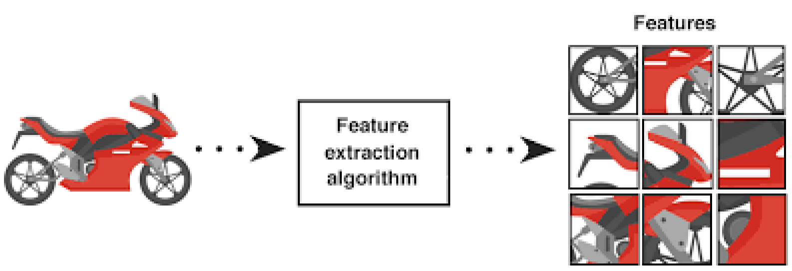
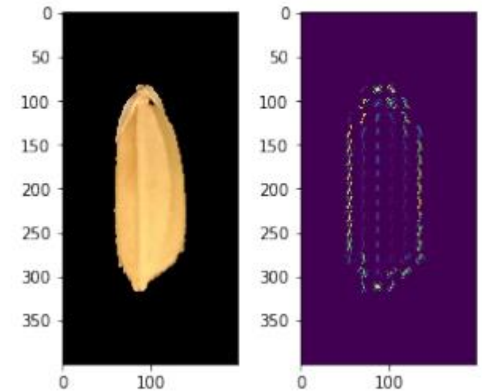
TRƯỜNG ĐẠI HỌC CẦN THƠ
TRƯỜNG CÔNG NGHỆ THÔNG TIN & TRUYỀN THÔNG
KHOA KHOA HỌC MÁY TÍNH

HỌC PHẦN MÁY HỌC ỨNG DỤNG

MSHP: CT294

SUPPORT VECTOR MACHINE (SVM)
MÁY HỌC VÉC-TƠ HỖ TRỢ





T07/2023

Nội dung

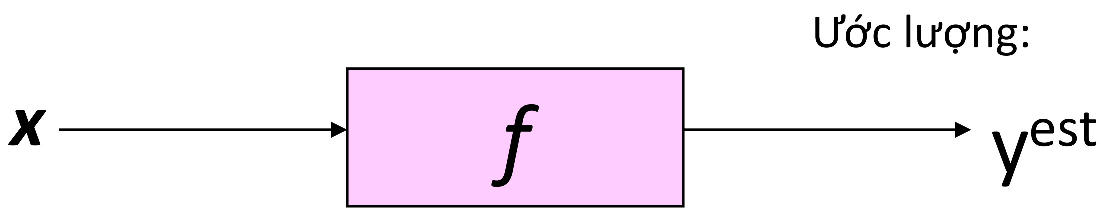
1. Phân lớp nhị phân– Bộ phân lớp tuyến tính
2. Tối ưu hoá margin
3. Nhãn nhiễu– Soft Margin
4. Phân lớp phi tuyến– Kernel
5. Ước lượng xác suất từng lớp
6. Kết luận

Lịch sử của SVM

- SVM có liên quan đến lý thuyết học thống kê [3]
- SVM lần đầu tiên được giới thiệu vào năm 1992 [1]
- SVM trở nên phổ biến nhờ thành công của nó trong nhận dạng chữ số viết tay
- Tỷ lệ lỗi kiểm tra 1,1% đối với SVM. Tỷ lệ này giống như tỷ lệ lỗi của mạng thần kinh được xây dựng, LeNet 4.
- SVM hiện được coi là một ví dụ quan trọng về "**các phương pháp kernel**", một trong những lĩnh vực chính trong học máy

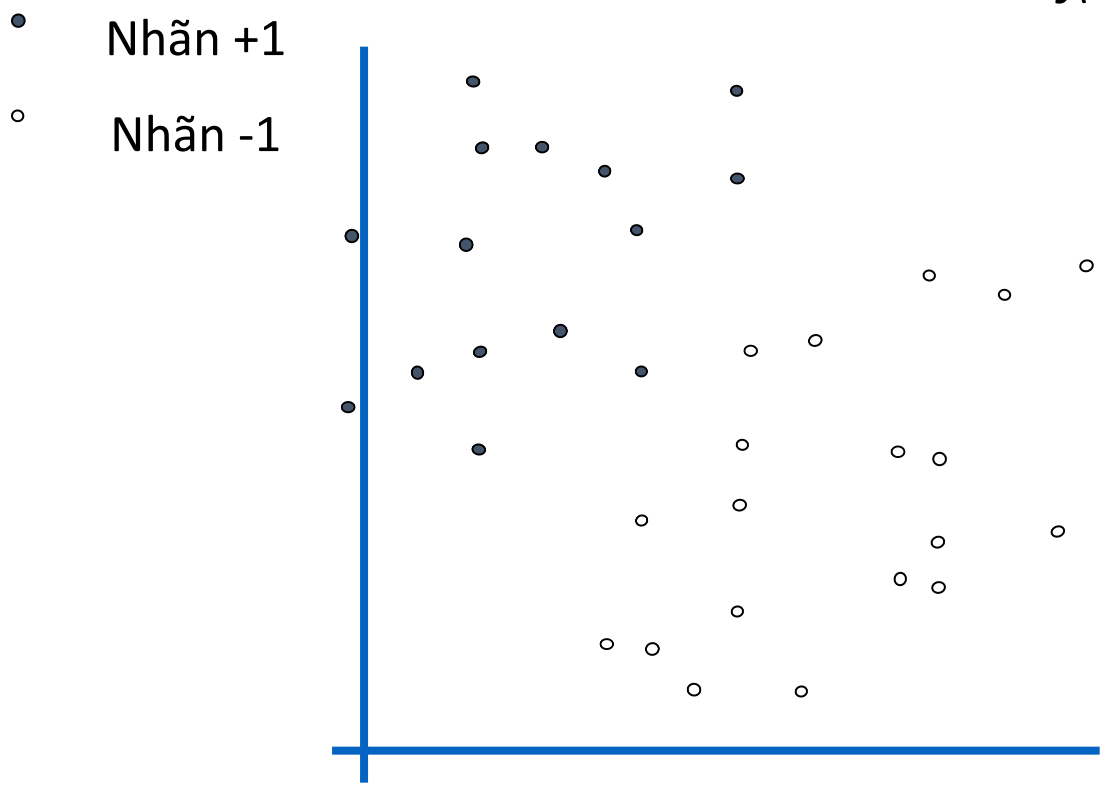
[1] B.E. Boser *et al*. A Training Algorithm for Optimal Margin Classifiers. Proceedings of the Fifth Annual Workshop on Computational Learning Theory 5 144-152, Pittsburgh, 1992.
[2] L. Bottou *et al*. Comparison of classifier methods: a case study in handwritten digit recognition. Proceedings of the 12th IAPR International Conference on Pattern Recognition, vol. 2, pp. 77-82.
[3] V. Vapnik. The Nature of Statistical Learning Theory. 2nd edition, Springer, 1999.

BỘ PHÂN LỚP TUYẾN TÍNH



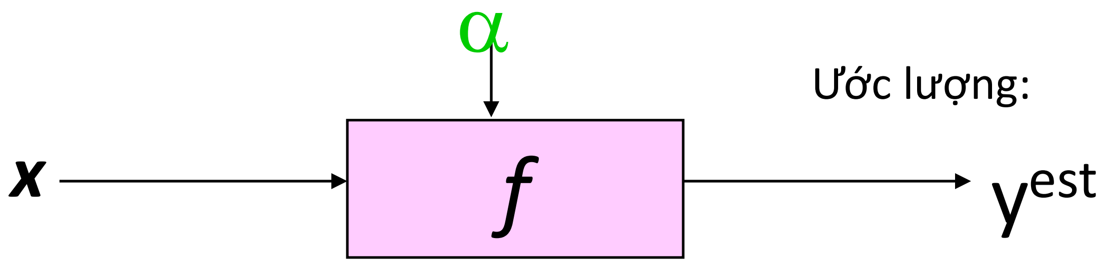
$f(\mathbf{x},\mathbf{w},b) = sign(\mathbf{w}.\ \mathbf{x} - b)$

w: trọng số
x: dữ liệu



PHÂN LỚP DỮ LIỆU NÀY NHƯ THẾ NÀO?

