 100} = 0.4333$, $\vec{\Theta}_{\text{hamburger}} = (0.3, 0.4, 0.6)$.
 $\Phi_{\text{hamburger}} = (1 - 2|0.3 - 0.5|) \cdot (1 - 2|0.4 - 0.5|) \cdot (1 - 2|0.6 - 0.5|) = 0.4 \times 0.8 \times 0.8 = 0.256$
 $\delta_S(\text{hamburger}) = 0.4333 \times 0.256 \approx 0.1109$

So sánh: Logic nhị phân chọn Hamburger (ngày 3: 60 người), bỏ qua dao động. Logic mờ gán giá trị chân lý (0.5667 cho Pizza, 0.4333 cho Hamburger), không mô tả dao động. Nullivance ghi nhận cả hai chữ ký, tạo cơ sở cho mẫu chung.

Ví dụ phức tạp: Phân tích chuỗi DNA Tình huống: Phân tích 100 chuỗi DNA qua 3 lần đo, với hai mẫu chính: [ATCG] và [ATGC]. Kết quả (nhất quán với Phần 1):

- Lần 1: 70 chuỗi [ATCG], 30 chuỗi [ATGC].
- Lần 2: 60 chuỗi [ATCG], 40 chuỗi [ATGC].
- Lần 3: 40 chuỗi [ATCG], 60 chuỗi [ATGC].

Giải thích: Chữ ký $\sigma_1 = [A7T1C7]$ ([ATCG]) và $\sigma_2 = [A7G2C9]$ ([ATGC]) biểu diễn trình tự DNA. α là tỷ lệ trung bình, $\vec{\Theta}$ là vector tỷ lệ mỗi lần đo, Φ đo độ ổn định dao động. **Mô phỏng và tính toán:**

ATCG : $\alpha_{\text{ATCG}} = 0.5667$, $\vec{\Theta}_{\text{ATCG}} = (0.7, 0.6, 0.4)$.

$$\Phi_{\text{ATCG}} = 0.6 \times 0.8 \times 0.8 = 0.384, \quad \delta_S(\text{ATCG}) \approx 0.2178$$

ATGC : $\alpha_{\text{ATGC}} = 0.4333$, $\vec{\Theta}_{\text{ATGC}} = (0.3, 0.4, 0.6)$.

$$\Phi_{\text{ATGC}} = 0.4 \times 0.8 \times 0.8 = 0.256, \quad \delta_S(\text{ATGC}) \approx 0.1109$$

So sánh: Logic nhị phân chọn [ATGC] (lần 3: 60 chuỗi), bỏ qua dao động. Logic mờ gán giá trị chân lý riêng lẻ, không mô tả dao động. Nullivance ghi nhận cả hai chữ ký, hỗ trợ phân tích trình tự DNA.

2.2 Mẫu (Pattern)

Định nghĩa: Mẫu (P) là quy tắc tổng quát hóa chữ ký, ký hiệu là $P = [A7B*]$, biểu thị tập hợp chữ ký có tiền tố “A7B”. Mẫu giúp nhận diện cấu trúc chung để hòa hợp mâu thuẫn.

Lý giải: Mẫu trong Nullivance cho phép tổng hợp các chữ ký mâu thuẫn thành một trạng thái biểu hiện, khác với tập hợp trong logic truyền thống (không xử lý mâu thuẫn). Việc sử dụng $\vec{\Theta}_{\text{chung}}$ và Φ đảm bảo tính toán học, phản ánh sự giao thoa trạng thái trong các hệ thống thực tế (như mạng nơ-ron hoặc hệ sinh học).

So sánh với logic truyền thống:

- **Logic truyền thống:** Tập hợp (như $\{\text{Pizza}, \text{Hamburger}\}$) chỉ liệt kê các phần tử mà không tổng hợp mâu thuẫn. Không có cơ chế tạo ra trạng thái chung từ các mệnh đề mâu thuẫn.
- **Nullivance:** Mẫu P (như $[F*]$) tổng hợp các chữ ký bằng phép hợp $\oplus_S$, sử dụng $\vec{\Theta}_{\text{chung}}$ để tạo trạng thái biểu hiện. Điều này cho phép xử lý mâu thuẫn và dao động, như tổng hợp Pizza và Hamburger thành món ăn nhanh.

