



**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN**

KHOÁ LUẬN TỐT NGHIỆP

**Học có giám sát với dữ liệu có phân bố thay
đổi bằng mô hình dựa trên quan hệ nhân quả**

GVHD: ThS. Trần Trung Kiên và TS. Nguyễn Ngọc Thảo

Nhóm sinh viên thực hiện:

20120032 – Phan Trường An

20120061 – Phạm Dương Trường Đức

Tháng 7/2024

Nội dung

1. Giới thiệu bài toán
2. Cơ sở lí thuyết
3. Causally Adaptive Constraint Minimization (CACM)
4. Thực nghiệm
5. Kết luận & hướng phát triển

1. Giới thiệu bài toán

Dữ liệu có phân bố thay đổi (Data distribution shift):

- Dữ liệu mà mô hình phải dự đoán có sự thay đổi phân bố so với dữ liệu được dùng để huấn luyện mô hình.
- Hệ quả: độ chính xác của kết quả dự đoán của mô hình giảm.

Khóa luận này tập trung vào bài toán dữ liệu có phân bố **đầu vào** thay đổi.

Định nghĩa bài toán

Cho dữ liệu của K miền huấn luyện, dữ liệu có dạng $\{(x_i, y_i)_{i=1}^n\}$. Trong đó:

- (x, y) phát sinh từ $P(X, Y) = P(X)P(Y|X)$.
- Các miền này có cùng phân bố $P(Y|X)$ nhưng khác phân bố $P(X)$.
- Các cặp (x, y) có thể có thêm thuộc tính a (địa điểm, thời gian,...).

Yêu cầu: tìm một hàm dự đoán $g: \mathcal{X} \rightarrow \mathcal{Y}$ có kết quả dự đoán tốt trên miền mục tiêu. Miền mục tiêu có cùng phân bố $P(Y|X)$ nhưng khác phân bố $P(X)$ so với các miền huấn luyện.

Khó khăn và thách thức của bài toán

- Dữ liệu thực tế có thể xuất hiện nhiều loại phân bố thay đổi, phân bố thay đổi có thể xảy ra trên nhiều thuộc tính.
- Các thuật toán khái quát miền thường chỉ chống chịu tốt với một loại phân bố thay đổi.
- Các thuật toán khái quát miền thường chỉ giải quyết phân bố thay đổi trên một thuộc tính.

Ý nghĩa

- Giúp mô hình học máy giữ được hiệu suất tốt và đưa ra dự đoán đủ tốt với yêu cầu của người dùng.
- Giúp mở rộng mô hình học máy.

Một số hướng tiếp cận

Empirical Risk Minimization (ERM)

- Xem như dữ liệu huấn luyện và kiểm tra có cùng phân bố với nhau.
- Ước lượng độ lỗi thực nghiệm (empirical risk) bằng cách tính trung bình của hàm lỗi trên dữ liệu huấn luyện.
- Tìm cách tối thiểu độ lỗi trung bình trên dữ liệu huấn luyện.

Correlation Alignment (CORAL) (2016) [1]

- **Căn chỉnh** dữ liệu huấn luyện theo dữ liệu của miền mục tiêu dựa trên hiệp phương sai.
- Dữ liệu huấn luyện sẽ có phân bố giống với dữ liệu của miền mục tiêu.

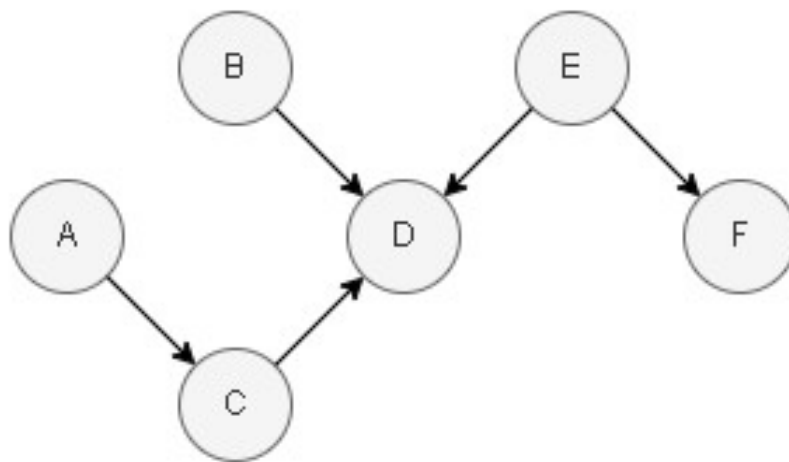
Causally Adaptive Constraint Minimization (CACM) (2022)

