1. Perceptron

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.

|  |  |
| --- | --- |
| https://machinelearningcoban.com/assets/pla/pla1.png | https://machinelearningcoban.com/assets/pla/pla2.png |

Có thể hiểu rằng chúng ta cần tìm lãnh thổ của mỗi class bằng cách tìm biên giới (boundary) giữa hai lãnh thổ.

Cơ chế hoạt động:

B1: Chọn ngẫu nhiên một vector hệ số và bias ­­ (thường bằng 0 hoặc giá trị nhỏ ngẫu nhiên)

B2: Duyệt ngẫu nhiên từng , tính được

* Nếu được phân lớp đúng, tức là , chúng ta không cần làm gì
* Nếu được phân lớp lỗi, cập nhật và theo công thức:

hay

B3: Kiểm tra xem có bao nhiêu điểm bị phân lớp lỗi. Nếu không còn điểm nào, dừng thuật toán. Nếu còn, quay lại bước 2.

Ưu điểm:

* Đơn giản và dễ hiểu
* Hiệu quả trên những dữ liệu tuyến tính phân tách
* Tính toán nhanh chóng
* Cơ sở cho các thuật toán nơ-ron phức tạp hơn

Nhược điểm:

* Không xử lý tốt dữ liệu phi tuyến tính phân tách
* Khả năng hội tụ phụ thuộc vào dữ liệu
* Thiếu khả năng tổng quát
* Cập nhật trọng số không tối ưu

\*\*Chú thích:

* Bias (độ dịch) ­­ là một số vô hướng, giúp điều chỉnh ranh giới phân loại, hạn chế việc biên giới của hai class đi qua gốc tọa độ
* Trong đó:
* : vector trọng số
* : bias, một số vô hướng
* : là tích vô hướng giữa vector trọng số và vector đặc trưng (còn gọi là tổng có trọng số).
* : trả về 1 nếu , trả về -1 nếu
* : hệ số học (learning rate),

1. ID3

ID3 (Iterative Dichotomiser 3): Là một thuật toán phân loại theo cách tiếp cận tham lam bằng cách chọn thuộc tính tốt nhất nhằm mang lại Information Gain (IG - lợi ích của thông tin) tối đa hoặc Entropy tối thiểu (entropy dùng để chỉ trạng thái ngẫu nhiên hoặc không có trật tự).

Bản chất của thuật toán ID3 là xây dựng một cây quyết định nhằm phân loại dữ liệu dựa trên các thuộc tính của nó.

Cơ chế hoạt động:

Sử dụng hàm “entropy“ để đo độ tinh khiết (purity) / độ vẩn đục (impurity) của 1 phép phân chia

B1: Tại root node (nút gốc), tính entropy để đo lường mức độ hỗn loạn của toàn bộ tập dữ liệu trước khi nó được phân chia theo bất kỳ thuộc tính nào.

B2: Khi đã phân chia tập dữ liệu 𝑆 dựa trên một thuộc tính x, chúng ta sẽ có K tập con (child nodes), mỗi tập tương ứng với một giá trị cụ thể của thuộc tính x. Ta tính entropy tại child node:

B3: Information Gain(IG) là lượng thông tin thu được sau khi phân chia tập dữ liệu dựa trên một thuộc tính x, tức là nó đo lường sự giảm entropy khi chia dữ liệu theo x. Công thức tính IG dựa trên x là:

Và tại mỗi node thuộc tính được chọn được xác định dựa vào:

Ưu điểm:

* Đơn giản, dễ hiểu, tốc độ nhanh
* Không cần tham số đặc biệt

Nhược điểm:

* Có xu hướng tạo cây quá phức tạp (overfitting)
* Không hoạt động tốt với dữ liệu liên tục hoặc dữ liệu có quá nhiều thuộc tính phân loại

1. Neural Networks

Mạng nơ-ron nhân tạo (Artificial Neural Network - ANN) là một mô hình mô phỏng cách mà hệ thống thần kinh con người hoạt động. ANN gồm các nơ-ron kết nối với nhau thông qua các trọng số. Khi một dữ liệu được đưa vào, nó được lan truyền qua các lớp nơ-ron, qua quá trình huấn luyện, mô hình học cách phân loại và dự đoán dữ liệu mới.

Cấu trúc:

* Lớp đầu vào (Input layer): Chứa các nơ-ron đầu vào, mỗi nơ-ron tương ứng với một đặc trưng (feature) từ dữ liệu.
* Lớp ẩn (Hidden layer): Mạng nơ-ron có thể có một hoặc nhiều lớp ẩn, mỗi lớp bao gồm nhiều nơ-ron. Các nơ-ron ở lớp ẩn được kết nối với các nơ-ron ở lớp đầu vào và lớp đầu ra. Chúng xử lý thông tin và tạo ra các đặc trưng mới dựa trên trọng số liên kết giữa các nơ-ron.
* Lớp đầu ra (Output layer): Dự đoán kết quả cuối cùng của bài toán dựa trên các thông tin được xử lý từ lớp ẩn.

Hàm kích hoạt (Activation Function):

* Sigmoid: Được sử dụng chủ yếu trong các bài toán phân loại nhị phân.
* ReLU (Rectified Linear Unit): Thường được sử dụng trong các mạng nơ-ron có nhiều lớp vì khả năng khắc phục hiện tượng mất mát gradient trong quá trình huấn luyện.
* Softmax: Dùng trong các bài toán phân loại đa lớp như bài toán dự đoán thời tiết này.