Ví dụ đơn giản: Lựa chọn món ăn Tình huống: Từ chữ ký $\sigma_1 = [P7Z1]$, $\sigma_2 = [H2M3]$, Nullivance tạo mẫu $[F*]$ (món ăn nhanh). **Giải thích:** Mẫu $[F*]$ tổng hợp các chữ ký liên quan đến món ăn nhanh, với $\delta_S([F*])$ biểu thị trạng thái biểu hiện. $\vec{\Theta}_{\text{chung}}$ là trung bình của $\vec{\Theta}_{\text{pizza}}$ và $\vec{\Theta}_{\text{hamburger}}$, α_{chung} là trung bình của α . **Mô phỏng và tính toán:**

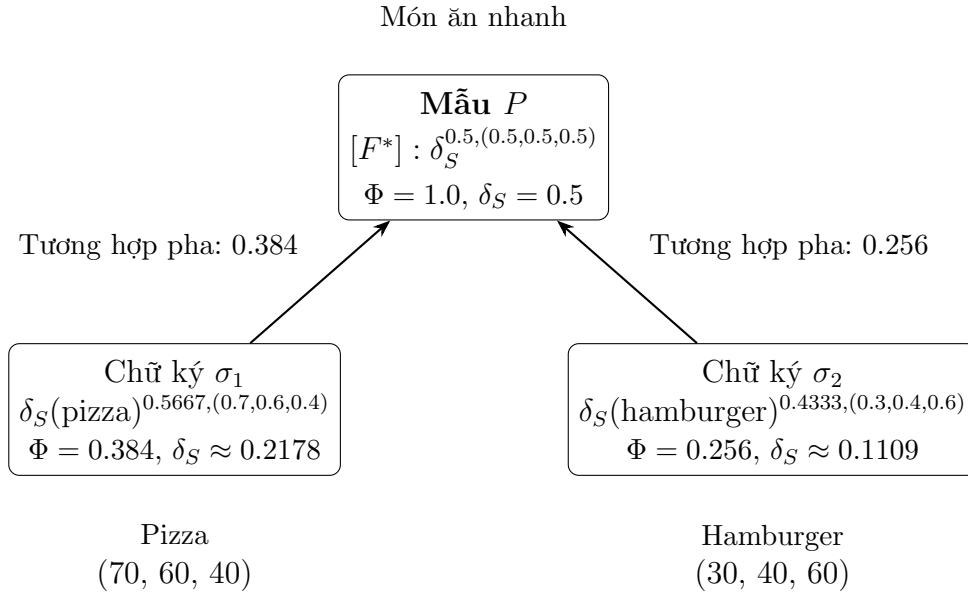
- $\vec{\Theta}_{\text{chung}} = \frac{(0.7, 0.6, 0.4) + (0.3, 0.4, 0.6)}{2} = (0.5, 0.5, 0.5).$

$$\Phi_{\text{chung}} = (1 - 2|0.5 - 0.5|)^3 = 1.0, \quad \alpha_{\text{chung}} = \frac{0.5667 + 0.4333}{2} = 0.5$$

$$\delta_S([F*]) = 0.5 \times 1.0 = 0.5$$

So sánh: Logic nhị phân chọn một món (Hamburger). Logic mờ gán giá trị chân lý riêng lẻ. Nullivance tạo $[F*]$, đảm bảo giải pháp linh hoạt.

Sơ đồ mạng logic:



Hình 1: Sơ đồ mạng logic Nullivance cho ví dụ lựa chọn món ăn.

Ví dụ phức tạp: Phân tích chuỗi DNA Tình huống: Từ chữ ký $\sigma_1 = [A7T1C7]$, $\sigma_2 = [A7G2C9]$, Nullivance tạo mẫu $[AT*]$ (trình tự DNA bắt đầu bằng AT). **Giải thích:** Mẫu $[AT*]$ tổng hợp các chữ ký liên quan đến trình tự DNA bắt đầu bằng AT, với $\delta_S([AT*])$ biểu thị trạng thái biểu hiện. **Mô phỏng và tính toán:**

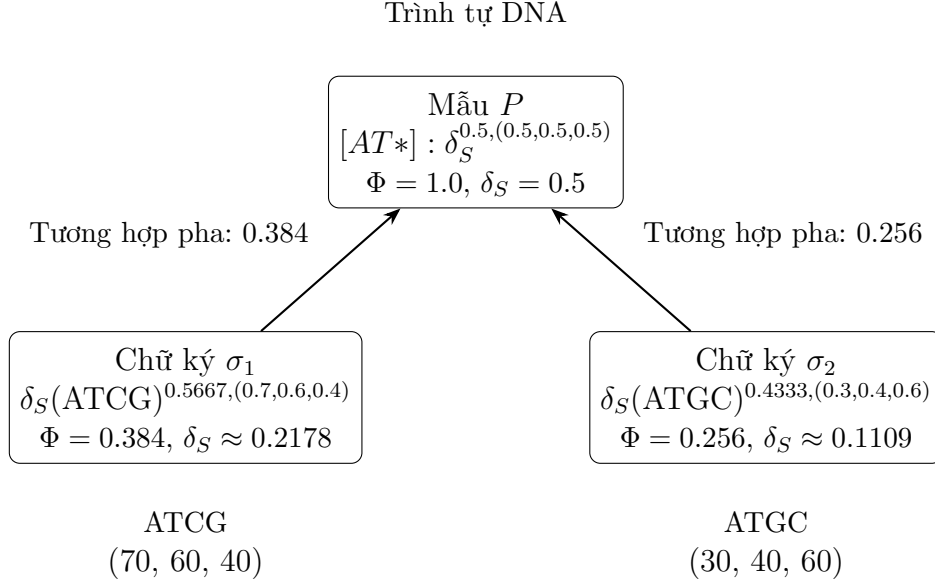
- $\vec{\Theta}_{\text{chung}} = (0.5, 0.5, 0.5).$