- Được đề xuất trong bài báo “Modeling the data-generating process is necessary for out-of-distribution generalization” ở hội nghị ICLR bởi nhóm tác giả của Microsoft Research [2].
- Tận dụng kiến thức về quá trình tạo ra dữ liệu và **đồ thị nhân quả** để tìm ra ràng buộc đúng.
- Vượt trội hơn các thuật toán khái quát miền khác trong nhiều trường hợp.
- Có thể hoạt động tốt trên dữ liệu có phân bố thay đổi **trên nhiều thuộc tính**.
- CACM là phương pháp chính được tìm hiểu sâu trong khóa luận này.

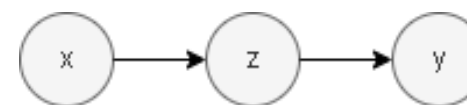
Nội dung

1. Giới thiệu bài toán
2. Cơ sở lí thuyết
3. Causally Adaptive Constraint Minimization (CACM)
4. Thực nghiệm
5. Kết luận & hướng phát triển

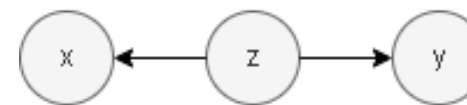
Đồ thị nhân quả



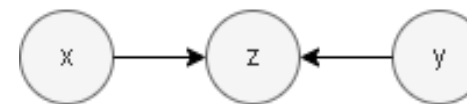
a)



Chain



Fork



Collider

b)

Hình 1: Ví dụ về đồ thị nhân quả và các kiến trúc đồ thị trong một đồ thị nhân quả

D-separation [3]

Nếu G là một đồ thị có hướng trong đó X, Y và S là các tập hợp đỉnh rời rạc. Một đường đi vô hướng p giữa X và Y được xem là bị chặn (d-separated) bởi S khi và chỉ khi có ít nhất một trong hai điều kiện sau thoả mãn:

1. p chứa một chain $x \rightarrow z \rightarrow y$ hoặc một fork $x \leftarrow z \rightarrow y$ sao cho nút nằm giữa z thuộc S .
2. p chứa một collider $x \rightarrow z \leftarrow y$ sao cho nút nằm giữa z và “con cháu” của z không thuộc S .

Nếu S chặn mọi đường đi từ một nút trong X đến một nút trong Y thì X và Y d-separated bởi S nghĩa là X và Y độc lập có điều kiện trên S ($X \perp\!\!\!\perp Y \mid S$).

Nội dung

1. Giới thiệu bài toán
2. Cơ sở lí thuyết
3. Causally Adaptive Constraint Minimization (CACM)
4. Thực nghiệm
5. Kết luận & hướng phát triển