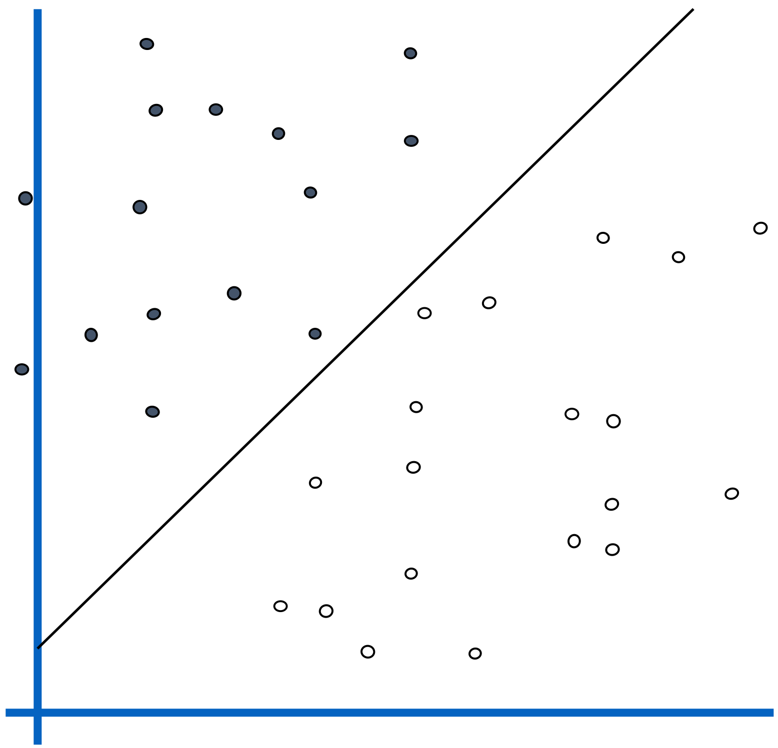
BỘ PHÂN LỚP TUYẾN TÍNH



$f(\boldsymbol{x}, \mathbf{w}, b) = sign(\mathbf{w} \cdot \boldsymbol{x} - b)$

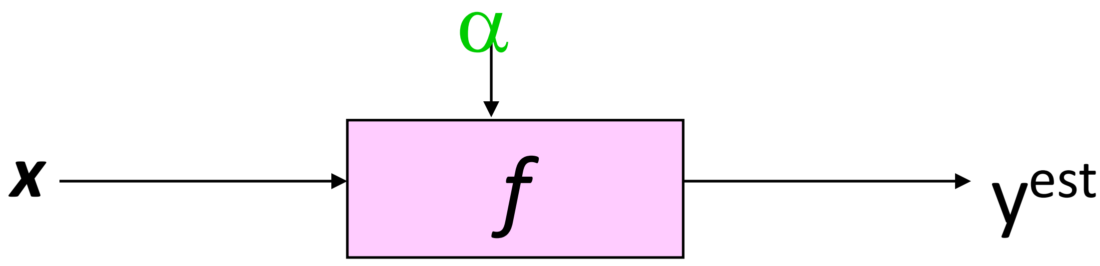
w: trọng số
x: dữ liệu

- ● Nhãn +1
- ○ Nhãn -1



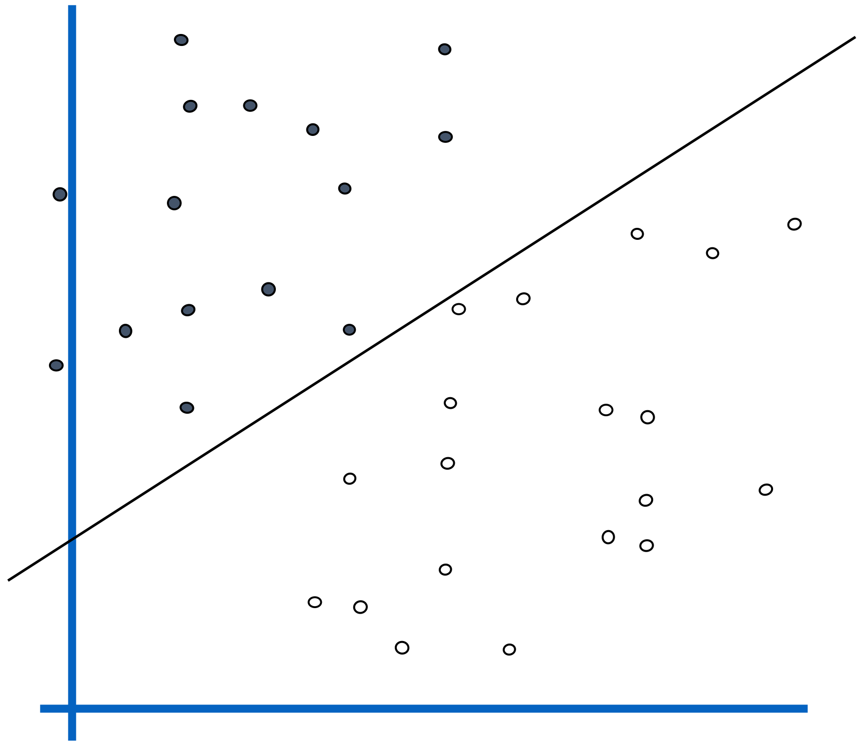
PHÂN LỚP DỮ LIỆU NÀY NHƯ THẾ NÀO?

BỘ PHÂN LỚP TUYẾN TÍNH



$f(\mathbf{x},\mathbf{w},b) = sign(\mathbf{w}.\ \mathbf{x} - b)$

- Nhãn +1
- Nhãn -1

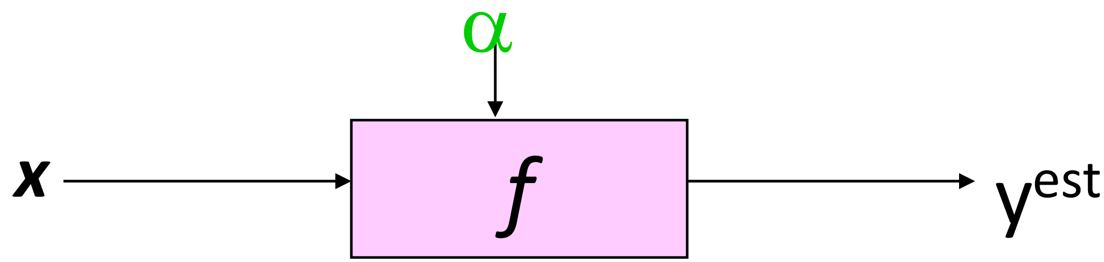


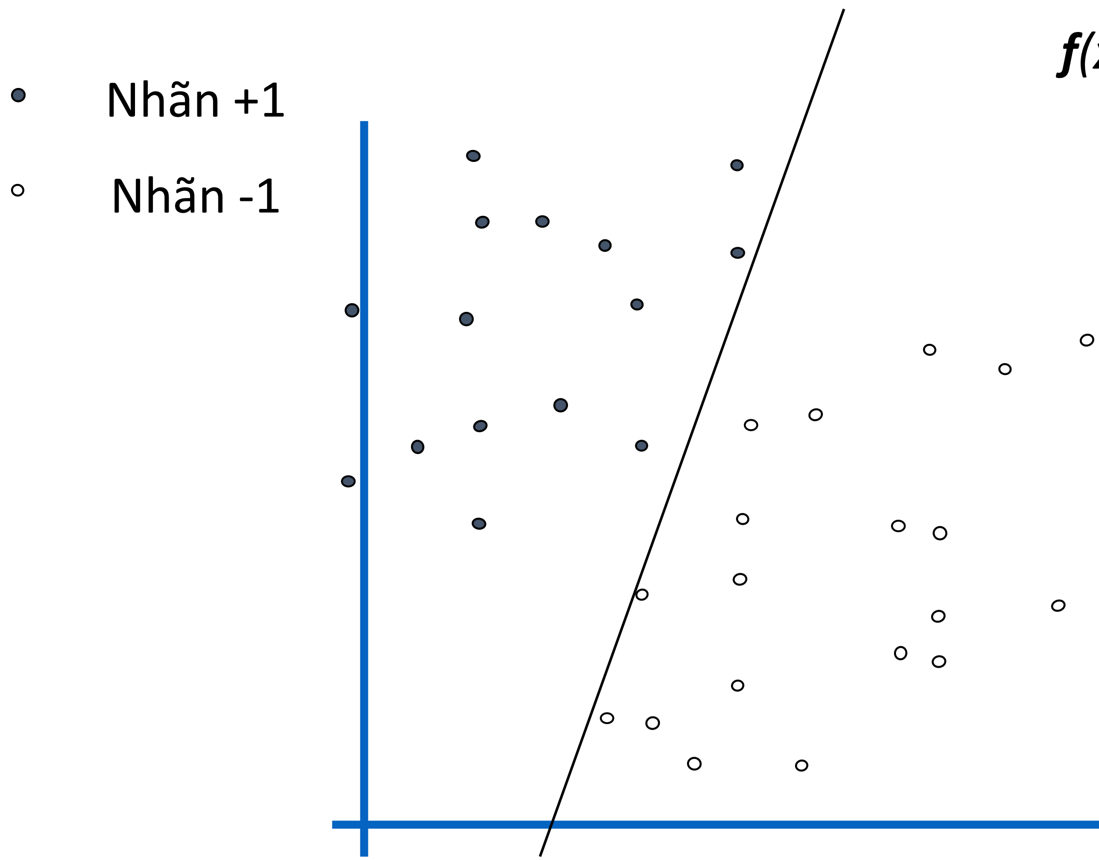
w: trọng số

x: dữ liệu

PHÂN LỚP DỮ LIỆU NÀY NHƯ THẾ NÀO?

BỘ PHÂN LỚP TUYẾN TÍNH



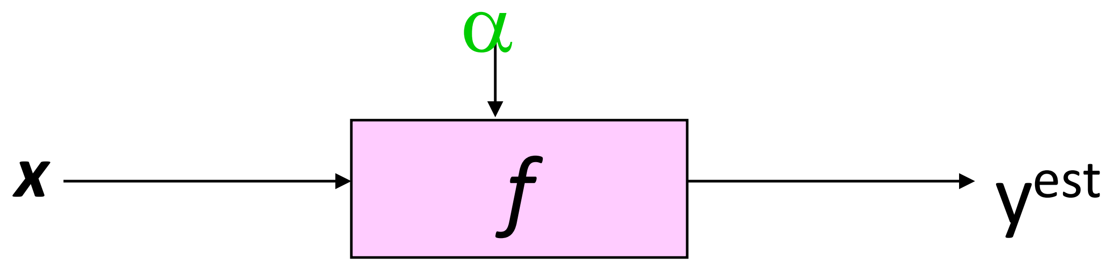


$f(\mathbf{x},\mathbf{w},b) = sign(\mathbf{w}.\ \mathbf{x} - b)$

w: trọng số
x: dữ liệu

PHÂN LỚP DỮ LIỆU NÀY NHƯ THẾ NÀO?

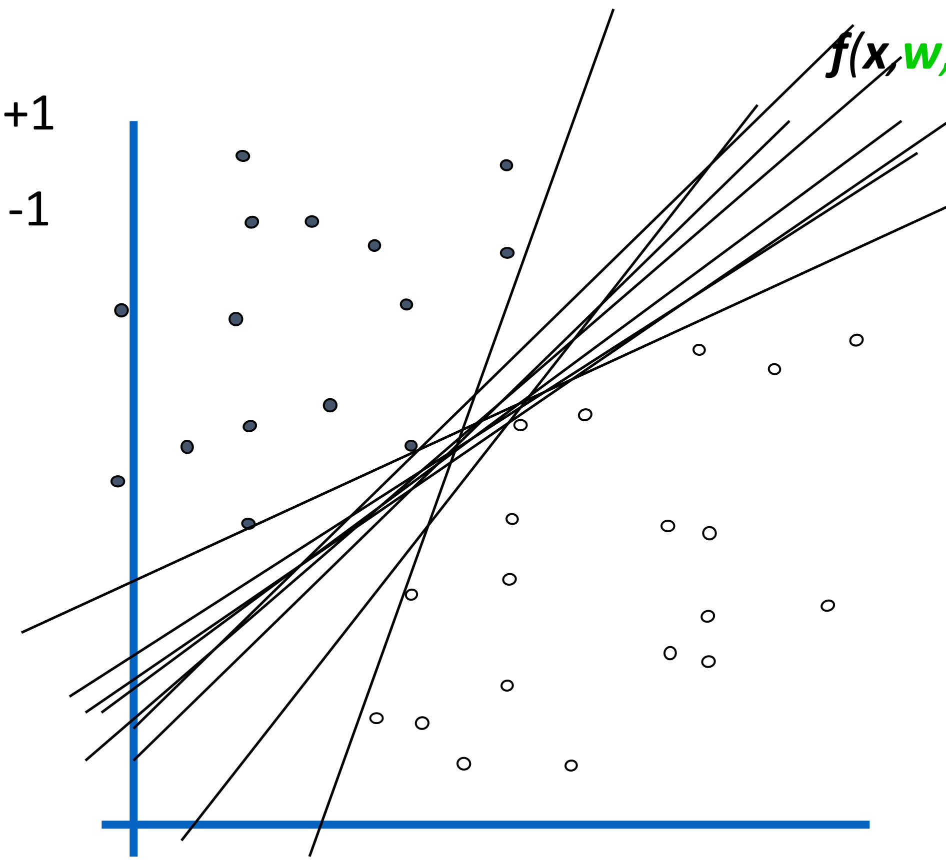
BỘ PHÂN LỚP TUYẾN TÍNH



$f(\mathbf{x},\mathbf{w},b) = sign(\mathbf{w}.\ \mathbf{x} - b)$

● Nhãn +1

○ Nhãn -1



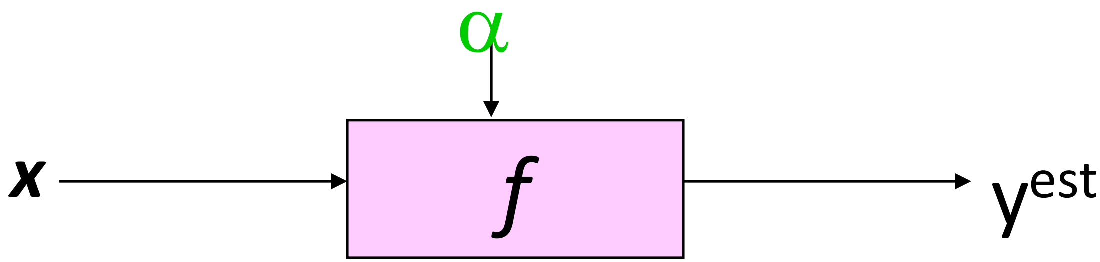
w: trọng số

x: dữ liệu

Những siêu phẳng này là tốt

….Nhưng cái nào là tốt nhất?