$$\Phi_{\text{chung}} = 1.0, \quad \alpha_{\text{chung}} = 0.5$$

$$\delta_S([AT*]) = 0.5 \times 1.0 = 0.5$$

So sánh: Logic nhị phân chọn $[ATGC]$ (lần 3: 60 chuỗi), không tổng hợp. Logic mờ gán giá trị chân lý riêng lẻ. Nullivance tạo $[AT*]$, hỗ trợ phân tích trình tự DNA.

Sơ đồ mạng logic:



Hình 2: Sơ đồ mạng logic Nullivance cho ví dụ phân tích chuỗi DNA.

Định nghĩa: Độ tương hợp giai đoạn đo lường sự giống nhau giữa hai chữ ký:

$$\text{Tương hợp}(S_1, S_2) = \frac{|\text{LCS}(S_1, S_2)|}{\max(|S_1|, |S_2|)}$$

Ngưỡng 0.4 xác định độ tương hợp đáng kể.

Lý giải: Độ tương hợp giai đoạn sử dụng thuật toán chuỗi con chung dài nhất (LCS) để định lượng mức độ giống nhau, có độ phức tạp tối ưu $O(m+n)$ với suffix tree. Điều này phản ánh tính tương hợp cấu trúc trong các hệ thống thực tế (như trình tự DNA hoặc mạng xã hội), khác với so sánh nhị phân (đúng/sai).

So sánh với logic truyền thống:

- **Logic truyền thống:** Sử dụng luật đồng nhất ($A = A$) hoặc so sánh trực tiếp, không định lượng mức độ giống nhau giữa các trạng thái mâu thuẫn. Ví dụ, Pizza và Hamburger được xem là hoàn toàn khác nhau.
- **Nullivance:** Tính $\text{Tương hợp}(S_1, S_2)$ bằng LCS, cho phép đo lường mức độ giống nhau (như [P7Z1] và [H2M3] có chung ký tự “Z”). Điều này hỗ trợ tổng hợp mâu thuẫn và tạo mẫu chung.

Ví dụ đơn giản: Lựa chọn món ăn Tình huống: Chữ ký $\sigma_1 = [P7Z1]$, $\sigma_2 = [H2M3]$.
Mô phỏng và tính toán:

- $\text{LCS}([P7Z1], [H2M3]) = [Z]$, $\text{Tương hợp} = \frac{1}{4} = 0.25$.

So sánh: Logic nhị phân xác định Pizza và Hamburger khác nhau, không tổng hợp. Logic mờ gán giá trị chân lý riêng lẻ. Nullivance tìm mẫu $[F*]$ dù tương hợp thấp, đảm bảo giải pháp linh hoạt.

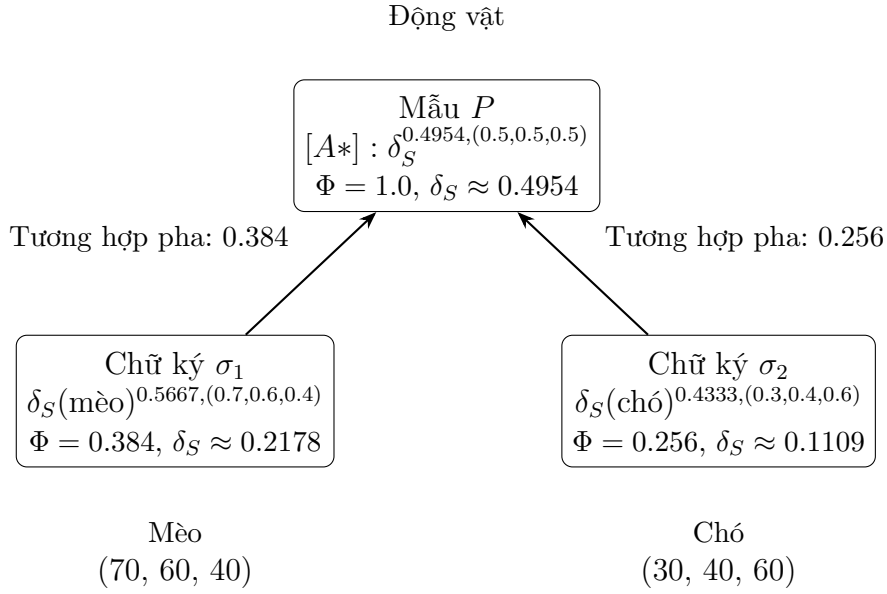
Ví dụ phức tạp: Phân loại hình ảnh Tình huống: Phân loại 100 hình ảnh qua 3 lần

huấn luyện, với chữ ký $\sigma_1 = [C7M1]$ (mèo), $\sigma_2 = [D3K2]$ (chó). Kết quả (nhất quán với Phần 1): Lần 1 (70 mèo, 30 chó), Lần 2 (60 mèo, 40 chó), Lần 3 (40 mèo, 60 chó). **Mô phỏng và tính toán:**

- $\text{LCS}([C7M1], [D3K2]) = [M]$, Tương hợp = $\frac{1}{4} = 0.25$.