3. Causally Adaptive Constraint Minimization (CACM) (2022)

Thuật toán *CACM*

Đầu vào: Bộ dữ liệu $(x_i, a_i, y_i)_{i=1}^n$, đồ thị nhân quả *DAG*

Đầu ra: Hàm dự đoán $g(x): \mathcal{X} \rightarrow \mathcal{Y}$

Pha 1: Rút ra các ràng buộc độc lập chính xác từ đồ thị nhân quả

Pha 2: Áp dụng Regularization penalty sử dụng các ràng buộc đã rút ra

3. Causally Adaptive Constraint Minimization (CACM) (2022)

Thuật toán *CACM*

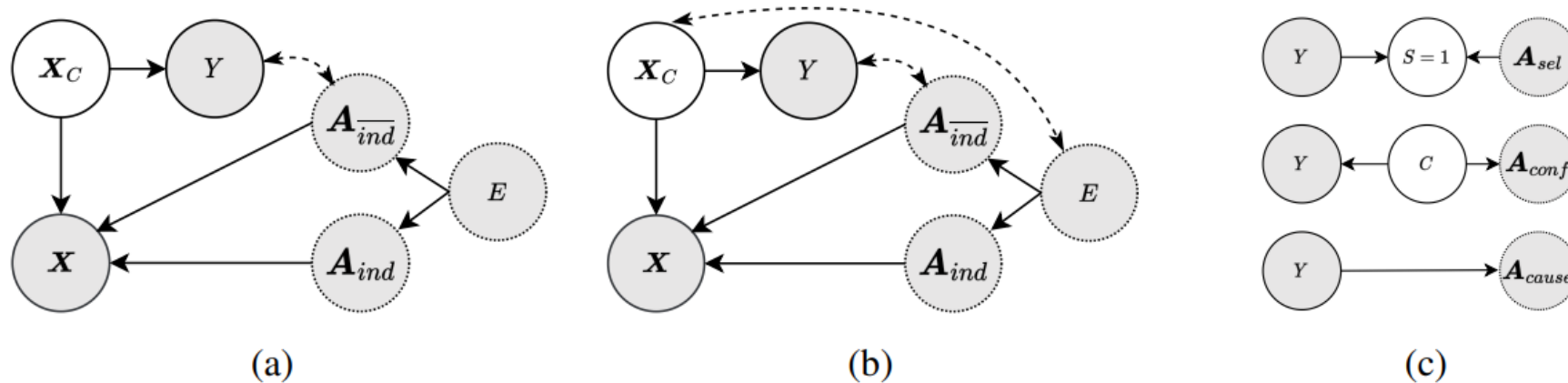
Đầu vào: Bộ dữ liệu $(x_i, a_i, y_i)_{i=1}^n$, đồ thị nhân quả *DAG*

Đầu ra: Hàm dự đoán $g(x): \mathcal{X} \rightarrow \mathcal{Y}$

Pha 1: Rút ra các ràng buộc độc lập chính xác từ đồ thị nhân quả

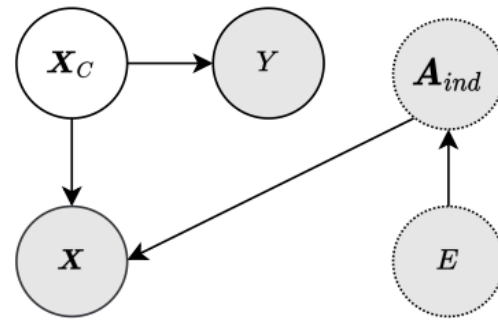
Pha 2: Áp dụng Regularization penalty sử dụng các ràng buộc đã rút ra

3. Causally Adaptive Constraint Minimization (CACM) (2022)

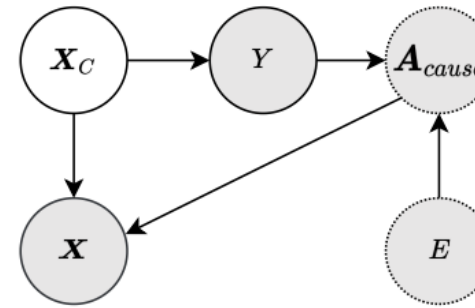


Hình 2: Canonical causal graph

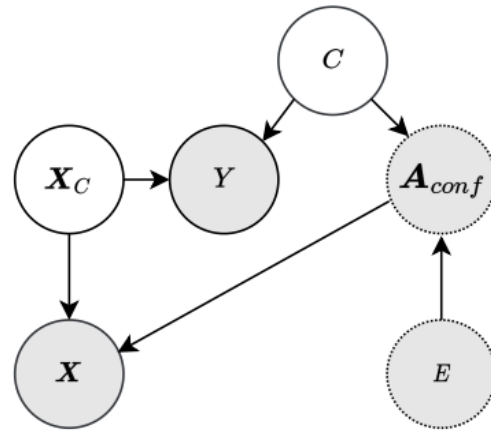
Nguồn: Modeling the Data-Generating Process is Necessary for Out-of-Distribution Generalization



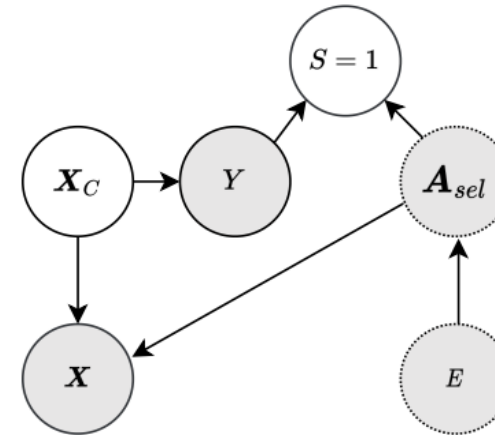
(a) *Independent shift*



(b) *Causal shift*



(c) *Confounded shift*



(d) *Selected shift*

Hình 3: Đồ thị nhận dạng nhân quả

Nguồn: Modeling the Data-Generating Process is Necessary for Out-of-Distribution Generalization

