Hồi quy Logistic, Máy véc-tơ hỗ trợ (SVM)

Nguyễn Thanh Tùng
Khoa Công nghệ thông tin – Đại học Thủy Lợi
tungnt@tlu.edu.vn

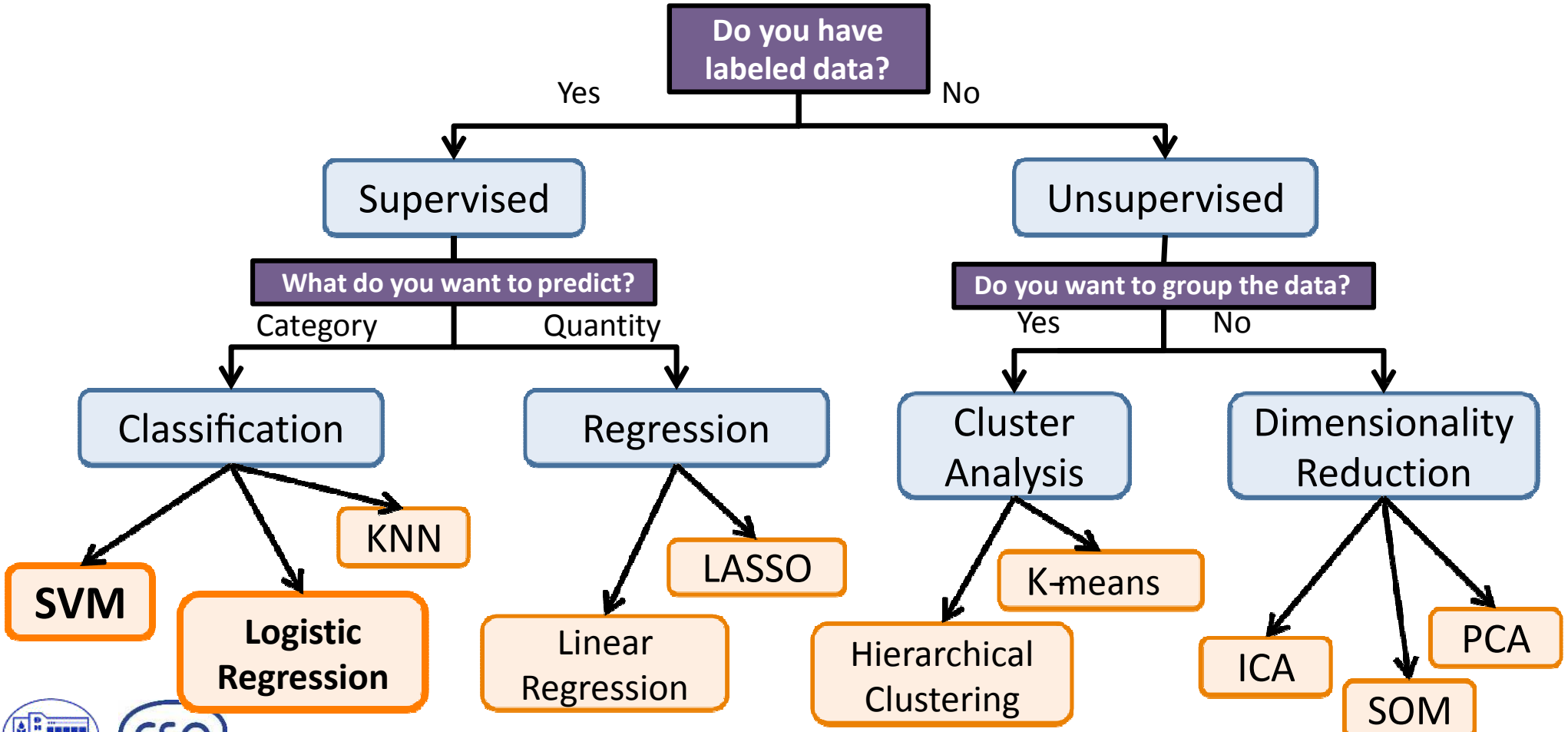
Website môn học: https://sites.google.com/a/wru.vn/cse445Fall2017

Bài giảng có sử dụng hình vẽ trong cuốn sách "An Introduction to Statistical Learning with Applications in R" với sự cho phép của tác giả, có sử dụng slides các khóa học CME250 của ĐH Stanford và IOM530 của ĐH Southern California

Hồi quy Logit

(Logistic Regression)