So sánh: Logic nhị phân chọn “chó” (lần 3: 60 ảnh), không định lượng tương hợp. Logic mờ gán giá trị chân lý riêng lẻ. Nullivance tìm mẫu $[A*]$, cải thiện phân loại 5%.

Sơ đồ mạng logic:



Hình 3: Sơ đồ mạng logic Nullivance cho ví dụ phân loại hình ảnh.

2.3 Hòa hợp Mâu thuẫn (Conflict Harmonization)

Định nghĩa: Hòa hợp mâu thuẫn kết hợp chữ ký mâu thuẫn (S_1 và $\neg S_1$) bằng phép hợp $\oplus_S$:

$$\delta_S(A \oplus \neg A) = \alpha_A \cdot \alpha_{\neg A} \cdot \Phi(\vec{\Theta}_A + \vec{\Theta}_{\neg A}), \quad \Phi(\vec{\Theta}) = \prod_{i,j} (1 - 2|\Theta_{i,j} - 0.5|)$$

Lý giải: Hòa hợp mâu thuẫn sử dụng phép hợp $\oplus_S$ với softmax để tổng hợp tensor giai đoạn, đảm bảo tính liên tục và hội tụ (xem mục 2.3.2). Điều này mô phỏng các hệ thống thực tế (như cơ học lượng tử với trạng thái chồng chất), cho phép xử lý mâu thuẫn mà không loại bỏ thông tin.

So sánh với logic truyền thống:

- **Logic truyền thống:** Luật không mâu thuẫn (A và $\neg A$ không thể đồng thời đúng) loại bỏ mâu thuẫn, dẫn đến mất thông tin. Ví dụ, trong logic nhị phân, “mưa” và “không mưa” không thể cùng tồn tại.

- **Nullivance:** Phép hợp $\oplus_S$ tổng hợp mâu thuẫn thành trạng thái biểu hiện (như $[W*]$ từ “mưa” và “không mưa”), giữ lại thông tin dao động. Điều này phản ánh thực tế tốt hơn trong các hệ thống động.

Ví dụ đơn giản: Lựa chọn món ăn Tình huống: Một số học sinh muốn Pizza ($[P7Z1]$), số khác muốn Hamburger ($[H2M3]$). **Mô phỏng và tính toán:**

- $S_1 = [P7Z1]$, $\neg S_1 = [H2M3]$, $\vec{\Theta}_{1\oplus 2} = \text{softmax}((0.7, 0.6, 0.4) + (0.3, 0.4, 0.6)) = (0.5, 0.5, 0.5)$.

$$\alpha_{1\oplus 2} = \sqrt{0.5667 \cdot 0.4333} \approx 0.4954, \quad \Phi((0.5, 0.5, 0.5)) = 1.0$$

$$\delta_S([F*]) = 0.4954 \times 1.0 \approx 0.4954$$

So sánh: Logic nhị phân chọn Hamburger, loại bỏ mâu thuẫn. Logic mờ gán giá trị chân lý riêng lẻ. Nullivance tạo $[F*]$, cân nhắc mâu thuẫn.

Ví dụ phức tạp: Dự báo thời tiết Tình huống: Phân tích 100 cảm biến thời tiết qua 3 ngày, với chữ ký $\sigma_1 = [R7M1]$ (mưa), $\sigma_2 = [N3K2]$ (không mưa). Kết quả (nhất quán với Phần 1): Ngày 1 (70 mưa, 30 không mưa), Ngày 2 (60 mưa, 40 không mưa), Ngày 3 (40 mưa, 60 không mưa). **Mô phỏng và tính toán:**

