Machine Learning Tutorial

nghia.pd225896

March 2025

Mục lục

Danh sách hình vẽ

Danh sách bảng

Tóm tắt nội dung

Đây là phần tóm tắt nội dung báo cáo.

Chương 1

Giới thiệu về Machine Learning

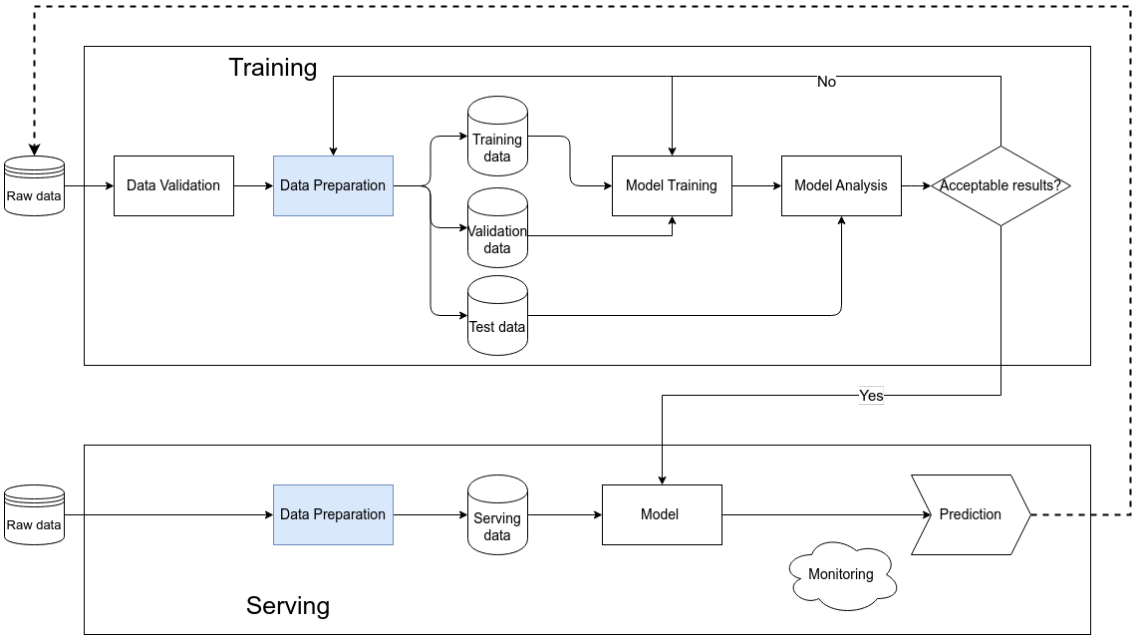
Machine Learning là một tập con của AI. Nói đơn giản, Machine Learning là một lĩnh vực nhỏ của Khoa Học Máy Tính, nó có khả năng tự học hỏi dựa trên dữ liệu đưa vào mà không cần phải được lập trình cụ thể.

Những năm gần đây, khi mà khả năng tính toán của các máy tính được nâng lên một tầm cao mới và lượng dữ liệu khổng lồ được thu thập bởi các hãng công nghệ lớn, Machine Learning đã tiến thêm một bước dài và một lĩnh vực mới được ra đời gọi là Deep Learning, đã giúp máy tính thực thi những việc tưởng chừng như không thể vào 10 năm trước: phân loại cả ngàn vật thể khác nhau trong các bức ảnh, tự tạo chú thích cho ảnh, bắt chước giọng nói và chữ viết của con người, giao tiếp với con người, hay thậm chí cả sáng tác văn hay âm nhạc...

Chương 2

Machine Learning pipeline

Machine Learning Pipeline gồm hai pha chính: Training và Serving, thể hiện luồng dữ liệu từ quá trình huấn luyện đến triển khai thực tế.



Hình 2.1: Machine Learning pipeline (Source: [1])

2.1 Pha Training

- Data Validation: Kiểm định dữ liệu đầu vào, đảm bảo tính tương đồng với dữ liệu trước đó để tránh sai lệch. Ví dụ, nếu tỉ lệ click quảng cáo đột ngột tăng từ 1
- Data Preparation: Làm sạch, tạo đặc trưng, chia thành tập huấn luyện, kiểm định và có thể có tập kiểm tra. Trong trường hợp dữ liệu ít, tập kiểm định có thể được dùng làm tập kiểm tra nhưng cần tránh overfitting.

