Báo cáo tiến trình học tập tuần 1

1. **Giới thiệu về Machine learning**

Machine learning (Máy học) là một mảng con (Artificial Intelligence), nó có khả năng tự học hỏi dựa trên các dữ liệu đưa vào mà không cần lập trình cụ thể.

1. **Phân nhóm các thuật toán Machine learning dựa trên phương thức học**

Theo phương thức học Machine learning được chia thành 4 nhóm: Supervised learning, Unsupervised learning, Semi-supervised learning, Reinforcement learning.

1. **Supervised learning**

Là thuật toán dự đoán đầu ra (outcome) của một dữ liệu mới (new input) dựa trên các cặp (input, output) đã biết từ trước. Các cặp này còn được gọi là (data, label).

Nghĩa là từ các dữ liệu đầu vào và đầu ra có từ trước (gọi là traing data), ta cần tạo ra một hàm ánh xạ để từ một đầu vào mới ta cho ra (xấp xỉ) một đầu ra tương ứng.

Thuật toán supervised learning được chia thành 2 loại chính:

* Regression:

Khi các label không phải là các nhóm hữu hạn mà là các giá trị thực cụ thể. VD: Giá một căn nhà.

* Classification (Phân loại):

Khi các label được chia thành các nhóm hữu hạn. VD: Một bệnh nhân được xác nhận là ung thư hay không - phân thành 2 nhóm là có và không.

1. **Unsupervised learning**

Trong unsupervised learning, chúng ta chỉ có dữ liệu đầu vào mà không có dữ liệu đầu ra (outcome hay nhãn), chủ yếu dựa vào cấu trúc để thực hiện một số công việc như phân nhóm (clustering) hoặc giảm số chiều của dữ liệu (dimension reduction).

Thuật toán unsupervised learning được chia thành 2 loại chính:

* Clustering (phân nhóm):

Phân nhóm toàn bộ dữ liệu ban đầu thành các nhóm nhỏ hơn có sự tương quan giữa các dữ liệu trong nhóm. VD: Một bài toán thực tế cho trẻ em như có nhiều mảnh ghép gồm nhiều loại hình dạng, công việc là tìm các hình ghép giống nhau rồi gom chúng thành một nhóm.

* Association (Mối liên kết):

Là bài toán khi chúng ta muốn khám phá ra một quy luật dựa trên dữ liệu cho trước. VD: Những người biết đánh bóng chuyền thường bắt gôn hay.

1. **Semi-Supervised learning (Học bán giám sát)**

Có một lượng lớn dữ liệu, nhưng chỉ có một phần trong chúng được gán nhãn.

Thực tế cho thấy việc thu thập dữ liệu có nhãn tốn thời gian và có chí phí cao, còn dữ liệu chưa có nhãn thì thu thập được với chi phí thấp.

1. **Reinforcement learning (Học củng cố)**

Là bài toán giúp hệ thống tự động xác định hành vi dựa trên hoàn cảnh để đạt được lợi ích cao nhất.

Nó thường được sử dụng trong lý thuyết trò chơi (Game Theory), thuật toán xác định nước đi tiếp theo để đạt được điểm số cao nhất.

1. **Thuật toán Linear regression**
2. **Giới thiệu bài toán**

Giả sử một căn nhà rộng , có phòng ngủ và cách trung tâm thành phố km có giá là bao nhiêu. Giả sử chúng ta có sẵn số liệu thống kê của 1000 căn nhà về các thông số này, bao gồm thêm cả giá nhà.

Ta có hàm dự đoán giá nhà là:

Với là các hằng số, còn được gọi là bias. Mối quan hệ giữa

là một mối quan hệ tuyến tính, bài toán chúng ta đang đi tìm là bài toán hồi quy, nên được gọi là hồi quy tuyến tính.

1. **Phân tích toán học**
2. **Dạng của Linear regression**

Đặt , hệ số cần tối ưu là ,khi đó phương trình hàm dự đoán được viết lại là:

1. **Ý tưởng xây dựng**

Chúng ta mong muốn sự sai khác giữa giá trị thực y và giá trị dự đoán là nhỏ nhất, nên ta có:

Lấy của bình phương sai số giữa outcome y từ tập traning data và hàm dự đoán f(x).

Trong đó là để thuận tiện trong việc tính toán đạo hàm (khử 2 khi đạo hàm), chúng ta cần bình phương vì sai số có thể là một số âm, và khi đó dù cho giá trị sai số là rất nhỏ nhưng khoảng cách sai số thực sự lớn. Không dùng trị tuyệt đối vì với phương trình có hàm trị tuyệt đối không phải khi nào cũng tồn tại khi đạo hàm (đạo hàm trị tuyệt đối không xác định tại 0).