Types of Algorithms



Phân lớp

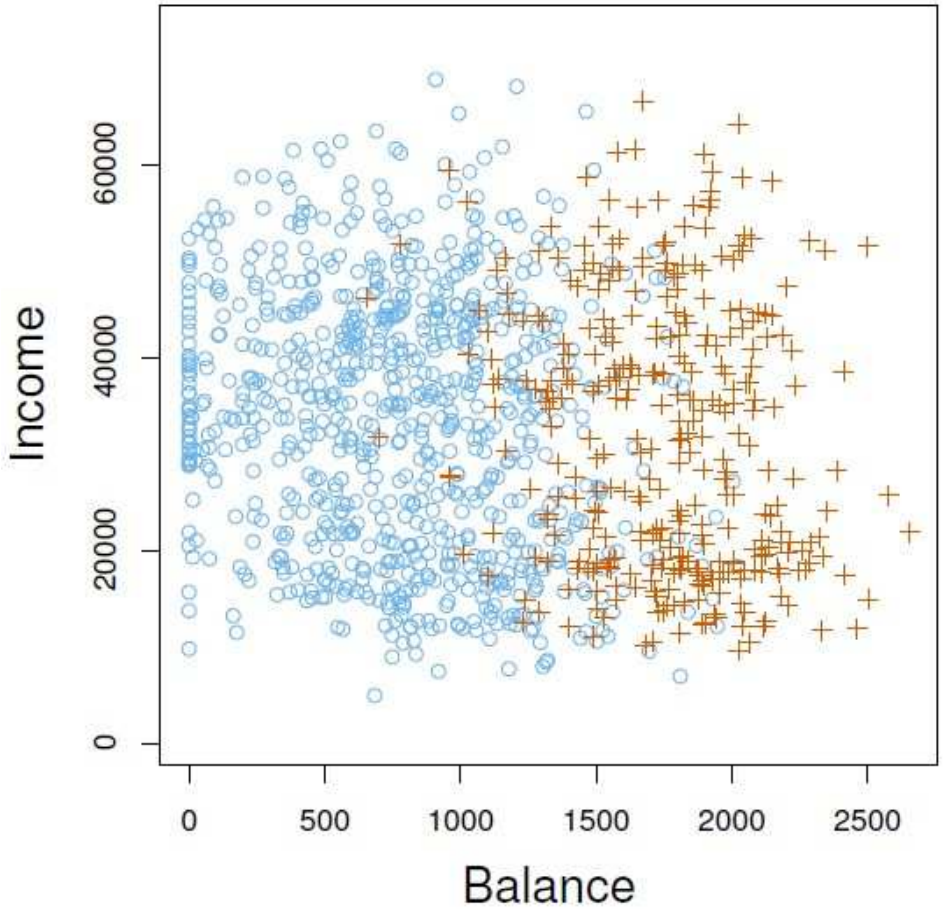
- Hồi quy – dự đoán biến định lượng (liên tục) Y
  - Trong nhiều ứng dụng, biến đầu ra là định tính hoặc kiểu định danh/hạng mục

- Phân lớp: Dự đoán biến đầu ra định tính
  - Gán mỗi quan sát cho một lớp/mục
  - vd: Bộ phân lớp *K-láng giềng gần nhất* trong bài học trước

Ví dụ về phân lớp

- *Các giao dịch thẻ tín dụng*
  - Có phải dịch gian lận hay không khi ta dựa trên thông tin lịch sử giao dịch của chúng?
- *Rủi ro tín dụng*
  - Liệu một cá nhân có bị vỡ nợ với tài khoản tín dụng của mình không?
- *Thị giác máy (Computer Vision)*
  - Hiểu được các đối tượng xuất hiện trong ảnh

Ví dụ về phân lớp



Hình 4.1 , ISL 2013*

Phân lớp và Hồi quy

- Phân lớp và Hồi quy có liên quan với nhau lớn.
- Phân lớp hoạt động như hồi quy:
  - Dự đoán xác suất của 1 mẫu dữ liệu thuộc vào một lớp, ta gán vào 1 lớp có xác suất cao nhất

Hồi quy Logistic

- *Phân lớp nhị phân*: Y nhận 2 giá trị ("0" hoặc "1") với 2 lớp tương ứng
- Mô hình hồi quy Logistic đối với bài toán phân lớp nhị phân