- Model Training: Huấn luyện mô hình, sử dụng tập kiểm định để tinh chỉnh tham số.
- Model Analysis: Đánh giá mô hình dựa trên các tiêu chí như độ chính xác, tính cân bằng giữa các nhóm dữ liệu, tốc độ dự đoán và tránh rò rỉ dữ liệu. Nếu mô hình chưa đạt yêu cầu, quay lại điều chỉnh hoặc thu thập thêm dữ liệu.

2.2 Pha Serving

- Triển khai mô hình: Mô hình mới không được áp dụng ngay cho toàn bộ dữ liệu mà chạy thử trên một phần nhỏ để so sánh với mô hình cũ. Nếu hiệu quả, dần mở rộng phạm vi áp dụng.
- Data Preparation: Quy trình xử lý dữ liệu đầu vào phải giống hệt như trong pha Training để đảm bảo tính nhất quán.
- Monitoring: Theo dõi phản ứng của người dùng đối với kết quả dự đoán, sử dụng dữ liệu này để cải thiện các mô hình tiếp theo. Nếu không giám sát chặt chẽ, sự thay đổi chất lượng mô hình có thể gây ảnh hưởng nghiêm trọng đến hệ thống.

Chương 3

Phân nhóm các thuật toán Machine Learning

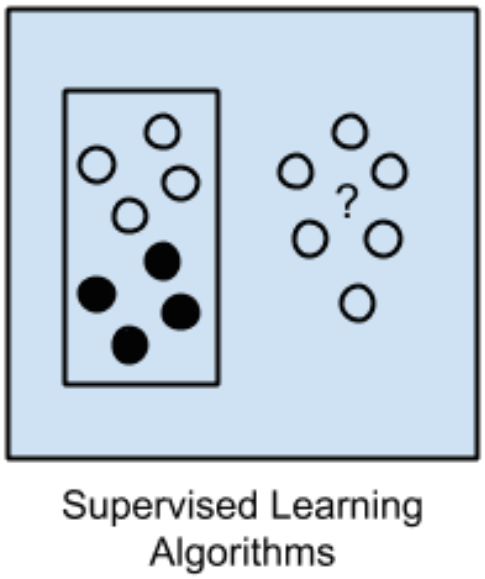
Có hai cách phổ biến phân nhóm các thuật toán Machine learning. Một là dựa trên phương thức học (learning style), hai là dựa trên chức năng (function) (của mỗi thuật toán).

3.1 Phân nhóm dựa trên phương thức học

Theo phương thức học, các thuật toán Machine Learning thường được chia làm 4 nhóm: Supervised learning, Unsupervised learning, Semi-supervised learning và Reinforcement learning. Có một số cách phân nhóm không có Semi-supervised learning hoặc Reinforcement learning.

3.1.1 Supervised Learning (Học có giám sát)

Supervised learning là thuật toán dự đoán đầu ra (outcome) của một dữ liệu mới (new input) dựa trên các cặp (input, outcome) đã biết từ trước. Cặp dữ liệu này còn được gọi là (data, label), tức (dữ liệu, nhãn). Supervised learning là nhóm phổ biến nhất trong các thuật toán Machine Learning.



Hình 3.1: Supervised Learning (Source: [2])

Một cách toán học, Supervised learning là khi chúng ta có một tập hợp biến đầu vào $\mathcal{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N\}$ và một tập hợp nhãn tương ứng $\mathcal{Y} = \{\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_N\}$, trong đó $\boldsymbol{x}_i, \boldsymbol{y}_i$ là các vector. Các cặp dữ liệu biết trước $(\boldsymbol{x}_i, \boldsymbol{y}_i) \in \mathcal{X} \times \mathcal{Y}$ được gọi là tập *training data* (dữ liệu huấn luyện). Từ tập training data này, chúng ta cần tạo ra một hàm số ánh xạ mỗi phần tử từ tập $\mathcal{X}$ sang một phần tử (xấp xỉ) tương ứng của tập $\mathcal{Y}$:

$$\boldsymbol{y}_i \approx f(\boldsymbol{x}_i), \quad \forall i = 1, 2, \ldots, N$$

Mục đích là xấp xỉ hàm số f thật tốt để khi có một dữ liệu $\boldsymbol{x}$ mới, chúng ta có thể tính được nhãn tương ứng của nó $\boldsymbol{y} = f(\boldsymbol{x})$.