BỘ PHÂN LỚP TUYẾN TÍNH

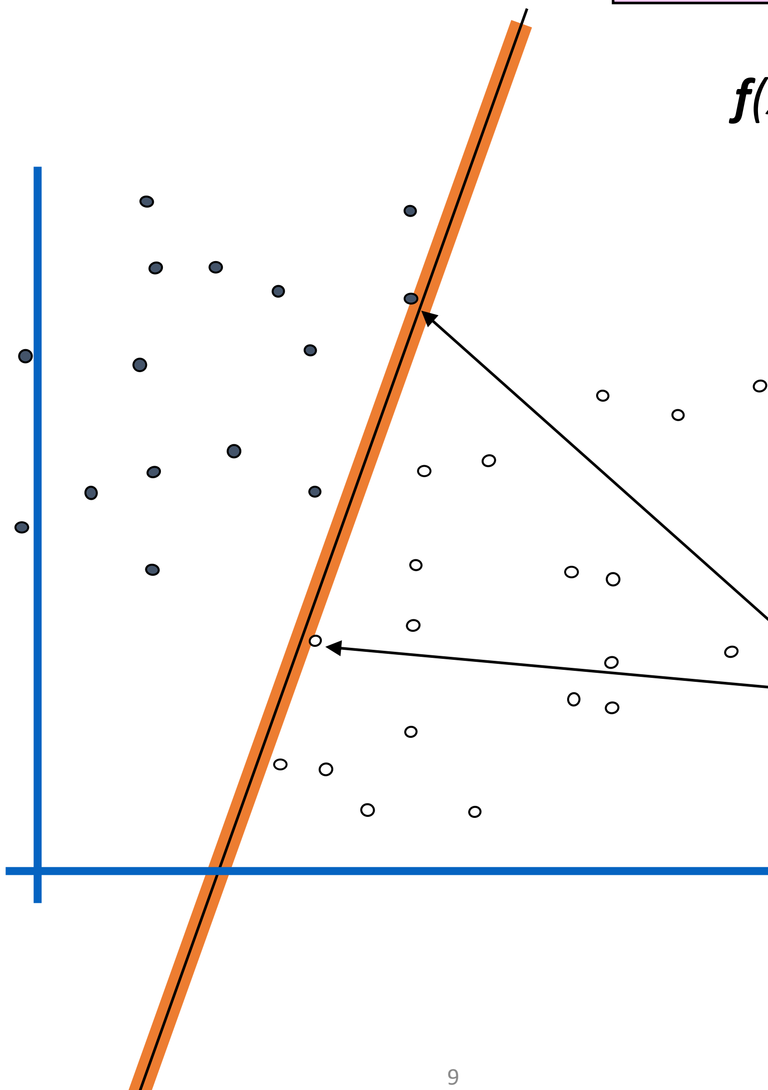


$f(\mathbf{x},\mathbf{w},b) = sign(\mathbf{w}.\ \mathbf{x} - b)$

w: trọng số
x: dữ liệu

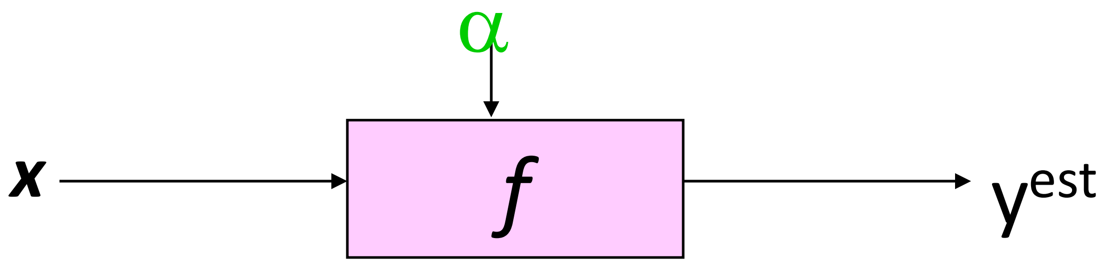
● Nhãn +1

○ Nhãn -1

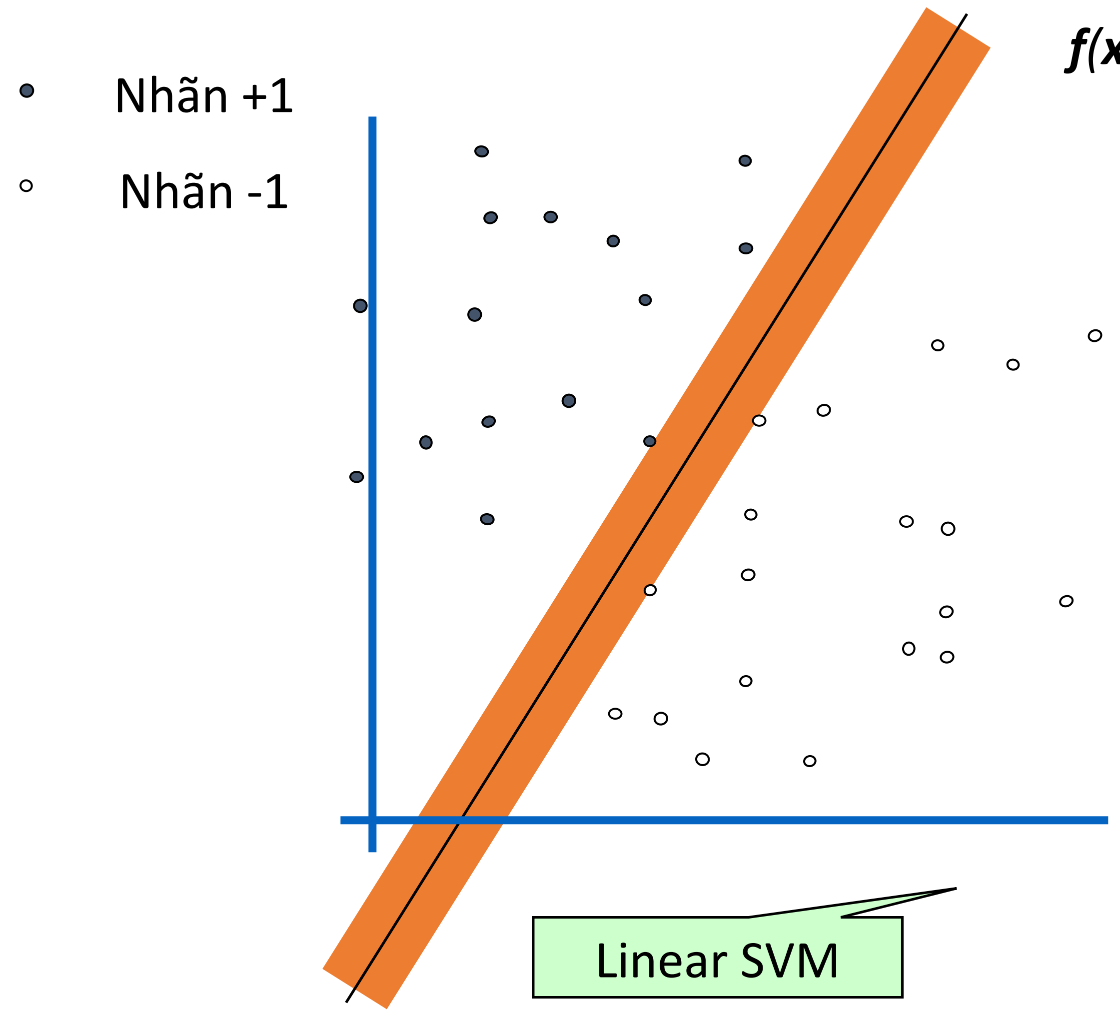


Xác định lề của phân loại tuyến tính là chiều rộng mà ranh giới có thể được tăng lên **trước khi chạm vào một điểm dữ liệu.**

Maximum Margin



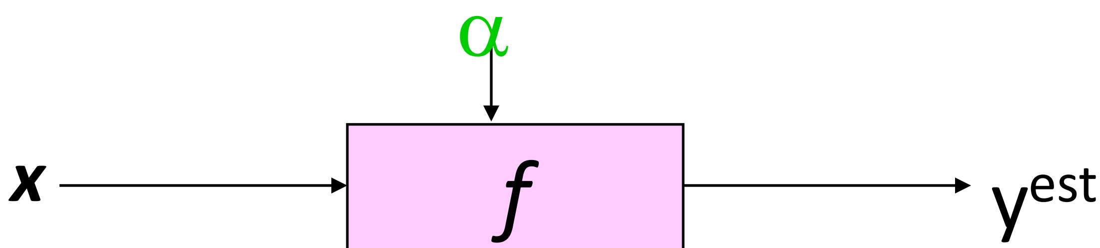
$f(\mathbf{x},\mathbf{w},b) = sign(\mathbf{w}.\ \mathbf{x} - b)$



Bộ phân loại tuyến tính lề tối đa là bộ phân loại tuyến tính có lề tối đa.