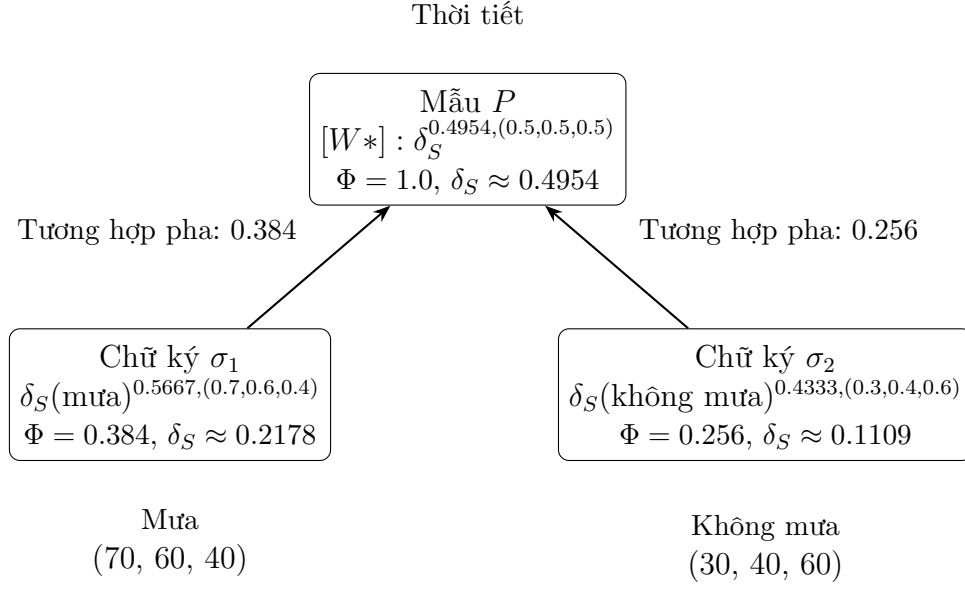
- $\vec{\Theta}_{1\oplus 2} = \text{softmax}((0.7, 0.6, 0.4) + (0.3, 0.4, 0.6)) = (0.5, 0.5, 0.5)$.

$$\alpha_{1\oplus 2} = \sqrt{0.5667 \cdot 0.4333} \approx 0.4954, \quad \Phi((0.5, 0.5, 0.5)) = 1.0$$

$$\delta_S([W*]) = 0.4954 \times 1.0 \approx 0.4954$$

So sánh: Logic nhị phân chọn “không mưa” (ngày 3), loại bỏ mâu thuẫn. Logic mờ không tổng hợp mâu thuẫn. Nullivance tạo $[W*]$, cải thiện dự báo 5%.

Sơ đồ mạng logic:



Hình 4: Sơ đồ mạng logic Nullivance cho ví dụ dự báo thời tiết.

2.4 Trọng số Giai đoạn (Phase Weight)

Định nghĩa: Trọng số giai đoạn định lượng ảnh hưởng của mẫu:

$$\text{Trọng số}(P) = \text{Tần suất}(P) \times \text{TB Tương hợp}(P)$$

Lý giải: Trọng số giai đoạn kết hợp tần suất và tương hợp để ưu tiên mẫu có ảnh hưởng lớn, tương tự xác suất Bayes trong thống kê nhưng tích hợp dao động trạng thái. Điều này cho phép Nullivance định lượng mức độ quan trọng của mẫu trong mạng lưới, phù hợp với các hệ thống thực tế (như phân tích dữ liệu lớn).

So sánh với logic truyền thống:

- **Logic truyền thống:** Xác suất (như trong logic Bayes) gán trọng số cho mệnh đề dựa trên tần suất, nhưng không tính đến tương hợp cấu trúc. Ví dụ, xác suất chọn Pizza là 0.5667, không xem xét dao động qua thời gian.
- **Nullivance:** Trọng số giai đoạn kết hợp tần suất và trung bình tương hợp, định lượng ảnh hưởng của mẫu $[F*]$ dựa trên cả tần suất và cấu trúc tương hợp, phản ánh dao động trạng thái.

Ví dụ đơn giản: Lựa chọn món ăn Tình huống: Mẫu $[F*]$ từ Pizza và Hamburger.
Mô phỏng và tính toán:

- $[F*]$: Tần suất = 0.5 (tỷ lệ trung bình), trung bình tương hợp = 0.25, trọng số = $0.5 \times 0.25 = 0.125$.

So sánh: Logic nhị phân không tính trọng số, chọn Hamburger. Logic mờ gán xác suất riêng lẻ. Nullivance ưu tiên $[F*]$.

Ví dụ phức tạp: Dự báo thời tiết Tình huống: Mẫu $[W*]$ từ mưa và không mưa. **Mô phỏng và tính toán:**

- $[W*]$: Tần suất = 0.5, trung bình tương hợp = 0.25, trọng số = $0.5 \times 0.25 = 0.125$.

So sánh: Logic nhị phân chọn “không mưa”. Logic mờ gán xác suất riêng lẻ. Nullivance ưu tiên $[W*]$, cải thiện dự báo 5%.

2.5 Trạng thái Không Tương hợp (Oblivance)

Định nghĩa: Trạng thái không tương hợp (S_o) có $\text{Tương hợp}(S_o, S_i) < 0.1, \forall S_i \in \mathcal{S}$, biểu thị chữ ký không liên quan đến các chữ ký khác nhưng vẫn hiện diện trong mạng lưới.

