

**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN
KHOA KHOA HỌC MÁY TÍNH**

NGUYỄN TRƯỜNG THỊNH

KHÓA LUẬN TỐT NGHIỆP

**CẢI TIẾN CÁC PHƯƠNG PHÁP DATA
VALUATION VÀ ỨNG DỤNG**

**ENHANCING DATA VALUATION
METHODS AND APPLICATIONS**

CỬ NHÂN NGÀNH KHOA HỌC MÁY TÍNH

TP. HỒ CHÍ MINH, NĂM 2025

**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN
KHOA KHOA HỌC MÁY TÍNH**

NGUYỄN TRƯỜNG THỊNH - 21520110

KHÓA LUẬN TỐT NGHIỆP

**CẢI TIẾN CÁC PHƯƠNG PHÁP DATA
VALUATION VÀ ỨNG DỤNG**

**ENHANCING DATA VALUATION
METHODS AND APPLICATIONS**

CỦ NHÂN NGÀNH KHOA HỌC MÁY TÍNH

**GIẢNG VIÊN HƯỚNG DẪN
TS. VÕ NGUYỄN LÊ DUY**

TP. HỒ CHÍ MINH, NĂM 2025

Thông tin hội đồng chấm khóa luận tốt nghiệp

Hội đồng chấm khóa luận tốt nghiệp, thành lập theo quyết định số 24/QĐ-ĐHCNTT ngày 09 tháng 01 năm 2025 của Hiệu trưởng Trường Đại học Công nghệ Thông tin.

- | | |
|------------------------------|----------|
| 1. TS. Dương Việt Hằng | Chủ tịch |
| 2. TS. Nguyễn Duy Khánh | Ủy viên |
| 3. ThS. Nguyễn Thị Ngọc Diễm | Thư ký |

ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC

CÔNG NGHỆ THÔNG TIN

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM

Độc Lập - Tự Do - Hạnh Phúc

TP. HCM, ngày...10..tháng...1..năm...2024.....

NHẬN XÉT KHÓA LUẬN TỐT NGHIỆP

(CỦA CÁN BỘ HƯỚNG DẪN/PHẢN BIỆN)

Tên khóa luận:

CẢI TIẾN CÁC PHƯƠNG PHÁP DATA VALUATION VÀ ÚNG DỤNG

Nhóm SV thực hiện:

NGUYỄN TRƯỜNG THỊNH

Cán bộ hướng dẫn/phản biện:

21520110 TS. VÕ NGUYỄN LÊ DUY

Đánh giá Khóa luận

1. Về cuốn báo cáo:

Số trang _____
Số bảng số liệu _____
Số tài liệu tham khảo _____

Số chương _____
Số hình vẽ _____
Sản phẩm _____

Một số nhận xét về hình thức cuốn báo cáo:

<nhận xét về định dạng, cách thức viết báo cáo, phân bố nội dung, chương mục có hợp lý không..>

2. Về nội dung nghiên cứu:

<nhận xét về kiến thức, phương pháp mà sinh viên đã tìm hiểu, nghiên cứu nhận xét ưu điểm và hạn chế>

.....
.....
.....
.....
.....
.....
.....

3. Về chương trình ứng dụng:

<nhận xét về việc xây dựng ứng dụng demo, nhận xét ưu điểm và hạn chế>

.....
.....
.....
.....

4. Về thái độ làm việc của sinh viên:

<nhận xét về thái độ, ưu khuyết điểm của từng sinh viên tham gia>

.....
.....
.....
.....

Đánh giá chung: Khóa luận đạt/không đạt yêu cầu của một khóa luận tốt nghiệp kỹ sư/ cử nhân, xếp loại Giỏi/ Khá/ Trung bình

.....
.....
.....

Điểm từng sinh viên:

<Tên sinh viên 1>:/10

ĐỀ CƯƠNG CHI TIẾT

TÊN ĐỀ TÀI TIẾNG VIỆT: CẢI TIẾN CÁC PHƯƠNG PHÁP DATA VALUATION VÀ ỨNG DỤNG

TÊN ĐỀ TÀI TIẾNG ANH: ENHANCING DATA VALUATION METHODS AND APPLICATIONS

Cán bộ hướng dẫn: TS. Võ Nguyễn Lê Duy, trường Đại học Công nghệ Thông tin

Thời gian thực hiện: Từ ngày 02/09/2024 đến ngày 28/12/2024

Sinh viên thực hiện:

Nguyễn Trường Thịnh - 21520110

Hệ đào tạo: Tài năng

Nội dung đề tài: (Mô tả chi tiết về tổng quan đề tài, mục tiêu, phạm vi, đối tượng, phương pháp thực hiện, kết quả mong đợi của đề tài, ...)

Tổng quan đề tài (sinh viên cần nêu rõ các đề tài, sản phẩm liên quan đã được nghiên cứu hoặc đã có trên thị trường trước thời điểm hiện tại và nêu thực trạng của chúng, để từ đó nêu bật được lý do thực hiện nghiên cứu trong KLTN này):

Trong bối cảnh dữ liệu trở thành yếu tố quan trọng trong công nghệ và kinh tế, có nhiều thách thức trong việc định lượng giá trị dữ liệu trong các dự đoán và quyết định thuật toán. Đã có nhiều nghiên cứu về phương pháp định giá dữ liệu công bằng nhưng hướng này khá mới nên vẫn còn nhiều hạn chế và chưa được ứng dụng rộng rãi.

Một số sản phẩm trước đó:

1. Leave-one-out:

Leave-one-out là một phương pháp đơn giản để đo lường tầm quan trọng của một điểm dữ liệu bằng cách loại bỏ nó ra khỏi tập dữ liệu và kiểm tra tác động của việc loại bỏ đó đối với hiệu suất của mô hình.

Ưu điểm: Phương pháp trực quan và đơn giản.

Nhược điểm:

- Tốn thời gian tính toán
- Không phù hợp với dữ liệu lớn

2. Data Shapley:

Data Shapley là một phương pháp phức tạp hơn so với Leave-one-out, dựa trên lý thuyết trò chơi. Nó được sử dụng để định giá dữ liệu dựa trên ý tưởng về **Shapley values**, trong

đó giá trị của mỗi điểm dữ liệu được xác định bằng cách xem xét tất cả các tổ hợp con mà nó tham gia.

Ưu điểm:

- **Công bằng:** Shapley values là một phương pháp chính thống từ lý thuyết trò chơi, đảm bảo sự đóng góp của từng điểm dữ liệu được đo lường công bằng.
- **Ứng dụng linh hoạt:** Data Shapley có thể được áp dụng cho nhiều mô hình học máy khác nhau và cho nhiều loại dữ liệu.

Nhược điểm:

- **Tốn kém thời gian và tài nguyên:** độ phức tạp thuật toán là hàm mũ 2.
- **Phụ thuộc vào mô hình và phân phối dữ liệu.**

Các biến thể:

- **Class-wise Shapley:** Xem xét các lớp (class) khác nhau và tính toán giá trị Shapley cho từng lớp riêng biệt, giúp tối ưu hóa khi có các lớp dữ liệu không đồng đều.
- **Beta Shapley** là một biến thể của thuật toán **Shapley values** nhằm đối phó với sự **không chắc chắn** trong dữ liệu, đặc biệt trong trường hợp dữ liệu nhiễu hoặc không rõ ràng. Beta Shapley sử dụng **phân phối Beta** để điều chỉnh giá trị Shapley bằng cách áp dụng các tham số α và β nhằm phản ánh sự không chắc chắn hoặc tín hiệu nhiễu trong dữ liệu. **Tăng tính linh hoạt:** Bằng cách thay đổi các tham số α và β , Beta Shapley có thể điều chỉnh sự quan trọng của dữ liệu tùy thuộc vào cấu trúc của dữ liệu và sự nhiễu.
- **KNN Shapley:** Dựa trên thuật toán **K-Nearest Neighbors (KNN)** để ước lượng giá trị Shapley một cách nhanh chóng bằng cách chỉ xem xét các điểm dữ liệu gần nhất, giúp giảm thời gian tính toán.

LAVA (Learning Algorithm without Pre-specified Algorithms):

LAVA là một phương pháp mới sử dụng **Optimal Transport (OT)** để định giá dữ liệu mà không yêu cầu thuật toán học cụ thể. LAVA dựa trên việc đo lường sự khác biệt giữa các phân phối dữ liệu bằng cách sử dụng **Wasserstein distance**. Wasserstein distance là một loại khoảng cách OT được sử dụng để đo sự khác biệt giữa các phân phối dữ liệu.

Ưu điểm:

- **Tính toán nhanh hơn:** LAVA không yêu cầu huấn luyện lại mô hình nhiều lần mà chỉ dựa vào việc tính toán khoảng cách giữa các phân phối dữ liệu.
- **Không phụ thuộc vào mô hình cụ thể:** LAVA không yêu cầu một thuật toán học cụ thể, điều này giúp phương pháp này linh hoạt hơn trong nhiều ngữ cảnh khác nhau.
- **Giảm chi phí tính toán so với Data Shapley**

Nhược điểm:

- **Độ chính xác phụ thuộc vào OT:** LAVA sử dụng xấp xỉ **Optimal Transport**, do đó có thể không đạt được độ chính xác cao như Data Shapley khi đánh giá sự đóng góp của từng điểm dữ liệu.
- **Chưa được ứng dụng rộng rãi:** LAVA vẫn là một phương pháp tương đối mới và chưa được triển khai rộng rãi trong thực tế.

Như vậy chưa có một phương pháp nào thật sự ưu việt cho phạm vi bài toán này. Qua thực nghiệm sơ bộ em thấy nhiều điểm yếu của chúng có thể bù trừ lẫn nhau, có thể kết hợp hoặc tùy mục đích sử dụng khác nhau mà dùng thuật toán khác nhau. Dữ liệu là một phần quan trọng và cũng là vấn đề nhức nhối với nền AI của Việt Nam khi luôn gặp khó khăn là bộ dữ liệu kém chất lượng. Nên em muốn nghiên cứu đề tài này kết hợp với kiến thức của một nhà khoa học dữ liệu để tạo ra sản phẩm hoàn chỉnh.

Mục tiêu của đề tài (Nêu cụ thể mục tiêu của KLTN, đặc biệt phải nêu được mục tiêu cài tiến sẽ là gì so với thực trạng nêu trong phần tổng quan đề tài):

Mục tiêu là đi cài tiến thời gian xử lý và bộ nhớ (tránh trường hợp tràn bộ nhớ khi tính toán dữ liệu lớn) của các phương pháp, nhưng độ chính xác đánh đổi là tối thiểu.

Xem xét các thuật toán trong các bối cảnh khác nhau đặc biệt là trên bộ dữ liệu mất cân bằng.

Phương pháp thực hiện (Nêu tổng quan phương pháp thực hiện):

□ **Tối ưu thời gian cho Data Shapley:**

- Thay vì tính toán tất cả các tổ hợp con của dữ liệu, nghiên cứu sẽ sử dụng phương pháp chọn ngẫu nhiên một số lượng nhỏ các tập con để tạo ra ước lượng gần đúng cho Shapley values.
- Xem xét các biến thể của Shapley như **Class-wise Shapley**, để gán trọng số khác nhau cho các lớp dữ liệu, giúp cải thiện độ chính xác mà không cần tăng thời gian tính toán.

□ **Tối ưu hóa phương pháp LAVA:**

- Sử dụng **hierarchical Wasserstein distance** để đánh giá khoảng cách giữa các điểm dữ liệu ở hai tập khác nhau, từ đó định giá từng điểm dữ liệu trong tập huấn luyện.
- Để tối ưu hóa, nghiên cứu sẽ chia nhỏ dữ liệu và thực thi LAVA trên từng batch nhỏ, từ đó giảm chi phí bộ nhớ và thời gian.

□ **Phương pháp đánh giá:**

- Các thí nghiệm sẽ được thực hiện trên các bộ dữ liệu phổ biến như **MNIST** và **CIFAR**, **các loại dữ liệu tạo sinh**. Các giá trị Shapley và LAVA của từng điểm dữ liệu sẽ được tính toán để so sánh tầm quan trọng của các điểm dữ liệu.

- Kết quả được đánh giá dựa trên việc loại bỏ các điểm dữ liệu có giá trị đóng góp thấp và so sánh hiệu suất của mô hình khi xóa hoặc giữ lại các điểm dữ liệu đó. Áp dụng **Metric** chuyên biệt và vẽ biểu đồ so sánh sẽ được sử dụng để đánh giá hiệu suất.

Các nội dung chính và giới hạn của đề tài:

Nội dung nghiên cứu tập trung vào kiểm định và so sánh phương pháp định giá dữ liệu được đề xuất với các phương pháp trước đây (như Data Shapley, LAVA, v.v.) trên nhiều khía cạnh, bao gồm cả hiệu suất tính toán và khả năng cải thiện chất lượng dữ liệu. Nghiên cứu sẽ được triển khai qua các bước chính như sau:

a. Kiểm định trên các bộ dữ liệu chuẩn:

- **Bộ dữ liệu:** Các bộ dữ liệu phổ biến trong học máy như **MNIST**, **CIFAR**,... sẽ được sử dụng để kiểm định. Các tập dữ liệu sẽ được chọn ngẫu nhiên nhằm đảm bảo sự đa dạng và tính khách quan trong đánh giá phương pháp.
- **Mục tiêu:** Đánh giá giá trị đóng góp của từng điểm dữ liệu vào mô hình, từ đó xác định độ quan trọng của các điểm dữ liệu trong tập huấn luyện và kiểm tra. Điều này sẽ giúp làm rõ vai trò của từng điểm dữ liệu trong việc cải thiện hoặc làm giảm hiệu suất của mô hình học máy.

b. Đánh giá hiệu suất và loại bỏ dữ liệu kém chất lượng:

- **Kết quả dự kiến:** Sau khi thực hiện mô hình, mỗi điểm dữ liệu sẽ được gán một **giá trị đóng góp** thể hiện độ mạnh/yếu của nó đối với hiệu suất của mô hình. Giá trị này sẽ giúp phân loại các điểm dữ liệu quan trọng và không quan trọng trong tập huấn luyện.
- **Metric đánh giá:** Các thước đo chuyên biệt sẽ được áp dụng để đánh giá hiệu suất của mô hình khi **xóa bỏ các điểm dữ liệu từ cao xuống thấp** (theo giá trị đóng góp). Ngoài ra, các biểu đồ sẽ được vẽ để so sánh hiệu suất khi loại bỏ hoặc giữ lại các điểm dữ liệu quan trọng.
 - Ví dụ: Nếu các điểm có giá trị đóng góp thấp bị loại bỏ, mô hình có thể được cải thiện về mặt tốc độ xử lý và chất lượng dữ liệu. Ngược lại, loại bỏ các điểm có giá trị đóng góp cao có thể làm giảm hiệu suất dự đoán của mô hình.
- **Lợi ích:** Giúp xác định và loại bỏ các điểm dữ liệu nhiễu, kém chất lượng khỏi tập huấn luyện, từ đó tăng độ chính xác của mô hình và giảm chi phí tính toán. Những điểm dữ liệu bị nhận diện là **nhiễu** sẽ được xuất ra và đánh giá lại qua các hình ảnh minh họa hoặc phân tích chi tiết.

c. Xây dựng hệ thống demo ứng dụng:

- **Mục tiêu của demo:** Xây dựng một sản phẩm ứng dụng trực quan giúp người dùng dễ dàng đánh giá và định giá dữ liệu.
- **Hệ thống demo:**
 - Hệ thống sẽ nhận **input là một bộ dữ liệu** từ nguồn trên mạng hoặc do người dùng tự thu thập.
 - Tập huấn luyện (train) ban đầu có thể chưa được chuẩn hóa, trong khi tập kiểm tra (test) sẽ là tập dữ liệu đã được đánh giá là chất lượng chuẩn.
- **Chức năng:**
 - Hệ thống cho phép người dùng **nhập thuật toán định giá dữ liệu**, các thông tin tham số (như kích thước batch, loại mô hình, số lớp), và chọn phương pháp định giá dữ liệu (Data Shapley, LAVA, hoặc phương pháp đề xuất).
 - Sau khi thực hiện định giá, hệ thống sẽ trả về **biểu đồ** hiển thị kết quả về giá trị đóng góp của từng điểm dữ liệu. Những điểm dữ liệu kém chất lượng hoặc nhiễu sẽ được nhận diện và hiển thị cụ thể, cùng với các thông tin về ảnh hưởng của chúng đối với mô hình học máy.
 - **Biểu đồ và hình ảnh** trả về sẽ giúp người dùng hình dung được tác động của từng điểm dữ liệu đối với mô hình và dễ dàng thực hiện các thao tác tiếp theo như loại bỏ dữ liệu nhiễu hoặc tinh chỉnh dữ liệu để cải thiện chất lượng mô hình.

d. Phương pháp đánh giá và thử nghiệm hệ thống:

- **Thử nghiệm:** Thử nghiệm hệ thống với các bộ dữ liệu phổ biến như **MNIST**, **CIFAR**, và một số tập dữ liệu khác để đảm bảo tính chính xác và khả năng tổng quát của phương pháp đề xuất.
- **Đánh giá:** So sánh hiệu suất của các phương pháp định giá dữ liệu trên các tiêu chí như **thời gian xử lý**, **bộ nhớ tiêu thụ**, và **độ chính xác** của mô hình sau khi loại bỏ dữ liệu nhiễu.
- **Biểu đồ đánh giá:** Sử dụng các biểu đồ trực quan để so sánh kết quả từ các phương pháp khác nhau, đồng thời cho thấy lợi ích của việc cải thiện dữ liệu đầu vào đối với chất lượng tổng thể của mô hình.

e. Giới hạn của nghiên cứu:

- **Dữ liệu giới hạn:** Các bộ dữ liệu thử nghiệm chủ yếu là các tập dữ liệu ảnh phổ biến, giới hạn phạm vi nghiên cứu trong các ứng dụng thị giác máy tính.
- **Giới hạn về thời gian và tài nguyên:** Phương pháp tối ưu hóa thời gian và bộ nhớ sẽ được ưu tiên thử nghiệm, nhưng vẫn cần cân nhắc các hạn chế về tài nguyên và

tính khả thi của hệ thống demo khi áp dụng cho các bài toán thực tiễn với quy mô lớn hơn.

f. Tài liệu tham khảo:

[1] Neil Jethani, Mukund Sudarshan, Ian Connick Covert, Su-In Lee, Rajesh Ranganath: Fast-

SHAP: Real-Time Shapley Value Estimation. ICLR 2022

[2] Hoang Anh Just, Feiyang Kang, Tianhao Wang, Yi Zeng, Myeongseob Ko, Ming Jin, Ruoxi

Jia: LAVA: Data Valuation without Pre-Specified Learning Algorithms. ICLR 2023

[3] Amirata Ghorbani, James Y. Zou: DataShapley: Equitable Valuation of Data for Machine

Learning. ICML 2019: 2242-2251

[4] Siyi Tang, Amirata Ghorbani, Rikiya Yamashita, Sameer Rehman, Jared A. Dunnmon,

James Y. Zou, Daniel L. Rubin: Data Valuation for Medical Imaging Using Shapley Value:

Application on A Large-scale Chest X-ray Dataset. CoRR abs/2010.08006 (2020)

[5] Stephanie Schoch, Haifeng Xu, Yangfeng Ji: CS-Shapley: Class-wise Shapley Values for

Data Valuation in Classification. NeurIPS 2022

[6] Yongchan Kwon, Manuel A. Rivas, James Zou: Efficient Computation and Analysis of

Distributional Shapley Values. AISTATS 2021: 793-801

Kế hoạch thực hiện: (Mô tả kế hoạch làm việc và phân công công việc cho từng sinh viên tham gia)

Giai đoạn	Thời gian	Mục tiêu
1	Từ tháng 9 đến giữa tháng 10	Thực nghiệm những nghiên cứu trước đó trên các bộ dataset phổ biến về ảnh, và dataset tạo sinh
2	Từ giữa tháng 10 đến tháng 12	So sánh với những ý tưởng đề xuất
3	Từ tháng 12 đến giữa tháng 12	Demo các phương pháp đã thực hiện

4	Từ giữa tháng 12 đến hết	Viết báo cáo
Xác nhận của CBHD (Ký tên và ghi rõ họ tên)	TP. HCM, ngày 10 tháng 9 năm 2024 Sinh viên (Ký tên và ghi rõ họ tên)	

Lời cảm ơn

Không ai đạt được điều gì đó to lớn mà không nhờ sự giúp đỡ của những người xung quanh, cho dù là trực tiếp hay gián tiếp đi nữa. Để hoàn thành được khóa luận này, tác giả may mắn nhận được nhiều sự giúp đỡ và hỗ trợ từ quý thầy, cô, anh chị, bạn bè và người thân. Tác giả xin dành những trang đầu tiên này để bày tỏ lòng tri ân của mình tới tất cả mọi người, những người đã đồng hành cùng nhóm trong khoảng thời gian vừa qua.

Đầu tiên, tác giả xin gửi lời cảm ơn sâu sắc đến toàn thể các thầy cô của Trường Đại học Công nghệ Thông tin nói chung và các thầy cô Khoa học máy tính nói riêng. Nhờ những kiến thức quý giá mà thầy cô đã truyền đạt, cũng như việc hỗ trợ tận tình trong suốt khoảng thời gian thực hiện, nhóm đã hoàn thành khóa luận và đạt được các kết quả đáng ghi nhận.

Tác giả xin đặc biệt cảm ơn TS. Võ Nguyễn Lê Duy là người đã truyền cảm hứng, tận tình hướng dẫn và hỗ trợ tận tình về kiến thức, tạo môi trường thuận lợi để nhóm có thể học hỏi, trao đổi với các bạn, các em trong nhóm nghiên cứu. Đây là những kiến thức, kinh nghiệm quý giá, không chỉ có tác dụng trong khóa luận tốt nghiệp này mà còn trong khoảng thời gian làm việc trong chặng đường tiếp theo.

Cuối cùng, Chúng em xin bày tỏ lòng tri ân đến gia đình và người thân, những người đã luôn là những hậu phương vững chắc và luôn ủng hộ từng quyết định mà nhóm đưa ra.

Mặc dù đã nỗ lực rất nhiều để luận văn được hoàn thiện nhất, song khó có thể tránh khỏi thiếu sót và hạn chế. Kính mong nhận được sự thông cảm và ý kiến đóng góp từ quý thầy cô và các bạn.

TP. Hồ Chí Minh, ngày 1 tháng 1 năm 2025
Nguyễn Trường Thịnh

Mục lục

Thông tin hội đồng chấm khóa luận tốt nghiệp	i
Lời cảm ơn	xii
Mục lục	xii
Danh sách hình vẽ	xv
Danh sách bảng	xviii
Tóm tắt khóa luận	1
1 Giới thiệu	3
1.1 Ngữ cảnh	3
1.2 Data-centric AI	3
1.2.1 Phân biệt giữa Model-centric AI và Data-centric AI	4
1.2.2 Phân biệt giữa Data-centric AI và Adversarial Machine Learning	5
1.3 Data valuation	7
1.3.1 Phương pháp truyền thống	7
1.3.2 Giới hạn của phương pháp truyền thống	8
1.4 Ứng dụng	9
1.4.1 Ứng dụng tiềm năng	9
1.4.2 Ứng dụng thực tiễn	10
2 Tổng quan	13
2.1 Phát biểu bài toán	14

2.2	Leave-one-out	14
2.3	Data Shapley	15
2.3.1	Định Nghĩa Data Shapley	15
2.3.2	Ba Tính Chất Công Bằng trong Data Shapley	17
2.4	Truncated Monte Carlo Shapley	18
2.4.1	Công thức dựa trên giá trị kỳ vọng	18
2.4.2	Monte Carlo Sampling	19
2.4.3	Vai trò của Truncation trong TMC-Shapley	19
2.4.4	Sai số kỳ vọng của Monte Carlo	19
2.5	Một số biến thể của Data Shapley	20
2.5.1	Beta shapley	20
2.5.2	Class-wise shapley	21
2.6	KNN-Shapley	22
2.6.1	Tại sao lại có công thức này:	24
2.7	LAVA	26
2.7.1	Kiến thức về Optimal transport	26
2.7.2	Định nghĩa bài toán về mặt toán học	27
2.7.3	Định nghĩa trên phân phối rời rạc	27
2.7.4	Phương pháp Sinkhorn-Knopp	29
2.8	Phân cấp trong Optimal Transport	33
2.8.1	Phân phối có điều kiện	33
2.8.2	Label Distance trong hàm Chi phí	33
2.9	Đánh giá điểm dữ liệu thông qua Gradient được hiệu chỉnh	35
2.9.1	Đánh giá điểm dữ liệu dựa trên Gradient	35
2.9.2	Gradient được hiệu chỉnh (Calibrated Gradients)	35
2.9.3	Công thức Gradient được hiệu chỉnh	36
3	Đề xuất cải tiến	39
3.1	Cải tiến KNN-shapley	39
3.2	Cải tiến LAVA	40
4	Thực nghiệm	43
4.1	Phát hiện dữ liệu nhiễu	43
4.2	Đánh giá hiệu suất	44

4.3	Độ đo các phương pháp	44
4.3.1	F1-score	44
4.3.2	Độ đo WAD	45
4.4	Thiết lập thí nghiệm	46
4.4.1	Tập dữ liệu và tiền xử lý	46
4.4.2	Cách chia dữ liệu	50
4.4.3	Thuật toán so sánh	51
4.5	Kết quả	51
4.5.1	Kết quả trên F1-score	52
4.5.2	Thí nghiệm trên bộ dữ liệu time series:	57
4.6	Thí nghiệm so sánh KNN-shapley và LAVA	57
4.6.1	Table	58
4.6.2	NLP	59
4.6.3	CV	60
4.6.4	Regression	61
4.6.5	Y tế (Healthcare)	62
4.6.6	Dữ liệu chuỗi thời gian (Time Series)	63
4.7	Thí nghiệm thực hiện thêm/xóa lần lượt dữ liệu được xem là tốt hoặc xấu	64
4.8	So sánh tìm nhiều với các phương pháp cổ điển	66
5	Thực nghiệm các khía cạnh khác	69
5.1	Độ phức tạp thời gian	69
5.2	Cân bằng lại dữ liệu mất cân bằng	70
5.3	Chất lượng embedding có tác động đến kết quả thế nào	71
5.4	Khảo sát batch size trong LAVA cải tiến	74
5.5	Khảo sát Threshold trong KNN-shapley cải tiến	75
5.6	Ý nghĩa của việc xử lý nhiều nhãn	75
6	Kết luận và hướng nghiên cứu tiếp theo	77
6.1	Kết quả đạt được trong nghiên cứu	77
6.2	Bàn luận về hướng phát triển	78
Tài liệu tham khảo		79

Danh sách hình vẽ

1.1	So sánh giữa AI tập Trung vào mô hình (Model-centric AI) và AI tập trung Vào Dữ Liệu (Data-centric AI).	4
1.2	So sánh giữa AI Tập Trung Vào Dữ Liệu (data-centric AI) và Học Máy Đối Kháng (Adversarial machine learning).	5
1.3	Hình minh họa rằng khi huấn luyện trên một tập có nhiều, mô hình sẽ không tăng mạnh độ chính xác bằng so với việc áp dụng mô hình tương tự và khử nhiễu. Nguồn: transfer-lab	7
1.4	Quy trình mô tả mô hình của các thuật toán data valuation truyền thống. .	8
1.5	Mô tả cách sử dụng optimal transport áp dụng vào quy trình xử lí data valuation của thuật toán truyền thống. Điểm khác biệt là thuật toán học được thay thế bằng một utility khác (optimal transport).	8
1.6	Ví dụ về ứng dụng của data valuation trong việc mua bán dữ liệu, các nhà đóng góp dữ liệu sẽ được phân chia lợi nhuận theo chất lượng dữ liệu họ có.	11
2.1	Hình minh họa ví dụ đơn giản về cách tính một điểm giá trị data value trên bộ dữ liệu gồm 3 điểm: con cá, con mèo, con chó. Ở đây ta tính giá trị con cá bằng tổng đóng góp khi thêm con cá vào các tập con không chứa nó. . .	16
2.2	Đồ thị phân phối Beta, nguồn [12]	20
2.3	Mô tả ví dụ sự khác nhau khi sử dụng hàm entropic vào bên phải, và không sử dụng bên trái. Ta thấy rằng bên phải từ một điểm nguồn (màu xanh) có thể nối đến nhiều điểm đích (màu đỏ) trong khi hình bên trái nghiêm ngặt hơn.	29
2.4	Minh họa cách hoạt động của thuật toán sinkhorn-knopp, ta chuẩn hóa giá trị u, v để biến K đến miền phân phối chung $\Pi(\mu, v)$ để tìm γ	32