1. **Hàm mất mát**

Một cách tổng quát, chúng ta muốn tổng sai số của tất các các cặp (input, output) là nhỏ nhất, tương đương với việc tìm w để hàm số sau đạt min:

Đặt là một vector cột chứa tất cả các output của traning data; là ma trận dữ liệu đầu vào mà mỗi hàng của nó là một điểm dữ liệu. Khi đó hàm số mất mát L(w) được viết dưới dạng đơn giản hơn:

1. **Nghiệm của bài toán**

Để tìm cực trị cho hàm số, ta giải phương trình đạo hàm bằng 0.

Đạo hàm theo w của hàm mất mát L(w) là:

* Vector tham số w để phương trình dự đoán f(x) có sai số nhỏ nhất được sinh ra bằng phương trình:

1. **Hạn chế của Linear regression**

* Nhạy cảm với nhiễu (sensitive to noise): trước khi thực hiện Linear Regression, các outlier cần phải được loại bỏ. Bước này gọi là tiền xử lý.
* Không biểu diễn được các mô hình phức tạp.

1. **Thuật toán K-means clustering**
2. **Giới thiệu bài toán**

Trong thuật toán K-means clustering, chúng ta không biết nhãn (label – output) của từng điểm dữ liệu. Mục đích là làm thế nào để phân dữ liệu thành các cụm (cluster) khác nhau sao cho dữ liệu trong cùng một cụm có tính chất giống nhau.



Giả sử mỗi cluster có một điểm đại diện (center) màu vàng. Và những điểm xung quanh mỗi center thuộc vào cùng một nhóm với center đó. Một cách đơn giản nhất, xét một điểm bất kỳ, ta xét xem điểm đó gần với center nào nhất thì nó thuộc về cùng nhóm với center đó.

Có một cách phân chia lãnh thổ được gọi là Voronoi Diagram, chúng là các đường trung trực của các cặp điểm gần nhau.

**2. Phân tích toán học**

**a. Một số ký hiệu toán học**

Giả sử có N điểm dữ liệu với K < N là số cluster chúng ta muốn phân chia.

Chúng ta cần tìm các center .

Với mỗi điểm dữ liệu đặt là các label vector của nó, trong đó nếu được phân vào cluster k thì và . Cách biểu diễn này được gọi là biểu diễn one-hot.

Ràng buộc của có thể viết dưới dạng toán học như sau:

**b. Hàm mất mát và bài toán tối ưu**

Giả sử ta coi center là center của mỗi cluster và ước lượng tất cả các điểm được phân vào cluster này bởi , thì một điểm dữ liệu được phân vào cluster k sẽ bị sai số là . Chúng ta muốn sai số này có giá trị nhỏ nhất nên, ta sẽ tìm cách để đại lượng sau đây đạt giá trị nhỏ nhất:

Hơn nữa, vì được phân vào cluster k nên **.** Khi đó, biểu thức trên sẽ được viết lại là:

Suy ra sai số cho toàn bộ dữ liệu là:

Chúng ta cần tối ưu bài toán sau:

**c. Thuật toán tối ưu hàm mất mát**

- Cố định M, tìm Y

Khi các centers là cố định, ta có thể phát biểu bài toán tìm label cho từng điểm dữ liệu như sau:

Vì là bình phương khoảng cách từ tới center , ta có thể kết luận rằng mỗi điểm thuộc vào cluster có center gần nó nhất.

* Cố định Y, tìm M

Giả sử đã tìm được cluster cho từng điểm, hãy tìm center mới cho mỗi cluster để hàm mất mát đạt giá trị nhỏ nhất.

Bài toán tìm center cho mỗi cluster được rút gọn thành:

Tìm nghiệm bằng phương pháp giải đạo hàm bằng 0:

Đặt là hàm bên trong dấu argmin, ta có đạo hàm:

Từ kết quả đạo hàm, ta thấy tử số là tổng các điểm thuộc vào center , mẫu là số phần tử các điểm dữ liệu thuộc cluster j. Hay nói cách khác,  **mới là trung bình cộng của các điểm trong cluster j.**

**d. Thuật toán tối ưu hàm mất mát**

Đầu vào: Dữ liệu X và số lượng cluster cần tìm K

Đầu ra: các center M và các label vector cho từng điểm dữ liệu X

1. Chọn K điểm bất kỳ làm các center ban đầu.
2. Phân mỗi điểm dữ liệu vào cluster có center gần nó nhất.
3. Nếu việc gán dữ liệu vào từng cluster ở bước 2 không thay đổi so với vòng lặp trước nó thì ta dừng thuật toán.
4. Cập nhật center cho từng cluster bằng cách lấy trung bình cộng của tất các các điểm dữ liệu đã được gán vào cluster đó sau bước 2.
5. Quay lại bước 2.

**e. Hạn chế**

- Chúng ta cần biết số lượng cluster cần clustering

- Nghiệm cuối cùng phụ thuộc vào các centers được khởi tạo ban đầu: cách khắc phục là chạy nhiều lần với các center ban đầu khác nhau rồi lấy hàm mất mát đạt giá trị nhỏ nhất.