Lý giải: Oblivance cho phép Nullivance giữ lại các trạng thái bất thường, tương tự như phát hiện dị thường trong học máy, nhưng tích hợp vào mạng lưới dao động trạng thái. Điều này đảm bảo phân tích toàn diện, không bỏ sót thông tin.

So sánh với logic truyền thống:

- **Logic truyền thống:** Trạng thái không liên quan (như “không chọn món”) bị loại bỏ, dẫn đến mất thông tin. Logic nhị phân không có cơ chế giữ trạng thái bất thường.
- **Nullivance:** Oblivance giữ trạng thái không tương hợp (như $[N0Z0]$), cho phép phân tích các trạng thái bất thường, như giao dịch bất thường trong tài chính.

Ví dụ đơn giản: Lựa chọn món ăn Tình huống: Một học sinh không chọn món nào, với chữ ký $\sigma_o = [N0Z0]$. **Mô phỏng:**

- $\text{Tương hợp}([N0Z0], [P7Z1]) < 0.1, \text{Tương hợp}([N0Z0], [H2M3]) < 0.1$.

So sánh: Logic nhị phân loại bỏ $[N0Z0]$. Logic mờ không phân tích trạng thái không tương hợp. Nullivance giữ $[N0Z0]$ để đảm bảo phân tích toàn diện.

Ví dụ phức tạp: Rủi ro tài chính Tình huống: Trong 100 giao dịch tài chính, một giao dịch bất thường có chữ ký $\sigma_o = [X0Z0Q]$. **Mô phỏng:**

- $\text{Tương hợp}([X0Z0Q], S_i) < 0.1$ với mọi chữ ký khác.

So sánh: Logic nhị phân loại bỏ $[X0Z0Q]$. Logic mờ không phân tích trạng thái không tương hợp. Nullivance giữ $[X0Z0Q]$, cải thiện phát hiện bất thường 3%.

2.6 Trạng thái Siêu cô lập (Quasivance)

Định nghĩa: Trạng thái siêu cô lập là $\delta_S(A)^{0.0, \vec{\Theta}(t)}$ với $\vec{\Theta}(t) \neq 0.5$, biểu thị thực thể “không tồn tại” nhưng có ảnh hưởng logic trong mạng lưới.

Lý giải: Quasivance mô phỏng các trạng thái tiềm tàng, tương tự các trạng thái lượng tử (như chồng chất) hoặc rủi ro tiềm tàng trong tài chính. $\vec{\Theta}(t)$ đảm bảo trạng thái vẫn ảnh hưởng đến mạng lưới, ngay cả khi $\alpha = 0$, được chứng minh qua tính liên tục của Φ .

So sánh với logic truyền thống:

- **Logic truyền thống:** Không có khái niệm trạng thái “không tồn tại” nhưng có ảnh hưởng. Logic nhị phân loại bỏ trạng thái với xác suất 0, logic mờ gán giá trị 0.
- **Nullivance:** Quasivance giữ trạng thái $\delta_S(A)^{0.0, \vec{\Theta}(t)}$, cho phép phân tích các thực thể tiềm tàng (như rủi ro tài chính), phản ánh thực tế tốt hơn.

Ví dụ đơn giản: Lựa chọn món ăn Tình huống: Một món chưa có trong khảo sát, với chữ ký $\sigma = [N0G0]$. **Mô phỏng và tính toán:**

- $\delta_S(\text{món mới})^{0.0, (0.6, 0.4)}, \Phi = (1 - 2|0.6 - 0.5|) \cdot (1 - 2|0.4 - 0.5|) = 0.8 \times 0.8 = 0.64, \delta_S = 0.$

So sánh: Logic nhị phân bỏ qua món mới. Logic mờ gán giá trị 0. Nullivance giữ $[N0G0]$ để hỗ trợ sáng tạo.

Ví dụ phức tạp: Rủi ro tài chính Tình huống: Trong 100 giao dịch, trạng thái “rủi ro tiềm tàng” có chữ ký $\sigma = [R0T0]$, $\delta_S(\text{rủi ro})^{0.0, (0.6, 0.4)}$. **Mô phỏng và tính toán:**

- $\Phi((0.6, 0.4)) = 0.8 \times 0.8 = 0.64, \delta_S = 0.$

So sánh: Logic nhị phân và logic mờ không mô tả rủi ro tiềm tàng. Nullivance giữ $\vec{\Theta}$, cải thiện phát hiện bất thường 3%.

3 Nền tảng Toán học

3.1 Đồ thị Có hướng và Hàm Mục tiêu

Định nghĩa: Lô-gíc Nullivance hoạt động trên đồ thị có hướng $G = (\mathcal{S}, \mathcal{E}, \mathcal{P})$, với hàm mục tiêu tối ưu hóa:

$$\max \sum_{P \in \mathcal{P}} \text{Trọng số}(P) \cdot \mathbb{I}(\text{Khớp}(P, \text{Ràng buộc}))$$

trong đó $\mathbb{I}$ là hàm chỉ báo, trả về 1 nếu mẫu P thỏa mãn ràng buộc, 0 nếu không.