$$Pr\left(Y \text{ belongs to class } 1 \mid X\right)$$

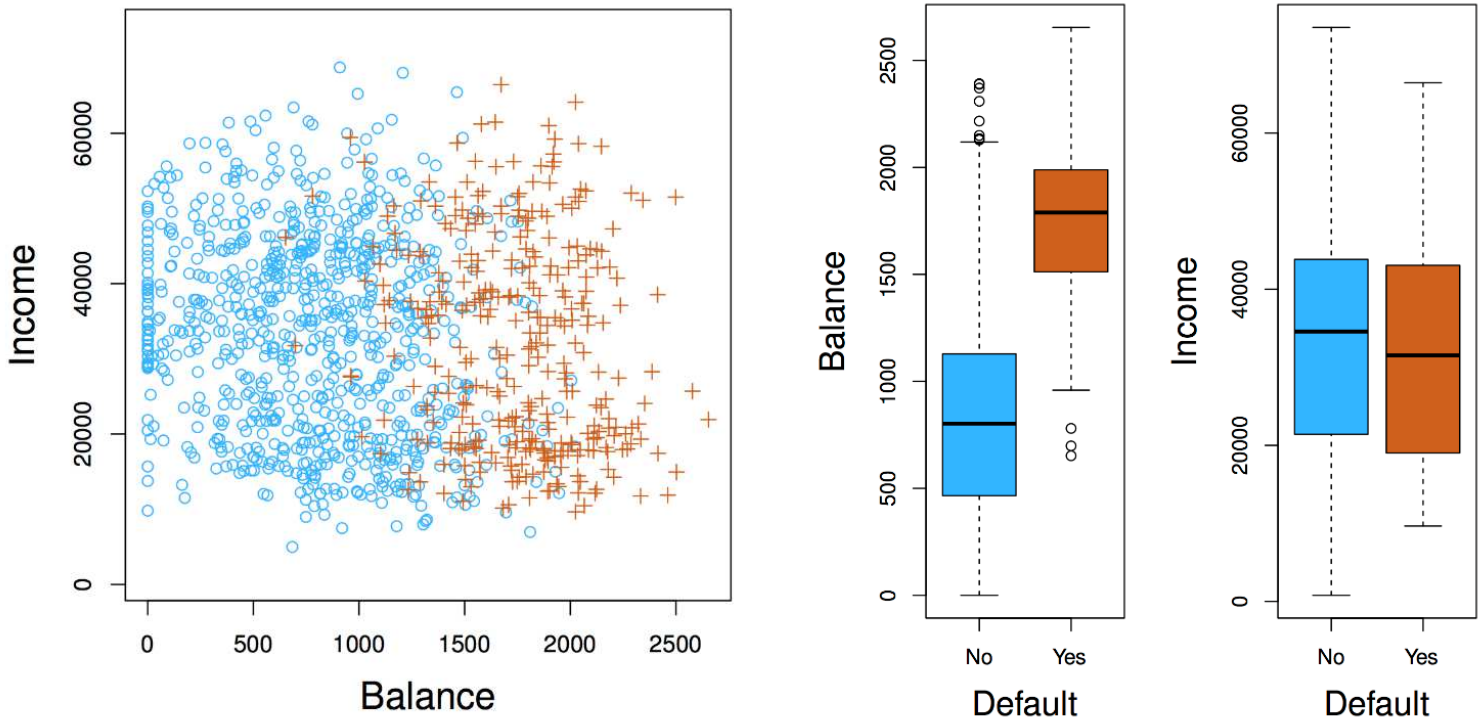
  - Ngưỡng để đạt được các quyết định phân lớp
  - Là mô hình hồi quy tuyến tính có chỉnh sửa để dự đoán xác suất trong [0, 1]


Ví dụ: Dữ liệu Credit Card Default

- Ta cần dự đoán các khách hang có nguy cơ phá sản (default)
- Các biến X là:
  - Thu nhập thường xuyên (Annual Income)
  - Cân đối thẻ hàng tháng (Monthly credit card balance)
- Biến Y (Default) có kiểu rời rạc (categorical): Yes hoặc No
- Làm sao để tìm quan hệ giữa Y và X?

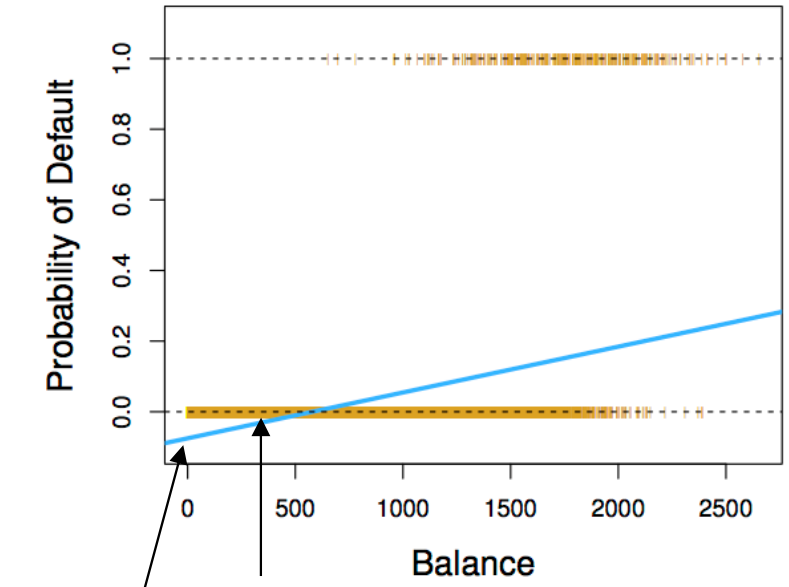
	default	student	balance	income
1	No	No	729.52650	44361.625
2	No	Yes	817.18041	12106.135
3	No	No	1073.54916	31767.139
4	No	No	529.25060	35704.494
5	No	No	785.65588	38463.496
6	No	Yes	919.58853	7491.559
7	No	No	825.51333	24905.227
8	No	Yes	808.66750	17600.451
9	No	No	1161.05785	37468.529
10	No	No	0.00000	29275.268
11	No	Yes	0.00000	21871.073
12	No	Yes	1220.58375	13268.562
13	No	No	237.04511	28251.695
14	No	No	606.74234	44994.556
15	No	No	1112.96840	23810.174
16	No	No	286.23256	45042.413
17	No	No	0.00000	50265.312

