Thuật toán K – Nearest – Neighbors

1. Giới thiệu

\_ ***K-nearest neighbor*** là một trong những thuật toán supervised-learning đơn giản nhất (mà hiệu quả trong một vài trường hợp) trong Machine Learning. Khi training, thuật toán này **không học** một điều gì từ dữ liệu training (đây cũng là lý do thuật toán này được xếp vào loại ***lazy learning***), mọi tính toán được thực hiện khi nó cần dự đoán kết quả của dữ liệu mới. K-nearest neighbor có thể áp dụng được vào cả hai loại của bài toán Supervised learning là Classification và Regression.

\_ Với KNN, trong bài toán Classification, label của một điểm dữ liệu mới (hay kết quả của câu hỏi trong bài thi) được suy ra trực tiếp từ K điểm dữ liệu ***gần nhất*** trong training set. Label của một test data có thể được quyết định bằng ***major voting*** (bầu chọn theo số phiếu) giữa các điểm gần nhất, hoặc nó có thể được suy ra bằng cách đánh trọng số khác nhau cho mỗi trong các điểm gần nhất đó rồi suy ra label. Chi tiết sẽ được nêu trong phần tiếp theo.

\_ Trong bài toán Regresssion, đầu ra của một điểm dữ liệu sẽ bằng chính đầu ra của điểm dữ liệu đã biết gần nhất (trong trường hợp K=1), hoặc là trung bình có trọng số của đầu ra của những điểm gần nhất, hoặc bằng một mối quan hệ dựa trên khoảng cách tới các điểm gần nhất đó.

1. Khoảng cách trong không gian vector

\_ Trong không gian một chiều, khoảng cách giữa hai điểm là trị tuyệt đối giữa hiệu giá trị của hai điểm đó. Trong không gian nhiều chiều, khoảng cách giữa hai điểm có thể được định nghĩa bằng nhiều hàm số khác nhau, trong đó độ dài đường thằng nổi hai điểm chỉ là một trường hợp đặc biệt trong đó.

1. Các bước thực hiện

\_ B1: Ta có D là tập các điểm dữ liệu đã được gắn nhãn và A là dữ liệu chưa được phân loại.

\_ B2: Đo khoảng cách (Euclidian, Manhattan, Minkowski, Minkowski hoặc Trọng số) từ dữ liệu mới A đến tất cả các dữ liệu khác đã được phân loại trong D.

\_ B3: Chọn K (K là tham số mà bạn định nghĩa) khoảng cách nhỏ nhất.

\_ B4: Kiểm tra danh sách các lớp có khoảng cách ngắn nhất và đếm số lượng của mỗi lớp xuất hiện.

\_ B5: Lấy đúng lớp (lớp xuất hiện nhiều lần nhất).

\_ B6: Lớp của dữ liệu mới là lớp mà bạn đã nhận được ở bước 5.

1. Chuẩn hóa dữ liệu

\_ Khi có một thuộc tính trong dữ liệu (hay phần tử trong vector) lớn hơn các thuộc tính khác rất nhiều (ví dụ thay vì đo bằng cm thì một kết quả lại tính bằng mm), khoảng cách giữa các điểm sẽ phụ thuộc vào thuộc tính này rất nhiều. Để có được kết quả chính xác hơn, một kỹ thuật thường được dùng là **Data Normalization** (chuẩn hóa dữ liệu) để đưa các thuộc tính có đơn vị đo khác nhau về cùng một khoảng giá trị, thường là từ 0 đến 1, trước khi thực hiện KNN. Các kỹ thuật chuẩn hóa được áp dụng với không chỉ KNN mà còn với hầu hết các thuật toán khác.

1. Ưu điểm và nhược điểm

* Ưu điểm:

\_ Độ phức tạp tính toán của quá trình training là bằng 0.

\_ Việc dự đoán kết quả của dữ liệu mới rất đơn giản.

\_ Không cần giả sử gì về phân phối của các class.

* Nhược điểm:

\_ KNN rất nhạy cảm với nhiễu khi K nhỏ.

\_ KNN là một thuật toán mà mọi tính toán đều nằm ở khâu test. Trong đó việc tính khoảng cách tới **từng *điểm dữ liệu*** trong training set sẽ tốn rất nhiều thời gian, đặc biệt là với các cơ sở dữ liệu có số chiều lớn và có nhiều điểm dữ liệu. Với K càng lớn thì độ phức tạp cũng sẽ tăng lên. Ngoài ra, việc lưu toàn bộ dữ liệu trong bộ nhớ cũng ảnh hưởng tới hiệu năng của KNN.

1. Tăng tốc cho KNN

\_ Sử dụng Approximate Nearest Neighbor

\_ K-D Tree

\_ Ball Tree