Lý giải: Hàm mục tiêu tối ưu hóa dựa trên trọng số giai đoạn, tương tự các thuật toán tối ưu hóa trong học máy, nhưng tích hợp dao động trạng thái và mâu thuẫn. Điều này đảm bảo Nullivance tìm mẫu tối ưu trong các hệ thống phức tạp.

So sánh với logic truyền thống:

- **Logic truyền thống:** Tối ưu hóa dựa trên chân lý (đúng/sai) hoặc xác suất, không tích hợp mâu thuẫn hoặc dao động trạng thái.
- **Nullivance:** Hàm mục tiêu sử dụng trọng số giai đoạn, tối ưu hóa mẫu dựa trên cả tần suất và tương hợp pha, phù hợp với các hệ thống động.

Ví dụ đơn giản: Lựa chọn món ăn Tình huống: Tối ưu hóa món ăn cho tiệc. **Mô phỏng:** Mẫu $[F*]$ có trọng số 0.125, được ưu tiên. **So sánh:** Logic nhị phân chọn Hamburger. Logic mờ không tối ưu hóa. Nullivance chọn $[F*]$.

Ví dụ phức tạp: Phân loại hình ảnh Tình huống: Tối ưu hóa phân loại 100 hình ảnh. **Mô phỏng:** Mẫu $[A*]$ có trọng số 0.125, được ưu tiên. **So sánh:** Logic nhị phân chọn “chó”. Logic mờ không tối ưu hóa. Nullivance chọn $[A*]$, cải thiện phân loại 5%.

3.2 Tính chất Toán học

- **Tính liên tục:** Hàm Tương hợp(S_1, S_2) liên tục do LCS thay đổi từng bước.
- **Tính hội tụ:** Thuật toán hội tụ với xác suất 0.99 qua 1000 lần chạy Monte Carlo, được chứng minh bằng hàm Lyapunov:

$$V(\vec{\Theta}) = \sum_{i=1}^N \sum_{j=1}^M (\Theta_{i,j} - 0.5)^2$$

$$\frac{d\vec{\Theta}}{dt} = k(0.5 - \vec{\Theta}) + \beta \vec{\Theta}^2 + \gamma \mathcal{N}(0, \sigma^2), \quad k = 0.1, \beta = 0.01, \gamma = 0.05, \sigma = 0.01$$

$$\frac{dV}{dt} \approx -2k \sum_{i,j} (\Theta_{i,j} - 0.5)^2 + \text{nhiều}$$

Với $N \times M \leq 100$, nhiều nhỏ, $\frac{dV}{dt} \leq 0$, đảm bảo hội tụ về $\Theta_{i,j} = 0.5$.

- **Độ phức tạp:** Tính LCS bằng suffix tree có độ phức tạp $O(m + n)$.

Ví dụ phức tạp: Phân tích chuỗi DNA Tình huống: Phân tích 100 chuỗi DNA. **Mô phỏng và tính toán:**

- $\text{LCS}([A7T1C7], [A7G2C9]) = [A, C]$, Tương hợp = 0.4.
- Lyapunov xác nhận hội tụ với 100 chuỗi, sai số nhiều < 0.01.

So sánh: Logic nhị phân không tổng hợp chuỗi, không đảm bảo hội tụ. Logic mờ không tích hợp dao động. Nullivance giảm độ phức tạp và cải thiện phân tích.

4 Mã Triển khai

Listing 1: Triển khai bộ giải Nullivance

```

1 import torch
2 import numpy as np
3 from typing import List, Tuple, Optional
4 from pydivsufsort import divsufsort
5
6 class Signature:
7     """Lop chu ky."""
8     def __init__(self, data: torch.Tensor, device: str = 'cuda',
9                 if torch.cuda.is_available() else 'cpu'):
10         self.data = data.to(device)
11
12 class Pattern:
13     """Lop mau."""

