**Lesson 6: Classification With K Nearest Neighbour & Support Vector Machine**

**1.KNN**

KNN là một thuật toán được xếp vào loại lazy learning. Vì thực sự KNN sẽ không có tính toán , hay học ra điều gì từ dữ liệu đầu vào cả, KNN chỉ ghi nhớ dữ liệu đó và bắt dầu tính toán khi nó cần dự đoán kết quả đầu ra. ( Model bình thường của chúng ta hoạt động theo quy trình : đưa dữ liệu vào => Tính toán , tìm ra quy luật , ... => Sau đó dùng quy luật / kiến thức được học từ dữ liệu được đưa vào => dự đoán kết quả đầu ra )

KNN là thuật toán đi tìm đầu ra của một điểm dữ liệu mới chỉ dựa trên thông tin của K điểm dữ liệu gần nhất trong tập huấn luyện

Ví dụ :

Có một anh bạn chuẩn bị đến ngày thi cuối kỳ. Vì môn này được mở tài liệu khi thi nên anh ta không chịu ôn tập để hiểu ý nghĩa của từng bài học và mối liên hệ giữa các bài. Thay vào đó, anh thu thập tất cả các tài liệu trên lớp, bao gồm ghi chép bài giảng (lecture notes), các slides và bài tập về nhà + lời giải. Để cho chắc, anh ta ra thư viện và các quán Photocopy quanh trường mua hết tất cả các loại tài liệu liên quan (khá khen cho cậu này chịu khó tìm kiếm tài liệu). Cuối cùng, anh bạn của chúng ta thu thập được một chồng cao tài liệu để mang vào phòng thi.

Vào ngày thi, anh tự tin mang chồng tài liệu vào phòng thi. Aha, đề này ít nhất mình phải được 8 điểm. Câu 1 giống hệt bài giảng trên lớp. Câu 2 giống hệt đề thi năm ngoái mà lời giải có trong tập tài liệu mua ở quán Photocopy. Câu 3 gần giống với bài tập về nhà. Câu 4 trắc nghiệm thậm chí cậu nhớ chính xác ba tài liệu có ghi đáp án. Câu cuối cùng, 1 câu khó nhưng anh đã từng nhìn thấy, chỉ là không nhớ ở đâu thôi.

Kết quả cuối cùng, cậu ta được 4 điểm, vừa đủ điểm qua môn. Cậu làm chính xác câu 1 vì tìm được ngay trong tập ghi chú bài giảng. Câu 2 cũng tìm được đáp án nhưng lời giải của quán Photocopy sai! Câu ba thấy gần giống bài về nhà, chỉ khác mỗi một số thôi, cậu cho kết quả giống như thế luôn, vậy mà không được điểm nào. Câu 4 thì tìm được cả 3 tài liệu nhưng có hai trong đó cho đáp án A, cái còn lại cho B. Cậu chọn A và được điểm. Câu 5 thì không làm được dù còn tới 20 phút, vì tìm mãi chẳng thấy đáp án đâu - nhiều tài liệu quá cũng mệt!!

Không phải ngẫu nhiên mà tôi dành ra ba đoạn văn để kể về chuyện học hành của anh chàng kia. Hôm nay tôi xin trình bày về một phương pháp trong Machine Learning, được gọi là K-nearest neighbor (hay KNN), một thuật toán được xếp vào loại lazy (machine) learning (máy lười học). Thuật toán này khá giống với cách học/thi của anh bạn kém may mắn kia.

( nguồn : machine learning cơ bản, Vũ Hữu Tiệp )

Graphical user interface

Description automatically generated with medium confidence

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.

Một cách ngắn gọn, KNN là thuật toán đi tìm đầu ra của một điểm dữ liệu mới bằng cách chỉ dựa trên thông tin của K điểm dữ liệu trong training set gần nó nhất (K-lân cận), không quan tâm đến việc có một vài điểm dữ liệu trong những điểm gần nhất này là nhiễu

Ví dụ đơn giản khác : Bạn đi mua đất, tuy nhiên, miếng đất bạn mua lại nằm ở riêng 1 chỗ và bạn không biết giá trị thực của cái miếng đất bạn muốn mua là bao nhiêu. Vì vậy , bạn đi xung quanh khu vực đó để hỏi giá.

Khu dân cư gần đó 1km : Giá đất trung bình là 30 triệu 1 m vuông.

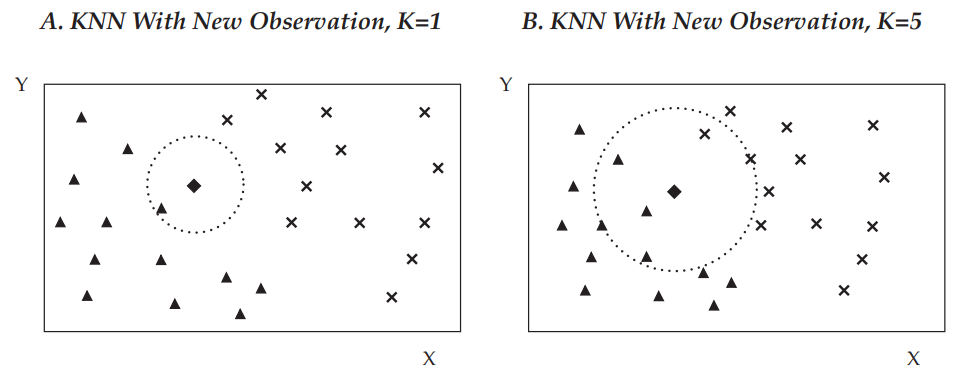
Khu dân cư cách đó 2km : Giá đất trung bình là 50 triệu 1 m vuông.

* Vì miếng đất của bạn gần với khu dân cư 1km kia hơn , nên miếng đất bạn mua sẽ có giá dao động trong khoảng 30 triệu 1 m vuông, chứ không thể 50 triệu 1 m vuông được.

Tuy nhiên, sẽ có nhiều vấn đề khác ảnh hưởng tới giá cả, ví dụ như quy hoạch tương lai, gần đường , chợ , ... những thứ đó gọi là nhiều và có thể ảnh hưởng tới kết quả dự đoán.

* Vì vậy mô hình KNN rất nhạy cảm với nhiễu.

Để xử lý nhiễu , chúng ta cần phải tăng các cụm cần xét xung quanh điểm dữ liệu lên => K Fold



Nếu k = 1, hình thoi sẽ được phân loại vào cùng loại với điểm dữ liệu gần nhất của nó (tức là hình tam giác trong bảng bên trái - bảng A).

- Bảng bên phải (bảng B) thể hiện trường hợp k = 5, thuật toán sẽ xem xét 5 điểm dữ liệu gần hình thoi nhất, đó là 3 hình tam giác và 2 hình chữ thập. Qui tắc quyết định là chọn phân loại có số lượng lớn nhất trong 5 điểm dữ liệu được xem xét. Vì vậy, trong trường hợp này, hình thoi cũng được xếp vào phân loại tam giác.

**2. SVM**

Trước khi tìm hiểu về SVM, chúng ta nhắc lại một tí về kiến thức hình học.

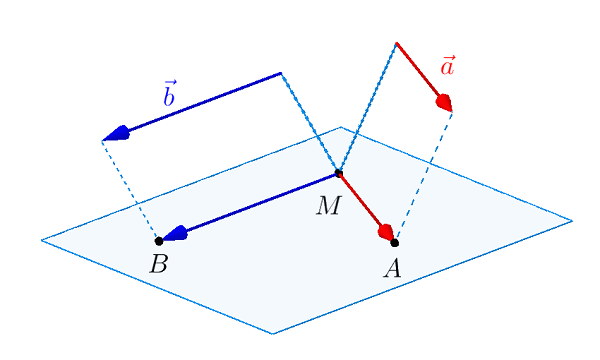
Chart, scatter chart

Description automatically generated

Véc tơ là gì

Véc tơ là những đoạn thẳng có hướng ( bắt đầu từ một điểm đi tới vô cực).