Tập dữ liệu Default



Tại sao không dùng hồi quy tuyến tính?

- Khi Y chỉ nhận giá trị Yes hoặc No (1 hoặc 0), tại sao mô hình hồi quy tuyến tính không thích hợp?
- Nếu ta xây dựng mô hình hồi quy tuyến tính trên dữ liệu Default, thì với những cân đối tài chính thấp (low balances) ta sẽ dự đoán một xác suất âm, và với cân đối cao ta sẽ dự đoán xác suất trên 1!



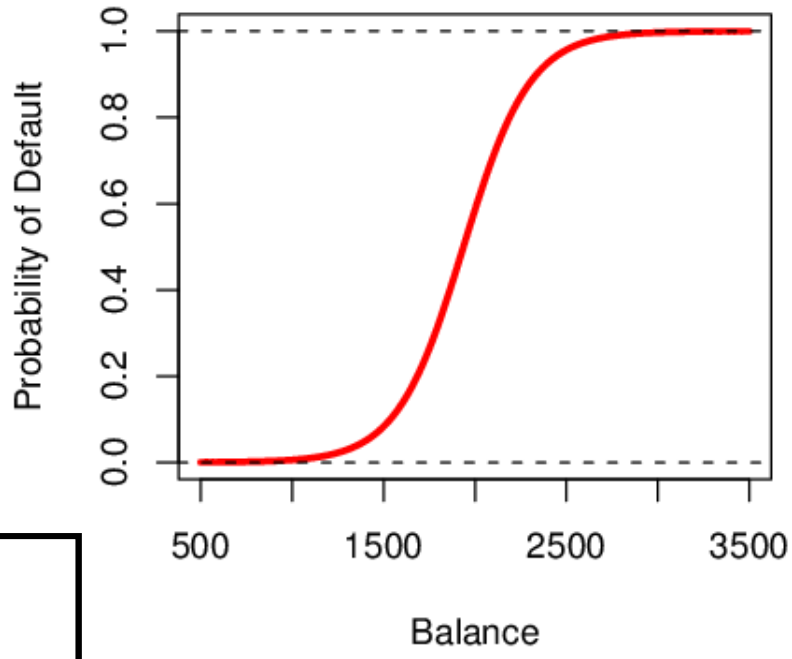
Khi biến Balance < 500,

Pr(default) là số âm!

Diễn giải giá trị nhỏ hơn 0 thế nào?

Hàm Logistic trên dữ liệu Default

Xác suất của việc phá sản sát 0 nhưng không âm đối với các tài khoản có cân bằng tài chính thấp, tương tự với cân bằng tài chính cao sẽ sát nhưng không lớn hơn 1