2.5	Sự quan trọng của khoảng cách nhãn: Các cặp dữ liệu thứ hai gần nhau hơn trong khoảng cách OT thông thường (không dựa trên nhãn) (màu vàng), trong khi điều ngược lại mới là đúng với khoảng cách có ý thức nhãn (màu xám) . Nguồn ảnh [1]	34
2.6	Minh họa tưởng tượng về sự thay đổi gradient của một điểm sẽ làm cho khoảng cách optimal transport giữa hai phân phối gần lại hay xích ra. Ví sự thay đổi nó như một cái cân (5kg) kéo khoảng cách xích lại hoặc giãn ra nếu như ta thay đổi trọng lượng của chiếc cân.	38
3.1	Minh họa việc chia nhỏ tập train và tập test để có thể tính LAVA chéo, giúp thuật toán được học cục bộ hơn và cung cấp phần giảm gánh nặng bộ nhớ.	42
4.1	Thực nghiệm trên các bộ dữ liệu khác nhau (2dplan, credit, digits, iris, pol) Trên một biểu đồ ta có ba đường lần lượt là: màu xanh nước biển là tỉ lệ phát hiện nhiễu của thuật toán trên cho lượng điểm dữ liệu chúng ta lấy ra để xét nhiễu (giả định rằng đó là nhiễu) (cột ngang). Hình màu vàng và xanh lá cây nét đứt lần lượt là kết quả tối ưu có thể đạt được, và kết quả có thể đạt được nhờ chọn random. Tên mô tả các thí nghiệm lần lượt là: tên bộ dữ liệu, nhiễu đặc trưng (feature) hoặc nhiễu nhãn (label) và cuối cùng là tên phương pháp.	58
4.2	Thực nghiệm trên các bộ dữ liệu ngôn ngữ tự nhiên (BBC, IMDB, SST2). Xem mô tả tại 4.1	59
4.3	Thực nghiệm trên các bộ dữ liệu thị giác máy tính (CIFAR-10, Fashion MNIST, MNIST, STL-10, SVHN). Xem mô tả tại 4.1	60
4.4	Thực nghiệm trên các bộ dữ liệu hồi quy (Regression) (Creditcard, Diabetes, Echomonths, MV, Stock, Wave Energy). Xem mô tả tại 4.1	61
4.5	Thực nghiệm trên bộ dữ liệu y tế (Chest X-ray).Xem mô tả tại 4.1	62
4.6	Thực nghiệm trên dữ liệu Time series. So sánh với một thuật toán khác là time-series shapley đề xuất trong [25] Xem mô tả tại 4.1	63

Danh sách bảng

1	Danh sách thuật ngữ và định nghĩa	xxi
4.1	Bảng các bộ dữ liệu thực nghiệm dạng bảng	46
4.2	Bảng các bộ dữ liệu ngôn ngữ tự nhiên được thực nghiệm (BBC, IMDB, SST-2)	48
4.3	Bảng các bộ dữ liệu thị giác máy tính được thực nghiệm nằm trong <code>torchvision.datasets</code>	49
4.4	Đánh giá hiệu suất F1-score trên các bộ dữ liệu có nhiều đặc trưng (feature) và nhiều nhãn (label). Lần lượt đưa ra kết quả precision, recall, F1-score và hiệu suất f1 của mô hình sau khi ta loại bỏ dữ liệu được cho là có khả năng cao là nhiều nhất theo các thuật toán. Thuật toán huấn luyện là logistic regression. Baseline performance là độ chính xác trên toàn bộ tập dữ liệu.	53
4.5	Thực nghiệm trên CIFAR với tỉ lệ nhiều đặc trưng là 0.2	53
4.6	Thực nghiệm trên CIFAR với tỉ lệ nhiều nhãn là 0.3. Mô tả bảng 4.4 . . .	54
4.7	Thực nghiệm trên FashionMnist với tỉ lệ nhiều đặc trưng là 0.2. Mô tả bảng 4.4	54
4.8	Thực nghiệm trên FashionMnist với tỉ lệ nhiều đặc trưng là 0.3. Mô tả bảng 4.4	54
4.9	Thực nghiệm trên FashionMnist với tỉ lệ nhiều nhãn là 0.2. Mô tả bảng 4.4	55
4.10	Thực nghiệm trên FashionMnist với tỉ lệ nhiều nhãn là 0.3. Mô tả bảng 4.4	55
4.11	Thực nghiệm trên BBC với nhiều đặc trưng là 0.2. Mô tả bảng 4.4	55
4.12	Thực nghiệm trên BBC với nhiều nhãn là 0.2. Mô tả bảng 4.4	56
4.13	Thực nghiệm trên Digits với tỉ lệ nhiều nhãn là 0.2. Mô tả bảng 4.4	56
4.14	Thực nghiệm trên Digits với tỉ lệ nhiều đặc trưng là 0.2. Mô tả bảng 4.4 . .	56

Thuật ngữ chuyên ngành

Thuật ngữ	Định nghĩa
Couplings	Lý thuyết xác suất về phân phối chung
Data-Centric AI	Một hướng nghiên cứu tập trung vào cải tiến dữ liệu
Data valuation	Đánh giá dữ liệu
Data value	Giá trị của thuật toán cần tìm kiếm
Data Shapley	Thuật toán tìm giá trị data value liên quan đến lý thuyết trò chơi
Empirical Risk Minimizer (ERM)	Một phương pháp học máy nhằm giảm thiểu rủi ro thực nghiệm bằng cách tối ưu hóa một hàm mất mát trên tập huấn luyện
K-Nearest Neighbours algorithm (KNN)	Thuật toán tìm kiếm các điểm lân cận gần nhất
Leave-one-out (LOO)	Thuật toán ngay thơ loại bỏ từng điểm dữ liệu rồi đánh giá
Marginal	Đóng góp cận biên
Optimal Transport (OT)	Bài toán vận chuyển tối ưu
Time-series	Bộ dữ liệu có hướng, thu thập theo trình tự thời gian
Truncated Monte Carlo sampling (TMC)	Phương pháp lấy mẫu Monte Carlo
Utility	Hàm tiện ích đo lường độ hiệu quả của một mô hình

Bảng 1: Danh sách thuật ngữ và định nghĩa

Tóm tắt khóa luận

Trong bối cảnh dữ liệu trở thành yếu tố cốt lõi trong công nghệ và kinh tế, việc đánh giá giá trị dữ liệu để xác định những đóng góp thực sự của chúng vào dự đoán của mô hình vẫn là một hướng tìm năng và thách thức lớn. Mặc dù đã có nhiều nghiên cứu về các phương pháp đánh giá dữ liệu công bằng, lĩnh vực này vẫn còn khá mới, với nhiều hạn chế và chưa được ứng dụng rộng rãi. Một trong những bước tiến đầu tiên là Data Shapley được giới thiệu bởi [7], đặt nền móng cho các phương pháp đánh giá dữ liệu. Theo cách tiếp cận truyền thống, việc đánh giá dữ liệu được hiểu là quá trình phân bổ công bằng mức độ đóng góp của từng điểm dữ liệu trong tập huấn luyện vào hiệu suất của mô hình học máy.

Tuy nhiên, phương pháp này gặp phải một số hạn chế lớn. Giá trị của dữ liệu phụ thuộc mạnh mẽ vào các lựa chọn thiết kế của thuật toán học, điều này không phù hợp với nhiều tình huống sử dụng thực tế. Chẳng hạn, trong các quy trình thu thập dữ liệu, việc ưu tiên các nguồn dữ liệu khác nhau hoặc thiết lập cơ chế đánh giá cho thị trường dữ liệu yêu cầu dữ liệu được đánh giá trước khi lựa chọn thuật toán học cụ thể. Một hạn chế khác là việc phải chạy lại thuật toán học có và không có từng điểm dữ liệu để đánh giá giá trị của chúng, gây ra gánh nặng tính toán lớn và không khả thi với các tập dữ liệu lớn.

Vì những khó khăn này, nghiên cứu của em tập trung vào việc phát triển các phương pháp đánh giá dữ liệu độc lập với quy trình huấn luyện một thuật toán học. Cụ thể, nghiên cứu hướng tới hai phương pháp chính: (1) sử dụng thuật toán K-Nearest Neighbours mô phỏng Data Shapley truyền thống (phần 2.6), và (2) phương pháp sử dụng khoảng cách Optimal Transport (OT) (phần 2.7). Với các khung lý thuyết này, giá trị của từng điểm dữ liệu có thể được tính toán hiệu quả, từ đó ứng dụng để phát hiện các điểm ngoại lệ hoặc nhiễu trong tập huấn luyện khi so sánh với tập xác thực.

Những ứng dụng của phương pháp này có tiềm năng được triển khai rộng rãi trong

nhiều lĩnh vực như thị giác máy tính, xử lý ngôn ngữ tự nhiên, y tế, chuỗi thời gian (time-series), và nhiều bối cảnh khác. Thông qua việc đánh giá dữ liệu hiệu quả, nghiên cứu kỳ vọng sẽ mở ra các hướng đi mới trong quản lý, đánh giá, và sử dụng dữ liệu một cách tối ưu.

Tóm tắt các phần có trong đồ án:

- Chương 1: Giới thiệu ngữ cảnh, thể loại nghiên cứu, tiềm năng của nghiên cứu.
- Chương 2: Tổng quan sơ lược về các phương pháp trước đó.
- Chương 3: Đề xuất cải tiến, áp dụng vào thuật toán cũ.
- Chương 4: Thực nghiệm và phân tích kết quả trên nhiều ứng dụng.
- Chương 5: Thực nghiệm các khía cạnh khác, các hướng tìm năng liên quan đến data valuation.
- Chương 6: Thảo luận và hướng nghiên cứu tương lai.

Code vào demo được đặt tại: [github: thinhhsama](#)

Chương 1

Giới thiệu

1.1 Ngữ cảnh

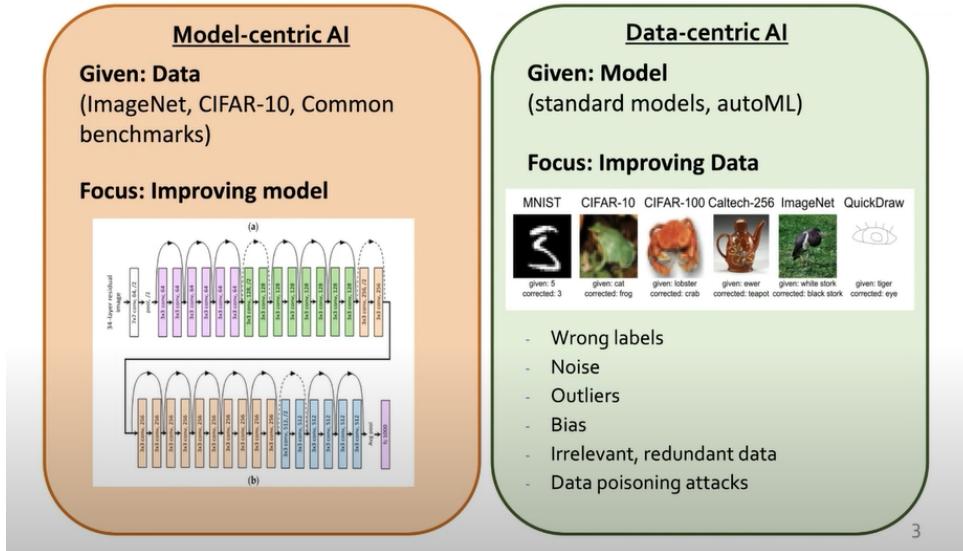
Khi chúng ta tiến hành phân tích dữ liệu trên những ứng dụng thực tế ví dụ như hình ảnh trên camara của một công ty. Việc những dữ liệu bị lỗi hay mất mát thông tin là điều không thể tránh khỏi, hay trong lĩnh vực y tế dữ liệu có thể rất đắt đỏ và cần được nhiều chuyên gia đánh giá. Như vậy kể cả trường hợp bộ dữ liệu là khan hiếm hay có nhiều dữ liệu chúng ta cũng tò mò muốn biết "dữ liệu nào là quan trọng?" hay "làm thế nào để đánh giá ảnh hưởng của từng điểm dữ liệu đến hiệu suất mô hình?" điều này làm dẫn đến nghiên cứu về đánh giá dữ liệu của nhóm chúng em.

Cần phải xác định nghiên cứu đánh giá dữ liệu tuy không phải xu hướng ở thời điểm hiện tại nhưng nó là một hướng quan trọng nằm trong Data-Centric AI. Vậy Data-Centric AI khác gì Model-Centric AI? Tiếp theo đây chúng em muốn phân biệt một chút về hướng tiếp cận chủ đạo và khác biệt giữa Data-centric AI, Model-centric AI và adversarial machine learning.

1.2 Data-centric AI

Data-centric AI là hướng tiếp cận trong học máy tập trung chủ yếu vào việc **nâng cao chất lượng** (và đôi khi cả **số lượng**) của dữ liệu huấn luyện, thay vì đầu tư quá nhiều vào việc tinh chỉnh mô hình. Mục tiêu của Data-centric AI là xây dựng một quy trình chuẩn để **chuẩn hóa, làm sạch, và tăng cường** dữ liệu, qua đó giúp mô hình học tốt hơn mà không nhất thiết phải thay đổi cấu trúc hay thuật toán học.

1.2.1 Phân biệt giữa Model-centric AI và Data-centric AI



Hình 1.1: So sánh giữa AI tập Trung vào mô hình (Model-centric AI) và AI tập trung Vào Dữ Liệu (Data-centric AI).

Model-centric AI

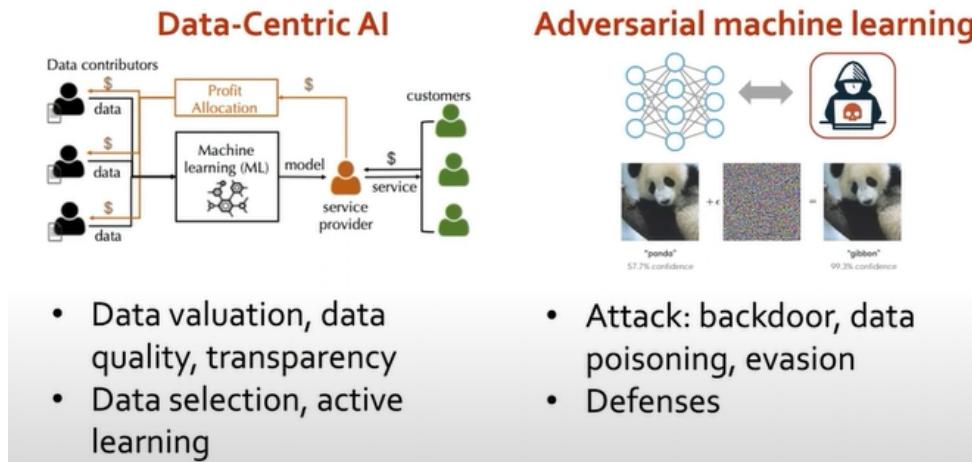
- **Dữ liệu đầu vào:** Các bộ dữ liệu được kiểm chứng và có chất lượng cao (ví dụ: ImageNet, CIFAR-10).
- **Mục tiêu chính:** Cải thiện mô hình thông qua:
 - Tối ưu hóa kiến trúc mạng nơ-ron.
 - Điều chỉnh các siêu tham số (hyperparameters).
 - Nâng cao thuật toán huấn luyện.

Data-centric AI

- **Mô hình đầu vào:** Các mô hình tiêu chuẩn hoặc hệ thống học máy tự động (AutoML).
- **Mục tiêu chính:** Cải thiện chất lượng dữ liệu bằng cách:
 - Sửa các nhãn sai.
 - Loại bỏ nhiễu và các điểm dữ liệu bất thường.

- Giải quyết thiên vị trong bộ dữ liệu.
- Loại bỏ các điểm dữ liệu không liên quan hoặc dư thừa.

1.2.2 Phân biệt giữa Data-centric AI và Adversarial Machine Learning



Hình 1.2: So sánh giữa AI Tập Trung Vào Dữ Liệu (data-centric AI) và Học Máy Đôi Kháng (Adversarial machine learning).

Data-centric AI

- **Mục tiêu chính:** Nâng cao chất lượng và tính hữu dụng của dữ liệu để cải thiện hiệu suất của các quy trình học máy.
- **ĐỊNH GIÁ DỮ LIỆU (DATA VALUATION):** Đánh giá chất lượng và giá trị của dữ liệu.
- **Học chủ động (Active learning):** Ưu tiên chọn các điểm dữ liệu có giá trị cao để gán nhãn trước.
- **Lựa chọn dữ liệu (Data selection):** Loại bỏ các điểm dữ liệu thừa hoặc không liên quan.
- **Tính minh bạch (Transparency):** Tăng cường độ tin cậy và hiểu biết về dữ liệu.

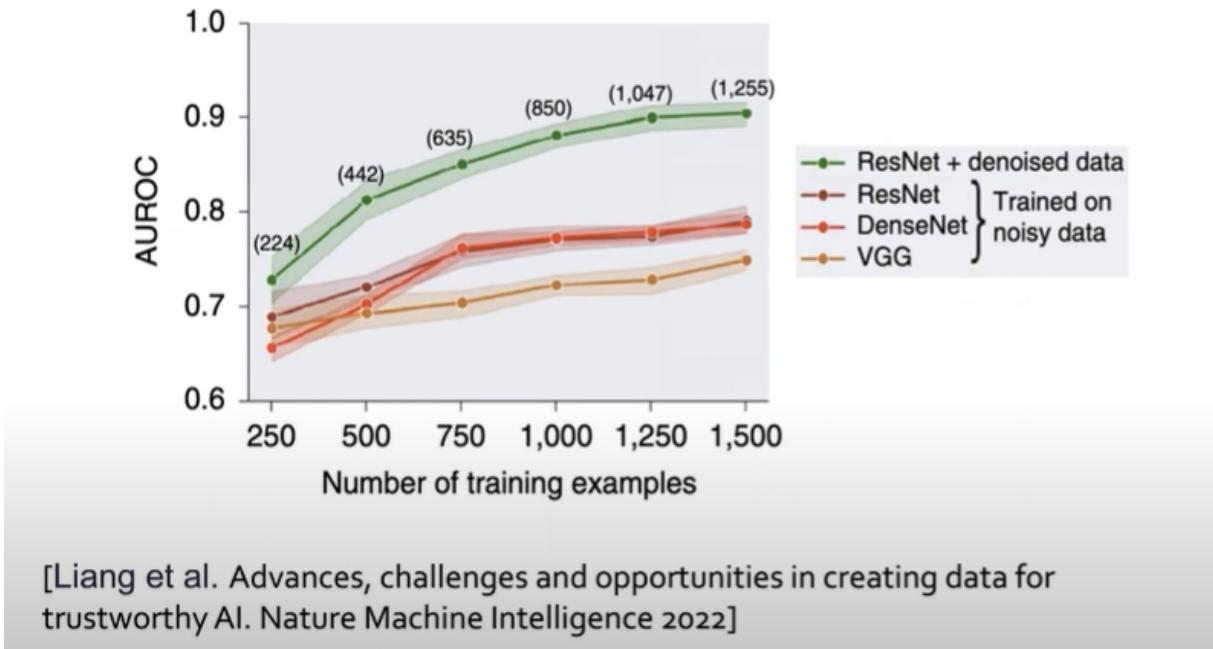
Học Máy Đôi Kháng (Adversarial Machine Learning)

1.2. Data-centric AI

- **Mục tiêu chính:** Bảo vệ hệ thống học máy khỏi các cuộc tấn công từ đối thủ.
- **Các loại tấn công:**
 - Tấn công cài cửa hậu (Backdoor attacks).
 - Tấn công làm nhiễm độc dữ liệu (Data poisoning attacks).
 - Tấn công né tránh (Evasion attacks).
- **Phòng thủ:** Phát triển các phương pháp mạnh mẽ để giảm thiểu mối đe dọa từ đối thủ.

Nhận xét: Cả AI tập trung vào mô hình và AI tập trung vào dữ liệu đều là những phương pháp quan trọng để phát triển các hệ thống AI mạnh mẽ. Trong khi AI tập trung vào mô hình chú trọng vào việc tinh chỉnh thuật toán và kiến trúc, thì AI tập trung vào dữ liệu nhấn mạnh việc cải thiện chất lượng dữ liệu nhằm đạt được hiệu suất mô hình tốt hơn. Học máy đối kháng (Adversarial Machine Learning) đóng vai trò bổ trợ, đảm bảo an ninh và khả năng chống lại các cuộc tấn công nhắm vào hệ thống AI.

Như vậy mục tiêu đề ra của data valuation nói riêng và Data-centric AI nói chung sẽ áp dụng các thuật toán machine learning săn có và quan sát sự thay đổi của dữ liệu có tác động thế nào đến hiệu suất mô hình. Ví dụ trong hình 1.3:



Hình 1.3: Hình minh họa rằng khi huấn luyện trên một tập có nhiều, mô hình sẽ không tăng mạnh độ chính xác bằng so với việc áp dụng mô hình tương tự và khử nhiễu.

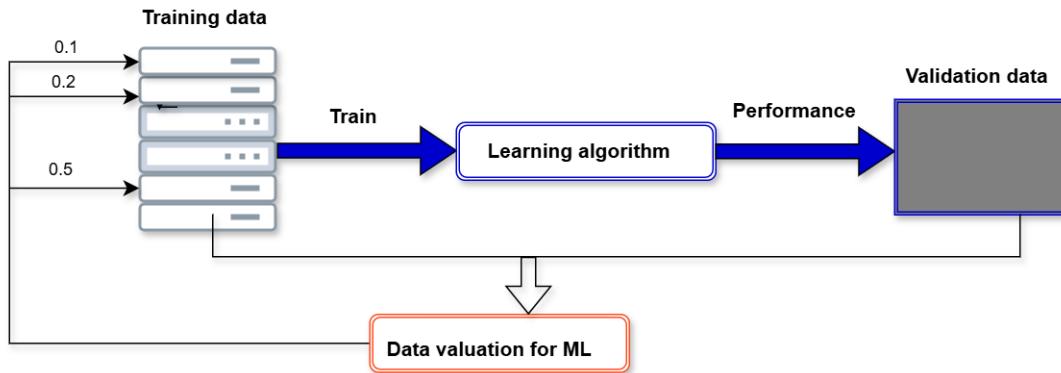
Nguồn: transfer-lab

1.3 Data valuation

1.3.1 Phương pháp truyền thống

Điểm khác nhau giữa phương pháp truyền thống và phương pháp đề xuất là có và không Learning algorithm hình 1.4. Thuật toán học có tác dụng để tìm ra mối quan hệ trong dữ liệu.

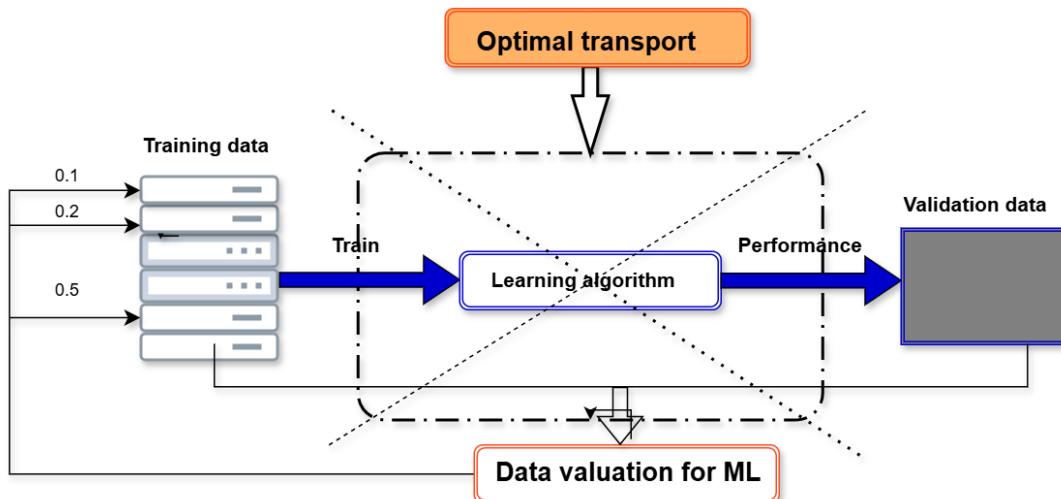
1.3. Data valuation



Hình 1.4: Quy trình mô tả mô hình của các thuật toán data valuation truyền thống.

1.3.2 Giới hạn của phương pháp truyền thống

Việc huấn luyện đi huấn luyện lại mô hình gây mất rất nhiều chi phí. Cho nên đề xuất là thay learning algorithm bằng một độ đo ví dụ Optimal transport (trong LAVA) (Hình 1.5).



Hình 1.5: Mô tả cách sử dụng optimal transport áp dụng vào quy trình xử lí data valuation của thuật toán truyền thống. Điểm khác biệt là thuật toán học được thay thế bằng một utility khác (optimal transport).

1.4 Ứng dụng

1.4.1 Ứng dụng tiềm năng

Data valuation cung cấp hai ứng dụng tiềm năng chính trong việc đánh giá dữ liệu:

- **Đánh giá chất lượng dữ liệu:** Dựa trên định nghĩa của Data Shapley 2.3, giá trị của từng điểm dữ liệu có thể được xem như một chỉ số đo lường chất lượng gọi là shapley value. Cụ thể, các điểm dữ liệu được định giá cao là những điểm có đóng góp đáng kể vào hiệu suất của mô hình dự đoán. Ngược lại, những điểm có giá trị thấp hoặc âm có thể làm giảm hiệu suất. Thông tin này có thể được sử dụng để:
 - Loại bỏ các dữ liệu gây hại hoặc nhiễu.
 - Xác định dữ liệu có giá trị cao và thu thập thêm các dữ liệu tương tự.
- **Thích nghi miền (domain adaptation):** Data valuation cung cấp một phương pháp đơn giản để điều chỉnh tập dữ liệu huấn luyện sao cho phù hợp hơn với tập kiểm tra. Điều này giúp cải thiện hiệu suất trong các tình huống mà dữ liệu huấn luyện và dữ liệu kiểm tra có sự khác biệt về phân phối.
- **Giá trị của dữ liệu chất lượng thấp có thể là:**
 - **Dữ liệu bị gán nhãn sai:** Việc gán nhãn sai trong tập dữ liệu huấn luyện là phổ biến và có thể xảy ra do con người gán nhãn hoặc cố tình phá hoại (ví dụ: tấn công gây nhiễu dữ liệu - data poisoning). Những điểm dữ liệu bị gán nhãn sai này thường có giá trị Data Shapley thấp, vì chúng cung cấp thông tin không chính xác cho nhiệm vụ học máy và làm giảm hiệu suất mô hình.
 - **Dữ liệu nhiễu:** Những điểm dữ liệu khác biệt hoặc không liên quan đến nhiệm vụ có thể ảnh hưởng tiêu cực đến mô hình.

Sử dụng data valuation, chúng ta có thể:

1. **Loại bỏ dữ liệu có giá trị âm:** Những điểm dữ liệu có giá trị Shapley âm được xem là không đóng góp hoặc làm giảm hiệu suất mô hình, do đó cần được loại bỏ.
2. **Tăng trọng số cho dữ liệu có giá trị cao:** Tăng cường ảnh hưởng của các điểm dữ liệu đóng góp tích cực bằng cách sử dụng hàm matsu có trọng số, trong đó trọng số của mỗi điểm tỉ lệ với giá trị shapley value của nó.