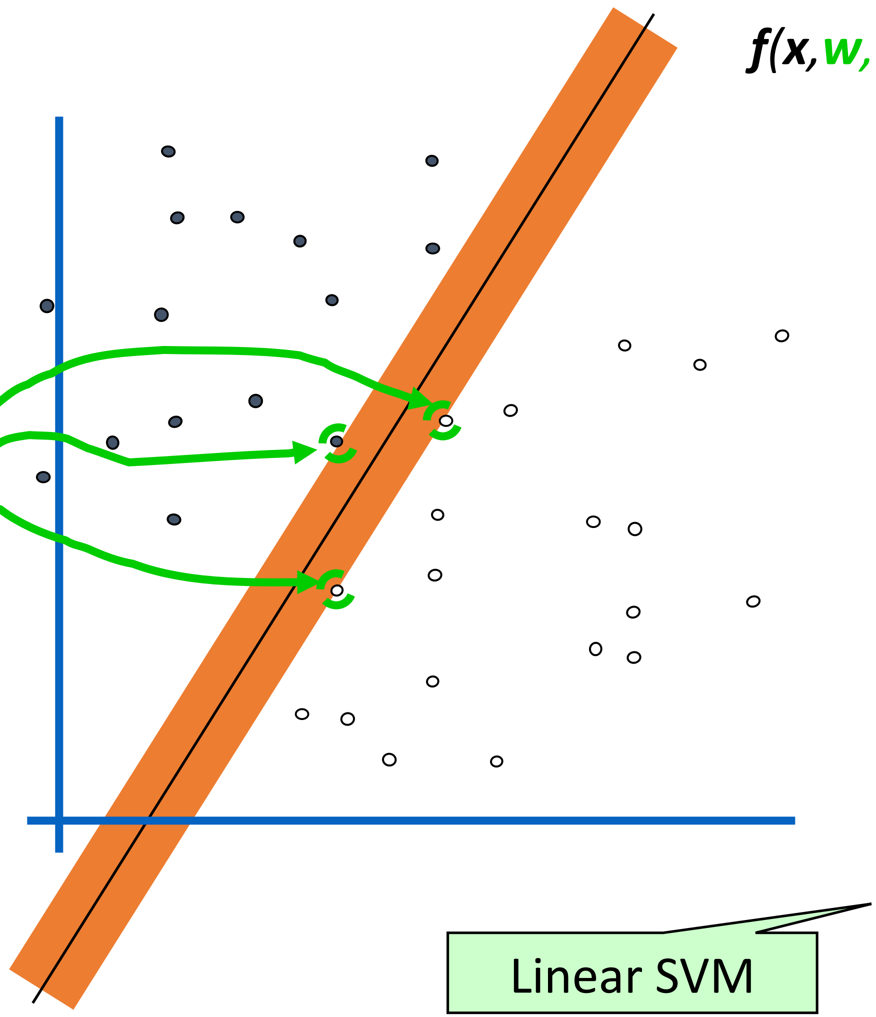
Đây là loại SVM đơn giản nhất (Được gọi là LSVM)

Maximum Margin



$f(x, w, b) = sign(w \cdot x + b)$

- ● Nhãn +1
- ○ Nhãn -1

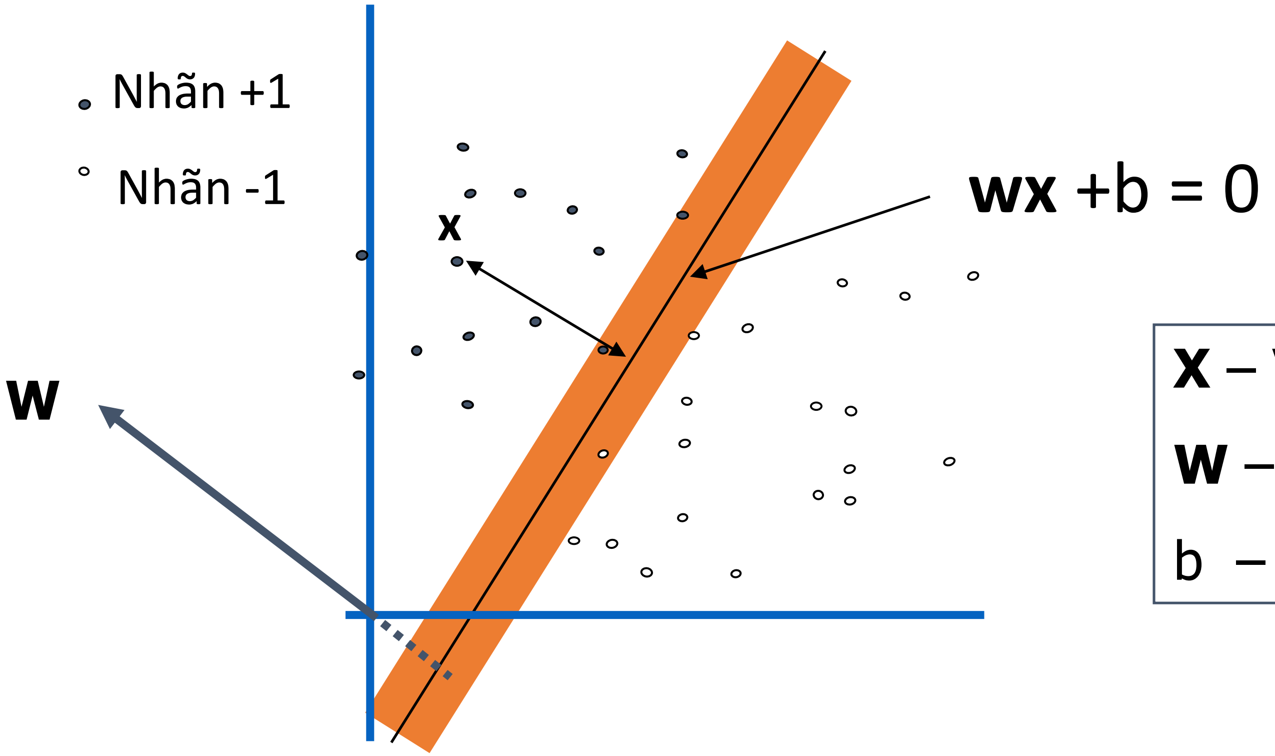


Support Vectors là những điểm dữ liệu mà có margin đẩy lên

Bộ phân loại tuyến tính lề tối đa là bộ phân loại tuyến tính có lề tối đa.

Đây là loại SVM đơn giản nhất (Được gọi là LSVM)

Tính khoảng cách từ một điểm đến một đường như thế nào?



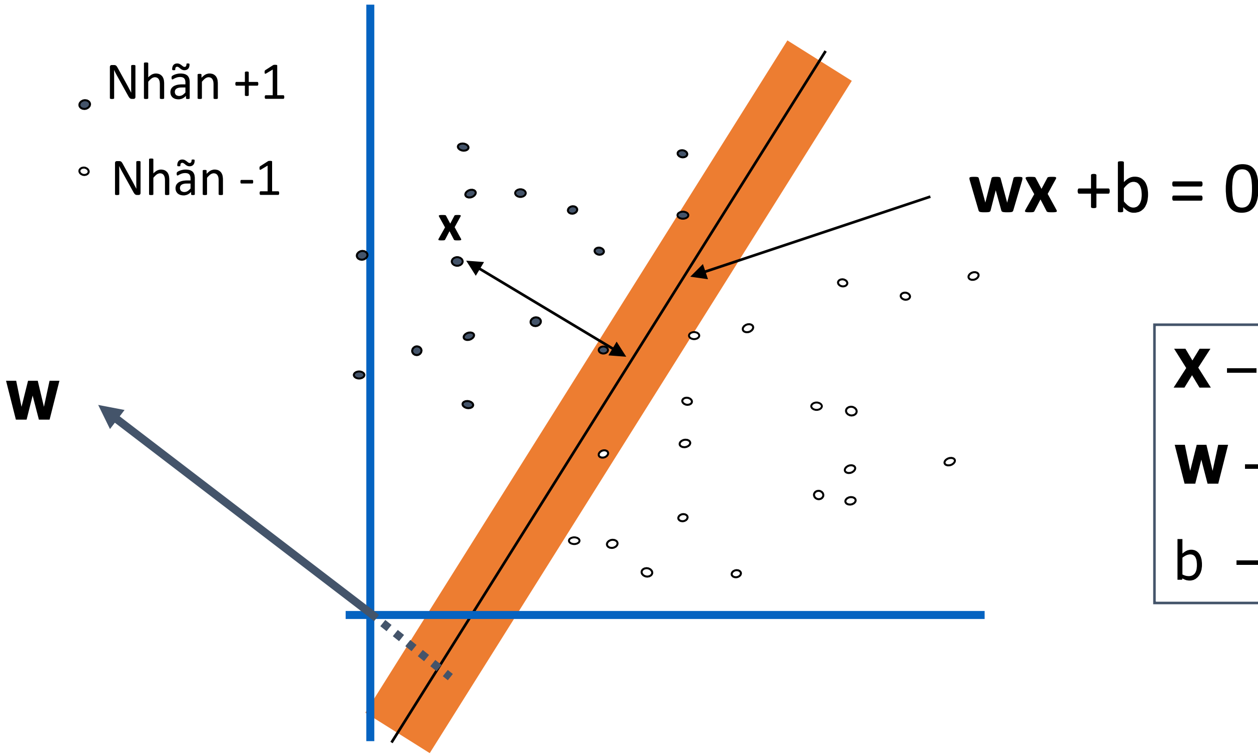
X – Vector

W – Normal Vector

b – Scale Value

- Trong trường hợp này, $w_1*x_1+w_2*x_2+b=0$,
- → $\mathbf{w}=(w_1,w_2)$, $\mathbf{x}=(x_1,x_2)$

Estimate the Margin



X – Vector

W – Normal Vector

b – Scale Value

- Khoảng cách từ một điểm **x** đến một đường thẳng **wx**+b= 0?

$$d(\mathbf{x}) = \frac{\left|\mathbf{x}\cdot\mathbf{w}+b\right|}{\sqrt{\left\|\mathbf{w}\right\|_2^2}} = \frac{\left|\mathbf{x}\cdot\mathbf{w}+b\right|}{\sqrt{\sum_{i=1}^{d} w_i^2}}$$

Nội dung

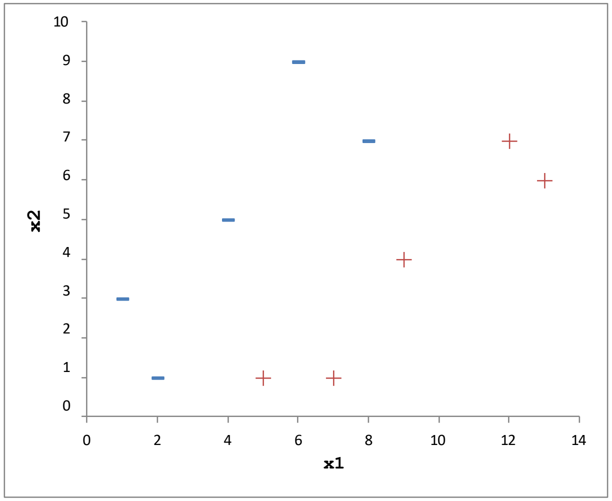
SVM TUYẾN TÍNH

Phân lớp Nhị phân (Binary Classification)

PHÂN LỚP NHỊ PHÂN
Dữ liệu rời rạc tuyến tính

Học giám sát: $Y = f(x_1, x_2, \ldots, x_p ; \beta)$ với vấn đề phân lớp nhị phân, chẳng hạn: $Y \in \{+, -\}$ or $Y \in \{+1, -1\}$

x1	x2	y
1	3	-1
2	1	-1
4	5	-1
6	9	-1
8	7	-1
5	1	1
7	1	1
9	4	1
12	7	1
13	6	1



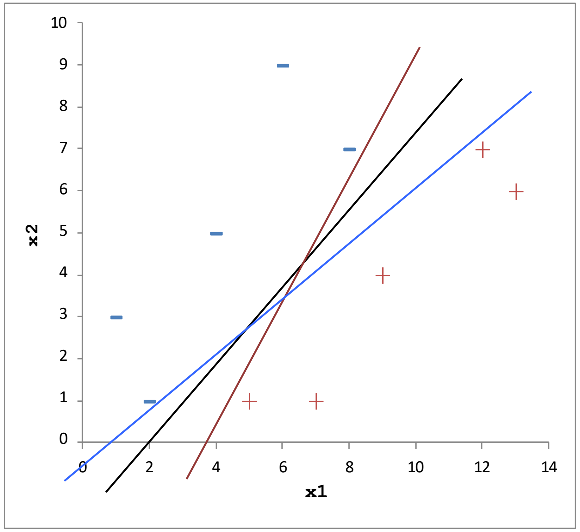
Mục tiêu là tìm "a hyperplane" để phân tách hai lớp “+” and “-”.