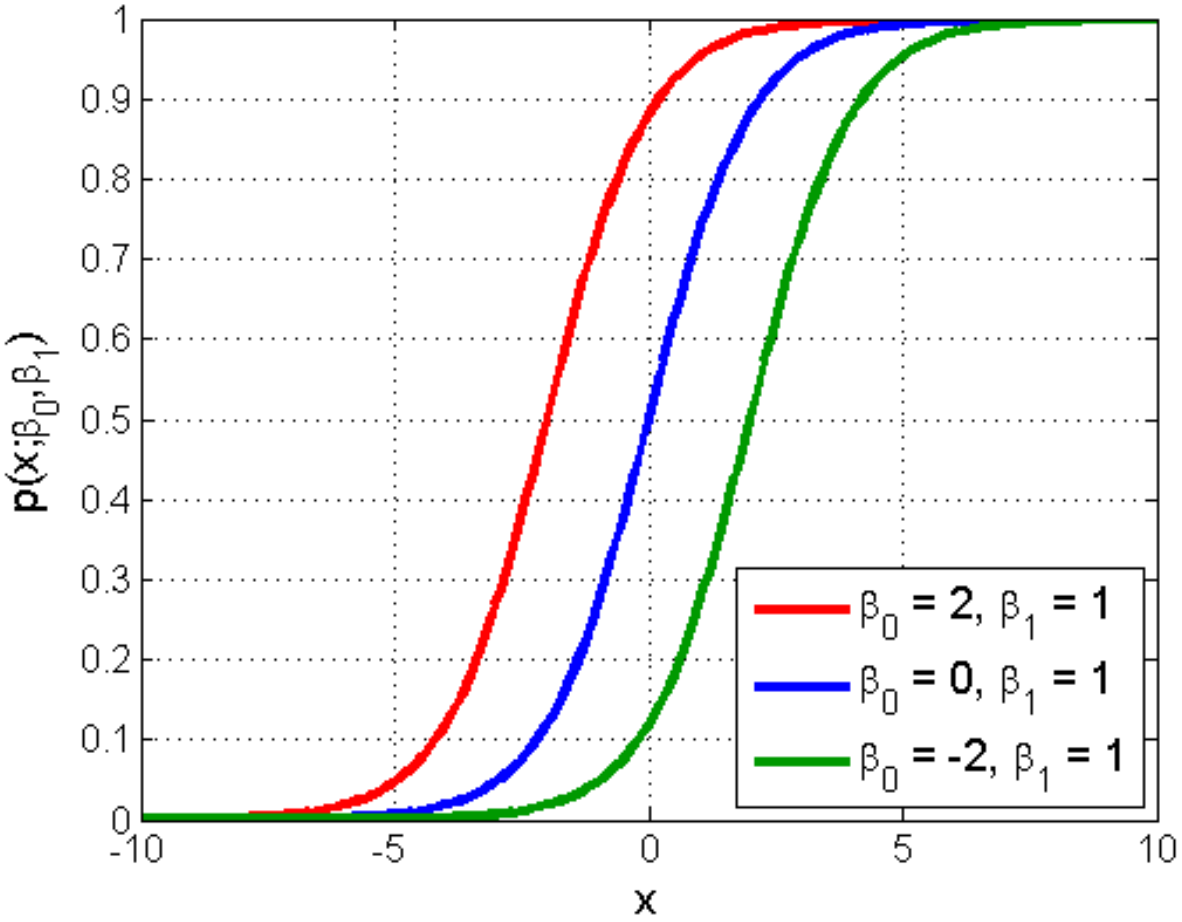


$$p = P(Y = 1) = \frac{e^{\beta_0 + \beta_1 X}}{1 + e^{\beta_0 + \beta_1 X}}$$

Diễn giải giá trị β_1

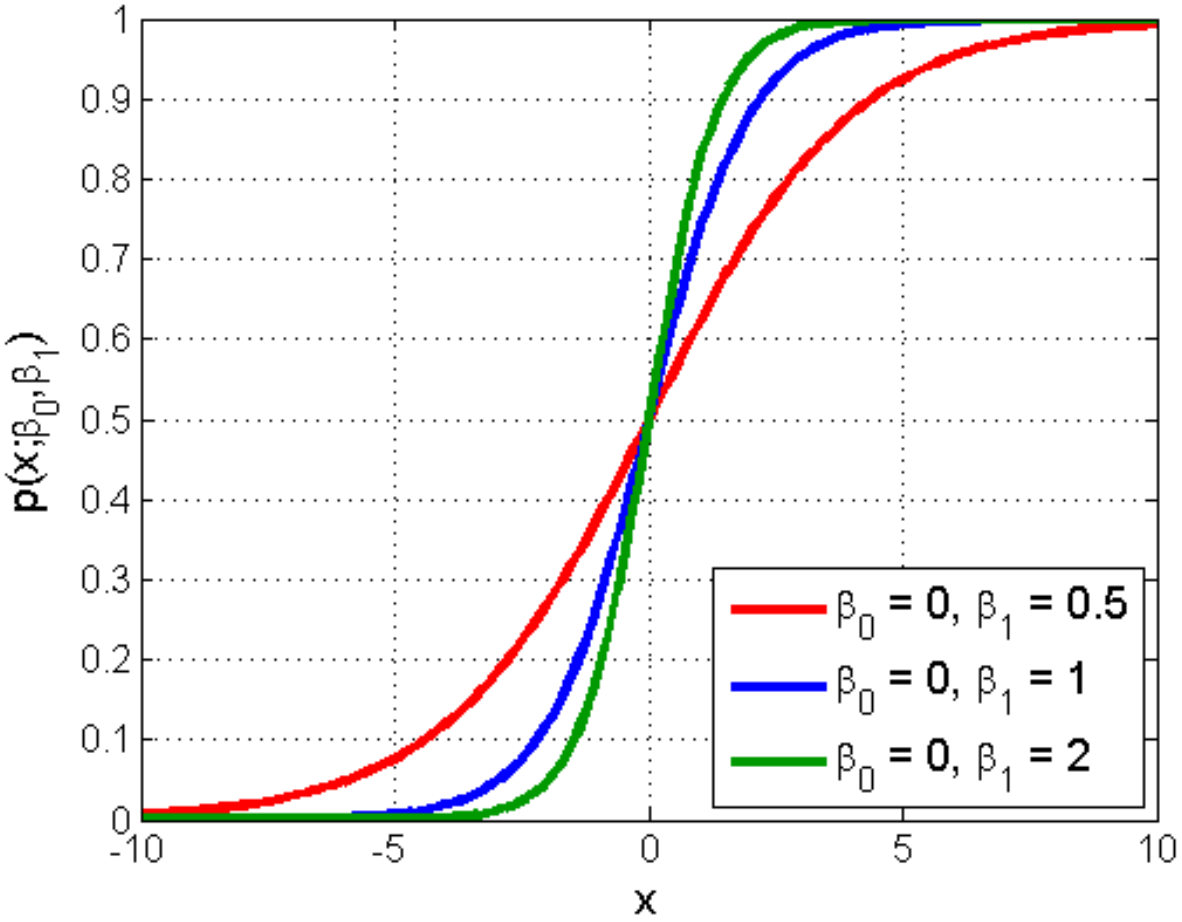
- Việc diễn giải ý nghĩa của β_1 không dễ đối với hồi quy logistic, bởi vì ta đang dự đoán xác suất P(Y), không phải giá trị Y.
- Nếu β_1 =0, có nghĩa là không tồn tại mối quan hệ giữa Y và X.
- Nếu β_1 >0, nghĩa là khi X nhận giá trị lớn hơn đồng nghĩa với việc tăng xác suất của Y = 1.
- Nếu β_1 <0, nghĩa là khi X nhận giá trị lớn hơn, xác suất mà Y = 1 nhỏ đi.
- Tuy nhiên giá trị lớn hoặc nhỏ hơn là bao nhiêu lại phụ thuộc vào vị trí ta đang đứng ở độ dốc (the slope) nào

Hồi quy Logistic



$$p(X) = \frac{e^{(\beta_0 + \beta_1 X)}}{1 + e^{(\beta_0 + \beta_1 X)}}$$

Hồi quy Logistic



$$p(X) = \frac{e^{(\beta_0 + \beta_1 X)}}{1 + e^{(\beta_0 + \beta_1 X)}}$$

Ý nghĩa của các hệ số?

- Ta thực hiện kiểm định giả thuyết để xem ý nghĩa của các hệ số β_0 và β_1.
- Ta dùng kiểm định Z thay thế cho T-test, nhưng việc diễn giải p-value không thay đổi
- Trong ví dụ này, p-value cho biến balance rất nhỏ, và β_1 dương, vì vậy ta có thể khẳng định rằng khi biến balance tăng thì xác suất phá sản cũng tăng theo

	Coefficient	Std. Error	Z-statistic	P-value