Ví dụ ứng dụng trong thích nghi miền (domain adaptation)

Trong thực tế, dữ liệu huấn luyện thường khác biệt về mặt thống kê hoặc chất lượng so với dữ liệu kiểm tra. Điều này thường xảy ra khi việc thu thập dữ liệu kiểm tra tương đồng với dữ liệu triển khai trở nên tốn kém, buộc chúng ta phải sử dụng dữ liệu huấn luyện từ một nguồn khác. Từ đó, hai câu hỏi quan trọng được đặt ra:

- *Những nguồn dữ liệu nào trong tập huấn luyện thực sự đóng góp tích cực cho việc thích nghi?*
- *Ngược lại, những nguồn dữ liệu nào có thể gây tác động tiêu cực đến quá trình thích nghi?*

1.4.2 Ứng dụng thực tiễn

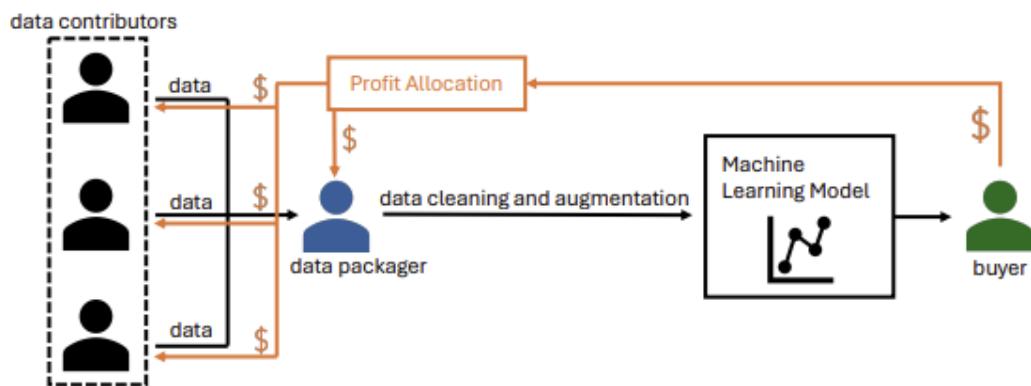
Các phương pháp định giá dữ liệu đã được áp dụng thành công trong nhiều lĩnh vực khác nhau được chúng em kiểm tra, bao gồm:

- **Thị giác máy tính (Computer Vision):**
 - Phát hiện và loại bỏ các điểm dữ liệu nhiễu trong các bài toán phân loại hình ảnh.
 - Tăng cường chất lượng của tập dữ liệu huấn luyện bằng cách ưu tiên các điểm dữ liệu quan trọng.
- **Xử lý ngôn ngữ tự nhiên (Natural Language Processing):**
 - Xác định các mẫu văn bản có ảnh hưởng lớn đến việc học từ vựng hoặc ngữ nghĩa.
 - Loại bỏ các điểm dữ liệu ít giá trị, giúp cải thiện hiệu quả huấn luyện.
- **Y tế (Healthcare):**
 - Phân tích dữ liệu bệnh nhân để xác định các mẫu quan trọng, hỗ trợ chẩn đoán và dự đoán bệnh.
 - Phát hiện dữ liệu ngoại lệ trong các nghiên cứu lâm sàng.
- **Chuỗi thời gian (Time-series):**

1.4. Ứng dụng

- Xử lý dữ liệu cảm biến để phát hiện lỗi hoặc dự đoán thời tiết, giao thông.
- Đánh giá giá trị của các điểm dữ liệu trong chuỗi thời gian để tối ưu hóa mô hình dự đoán.

Các ứng dụng trên không chỉ minh họa tiềm năng rộng lớn của data valuation trong việc định giá dữ liệu mà còn chỉ ra khả năng giải quyết các vấn đề phức tạp như nhiều dữ liệu, thích nghi miền, và tối ưu hóa dữ liệu huấn luyện. Tuy nhiên, để mở rộng tính ứng dụng, cần cải tiến các thuật toán nhằm giảm chi phí tính toán và tăng khả năng thích nghi với nhiều loại dữ liệu.



Hình 1.6: Ví dụ về ứng dụng của data valuation trong việc mua bán dữ liệu, các nhà đóng góp dữ liệu sẽ được phân chia lợi nhuận theo chất lượng dữ liệu họ có.

Chương 2

Tổng quan

Phần này chúng em sẽ đi phân tích sơ lược về một số phương pháp từ trước đến nay đưa ra điểm yếu của các phương pháp ví dụ như về thời gian tính toán, về khả năng phát hiện nhiễu hoặc ngoại lệ hay cả về khả năng cải thiện hiệu suất mô hình.

Nói sơ qua về các phương pháp, đầu tiên lấy ý tưởng từ việc đánh giá mức độ đóng góp hoặc tầm quan trọng của từng điểm dữ liệu bằng cách loại bỏ nó ra khỏi tập dữ liệu huấn luyện và quan sát sự thay đổi trong hiệu suất mô hình. Ý tưởng này xuất phát từ các khái niệm cơ bản trong lý thuyết giá trị cận biên (marginal contribution) trong toán kinh tế và thống kê, cụ thể vào năm 1953 Lloyd Shapley đã đề xuất một lý thuyết trò chơi và đo mức độ đóng góp của người chơi vào kết quả cuối cùng của trò chơi. Ở trong bối cảnh dữ liệu mỗi điểm dữ liệu được coi như một người chơi, còn trò chơi là việc xây dựng mô hình. Giá trị của từng điểm dữ liệu được xác định bằng cách loại bỏ nó và đánh giá sự thay đổi trong hiệu suất mô hình. Một loạt các phương pháp liên quan đến marginal ra đời từ phần 2.2 đến 2.6. Một cách tiếp cận khác là LAVA 2.7 đề xuất về sự thay đổi các điểm dữ liệu trên gradient, các phương pháp gradient tập trung vào việc đánh giá độ nhạy của một giá trị tiện ích (utility value) đối với sự thay đổi trọng số của một điểm dữ liệu cụ thể. Cụ thể phương pháp tính mức độ thay đổi của giá trị tiện ích (thước đo hiệu suất mô hình hoặc chi phí) khi điểm dữ liệu được gán trọng số cao hơn hoặc thấp hơn. Giá trị tiện ích đại diện cho một thước đo cụ thể phản ánh hiệu quả hoặc hiệu suất mô hình ví dụ như: độ chính xác trên tập kiểm tra đo khả năng tổng quát hóa của mô hình, hàm mất mát đo mức độ sai số so với giá trị thực, hay một chi phí sử dụng optimal transport(OT) đo sự khác biệt giữa hai phân phối. LAVA [10] đã ra đời sử dụng optimal transport làm hàm tiện ích.

2.1 Phát biểu bài toán

Bài toán nằm trong phạm vi học có giám sát, bao gồm ba thành phần chính:

Thành phần đầu tiên là tập huấn luyện $D = \{(x_i, y_i)\}_{i=1}^n$, trong đó D chứa n dữ liệu nguồn. Mỗi cặp (x_i, y_i) đại diện cho dữ liệu thứ i với x_i là đặc trưng đầu vào và y_i là đầu ra. Đầu ra y_i có thể là giá trị phân loại (categorical) trong các bài toán phân loại (classification) hoặc giá trị thực (real) trong bài toán hồi quy (regression).

Thành phần thứ hai là mô hình huấn luyện, được ký hiệu là A . Mô hình này được coi như một "hộp đen" (black-box), trong đó tập dữ liệu D được đưa vào để tạo ra một bộ dự đoán (predictor).

Thành phần cuối cùng là thuật toán đánh giá chất lượng của predictor đã được huấn luyện, được gọi là điểm hiệu suất (performance score) V . Thuật toán V cũng được xem như một hộp đen, nhận bộ dự đoán làm đầu vào và trả về điểm đánh giá.

Công thức hóa bài toán này như sau: chúng ta biểu diễn điểm hiệu suất là $V(S, A)$ hoặc đơn giản hơn là $V(S)$, nơi S là một tập con (subset) chỉ số của các phần tử trong $D = \{1, 2, \dots, n\}$. Điểm hiệu suất V đánh giá chất lượng của predictor được huấn luyện trên tập dữ liệu S bằng thuật toán học A .

Mục tiêu của bài toán là xác định giá trị dữ liệu của từng điểm (x_i, y_i) trong tập D . Giá trị này, ký hiệu là $\phi_i(D, A, V)$ hoặc đơn giản hơn là $\phi_i(V)$ hay ϕ_i , phản ánh mức độ quan trọng của điểm dữ liệu đó đối với hiệu suất chung của tập dữ liệu D .

Hình minh họa xem trong hình 1.4.

2.2 Leave-one-out

Phương pháp đơn giản đầu tiên được biết đến để đo lường tầm quan trọng của một điểm dữ liệu bằng cách loại bỏ nó ra khỏi tập dữ liệu và kiểm tra tác động của việc loại bỏ đó đối với hiệu suất của mô hình.

Biểu diễn công thức:

$$\phi_i = V(A(D)) - V(A(D \setminus x_i)),$$

trong đó:

- x_i là điểm dữ liệu cần loại bỏ,

- D là tập dữ liệu ban đầu,
- $V(A(D))$ độ đo hiệu suất khi huấn luyện bằng mô hình A trên tập D (measure)
- $V(A(D \setminus x_i))$ độ đo hiệu suất khi huấn luyện bằng mô hình A trên tập D sau khi loại bỏ x_i .

Sự thay đổi của V có thể là thay đổi hiệu suất của mô hình hay hàm loss ví dụ như sau:

Chúng ta ký hiệu tập dữ liệu kiểm thử là $\{(\tilde{x}_i, \tilde{y}_i)\}_{i=1}^{n_{val}}$

- Phân loại: độ chính xác phân loại (giả định theo ERM) được huấn luyện trên một subset của D . $V(S) = \frac{1}{n_{val}} \sum_1^{n_{val}} \mathbf{1}(\tilde{y}_i - \hat{f}_A \tilde{x}_i)$
- Hồi quy: lỗi NegMSE (negative mean squared error) (giả định theo ERM) được huấn luyện trên một subset của D . $V(S) = -\frac{1}{n_{val}} \sum_1^{n_{val}} (\tilde{y}_i - \hat{f}_A \tilde{x}_i)^2$

Ý tưởng chính của phương pháp LOO là đánh giá mức độ đóng góp riêng lẻ của từng điểm dữ liệu x_i trong tập D :

- Nếu việc loại bỏ x_i dẫn đến sự suy giảm đáng kể hiệu suất của mô hình, điểm dữ liệu này được coi là quan trọng.
- Ngược lại, nếu hiệu suất không thay đổi nhiều, x_i có thể không đóng góp đáng kể vào kết quả của mô hình.

Thuật toán có ưu điểm là dễ hiểu trực quan, không yêu cầu sửa đổi kiến trúc hoặc cách huấn luyện. Tuy nhiên độ phức tạp có thể cao khi tập dữ liệu lớn hoặc mô hình phức tạp vì có thể huấn luyện lại mô hình nhiều lần, qua phần data shapley (2.3), ta sẽ thấy phương pháp này mang nhiều hạn chế. Nó không thật sự đánh giá dữ liệu công bằng.

2.3 Data Shapley

2.3.1 Định Nghĩa Data Shapley

Data Shapley được xây dựng dựa trên lý thuyết trò chơi được phát biểu lần đầu bởi Lloyd Shapley. Được Ghorbani và Zou 2019 ứng dụng vào việc đánh giá đóng góp của từng điểm dữ liệu vào hiệu suất mô hình. Khác với LOO phương pháp này tính toán

2.3. Data Shapley

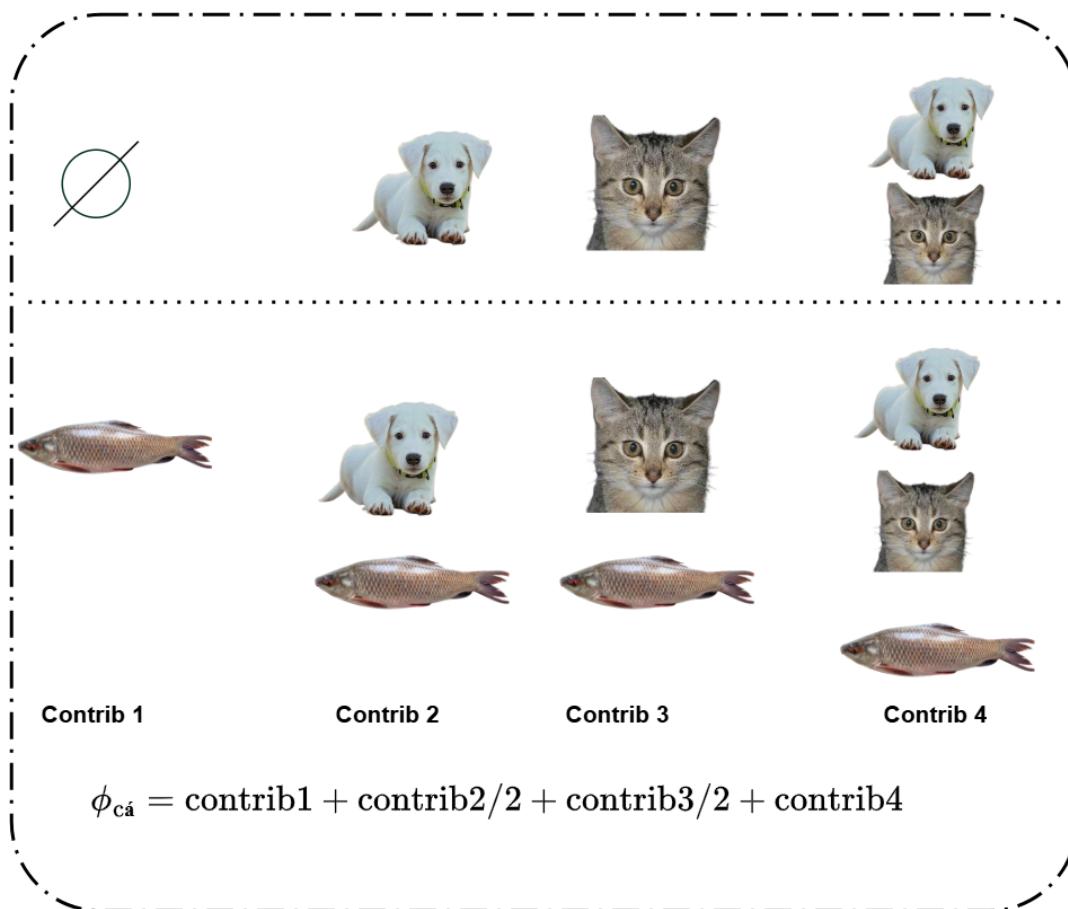
trung bình đóng góp cận biên của x_i trên tất cả các tập con S không chứa x_i .

Công thức định nghĩa giá trị Shapley của một điểm dữ liệu x_i trong một tập dữ liệu D như sau:

$$\phi_i = C \sum_{S \subseteq D \setminus \{i\}} \frac{V(S \cup \{i\}) - V(S)}{\binom{n-1}{|S|}} \quad (2.1)$$

trong đó:

- $V(S)$: Giá trị tiện ích của mô hình khi được huấn luyện trên tập con $S \subseteq D$.
- C : Hệ số tỉ lệ không đổi (Hằng số).
- $\binom{n-1}{|S|}$: Số lượng tập con S có kích thước $|S|$ trong tập $D \setminus \{i\}$, mục đích chia đều để đảm bảo tính công bằng.



Hình 2.1: Hình minh họa ví dụ đơn giản về cách tính một điểm giá trị data value trên bộ dữ liệu gồm 3 điểm: con cá, con mèo, con chó. Ở đây ta tính giá trị con cá bằng tổng đóng góp khi thêm con cá vào các tập con không chứa nó.

2.3.2 Ba Tính Chất Công Bằng trong Data Shapley

Phương pháp Data Shapley thỏa mãn ba tính chất công bằng cơ bản trong lý thuyết trò chơi:

1. Tính Đối Xứng (Symmetry)

Nếu hai điểm dữ liệu x_i và x_j có cùng đóng góp vào mô hình, nghĩa là:

$$V(S \cup \{x_i\}) = V(S \cup \{x_j\}), \forall S \subseteq D \setminus \{i, j\},$$

thì giá trị Shapley của chúng phải bằng nhau:

$$\phi_i = \phi_j.$$

2. Tính Không Thiên Vị (Dummy Property)

Nếu một điểm dữ liệu x_k không làm thay đổi giá trị tiện ích của bất kỳ tập con nào, nghĩa là:

$$V(S \cup \{x_k\}) = V(S), \forall S \subseteq D \setminus \{k\},$$

thì giá trị Shapley của x_k phải bằng 0:

$$\phi_k = 0.$$

3. Tính Thêm Vào (Additivity)

Nếu giá trị tiện ích của tập dữ liệu D là tổng giá trị của hai hàm tiện ích V_1 và V_2 , nghĩa là:

$$V(S) = V_1(S) + V_2(S), \forall S \subseteq D,$$

thì giá trị Shapley của mỗi điểm dữ liệu cũng phải là tổng giá trị Shapley tương ứng:

$$\phi_i(V) = \phi_i(V_1) + \phi_i(V_2), \forall i \in D.$$

Thỏa mãn 3 tính chất công bằng này được xem là cơ sở cho các phương pháp data valuation hướng tới, có thể thấy phương pháp LOO chưa thỏa mãn phương pháp này. Tuy nhiên các phương pháp sau có thể cũng đã bỏ đi các ràng buộc chặt chẽ bởi ba tính chất này để ưu tiên những khả năng khác (thời gian). Vì bởi lẽ thuật toán này có độ phức tạp lên tới hàm mũ khi phải xét trên tất cả tập con của D .

2.4 Truncated Monte Carlo Shapley

Việc cài đặt Data Shapley như trên có vẻ khá đơn giản, nhưng độ phức tạp lại tồn theo hàm mũ (5.1). Để tối ưu thời gian ta sẽ đi xấp xỉ thuật toán này bằng kĩ thuật Truncated Monte Carlo Shapley. Monte Carlo được xem như là một phương pháp lấy mẫu ngẫu nhiên tại đây cách làm của ta là đi chọn một hoán vị ngẫu nhiên và tính đóng góp của điểm thứ i dựa trên các điểm trước nó trong hoán vị. Sau một số bước giá trị ϕ_i sẽ hội tụ tại giá trị tối ưu. Mô tả:

Algorithm 1 Truncated Monte Carlo Shapley (TMC-Shapley), nguồn [7]

Require: Dữ liệu huấn luyện $D = \{1, \dots, n\}$, thuật toán học \mathcal{A} , hàm đánh giá V

Ensure: Giá trị Shapley value của các điểm dữ liệu huấn luyện: ϕ_1, \dots, ϕ_n

```

1: Khởi tạo  $\phi_i = 0$  cho  $i = 1, \dots, n$  và  $t = 0$ 
2: while Chưa thỏa mãn tiêu chí hội tụ do
3:    $t \leftarrow t + 1$ 
4:    $\pi^t \leftarrow$  Hoán vị ngẫu nhiên các điểm dữ liệu huấn luyện
5:    $v_0^t \leftarrow V(\emptyset, \mathcal{A})$ 
6:   for  $j \in \{1, \dots, n\}$  do
7:     if  $|V(D) - v_{j-1}^t| <$  Ngưỡng sai số của hàm đánh giá then
8:        $v_j^t \leftarrow v_{j-1}^t$ 
9:     else
10:       $v_j^t \leftarrow V(\{\pi^t[1], \dots, \pi^t[j]\}, \mathcal{A})$ 
11:    end if
12:     $\phi_{\pi^t[j]} \leftarrow \frac{t-1}{t} \phi_{\pi^t[j]} + \frac{1}{t} (v_j^t - v_{j-1}^t)$ 
13:  end for
14: end while

```

2.4.1 Công thức dựa trên giá trị kỳ vọng

Giá trị Shapley được định nghĩa như sau:

$$\phi_i = \mathbb{E}_\pi [V(S_\pi^i \cup \{i\}) - V(S_\pi^i)], \quad (2.2)$$

trong đó:

- π là một hoán vị ngẫu nhiên của tập dữ liệu.
- S_π^i là tập con của các điểm dữ liệu xuất hiện trước i trong hoán vị π .
- $V(S)$ là giá trị hiệu suất của mô hình khi được huấn luyện trên tập con S .

2.4. Truncated Monte Carlo Shapley

Giá trị Shapley có thể được xem là kỳ vọng toán học của đóng góp cận biên khi thêm điểm i vào một tập con ngẫu nhiên S_π^i .

2.4.2 Monte Carlo Sampling

Trong Monte Carlo Sampling, giá trị Shapley được xấp xỉ bởi trung bình đóng góp cận biên qua t hoán vị ngẫu nhiên:

$$\hat{\phi}_i = \frac{1}{t} \sum_{k=1}^t (v(S_{\pi_k}^i \cup \{i\}) - v(S_{\pi_k}^i)). \quad (2.3)$$

Theo **Luật số lớn**, nếu các hoán vị π_k được sinh ngẫu nhiên và độc lập:

$$\lim_{t \rightarrow \infty} \hat{\phi}_i = \phi_i.$$

Điều này đảm bảo rằng trung bình $\hat{\phi}_i$ hội tụ về kỳ vọng ϕ_i khi số lượng mẫu t tăng.

2.4.3 Vai trò của Truncation trong TMC-Shapley

Truncation giúp giảm tải tính toán bằng cách dừng sớm khi giá trị $V(S)$ hội tụ trong một ngưỡng $\epsilon > 0$.

Cụ thể:

- Nếu $|V(S \cup \{i\}) - V(S)| < \epsilon$, việc tính toán thêm đóng góp cận biên có thể bị dừng.
- Sai số từ truncation được kiểm soát bởi số lượng hoán vị t , vì ảnh hưởng của một lần truncation giảm khi t tăng:

$$|\hat{\phi}_i^{\text{truncated}} - \phi_i| \leq \epsilon.$$

Khi $t \rightarrow \infty$, sai số này trở nên không đáng kể vì mỗi $V(S)$ bị ảnh hưởng bởi truncation chỉ chiếm một phần rất nhỏ trong tổng trung bình.

2.4.4 Sai số kỳ vọng của Monte Carlo

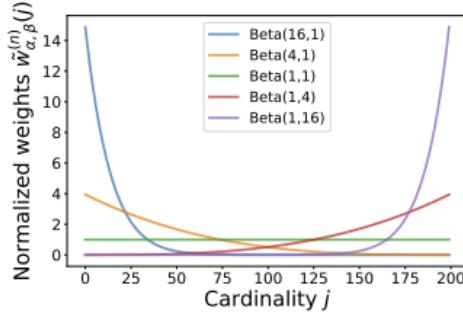
Sai số trong ước lượng Monte Carlo có thể được đo bằng phương sai:

$$\text{Var}(\hat{\phi}_i) = \frac{1}{t} \cdot \text{Var}(v(S_\pi^i \cup \{i\}) - v(S_\pi^i)).$$

Khi t tăng, sai số giảm tỷ lệ nghịch với $\frac{1}{t}$, đảm bảo tính hội tụ:

$$\lim_{t \rightarrow \infty} \text{Var}(\hat{\phi}_i) = 0.$$

2.5. Một số biến thể của Data Shapley



Hình 2.2: Đồ thị phân phối Beta, nguồn [12]

2.5 Một số biến thể của Data Shapley

Một vài phương pháp mới khắc phục một số điểm yếu nhưng nhìn chung tương tự Data Shapley như Beta-Shapley, Class-wise được ra đời.

2.5.1 Beta shapley

Phương pháp của [7] giả định kích thước của mỗi tập con có tác động như nhau trong việc tính đóng góp cận biên. Nhưng ta dễ thấy là nếu một tập huấn luyện càng lớn thì đóng góp biên của một điểm riêng lẻ sẽ giảm đi. Hay ở đây là dù có ném một điểm đi thì hiệu suất cũng không thay đổi nhiều. Điều này làm động lực để thiết kế lại trọng số đóng góp của mỗi tập con. Phương pháp beta shapley [12] thiết kế trọng số dựa trên phân phối beta (hình 2.2). Viết lại công thức tính shapley value:

$$\phi_i = \frac{1}{n} \sum_{j=1}^n \binom{n-1}{j-1} w^{(n)}(j) (V(S \cup \{i\}) - V(S)) \quad (2.4)$$

Điểm khác biệt là ở mỗi về ta thêm trọng số vào công thức. Khi đặt $w^{(n)}(j) = \binom{n-1}{j-1}^{-1}$, ta có thể chuyển sang Data Shapley gốc. Nếu đặt $w^{(n)}(j) = n \cdot \mathbb{1}(j=n)$, ta sẽ nhận được Leave-One-Out (LOO). Công thức $w^{(n)}(j)$ có thể tính như sau:

$$w_{\alpha,\beta}^{(n)}(j) = n \frac{\prod_{k=1}^{j-1} (\beta + k - 1) \prod_{k=1}^{n-j} (\alpha + k - 1)}{\prod_{k=1}^{n-1} (\alpha + \beta + k - 1)}. \quad (2.5)$$

(là công thức dựa theo phân phối beta hình 2.2)

2.5. Một số biến thể của Data Shapley

Beta Shapley cũng có thể sử dụng phương pháp xấp xỉ Monte Carlo kết hợp với phân phối Beta để ước tính giá trị data shapley của một điểm dữ liệu mà không cần phải tính toán tất cả các tập con.

Ưu điểm Dựa vào hàm Beta ta có thể căn chỉnh tham số α, β sao cho phù hợp. Giá trị thường dùng là (16,1), (4,1).

2.5.2 Class-wise shapley

Một khó khăn nữa cho giá trị Shapley truyền thống là chỉ đo lường đóng góp dựa trên hiệu suất tổng thể của mô hình, nhưng điều này không đủ để phân biệt giữa các đóng góp *theo từng lớp* để phân biệt rõ hơn các điểm dữ liệu dựa trên mức độ cải thiện hiệu suất trong lớp của chúng. Phương pháp class-wise shapley [18] được đề xuất nhằm cải thiện hiệu suất theo từng lớp.

Hàm giá trị Class-wise:

CS-Shapley định nghĩa hai loại độ chính xác:

- **Độ chính xác trong lớp:** $V_S(D_{y_i})$, là tỷ lệ dự đoán đúng trên dữ liệu lớp y_i trên tập kiểm thử.
- **Độ chính xác ngoài lớp:** $V_S(D_{-y_i})$, là tỷ lệ dự đoán đúng trên dữ liệu ngoài lớp y_i trên tập kiểm thử.

Hàm tính độ chính xác một lớp được xây dựng bằng cách kết hợp hai loại độ chính xác này:

$$V_{y_i}(S) = V_S(D_{y_i}) \cdot e^{V_S(D_{-y_i})}.$$

Ý nghĩa:

- $V_S(D_{y_i})$ tập trung vào hiệu suất trong lớp.
- $e^{V_S(D_{-y_i})}$ điều chỉnh để đảm bảo không hy sinh hiệu suất ngoài lớp.
- Sự kết hợp này giúp cân bằng giữa việc tối ưu hóa hiệu suất trong lớp và duy trì hiệu suất tổng thể trên toàn bộ tập dữ liệu.

Giá trị CS-Shapley

Giá trị CS-Shapley cho điểm dữ liệu i được định nghĩa:

$$\phi_i^{CS} = \frac{1}{2^{|D_{-y_i}|}} \sum_{S_{-y_i} \subseteq D_{-y_i}} \left(\frac{1}{|D_{y_i}|} \sum_{S_{y_i} \subseteq D_{y_i}} \left(\binom{|D_{y_i}| - 1}{|S_{y_i}|} \right)^{-1} (V(S_{y_i} \cup S_{-y_i} \cup \{i\}) - V(S_{y_i} \cup S_{-y_i})) \right)$$

Trong đó:

- ϕ_i^{CS} : Giá trị Shapley class-wise của điểm dữ liệu i .
- D_{y_i} : Tập hợp dữ liệu trong cùng lớp với i (cùng nhãn y_i).
- D_{-y_i} : Tập hợp dữ liệu ngoài lớp của i (khác nhãn y_i).
- $S_{y_i} \subseteq D_{y_i}$: Tập con dữ liệu cùng lớp y_i .
- $S_{-y_i} \subseteq D_{-y_i}$: Tập con dữ liệu ngoài lớp y_i .
- $V(S)$: Hàm hiệu suất hoặc giá trị mô hình khi sử dụng tập dữ liệu S .
- Công thức này tính giá trị Shapley cho điểm dữ liệu i bằng cách đánh giá đóng góp của nó vào mô hình thông qua việc kết hợp dữ liệu trong lớp và ngoài lớp (cite công thức kề trước)

Như vậy phương pháp CS-Shapley có một số đặc điểm nổi bật như:

- Áp dụng hiệu quả hơn cho bài toán có bộ dữ liệu mất cân bằng.
- Có thể sử dụng Monte-carlo và truncation để tăng tốc độ tính toán.
- Phân biệt giữa đóng góp trong lớp (D_{y_i}) và ngoài lớp (D_{-y_i}).

2.6 KNN-Shapley

Áp dụng tư tưởng của data shapley nhưng tận dụng khả năng của thuật toán phân loại KNN là không cần phải thực hiện lại quá trình huấn luyện mô hình trên mỗi tập con. Giúp cho phương pháp shapley trở nên khả thi về mặt tính toán và cũng đảm bảo độ chính xác tương đối so với việc chạy các mô hình phức tạp khác. Phương pháp KNN-Shapley [8] đê xuất như sau.

Nhắc lại về thuật toán KNN, khi muốn xác định kiểm tra x_{test} có nhãn dự đoán là y_{test} , thuật toán KNN sẽ tìm ra K điểm gần nhất $\{x_{a_1}, x_{a_2}, \dots, x_{a_k}\}$ trong đó a_k là chỉ số huấn luyện thứ K gần nhất. Xác suất để nhãn x_{test} nhận nhãn đúng là:

$$P[x_{\text{test}} \rightarrow y_{\text{test}}] = \frac{1}{K} \sum_{k=1}^K \mathbb{1}_{y_{\alpha_k} = y_{\text{test}}}$$

Một cách tự nhiên để định nghĩa hàm tiện ích (**utility**) của một bộ phân loại KNN là xác suất đúng của nhãn:

$$V(S) = \frac{1}{\min(K, |S|)} \sum_{k=1}^{\min(K, |S|)} \mathbb{1}_{y_{\alpha_k(S)} = y_{\text{test}}}$$

trong đó S là tập con của dữ liệu huấn luyện, $\alpha_k(S)$ là chỉ số của đặc trưng huấn luyện gần thứ k trong tập S .