$\beta=(\beta_1, \beta_2, \ldots, \beta_p)$ và β_0 là (p+1) parameters (coefficients) để ước lượng

$$f(x) = x^T \beta + \beta_0$$
$$= x_1\beta_1 + x_2\beta_2 + \ldots + \beta_0$$

TÌM GIẢI PHÁP TỐI ƯU

Sau khi "hình dạng - shape" của ranh giới quyết định được xác định, chúng ta phải chọn một giải pháp trong số vô số giải pháp khả thi.



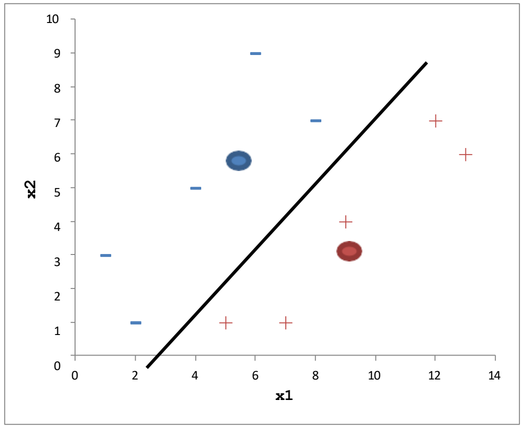
Hai vấn đề quan trọng của framework cho supervised learning :

(1) Chọn "Representation bias" or "hypothesis bias" → we define the shape of the separator

(2) Choosing the search bias i.e. the way to select the best solution among all the possible solutions → it often boils down to set the objective function to optimize

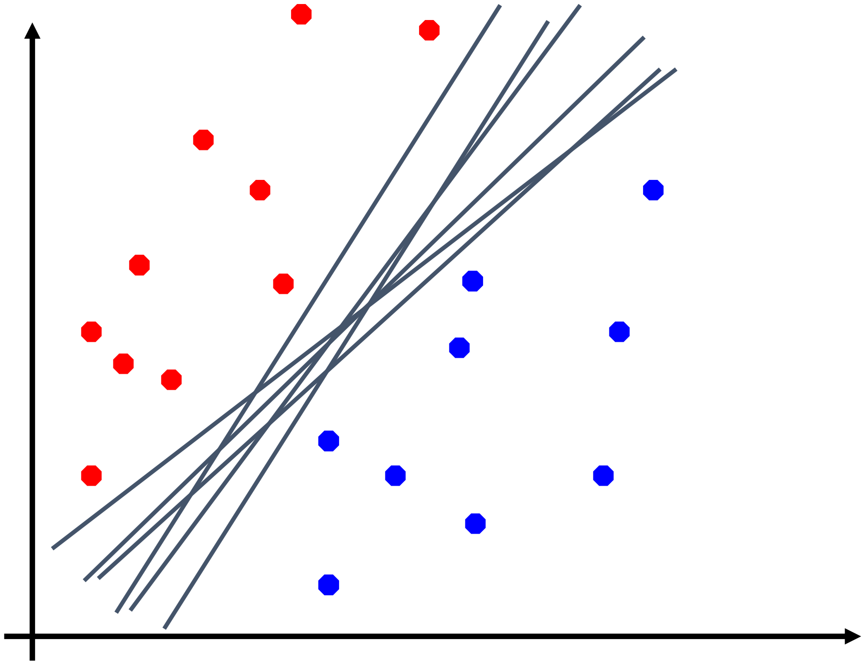
Ví dụ: Linear Discriminant Analysis (LDA)

Siêu phẳng phân tách nằm ở giữa hai trọng tâm có điều kiện theo nghĩa của khoảng cách Mahalanobis.



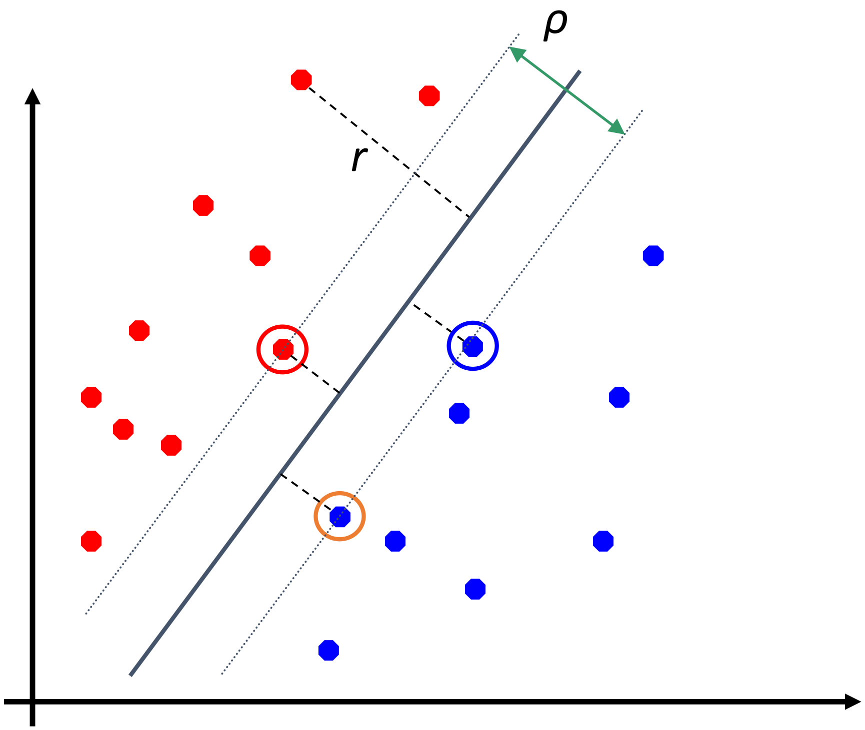
LINEAR SEPARATORS

- Đường nào trong số các dải phân cách tuyến tính là tối ưu?



CLASSIFICATION MARGIN

- Khoảng cách từ ví dụ $\mathbf{x}_i$ đến dấu phân cách là $r = \frac{\mathbf{w}^T \mathbf{x}_i + b}{\|\mathbf{w}\|}$
- Các ví dụ gần với siêu phẳng nhất là các vectơ hỗ trợ.
- Biên ρ của dấu phân cách là khoảng cách giữa các vectơ hỗ trợ.



Nội dung

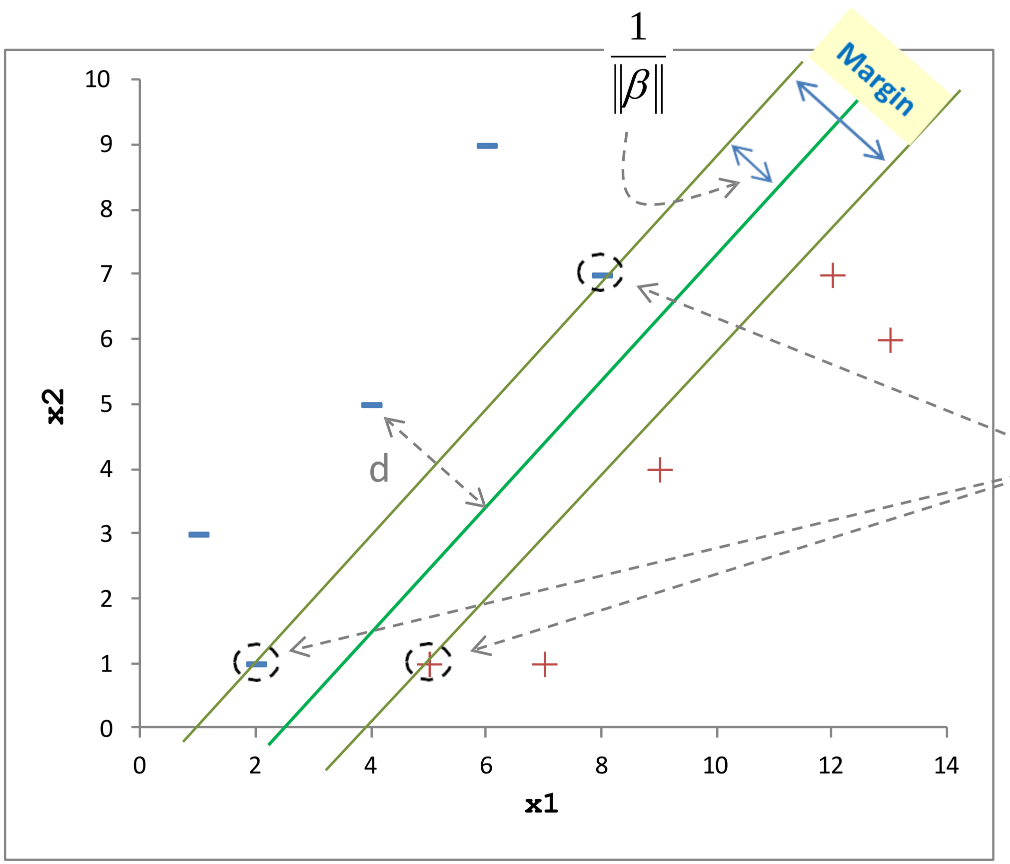
TỐI ƯU HOÁ MARGIN (I)

VẤN ĐỀ CƠ BẢN

NGUYÊN TẮC HARD-MARGIN

LAYOUT

Siêu phẳng phân tách tối ưu phân tách hai lớp và tối đa hóa khoảng cách đến điểm gần nhất từ một trong hai lớp (Vapnik, 1996)



- Khoảng cách từ bất kỳ điểm x nào với ranh giới.

$$d = \frac{|x^T\beta + \beta_0|}{\|\beta\|}$$

- Margin tối đa là:

$$\delta = \frac{2}{\|\beta\|}$$

- Các trường hợp dựa vào lợi nhuận là "vector hỗ trợ". Nếu chúng ta loại bỏ chúng khỏi mẫu, giải pháp tối ưu sẽ được sửa đổi.