Intercept	-10.6513	0.3612	-29.5	< 0.0001
balance	0.0055	0.0002	24.9	< 0.0001

Dự đoán

- Giả sử mỗi cá nhân có cân đối tài chính trung bình là $1000. Xác suất phá sản là bao nhiêu?

$$\hat{p}(X) = \frac{e^{\hat{\beta}_0+\hat{\beta}_1 X}}{1+e^{\hat{\beta}_0+\hat{\beta}_1 X}} = \frac{e^{-10.6513+0.0055\times 1000}}{1+e^{-10.6513+0.0055\times 1000}} = 0.00576$$

- Xác suất phá sản dự đoán cho mỗi cá nhân với cân đối tài chính $1000 là nhỏ hơn 1%.
- Với cân đối là $2000, xác suất lớn hơn và kết quả là 0.586 (58.6%).

Biến X rời rạc trong Hồi quy Logistic

- Ta có thể dự đoán từng cá nhân phá sản với việc kiểm tra xem người đó có phải là sinh viên hay không. Do đó, ta sử dụng biến rời rạc "Student" được mã như sau: Student = 1, Non-student =0.
- β_1 dương: Điều này chỉ ra rằng sinh viên có xu hướng xác suất vỡ nợ cao hơn là người không phải là sinh viên

	Coefficient	Std. Error	Z-statistic	P-value
Intercept	-3.5041	0.0707	-49.55	< 0.0001
student[Yes]	0.4049	0.1150	3.52	0.0004

$$\widehat{\Pr}(\texttt{default=Yes}|\texttt{student=Yes}) = \frac{e^{-3.5041+0.4049\times 1}}{1+e^{-3.5041+0.4049\times 1}} = 0.0431,$$

$$\widehat{\Pr}(\texttt{default=Yes}|\texttt{student=No}) = \frac{e^{-3.5041+0.4049\times 0}}{1+e^{-3.5041+0.4049\times 0}} = 0.0292.$$