Ví dụ: trong nhận dạng chữ viết tay, ta có ảnh của hàng nghìn ví dụ của mỗi chữ số được viết bởi nhiều người khác nhau. Chúng ta đưa các bức ảnh này vào trong một thuật toán và chỉ cho nó biết mỗi bức ảnh tương ứng với chữ số nào. Sau khi thuật toán tạo ra một mô hình, tức một hàm số mà đầu vào là một bức ảnh và đầu ra là một chữ số, khi nhận được một bức ảnh mới mà mô hình chưa nhìn thấy bao giờ, nó sẽ dự đoán bức ảnh đó chứa chữ số nào.

Classification (Phân loại)

Một bài toán được gọi là classification nếu các label của input data được chia thành một số hữu hạn nhóm. Ví dụ: Gmail xác định xem một email có phải là spam hay không; các hãng tín dụng xác định xem một khách hàng có khả năng thanh toán nợ hay không.

Regression (Hồi quy)

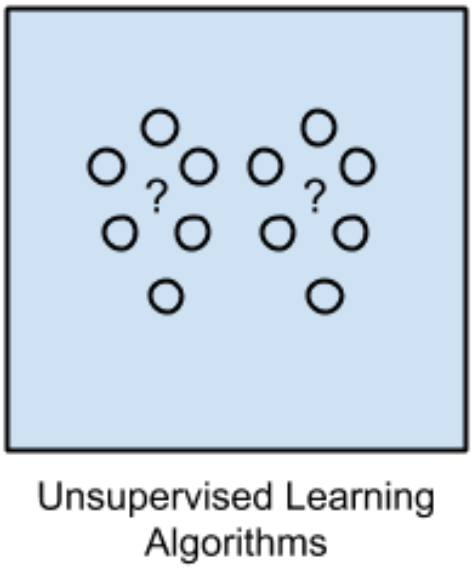
Nếu label không được chia thành các nhóm mà là một giá trị thực cụ thể. Ví dụ: một căn nhà rộng $x\,m^2$, có y phòng ngủ và cách trung tâm thành phố $z\,km$ sẽ có

giá là bao nhiêu?

Gần đây Microsoft có một ứng dụng dự đoán giới tính và tuổi dựa trên khuôn mặt. Phần dự đoán giới tính có thể coi là thuật toán Classification, phần dự đoán tuổi có thể coi là thuật toán Regression. Chú ý rằng phần dự đoán tuổi cũng có thể coi là Classification nếu ta coi tuổi là một số nguyên dương không lớn hơn 150, chúng ta sẽ có 150 class (lớp) khác nhau.

3.1.2 Unsupervised Learning (Học không giám sát)

Trong thuật toán này, chúng ta không biết được outcome hay nhãn mà chỉ có dữ liệu đầu vào. Thuật toán unsupervised learning sẽ dựa vào cấu trúc của dữ liệu để thực hiện một công việc nào đó, ví dụ như phân nhóm (clustering) hoặc giảm số chiều của dữ liệu (dimension reduction) để thuận tiện trong việc lưu trữ và tính toán.



Hình 3.2: Unsupervised Learning (Source: [2])

Một cách toán học, Unsupervised learning là khi chúng ta chỉ có dữ liệu vào $\mathcal{X}$ mà không biết nhãn $\mathcal{Y}$ tương ứng.

Các bài toán Unsupervised learning được tiếp tục chia nhỏ thành hai loại:

Clustering (phân nhóm)