Quan hệ Causal và Independent

Hình 4: Ví dụ về quan hệ Causal và Independent

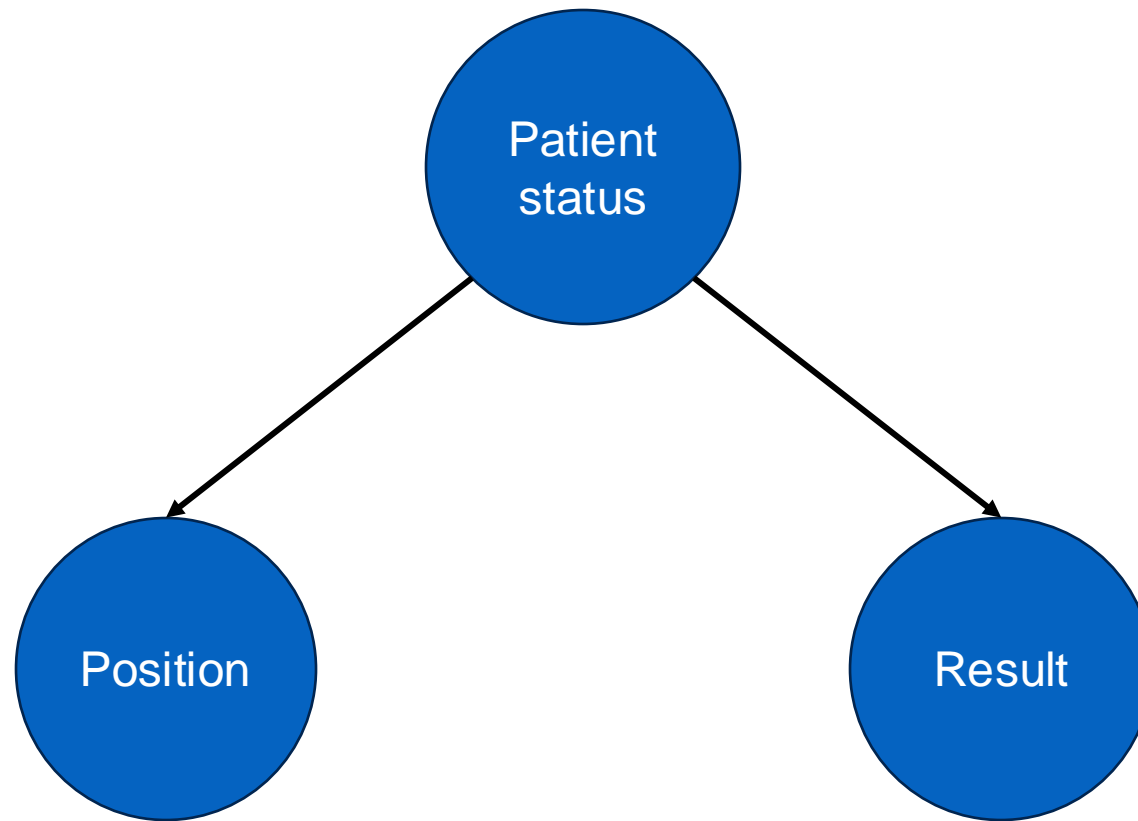
	Train				Test	
	0.9		0.8		0.1	
Y=0						
Y=1						

Colored MNIST: $A_{cause} = color$
(*color* bị ảnh hưởng bởi nhãn Y)