Hồi quy Logistic đa biến

- Ta có thể mở rộng hồi quy logistic với trường hợp nhiều biến đầu vào:

$$Pr(Y = 1 \mid X) = \sigma(\beta_0 + \beta_1 X_1 + \cdots + \beta_d X_d)$$

$$= \frac{e^{(\beta_0 + \beta_1 X_1 + \cdots + \beta_d X_d)}}{1 + e^{(\beta_0 + \beta_1 X_1 + \cdots + \beta_d X_d)}}$$

Hồi quy Logistic đa biến- Default Data

Dự đoán khả năng vỡ nợ (Default) dùng:

- Balance (dữ liệu số, quantitative)
- Income (dữ liệu số, quantitative)
- Student (rời rạc, qualitative)

	Coefficient	Std. Error	Z-statistic	P-value
Intercept	-10.8690	0.4923	-22.08	< 0.0001
balance	0.0057	0.0002	24.74	< 0.0001
income	0.0030	0.0082	0.37	0.7115
student[Yes]	-0.6468	0.2362	-2.74	0.0062

Dự đoán

Một sinh viên với mức cân đối thẻ tín dụng là $1,500 và tổng thu nhập là $40,000 có xác suất dự đoán khả năng vỡ nợ như sau

$$\hat{p}(X) = \frac{e^{-10.869+0.00574\times 1500+0.003\times 40-0.6468\times 1}}{1+e^{-10.869+0.00574\times 1500+0.003\times 40-0.6468\times 1}} = 0.058.$$

Mâu thuẫn kiểu biến!

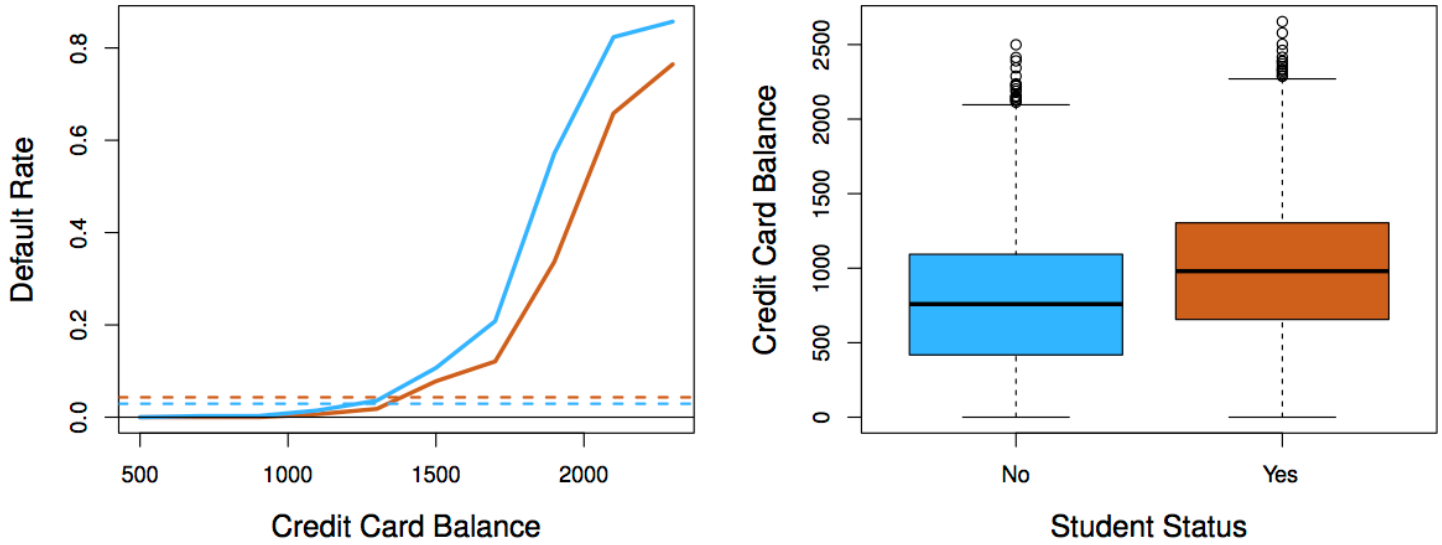
	Coefficient	Std. Error	Z-statistic	P-value
Intercept	-3.5041	0.0707	-49.55	< 0.0001
student[Yes]	0.4049	0.1150	3.52	0.0004

Dương

	Coefficient	Std. Error	Z-statistic	P-value
Intercept	-10.8690	0.4923	-22.08	< 0.0001
balance	0.0057	0.0002	24.74	< 0.0001
income	0.0030	0.0082	0.37	0.7115
student[Yes]	-0.6468	0.2362	-2.74	0.0062

Âm

Sinh viên (màu cam) vs. Không phải sinh viên (màu xanh)



- Sinh viên rủi ro hơn là người không phải sinh viên nếu không có thông tin về mức cân đối thẻ tín dụng
- Tuy nhiên, sinh viên ít rủi ro hơn với đối tượng không phải là sinh viên nếu có cùng mức cân đối thẻ tín dụng!