Công thức tính giá trị Shapley

Dựa trên hàm utility trên, giá trị Shapley cho từng điểm huấn luyện có thể được tính đê quy như sau:

Trường hợp cơ sở cho một điểm test:

$$\phi_{\alpha_N} = \frac{\mathbb{1}_{y_{\alpha_N} = y_{\text{test}}}}{N} \quad (2.6)$$

Trường hợp tổng quát cho một điểm test:

$$\phi_{\alpha_i} = \phi_{\alpha_{i+1}} + \frac{\mathbb{1}_{y_{\alpha_i} = y_{\text{test}}} - \mathbb{1}_{y_{\alpha_{i+1}} = y_{\text{test}}}}{K} \frac{\min(K, i)}{i} \quad (2.7)$$

Trường hợp nhiều điểm kiểm tra

Ở trên là cập nhật giá trị shapley của tập train so với một điểm trong tập test. Kết quả trên có thể dễ dàng mở rộng cho trường hợp nhiều điểm test khác. Khi đó, hàm utility được định nghĩa là:

$$V(S) = \frac{1}{N_{\text{test}}} \sum_{j=1}^{N_{\text{test}}} \frac{1}{\min(K, |S|)} \sum_{k=1}^{\min(K, |S|)} \mathbb{1}_{y_{\alpha_k^{(j)}(S)} = y_{\text{test}, j}} \quad (2.8)$$

2.6. KNN-Shapley

trong đó $\alpha_k^{(j)}(S)$ là chỉ số của k -NN của $x_{\text{test},j}$ trong tập S .

$$\phi_i^{\text{all}} = \frac{1}{N_{\text{test}}} \sum_{j=1}^{N_{\text{test}}} \phi_{ji} \quad (2.9)$$

2.6.1 Tại sao lại có công thức này:

Phần này là giải thích tính đúng đắn của công thức KNN-shapley và cũng liên quan đến chap 3 đề xuất của chúng em.

Từ công thức Data Shapley:

$$\phi_i = C \sum_{S \subseteq D \setminus \{i\}} \frac{V(S \cup \{i\}) - V(S)}{\binom{n-1}{|S|}} \quad (2.10)$$

Với bất kì $i, j \in S$ ta lấy hiệu hai giá trị shapley:

$$\phi_i - \phi_j = \frac{1}{N-1} \sum_{S \subseteq D \setminus \{i,j\}} \frac{V(S \cup \{i\}) - V(S \cup \{j\})}{\binom{N-2}{|S|}}. \quad (2.11)$$

Định lí 1 Ta giả định rằng x_1, \dots, x_N được sắp xếp theo thứ tự tăng dần theo khoảng cách đến x_{test} . Cho bất kì tập con nào $S \subseteq \{1, 2, \dots, N\}$ có kích thước k , chúng ta chia tập con thành hai phần S_1 and S_2 sao cho $S = S_1 \cup S_2$ và $|S_1| + |S_2| = |S| = k$. Với hai điểm liền kề nhau $i, i+1 \in S$ ta có $S_1 \subseteq \{1, \dots, i-1\}$ và $S_2 \subseteq \{i, \dots, N\}$. Chúng ta phân tích $V(S \cup \{i\}) - V(S \cup \{i+1\})$ theo các trường hợp:

TH 1: $|S_1| \geq K$. Trong trường hợp này $i, i+1 > K$ vì vậy $V(S \cup \{i\}) = V(S \cup \{i+1\}) = V(S)$, suy ra:

$$v(S \cup \{i\}) - v(S \cup \{i+1\}) = 0. \quad (2.12)$$

TH 2: $|S_1| < K$. Trong trường hợp này, chúng ta biết rằng $i \leq K$ vì vậy $V(S \cup \{i\}) - V(S)$ có lẽ sẽ không nhất thiết bằng 0. Việc thêm i vào làm K-th neighbor sẽ loại bỏ đi điểm lân cận gần nhất thứ K suy ra:

$$V(S \cup \{i\}) - V(S) = \frac{1}{K} [\mathbb{1}[y_i = y_{\text{test}}] - \mathbb{1}[y_K = y_{\text{test}}]]. \quad (2.13)$$

Tương tự $i+1$:

$$V(S \cup \{i+1\}) - V(S) = \frac{1}{K} [\mathbb{1}[y_{i+1} = y_{\text{test}}] - \mathbb{1}[y_K = y_{\text{test}}]]. \quad (2.14)$$

2.6. KNN-Shapley

Trừ 2 vế ta có:

$$V(S \cup \{i\}) - V(S \cup \{i+1\}) = \frac{\mathbb{1}[y_i = y_{\text{test}}] - \mathbb{1}[y_{i+1} = y_{\text{test}}]}{K}. \quad (2.15)$$

Áp dụng vào (2.11) kết hợp cả hai trường hợp ta sẽ có công thức:

$$\phi_i - \phi_{i+1} = \frac{\mathbb{1}[y_i = y_{\text{test}}] - \mathbb{1}[y_{i+1} = y_{\text{test}}]}{K} \frac{\min(K, i)}{i}. \quad (2.16)$$

Công thức viết mã giả:

$$\phi_{a_i} = \phi_{a_{i+1}} + \frac{\mathbb{1}[y_i = y_{\text{test}}] - \mathbb{1}[y_{i+1} = y_{\text{test}}]}{K} \frac{\min(K, i)}{i}. \quad (2.17)$$

Giá trị khởi tạo ϕ_{a_N} :

$$\phi_{a_N} = \frac{\mathbb{1}[y_N = y_{\text{test}}]}{N}. \quad (2.18)$$

Mã giả:

Algorithm 2 Thuật toán chính xác để tính toán giá trị Shapley (SV) cho một bộ phân loại KNN.

Đầu vào: Dữ liệu huấn luyện $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$, dữ liệu kiểm tra $\mathcal{D}_{\text{test}} = \{(x_{\text{test},j}, y_{\text{test},j})\}_{j=1}^{N_{\text{test}}}$

Đầu ra: Giá trị Shapley $\{s_i\}_{i=1}^N$

- 1: **for** $j \leftarrow 1$ to N_{test} **do**
 - 2: $(\alpha_1, \dots, \alpha_N) \leftarrow$ Chỉ số của các điểm dữ liệu huấn luyện được sắp xếp tăng dần theo khoảng cách $d(\cdot, x_{\text{test},j})$
 - 3: $s_{j,\alpha_N} \leftarrow \frac{1}{N} \mathbb{1}[y_{\alpha_N} = y_{\text{test},j}]$
 - 4: **for** $i \leftarrow N - 1$ to 1 **do**
 - 5: $s_{j,\alpha_i} \leftarrow s_{j,\alpha_{i+1}} + \frac{1}{K} \left(\mathbb{1}[y_{\alpha_i} = y_{\text{test},j}] - \mathbb{1}[y_{\alpha_{i+1}} = y_{\text{test},j}] \right) \frac{\min(K, i)}{i}$
 - 6: **end for**
 - 7: **end for**
 - 8: **for** $i \leftarrow 1$ to N **do**
 - 9: $s_i \leftarrow \frac{1}{N_{\text{test}}} \sum_{j=1}^{N_{\text{test}}} s_{j,i}$
 - 10: **end for**
-

2.7 LAVA

LAVA (Hình 1.5) là một thuật toán không phụ thuộc vào mô hình huấn luyện cụ thể, thay vào đó sử dụng optimal transport (OT), được đề xuất bởi [10].

Optimal transport là một phương pháp có nền tảng lý thuyết vững chắc, được ứng dụng rộng rãi trong các lĩnh vực như thị giác máy tính, mạng tạo sinh, và nhiều lĩnh vực khác. Với khả năng mạnh mẽ trong việc so sánh các phân phối, OT giúp LAVA trở thành một lựa chọn lý tưởng cho các bài toán yêu cầu thích nghi miền hoặc căn chỉnh phân phối.

Nhờ sử dụng chi phí OT, LAVA không bị ràng buộc bởi một mô hình cụ thể hoặc một giá trị tiện ích cố định, mang lại sự linh hoạt cao, đặc biệt trong các thiết lập **không giám sát** hoặc **bán giám sát**. Ngày nay, có nhiều thư viện tối ưu hóa hỗ trợ OT, chẳng hạn như GeomLoss và POT, giúp việc triển khai LAVA trở nên dễ dàng hơn.

2.7.1 Kiến thức về Optimal transport

Optimal transport là một phương pháp nổi bật dùng để đo lường sự khác biệt giữa hai phân phối xác suất [2]. So với các thước đo khác như Kullback-Leibler hoặc Maximum Mean Discrepancies khoảng cách OT có các tính chất vượt trội hơn nhờ định nghĩa toán học chặt chẽ. Trong bài toán này OT giống một metric đo lường khoảng cách, có thể tính toán được các mẫu hữu hạn và khả thi về thời gian tính toán. (xem phần 5.1)

Định nghĩa bài toán OT trong thực tế: Bài toán optimal transport có thể được mô tả qua ngữ cảnh bài toán vận chuyển hàng hóa như sau:

- **Nguồn cung và nhu cầu:** Giả sử có một số lượng nhà kho (nguồn cung) với một lượng hàng hóa cố định và một số lượng cửa hàng hoặc khách hàng (điểm nhận) với nhu cầu hàng hóa cụ thể.
- **Chi phí vận chuyển:** Mỗi đơn vị hàng hóa khi được vận chuyển từ một nhà kho đến một cửa hàng có một chi phí nhất định. Chi phí này có thể phụ thuộc vào khoảng cách địa lý hoặc các yếu tố khác như giao thông hoặc chi phí nhiên liệu.
- **Mục tiêu:** Tìm cách phân phối hàng hóa từ các nhà kho đến các cửa hàng sao cho tổng chi phí vận chuyển là thấp nhất, đồng thời đảm bảo tất cả nguồn cung được sử dụng hết và tất cả nhu cầu được đáp ứng. (*tựa như bài toán luồng trong cấu trúc dữ liệu và giải thuật*)

2.7.2 Định nghĩa bài toán về mặt toán học

Giả sử ta có hai phân phối xác suất μ và ν trên không gian X , và ta muốn tìm một cách “vận chuyển” khôi lượng từ μ đến ν sao cho chi phí vận chuyển là nhỏ nhất. Chi phí này được đo bằng một hàm chi phí $c(x, y)$, với $x \in X$ và $y \in Y$ là các điểm trong không gian đầu và không gian đích.

Bài toán tối ưu vận chuyển có thể được mô tả như sau:

$$\min_{\gamma \in \Pi(\mu, \nu)} \int_{X \times Y} c(x, y) d\gamma(x, y),$$

trong đó:

- $\Pi(\mu, \nu)$ là tập hợp các phân phối chung (couplings) có các biên (marginals) là μ và ν . tức là $\gamma \in \Pi$ phải thỏa mãn:

$$\int_X d\gamma(x, y) = \mu(x) \quad \text{và} \quad \int_Y d\gamma(x, y) = \nu(y)$$

Với $\mu(x)$ và $\nu(y)$ là mật độ xác suất của μ và ν tương ứng.

- $c(x, y)$ là hàm chi phí vận chuyển giữa điểm x và điểm y .
- $\gamma(x, y)$ là phân phối chung giữa μ và ν .

2.7.3 Định nghĩa trên phân phối rời rạc

Giả sử ta có hai phân phối rời rạc $\mathbf{p} = (p_1, p_2, \dots, p_n)$ và $\mathbf{q} = (q_1, q_2, \dots, q_m)$, đại diện cho các phân phối xác suất trên các điểm rời rạc x_1, x_2, \dots, x_n và y_1, y_2, \dots, y_m , tương ứng. Mục tiêu là tìm phân phối γ (các phép vận chuyển từ x_i sang y_j) sao cho chi phí vận chuyển là tối thiểu.

Bài toán tối ưu vận chuyển cho phân phối rời rạc có thể được mô tả như sau:

$$\mathcal{L}(\gamma) = \min_{\gamma} \sum_{i=1}^n \sum_{j=1}^m c(x_i, y_j) \gamma_{ij} \tag{2.19}$$

Trong đó:

- γ_{ij} là lượng khôi lượng vận chuyển từ điểm x_i đến điểm y_j .
- $c(x_i, y_j)$ là chi phí vận chuyển từ x_i đến y_j .

- Các ràng buộc:

$$\sum_{j=1}^m \gamma_{ij} = p_i \quad \text{và} \quad \sum_{i=1}^n \gamma_{ij} = q_j$$

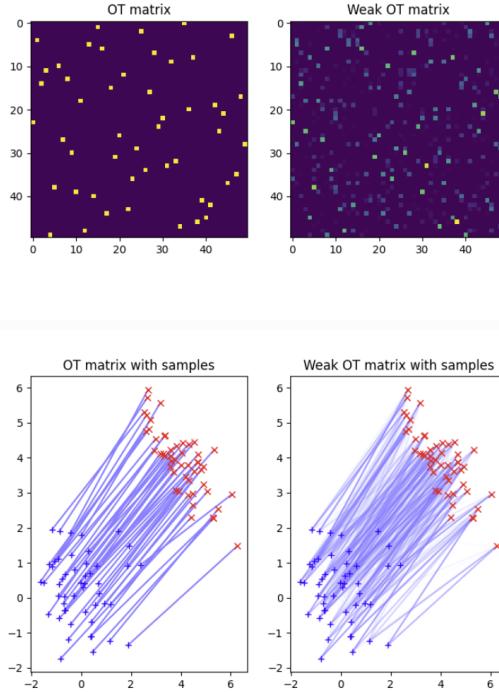
Tức là, tổng khối lượng vận chuyển từ mỗi điểm x_i phải bằng p_i , và tổng khối lượng vận chuyển đến mỗi điểm y_j phải bằng q_j .

Để giúp bài toán hội tụ ta thêm một hàm entropic trên γ , bài toán tối ưu vận chuyển với điều chuẩn entropic có thể được mô tả như sau (hình 2.3):

$$\min_{\gamma} \sum_{i=1}^n \sum_{j=1}^m c(x_i, y_j) \gamma_{ij} + \varepsilon \sum_{i=1}^n \sum_{j=1}^m \gamma_{ij} \ln(\gamma_{ij}) \quad (2.20)$$

Trong đó:

- γ_{ij} là lượng khối lượng vận chuyển từ x_i đến y_j .
- $c(x_i, y_j)$ là chi phí vận chuyển từ x_i đến y_j .
- ε là hệ số điều chuẩn entropic, điều chỉnh mức độ điều chuẩn trong bài toán.
- p_i và q_j là các yếu tố xác suất của phân phối **p** và **q**, tương ứng.
- Điều chuẩn entropic $\gamma_{ij} \ln \left(\frac{\gamma_{ij}}{p_i q_j} \right)$ giúp giảm thiểu sự không chắc chắn trong phân phối γ và làm cho bài toán tối ưu dễ giải hơn.



Hình 2.3: Mô tả ví dụ sự khác nhau khi sử dụng hàm entropic vào bên phải, và không sử dụng bên trái. Ta thấy rằng bên phải từ một điểm nguồn (màu xanh) có thể nối đến nhiều điểm đích (màu đỏ) trong khi hình bên trái nghiêm ngặt hơn.

2.7.4 Phương pháp Sinkhorn-Knopp

Phương pháp **Sinkhorn-Knopp** thực chất là một phương pháp lặp lại để giải bài toán tối ưu vận chuyển với điều kiện chuẩn entropic thông qua các phép chuẩn hóa theo hai chiều. Các bước chuẩn hóa này được áp dụng nhằm duy trì các điều kiện biên, tức là tổng khối lượng vận chuyển từ mỗi điểm x_i phải bằng p_i và tổng khối lượng đến mỗi điểm y_j phải bằng q_j . Để hiểu rõ về sinkhorn-knopp ta cần phải biết dạng dual form và cách duy trì điều kiện biên.

Bài toán Dual

Bài toán được đề xuất ban đầu (công thức 2.19) được gọi là primal form nó đi tối ưu hóa trên γ , dual form là phương pháp đẩy ma trận γ thành hai vector α, β . Thay vì tối ưu hóa primal form trên γ ta đi tối ưu hóa α và β sử dụng chuyển hóa lagrange như sau:

$$\mathcal{L}(\gamma, \alpha, \beta) = \sum_{i,j} \gamma_{ij} c_{ij} + \varepsilon \sum_{i,j} \gamma_{ij} (\ln \gamma_{ij}) - \sum_i \alpha_i (\sum_j \gamma_{ij} - p_i) - \sum_j \beta_j (\sum_i \gamma_{ij} - q_j) - \varepsilon (\sum_{i,j} \gamma_{ij} - 1). \quad (2.21)$$

Trong đó:

$$\gamma \mathbf{1}_n = p, \quad \gamma^T \mathbf{1}_m = q \quad (\text{vector n chiều}), \quad \gamma_{ij} \geq 0$$

Lấy đạo hàm của \mathcal{L} cho γ_{ij} và đặt nó bằng 0:

$$\frac{\partial \mathcal{L}}{\partial \gamma_{ij}} = c_{ij} + \varepsilon(\ln \gamma_{ij} + 1) - \alpha_i - \beta_j - \varepsilon = 0. \quad (2.22)$$

Giải cho γ_{ij} :

$$\ln \gamma_{ij}^* = - \left(\frac{c_{ij} - \alpha_i - \beta_j}{\varepsilon} \right). \quad (2.23)$$

(2.24)

$$\gamma_{ij}^* = \exp \left(\frac{\alpha_i}{\varepsilon} \right) \exp \left(-\frac{c_{ij}}{\varepsilon} \right) \exp \left(\frac{\beta_j}{\varepsilon} \right)$$

Đặt

$$K_{ij} = \exp \left(-\frac{c_{ij}}{\varepsilon} \right), \quad u_i = \exp \left(\frac{\alpha_i}{\varepsilon} \right), \quad v_j = \exp \left(\frac{\beta_j}{\varepsilon} \right);$$

$$\gamma_{ij} = u_i K_{ij} v_j, \quad (2.25)$$

$$\gamma = \text{diag}(u) K \text{diag}(v) \quad (2.26)$$

Ta thấy

$$\gamma \mathbb{1} = p \text{ và } \gamma^T \mathbb{1} = q, \text{ vì vậy}$$

$$\gamma \mathbb{1} = [\text{diag}(u) K \text{diag}(v)] \mathbb{1} = p, \quad (2.27)$$

$$\gamma^T \mathbb{1} = [\text{diag}(v) K^T \text{diag}(u)] \mathbb{1} = q \quad (2.28)$$

$$\gamma \mathbb{1} = u \odot (Kv) = p$$

$$\gamma^T \mathbb{1} = v \odot (K^T u) = q$$

$$u = p \oslash (Kv), \quad v = q \oslash (K^T u)$$

Ký hiệu:

- \odot : Phép nhân từng phần tử (element-wise product).
- \oslash : Phép chia từng phần tử (element-wise division).

Tại vòng lặp thứ $(i + 1)$, chúng ta sử dụng $v^{(i)}$ để cập nhật $u^{(i)}$ thành $u^{(i+1)}$, dùng $u^{(i+1)}$ cập nhật $v^{(i)}$ thành $v^{(i+1)}$.

$$u^{(i+1)} = \frac{p}{Kv^{(i)}} \quad \rightarrow \quad v^{(i+1)} = \frac{q}{K^T u^{(i+1)}}.$$

Mã giả:

Algorithm 3 Thuật toán Sinkhorn-Knopp

Require: Ma trận chi phí $C = [c(x_i, y_j)]$, các vector biên $p = [p_i]$, $q = [q_j]$, tham số $\varepsilon > 0$, ngưỡng hội tụ $\delta > 0$

Ensure: Ma trận vận chuyển tối ưu $\gamma = [\gamma_{ij}]$

- 1: Khởi tạo $K_{ij} \leftarrow \exp\left(-\frac{c(x_i, y_j)}{\varepsilon}\right)$
 - 2: Khởi tạo $u_i \leftarrow 1$ với mọi i , $v_j \leftarrow 1$ với mọi j
 - 3: **repeat**
 - 4: **Cập nhật u :**
 - 5: **for** $i = 1$ đến n **do**
 - 6: $u_i \leftarrow \frac{p_i}{\sum_{j=1}^m K_{ij} v_j}$
 - 7: **end for**
 - 8: **Cập nhật v :**
 - 9: **for** $j = 1$ đến m **do**
 - 10: $v_j \leftarrow \frac{q_j}{\sum_{i=1}^n K_{ij} u_i}$
 - 11: **end for**
 - 12: **Cập nhật ma trận γ :**
 - 13: **for** $i = 1$ đến n **do**
 - 14: **for** $j = 1$ đến m **do**
 - 15: $\gamma_{ij} \leftarrow u_i K_{ij} v_j$
 - 16: **end for**
 - 17: **end for**
 - 18: **until** Hội tụ: $\|\sum_{j=1}^m \gamma_{ij} - p_i\| < \delta$ và $\|\sum_{i=1}^n \gamma_{ij} - q_j\| < \delta$
-

Giải thích mã giả: Để giải bài toán dual này, phương pháp **Sinkhorn-Knopp** sử dụng các phép chuẩn hóa theo hai chiều (hàng và cột). Mục tiêu của các phép chuẩn hóa này là duy trì các điều kiện biên, tức là tổng khối lượng vận chuyển từ mỗi điểm x_i phải bằng p_i và tổng khối lượng đến mỗi điểm y_j phải bằng q_j .

Bước 1: Cập nhật u_i (chuẩn hóa theo hàng)

Cập nhật u_i sao cho tổng khối lượng từ mỗi điểm x_i phải bằng p_i . Cập nhật này được thực hiện theo công thức:

$$u_i \leftarrow \frac{p_i}{\sum_{j=1}^m K_{ij} v_j}$$

Trong đó, $K_{ij} = \exp\left(-\frac{c(x_i, y_j)}{\epsilon}\right)$ là ma trận chi phí đã điều chuẩn.

Bước 2: Cập nhật v_j (chuẩn hóa theo cột)

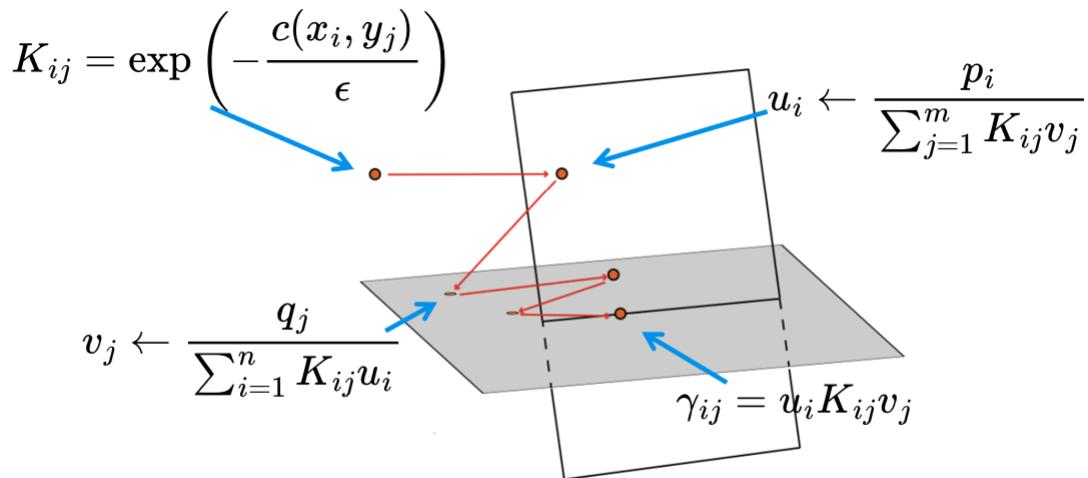
Cập nhật v_j sao cho tổng khối lượng đến mỗi điểm y_j phải bằng q_j . Điều này được thực hiện qua phép chuẩn hóa sau:

$$v_j \leftarrow \frac{q_j}{\sum_{i=1}^n K_{ij} u_i}$$

Bước 3: Cập nhật ma trận γ_{ij}

Cuối cùng, ma trận vận chuyển γ_{ij} được cập nhật theo công thức:

$$\gamma_{ij} = u_i K_{ij} v_j$$



Hình 2.4: Minh họa cách hoạt động của thuật toán sinkhorn-knopp, ta chuẩn hóa giá trị u, v để biến K đến miền phân phối chung $\Pi(\mu, v)$ để tìm γ

Phương pháp Sinkhorn-Knopp hội tụ sau một số vòng lặp, và khi đạt được sự hội tụ, các giá trị u_i và v_j đạt được tối đa của hàm mục tiêu trong bài toán dual, đồng thời các điều kiện biên được duy trì. Đây là lý do tại sao phương pháp Sinkhorn-Knopp có thể tìm được nghiệm tối ưu của bài toán tối ưu vận chuyển, mặc dù nó sử dụng các phép chuẩn hóa thay vì giải trực tiếp bài toán tối ưu.

2.8 Phân cấp trong Optimal Transport

Kỹ thuật này được gọi là hierarchically-defined wasserstein distance [1]. Optimal transport (OT) thường được tính toán dựa trên không gian đặc trưng (feature space) và sử dụng khoảng cách Euclidean hoặc Cosine để tính khoảng cách. Tuy nhiên, khi làm việc với dữ liệu có nhãn, OT sẽ không bắt giữ được thông tin giữa nhãn và cấu trúc quan trọng bên trong dữ liệu (xem hình 2.5). Do đó, khoảng cách nhãn (*label distance*) là điều cần thiết để định nghĩa mối quan hệ giữa sự phân bố các đặc trưng có điều kiện trên các nhãn khác nhau. Ví dụ, [24] sử dụng khoảng cách OT phân cấp để đo sự tương đồng giữa tài liệu, với khoảng cách cấp độ bên trong cho các chủ đề và cấp độ bên ngoài cho tài liệu. [4] thì áp dụng khoảng cách Wasserstein lồng nhau như một hàm tổn thất, đặc biệt phù hợp để so sánh hình ảnh hơn metric L_2 thông thường áp dụng vào generative models. Hai phương pháp trên là ý tưởng để tạo nên OT phân cấp được áp dụng cho bài toán của chúng ta.

2.8.1 Phân phối có điều kiện

Để định nghĩa mới quan hệ giữa đặc trưng (*feature*) và nhãn (*label*), cấu trúc phân phối có điều kiện được định nghĩa như sau:

$$\mu_t(x | y) := \frac{\mu_t(x) \mathbb{1}[f_t(x) = y]}{\int \mu_t(x) \mathbb{1}[f_t(x) = y] dx}$$

và

$$\mu_v(x | y) := \frac{\mu_v(x) \mathbb{1}[f_v(x) = y]}{\int \mu_v(x) \mathbb{1}[f_v(x) = y] dx}$$

trong đó:

- $\mu_v(x)$ là phân phối feature trong dataset v ,
- $f_v(x)$ là hàm dự đoán (*predictor label*),

2.8.2 Label Distance trong hàm Chi phí

Để tính khoảng cách OT giữa các nhãn, hàm chi phí giữa hai cặp (feature, label) (x_t, y_t) và (x_v, y_v) được định nghĩa như sau:

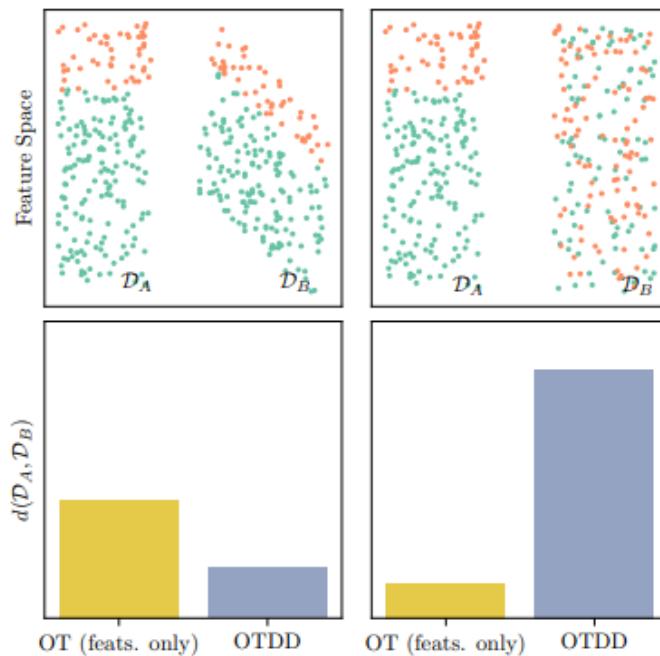
$$C((x_t, y_t), (x_v, y_v)) := d(x_t, x_v) + cOT_d(\mu_t(\cdot | y_t), \mu_v(\cdot | y_v)),$$

trong đó:

2.8. Phân cấp trong Optimal Transport

- $d(x_t, x_v)$ là một metric (ví dụ: khoảng cách Euclidean) trong không gian đặc trưng,
- OT_d (hay còn gọi là khoảng cách Wasserstein) khoảng cách giữa các phân phối có điều kiện $\mu_t(\cdot | y_t)$ và $\mu_v(\cdot | y_v)$,
- $c > 0$ là một hệ số trọng số kiểm soát ảnh hưởng của khoảng cách nhãn.