- Một số khu vực được xác định trong không gian đại diện

f(x) = 0, → he maximum margin hyperplane

f(x) > 0 → khu vực của các trường hợp « + »

f(x) < 0 →khu vực của các trường hợp « - »

f(x) = +1 or -1, → những siêu phẳng này là lề (MARGIN)

TỐI ƯU MARGIN

Toán học

Cực đại hóa lề tương đương với cực tiểu hóa định mức của vectơ tham số β

$$\max \frac{2}{\|\beta\|} \Leftrightarrow \min \|\beta\|$$

$$\min_{\beta,\beta_0} \|\beta\|$$

Subject to

$$y_i \times f(x_i) \geq 1,\ i = 1, \ldots, n$$

- Norm: $$\|\beta\| = \sqrt{\beta_1^2 + \ldots + \beta_p^2}$$

- Các ràng buộc chỉ ra rằng tất cả các điểm đều ở phía bên phải, ít nhất chúng nằm trên siêu phẳng của các vectơ hỗ trợ.

- Ta có bài toán tối ưu lồi (hàm mục tiêu bậc hai, ràng buộc tuyến tính). Một tối ưu toàn cầu tồn tại.

- Nhưng không có giải pháp theo nghĩa đen. Nó phải đi qua các chương trình tối ưu hóa số.

Lưu ý: cũng có thể viết

$$\min_{\beta,\beta_0} \frac{1}{2}\|\beta\|^2$$

TỐI ƯU MARGIN

Ví dụ với **EXCEL (!)**

Chúng tôi sử dụng SOLVER để giải quyết vấn đề tối ưu hóa.

	beta.1	beta.2	beta.0
	0.667	-0.667	-1.667

(p + 1) biến

n°	x1	x2	y	f(x)	f(x)*y
1	1	3	-1	-3	3
2	2	1	-1	-1	1
3	4	5	-1	-2.33333333	2.33333333
4	6	9	-1	-3.66666667	3.66666667
5	8	7	-1	-1	1
6	5	1	1	1	1
7	7	1	1	2.33333333	2.33333333
8	9	4	1	1.66666667	1.66666667
9	12	7	1	1.66666667	1.66666667
10	13	6	1	3	3

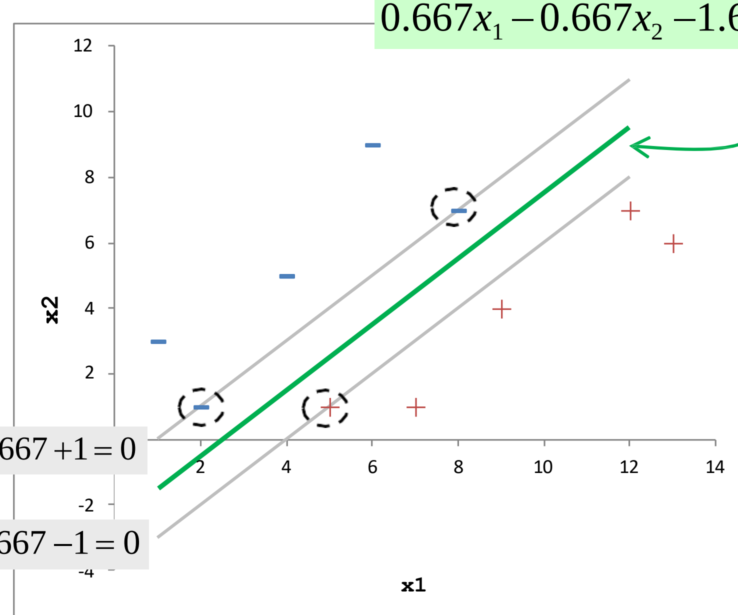
Ràng buộc bão hòa: Đã tìm thấy 3 vectơ hỗ trợ (n°2, 5 et 6)

Norme.Beta	0.943

n = 10 ràng buộc

Mục tiêu : $\|\beta\|$

$$0.667x_1 - 0.667x_2 - 1.667 = 0$$



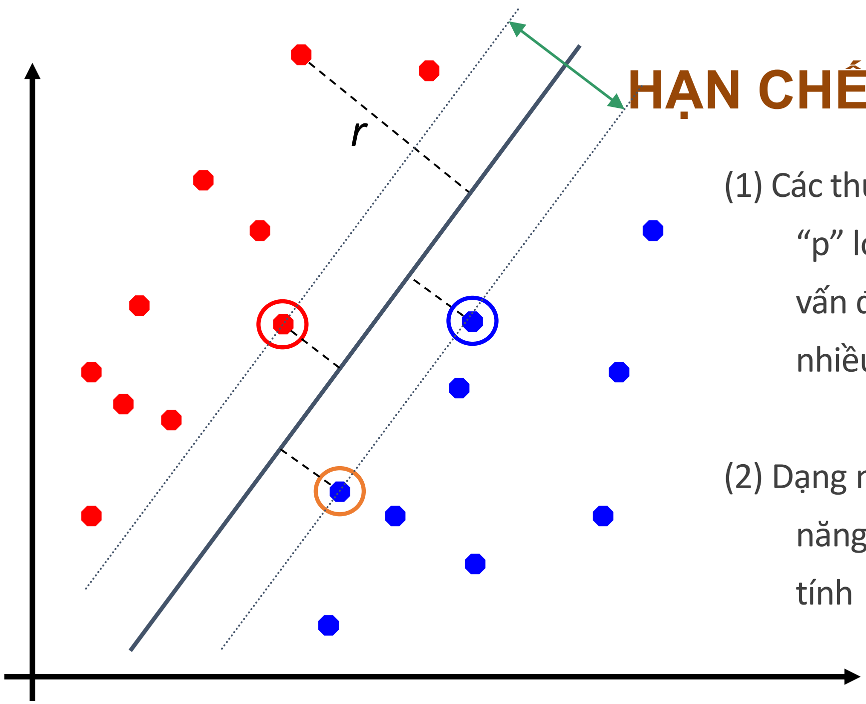
$$x^T\beta + \beta_0 \pm 1 = 0 \quad \begin{cases} 0.667x_1 - 0.667x_2 - 1.667 + 1 = 0 \\ 0.667x_1 - 0.667x_2 - 1.667 - 1 = 0 \end{cases}$$

VẤN ĐỀ CƠ BẢN
COMMENTS

Gán quy tắc cho trường hợp i* dựa trên các hệ số ước lượng $\hat{\beta}_j$

$$f(x_{i*}) = \begin{cases} \geq 0 \Rightarrow \hat{y}_{i*} = 1 \\ < 0 \Rightarrow \hat{y}_{i*} = -1 \end{cases}$$

beta.1	beta.2	beta.0		
0.667	-0.667	-1.667		
x1	**x2**	**y**	**f(x)**	**prediction**
1	3	-1	-3	-1
2	1	-1	-1	-1
4	5	-1	-2.3333	-1
6	9	-1	-3.6667	-1
8	7	-1	-1	-1
5	1	1	1	1
7	1	1	2.33333	1
9	4	1	1.66667	1
12	7	1	1.66667	1
13	6	1	3	1



HẠN CHẾ:

(1) Các thuật toán tối ưu hóa số (prog. bậc hai) không hoạt động khi "p" lớn (> vài trăm). Điều này thường xảy ra khi chúng ta xử lý các vấn đề thực tế (ví dụ: khai thác văn bản, hình ảnh,...) (ít ví dụ, nhiều mô tả)

(2) Dạng nguyên thủy này không làm nổi bật khả năng sử dụng các chức năng "hạt nhân" cho phép vượt ra ngoài các bộ phân loại tuyến tính

Nội dung

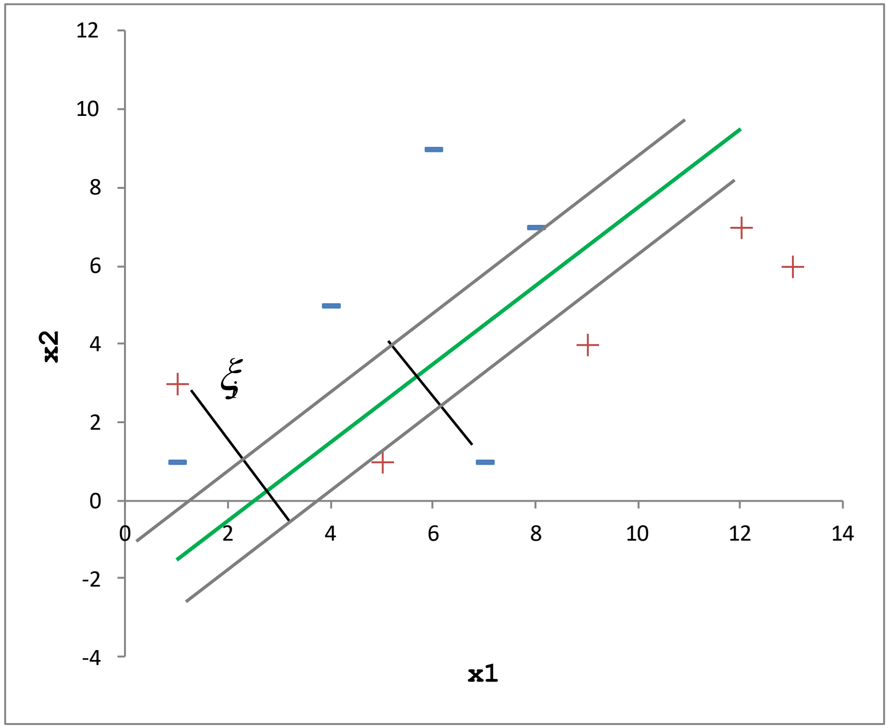
SOFT MARGIN

NHÃN DỮ LIỆU **NHIỄU**

BIẾN “SLACK"

SỬ DỤNG ξ_i để xử lý các trường hợp phân loại sai

Trong các vấn đề thực tế, một sự phân loại hoàn hảo là không khả thi.
Một số trường hợp ở phía sai của lề.



- ξ là một vectơ có kích thước n
- $\xi_i \geq 0$ đánh dấu các trường hợp phân loại sai
- $\xi_i = 0$, trường hợp nằm ở bên phải của lề
- $\xi_i < 1$, trường hợp nằm ở phía bên phải của
- siêu phẳng lề tối đa, nhưng nó vượt quá lề của nó
- $\xi_i > 1$, trường hợp bị phân loại sai, tức là nó nằm sai phía của siêu phẳng lề tối đa

Tái cấu trúc bài toán

Introduction of the cost parameter "C"

Chúng ta nên phạt lỗi, nhiều hay ít tùy thuộc vào việc bạn muốn một mô hình phù hợp hơn hay ít hơn với dữ liệu huấn luyện.

Form Cơ bản

$$\min_{\beta,\beta_0,\xi_i} \frac{1}{2}\|\beta\|^2 + C\sum_{i=1}^{n}\xi_i$$

$s.c.$

$$y_i \times (x_i^T\beta + \beta_0) \geq 1-\xi_i, \forall i = 1,\ldots,n$$

$$\xi_i \geq 0, \forall i$$

Hình thức kép

$$\max_{\alpha} L_D(\alpha) = \sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{i'=1}^{n}\alpha_i\alpha_{i'}y_i y_{i'}\langle x_i, x_{i'}\rangle$$

$s.c.$

$$\sum_{i=1}^{n}\alpha_i y_i = 0$$

$$0 \leq \alpha_i \leq C, \forall i$$

Khả năng chịu lỗi ít nhiều được nhấn mạnh với tham số C (tham số ''chi phí'')

➔ C quá cao: overfitting

➔ C quá thấp: underfitting

Soft-margin – Ví dụ

Dạng cơ bản:

Tối thiểu hóa hàm mục tiêu cho β và ξ

C là tham số, ta đặt C = 5

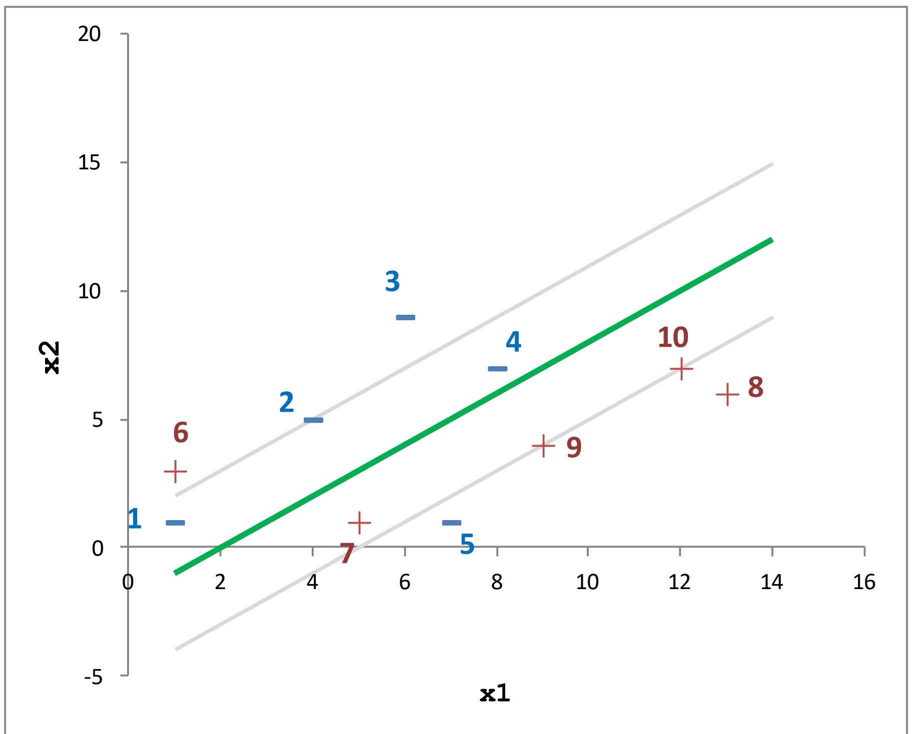
Giá trị của C là một vấn đề quan trọng trong thực tế

	beta.1	beta.2	beta.0			
	0.333	-0.333	-0.667			

n°	x1	x2	y	ksi	1-ksi	y*f(x)
1	1	1	-1	0.333	0.667	0.667
2	4	5	-1	0	1	1
3	6	9	-1	0	1	1.667
4	8	7	-1	0.667	0.333	0.333
5	7	1	-1	2.333	-1.333	-1.333
6	1	3	1	2.333	-1.333	-1.333
7	5	1	1	0.333	0.667	0.667
8	13	6	1	0	1	1.667
9	9	4	1	0	1	1
10	12	7	1	0	1	1

C 5

Fonc.Obj 30.1111



- $y_i(x_i^T\beta+\beta_0)=1 - \xi_i$: ràng buộc ổn đinh

vectơ hỗ trợ (nền màu vàng) tức là nếu chúng tôi loại bỏ trường hợp, giải pháp sẽ khác (8 trường hợp ở đây)

- $\xi_\iota = 0$: trường hợp là phía bên phải của lề
- $\xi_i \geq 1$: trường hợp nằm ở phía sai của siêu phẳng lề tối đa (2 trường hợp được phân loại sai ở đây)
- $0 < \xi_i < 1$: đối tượng nằm ở phía bên phải của siêu phẳng lề tối đa, nhưng nó vượt quá lề của nó

Nội dung

PHÂN LỚP PHI TUYẾN

NHỮNG HÀM NHÂN (KERNEL)

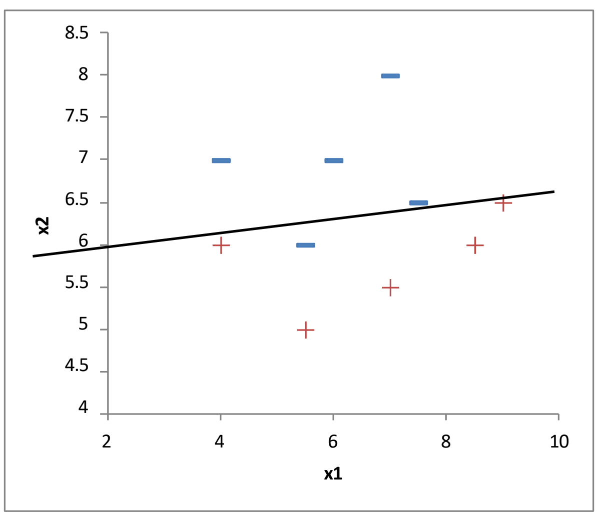
CHUYỂN ĐỔI KHÔNG GIAN FEATURES

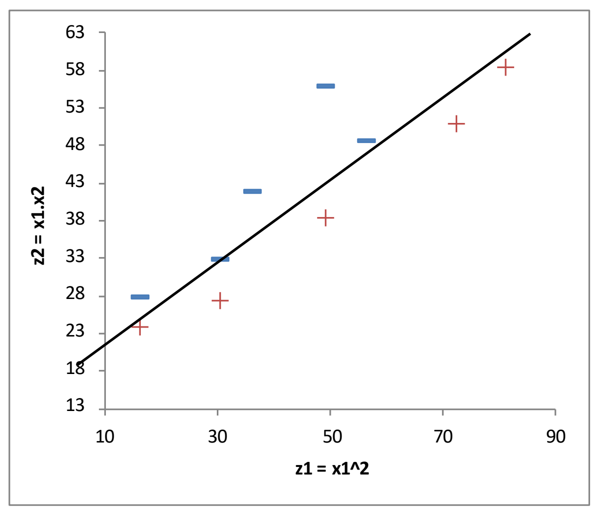
Bằng cách thực hiện các phép biến đổi thích hợp của các biến, chúng ta có thể làm cho vấn đề có thể tách tuyến tính trở thành vấn đề không thể tách tuyến tính trong không gian biểu diễn ban đầu.

n°	x1	x2	y
1	4	7	-1
2	7	8	-1
3	5.5	6	-1
4	6	7	-1
5	7.5	6.5	-1
6	5.5	5	1
7	4	6	1
8	7	5.5	1
9	8.5	6	1
10	9	6.5	1

n°	z1	z2	y
1	16	28	-1
2	49	56	-1
3	30.25	33	-1
4	36	42	-1
5	56.25	48.75	-1
6	30.25	27.5	1
7	16	24	1
8	49	38.5	1
9	72.25	51	1
10	81	58.5	1

$$z^1 = x_1^2$$
$$z_2 = x_1 x_2$$





Nhưng việc nhân các biến trung gian một cách cụ thể trong cơ sở dữ liệu là tốn kém, mà không có bất kỳ sự đảm bảo nào rằng chúng ta sẽ có được một phép biến đổi hiệu quả.

KERNEL ĐA THỨC

ỨNG DỤNG CHO SCALAR

The dot product between vectors has an important place in the calculations (dual form). SVM can take advantage of the "kernel" functions.

Let a transformation function $\phi(x)$ of initial variables

With the dual form, to optimize the Lagrangian, we calculate the scalar product $<\phi(x_i), \phi(x_{i'})>$ for each pair of instances (i, i')

Ex. $x = (x_1, x_2) \rightarrow \phi(x) = (x_1^2, \sqrt{2}x_1 x_2, x_2^2)$

We should handle 3 variables instead of 2, the calculations are more expensive, not to mention the storage of additional variables.

We can find a function K(.), called Kernel Function, such as

$$K(x_i, x_{i'}) = \langle \phi(x_i), \phi(x_{i'}) \rangle$$

The main consequence is that we simply calculate the scalar product $<x_i, x_{i'}>$, and we transform the result with the Kernel function.

We handle only the 2 initial variables for calculations. But the algorithm fits the classifier in a 3 dimensional space!

KERNEL ĐA THỨC

VÍ DỤ

Tích giữa hai trường hợp (vectơ) u và v với các giá trị sau

$$u = (4,7) \quad v = (2,5) \Rightarrow \langle u, v \rangle = 4\times 2 + 7\times 5 = 43$$

Transformation (1)

$$\phi(x) = (x_1^2, \sqrt{2}x_1x_2, x_2^2)$$

$$\left.\begin{matrix}\phi(u) = (16, 39.6, 49) \\ \phi(v) = (4, 14.1, 25)\end{matrix}\right\} \Rightarrow \langle \phi(u), \phi(v) \rangle = 1849$$

chức năng tương ứng (1)

$$K_1(u,v) = (\langle u, v \rangle)^2 = 43^2 = 1849$$

Kết quả là tương đương nhau. Với K (.), chúng ta làm việc trong không gian nhiều chiều hơn mà không cần phải tạo các biến một cách rõ ràng.

Transformation (2)

$$\phi(x) = (1, \sqrt{2}x_1, \sqrt{2}x_2, x_1^2, x_2^2, \sqrt{2}x_1x2)$$

$$\left.\begin{matrix}\phi(u) = (1, 5.7, 9.9, 16, 49, 39.6) \\ \phi(v) = (1, 2.8, 7.1, 4, 25, 14.1)\end{matrix}\right\} \Rightarrow \langle \phi(u), \phi(v) \rangle = 1936$$

Chức năng tương ứng (2)

$$K_2(u,v) = (1 + \langle u, v \rangle)^2 = (1 + 43)^2 = 1936$$

Chúng tôi làm việc trong không gian 5 chiều trong cấu hình này.