Hàm mất mát (Loss Function):

* Hàm mất mát là hàm giúp đo lường sự khác biệt giữa giá trị dự đoán và giá trị thực tế.
* Đối với bài toán phân loại, đặc biệt là phân loại đa lớp, hàm mất mát thường được sử dụng là Cross-Entropy Loss. Hàm này giúp tối ưu hóa mạng bằng cách giảm thiểu sự khác biệt giữa nhãn dự đoán và nhãn thực tế.

Lan truyền ngược (Backpropagation):

* Đây là quá trình tối ưu hóa mạng nơ-ron, giúp mạng điều chỉnh trọng số của các liên kết giữa các nơ-ron dựa trên giá trị của hàm mất mát. Lan truyền ngược sử dụng thuật toán Gradient Descent để điều chỉnh trọng số nhằm giảm thiểu hàm mất mát sau mỗi lần huấn luyện.

Ưu điểm:

* Khả năng học tốt các mối quan hệ phi tuyến
* Khả năng xử lý dữ liệu lớn
* Tính linh hoạt
* Tự động học các đặc trưng
* Khả năng tinh chỉnh và tinh vi

Nhược điểm:

* Yêu cầu về tài nguyên cao
* Thời gian huấn luyện dài
* Khó khăn trong việc giải thích
* Rủi ro overfitting
* Cần tinh chỉnh tham số kỹ
* Nhạy cảm với dữ liệu đầu vào

1. Mô tả bài toán

Vai trò của dự báo thời tiết:

Dự báo thời tiết đóng vai trò quan trọng trong nhiều lĩnh vực của đời sống, từ nông nghiệp, giao thông vận tải đến du lịch, và thậm chí cả việc bảo vệ sức khỏe. Nắm bắt được thông tin về thời tiết giúp con người chủ động trong mọi hoạt động, từ việc lên kế hoạch cho các hoạt động ngoài trời đến việc phòng tránh thiên tai.

Mô tả bài toán:

Với sự phát triển của khoa học máy tính và học máy, bọn mình đã chọn ba thuật toán Perceptron Learning, ID3 và Neural Networks để thực hiện một bài toán về dự báo thời tiết. Bọn mình sẽ sử dụng các thông số thời tiết phổ biến như precipitation (lượng mưa), temp\_max (nhiệt độ cao nhất), temp\_min (nhiệt độ thấp nhất), wind (sức gió) để dự báo tình trạng thời tiết

1. Tham số đánh giá

TP: True positive (dương tính thật) – dự đoán trùng với giá trị thật

FP: Flase positive (dương tính giả) – dự đoán dương nhg thực tế là âm

TN: True negative (âm tính thật) – dự đoán trùng với giá trị thật

FN: False negative (âm tính giả) – dự đoán âm nhg thực tế là dương

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | Actual | |
|  |  | Positive(1) | Negative(0) |
| Predicted | Positive(1) | TP | FP |
| Negative(0) | FN | TN |

* Accuracy ( Độ chính xác)

Công thức: =

Là một phép đo độ chính xác dễ hiểu và đơn giản, nhưng không thể phản ánh đúng với dữ liệu không cân bằng, tức là một dữ liệu vượt trội hơn các dữ liệu còn lại

* Precision (Độ chính xác dương tính)

Công thức:

Dùng phương pháp này khi quan tâm FP nhiều hơn, tối thiếu hóa FP. Ví dụ, trong việc phát hiện thư rác, không muốn thư bình thường bị gắn nhầm là thư rác.

* Recall

Công thức:

Dùng phương pháp này khi FN quan trọng hơn, mục đích là để tối thiểu hóa FN (min). Ví dụ, trong y học, khi phát hiện bệnh ung thư, không muốn bỏ sót bất kỳ bệnh nhân nào.

* F1 score

Công thức:

Một phương pháp trung bình, hài hòa của precision và recall

Được sử dụng trong một số TH: cân bằng hoặc không cân bằng

F1 score tùy vào dữ liệu, và vấn đề muốn dự đoán:

0.9: very good

0.8 – 0.9: good

0.5 – 0.8: accept

<0.5: Not good

* Macro average:

Macro average đơn giản là tính trung bình của các chỉ số (precision, recall, F1 score, v.v.) cho mỗi lớp mà không quan tâm đến kích thước của từng lớp. Mỗi lớp đều có trọng số bằng nhau.

Đo lường hiệu suất tổng thể của mô hình, mà không quan tâm đến sự phân bổ không đều giữa các lớp. Mỗi lớp được coi trọng như nhau.

* Weight avg

Weighted average cũng tính toán trung bình của các chỉ số cho từng lớp, nhưng có tính đến kích thước của mỗi lớp trong tập dữ liệu. Các lớp lớn sẽ có trọng số cao hơn và ảnh hưởng nhiều hơn đến giá trị trung bình.

Sử dụng khi bạn có tập dữ liệu bị mất cân bằng, và bạn muốn chỉ số tổng thể phản ánh đúng tầm quan trọng của từng lớp dựa trên kích thước của chúng. Các lớp lớn sẽ có ảnh hưởng lớn hơn đến giá trị cuối cùng.

Cách tính:

* Tính precision, recall, F1 score cho mỗi lớp.
* Tính trung bình có trọng số của các giá trị này, trong đó trọng số là tỷ lệ số lượng mẫu của mỗi lớp so với tổng số mẫu.

Công thức:

Weighted precision:

Trong đó: là số lượng mẫu của lớp i

Ví dụ:

Giả sử bạn có 3 lớp với số lượng mẫu lần lượt là 50, 30, và 20 và precision của từng lớp là 0.9, 0.8 và 0.7. Weighted Precision sẽ được tính như sau: