|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Phương pháp | Input | Output | Bài toán ưu (hoặc hàm mất mát) | Cách thực hiện |
| Linear Regression (Hồi quy tuyến tính) | Tập dữ liệu huấn luyện đã được gán nhãn (Xtrain,ytrain) | Hàm tuyến tính có dạng f(xi)=wxi + w0 |  | Tìm giá trị tối ưu của w giải phương trình đạo hàm của hàm mất mát theo w bằng 0 |
| Perceptron | Tập dữ liệu huấn luyện đã được gán nhãn (Xtrain,ytrain) | Siêu phẳng sao cho tất cả các mẫu cùng nhãn thì cùng thuộc một phía của siêu phẳng f(x,w)=wT.x |  | **Dùng gradient descent.**  1. Tại thời điểm t = 0, chọn ngẫu nhiên một vector hệ số w0  2. Tại thời điểm t, nếu không có điểm dữ liệu nào bị phân lớp lỗi, dừng thuật toán.  3. Giả sử xi là một điểm bị phân lớp lỗi. Cập nhật wt+1 = wt + 𝑦i xi  4. Thay đổi t = t + 1 rồi quay lại Bước 2 |
| SVM | Tập dữ liệu huấn luyện đã được gán nhãn (Xtrain,ytrain) | Tìm một siêu phẳng phân chia dữ liệu thành 2 lớp:  -Siêu phẳng sao cho tất cả các mẫu cùng nhãn thì cùng thuộc một phía của siêu phẳng f(x,w)=wT.x  -Khoảng cách từ điểm gần nhất của mỗi lớp tới đường phân chia là như nhau (margin, lề).  -Margin này phải là cực đại |  | **Tìm w và b sao cho hàm mất mát đạt giá trị nhỏ nhất**    Bài toán tối ưu này là bài toán lồi, và là một quadratic programming (phương trình bậc 2)  Suy ra nghiệm cho SVM là duy nhất  Để giải bài toán này, người ta thường giải bài toán đối ngẫu Lagrange |
| Cây quyết định | Tập dữ liệu huấn luyện đã được gán nhãn (Xtrain,ytrain) | Đưa ra 1 cây quyết định dự đoán nhãn dựa theo các thuộc tính, với mỗi node là một thuôc tính của dữ liệu | Hàm entropy: | Tìm các cách phân chia hợp lý (thứ tự chọn thuộc tính hợp lý) sao cho hàm mất mát cuối cùng đạt giá trị càng nhỏ càng tốt  1. Tạo một nút N  2. Nếu tất cả các mẫu thuộc cùng lớp C thì N được gán nhãn C; dừng thuật toán;  3. Nếu A là rỗng thì N được gán nhãn C là nhãn phổ biến nhất trong S; dừng thuật toán;  4. Chọn a∊A, có độ đo information gain cao nhất; Gán nhãn N theo a;  5. Với mỗi giá trị v của a:  a. Phát triển 1 nhánh từ N với điều kiện a=v;  b. Đặt Sv là tập con của S với a=v;  c. Nếu Sv là rỗng thì gắn một lá có nhãn phổ biến nhất trong S;  d. Ngược lại gắn một nút được tạo bởi GenDecTree(Sv , A-a) |
| Neural Network | Tập dữ liệu huấn luyện đã được gán nhãn (Xtrain,ytrain), kiến trúc mạng nơ ron (số lớp ẩn, số nơron của mỗi lớp ẩn), hàm kích hoạt (activation function), hàm mất mát (loss function) | Bộ vector trọng số của các liên kết giữa các nơ ron (W) để hàm mất mát đạt giá trị tối ưu |  | -Đạo hàm của hàm mất mát đạt giá trị nhỏ nhất, càng nhỏ mô hình càng tốt  -Phương pháp phổ biến nhất để tối ưu MLP vẫn là Gradient Descent (GD)  - Để áp dụng GD, chúng ta cần tính được gradient của hàm mất mát theo từng ma trận trọng số Wl , và vector bias bl  w = w-alpha\*L’(x,w)  tính đạo hàm riêng của hàm mất mát của các trọng số ở tầng (l+1) của mạng bằng phương pháp lan truyền ngược |
| Logistic Regression | Tập dữ liệu huấn luyện đã được gán nhãn (Xtrain,ytrain) | Được thể hiện dưới dạng xác suất (probability)  Đầu ra dự đoán: |  | Trực tiếp tối ưu hàm số này theo w khó  Một phương pháp thường được sử dụng đó là lấy logarit tự nhiên (cơ số e) của likelihood function biến phép nhân thành phép cộng và để tránh việc số quá nhỏ  Sau đó lấy ngược dấu để một hàm và coi nó là hàm mất mát. Lúc này bài toán tìm giá trị lớn nhất (maximum likelihood) trở thành bài toán tìm giá trị nhỏ nhất của hàm mất mát (hàm này còn được gọi là negative log likelihood):    Chúng ta sử dụng phương pháp Gradient Descent để tìm w |
| K-means | Tập dữ liệu huấn luyện chưa được gán nhãn (Xtrain).  Dữ liệu X và số lượng cluster cần tìm K | Các center M và label vector cho từng điểm dữ liệu Y |  | Giải bài toán (\*) là:  Xen kẽ giải 𝑌 và 𝑀 khi biến còn lại được cố định  Đây là thuật toán lặp, cũng là kỹ thuật phổ biến khi giải bài toán tối ưu  Sau khi giải, chúng ta có được Y và M |

|  |  |  |
| --- | --- | --- |
| Bài toán | Tập dữ liệu | Các phương pháp |
| Hồi quy | Tập dữ liệu đã được gán nhãn (Xtrain, ytrain). Miền giá trị của tập nhãn ytrain là liên tục và có vô số giá trị | Linear Regression, Lasso, Ridgre, K-Fold |
| Phân lớp | Tập dữ liệu đã được gán nhãn (Xtrain, ytrain). Miền giá trị của tập nhãn ytrain là rời rạc và hữu hạn | SVM, perceptron, cây quyết định, Neural Network, Logistic Regression |
| Phân Cụm | Tập dữ liệu huấn luyện không được gán nhãn (Xtrain) | K-means |

|  |  |  |  |
| --- | --- | --- | --- |
| Kiểu học | Tập dữ liệu | Bài toán | Phương pháp |
| Học có giám sát | Tập dữ liệu huấn luyện được gán nhãn (Xtrain, ytrain) | Hồi quy, phân lớp | Linear Regression, Lasso, Ridgre, K-Fold, SVM, perceptron, cây quyết định, Neural Network, Logistic Regression |
| Học không có giám sát | Tập dữ liệu huấn luyện không được gán nhãn (Xtrain) | Phân cụm | K-means |