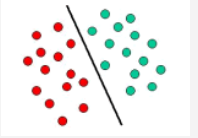
Support Vector Machine

# Support Vector Machine:

Support Vector Machine (SVM) là một phuơng pháp phân lớp dựa trên lý thuyết học thống kê, được đề xuất bởi Vapnik (1995).

Để đơn giản ta sẽ xét bài toán phân lớp nhị phân, sau đó sẽ mở rộng vấn đề ra cho bài toán phân nhiều lớp.

Xét một ví dụ của bài toán phân lớp như hình vẽ; ở đó ta phải tìm một đường thẳng sao cho bên trái nó toàn là các điểm đỏ, bên phải nó toàn là các điểm xanh. Bài toán mà dùng đường thẳng để phân chia này được gọi là phân lớp tuyến tính (linear classification).



Hàm tuyến tính phân biệt hai lớp như sau:

(1)

Trong đó:

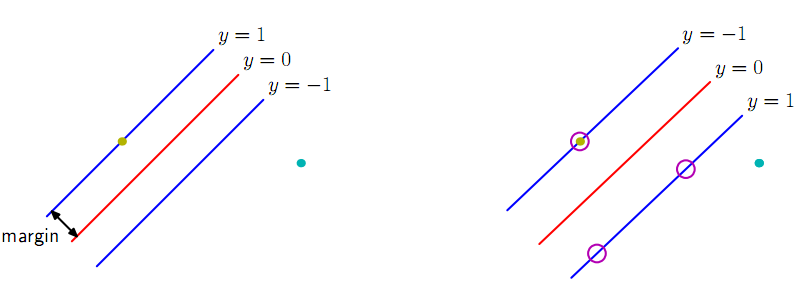
* là vector trọng số hay vector chuẩn của siêu phẳng phân cách, T là kí hiệu chuyển vị.
* là độ lệch
* là véc tơ đặc trưng, làm hàm ánh xạ từ không gian đầu vào sang không gian đặc trưng.

Tập dữ liệu đầu vào gồm N mẫu input vector {x1, x2,...,xN}, với các giá trị nhãn tương ứng là {t1,…,tN} trong đó .

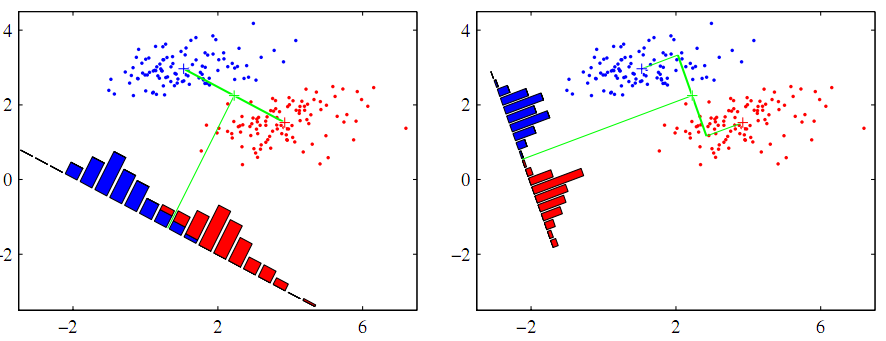
Lưu ý cách dùng từ ở đây: điểm dữ liệu, mẫu… đều được hiểu là input vector xi; nếu là không gian 2 chiều thì đường phân cách là đường thẳng, nhưng trong không gian đa chiều thì gọi đó là siêu phẳng.

Giả sử tập dữ liệu của ta có thể phân tách tuyến tính hoàn toàn (các mẫu đều được phân đúng lớp) trong không gian đặc trưng (feature space), do đó sẽ tồn tại giá trị tham số w và b theo (1) thỏa cho những điểm có nhãn và cho những điểm có , vì thế mà cho mọi điểm dữ liệu huấn luyện.

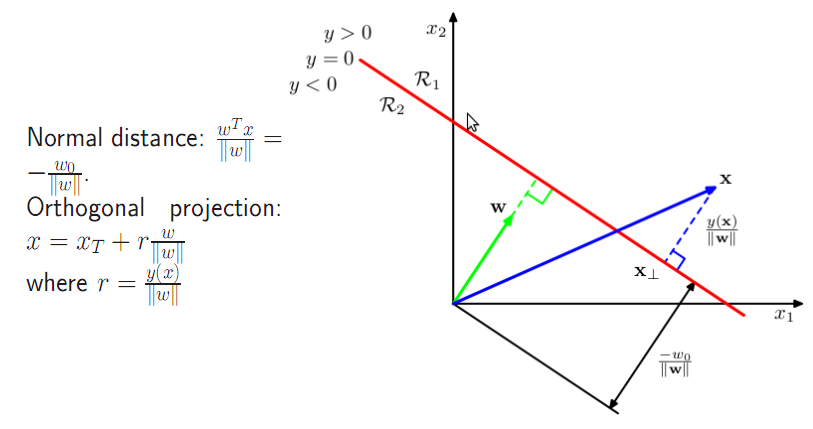
SVM tiếp cận giải quyết vấn đề này thông qua khái niệm gọi là lề, đường biên… (margin). Lề được chọn là khoảng cách nhỏ nhất từ đường phân cách đến mọi điểm dữ liệu hay là khoảng cách từ đường phân cách đến những điểm gần nhất.



Trong SVM, đường phân lớp tốt nhất chính là đường có khoảng cách margin lớn nhất (tức là sẽ tồn tại rất nhiều đường phân cách xoay theo các phương khác nhau, và ta chọn ra đường phân cách mà có khoảng cách margin là lớn nhất).



Ta có công thức tính khoảng cách từ điểm dữ liệu đến mặt phân cách như sau:



Do ta đang xét trong trường hợp các điểm dữ liệu đều được phân lớp đúng nên cho mọi n. Vì thế khoảng cách từ điểm xn đến mặt phân cách được viết lại như sau:

(2)

Lề là khoảng cách vuông góc đến điểm dữ liệu gần nhất xn từ tập dữ liệu, và chúng ta muốn tìm giá trị tối ưu của w và b bằng cách cực đại khoảng cách này. Vấn đề cần giải quyết sẽ được viết lại dưới dạng công thức sau:

(3)

Chúng ta có thể đem nhân tử ra ngoài bởi vì w không phụ thuộc n. Giải quyết vấn đề này một cách trực tiếp sẽ rất phức tạp, do đó ta sẽ chuyển nó về một vấn đề tương đương dễ giải quyết hơn. Ta sẽ scale và cho mọi điểm dữ liệu, từ đây khoảng cách lề trở thành 1, việc biến đổi này không làm thay đổi bản chất vấn đề.

(4)

Từ bây giờ, các điểm dữ liệu sẽ thỏa ràng buộc:

(5)

Vấn đề tối ưu yêu cầu ta cực đại được chuyển thành cực tiểu , ta viết lại công thức:

(6)

Việc nhân hệ số ½ sẽ giúp thuận lợi cho lấy đạo hàm về sau.

Lý thuyết Nhân tử Lagrange:

Vấn đề cực đại hàm f(x) thỏa điều kiện sẽ được viết lại dưới dạng tối ưu của hàm Lagrange như sau:

Trong đó x và λ phải thỏa điều kiện Karush-Kuhn-Tucker (KKT) như sau:

Nếu là cực tiểu hàm f(x) thì hàm Lagrange sẽ là

Để giải quyết bài toán trên, ta viết lại theo hàm Lagrange như sau:

(7)

Trong đó là nhân tử Lagrange.

Lưu ý dấu (–) trong hàm Lagrange, bởi vì ta cực tiểu theo biến w và b, và là cực đại theo biến a.

Lấy đạo hàm L(w,b,a) theo w và b ta có:

(8)

(9)

Loại bỏ w và b ra khỏi L(w,b,a) bằng cách thế (8), (9) vào. Điều này sẽ dẫn ta đến vấn đề tối ưu:

(10)

Thỏa các ràng buộc:

(11)

(12)

Ở đây hàm nhân (kernel function) được định nghĩa là .

Vấn đề tạm thời gác lại ở đây, ta sẽ thảo luận kỹ thuật giải quyết (10) thỏa (11), (12) này sau.

Để phân lớp cho 1 điểm dữ liệu mới dùng mô hình đã huấn luyện, ta tính dấu của y(x) theo công thức (1), nhưng thế w trong (8) vào:

(13)

Thỏa các điều kiện KKT sau:

(14)

(15)

(16)

Vì thế với mọi điểm dữ liệu, hoặc là hoặc là . Những điểm dữ liệu mà có sẽ không xuất hiện trong (13) và do đó mà không đóng góp trong việc dự đoán điểm dữ liệu mới.

Những điểm dữ liệu còn lại được gọi là support vector, chúng thỏa , đó là những điểm nằm trên lề của siêu phẳng trong không gian đặc trưng.

Support vector chính là cái mà ta quan tâm trong quá trình huấn luyện của SVM. Việc phân lớp cho một điểm dữ liệu mới sẽ chỉ phụ thuộc vào các support vector.

Giả sử rằng ta đã giải quyết được vấn đề (10) và tìm được giá trị nhân tử a, bây giờ ta cần xác định tham số b dựa vào các support vector xn có . Thế (13) vào:

(17)

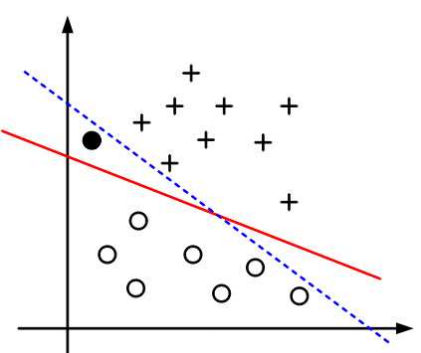
Trong đó S là tập các support vector. Mặc dù ta chỉ cần thế một điểm support vector xn vào là có thể tìm ra b, nhưng để đảm bảo tính ổn định của b ta sẽ tính b theo cách lấy giá trị trung bình dựa trên các support vector.

Đầu tiên ta nhân tn vào (17) (lưu ý , và giá trị b sẽ là:

(18)

Trong đó Ns là tổng số support vector.

Ban đầu để dễ trình bày thuật toán ta đã giả sử là các điểm dữ liệu có thể phân tách hoàn toàn trong không gian đặc trưng . Nhưng việc phân tách hoàn toàn này có thể dẫn đến khả năng tổng quát hóa kém, vì thực tế một số mẫu trong quá trình thu thập dữ liệu có thể bị gán nhãn sai, nếu ta cố tình phân tách hoàn toàn sẽ làm cho mô hình dự đoán quá khớp.



Để chống lại sự quá khớp, chúng ta chấp nhận cho một vài điểm bị phân lớp sai.

Để làm điều này, ta dùng các biến slack variables cho mọi điểm dữ liệu.

* cho những điểm nằm trên lề hoặc phía trong của lề
* cho những điểm còn lại.
* Do đó những điểm nằm trên đường phân cách sẽ có
* Còn những điểm phân lớp sai sẽ có



Công thức (5) sẽ viết lại như sau:

(20)

Mục tiêu của ta bây giờ là cực đại khoảng cách lề, nhưng đồng thời cũng đảm bảo tính mềm mỏng cho những điểm bị phân lớp sai. Ta viết lại vấn đề cần cực tiểu:

(21)

Trong đó C > 0 đóng vai trò quyết định đặt tầm quan trọng vào biến hay là lề.

Bây giờ chúng ta cần cực tiểu (21) thỏa ràng buộc (20) và . Theo Lagrange ta viết lại:

(22)

Trong đó và là các nhân tử Lagrange.

Các điều kiện KKT cần thỏa là:

(23)

(24)

(25)

(26)

(27)

(28)

Với n = 1,…,N

Lấy đạo hàm (22) theo w, b và {}:

(29)

(30)

(31)

Thế (29), (30), (31) vào (22) ta được:

(32)

Từ (23), (26) và (31) ta có:

Vấn đề cần tối ưu giống hệt với trường hợp phân tách hoàn toàn, chỉ có điều kiện ràng buộc khác biệt như sau:

(33)

(34)

Thế (29) vào (1), ta sẽ thấy để dự đoán cho một điểm dữ liệu mới tương tự như (13).

Như trước đó, tập các điểm có không có đóng góp gì cho việc dự đoán điểm dữ liệu mới.

Những điểm còn lại tạo thành các support vector. Những điểm có và theo (25) thỏa:

(35)

Nếu theo (31) có , từ (28) suy ra và đó là những điểm nằm trên lề.