Khoảng cách OT phân cấp không chỉ cải thiện khả năng biểu diễn giữa các tập dữ liệu mà còn giúp giải quyết các bài toán phức tạp như phân loại đa nhãn hoặc các tập dữ liệu có cấu trúc phức tạp. Bằng cách kết hợp cả metric ở không gian đặc trưng và các khoảng cách có điều kiện, phương pháp này mang lại sự phân tích chi tiết hơn giữa các tập dữ liệu nguồn và mục tiêu.



Hình 2.5: Sự quan trọng của khoảng cách nhãn: Các cặp dữ liệu thứ hai gần nhau hơn trong khoảng cách OT thông thường (không dựa trên nhãn) (màu vàng), trong khi điều ngược lại mới là đúng với khoảng cách có ý thức nhãn (màu xám). Nguồn ảnh [1]

2.9 Đánh giá điểm dữ liệu thông qua Gradient được hiệu chỉnh

2.9.1 Đánh giá điểm dữ liệu dựa trên Gradient

Khoảng cách OT (Optimal Transport) được biết đến là ít nhạy cảm với các biến đổi nhỏ nhưng lại nhạy cảm với các sai lệch lớn. Điều này phù hợp tự nhiên để phát hiện các điểm dữ liệu bất thường — bỏ qua các biến đổi nhỏ giữa các dữ liệu sạch nhưng phản ứng mạnh với các điểm nằm ngoài (*outliers*).

Để đo lường mức độ đóng góp của từng điểm dữ liệu vào khoảng cách OT, thuật toán sử dụng gradient của khoảng cách OT trên xác suất của từng điểm. Gradient này cho phép dự đoán cách mà khoảng cách OT thay đổi khi thêm hoặc loại bỏ một điểm dữ liệu mà không cần tính toán lại toàn bộ.

Ngoài ra, gradient cũng cung cấp thông tin có hướng, giúp phân biệt giữa các điểm dữ liệu có ảnh hưởng tích cực hoặc tiêu cực. Điều này cho phép xếp hạng các điểm dựa trên giá trị gradient.

2.9.2 Gradient được hiệu chỉnh (Calibrated Gradients)

Gradient được hiệu chỉnh (*calibrated gradients*) dự đoán cách mà khoảng cách OT thay đổi khi khôi xác suất được chuyển hướng tới một điểm dữ liệu cụ thể. Đóng góp này có thể mang giá trị tích cực hoặc tiêu cực:

- Gradient tích cực ám chỉ rằng việc chuyển thêm khôi xác suất sẽ làm tăng khoảng cách OT.
- Gradient tiêu cực ám chỉ rằng việc chuyển thêm khôi xác suất sẽ làm giảm khoảng cách OT.

Nếu mục tiêu là khớp phân phối của tập huấn luyện với tập kiểm tra, thì việc loại bỏ các điểm dữ liệu có gradient tích cực lớn và tăng các điểm có gradient tiêu cực lớn sẽ giúp giảm khoảng cách OT giữa hai tập.

2.9. Đánh giá điểm dữ liệu thông qua Gradient được hiệu chỉnh

2.9.3 Công thức Gradient được hiệu chỉnh

Bài toán dual của Optimal Transport được viết lại như sau:

$$\text{OT}(\mu_t, \mu_v) = G = \max_{f,g} \left\{ \sum_i f_i \mu_t(z_i) + \sum_j g_j \mu_v(z'_j) \mid f_i + g_j \leq c_{ij}, \forall i, j \right\}.$$

Trong đó:

- f^* và g^* là nghiệm tối ưu của bài toán dual
- Giả sử γ^* và (f^*, g^*) là các nghiệm tối ưu tương ứng của bài toán gốc và bài toán đối ngẫu. Khi đó, định lý đối ngẫu mạnh chỉ ra rằng:

$$\text{OT}(\gamma^*(\mu_t, \mu_v)) = \text{OT}(f^*, g^*),$$

Giải thích: Từ công thức mục 2.19 và 2.24:

$$\gamma_{ij}^* >= 0 \Rightarrow c_{ij} >= u_i + v_j \quad (2.29)$$

Thay $c_{ij} = u_i + v_j$ có thể thấy $L = G$ (f, g tương ứng với u và v trong thuật toán sinkhorn). Như vậy $L >= G$ thay vì tìm tối thiểu trên L ta có thể tìm giá trị tối ưu trên G .

Bài toán đối ngẫu này đôi khi được gọi là bài toán của người vận chuyển (Shipper's problem). Giả sử mục tiêu ban đầu của chúng ta là vận chuyển p đến q . Một người vận chuyển tiếp cận chúng ta và đồng ý vận chuyển p đến q cho chúng ta, và chúng ta chỉ cần trả chi phí bốc dỡ. Người vận chuyển cho biết chi phí bốc dỡ một đơn vị hàng tại μ_t là u_i và tại μ_v là v_j .

Để chấp nhận thỏa thuận, có vẻ hợp lý khi yêu cầu rằng:

$$u_i + v_j \leq c_{ij}$$

(tức là chi phí chúng ta trả cho người vận chuyển để bốc và dỡ hàng phải thấp hơn hoặc bằng chi phí nếu chúng ta tự vận chuyển).

Người vận chuyển sẽ cố gắng tối đa hóa lợi nhuận (tổng giá bốc và dỡ hàng mà họ có thể tính cho chúng ta) với điều kiện chúng ta chấp nhận thỏa thuận. Vì vậy, người vận chuyển sẽ cố gắng giải bài toán đối ngẫu để quyết định chi phí bốc/dỡ.

2.9. Đánh giá điểm dữ liệu thông qua Gradient được hiệu chỉnh

Tính chất đối ngẫu yếu (weak duality) cho chúng ta biết rằng điều này sẽ luôn là một thỏa thuận có lợi (tức là tổng số tiền chúng ta trả cho người vận chuyển sẽ ít hơn so với việc tự vận chuyển). Trong trường hợp này, tính chất đối ngẫu mạnh (strong duality) sẽ cho biết rằng một người vận chuyển khéo léo (người giải được bài toán đối ngẫu) có thể khiến chúng ta phải trả số tiền tương đương với chi phí tự vận chuyển.

Tính chất nhạy cảm của nghiệm dual

Trong tối ưu hóa lồi, nghiệm dual f^* và g^* thể hiện mức độ “nhạy cảm” của hàm mục tiêu khi thay đổi **các trọng số xác suất** μ_t và μ_v .

Cụ thể:

- Khi tăng nhẹ trọng số $\mu_t(z_i)$ tại một điểm z_i , hàm mục tiêu OT sẽ thay đổi với tốc độ bằng f_i^* .
- Khi tăng nhẹ trọng số $\mu_v(z'_j)$ tại một điểm z'_j , hàm mục tiêu OT sẽ thay đổi với tốc độ bằng g_j^* .

Điều này dẫn đến:

$$\frac{\partial \text{OT}}{\partial \mu_t(z_i)} = f_i^*, \quad \frac{\partial \text{OT}}{\partial \mu_v(z'_j)} = g_j^*.$$

Gradient của khoảng cách OT trên mỗi điểm dữ liệu được hiệu chỉnh để thỏa ràng buộc giữa tất cả các điểm trong tập dữ liệu được tính như sau:

$$\frac{\partial \text{OT}(\mu_t, \mu_v)}{\partial \mu_t(x_i)} = f_i^* - \sum_{j \in \{1, \dots, N\} \setminus i} \frac{f_j^*}{N-1}, \quad \frac{\partial \text{OT}(\mu_t, \mu_v)}{\partial \mu_v(x_j)} = g_j^* - \sum_{i \in \{1, \dots, M\} \setminus j} \frac{g_i^*}{M-1}.$$

Trong đó:

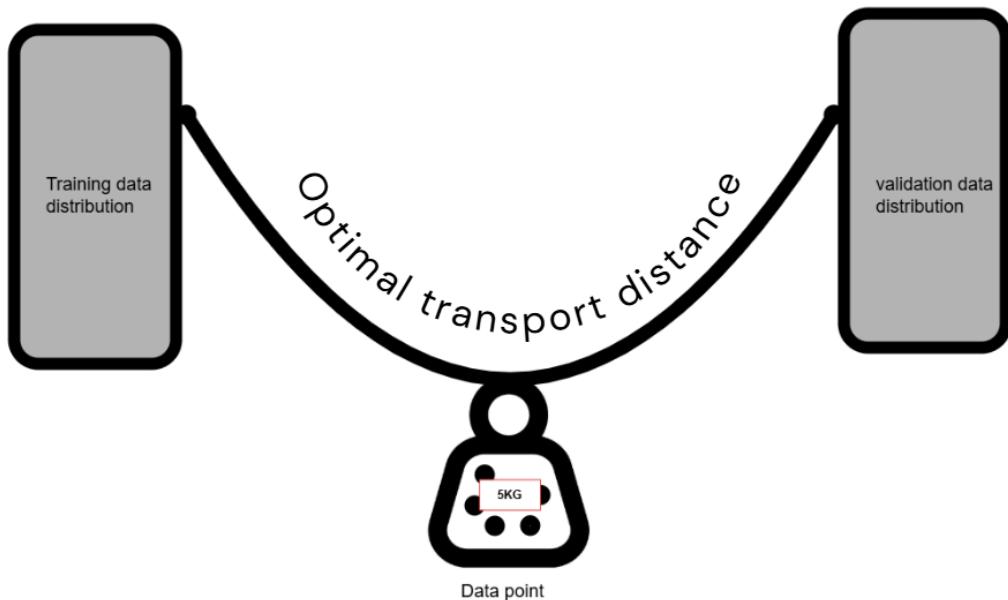
- f_i^* và g_i^* là các biến dual tối ưu tương ứng với các điểm dữ liệu x_i .
- Gradient này đo lường tốc độ thay đổi của khoảng cách OT theo hướng đảm bảo rằng tổng khối xác suất của tất cả các điểm dữ liệu vẫn bằng 1.

Như vậy qua việc chứng minh mối quan hệ giữa công thức Lagrange và dual form ta có thay vì tối ưu γ^* ta đi tối ưu f^* và g^* bằng sinkhorn và cùng lúc đó sử dụng giá trị này để tính calibrated gradient (gradient hiệu chỉnh).

2.9. Đánh giá điểm dữ liệu thông qua Gradient được hiệu chỉnh

Các giá trị tương ứng trong data shapley chính là các gradient của OT trên từng điểm dữ liệu, nhưng khác là các giá trị âm biểu thị tính tích cực của điểm dữ liệu vào đóng góp chung trái với giá trị data shapley.

Gradient được hiệu chỉnh là một công cụ quan trọng để phát hiện và xóa bỏ các điểm dữ liệu bất thường hoặc không liên quan, giúp cải thiện hiệu quả học máy trên nhiều ứng dụng thực tế.



Hình 2.6: Minh họa tưởng tượng về sự thay đổi gradient của một điểm sẽ làm cho khoảng cách optimal transport giữa hai phân phối gần lại hay xích ra. Ví dụ thay đổi nó như một cái cân (5kg) kéo khoảng cách xích lại hoặc giãn ra nếu như ta thay đổi trọng lượng của chiếc cân.

Chương 3

Đề xuất cải tiến

3.1 Cải tiến KNN-shapley

Với sự ra đời của KNN-Shapley, một công cụ thực tế cho phép tính toán các giá trị Shapley mà không cần huấn luyện lại mô hình đắt đỏ, các phương pháp dựa trên Shapley đã trở nên khả thi và được áp dụng rộng rãi. KNN-Shapley tận dụng bộ phân loại K-Nearest Neighbors làm một mô hình thay thế cho mô hình học gốc, tính toán đê quy giá trị Shapley cho từng mẫu huấn luyện.

Dù có tiềm năng, KNN-Shapley và các biến thể của nó vẫn đối mặt với vấn đề lạm phát giá trị. Một số giá trị Shapley trong KNN-Shapley mang giá trị dương nhưng có thể gây tiêu cực (làm giảm hiệu suất mô hình).

Giả định chúng ta có một mô hình KNN-classifier. Việc chúng ta làm là tính giá trị shapley trên toàn bộ subset trong tập train. Theo công thức từ KNN-shapley ta dùng một điểm test để đánh giá giá trị shapley.

Có một nhược điểm từ KNN-Shapley có thể mắc phải:

☞ Trên các subset mà chỉ toàn những điểm dữ liệu xa so với điểm test mà chúng ta đang xét, đóng góp của điểm test vào các điểm train trong subset có thể không đúng và làm giá trị Shapley bị lạm phát hoặc sai lệch.

Phương pháp đề xuất là đặt một ngưỡng T để ngăn chặn việc tính các tập hợp con mắc phải điều trên. Góp phần cải thiện kết quả quanh ngưỡng 0 (âm được xem là tiêu cực, dương được xem là tích cực với giá trị data shapley) và giảm đáng kể thời gian tính toán.

Công thức viết mã giả:

$$\phi_{a_N} = \phi_{a_{N-1}} = \dots = \phi_{a_{N-T+1}} = 0 \quad (3.1)$$

3.2. Cải tiến LAVA

$$\phi_{a_{N-T}} = \frac{1[y_{N-T} = y_{\text{test}}]}{N - T} \quad (3.2)$$

Công thức còn lại như cũ (2.17):

$$\phi_{a_i} = \phi_{a_{i+1}} + \frac{1[y_i = y_{\text{test}}] - 1[y_{i+1} = y_{\text{test}}]}{K} \frac{\min(K, i)}{i} \quad (3.3)$$

Như vậy so với phương pháp KNN-shapley thì chúng ta sẽ thêm một tham số T nữa, việc chọn tham số T có thể tùy chỉnh tương tự beta shapley ví dụ T bằng 2^*K , $N/2$, $N-2K$.

3.2 Cải tiến LAVA

Mối tương quan giữa cost label và cost feature trong Optimal Transport

Để thực nghiệm LAVA, chúng ta cần ba thành phần chính: hai phân phối rời rạc p, q (được mô tả trong phần trước) và ma trận chi phí *cost_matrix*.

Thông thường, p và q được giả định là phân phối chuẩn, biểu thị rằng khối lượng ban đầu của các điểm dữ liệu là đồng đều. Ma trận *cost_matrix* được định nghĩa như sau (2.7):

$$C((x_t, y_t), (x_v, y_v)) = d(x_t, x_v) + cOT_d(\mu_t(\cdot|y_t), \mu_v(\cdot|y_v)),$$

Công thức này có thể linh hoạt gán trọng số để điều chỉnh tầm quan trọng giữa khoảng cách nhãn (label distance) và khoảng cách đặc trưng (feature distance). Khi thực nghiệm, công thức được viết lại như sau:

$$\text{cost_matrix} = \text{feature_distance} \cdot \alpha + \text{label_distance} \cdot \beta$$

Trong đó:

- **Cân bằng giữa feature và label distance:** Để đạt hiệu suất tốt, cần cân bằng khoảng cách giữa đặc trưng và nhãn. Qua thực nghiệm, tỷ lệ 1:1 được chứng minh là hiệu quả. Tuy nhiên, khi mối tương quan giữa *feature_distance* và *label_distance* chưa tương đồng [6], chúng em đã thực hiện *normalize* để cân bằng đóng góp của chúng. Quá trình *normalize* sử dụng hàm *tanh* để đưa về khoảng $[-1, 1]$ và sau đó chuyển về $[0, 1]$ bằng công thức $(\tanh + 1)/2$.
- **Cải tiến với dữ liệu mất cân bằng:** Trong LAVA, giá trị p, q thường được giả định là phân phối chuẩn ($\frac{1}{\text{train_size}}$ và $\frac{1}{\text{test_size}}$). Tuy nhiên, với tập dữ liệu không cân bằng, có thể gán trọng số khác nhau cho các lớp (class).

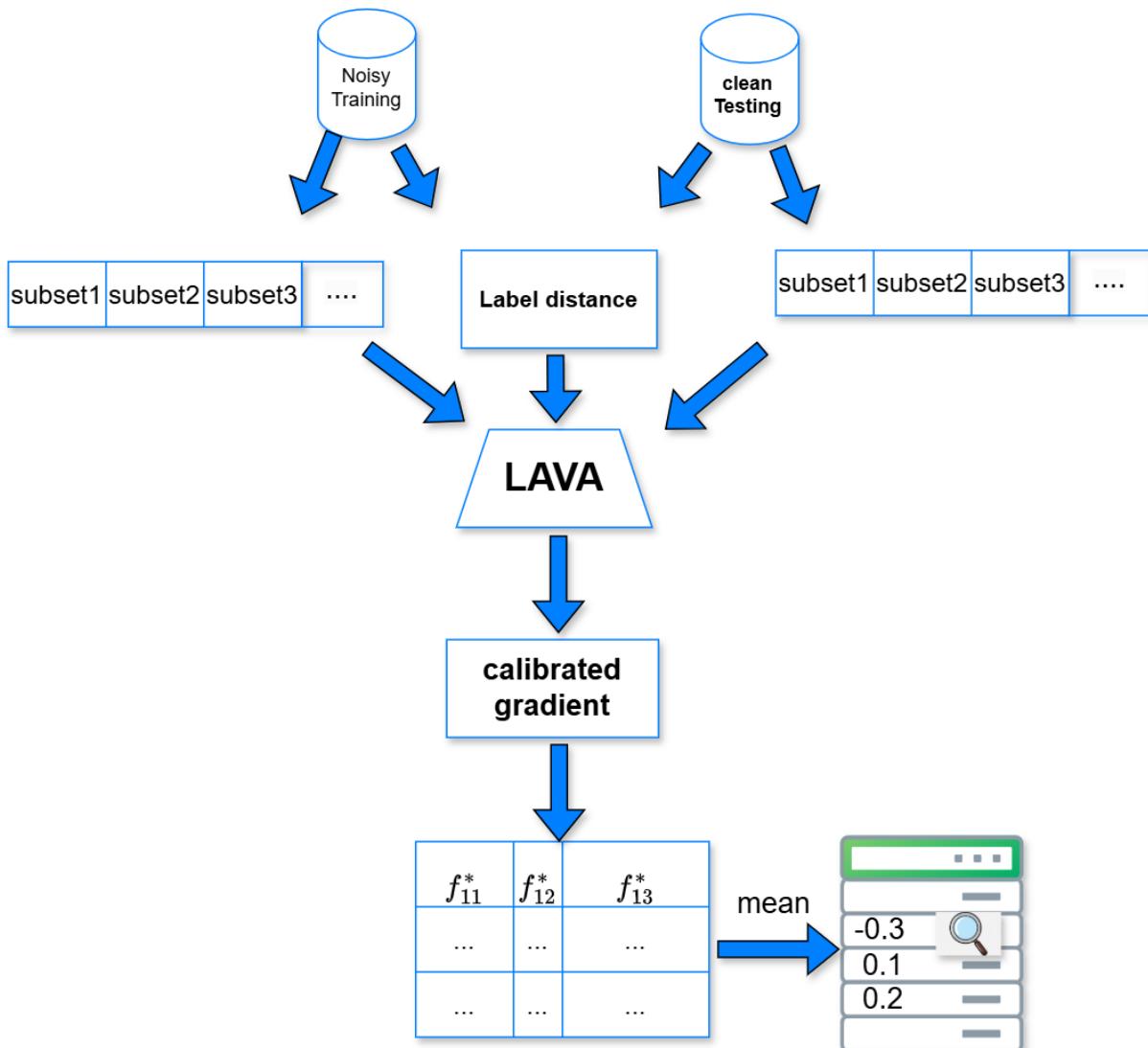
Đề xuất batch-LAVA:

Khi chuyển toàn bộ dữ liệu từ tập huấn luyện (train) sang tập kiểm tra (test), có thể xảy ra tình trạng một số điểm dữ liệu không phản ánh đầy đủ mối tương quan trong phân phối tổng thể. Để khắc phục điều này, chúng em đề xuất áp dụng cross-validation, trong đó tập train và test được chia ngẫu nhiên thành nhiều batch nhỏ hơn. Bằng cách này, thuật toán LAVA có thể được áp dụng nhiều lần trên các batch khác nhau, giúp đánh giá giá trị gradient của từng điểm trong tập train một cách toàn diện hơn. Cuối cùng, giá trị trung bình của các lần lặp được lấy để đảm bảo tính ổn định và giảm thiểu độ nhiễu trong quá trình đánh giá. Việc này không chỉ giúp mô hình tổng quát hóa tốt hơn mà còn giảm thiểu rủi ro overfitting khi đánh giá trên một tập test cố định. Hình minh họa có thể xem tại 3.1.

Tính trước khoảng cách nhãn:

Trong các bài toán có dữ liệu không cân bằng, việc tính toán khoảng cách nhãn trước khi đưa vào mô hình có thể giúp cải thiện đáng kể chất lượng dự đoán. Thay vì tính toán khoảng cách giữa các nhãn trong từng batch một cách độc lập, chúng em đề xuất tiền xử lý khoảng cách nhãn một lần trước khi huấn luyện, giúp tiết kiệm thời gian và giảm bớt gánh nặng tính toán trong quá trình chạy thuật toán LAVA. Kỹ thuật này đặc biệt hữu ích khi làm việc với các tập dữ liệu lớn, nơi việc tính toán khoảng cách động có thể trở thành một vấn đề về hiệu suất.

3.2. Cải tiến LAVA



Hình 3.1: Minh họa việc chia nhỏ tập train và tập test để có thể tính LAVA chéo, giúp thuật toán được học cục bộ hơn và cũng góp phần giảm gánh nặng bộ nhớ.

Chương 4

Thực nghiệm

Vì data valuation là một nhiệm vụ nằm trong data-centric (phần 1.2) các nhà khoa học đã tạo ra những tác vụ đánh giá riêng thể hiện độ hiệu quả của các thuật toán data valuation. Nghĩa là các thuật toán được so sánh qua các tác vụ như phát hiện nhiễu, lựa chọn dữ liệu (data selection),.. được tóm tắt trong [13]. Chúng em sẽ đề xuất một số cách dưới đây.

4.1 Phát hiện dữ liệu nhiễu

Chúng em so sánh khả năng phát hiện dữ liệu nhiễu của các thuật toán định giá dữ liệu khác nhau trên các tập dữ liệu nhiễu được tạo ra tổng hợp. Chúng em xem xét hai loại nhiễu nhân tạo:

- **Nhiễu nhãn:** Lật nhãn ban đầu thành nhãn khác.
- **Nhiễu đặc trưng:** Chuẩn hóa và thêm lỗi ngẫu nhiên từ phân phối Gaussian với trung bình bằng 0 và độ lệch chuẩn là 2.

Tỷ lệ nhiễu được chọn là $\rho_{noise} \in \{20\%\}$. Sau khi cho qua data valuation ta được một tập có kích thước bằng tập train, ta sắp xếp lại thứ tự và chia thành hai nhóm:

- **Có lợi (beneficial):** Nhóm có giá trị dữ liệu cao hơn.
- **Có hại (detrimental):** Nhóm có giá trị dữ liệu thấp hơn.

Chúng em sẽ sử dụng **F1-score** để đánh giá độ hiệu quả của thuật toán (phần 4.3.1)

4.2 Đánh giá hiệu suất

Để đánh giá hiệu suất của thuật toán data valuation ngoài phát hiện nhiễu ta sẽ đánh giá hiệu suất mô hình khi loại bỏ điểm dữ liệu kém chất lượng hoặc thêm điểm dữ liệu chất lượng xem nó thay đổi thế nào so với hiệu suất gốc. (phần 4.3.2)

4.3 Độ đo các phương pháp

Sau khi áp dụng thuật toán **data valuation**, chúng ta có được thứ tự các điểm dữ liệu theo mức độ quan trọng. Để kiểm tra hiệu quả của quá trình định giá dữ liệu, chúng ta thực nghiệm trên các tác vụ sau:

- **phát hiện nhiễu:** dùng f1-score.
- **Xóa dữ liệu:** Lần lượt loại bỏ các điểm dữ liệu kém chất lượng theo thứ tự đã sắp xếp và ngược lại.
- **Thêm dữ liệu:** Dần dần thêm vào các điểm dữ liệu từ tập hợp có giá trị cao đến thấp và ngược lại.

Và để dễ dàng cho việc đánh giá kết quả hơn thay vì mất công phân tích, chúng em đề xuất hai độ đo để giải quyết vấn đề trên là F1-score, và WAD.

4.3.1 F1-score

Phát hiện nhãn sai (Corrupted Label Detection): chúng em đo hiệu suất của thuật toán phát hiện nhãn sai (hay còn gọi là phát hiện lỗi nhãn) bằng F_1 -score của *các điểm dữ liệu bị phát hiện là nhãn sai*, là trung bình điều hòa của độ chính xác (precision) và độ hồi tưởng (recall), cụ thể:

$$F_1 = \frac{2}{\text{Precision}^{-1} + \text{Recall}^{-1}}.$$

$$\text{F1}(S_\phi^{(\text{low})}, S^{(\text{mislabeled})}) := 2 \frac{|S_\phi^{(\text{low})} \cap S^{(\text{mislabeled})}|}{|S_\phi^{(\text{low})}| + |S^{(\text{mislabeled})}|}.$$

4.3. Độ đo các phương pháp

Ý nghĩa:

$$\text{Precision} = \frac{|S_{\phi}^{(\text{low})} \cap S^{(\text{mislabeled})}|}{|S_{\phi}^{(\text{low})}|}, \quad \text{Recall} = \frac{|S_{\phi}^{(\text{low})} \cap S^{(\text{mislabeled})}|}{|S^{(\text{mislabeled})}|}.$$

- Precision: Tỉ lệ tìm thấy nhiễu trên cho tỉ lệ các điểm được chọn ra để đánh giá.
- Recall: Tỉ lệ tìm thấy nhiễu trên tất cả các nhiễu.

4.3.2 Độ đo WAD

Để đo tầm quan trọng giữa thứ hạng của các điểm dữ liệu, mô hình có phản ánh đúng thứ tự của điểm nào chất lượng thấp, điểm nào có chất lượng tốt thật sự không chung em dùng độ đo được đề xuất trong [18] như sau:

$$WAD_D = \sum_{j=1}^n \left(\frac{1}{j} \sum_{i=1}^j [V - V_{D \setminus \{1:i\}}] \right)$$

- $D = \{(x_i, y_i)\}_{i=1}^n$ là tập huấn luyện, với x_i là các điểm dữ liệu và y_i là nhãn tương ứng.
- Tập D được sắp xếp từ giá trị cao nhất đến giá trị thấp nhất dựa trên giá trị đánh giá dữ liệu.
- V là độ chính xác trên tập test khi huấn luyện trên tập train được chỉ định.

Ý nghĩa:

- Khi $i = 1$, $V_{D \setminus \{1:i\}}$ tương đương với độ chính xác dự đoán khi sử dụng toàn bộ tập huấn luyện D .
- WAD cho phép gán tầm quan trọng cao hơn cho các điểm dữ liệu có thứ hạng cao, đồng thời vẫn nắm bắt được hiệu suất tổng thể của mô hình qua các lần loại bỏ dữ liệu.
- Thuật toán dùng để đánh giá sử dụng Logistic Regression với CrossEntropyLoss.

4.4 Thiết lập thí nghiệm

4.4.1 Tập dữ liệu và tiền xử lý

Dữ liệu dạng bảng:

Mỗi đặc trưng trong dataset bảng được chuẩn hóa về trung bình bằng 0 và độ lệch chuẩn bằng 1.

Dataset Name	Samples	Features	Classes/Task
Iris	150	4	3 (Setosa, Versicolor, Virginica)
Digits	1,797	64 (8x8 Pixel)	10 (0-9 Digits)
Breast Cancer	569	30	2 (Benign, Malignant)
Gaussian Classifier	10,000 (Synthetic)	10	Binary
Election	198,000	50+	3 (Political Parties)
2D Planes	4,078	10	Binary
POL	15,000	26	Binary
Fried	5,000	10	Binary
Nomao	34,465	118	Binary
Credit Card Fraud	284,807	30	Binary (Fraud/No Fraud)

Bảng 4.1: Bảng các bộ dữ liệu thực nghiệm dạng bảng

Dữ liệu của chúng em được thu thập từ thư viện OpenML và scikit-learn, phần lớn thuộc dạng bài toán phân loại. Bên cạnh đó, chúng em cũng sẽ đề xuất phương pháp giải quyết dữ liệu cho bài toán hồi quy trong mục 4.6.4.