	Train				Test	
	15°		60°		90°	
Y=0						
Y=1						

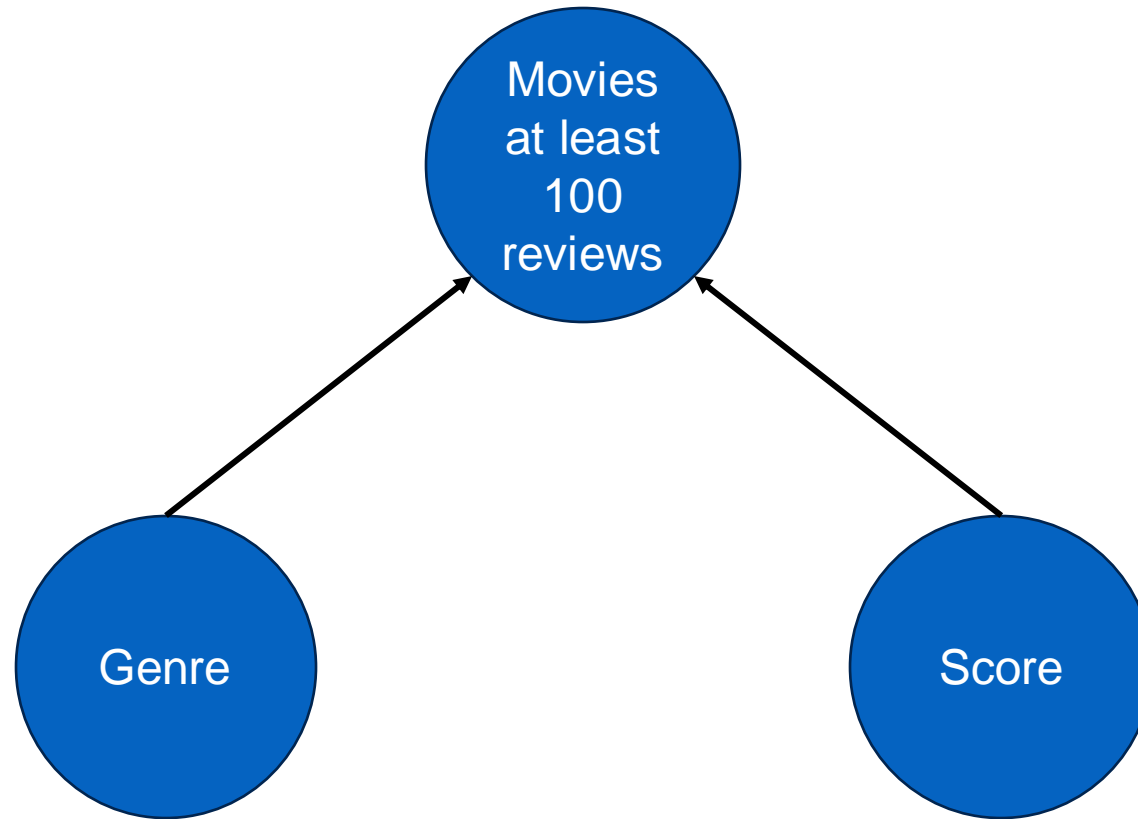
Rotated MNIST: $A_{ind} = rotate$
(*rotate* độc lập với nhãn Y)

Quan hệ Confounded



Hình 5: Ví dụ về quan hệ Confounded

Quan hệ Selected



Hình 6: Ví dụ về quan hệ Selected

3. Causally Adaptive Constraint Minimization (CACM) (2022)

Thuật toán **CACM**

Đầu vào: Bộ dữ liệu $(x_i, a_i, y_i)_{i=1}^n$, đồ thị nhân quả DAG

Đầu ra: Hàm dự đoán $g(x): \mathcal{X} \rightarrow \mathcal{Y}$

Pha 1: Rút ra các ràng buộc độc lập chính xác từ đồ thị nhân quả

Pha 2: Áp dụng Regularization penalty sử dụng các ràng buộc đã rút ra

3. Causally Adaptive Constraint Minimization (CACM) (2022)

Thuật toán CACM

Pha 1: Rút ra các ràng buộc độc lập chính xác từ đồ thị nhân quả

Xét các thuộc tính $A \in \mathcal{A}$:

Nếu X_c và A d-seperated **thì**

$X_c \perp\!\!\!\perp A$ là ràng buộc độc lập hợp lệ

hoặc nếu X_c và A d-seperated điều kiện A_s^* **thì**

$X_c \perp\!\!\!\perp A | A_s$ là ràng buộc độc lập hợp lệ

* (A_s là tập con bất kì của $\mathcal{A} \setminus \{A\} \cup \{Y\}$)

Các điều kiện độc lập có ràng buộc thoả mãn bởi X_c là:

1. *Independent*: $X_c \perp\!\!\!\perp A_{ind}; X_c \perp\!\!\!\perp E; X_c \perp\!\!\!\perp A_{ind}|Y; X_c \perp\!\!\!\perp A_{ind}|E; X_c \perp\!\!\!\perp A_{ind}|Y, E$
2. *Causal*: $X_c \perp\!\!\!\perp A_{caus}|Y; X_c \perp\!\!\!\perp E; X_c \perp\!\!\!\perp A_{caus}|Y, E$
3. *Confounded*: $X_c \perp\!\!\!\perp A_{conf}; X_c \perp\!\!\!\perp E; X_c \perp\!\!\!\perp A_{conf}|E$
4. *Selected*: $X_c \perp\!\!\!\perp A_{sel}|Y; X_c \perp\!\!\!\perp A_{sel}|Y, E$

3. Causally Adaptive Constraint Minimization (CACM) (2022)

Thuật toán **CACM**

Đầu vào: Bộ dữ liệu $(x_i, a_i, y_i)_{i=1}^n$, đồ thị nhân quả DAG

Đầu ra: Hàm dự đoán $g(x): \mathcal{X} \rightarrow \mathcal{Y}$

Pha 1: Rút ra các ràng buộc độc lập chính xác từ đồ thị nhân quả

Pha 2: Áp dụng **Regularization penalty** sử dụng các ràng buộc đã rút ra

3. Causally Adaptive Constraint Minimization (CACM) (2022)

Thuật toán CACM

Pha 2: Áp dụng Regularization penalty sử dụng các ràng buộc đã rút ra

Xét các thuộc tính $A \in \mathcal{A}$:

Nếu $X_c \perp\!\!\!\perp A$ thì

$$RegPenalty_A = \sum_{|E|} \sum_{i=1}^{|A|} \sum_{j>i} MMD \left(P(\phi(x)|A_i), P(\phi(x)|A_j) \right)$$

hoặc nếu $X_c \perp\!\!\!\perp A|A_s$ thì

$$RegPenalty_A = \sum_{|E|} \sum_{a \in A_s} \sum_{i=1}^{|A|} \sum_{j>i} MMD \left(P(\phi(x)|A_i, a), P(\phi(x)|A_j, a) \right)$$













$$RegPenalty = \sum_{A \in \mathcal{A}} \lambda_A RegPenalty_A$$

$$g_1, \phi = \operatorname{argmin}_{g_1, \phi} L(g_1(\phi(x)), y) + RegPenalty$$













Nội dung

1. Giới thiệu bài toán
2. Cơ sở lí thuyết
3. Causally Adaptive Constraint Minimization (CACM)
4. Thực nghiệm
5. Kết luận & hướng phát triển

Bộ dữ liệu MNIST

	Train				Test	
	0.9		0.8		0.1	
Y=0						
Y=1						

Hình 7: Colored MNIST

	Train				Test	
	15°		60°		90°	
Y=0						
Y=1						

Hình 8: Rotated MNIST

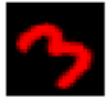
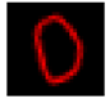

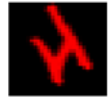





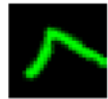


2 lớp:

- 0: số bé hơn 5
- 1: số lớn hơn hoặc bằng 5

Colored MNIST: $A_{cause} = color$
(thuộc tính *color* bị ảnh hưởng bởi nhãn)

Rotated MNIST: $A_{ind} = rotate$
(thuộc tính *rotate* độc lập với nhãn)

Bộ dữ liệu MNIST

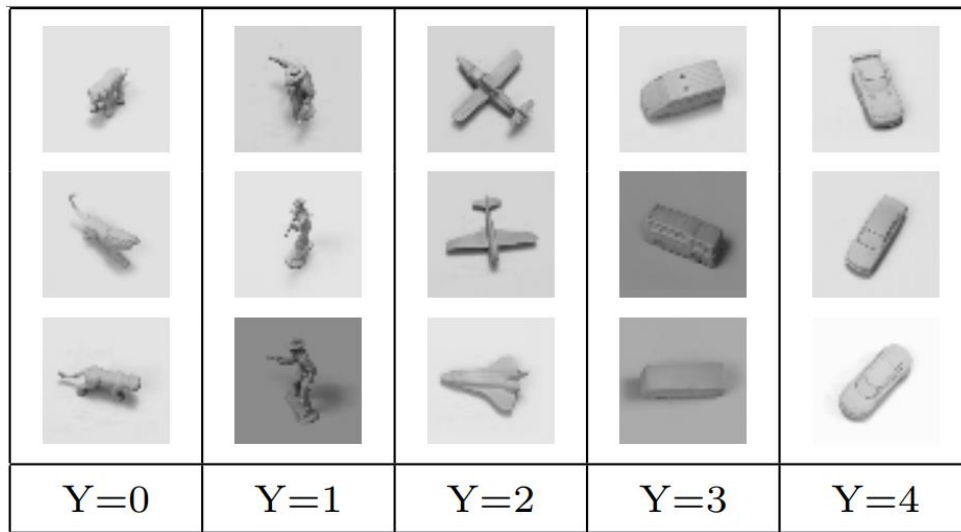
	Train				Test	
	(0.9, 15°)		(0.8, 60°)		(0.1, 90°)	
Y=0						
Y=1						

Hình 9: Một số mẫu dữ liệu
Colored+Rotated MNIST

Colored+Rotated MNIST: thay đổi phân bố đa thuộc tính:

- $A_{cause} = color$
- $A_{ind} = rotate$

Bộ dữ liệu small NORB



Hình 10: Một số mẫu dữ liệu **small NORB**

5 lớp: động vật bốn chân, nhân vật, máy bay, xe tải, xe ô tô

Lighting small NORB: $A_{cause} = lighting$
(*lighting* bị ảnh hưởng bởi nhãn)

Azimuth small NORB: $A_{ind} = azimuth$
(*azimuth* độc lập với nhãn)

Lighting+Azimuth small NORB: thay đổi phân bố đa thuộc tính

- $A_{cause} = lighting$
- $A_{ind} = azimuth$

So sánh kết quả thí nghiệm với bài báo gốc

Algo	MNIST			small NORB		
	<i>color</i>	Accuracy <i>rotation</i>	<i>col+rot</i>	<i>lighting</i>	Accuracy <i>azimuth</i>	<i>light+azi</i>
Các kết quả trong bài báo gốc						
ERM	30.9 \pm 1.6	61.9 \pm 0.5	25.2 \pm 1.3	65.5 \pm 0.7	78.6 \pm 0.7	64.0 \pm 1.2
CORAL	28.5 \pm 0.8	62.5 \pm 0.7	23.5 \pm 1.1	64.7 \pm 0.5	77.2 \pm 0.7	62.9 \pm 0.3
CACM	70.4 \pm 0.5	62.4 \pm 0.4	54.1 \pm 1.3	85.4 \pm 0.5	80.5 \pm 0.6	69.6 \pm 1.6
Các kết quả từ thí nghiệm của khóa luận						
ERM	30.8 \pm 0.6	61.6 \pm 0.2	25.5 \pm 1.0	70.8 \pm 1.7	77.6 \pm 0.7	64.1 \pm 4.3
CORAL	30.7 \pm 0.3	61.3 \pm 0.9	24.9 \pm 1.6	72.2 \pm 2.5	77.8 \pm 1.3	68.0 \pm 3.2
CACM	72.0 \pm 0.7	61.9 \pm 0.6	49.5 \pm 0.3	86.8 \pm 3.4	77.5 \pm 1.8	76.7 \pm 4.7

Bảng 1: Kết quả thí nghiệm trên các bộ dữ liệu MNIST và small NORB.
Các kết quả ở đây là độ chính xác dự đoán (%) trên tập kiểm tra.

So sánh CACM ràng buộc đúng với CACM ràng buộc sai

Algo	MNIST Accuracy		
	<i>color</i>	<i>rotation</i>	<i>col+rot</i>
CACM ràng buộc đúng	72.0 \pm0.7	61.9 \pm0.6	49.5 \pm0.3
CACM ràng buộc sai	67.2 \pm 3.4	61.6 \pm 0.4	27.9 \pm 1.3