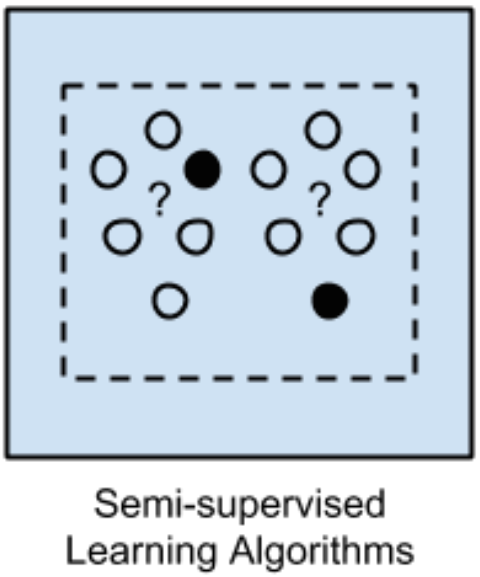
Một bài toán phân nhóm toàn bộ dữ liệu $\mathcal{X}$ thành các nhóm nhỏ dựa trên sự liên quan giữa các dữ liệu trong mỗi nhóm. Ví dụ: phân nhóm khách hàng dựa trên hành vi mua hàng. Điều này cũng giống như việc ta đưa cho một đứa trẻ rất nhiều mảnh ghép với các hình thù và màu sắc khác nhau, ví dụ tam giác, vuông, tròn với màu xanh và đỏ, sau đó yêu cầu trẻ phân chúng thành từng nhóm. Mặc dù không cho trẻ biết mảnh nào tương ứng với hình nào hoặc màu nào, nhiều khả năng chúng vẫn có thể phân loại các mảnh ghép theo màu hoặc hình dạng.

Association

Là bài toán khi chúng ta muốn khám phá ra một quy luật dựa trên nhiều dữ liệu cho trước. Ví dụ: những khách hàng nam mua quần áo thường có xu hướng mua thêm đồng hồ hoặc thắt lưng; những khán giả xem phim Spider Man thường có xu hướng xem thêm phim Bat Man, dựa vào đó tạo ra một hệ thống gợi ý khách hàng (Recommendation System), thúc đẩy nhu cầu mua sắm.

3.1.3 Semi-Supervised Learning (Học bán giám sát)

Các bài toán khi chúng ta có một lượng lớn dữ liệu $\mathcal{X}$ nhưng chỉ một phần trong chúng được gán nhãn được gọi là Semi-Supervised Learning. Những bài toán thuộc nhóm này nằm giữa hai nhóm được nêu bên trên.



Hình 3.3: Semi-Supervised Learning (Source: [2])

3.1.4 Reinforcement Learning (Học Củng Cố)

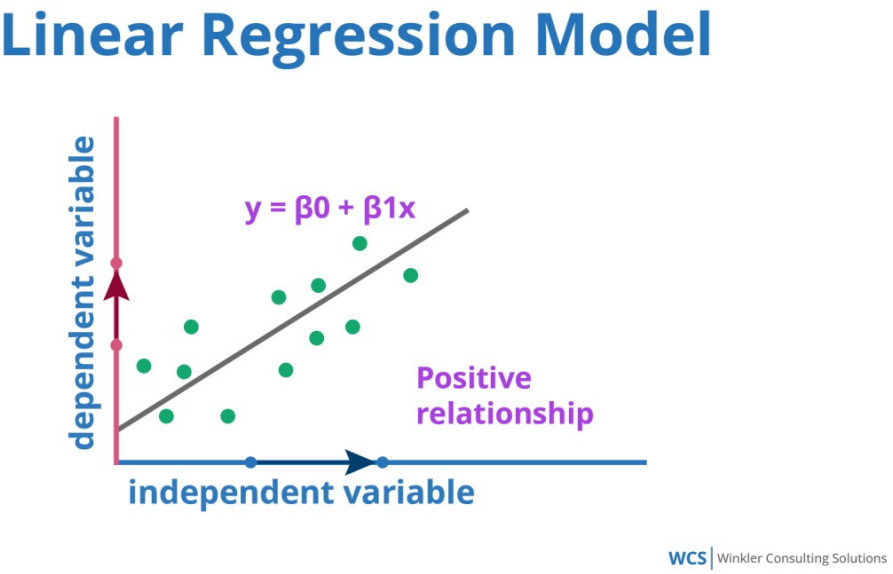
Reinforcement learning là các bài toán giúp cho một hệ thống tự động xác định hành vi dựa trên hoàn cảnh để đạt được lợi ích cao nhất (maximizing the performance). Hiện tại, Reinforcement learning chủ yếu được áp dụng vào Lý Thuyết Trò Chơi (Game Theory), các thuật toán cần xác định nước đi tiếp theo để đạt được điểm số cao nhất.

Ví dụ: Huấn luyện cho máy tính chơi game Mario, AlphaGo gần đây nổi tiếng với việc chơi cờ vây thắng cả con người.