Dữ liệu không phải dạng bảng:

Đối với dữ liệu không ở dạng bảng (chẳng hạn ảnh, văn bản, âm thanh...), ta có thể sử dụng embedding đã được huấn luyện trước (pretrained embedding) để trích xuất hoặc lọc ra các đặc trưng ẩn. Nhờ đó, khoảng cách hình học giữa các điểm dữ liệu trong không gian được điều chỉnh, phản ánh sát hơn sự tương đồng ngữ nghĩa của chúng. Bước này cũng giúp các thuật toán đánh giá dữ liệu (data valuation) hoạt động chính xác và hợp lý

4.4. Thiết lập thí nghiệm

hơn, vì các đặc trưng đầu vào đã mang nhiều thông tin ý nghĩa trước khi đi vào quá trình tính toán giá trị. Tóm tắt việc sử dụng embedding trên các bộ dataset không phải dạng bảng:

- Ảnh: Sử dụng ResNet50 được huấn luyện trước trên imageNet để trích xuất đặc trưng.
- Ngôn ngữ tự nhiên: Sử dụng DistilBERT được huấn luyện trước cho văn bản.
- Ảnh y tế: chestXrayNet [16].
- Time-series: tiền xử lí

1.Bộ dữ liệu ngôn ngữ tự nhiên

- **BBC Dataset:** Tập dữ liệu BBC bao gồm các bài viết tin tức từ trang BBC, được phân loại thành 5 chủ đề: *business, entertainment, politics, sport, tech*. Đây là tập dữ liệu phổ biến trong các bài toán phân loại văn bản đa lớp và phân tích chủ đề (topic modeling). Tập dữ liệu thường được sử dụng để đánh giá hiệu suất của các mô hình phân loại văn bản truyền thống và hiện đại.

- **IMDB Dataset:** IMDB là tập dữ liệu gồm 50,000 bài đánh giá phim, với các nhãn là *positive (tích cực)* hoặc *negative (tiêu cực)*. Đây là tập dữ liệu tiêu chuẩn trong các bài toán phân tích cảm xúc (sentiment analysis). Các bài đánh giá có độ dài khác nhau, giúp kiểm tra khả năng của mô hình trong việc xử lý văn bản dài và ngữ cảnh phức tạp. IMDB là một trong những tập dữ liệu chính được sử dụng để đánh giá hiệu suất của mô hình LSTM, GRU, và BERT.

- **SST-2 (Stanford Sentiment Treebank):** SST-2 là tập con của Stanford Sentiment Treebank, bao gồm các câu được trích từ bài đánh giá phim. Nhiệm vụ chính là phân loại cảm xúc thành *positive (tích cực)* hoặc *negative (tiêu cực)*. SST-2 là một trong những tập dữ liệu quan trọng trong GLUE Benchmark, thường được sử dụng để đánh giá hiệu suất của các mô hình Transformer như BERT, RoBERTa.

2.Bộ dữ liệu thị giác máy tính thông dụng

Sử dụng một số tập dữ liệu ảnh phổ biến có sẵn trong torchvision như CIFAR-10, MNIST, STL10, SVHN, FashionMNIST. Các tập dữ liệu này thường được sử dụng để

4.4. Thiết lập thí nghiệm

Dataset Name	Samples	Categories	Task Type
BBC	2,225	5 (business, entertainment, politics, sport, tech)	Multi-class Classification
IMDB	50,000	2 (positive, negative)	Binary Classification
SST-2	67,349	2 (positive, negative)	Binary Classification

Bảng 4.2: Bảng các bộ dữ liệu ngôn ngữ tự nhiên được thực nghiệm (BBC, IMDB, SST-2)

huấn luyện và đánh giá các mô hình thị giác máy tính (Computer Vision).

- **CIFAR-10:** Tập dữ liệu CIFAR-10 chứa 60,000 hình ảnh màu, kích thước 32×32 pixel, được chia thành 10 lớp, mỗi lớp có 6,000 hình ảnh. CIFAR-10 là tập dữ liệu chuẩn cho các bài toán phân loại ảnh.
- **MNIST:** Tập dữ liệu MNIST bao gồm 70,000 hình ảnh chữ số viết tay từ 0 đến 9, với kích thước 28×28 pixel. Đây là tập dữ liệu cơ bản để kiểm tra các mô hình nhận dạng ký tự.
- **STL10:** STL10 là một tập dữ liệu ảnh lớn hơn CIFAR, với hình ảnh có độ phân giải 96×96 pixel, chia thành 10 lớp. STL10 đặc biệt phù hợp với bài toán học có giám sát và học không giám sát.
- **SVHN (Street View House Numbers):** SVHN là tập dữ liệu gồm các hình ảnh chữ số được chụp từ biển số nhà, với kích thước 32×32 pixel, chứa hơn 600,000 hình ảnh. Đây là tập dữ liệu có tính thực tế cao và thường được dùng cho bài toán nhận dạng chữ số trong môi trường thực tế.
- **FashionMNIST:** Tập dữ liệu FashionMNIST là phiên bản nâng cấp của MNIST, bao gồm 70,000 hình ảnh của các sản phẩm thời trang thuộc 10 lớp khác nhau. Tập dữ liệu này có độ khó cao hơn MNIST và được sử dụng để đánh giá các mô hình nhận dạng hình ảnh phức tạp.

4.4. Thiết lập thí nghiệm

Bộ dữ liệu	Số lượng mẫu	Số lớp	Kích thước ảnh	Loại ảnh
CIFAR-10	60,000	10	32×32	Màu
CIFAR-100	60,000	100	32×32	Màu
MNIST	70,000	10	28×28	Xám
FashionMNIST	70,000	10	28×28	Xám
STL-10	113,000	10	96×96	Màu
SVHN	630,420	10	32×32	Màu

Bảng 4.3: Bảng các bộ dữ liệu thị giác máy tính được thực nghiệm nằm trong `torchvision.datasets`

3. Bộ dữ liệu ảnh y tế (Healthcare)

Tập dữ liệu ChestX-ray14. Đây là một tập dữ liệu X-quang ngực lớn, trong đó các nhãn bệnh lý được trích xuất từ các báo cáo X-quang bằng các kỹ thuật khai thác văn bản và các chuyên gia [16]. Chúng em lấy 2.000 ảnh X-quang ngực làm tập huấn luyện để thực nghiệm thuật toán định giá dữ liệu và tính toán các giá trị Shapley. 500 ảnh X-quang ngực được sử dụng làm tập kiểm định để tính toán hiệu suất dự đoán trong quá trình huấn luyện, và 610 ảnh X-quang ngực là tập kiểm tra.

Do sự phân bố nhãn trong tập dữ liệu ChestX-ray14 có độ mất cân bằng cao, chúng em đã lấy mẫu với tỷ lệ lớn hơn các trường hợp viêm phổi trong tập huấn luyện, tập kiểm định và kiểm thử được giữ lại như cũ.

4. Bộ dữ liệu có yếu tố thời gian (Time-series)

Chúng em sử dụng bộ dữ liệu lưu lượng giao thông Metro Interstate (Hogue, 2019) được lưu trữ trên UC Irvine Machine Learning Repository, giai đoạn từ tháng 7 đến tháng 9 năm 2013. Cụ thể:

4.4. Thiết lập thí nghiệm

- 748 quan sát trong tháng 7.
- 650 quan sát trong tháng 8.
- 478 quan sát trong tháng 9.

Dữ liệu của mỗi tháng được chia theo tỉ lệ 70-30 (train-test).

Cột đầu ra trong dữ liệu ban đầu thể hiện lưu lượng xe trong một khoảng thời gian cụ thể. Để xác định liệu thời điểm đó có lưu lượng xe đông hay không, chúng em đưa ra một ngưỡng phân loại.

Chúng em chuyển bài toán sang **phân loại nhị phân** bằng cách đặt ngưỡng 3.500:

- Nếu lưu lượng xe vượt quá 3.500 (xe/giờ), gán nhãn “1”.
- Ngược lại, gán nhãn “0”.

Quy trình thí nghiệm:

Thực nghiệm 1:

1. Huấn luyện mô hình ban đầu trên tập tháng 7 (train) và kiểm thử trên tập tháng 7 (test).
2. Lần lượt thêm dữ liệu từ tháng 8 vào dữ liệu huấn luyện (tháng 7), sau đó đánh giá hiệu suất trên tập kiểm thử gộp tháng 7 + tháng 8.

Thực nghiệm 2:

1. Chọn 25% điểm dữ liệu tháng 8 (những điểm giúp mô hình đạt hiệu suất tốt nhất) kết hợp với dữ liệu tháng 7 để đánh giá trên tháng 9.
2. Cuối cùng, chúng em thêm dần dữ liệu tháng 9 và theo dõi sự thay đổi hiệu suất trên tập kiểm thử gộp tháng 7 + 8 + 9.

4.4.2 Cách chia dữ liệu

Vì để phục vụ cho việc phân tích phương pháp data valuation nên chúng ta sẽ không cần cố gắng chạy hết tất cả các điểm dữ liệu (chúng em đã có chạy thử toàn bộ một vài bộ để đánh giá về mặt thời gian có khả thi không).

Dữ liệu được chia thành:

- Tập dataset: $n = 1000 - 2000$
- Riêng tập Cifar10 sử dụng $n = 10000$
- Tỉ lệ train-val-test: $6 : 2 : 2$
- $p_{\text{noise}} = 0.2$ hoặc 0.3

4.4.3 Thuật toán so sánh

- LAVA, KNNShapley, KNNShapley dùng độ đo cosine, KNN-Shapley để xuất, LAVA để xuất.

Để đảm bảo so sánh công bằng, mô hình để đánh giá là LogisticRegression (crossEntropyLoss) với số lần huấn luyện mô hình được giới hạn ở 1000 epochs.

4.5 Kết quả

Sau khi chạy hầu hết các bộ dữ liệu khác nhau ta có nhận xét trên thuật toán LAVA và KNN-shapley:

- **Khả năng phát hiện nhiễu trên nhãn (Label Noise):** kết quả KNN-shapley sẽ tốt hơn về việc xác định các lớp bị nhiễu điều này do mô hình KNN classifier. Khi một mẫu có nhãn bị nhiễu, khoảng cách giữa mẫu đó với các điểm test trong cùng lớp sẽ tăng lên đáng kể, dẫn đến việc thuật toán KNN dễ dàng nhận diện sự bất thường.
- **Khả năng phát hiện nhiễu trên đặc trưng (Feature Noise):** Ngược lại, LAVA cho thấy khả năng phát hiện các mẫu có đặc trưng bị nhiễu tốt hơn. Điều này xuất phát từ việc LAVA nhạy cảm với các sự khác biệt trong không gian đặc trưng (feature space). Khi các đặc trưng của dữ liệu bị thay đổi hoặc thêm nhiễu, LAVA dễ dàng nhận ra các điểm không nhất quán và xử lý hiệu quả các loại dữ liệu có độ biến thiên cao về đặc trưng.

Kiểm tra hiệu suất mô hình thay đổi: Khi chạy tác vụ thêm và xóa điểm dữ liệu (kết quả phần 4.7), trên các bộ dataset như cifar, 2plan, bbc, imdb ta có nhận xét như sau:

- Khi xóa dữ liệu kém chất lượng mô hình có xu hướng độ chính xác tăng lên một chút sau đó mới giảm nhẹ dần. Chứng tỏ việc loại bỏ dữ liệu kém chất lượng khiến mô hình được cải thiện hiệu suất và giảm kích thước tập huấn luyện mà không quá đánh mất nhiều độ chính xác.
- Khi xóa dữ liệu chất lượng tốt mô hình tụt giảm độ chính xác nhanh xuống đáy.
- Khi thêm vào dữ liệu chất lượng kém trước ta thấy mô hình chỉ tăng nhẹ về hiệu suất, có khi đi ngang.
- Khi thêm dữ liệu chất lượng tốt trước mô hình như chóng đạt đỉnh với độ chính xác vốn có của nó.
- Nếu xét về LAVA với KNN-shapley mô hình vẫn phụ thuộc vào hai trường hợp nhiều đặc trưng và nhiều nhãn. Thuật KNN-shapley được đề xuất bám sát với thuật KNN-shapley.

Về phương pháp đề xuất:

- Threshold KNN-shapley: Cho kết quả tốt hơn KNN-shapley ở nhiệm vụ nhiều đặc trưng. Tương đối ngang ở nhiệm vụ nhiều nhãn.
- Batch LAVA: Cho kết quả kém hơn LAVA lý do vì thực hiện trên dữ liệu nhỏ, các batch làm cho mô hình học không được tổng quát.
- Batch LAVA (label-to-label): Nhờ tính sẵn khoảng cách nhãn nên mô hình có kết quả ổn định so với LAVA gốc.
- Ưu điểm của việc chia batch: Giúp mô hình tối ưu thời gian và bộ nhớ khi thực hiện tính toán trên GPU.

Chi tiết như sau:

4.5.1 Kết quả trên F1-score

Thực nghiệm so sánh các phương pháp KNN-shapley, LAVA và phương pháp cải tiến. Một số kí hiệu về thuật toán như sau:

- KNN shapley: KNN shapley bình thường.

4.5. Kết quả

- KNN shapley cosine: KNN shapley dùng khoảng cách cosine.
- KNN shapley ($T = N/2$): KNN shapley để xuất dùng threshold với $T=2$.
- KNN shapley (Best Threshold): KNN shapley để xuất dùng grid search tìm threshold.
- LAVA: LAVA bình thường.
- LAVA (OT): dùng thư viện POT để implement lại thuật toán.
- LAVA (Best Batch Size): LAVA để xuất bằng cách chia nhỏ batch size.
- LAVA (Best label-to-label): LAVA để xuất tương tự LAVA batch size. Thêm vào việc tính toán trước khoảng cách nhãn.

Bảng 4.4: Đánh giá hiệu suất F1-score trên các bộ dữ liệu có nhiều đặc trưng (feature) và nhiều nhãn (label). Lần lượt đưa ra kết quả precision, recall, F1-score và hiệu suất f1 của mô hình sau khi ta loại bỏ dữ liệu được cho là có khả năng cao là nhiễu nhất theo các thuật toán. Thuật toán huấn luyện là logistic regression. Baseline performance là độ chính xác trên toàn bộ tập dữ liệu.

Evaluator	Precision	Recall	F1-Score	Improvement
<i>Baseline Performance = 0.8927</i>				
KNN Shapley	0.5165	0.5165	0.5165	0.8919
KNN Shapley ($T = N/2$)	0.6513	0.9770	0.7816	0.8923
LAVA	1.0000	1.0000	1.0000	0.8924
LAVA (Best Batch Size)	1.0000	1.0000	1.0000	0.8924
LAVA (OT Library Implementation)	1.0000	1.0000	1.0000	0.8923

Bảng 4.5: Thực nghiệm trên CIFAR với tỉ lệ nhiễu đặc trưng là 0.2

4.5. Kết quả

Evaluator	Precision	Recall	F1-Score	WAD	Improvement
<i>Baseline Performance = 0.8840</i>					
KNN Shapley	0.6523	0.9785	0.7828	0.01931	0.8873
KNN Shapley ($T = N/2$)	0.6513	0.9770	0.7816	0.01925	0.8888
LAVA	0.4220	0.6330	0.5064	0.02074	0.8859
LAVA (Best Batch Size)	0.4250	0.6375	0.5100	0.01726	0.8823
LAVA (OT Library Implementation)	0.4626	0.6940	0.5551	0.02485	0.8757

Bảng 4.6: Thực nghiệm trên CIFAR với tỉ lệ nhiễu nhãn là 0.3. Mô tả bảng 4.4

Evaluator	Precision	Recall	F1-Score	Improvement
<i>Base Performance = 0.8565</i>				
KNN shapley cosine	0.5750	0.5750	0.5750	0.8588
KNN Shapley	0.5675	0.5675	0.5675	0.8573
KNN Shapley (Best Threshold)	0.6425	0.6425	0.6425	0.8600
LAVA	1.0000	1.0000	1.0000	0.8595
LAVA (Best Batch Size)	1.0000	1.0000	1.0000	0.8595
LAVA (label-to-label)	1.0000	1.0000	1.0000	0.8595
LAVA (OT Library Implementation)	1.0000	1.0000	1.0000	0.8595

Bảng 4.7: Thực nghiệm trên FashionMnist với tỉ lệ nhiễu đặc trưng là 0.2. Mô tả bảng 4.4

Evaluator	Precision	Recall	F1-Score	Improvement
<i>Base Performance = 0.8565</i>				
KNN Shapley	0.6600	1.0000	0.8000	0.8559
KNN shapley cosine	0.6600	1.0000	0.8000	0.8574
KNN Shapley (Best Threshold)	0.6600	1.0000	0.8000	0.8585
LAVA	0.6600	1.0000	0.8000	0.8607
LAVA (Best Batch Size)	0.6600	1.0000	0.8000	0.8608
LAVA (label-to-label)	0.6600	1.0000	0.8000	0.8624
LAVA (OT Library Implementation)	0.6600	1.0000	0.8000	0.8615

Bảng 4.8: Thực nghiệm trên FashionMnist với tỉ lệ nhiễu đặc trưng là 0.3. Mô tả bảng 4.4

4.5. Kết quả

Evaluator	Precision	Recall	F1-Score	Improvement
<i>Base Performance = 0.8320</i>				
KNN shapley	0.8450	0.8450	0.8450	0.8406
KNN Shapley cosine	0.86	0.86	0.86	0.8426
KNN Shapley (Best Threshold)	0.8475	0.8475	0.8475	0.8406
LAVA	0.6775	0.6775	0.6775	0.8358
LAVA (Best Batch Size)	0.5950	0.5950	0.5950	0.8346
LAVA (label-to-label)	0.605	0.605	0.605	0.8368
LAVA (OT Library Implementation)	0.6625	0.6625	0.6625	0.8335

Bảng 4.9: Thực nghiệm trên FashionMnist với tỉ lệ nhiễu nhãn là 0.2. Mô tả bảng 4.4

Evaluator	Precision	Recall	F1-Score	Improvement
<i>Base Performance = 0.8320</i>				
KNN shapley	0.6416	0.9625	0.7700	0.8427
KNN Shapley cosine	0.6450	0.9675	0.7740	0.8415
KNN Shapley (Best Threshold)	0.6416	0.9625	0.7700	0.8424
LAVA	0.5250	0.7875	0.6300	0.8378
LAVA (Best Batch Size)	0.4883	0.7325	0.5860	0.8339
LAVA (label-to-label)	0.4600	0.6900	0.5520	0.8367
LAVA (OT Library Implementation)	0.5050	0.7575	0.6060	0.8365

Bảng 4.10: Thực nghiệm trên FashionMnist với tỉ lệ nhiễu nhãn là 0.3. Mô tả bảng 4.4

Evaluator	Precision	Recall	F1-Score	Improvement
<i>Baseline Performance = 0.9799</i>				
KNN shapley	0.8050	0.8050	0.8050	0.9795
KNN Shapley (Best Threshold)	0.8600	0.8600	0.8600	0.9825
LAVA	1.0000	1.0000	1.0000	0.9799
LAVA (Best Batch Size)	1.0000	1.0000	1.0000	0.9799
LAVA (label-to-label)	1.0000	1.0000	1.0000	0.9799

Bảng 4.11: Thực nghiệm trên BBC với nhiễu đặc trưng là 0.2. Mô tả bảng 4.4

4.5. Kết quả

Evaluator	Precision	Recall	F1-Score	Improvement
<i>Baseline Performance = 0.9699</i>				
KNN shapley	0.9400	0.9400	0.9400	0.9642
KNN Shapley (Best Threshold)	0.9450	0.9450	0.9450	0.9642
LAVA	0.6550	0.6550	0.6550	0.9638
LAVA (Best Batch Size)	0.6450	0.6450	0.6450	0.9637
LAVA (label-to-label)	0.6600	0.6600	0.6600	0.9660

Bảng 4.12: Thực nghiệm trên BBC với nhiễu nhãn là 0.2. Mô tả bảng 4.4

Evaluator	Precision	Recall	F1-Score	Improvement
<i>Base Performance = 0.6469</i>				
KNN shapley	0.9553	0.9553	0.9553	0.6605
KNN Shapley (Best Threshold)	0.9553	0.9553	0.9553	0.6605
LAVA	0.8044	0.8044	0.8044	0.6583
LAVA (Best Batch Size)	0.6703	0.6703	0.6703	0.6583
LAVA (label-to-label)	0.7318	0.7318	0.7318	0.6840

Bảng 4.13: Thực nghiệm trên Digits với tỉ lệ nhiễu nhãn là 0.2. Mô tả bảng 4.4

Evaluator	Precision	Recall	F1-Score	Improvement
<i>Base Performance = 0.9473</i>				
KNN shapley	0.6480	0.6480	0.6480	0.9413
KNN Shapley (Best Threshold)	0.6592	0.6592	0.6592	0.9413
LAVA	0.9217	0.9217	0.9217	0.9417
LAVA (Best Batch Size)	0.8994	0.8994	0.8994	0.9417
LAVA (label-to-label)	0.8994	0.8994	0.8994	0.9417

Bảng 4.14: Thực nghiệm trên Digits với tỉ lệ nhiễu đặc trưng là 0.2. Mô tả bảng 4.4

4.6. Thí nghiệm so sánh KNN-shapley và LAVA

4.5.2 Thí nghiệm trên bộ dữ liệu time series:

Evaluator	Peak Performance	25% of Data Points Remain
<i>Baseline Performance = 0.640</i>		
KNN Shapley	0.67	0.67
KNN Shapley (Best Threshold)	0.68	0.68
LAVA	0.720	0.730
LAVA (Best Batch Size)	0.728	0.725
LAVA+label-to-label	0.735	0.730
LAVA (OT Library Implementation)	0.740	0.730

Bảng 4.15: Thực nghiệm 1 (Tháng 8): Thực hiện thí nghiệm lưu lượng giao thông khi thêm dần các điểm chất lượng cao theo các thuật toán. Peak performance là hiệu suất cực đỉnh có được và 25% of data points remain là hiệu suất đạt được khi chỉ lấy trên 25% điểm có giá trị tốt nhất.

Evaluator	Peak Performance	25% of Data Points Remain
<i>Baseline Performance = 0.650</i>		
KNN Shapley	0.643	0.615
KNN Shapley (Best Threshold)	0.65	0.62
LAVA	0.715	0.705
LAVA+label-to-label	0.715	0.705
LAVA (OT Library Implementation)	0.715	0.64

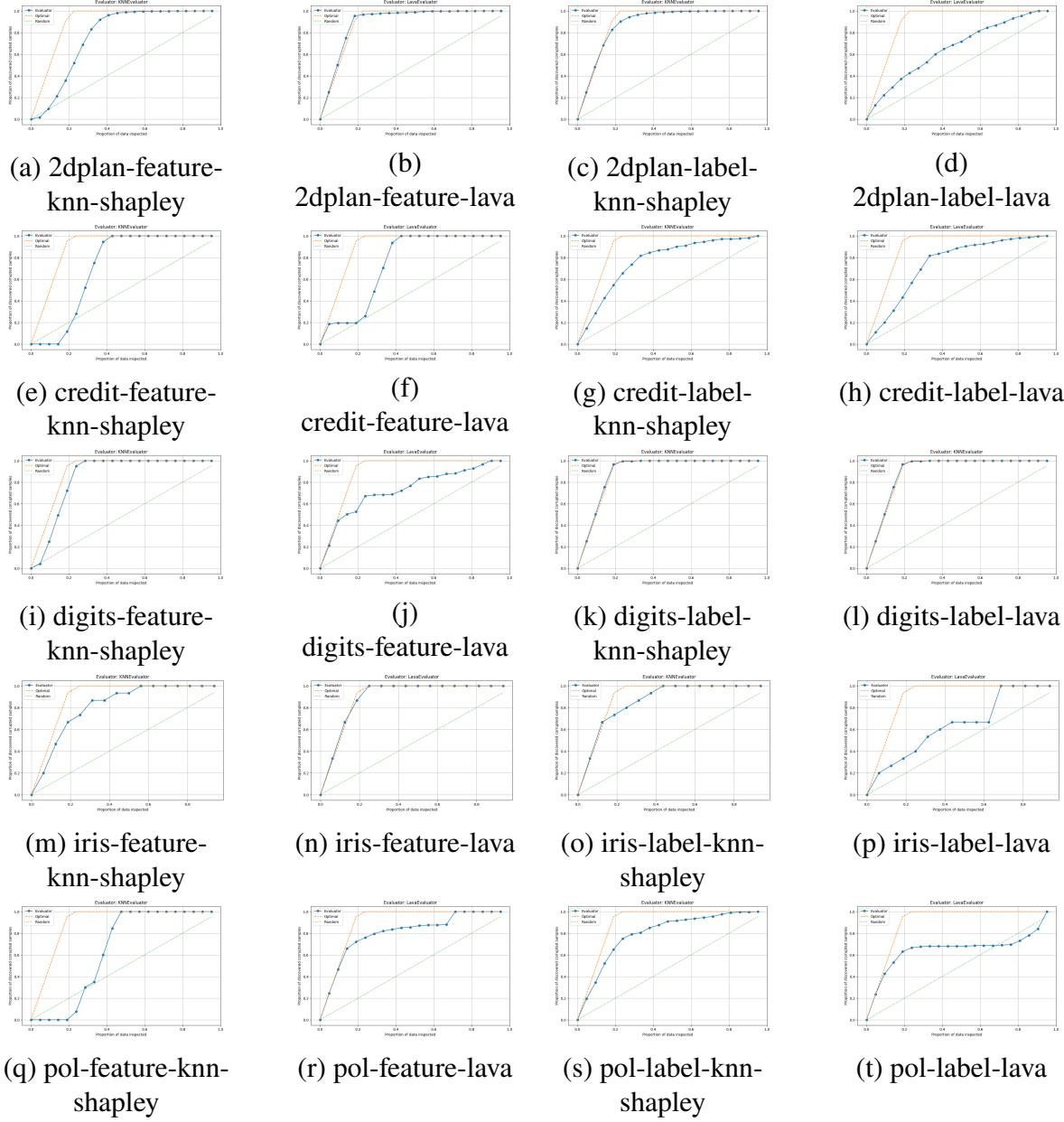
Bảng 4.16: Thực nghiệm 2 (Tháng 9): Nối 25% dữ liệu tháng 8 tốt nhất ở thực nghiệm 1 vào tháng 7. Tiến hành thêm dần tháng 9 vào để đánh giá hiệu suất cực đỉnh và hiệu suất giữ 25% điểm tốt nhất. Nhận thấy các phương pháp tốt nhất là KNN Shapley đề xuất và LAVA đề xuất.

4.6 Thí nghiệm so sánh KNN-shapley và LAVA

Ta chạy hai thuật toán KNN-shapley và LAVA trên mọi loại bộ dữ liệu để thấy được thế mạnh khác nhau của hai thuật toán. KNN-shapley hoạt động tốt trên nhiều nhãn (label), ngược lại LAVA lại ổn định trên nhiều đặc trưng (feature).

4.6. Thí nghiệm so sánh KNN-shapley và LAVA

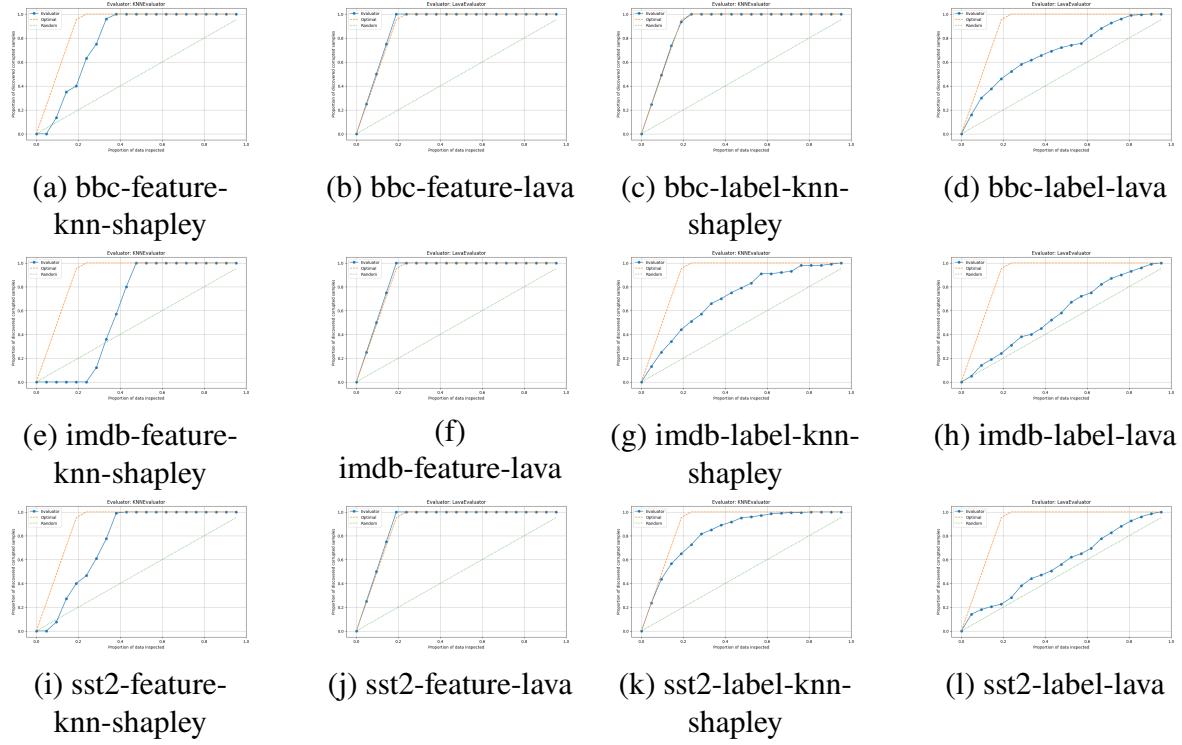
4.6.1 Table



Hình 4.1: Thực nghiệm trên các bộ dữ liệu khác nhau (2dplan, credit, digits, iris, pol)
Trên một biểu đồ ta có ba đường lần lượt là: màu xanh nước biển là tỉ lệ phát hiện nhiễu của thuật toán trên cho lượng điểm dữ liệu chúng ta lấy ra để xét nhiễu (giả định rằng đó là nhiễu) (cột ngang). Hình màu vàng và xanh lá cây nét đứt lần lượt là kết quả tối ưu có thể đạt được, và kết quả có thể đạt được nhờ chọn random. Tên mô tả các thí nghiệm lần lượt là: tên bộ dữ liệu, nhiễu đặc trưng (feature) hoặc nhiễu nhãn (label) và cuối cùng là tên phương pháp.