Giá trị của một vector là đường thẳng đi qua điểm đầu và điểm cuối của một vector đó.



Có thể thấy B , M , A là 3 điểm trên một mặt phẳng, đường thẳng BM , MA ,... hoàn toàn có thể được phóng thành các vector trong không gian.

Vậy phóng vector trong không gian để làm gì ? Có ý nghĩa gì trong bài toán phân lớp ?

Hãy thử vẽ 1 đường thẳng phân lớp 2 đối tượng Icon

Description automatically generated

Rất dễ đúng không ? Icon

Description automatically generated

( đường thẳng phân 2 lớp này gọi là hyper-plane)

Vậy còn như này thì sao ?

Chart, scatter chart

Description automatically generated

Dễ thôi, vẽ một trục Z , phóng nó thành không gian

Chart, scatter chart

Description automatically generated

Nếu nhìn bằng 2D

Chart, icon, scatter chart

Description automatically generated

Các bạn đã hình dung được vấn đề rồi đúng không ?

SVM là một thuận toán dùng máy móc tính ra 1 đường vector có thể phân 2 lớp thành 2 thái cực khác nhau.

Tuy nhiên trong thực tế, đôi khi mọi thứ không được hoàn hảo

Chart, scatter chart

Description automatically generated

Vẽ đường thẳng phân ra 2 điểm đen và xanh

Đương nhiên, chúng ta chỉ có thể vẽ như vầy

Chart, scatter chart

Description automatically generated

Nếu muốn nó hoàn hảo, thì nó không còn thẳng nữa

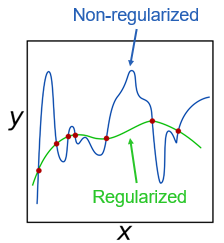
Chart, scatter chart

Description automatically generated

* Mô hình 1 nhanh , dễ , tuy nhiên có một vài outlier
* Mô hình 2 chính xác, nhưng phức tạp

Vấn đề này trong thực tế gọi là regularization parameter .

Regularization parameter



Regularization parameter là một tham số thể hiện độ chính xác của model. Nó ngược lại với loss function. Loss function càng thấp thì mô hình càng chính xác còn regularization càng cao thì mô hình càng chính xác. ( tham số regularization còn gọi là tham số C trong sklearn)

Ví dụ 1 : Mô hình này có regularization parameter thấp, vì vậy hyper plane của nó là một đường thẳng , và đường thẳng này không phải là tuyệt đối ( có 2 outlier )Chart, scatter chart

Description automatically generated

Ví dụ 2 : Mô hình có regularization cao nên đường thẳng cắt 2 class nó rất phức tạp, không phải chỉ là một đường thẳng

Chart, scatter chart

Description automatically generated

Ngoài ra còn có Gamma

Chart, scatter chart

Description automatically generated

Margin

Chart, scatter chart

Description automatically generated

Tóm tắt nguyên lý hoạt động

Chart, diagram

Description automatically generated

**2.2 SVR**

SVR vẫn là thuật toán Support Vector Machine , nhưng dùng cho bài toán Regression.

Đường thẳng Hyper-plane chúng ta cũng biết rồi, là một đường thẳng có dạng

Y = ax + b

Và cái đường hyper-plane là cái đường phân các data points ra thành 2 class, vậy cái đường đó nó nằm ở khoảng giữa, và có thể phân loại +Y hoặc -Y

Và khoảng cách từ hyper-plane đến Y hoặc -Y nó phải nằm trong khoảng cho phép , khoảng đó ký hiệu là margin of tolerance ( epsilon)

Chart, scatter chart

Description automatically generated

Non – Linear SVR

Diagram

Description automatically generated

( MAE và MSE sẽ được dùng để tính epsilon)

Practice : [Link](https://colab.research.google.com/drive/1kjZ4Jla8GLoovzwQnUiS1e0lcuoiH8Hi?usp=sharing)