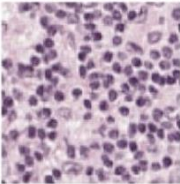
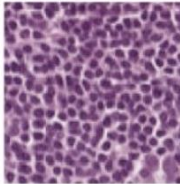
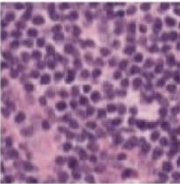
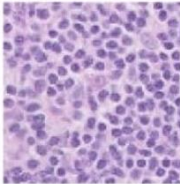
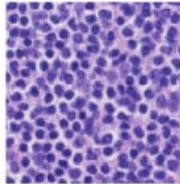
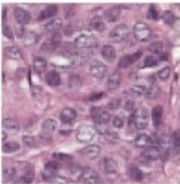
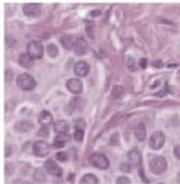
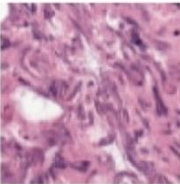
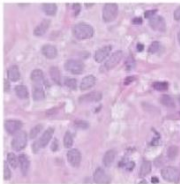
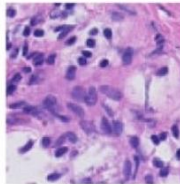
Bảng 2: Kết quả thí nghiệm CACM trên các bộ dữ liệu MNIST với ràng buộc sai.
Các kết quả ở đây là độ chính xác dự đoán (%) trên tập kiểm tra.

So sánh CACM ràng buộc đúng với CACM ràng buộc sai

Algo	small NORB		
	Accuracy		
	<i>lighting</i>	<i>azimuth</i>	<i>light+azi</i>
CACM ràng buộc đúng	86.8 ±3.4	77.5 ±1.8	76.7 ±4.7
CACM ràng buộc sai	85.3 ±3.4	78.1 ±1.4	66.1 ±5.0

Bảng 3: Kết quả thí nghiệm CACM trên các bộ dữ liệu small NORB với ràng buộc sai. Các kết quả ở đây là độ chính xác dự đoán (%) trên tập kiểm tra.

Bộ dữ liệu Camelyon17

Train				Val (OOD)	Test (OOD)
	d = Hospital 1	d = Hospital 2	d = Hospital 3	d = Hospital 4	d = Hospital 5
y = Normal					
y = Tumor					

Hình 11: Một số mẫu dữ liệu **Camelyon17**

2 lớp:

- 0: không có khối u
- 1: có khối u

Có sự khác biệt trong quá trình **nhuộm tiêu bản** và **thu thập dữ liệu** giữa các bệnh viện.

$$\Rightarrow A_{ind} = hospital$$

Kết quả thí nghiệm trên bộ dữ liệu Camelyon17

Algo	Camelyon17		
	Accuracy		
	Validation (IID)	Validation (OOD)	Test (OOD)
ERM	92.0 \pm 0.3	91.2 \pm 1.5	81.5 \pm 2.6
CORAL	85.3 \pm 8.1	71.0 \pm 18.1	71.8 \pm 5.7
CACM	92.2 \pm 0.9	93.3 \pm 0.5	83.2 \pm2.6

Bảng 4: Kết quả thí nghiệm trên bộ dữ liệu Camelyon17.
Các kết quả ở đây là độ chính xác dự đoán (%).

Nội dung

1. Giới thiệu bài toán
2. Cơ sở lí thuyết
3. Causally Adaptive Constraint Minimization (CACM)
4. Thực nghiệm
5. Kết luận & hướng phát triển

Kết luận

Kết quả đạt được:

- Cài đặt thành công CACM.
- Kết quả của khóa luận tương quan với bài báo gốc.
- Thí nghiệm đánh giá hiệu suất của CACM với ràng buộc sai.
- Thí nghiệm CACM với dữ liệu ảnh thực tế.

Kết luận

Kết luận:

- CACM vượt trội hơn ERM và CORAL trong đa số trường hợp.
- CACM hoạt động tốt trên dữ liệu có phân bố thay đổi trên nhiều thuộc tính do áp dụng ràng buộc riêng cho từng loại thay đổi.
- Ràng buộc sai làm giảm đáng kể hiệu suất của CACM.
- CACM hoạt động tốt trên dữ liệu ảnh thực tế.

Hướng phát triển

- Thí nghiệm CACM với loại dữ liệu khác (âm thanh, văn bản).
- Thí nghiệm để đánh giá điểm mạnh, điểm yếu của CACM.

Tài liệu tham khảo

- [1] B. Sun, J. Feng, and K. Saenko, “Return of frustratingly easy domain adaptation,” in Proceedings of the AAAI conference on artificial intelligence, 2016.
- [2] J. N. Kaur, E. Kiciman, and A. Sharma, “Modeling the data-generating process is necessary for out-of-distribution generalization,” International Conference on Learning Representations, 2022.
- [3] Hayduk, Leslie et al. “Pearl’s D-separation: One more step into causal thinking”. In: Structural Equation Modeling 10.2 (2003), pp. 289–311.