4.6. Thí nghiệm so sánh KNN-shapley và LAVA

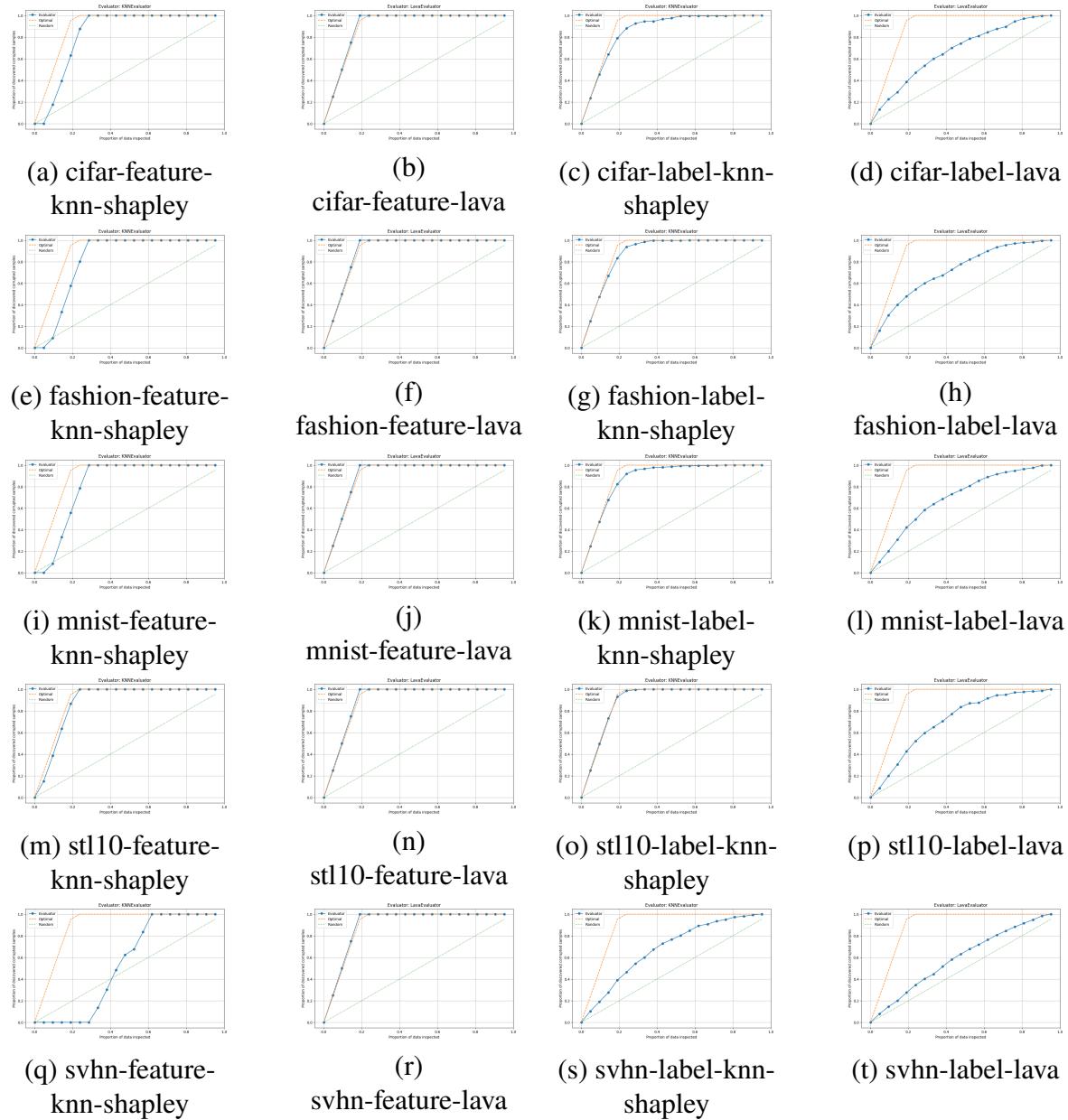
4.6.2 NLP



Hình 4.2: Thực nghiệm trên các bộ dữ liệu ngôn ngữ tự nhiên (BBC, IMDB, SST2).
Xem mô tả tại 4.1

4.6. Thí nghiệm so sánh KNN-shapley và LAVA

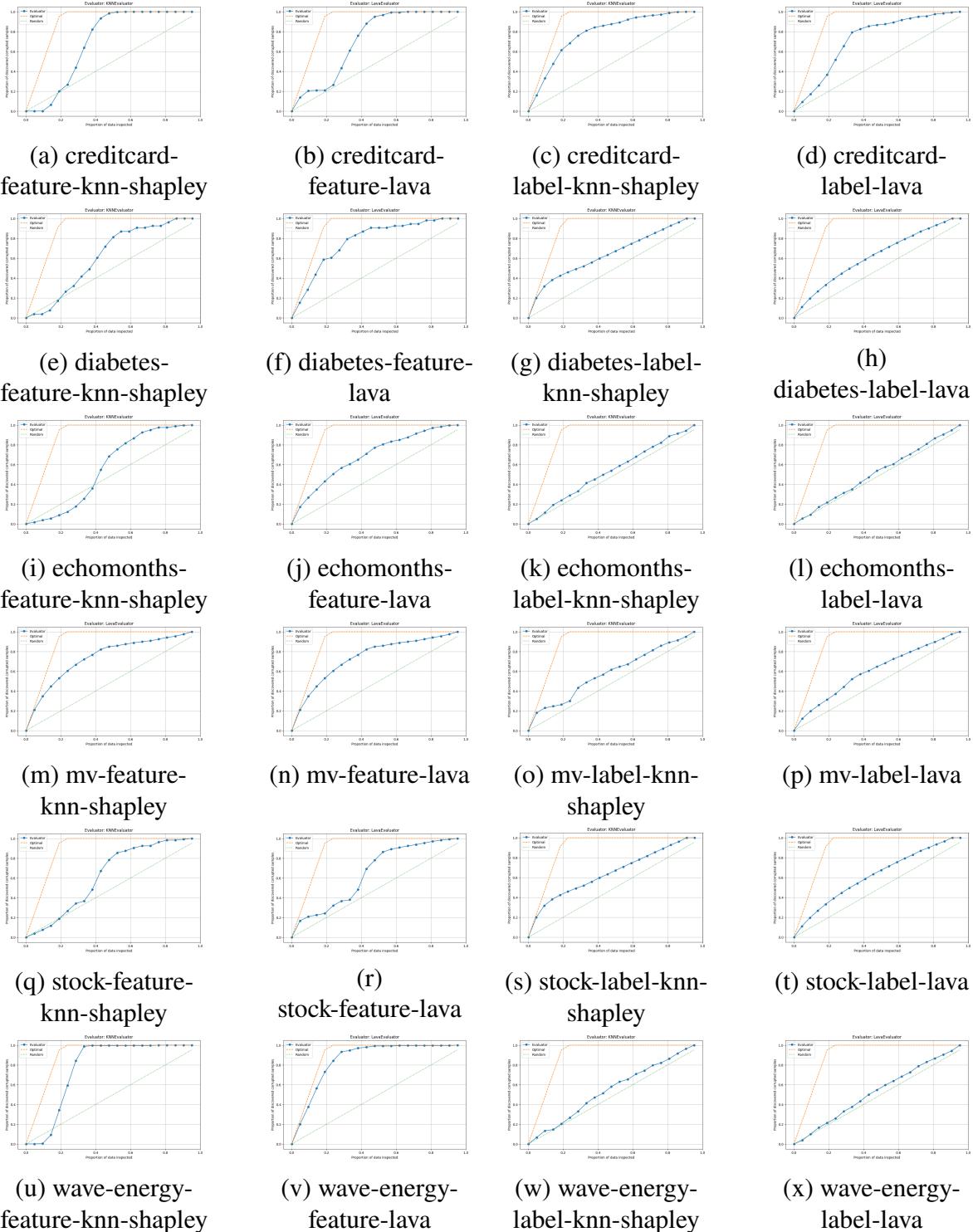
4.6.3 CV



Hình 4.3: Thực nghiệm trên các bộ dữ liệu thị giác máy tính (CIFAR-10, Fashion MNIST, MNIST, STL-10, SVHN). Xem mô tả tại 4.1

4.6. Thí nghiệm so sánh KNN-shapley và LAVA

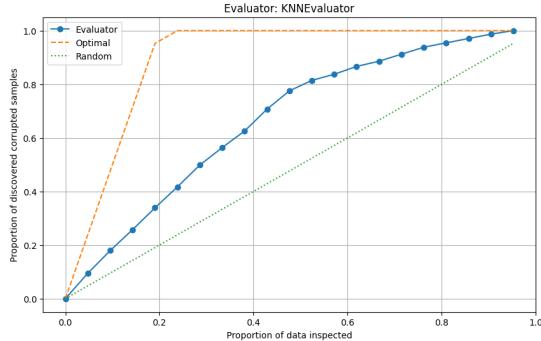
4.6.4 Regression



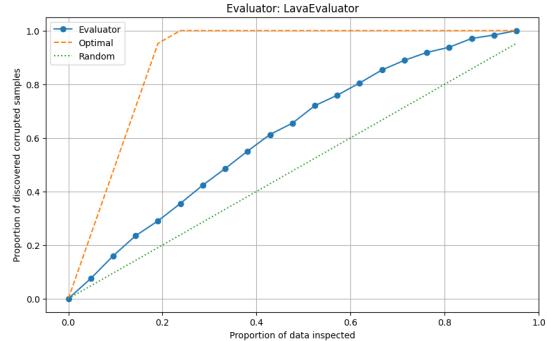
Hình 4.4: Thực nghiệm trên các bộ dữ liệu hồi quy (Regression) (Creditcard, Diabetes, Echomonths, MV, Stock, Wave Energy). Xem mô tả tại 4.1

4.6. Thí nghiệm so sánh KNN-shapley và LAVA

4.6.5 Y tế (Healthcare)



(a) Chest X-ray - KNN-Shapley label noise

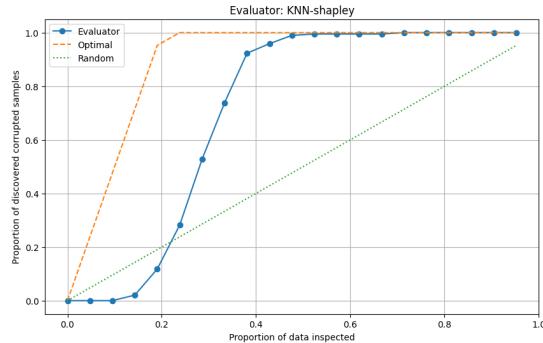


(b) Chest X-ray - LAVA label noise

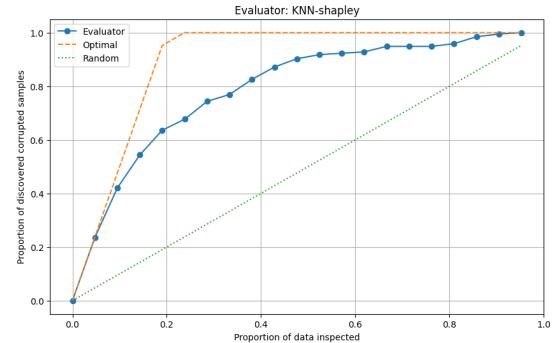
Hình 4.5: Thực nghiệm trên bộ dữ liệu y tế (Chest X-ray). Xem mô tả tại 4.1

4.6. Thí nghiệm so sánh KNN-shapley và LAVA

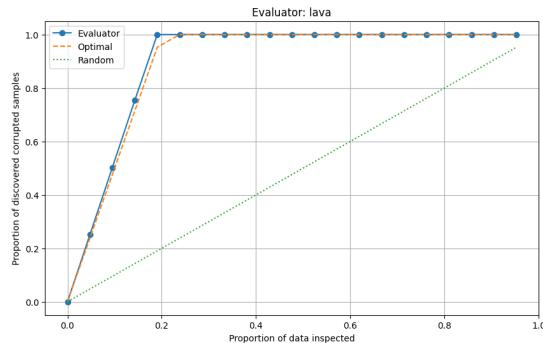
4.6.6 Dữ liệu chuỗi thời gian (Time Series)



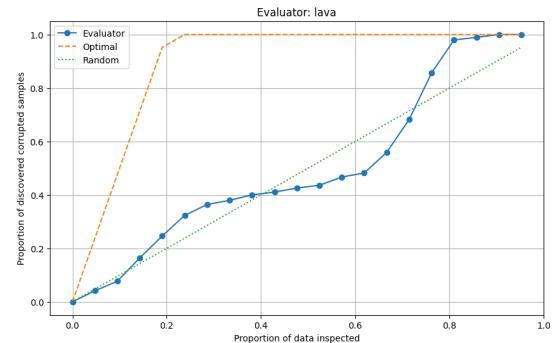
(a) KNN shapley - feature noise



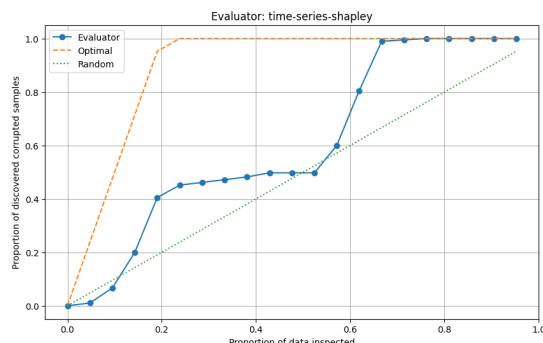
(b) KNN shapley - label noise



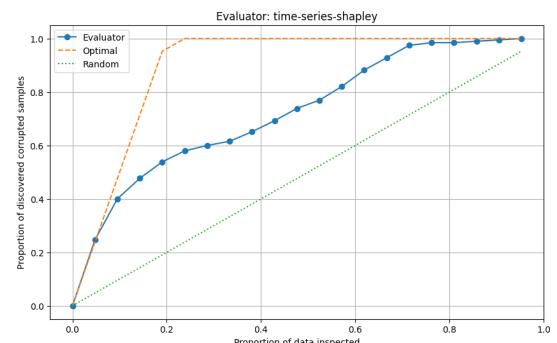
(c) LAVA - feature noise



(d) LAVA - label noise



(e) Time-series shapley - feature noise



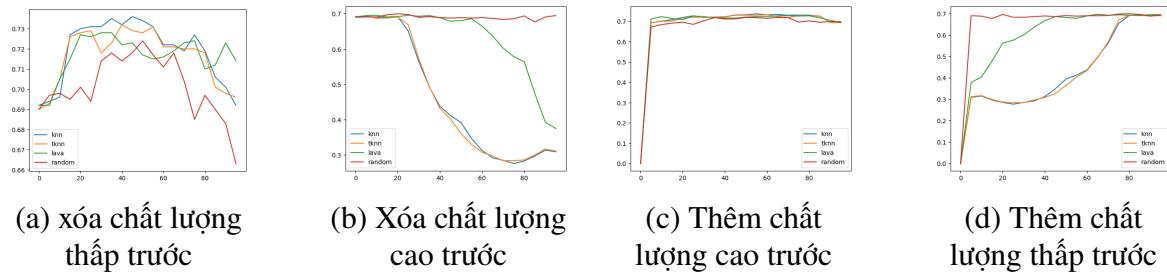
(f) Time-series shapley - label noise

Hình 4.6: Thực nghiệm trên dữ liệu Time series. So sánh với một thuật toán khác là time-series shapley đề xuất trong [25] Xem mô tả tại 4.1

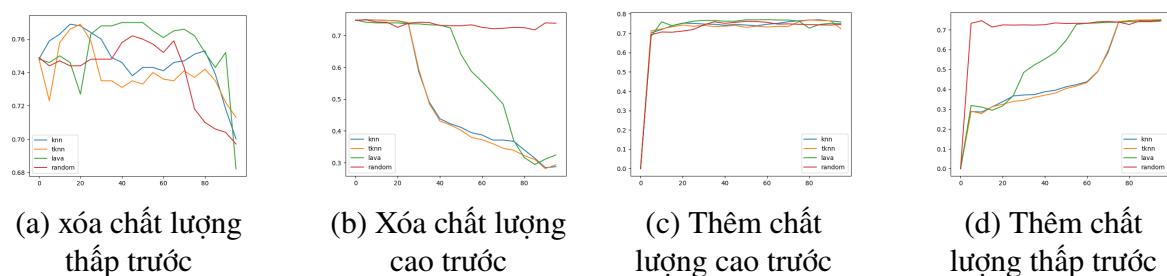
4.7. Thí nghiệm thực hiện thêm/xóa lần lượt dữ liệu được xem là tốt hoặc xấu

4.7 Thí nghiệm thực hiện thêm/xóa lần lượt dữ liệu được xem là tốt hoặc xấu

Thực hiện thêm, xóa lần lượt dữ liệu chất lượng cao hoặc thấp qua mô tả từng ảnh như sau:

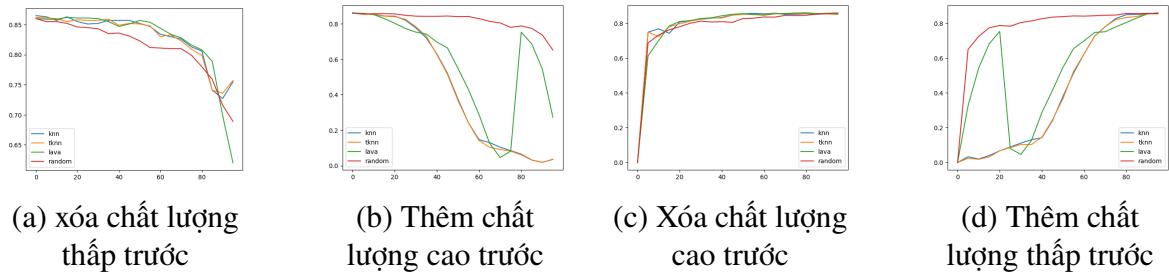


Hình 4.7: Thực nghiệm trên bộ 2dplan với nhiễu nhãn. Mô tả từng biểu đồ từ trái qua phải: ở (a) ta lần lượt đi xóa các điểm dữ liệu được xem là chất lượng thấp khi qua các thuật toán LAVA, KNN-shapley, KNN-shapley đề xuất. Ở (b) lần lượt xóa các điểm dữ liệu chất lượng cao trước, (c) thêm lần lượt điểm chất lượng cao trước, (d) thêm lần lượt điểm chất lượng thấp trước. Hiệu suất mô hình (cột dọc) là các điểm chọn ra được huấn luyện trên logistic regression. Đường màu xanh nước biển là thuật toán KNN-shapley, đường màu vàng là KNN-shapley chúng em đề xuất và hình màu xanh lá cây là thuật toán LAVA, cuối cùng màu đỏ là random nghĩa là xẽ xóa random các dữ liệu.

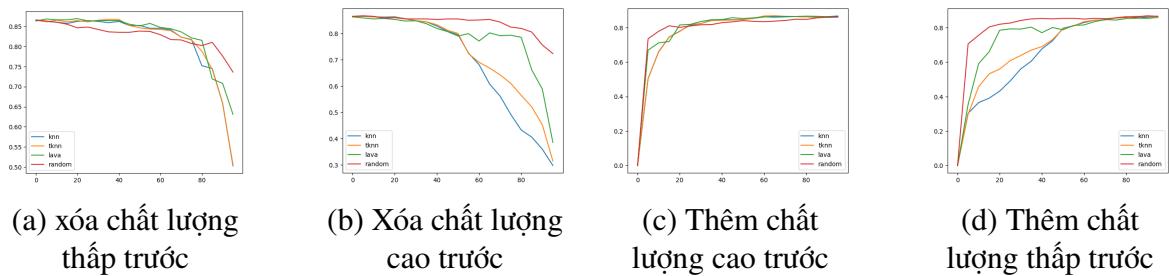


Hình 4.8: Trên bộ 2dplan với nhiễu đặc trưng. Mô tả xem ở 4.7

4.7. Thí nghiệm thực hiện thêm/xóa lần lượt dữ liệu được xem là tốt hoặc xấu

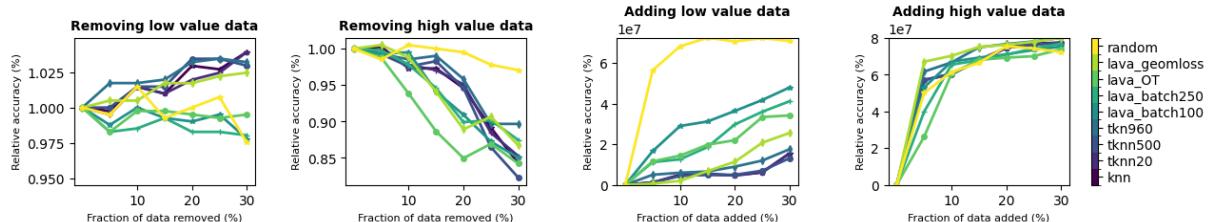


Hình 4.9: Trên bộ cifar với nhiễu nhãn. Mô tả xem ở 4.7



Hình 4.10: Trên bộ cifar với nhiễu đặc trưng. Mô tả xem ở 4.7

Tiến hành zoom lớn và xem sự thay đổi hiệu suất khi chọn ra 30% kích thước dữ liệu:



Hình 4.11: Thí nghiệm được thực hiện trên bộ Cifar10 bị nhiễu nhãn tỉ lệ 0.2. Ta chỉ xét 30% điểm dữ liệu được xóa hoặc thêm và xem sự tăng hiệu suất so với không làm gì (theo tỉ lệ phần trăm).

Kết quả cho thấy thuật toán KNN-Shapley cải tiến đạt kết quả tốt nhất khi nhìn hình 4.11. Ở tác vụ loại dữ liệu chất lượng thấp (trái nhât) ta thấy độ chính xác tăng lên so với ban đầu và KNN-shapley đề xuất (màu xanh đậm) cho kết quả tốt nhất.

4.8 So sánh tìm kiếm với các phương pháp cổ điển

Để minh họa hiệu quả của các phương pháp *data valuation* trong tác vụ tìm kiếm, ta tiến hành so sánh với các thuật toán tìm ngoại lệ (outlier) cổ điển không cần huấn luyện trước, tập trung vào phát hiện điểm bất thường hoặc không nhất quán. Các thuật toán đã được tìm hiểu và tham khảo trên thư viện pyOD (A Python Library for Outlier and Anomaly Detection, Integrating Classical and Deep Learning Techniques) trên github. Cụ thể, các phương pháp sau được sử dụng:

- **KNN Consistency:** Dựa trên ý tưởng: điểm có nhiều láng giềng mang nhãn khác thì nghi là ngoại lệ.
- **Entropy-based Consistency:** Dùng entropy của nhãn lân cận để đo độ lẩn lộn. Entropy càng cao, càng có khả năng điểm đó là ngoại lệ.
- **K-means Inconsistency:** Phân cụm K-means (số cụm = số lớp), gán nhãn majority cho mỗi cụm, điểm nào khác nhãn này được đánh dấu là ngoại lệ.
- **PCA Outlier Detection:** Chiếu xuống không gian PCA, tính khoảng cách đến tâm, điểm nào xa tâm hơn mức cho phép coi là ngoại lệ (outlier).
- **Isolation Forest:** Sử dụng cây chia rẽ ngẫu nhiên (random partitioning trees), điểm bị cô lập sớm thường là ngoại lệ (outlier).
- **Local Outlier Factor (LOF):** So sánh mật độ cục bộ của một điểm so với láng giềng, nếu chênh lệch cao điểm đó là ngoại lệ (outlier).

4.8. So sánh tìm nhiễu với các phương pháp cổ điển

Method	Precision	Recall	F1-Score
KNN-Shapley	0.7675	0.7675	0.7675
KNN-Shapley (Best Threshold)	0.7765	0.7765	0.7765
LAVA	0.4825	0.4825	0.4825
LAVA(batch size)	0.4030	0.4030	0.4030
LAVA(label-to-label)	0.4085	0.4085	0.4085
KNN Consistency	0.8145	0.8145	0.8145
Entropy-based	0.3865	0.3865	0.3865
K-means Inconsistency	0.2415	0.2415	0.2415
PCA Outlier	0.1960	0.1960	0.1960
Isolation Forest	0.1990	0.1990	0.1990
Local Outlier Factor	0.1990	0.1990	0.1990

Bảng 4.17: So sánh các phương pháp phát hiện nhiễu/outlier trên dữ liệu synthetic và thêm nhiễu nhãn.

Method	Precision	Recall	F1-Score
KNN-Shapley	0.4065	0.4065	0.4065
KNN-Shapley (Best Threshold)	0.5950	0.5950	0.5950
LAVA	0.7055	0.7055	0.7055
LAVA(batch size)	0.7095	0.7095	0.7095
LAVA(label-to-label)	0.7120	0.7120	0.7120
KNN Consistency	0.2020	0.2020	0.2020
Entropy-based	0.1990	0.1990	0.1990
K-means Inconsistency	0.1845	0.1845	0.1845
PCA Outlier	0.2180	0.2180	0.2180
Isolation Forest	0.2125	0.2125	0.2125
Local Outlier Factor	0.1990	0.1990	0.1990

Bảng 4.18: So sánh các phương pháp phát hiện nhiễu/outlier với dữ liệu cifar và thêm nhiễu đặc trưng.

4.8. So sánh tìm nhiễu với các phương pháp cổ điển

Method	Precision	Recall	F1-Score
KNN-Shapley	0.8145	0.8145	0.8145
KNN-Shapley(Best Threshold)	0.8260	0.8260	0.8260
LAVA	0.2905	0.2905	0.2905
LAVA(batch size)	0.2845	0.2845	0.2845
LAVA(label-to-label)	0.2900	0.2900	0.2900
KNN Consistency	0.8100	0.8100	0.8100
Entropy-based	0.3755	0.3755	0.3755
K-means Inconsistency	0.2880	0.2880	0.2880
PCA Outlier	0.2065	0.2065	0.2065
Isolation Forest	0.2125	0.2125	0.2125
Local Outlier Factor	0.1995	0.1995	0.1995

Bảng 4.19: So sánh các phương pháp phát hiện nhiễu/outlier với dữ liệu cifar và thêm nhiễu nhãn.

Có thể thấy mỗi phương pháp data valuation đều thể hiện ưu thế riêng khi xử lý từng loại nhiễu, và nhìn chung vượt trội hơn hẳn so với các cách tiếp cận truyền thống như phân cụm (K-means) hay giảm chiều (PCA). Riêng KNN Consistency — do ý tưởng tương đồng với KNN-Shapley trong việc xác định nhiễu — vẫn cho kết quả khả quan; còn lại, các phương pháp truyền thống khác hầu như kém hiệu quả hơn rất nhiều.

Chương 5

Thực nghiệm các khía cạnh khác

Đây là các khía cạnh xung quanh, so sánh và nghiên cứu thêm các vấn đề liên quan đến bài toán như thời gian của các giải thuật data valuation, đề xuất thực hiện trên bộ dữ liệu mất cân bằng hay chất lượng embedding ảnh hưởng thế nào đến các thuật toán data valuation.