MỘT SỐ KERNEL

Các chức năng phổ biến nhất trong các công cụ (ví dụ: gói Scikit-learn cho Python - SVC)

Đặt đúng giá trị của các tham số là vấn đề then chốt, bao gồm cả ''tham số chi phí - Cost parameter'' C.

Polynomial

$$K(u,v) = \left(\text{coef0} + \langle u,v \rangle\right)^{\text{degree}}$$

coef0 = 0 và độ = 1, ta có hạt nhân "tuyến tính"

Gaussian radial basis function (RBF)

$$K(u,v) = \exp\left(-\gamma . \times \|u - v\|^2\right)$$

nếu nó không được chỉ định, các công cụ được đặt theo mặc định (p: số lượng biến)

$$\gamma = \frac{1}{p}$$

Hyperbolic tangent

$$K(u,v) = \tanh\left(\gamma \times \langle u,v \rangle + \text{coef0}\right)$$

Có một chút đa nghĩa trong các tham số, nhưng chúng đã được phổ biến bởi gói **LIBSVM** nổi tiếng, được bao gồm trong một số công cụ khai thác dữ liệu (Scikit-Learn - Python, e1071 - R, Tanagra, v.v.)

Nội dung

XÁC SUẤT LỚP THÀNH VIÊN

ĐIỂM VÀ XÁC SUẤT

Xác suất lớp thành viên

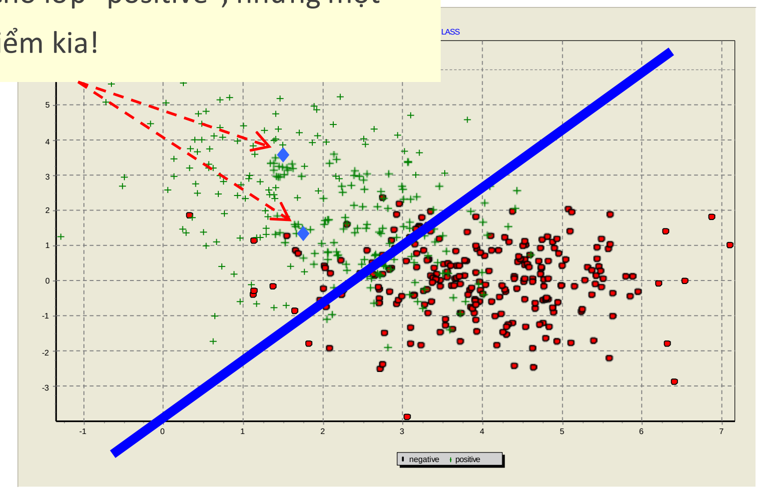
Đầu ra của SVM = điểm số, nhưng chúng không được hiệu chỉnh

Đầu ra của hàm phân loại f(x)

cho phép gán một lớp cho một thể hiện

$$f(x_{i*}) = \begin{cases} \geq 0 \Rightarrow \mathrm{f(x)} = 1 \\ < 0 \Rightarrow \mathrm{f(x)} = -1 \end{cases}$$

Hai điểm được gán cho lớp ”positive", nhưng một điểm tích cực hơn điểm kia!



Chúng tôi cần một dấu hiệu về độ tin cậy của phản hồi.

But...

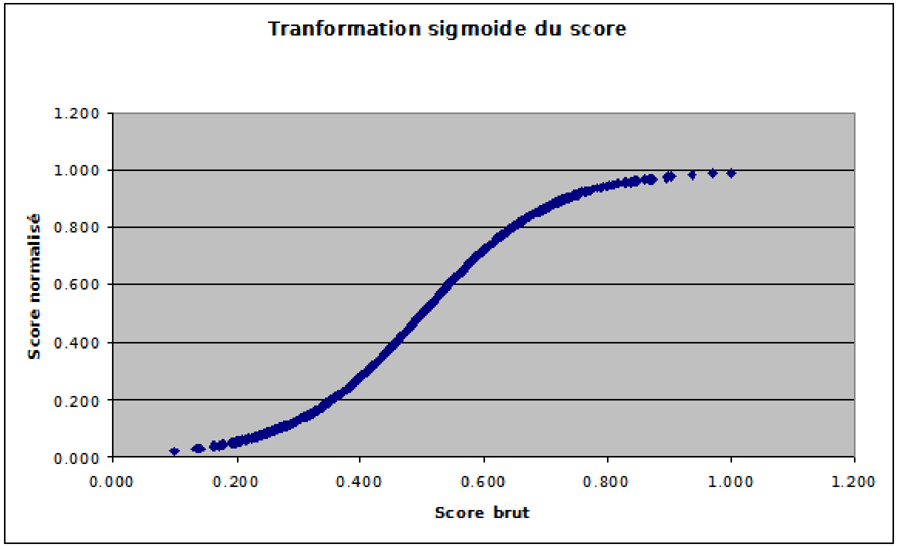
|f(x)| đã là một dấu hiệu tốt. Nó cho phép xếp hạng các cá nhân theo mức độ “sự tích cực” (ví dụ: tính điểm, nhắm mục tiêu, v.v.)

Trong nhiều lĩnh vực, chúng ta cần ước tính xác suất thành viên của lớp (ví dụ: diễn giải, kết hợp với ma trận chi phí, so sánh với đầu ra của các phương pháp khác, v.v.)

Platt scaling
Maximum likelihood estimation

Chúng tôi sử dụng hàm sigmoid để ánh xạ f(x) trong khoảng [0, 1]

$$P(Y=1/x)=\frac{1}{1+\exp[-f(x)]}$$



Chúng ta có thể phát triển một giải pháp phức tạp hơn bằng cách sử dụng một biểu thức được tham số hóa và ước tính các giá trị của các hệ số bằng ước lượng khả năng tối đa

$$P(Y=1/x)=\frac{1}{1+\exp[-(a\times f(x)+b)]}$$

Một chương trình hồi quy logistic có thể ước tính dễ dàng các giá trị của "a" và "b"

Platt scaling
Ví dụ

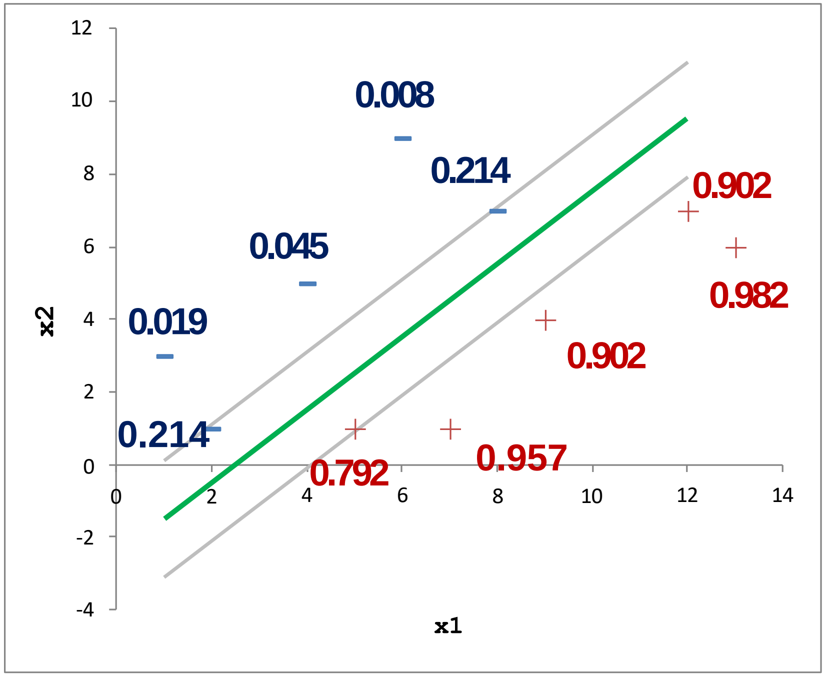
	beta.1	beta.2	beta.0		
	0.667	-0.667	-1.667		
n°	x1	x2	y	f(x)	P(y=1/x)
1	1	3	-1	-3.000	0.019
2	2	1	-1	-1.000	0.214
3	4	5	-1	-2.333	0.045
4	6	9	-1	-3.667	0.008
5	8	7	-1	-1.000	0.214
6	5	1	1	1.000	0.792
7	7	1	1	2.333	0.957
8	9	4	1	1.667	0.902
9	12	7	1	1.667	0.902
10	13	6	1	3.000	0.982

a	1.32
b	0.02

Xác suất thành viên của lớp phù hợp với vị trí của điểm và khoảng cách của nó với đường biên (siêu phẳng lề tối đa).

Chúng ta hãy xem ví dụ

$$P(Y=1/x)=\frac{1}{1+\exp[-(1.32\times f(x)+0.02)]}$$



Nội dung

KẾT LUẬN

Ưu khuyết điểm của SVM

SVM (Ưu khuyết điểm)

ƯU ĐIỂM

- Khả năng xử lý tập dữ liệu nhiều chiều (high #variables)
- Xử lý ngay cả khi tỷ lệ "#quan sát / #biến số" bị đảo ngược
- Xử lý hiệu quả các bài toán phi tuyến với cơ chế kernel
- Phi tham số
- Xử lý chống lại các ngoại lệ (được kiểm soát với tham số C)
- #vector hỗ trợ cung cấp một dấu hiệu tốt về complexity của vấn đề cần xử lý
- Khả năng điều chỉnh các tham số cho phép tính linh hoạt (ví dụ: tuyến tính so với phi tuyến tính, chính quy hóa, v.v.)

NHƯỢC ĐIỂM

- Việc xác định các giá trị tối ưu của các tham số là không rõ ràng (SVM có thể sensitive với các tham số)
- Khó xử lý tập dữ liệu lớn (#instance)
- Các sự cố khi chúng tôi xử lý các nhãn dữ liệu nhiễu (tăng số lượng vectơ hỗ trợ)
- Không có mô hình rõ ràng khi chúng ta sử dụng kernel (nhân) phi tuyến
- Giải thích bộ phân loại cho kernel phi tuyến. Khó khăn trong việc xác định ảnh hưởng của các mô tả.
- Việc xử lý vấn đề đa lớp vẫn là một vấn đề mở (hướng nghiên cứu kế tiếp)

SVM – Mở rộng

Trong phân loại, SVM có thể được áp dụng cho :

- Học bán giám sát (dữ liệu được gán nhãn một phần)
- Hồi quy vector hỗ trợ (vấn đề hồi quy)
- Hỗ trợ phân cụm vector (vấn đề phân cụm)

Các chức năng kernel cụ thể được phát triển (khai thác văn bản, khai thác hình ảnh, nhận dạng giọng nói,...). Chúng phải được điều chỉnh cho phù hợp với khái niệm về sự giống nhau giữa các quan sát trong lĩnh vực.

Phương pháp SVM rất gần với nghiên cứu, thường có những phát triển mới và cải tiến. Tất cả các biến thể mới này không phải lúc nào cũng có sẵn trong các công cụ thông thường.

The End