3.2 Phân nhóm dựa trên chức năng

3.2.1 Regression Algorithms

Linear Regression (Hồi Quy Tuyến Tính)



Hình 3.4: Linear Regression (Source: [3])

Bài toán Linear Regression xuất phát từ nhu cầu dự đoán giá trị đầu ra y (output) dựa trên các biến đầu vào $\mathbf{x} = [x_1, x_2, x_3]$ (input). Ví dụ cụ thể:

- x_1: Diện tích nhà (m^2)
- x_2: Số phòng ngủ
- x_3: Khoảng cách tới trung tâm (km)
- y: Giá nhà cần dự đoán

Mối quan hệ giữa đầu vào và đầu ra được biểu diễn bằng hàm tuyến tính:

$$\hat{y} = f(\mathbf{x}) = w_1 x_1 + w_2 x_2 + w_3 x_3 + w_0 \tag{3.1}$$

trong đó:

- w_1, w_2, w_3: Trọng số (weights)
- w_0: Hệ số điều chỉnh (bias)
- $\hat{y}$: Giá trị dự đoán
- y: Giá trị thực tế

ký hiệu:

- Vô hướng: x, y, N (không in đậm)
- Vector: $\mathbf{x}, \mathbf{y}$ (in đậm, chữ thường)
- Ma trận: $\mathbf{X}, \mathbf{W}$ (in đậm, chữ hoa)

Mở rộng vector đầu vào và trọng số:

$$\bar{\mathbf{x}} = [1, x_1, x_2, x_3] \quad \text{(vector hàng mở rộng)}$$
$$\mathbf{w} = [w_0, w_1, w_2, w_3]^T \quad \text{(vector cột trọng số)}$$

Mô hình tuyến tính có thể viết lại:

$$\hat{y} = \bar{\mathbf{x}}\mathbf{w} \tag{3.2}$$

Đo lường sai số giữa giá trị dự đoán và giá trị thực:

$$L(\mathbf{w}) = \frac{1}{2}\sum_{i=1}^{N}(y_i - \bar{\mathbf{x}}_i\mathbf{w})^2 = \frac{1}{2}\|\mathbf{y} - \bar{\mathbf{X}}\mathbf{w}\|_2^2 \tag{3.3}$$

trong đó:

- $L(\mathbf{w})$ được gọi là **hàm mất mát (loss function)** của bài toán Linear Regression.
- Giá trị của $\mathbf{w}$ làm cho hàm mất mát đạt giá trị nhỏ nhất được gọi là điểm tối ưu (optimal point),
- $\mathbf{y} = [y_1; y_2; \ldots; y_N]$: Vector cột output
- $\bar{\mathbf{X}} = [\bar{\mathbf{x}}_1; \bar{\mathbf{x}}_2; \ldots; \bar{\mathbf{x}}_N]$: Ma trận input mở rộng
- $\|\cdot\|_2$: Chuẩn Euclid

Tìm $\mathbf{w}^*$ sao cho $L(\mathbf{w})$ đạt giá trị nhỏ nhất:

$$\mathbf{w}^* = \arg\min_{\mathbf{w}} L(\mathbf{w}) \tag{3.4}$$

Tính gradient và giải phương trình:

$$\frac{\partial L(\mathbf{w})}{\partial \mathbf{w}} = \bar{\mathbf{X}}^T(\bar{\mathbf{X}}\mathbf{w} - \mathbf{y}) = 0 \tag{3.5}$$
$$\Rightarrow \bar{\mathbf{X}}^T\bar{\mathbf{X}}\mathbf{w} = \bar{\mathbf{X}}^T\mathbf{y} \tag{3.6}$$

- Nếu $\mathbf{A} = \bar{\mathbf{X}}^T\bar{\mathbf{X}}$ khả nghịch:

$$\mathbf{w} = \mathbf{A}^{-1}\mathbf{b} \quad \text{với } \mathbf{b} = \bar{\mathbf{X}}^T\mathbf{y} \tag{3.7}$$

- Nếu $\mathbf{A}$ không khả nghịch, sử dụng giả nghịch đảo (pseudo-inverse):

$$\mathbf{w} = \mathbf{A}^{\dagger}\mathbf{b} = (\bar{\mathbf{X}}^T\bar{\mathbf{X}})^{\dagger}\bar{\mathbf{X}}^T\mathbf{y} \tag{3.8}$$