5.1 Độ phức tạp thời gian

- Độ phức tạp thuật toán data shapley 2.3: $\mathcal{O}(2^N T)$, T là thời gian huấn luyện thuật toán cụ thể.
- Độ phức tạp thuật toán xấp xỉ TMC-shapley 2.4: $\mathcal{O}(N \log N T)$, T là thời gian huấn luyện thuật toán cụ thể.
- Độ phức tạp thuật toán KNN-shapley: là $\mathcal{O}((N \log N) N_{\text{test}})$ trong đó N là độ dài tập train, N_{test} độ dài tập test. Do việc sắp xếp lại khoảng cách giữa các điểm trên tập train với tập test.
- Độ phức tạp trên KNN-shapley cải tiến: giảm phụ thuộc vào tham số T, với $T=N/2$ thuật toán sẽ là $\mathcal{O}((N/2 \log N/2) N_{\text{test}})$
- Độ phức tạp của LAVA (Optimal Transport) sử dụng Sinkhorn-Knopp 2.7: $\mathcal{O}\left(\frac{N^2}{\varepsilon^2}\right)$ (cụ thể xem ở phần LAVA).

5.2. Cân bằng lại dữ liệu mất cân bằng

Phân tích:

- Dễ thấy hai thuật toán KNN-shapley và LAVA chúng ta đề cập đều có thời gian thực thi nhanh.
- Với tập dữ liệu nhỏ và ít chiều, LAVA thường nhanh hơn KNN-Shapley.
- Khi tập dữ liệu lớn và nhiều chiều, KNN-Shapley thường nhanh hơn LAVA.
- Khi dữ liệu có nhiều lớp, LAVA có thể tốn thời gian hơn do sự phức tạp trong việc tính toán optimal transport giữa các lớp.
- Nhìn chung cả hai thuật toán đều đạt thời gian lý tưởng, làm nổi bật tính khả thi của thuật toán khi áp dụng trên bộ dữ liệu thực.

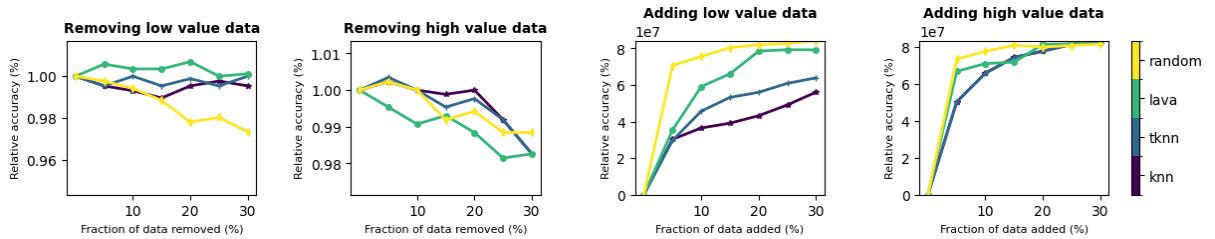
5.2 Cân bằng lại dữ liệu mất cân bằng

Chúng ta khi dạo trên kaggle thường thấy nhiều bộ dữ liệu chất lượng nhưng khi vào các code tiền xử lí ta thấy dữ liệu bị mất cân bằng và các nhà dữ liệu thường loại bỏ một số dữ liệu bị lệch để cân bằng lại các lớp. Vậy sử dụng phương pháp data valuation là một cách lí tưởng vừa vứt dữ liệu gây mất cân bằng, vừa đảm bảo hiệu suất mô hình.

Thực nghiệm trên bộ dữ liệu cifar, với việc lấy một lớp chênh lệch so với các lớp còn lại ở đây là frog với 2000 dữ liệu, trong khi các lớp còn lại là 500 dữ liệu. Sau đó tiến hành kiểm tra trên phương thức loại bỏ điểm có chất lượng thấp dựa trên thuật toán data valuation.

Kết quả cho thấy các thuật toán data valuation không chỉ làm giảm số lượng điểm dữ liệu mà còn làm tăng hiệu suất mô hình. Trong đó LAVA là hiệu quả nhất so với phương pháp KNN-shapley. (Hình 5.1)

5.3. Chất lượng embedding có tác động đến kết quả thế nào



Hình 5.1: Thực hiện tạo dữ liệu mất cân bằng trên cifar10 bằng cách lấy nhiều lớp Frog.

Ở kết quả hình đầu tiên là loại bỏ dữ liệu chất lượng thấp, ta thấy hiệu suất mô hình tăng một ít dù là đang loại bỏ dữ liệu. Mô tả giống 4.11.

5.3 Chất lượng embedding có tác động đến kết quả thế nào

Chúng em sẽ xem xét chất lượng của embedding trên dữ liệu không phải dạng bảng sẽ ảnh hưởng thế nào đến thuật toán data valuation. Từ đó sẽ nhận xét về sự khác nhau giữa các embedding.

Các embedding được sử dụng như resnet50, vgg16, Vi-T, dino (phiên bản mini có thể áp dụng được dễ dàng) và thực hiện trên tập Cifar10.

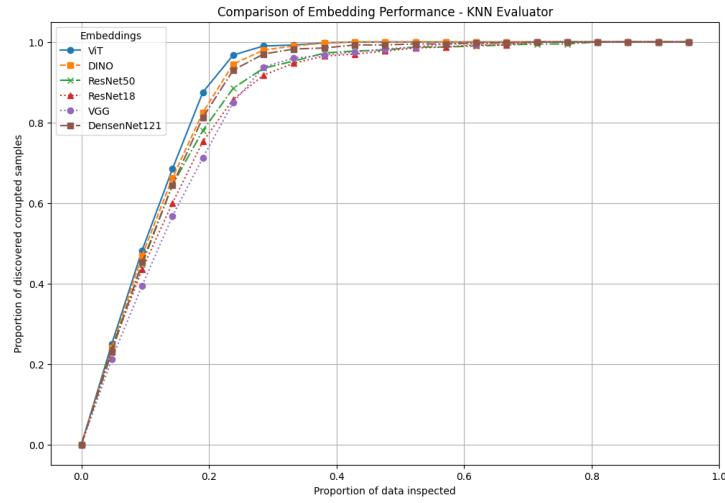
Kết quả cho thấy rằng dù khi sử dụng các embedding khác nhau để trích xuất ra các đặc trưng khác nhau, các thuật toán data valuation đều phát hiện nhiều tốt và đúng với chất lượng embedding nào tốt hơn thì cho kết quả tốt hơn (được đánh giá dựa vào bảng huấn luyện mô hình bằng Logistic Regression ở 5.1). Vì vậy có thể khẳng định tính ứng dụng của data valuation trên các bộ embedding khác nhau nhằm đánh giá chất lượng embedding giúp cho các nhà xây dựng mô hình dễ dàng áp dụng tùy thuộc vào mục đích của họ và tài nguyên họ có (hình 5.2).

5.3. Chất lượng embedding có tác động đến kết quả thế nào

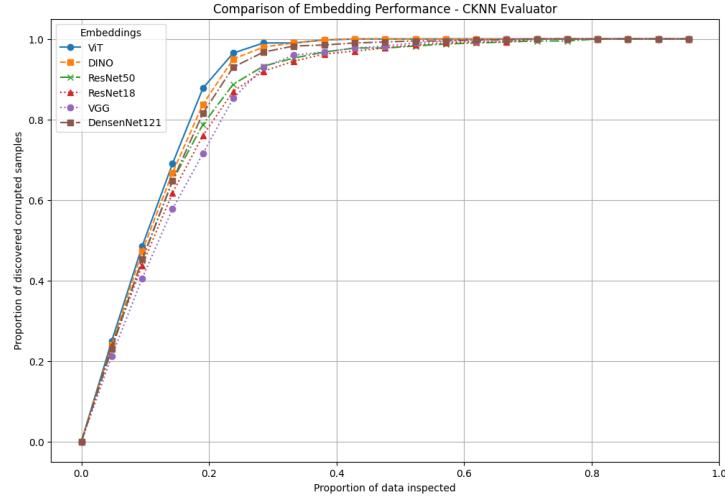
Bảng 5.1: Độ chính xác khi huấn luyện bộ dữ liệu CIFAR với nhiều nhãn bằng các embedding khác nhau trên logistic regression. Ta thấy độ chính xác khác nhau do các phương pháp chúng ta dùng có sức mạnh khác nhau.

Evaluator	F1-Score
ResNet50	0.4650
ResNet18	0.4650
ViT	0.9158
VGG-16	0.7736
DenseNet121	0.8201
DINO-Small	0.8426

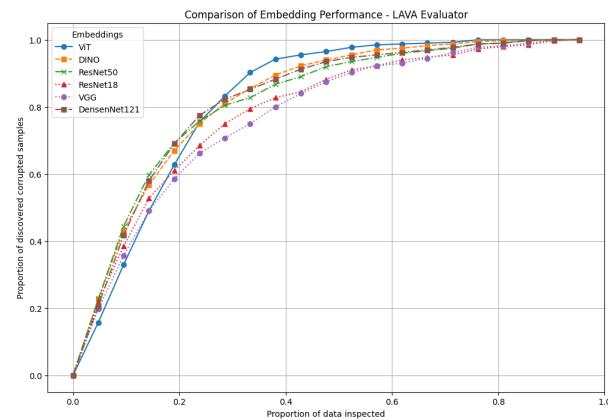
5.3. Chất lượng embedding có tác động đến kết quả thế nào



(a) Tỉ lệ phát hiện nhiễu trên KNN-shapley.



(b) Tỉ lệ phát hiện nhiễu trên KNN-shapley đề xuất.



(c) Tỉ lệ phát hiện nhiễu trên LAVA.

Hình 5.2: Đánh giá trên tỉ lệ phát hiện nhiễu nhãn của cifar10 bằng các embedding khác nhau. Hiệu suất khi sử dụng embedding khác nhau thay đổi không nhiều nhưng phản ánh đúng về chất lượng của embedding. Mô tả biểu đồ có thể xem ở 4.1

5.4. Khảo sát batch size trong LAVA cải tiến

5.4 Khảo sát batch size trong LAVA cải tiến

Evaluator	Precision	Recall	F1-Score
LAVA	0.525	0.7875	0.63
LAVA batch=32	0.2816	0.4225	0.338
LAVA batch=64	0.3833	0.575	0.4600
LAVA batch=128	0.44	0.66	0.5279
LAVA batch=256	0.46	0.69	0.552
LAVA batch=512	0.4883	0.7325	0.5860
LAVA batch=1024	0.5	0.75	0.6

Bảng 5.2: Thực nghiệm so sánh các batch size trên bộ FashionMnist với tỉ lệ nhiễu nhãn là 0.3

Nhận xét: Khi không thực hiện việc tính trung bình trước khoảng cách nhãn như LAVA label-to-label. Việc các batch size càng nhỏ càng khiến mô hình yếu đi do việc không nắm bắt được tốt các đặc trưng của nhãn.

5.5. Khảo sát Threshold trong KNN-shapley cải tiến

5.5 Khảo sát Threshold trong KNN-shapley cải tiến

Evaluator	Precision	Recall	F1-Score
<i>Base Performance = 0.8320</i>			
KNN_shapley	0.845	0.845	0.845
KNN_shapley cosine	0.86	0.86	0.86
KNN_shapley T=2K	0.845	0.845	0.845
KNN_shapley T=N/8	0.8475	0.8475	0.8475
KNN_shapley T=N/4	0.8475	0.8475	0.8475
KNN_shapley T=N/2	0.8475	0.8475	0.8475
KNN_shapley T=2N/3	0.85	0.85	0.85
KNN_shapley batch=N-2K	0.735	0.735	0.735

Bảng 5.3: Thực nghiệm so sánh các threshold khác nhau trên bộ FashionMnist với tỉ lệ nhiễu nhãn là 0.3

Nhận xét:

- Qua quá trình thực nghiệm với KNN-Shapley và phiên bản cải tiến, chúng em nhận thấy tham số $K_{neighbor}$ cho kết quả tốt nhất khi nhận giá trị 5 hoặc 10. Từ đó, chúng ta tiếp tục khảo sát tham số T trong trường hợp cố định $K_{neighbor} = 10$.
- Mô hình threshold KNN-shapley đạt được độ chính xác tốt nhất ở khoảng $N/2$ đến $2N/3$ và vượt trội hơn so với thuật toán KNN-shapley truyền thống.

5.6 Ý nghĩa của việc xử lý nhiễu nhãn

Khả năng xử lý tốt dữ liệu có nhiễu nhãn không chỉ cải thiện độ chính xác của mô hình mà còn mở ra tiềm năng áp dụng trong các bài toán học không giám sát (unsupervised learning), nơi việc gán nhãn có thể được học một cách tự động và dễ xảy ra sai sót.

5.6. Ý nghĩa của việc xử lý nhiễu nhăn

Trong thực tế, nhiễu nhăn trong các tập dữ liệu có thể dẫn đến sự hình thành các mô hình tương quan sai, làm suy giảm khả năng tổng quát hóa của mạng nơ-ron sâu (DNNs). Do đó, việc phát hiện và loại bỏ các mẫu dữ liệu bị lỗi một cách hiệu quả là điều tối quan trọng. Các phương pháp hiện nay chủ yếu tập trung vào việc phát triển các kỹ thuật huấn luyện mạnh mẽ nhằm ngăn DNNs ghi nhớ các mẫu dữ liệu bị nhiễu. Tuy nhiên, phần lớn các phương pháp này đòi hỏi quy trình huấn luyện phức tạp và có nguy cơ overfit với dữ liệu nhiễu, dẫn đến giảm hiệu quả trong việc phát hiện các mẫu sai lệch.

Các thuật toán tiên tiến như KNN-Shapley và LAVA (ứng dụng optimal transport) đã được chứng minh là có hiệu quả trong việc giải quyết vấn đề này. Thông qua việc áp dụng các kỹ thuật học máy như voting và ensemble, những phương pháp này có thể hỗ trợ cải tiến quá trình gán lại nhãn sai (vừa huấn luyện vừa gán lại nhãn qua từng epoch) và giảm thiểu tác động tiêu cực của nhiễu nhăn.

Công trình nghiên cứu ứng dụng KNN-Shapley có thể được tham khảo tại [26], trong khi ứng dụng của LAVA được trình bày trong [5].

Chương 6

Kết luận và hướng nghiên cứu tiếp theo

6.1 Kết quả đạt được trong nghiên cứu

Dù còn tồn tại một số hạn chế trong thực nghiệm, nghiên cứu của em đã đạt được hai mục tiêu chính là ứng dụng và cải tiến. Về mặt ứng dụng, nghiên cứu đã chứng minh rằng định giá dữ liệu (data valuation) là một hướng tiếp cận đầy tiềm năng với khả năng áp dụng rộng rãi trên nhiều loại tập dữ liệu khác nhau. Thuật toán mà nhóm em sử dụng có ưu điểm về tốc độ tính toán và không yêu cầu mô hình học có sẵn, giúp ích đáng kể cho các nhà khoa học dữ liệu áp dụng vào tiền xử lí, giải thích dữ liệu hay áp dụng cho real-time. Đây là một chủ đề mới, hứa hẹn nhiều ứng dụng và tiềm năng phát triển trong tương lai.

Lĩnh vực này hiện đang nhận được sự quan tâm lớn từ cộng đồng các nhà khoa học máy tính và toán học, với nhiều thư viện hỗ trợ đã được phát triển. Mặc dù vậy, em đã đề xuất một số cải tiến cho các phương pháp hiện có trong những điều kiện cụ thể. Tính khả thi của các phương pháp này đã được kiểm chứng và cho thấy không thua kém các thuật toán truyền thống.

6.2 Bàn luận về hướng phát triển

Bài toán định giá dữ liệu không chỉ hướng đến việc cải thiện hiệu suất của các mô hình học máy mà còn mở ra các ứng dụng lâu dài trong lưu trữ và quản lý dữ liệu. Ở một góc độ rộng hơn, việc triển khai thực tiễn các phương pháp định giá dữ liệu có thể đặt ra nhiều câu hỏi liên quan đến kinh tế và xã hội.

Chẳng hạn, trong các sàn giao dịch dữ liệu (data marketplaces), các nhà cung cấp dữ liệu có thể nhân bản, biến đổi hoặc thậm chí sửa đổi dữ liệu một cách bất hợp pháp nhằm tối đa hóa lợi nhuận. Hầu hết các thuật toán định giá dữ liệu hiện nay chưa được thiết kế để đối phó với các hình thức tấn công này, dẫn đến việc đánh giá sai giá trị dữ liệu, ảnh hưởng tiêu cực đến sự minh bạch và độ tin cậy của các sàn giao dịch. Do đó, việc phát triển các thuật toán định giá dữ liệu có khả năng chống lại các hành vi gian lận là một hướng nghiên cứu mới đầy thú vị.

Ngoài ra, vấn đề bảo mật dữ liệu trong các kịch bản học liên kết (federated learning) và học phân tán cũng là một mối quan tâm lớn. Các chủ sở hữu dữ liệu có thể ngần ngại chia sẻ dữ liệu của họ với máy chủ chính vì lo ngại vấn đề quyền riêng tư, đặc biệt là khi các chủ sở hữu này cạnh tranh trực tiếp với nhau. Việc đánh giá dữ liệu dựa trên các đặc trưng tổng hợp (chẳng hạn như gradient) mà không cần truy cập trực tiếp vào dữ liệu là một chủ đề thú vị và đang được quan tâm.

Nhóm em vẫn đang tiếp tục khám phá tiềm năng của các thuật toán hiện tại và mở rộng nghiên cứu sang các chủ đề khá giống data valuation như phân cụm (clustering), (data-agnostic). Các hướng đi xoay quanh các kỹ thuật như vận chuyển tối ưu (optimal transport), đóng góp cận biên (marginal contribution).

Hiện tại, nhóm đang tập trung thử nghiệm ứng dụng của các thuật toán trên các bài toán xử lý ngôn ngữ tự nhiên (NLP) và chuỗi thời gian (time-series). Đặc biệt, trong các bài toán dự báo chuỗi thời gian dữ liệu đã quan sát trước đó có thể ảnh hưởng mạnh mẽ đến các điểm dữ liệu tiếp theo. Điều này gây khó khăn trong việc áp dụng các kỹ thuật của chúng ta. Do đó, phát triển khái niệm định giá dữ liệu trong các kịch bản này là một vấn đề đầy hứa hẹn và có thể tạo ra nhiều ảnh hưởng trong tương lai.

Tài liệu tham khảo

- [1] D. Alvarez-Melis and N. Fusi, “Dataset dynamics via gradient flows in probability space,” in *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, M. Meila and T. Zhang, Eds., ser. Proceedings of Machine Learning Research, vol. 139, PMLR, 2021, pp. 219–230. [Online]. Available: <http://proceedings.mlr.press/v139/alvarez-melis21a.html>.
- [2] D. Alvarez-Melis and N. Fusi, “Geometric dataset distances via optimal transport,” in *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., 2020. [Online]. Available: <https://proceedings.neurips.cc/paper/2020/hash/f52a7b2610fb4d3f74b4106fb80b233d-Abstract.html>.
- [3] L. Djafri, “Pro-smoteboost: An adaptive smoteboost probabilistic algorithm for rebalancing and improving imbalanced data classification,” *Inf. Sci.*, vol. 690, p. 121548, 2025. DOI: [10.1016/j.ins.2024.121548](https://doi.org/10.1016/j.ins.2024.121548). [Online]. Available: <https://doi.org/10.1016/j.ins.2024.121548>.
- [4] Y. Dukler, W. Li, A. T. Lin, and G. Montúfar, “Wasserstein of wasserstein loss for learning generative models,” in *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, K. Chaudhuri and R. Salakhutdinov, Eds., ser. Proceedings of Machine Learning Research, vol. 97, PMLR, 2019, pp. 1716–1725. [Online]. Available: <http://proceedings.mlr.press/v97/dukler19a.html>.
- [5] C. Feng, Y. Ren, and X. Xie, “Ot-filter: An optimal transport filter for learning with noisy labels,” in *IEEE/CVF Conference on Computer Vision and Pat-*

- tern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023*, IEEE, 2023, pp. 16 164–16 174. DOI: [10.1109/CVPR52729.2023.01551](https://doi.org/10.1109/CVPR52729.2023.01551). [Online]. Available: <https://doi.org/10.1109/CVPR52729.2023.01551>.
- [6] J. Feydy, T. Séjourné, F. Vialard, S. Amari, A. Trouvé, and G. Peyré, “Interpolating between optimal transport and MMD using sinkhorn divergences,” in *The 22nd International Conference on Artificial Intelligence and Statistics, AISTATS 2019, 16-18 April 2019, Naha, Okinawa, Japan*, K. Chaudhuri and M. Sugiyama, Eds., ser. Proceedings of Machine Learning Research, vol. 89, PMLR, 2019, pp. 2681–2690. [Online]. Available: <http://proceedings.mlr.press/v89/feydy19a.html>.
- [7] A. Ghorbani and J. Y. Zou, “Data shapley: Equitable valuation of data for machine learning,” in *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, K. Chaudhuri and R. Salakhutdinov, Eds., ser. Proceedings of Machine Learning Research, vol. 97, PMLR, 2019, pp. 2242–2251. [Online]. Available: <http://proceedings.mlr.press/v97/ghorbani19c.html>.
- [8] R. Jia *et al.*, “Efficient task-specific data valuation for nearest neighbor algorithms,” *Proc. VLDB Endow.*, vol. 12, no. 11, pp. 1610–1623, 2019. DOI: [10.14778/3342263.3342637](https://doi.org/10.14778/3342263.3342637). [Online]. Available: <http://www.vldb.org/pvldb/vol12/p1610-jia.pdf>.
- [9] K. F. Jiang, W. Liang, J. Y. Zou, and Y. Kwon, “Opendataval: A unified benchmark for data valuation,” in *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, Eds., 2023. [Online]. Available: http://papers.nips.cc/paper%5C_files/paper/2023/hash/5b047c7d862059a5df623c1ce2982fca-Abstract-Datasets%5C_and%5C_Benchmarks.html.
- [10] H. A. Just *et al.*, “LAVA: data valuation without pre-specified learning algorithms,” in *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*, OpenReview.net, 2023. [Online]. Available: <https://openreview.net/forum?id=JJuP86nBl4q>.

- [11] S. Kessler, T. Le, and V. Nguyen, “SAVA: scalable learning-agnostic data valuation,” *CoRR*, vol. abs/2406.01130, 2024. DOI: [10.48550/ARXIV.2406.01130](https://doi.org/10.48550/ARXIV.2406.01130). arXiv: [2406.01130](https://arxiv.org/abs/2406.01130). [Online]. Available: <https://doi.org/10.48550/arXiv.2406.01130>.
- [12] Y. Kwon and J. Zou, “Beta shapley: A unified and noise-reduced data valuation framework for machine learning,” in *International Conference on Artificial Intelligence and Statistics, AISTATS 2022, 28-30 March 2022, Virtual Event*, G. Camps-Valls, F. J. R. Ruiz, and I. Valera, Eds., ser. Proceedings of Machine Learning Research, vol. 151, PMLR, 2022, pp. 8780–8802. [Online]. Available: <https://proceedings.mlr.press/v151/kwon22a.html>.
- [13] M. Mazumder *et al.*, “Dataperf: Benchmarks for data-centric AI development,” in *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, Eds., 2023. [Online]. Available: http://papers.nips.cc/paper%5C_files/paper/2023/hash/112db88215e25b3ae2750e9eefcded94-Abstract-Datasets%5C_and%5C_Benchmarks.html.
- [14] K. D. Pandl, F. Feiland, S. Thiebes, and A. Sunyaev, “Trustworthy machine learning for health care: Scalable data valuation with the shapley value,” in *ACM CHIL ’21: ACM Conference on Health, Inference, and Learning, Virtual Event, USA, April 8-9, 2021*, M. Ghassemi, T. Naumann, and E. Pierson, Eds., ACM, 2021, pp. 47–57. DOI: [10.1145/3450439.3451861](https://doi.org/10.1145/3450439.3451861). [Online]. Available: <https://doi.org/10.1145/3450439.3451861>.
- [15] K. Pham, K. Le, N. Ho, T. Pham, and H. Bui, “On unbalanced optimal transport: An analysis of sinkhorn algorithm,” in *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, ser. Proceedings of Machine Learning Research, vol. 119, PMLR, 2020, pp. 7673–7682. [Online]. Available: <https://proceedings.mlr.press/v119/pham20a.html>.
- [16] P. Rajpurkar *et al.*, “Chexnet: Radiologist-level pneumonia detection on chest x-rays with deep learning,” *CoRR*, vol. abs/1711.05225, 2017. arXiv: [1711.05225](https://arxiv.org/abs/1711.05225). [Online]. Available: [http://arxiv.org/abs/1711.05225](https://arxiv.org/abs/1711.05225).

- [17] N. Reimers and I. Gurevych, “Sentence-bert: Sentence embeddings using siamese bert-networks,” in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, K. Inui, J. Jiang, V. Ng, and X. Wan, Eds., Association for Computational Linguistics, 2019, pp. 3980–3990. DOI: [10.18653/v1/D19-1410](https://doi.org/10.18653/v1/D19-1410). [Online]. Available: <https://doi.org/10.18653/v1/D19-1410>.
- [18] S. Schoch, H. Xu, and Y. Ji, “Cs-shapley: Class-wise shapley values for data valuation in classification,” in *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, Eds., 2022. [Online]. Available: http://papers.nips.cc/paper%5C_files/paper/2022/hash/df334022279996b07e0870a629c18857-Abstract-Conference.html.
- [19] S. Tang *et al.*, “Data valuation for medical imaging using shapley value: Application on A large-scale chest x-ray dataset,” *CoRR*, vol. abs/2010.08006, 2020. arXiv: [2010.08006](https://arxiv.org/abs/2010.08006). [Online]. Available: <https://arxiv.org/abs/2010.08006>.
- [20] J. T. Wang and R. Jia, “A note on "efficient task-specific data valuation for nearest neighbor algorithms”,” *CoRR*, vol. abs/2304.04258, 2023. DOI: [10.48550/arXiv.2304.04258](https://doi.org/10.48550/arXiv.2304.04258). arXiv: [2304.04258](https://arxiv.org/abs/2304.04258). [Online]. Available: <https://doi.org/10.48550/arXiv.2304.04258>.
- [21] J. T. Wang, P. Mittal, and R. Jia, “Efficient data shapley for weighted nearest neighbor algorithms,” in *International Conference on Artificial Intelligence and Statistics, 2-4 May 2024, Palau de Congressos, Valencia, Spain*, S. Dasgupta, S. Mandt, and Y. Li, Eds., ser. Proceedings of Machine Learning Research, vol. 238, PMLR, 2024, pp. 2557–2565. [Online]. Available: <https://proceedings.mlr.press/v238/t-wang24a.html>.
- [22] J. T. Wang, Y. Zhu, Y. Wang, R. Jia, and P. Mittal, “Threshold knn-shapley: A linear-time and privacy-friendly approach to data valuation,” *CoRR*, vol. abs/2308.15709, 2023. DOI: [10.48550/arXiv.2308.15709](https://doi.org/10.48550/arXiv.2308.15709). arXiv: [2308.15709](https://arxiv.org/abs/2308.15709). [Online]. Available: <https://doi.org/10.48550/arXiv.2308.15709>.

- [23] X. Wang, Y. Peng, L. Lu, Z. Lu, M. Bagheri, and R. M. Summers, “Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases,” in *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, IEEE Computer Society, 2017, pp. 3462–3471. DOI: [10.1109/CVPR.2017.369](https://doi.org/10.1109/CVPR.2017.369). [Online]. Available: <https://doi.org/10.1109/CVPR.2017.369>.
- [24] M. Yurochkin, S. Claici, E. Chien, F. Mirzazadeh, and J. M. Solomon, “Hierarchical optimal transport for document representation,” in *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, H. M. Wallach, H. Larochelle, A. Beygelzimer, F. d’Alché-Buc, E. B. Fox, and R. Garnett, Eds., 2019, pp. 1599–1609. [Online]. Available: <https://proceedings.neurips.cc/paper/2019/hash/8b5040a8a5baf3e0e67386c2e3a9b903Abstract.html>.
- [25] X. Zheng, X. Chang, R. Jia, and Y. Tan, “Towards data valuation via asymmetric data shapley,” *CoRR*, vol. abs/2411.00388, 2024. DOI: [10.48550/ARXIV.2411.00388](https://doi.org/10.48550/ARXIV.2411.00388). arXiv: [2411.00388](https://arxiv.org/abs/2411.00388). [Online]. Available: <https://doi.org/10.48550/arXiv.2411.00388>.
- [26] Z. Zhu, Z. Dong, and Y. Liu, “Detecting corrupted labels without training a model to predict,” in *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvári, G. Niu, and S. Sabato, Eds., ser. Proceedings of Machine Learning Research, vol. 162, PMLR, 2022, pp. 27412–27427. [Online]. Available: <https://proceedings.mlr.press/v162/zhu22a.html>.