Họ và tên: Mai Minh Quân

MSSV: 20225661

# Naïve Bayes

**Source:**

* Naive Bayes in Python - Machine Learning From Scratch

1. Giới thiệu về Naive Bayes

Naive Bayes là một thuật toán học máy có giám sát được sử dụng cho các bài toán phân loại. Thuật toán này dựa trên Định lý Bayes, một nguyên tắc xác suất cơ bản mô tả mối quan hệ giữa các xác suất có điều kiện.

Định lý Bayes:

Trong ngữ cảnh phân loại:

: Lớp (class).

: Véc-tơ đặc trưng (feature vector).

1. Giả định của Naive Bayes

Tên gọi "Naive" xuất phát từ giả định rằng tất cả các đặc trưng đầu vào đều độc lập với nhau. Điều này có nghĩa là sự xuất hiện của một đặc trưng không ảnh hưởng đến sự xuất hiện của bất kỳ đặc trưng nào khác.

* Ví dụ: Trong bài toán dự đoán xem một người có đi chạy bộ hay không, các đặc trưng "trời nắng" và "sức khỏe tốt" được giả định là độc lập.
* Trong thực tế, giả định này thường không đúng, nhưng Naive Bayes vẫn hoạt động hiệu quả trong nhiều trường hợp.

1. Công thức và Giải thích

* : Xác suất hậu nghiệm (posterior probability), xác suất một mẫu dữ liệu thuộc lớp khi biết đặc trưng .
* : Xác suất điều kiện lớp (class conditional probability), xác suất xuất hiện đặc trưng khi biết lớp Y.
* : Xác suất tiên nghiệm của lớp (prior probability of ), xác suất xuất hiện lớp trong tập dữ liệu.
* : Xác suất tiên nghiệm của đặc trưng (prior probability of ).

Mục tiêu của Naive Bayes là tìm lớp Y có xác suất hậu nghiệm lớn nhất.

Vì P(X) là hằng số đối với tất cả các lớp, ta có thể bỏ qua nó trong quá trình so sánh.

Để tránh vấn đề tràn số do nhân nhiều xác suất nhỏ, ta thường sử dụng logarit.

1. Tính toán Xác suất

Xác suất tiên nghiệm được tính bằng tần suất xuất hiện của lớp trong tập dữ liệu huấn luyện.

Xác suất điều kiện lớp thường được mô hình hóa bằng phân phối Gaussian (phân phối chuẩn).

* : Phương sai của đặc trưng trong lớp .
* : Kì vọng của đặc trưng trong lớp .

# Support Vector Machine

**Source:**

* Support Vector Machine in Python - Machine Learning From Scratch

1. Giới thiệu về Support Vector Machine (SVM)

Support Vector Machine (SVM) là một thuật toán học máy có giám sát mạnh mẽ, được sử dụng rộng rãi cho cả các bài toán phân loại và hồi quy. Trong lĩnh vực phân loại, SVM đặc biệt hiệu quả trong việc tìm ra ranh giới quyết định tối ưu để phân tách các lớp dữ liệu.

1. Ranh giới quyết định tuyến tính và Hyperplane

SVM hoạt động dựa trên ý tưởng sử dụng một mô hình tuyến tính để xác định ranh giới quyết định. Ranh giới quyết định này, trong không gian n chiều (n là số chiều của dữ liệu), được gọi là hyperplane.

1. Hyperplane tối ưu và Margin

Mục tiêu của SVM là tìm ra hyperplane "tốt nhất" để phân tách các lớp dữ liệu. "Tốt nhất" ở đây được định nghĩa là hyperplane tạo ra sự phân tách lớn nhất giữa các lớp, hay nói cách khác, có "margin" lớn nhất.

* Margin**:** Khoảng cách giữa hyperplane và các điểm dữ liệu gần nhất của mỗi lớp. Các điểm dữ liệu này được gọi là "vector hỗ trợ" (support vectors).
* SVM tìm cách tối đa hóa margin, tức là tìm hyperplane sao cho khoảng cách đến các vector hỗ trợ của mỗi lớp là lớn nhất.

A diagram of a mathematical equation

AI-generated content may be incorrect.

1. Biểu diễn toán học của Hyperplane và Margin

* Phương trình Hyperplane:
* : Véc-tơ trọng số, xác định hướng của hyperplane.
* : Véc-tơ đặc trưng của một điểm dữ liệu.
* : Bias, xác định vị trí của hyperplane.
* Điều kiện phân loại:
  + cho các điểm thuộc lớp .
  + cho các điểm thuộc lớp .
* Kết hợp điều kiện:
  + : Nhãn lớp, nhận giá trị hoặc .

1. Hàm Chi phí (Cost Function) và Tối ưu hóa

Để tìm ra các giá trị tối ưu cho và , SVM sử dụng hàm chi phí và phương pháp tối ưu hóa.

* Hàm Chi phí Hinge Loss:
  + Hinge Loss đo lường "mức độ vi phạm" của một điểm dữ liệu so với điều kiện phân loại.
  + Hinge Loss bằng 0 nếu điểm dữ liệu được phân loại đúng và nằm ngoài margin.
* Tối ưu hóa Margin: Để tối đa hóa margin, SVM cần cực tiểu hóa độ lớn của véc-tơ trọng số ().
* Hàm Chi phí Tổng quát:
  + : Tham số điều chỉnh, cân bằng giữa mục tiêu phân loại đúng và tối đa hóa margin.

1. Phương pháp Gradient Descent và Đạo hàm

SVM sử dụng phương pháp gradient descent để tìm ra các giá trị của và cực tiểu hóa hàm chi phí.

* Đạo hàm của Hàm Chi phí:
  + Trường hợp 1: (Hinge Loss = 0)
  + Trường hợp 2:
* Cập nhật Tham số:
  + : Tốc độ học (learning rate), kiểm soát kích thước bước cập nhật.