Chú ý quan trọng

- Lý do sử dụng bình phương sai số thay vì trị tuyệt đối:
  - Đạo hàm luôn tồn tại (hàm trị tuyệt đối không khả vi tại 0)
  - Dễ dàng tối ưu hóa bằng giải tích
- Linear Regression giả định mối quan hệ tuyến tính giữa input và output
- Trường hợp đa chiều, xuất hiện khái niệm **siêu mặt phẳng (hyperplane)**

Logistic Regression

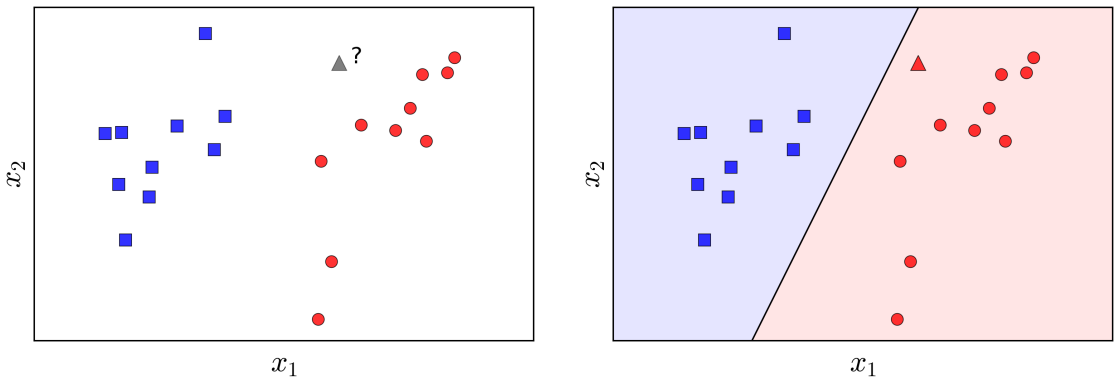
Stepwise Regression

3.2.2 Classification Algorithms

Perceptron Learning Algorithm

Perceptron là một thuật toán Classification cho trường hợp đơn giản nhất: chỉ có hai class (lớp) (bài toán với chỉ hai class được gọi là binary classification) và cũng chỉ hoạt động được trong một trường hợp rất cụ thể.

Giả sử chúng ta có hai tập hợp dữ liệu đã được gán nhãn được minh hoạ trong Hình bên trái dưới đây. Hai class của chúng ta là tập các điểm màu xanh và tập các điểm màu đỏ. Bài toán đặt ra là: từ dữ liệu của hai tập được gán nhãn cho trước, hãy xây dựng một classifier (bộ phân lớp) để khi có một điểm dữ liệu hình tam giác màu xám mới, ta có thể dự đoán được màu (nhãn) của nó.



Hình 3.5: Perceptron Example (Source: [4])

Hiểu theo một cách khác, chúng ta cần tìm lãnh thổ của mỗi class sao cho, với mỗi một điểm mới, ta chỉ cần xác định xem nó nằm vào lãnh thổ của class nào rồi quyết định nó thuộc class đó. Để tìm lãnh thổ của mỗi class, chúng ta cần đi tìm biên giới (boundary) giữa hai lãnh thổ này. Vậy bài toán classification có thể

coi là bài toán đi tìm boundary giữa các class. Và boundary đơn giản nhất trong không gian hai chiều là một đường thẳng, trong không gian ba chiều là một mặt phẳng, trong không gian nhiều chiều là một siêu mặt phẳng (hyperplane) (tôi gọi chung những boundary này là đường phẳng). Những boundary phẳng này được coi là đơn giản vì nó có thể biểu diễn dưới dạng toán học bằng một hàm số đơn giản có dạng tuyến tính, tức linear. Tất nhiên, chúng ta đang giả sử rằng tồn tại một đường phẳng để có thể phân định lãnh thổ của hai class. Hình 1 bên phải minh họa một đường thẳng phân chia hai class trong mặt phẳng. Phần có nền màu xanh được coi là lãnh thổ của lớp xanh, phần có nên màu đỏ được coi là lãnh thổ của lớp đỏ. Trong trường hợp này, điểm dữ liệu mới hình tam giác được phân vào class đỏ.