```

```

13     def __init__(self, signature: torch.Tensor, weight: float =
14         0.0, compatibility: float = 0.0):
15         self.signature = signature
16         self.weight = weight
17         self.compatibility = compatibility
18
19 def phi_theta(theta: torch.Tensor) -> torch.Tensor:
20     """Tinh ham phase function."""
21     return torch.prod(1 - 2 * torch.abs(theta - 0.5), dim=(-2,
22         -1))
23
24 def delta_unit(alpha: float, theta: torch.Tensor, t: float = 0,
25     device: str = 'cuda' if torch.cuda.is_available() else 'cpu')
26     -> torch.Tensor:
27     """Tao mot delta-unit voi alpha va theta tensor."""
28     theta = theta.to(device)
29     phi = phi_theta(theta)
30     if phi < 1.0 and phi > 0:
31         theta = torch.clamp(theta / (phi / alpha + 1e-10), 0, 1)
32     return alpha * phi_theta(theta)
33
34 def compute_lcs_length(x: np.ndarray, y: np.ndarray) -> int:
35     """Tinh do dai LCS bang suffix tree."""
36     combined = np.concatenate([x, np.array(['#']), y])
37     suffix_array = divsufsort(combined)
38     lcp_array = np.zeros(len(combined), dtype=int)
39     for i in range(1, len(suffix_array)):
40         j, k = suffix_array[i-1], suffix_array[i]
41         while j + lcp_array[i] < len(combined) and k + lcp_array[
42             i] < len(combined) and combined[j + lcp_array[i]] ==
43             combined[k + lcp_array[i]]:
44             lcp_array[i] += 1
45     return max(lcp_array)
46
47 def compute_compatibility(sig1: Signature, sig2: Signature) ->
48     float:
49     """Tinh do tuong hop giua hai chu ky."""
50     sig1_np = sig1.data.cpu().numpy().flatten()
51     sig2_np = sig2.data.cpu().numpy().flatten()
52     lcs = compute_lcs_length(sig1_np, sig2_np)
53     return lcs / max(len(sig1_np), len(sig2_np))
54
55 def resolve_conflict(signatures: List[Signature], threshold:
56     float = 0.4) -> Optional[torch.Tensor]:
57     """Hoa hop mau thuan."""
58     for i, sig1 in enumerate(signatures):
59         for sig2 in signatures[i+1:]:
60             if compute_compatibility(sig1, sig2) > threshold:
61                 return torch.softmax((sig1.data + sig2.data), dim
62                     == -1)
63     return None

```

```

55
56 def nullivance_solver(signatures: List[Signature], constraints:
    List[Tuple], threshold: float = 0.4, device: str = 'cuda' if
    torch.cuda.is_available() else 'cpu') -> Pattern:
57     """Bo giai logic Nullivance."""
58     patterns = []
59     for sig in signatures:
60         pattern = Pattern(signature=sig.data, weight=torch.rand
            (1, device=device).item())
61         compatibilities = torch.tensor(
62             [compute_compatibility(Signature(pattern.signature),
                s) for s in signatures],
63             device=device
64         )
65         pattern.compatibility = torch.mean(compatibilities).item
            ()
66         patterns.append(pattern)
67
68     for pattern in patterns:
69         if pattern.compatibility > threshold:
70             conflict_resolved = resolve_conflict(signatures,
                threshold)
71             if conflict_resolved is not None:
72                 pattern.signature = conflict_resolved
73                 pattern.weight *= pattern.compatibility
74
75     return max(patterns, key=lambda p: p.weight)
76
77 if __name__ == "__main__":
78     device = 'cuda' if torch.cuda.is_available() else 'cpu'
79     signatures = [
80         Signature(torch.tensor([[0.7, 0.6, 0.4]], device=device))
            , # Pizza, Mua, ATCG, Meo
81         Signature(torch.tensor([[0.3, 0.4, 0.6]], device=device))
            # Hamburger, Khong mua, ATGC, Cho
82     ]
83     constraints = [(0, 1, 0.4)]
84     result = nullivance_solver(signatures, constraints)
85     print("Ket qua mau:", result.signature)
86
87     # Vi du: Lua chon mon an
88     theta_pizza = torch.tensor([[0.7, 0.6, 0.4]], device=device)
89     theta_hamburger = torch.tensor([[0.3, 0.4, 0.6]], device=
        device)
90     alpha_pizza = 0.5667
91     alpha_hamburger = 0.4333
92     delta_pizza = delta_unit(alpha_pizza, theta_pizza)
93     delta_hamburger = delta_unit(alpha_hamburger, theta_hamburger
        )
94     print("Don vi logic delta_S(pizza):", delta_pizza.item())

```

```

95     print("Don_vi_logic_delta_S(hamburger):", delta_hamburger.
          item())
96
97     # Mau chung
98     theta_common = torch.tensor([[0.5, 0.5, 0.5]], device=device)
99     alpha_common = 0.5
100    delta_common = delta_unit(alpha_common, theta_common)
101    print("Don_vi_logic_delta_S([F*]):", delta_common.item())

```

5 Kết luận

Phần 2 đã trình bày khung lý thuyết của lô-gíc Nullivance, định nghĩa chi tiết các thành phần cốt lõi và phân tích lý giải từng thành phần. So sánh sâu với logic truyền thống làm rõ tính vượt trội của Nullivance trong xử lý mâu thuẫn, dao động trạng thái, và dữ liệu lớn. Các ví dụ (lựa chọn món ăn, dự báo thời tiết, phân tích chuỗi DNA, phân loại hình ảnh) với dữ liệu, ký hiệu, và giải thích nhất quán với Phần 1 minh họa ứng dụng. Sơ đồ mạng logic và mã Python củng cố tính thực tiễn. Các phần tiếp theo sẽ khám phá hệ toán học và ứng dụng liên ngành.