- Các cluster cần có só lượng điểm gần bằng nhau.

- Các cluster cần có dạng hình tròn.

-Khi một cluster nằm phía trong 1 cluster khác.

**3. Một số ứng dụng thuật toán K-means Clustering**

**a. Phân nhóm chữ viết tay**

- Dùng một bộ cơ sở dữ liệu lớn là **MNIST** là cơ sở dữ liệu lớn nhất về chữ số viết tay. Coi mỗi bức ảnh chữ số viết tay là một điểm dữ liệu (trước đó phải vector hóa nó về dạng vector, ban đầu nó đang ở dạng ma trận) sử dụng thuật toán K-means, ta thu được các chữ số được gán nhãn vào những center nào.

**b. Tách vật thể trong ảnh**



Dùng K-means để tách thành 3 phần khác nhau là khăn, màu da và màu đen nền, xem như là 3 cluster. Sau khi gán nhãn được cho mỗi pixel thì thay mỗi pixel thành center của cluster đó (Sử dụng thuật toán [Vector Quantization (VQ)](https://en.wikipedia.org/wiki/Vector_quantization), ta thu được ảnh:



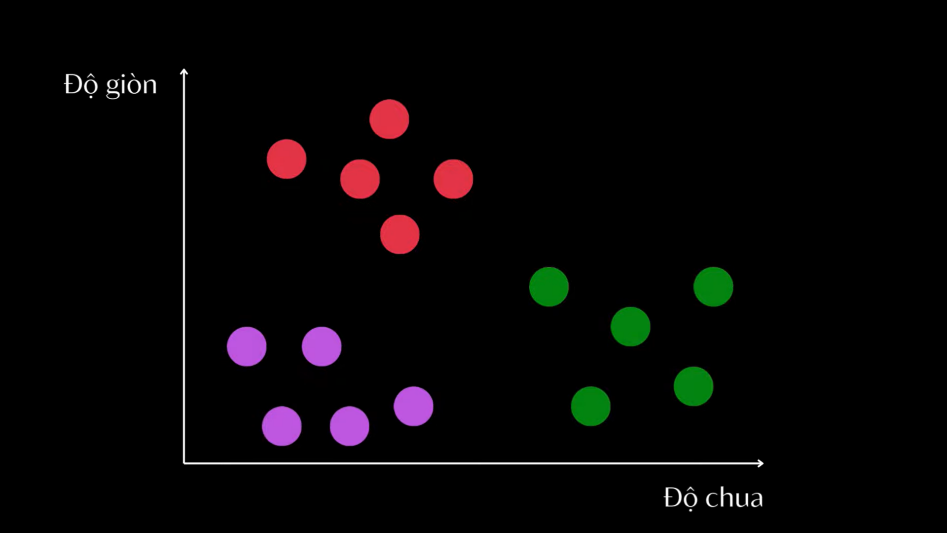
**c. Image Compression (nén ảnh và nén dữ liệu nói chung)**

Với một bức ảnh ban đầu với rất nhiều màu, ta lấy ra các màu chính yếu, và phân nhóm cho nó thì dù chất lượng rõ ràng đã giảm đi nhiều nhưng nhìn chung là vẫn chấp nhận được.



1. **Thuật toán K-nearest neighbors**
2. **Giới thiệu bài toán**

Ta biết rằng mỗi loại trái cây có độ chua và giòn khác nhau. Ví dụ như táo thì có độ chua thấp, độ giòn thì cao, chanh thì hoàn toàn ngược lại,… Giả sử ta đo được các chỉ số này và dữ liệu được biểu hiện dưới bảng sau:



Chú thích:

Đỏ: Táo

Xanh: Chanh

Tím: Thăng long

Nếu ta có một điểm dữ liệu mới và cần xác định nó là loại quả gì, KNN sẽ giải quyết bài toán đó.

1. **Phân tích toán học**

Với điểm dữ liệu mới, nó sẽ tính tất cả các khoảng cách từ nó đến các điểm còn lại (khoảng cách Euclid chẳng hạn), sau đó lấy ra K hàng xóm gần nó nhất (tùy chọn). Trong K điểm dữ liệu gần nhất, ta sẽ thuận theo đa số mà gán nhãn cho dữ liệu mới đó.

Lưu ý:

* Việc chọn K ảnh hưởng rất nhiều đến khả năng dự đoán của mô hình. Nếu K quá nhỏ dễ bị overfitting, K quá lớn thì mất các đặc trưng cục bộ quan trọng của dữ liệu.

1. **Cách lựa chọn K**

Chia toàn bộ dữ liệu thành 2 tập là tập dữ liệu test và dữ liệu train, sau đó thử với các K khác nhau và đánh giá độ chính xác để lựa chọn K phù hợp nhất.