Bài toán Perceptron được phát biểu như sau: Cho hai class được gán nhãn, hãy tìm một đường phẳng sao cho toàn bộ các điểm thuộc class 1 nằm về 1 phía, toàn bộ các điểm thuộc class 2 nằm về phía còn lại của đường phẳng đó. Với giả định rằng tồn tại một đường phẳng như thế.

Nếu tồn tại một đường phẳng phân chia hai class thì ta gọi hai class đó là linearly separable. Các thuật toán classification tạo ra các boundary là các đường phẳng được gọi chung là Linear Classifier.

Cũng giống như các thuật toán lặp trong **K-means Clustering** và **Gradient Descent,** ý tưởng cơ bản của PLA là xuất phát từ một nghiệm dự đoán nào đó, qua mỗi vòng lặp, nghiệm sẽ được cập nhật tới một ví trí tốt hơn. Việc cập nhật này dựa trên việc giảm giá trị của một hàm mất mát nào đó.

Ví dụ: Chúng ta có một tập dữ liệu gồm các bông hoa với hai đặc trưng: độ dài cánh hoa và màu sắc. Mỗi bông hoa thuộc một trong hai loại: Hoa A (1) hoặc Hoa B (0).

- Bước 1: Khởi tạo tham số
  - Trọng số w1 cho x
  - Trọng số w2 cho y
  - Giá trị ngưỡng b (bias)

  Trọng số được khởi tạo ngẫu nhiên, tương đương với phương trình đường thẳng:
  $$f(x, y) = w_1 x + w_2 y + b = 0 \tag{3.9}$$
  Nếu:
  - $f(x, y) \geq 0$: Hoa A (1)
  - $f(x, y) < 0$: Hoa B (0)
- Bước 3: Cập nhật trọng số
  - Nhãn thực tế = 0 Nhãn dự đoán = 0 : **Đúng**
  - Nhãn thực tế = 1 Nhãn dự đoán = 1 : **Đúng**
  - Nhãn thực tế = 0 Nhãn dự đoán = 1 : **Sai**
  - Nhãn thực tế = 1 Nhãn dự đoán = 0 : **Sai**

- Với mỗi điểm dữ liệu (x,y) dự đoán sai, thuật toán sẽ điều chỉnh các trọng số:
  - $w_1 = w_1 + lr \times (label_{true} - label_{predicted}) \times x$
  - $w_2 = w_2 + lr \times (label_{true} - label_{predicted}) \times y$
  - $b = b + lr \times (label_{true} - label_{predicted})$
  - Với labeltrue là nhãn thực tế, labelpredicted là nhãn dự đoán; lr (learning rate) quyết định tốc độ học của thuật toán (learning rate càng lớn w1,w2,b thay đổi càng nhanh)
  - Lặp lại Bước 2 và Bước 3 đến khi:
    * Không còn điểm dữ liệu nào bị phân loại sai.
    * Hoặc số lần lặp đạt đến giới hạn tối đa.

Linear Classifier

Support Vector Machine (SVM)

Kernel SVM

Sparse Representation-based classification (SRC)

3.2.3 Instance-based Algorithms

k-Nearest Neighbor (kNN)

Learning Vector Quantization (LVQ)

3.2.4 Regularization Algorithms

Ridge Regression

Least Absolute Shrinkage and Selection Operator (LASSO)

Least-Angle Regression (LARS)

3.2.5 Bayesian Algorithms

Naive Bayes

Gaussian Naive Bayes

3.2.6 Clustering Algorithms

k-Means clustering

k-Medians

Expectation Maximization (EM)

3.2.7 Artificial Neural Network Algorithms

Perceptron

Softmax Regression

Multi-layer Perceptron

Back-Propagation

3.2.8 Dimensionality Reduction Algorithms

Principal Component Analysis (PCA)

Linear Discriminant Analysis (LDA)

3.2.9 Ensemble Algorithms

Boosting

AdaBoost

Random Forest

Tài liệu tham khảo

[1] Machine Learning pipeline
https://machinelearningcoban.com/tabml_book/ch_intro/pipeline.html

[2] A Tour of Machine Learning Algorithms
https://machinelearningmastery.com/a-tour-of-machine-learning-algorithms/

[3] Linear Regression Model Application in Business
https://www.linkedin.com/pulse/linear-regression-model-application-6mqze

[4] Machine Learning pipeline
https://machinelearningcoban.com/2017/01/21/perceptron/