Những điểm có có thể là những điểm phân lớp đúng nằm giữa lề và đường phân cách nếu hoặc có thể là phân lớp sai nếu

Để xác định tham số b trong (1) ta sẽ dùng những support vector mà có vì thế :

(36)

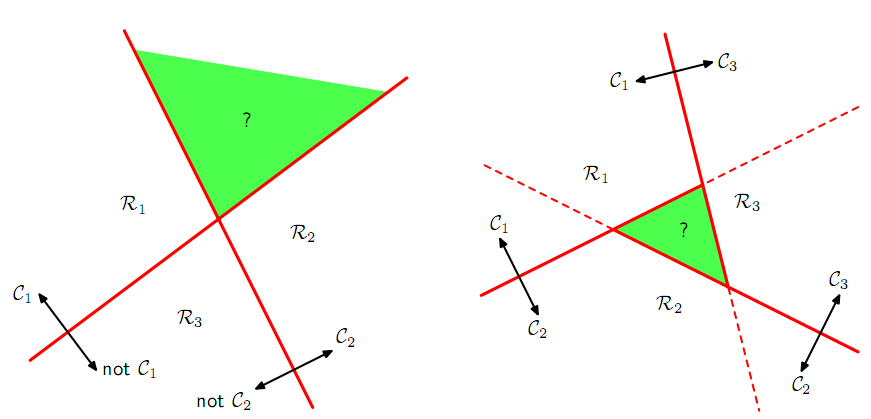
Lần nữa, để đảm bảo tính ổn định của b ta tính theo trung bình:

(37)

Trong đó M là tập các điểm có

Để giải quyết (10) và (32) ta dùng thuật toán Sequential Minimal Optimization (SMO) do Platt đưa ra vào 1999.

# MultiClass SVMs:



Bây giờ xét đến trường hợp phân nhiều lớp K > 2. Chúng ta có thể xây dựng việc phân K-class dựa trên việc kết hợp một số đường phân 2 lớp. Tuy nhiên, điều này sẽ dẫn đến một vài khó khăn (theo Duda and Hart, 1973).

Hướng one-versus-the-rest, ta sẽ dùng K-1 bộ phân lớp nhị phân để xây dựng K-class.

Hướng one-versus-one, dùng K(K-1)/2 bộ phân lớp nhị phân để xây dựng K-class.

Cả 2 hướng đều dẫn đến vùng mập mờ trong phân lớp (như hình vẽ).

Ta có thể tránh được vấn đề này bằng cách xây dựng K-Class dựa trên K hàm tuyến tình có dạng:

Và một điểm x được gán vào lớp Ck khi với mọi .

Một hướng tiếp cận khác do Wu (2004) đề xuất phương pháp ước lượng xác suất cho việc phân m lớp.

# Áp dụng cho bài toán phân loại văn bản:

Hướng dẫn cài đặt:

Mô tả vector đặc trưng của văn bản: Là vector có số chiều là số đặc trưng trong toàn tập dữ liệu, các đặc trưng này đôi một khác nhau. Nếu văn bản có chứa đặc trưng đó sẽ có giá trị 1, ngược lại là 0.

Việc cài đặt SVM khá phức tạp ta nên dùng các thư viện cài săn trên mạng như LibSVM, SVMLight.

Thuật toán gồm 2 giai đoạn huấn luyện và phân lớp:

1. Huấn luyện:

Đầu vào:

* Các vector đặc trưng của văn bản trong tập huấn luyện (Ma trận MxN, với M là số vector đặc trưng trong tập huấn luyện, N là số đặc trưng của vector).
* Tập nhãn/lớp cho từng vector đặc trưng của tập huấn luyện.
* Các tham số cho mô hình SVM: C, (tham số của hàm kernel, thường dùng hàm Gauss)

Đầu ra:

* Mô hình SVM (Các Support Vector, nhân tử Lagrange a, tham số b).

1. Phân lớp:

Đầu vào:

* Vector đặc trưng của văn bản cần phân lớp.
* Mô hình SVM

Đầu ra:

* Nhãn/lớp của văn bản cần phân loại.

# Tài liệu tham khảo:

[1] Christopher M. Bishop, *Pattern Recognition and Machine Learning,* Springer (2007)

.