Hồi quy Logistic

- Các tham số của mô hình β_0 và β_1 được ước lượng từ dữ liệu huấn luyện
  - Trong phương pháp hồi quy tuyến tính, ta *sử dụng bình phương nhỏ nhất*

- Tìm tham số mô hình hồi quy Logistic sử dụng phương pháp *Ước lượng hợp lý cực đại* (*maximum likelihood estimation)*


Câu hỏi?

Máy véc-tơ hỗ trợ
Support Vector Machines (SVMs)

Máy véc-tơ hỗ trợ

- Máy véc-tơ hỗ trợ được đề xuất bởi V. Vapnik và các đồng nghiệp của ông vào những năm 1970s ở Nga, và sau đó đã trở nên nổi tiếng và phổ biến vào những năm 1990s
- Phương pháp học phân loại có giám sát: Bài toán phân loại 2 lớp
- SVM nổi tiếng sau khi xử lý thành công bài toán nhận dạng chữ viết tay năm 1994.
- SVM được sử dụng rộng rãi cho bài toán phân lớp dữ liệu văn bản.

Bộ phân lớp có lề cực đại

- *Bộ phân lớp có lề cực đại*
  - Giả định quan trọng: Dữ liệu có 2 lớp tách được tuyến tính
    - SVM là một phương pháp **phân lớp tuyến tính** (linear classifier), với mục đích xác định một siêu phẳng để phân tách **hai lớp** của dữ liệu – ví dụ: lớp các mẫu có nhãn dương (positive) và lớp các mẫu có nhãn âm (negative)

- Tìm hiểu về siêu phẳng (*hyperplanes)…*

Các siêu phẳng

- Siêu phẳng là gì?
  - Trong không gian d-chiều, tồn tại một không gian con $(d-1)$-chiều

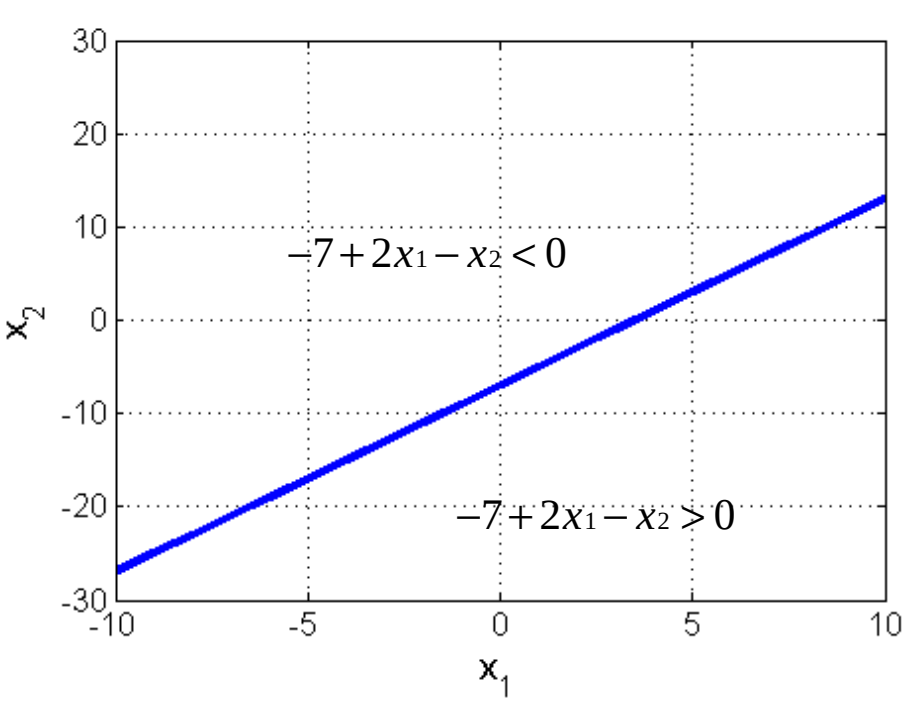
  Vd: đường thẳng trong 2D, máy bay trong không gian 3D
  - Siêu phẳng trong không gian *d*-chiều:

$$(\star)\ \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_d x_d = 0$$

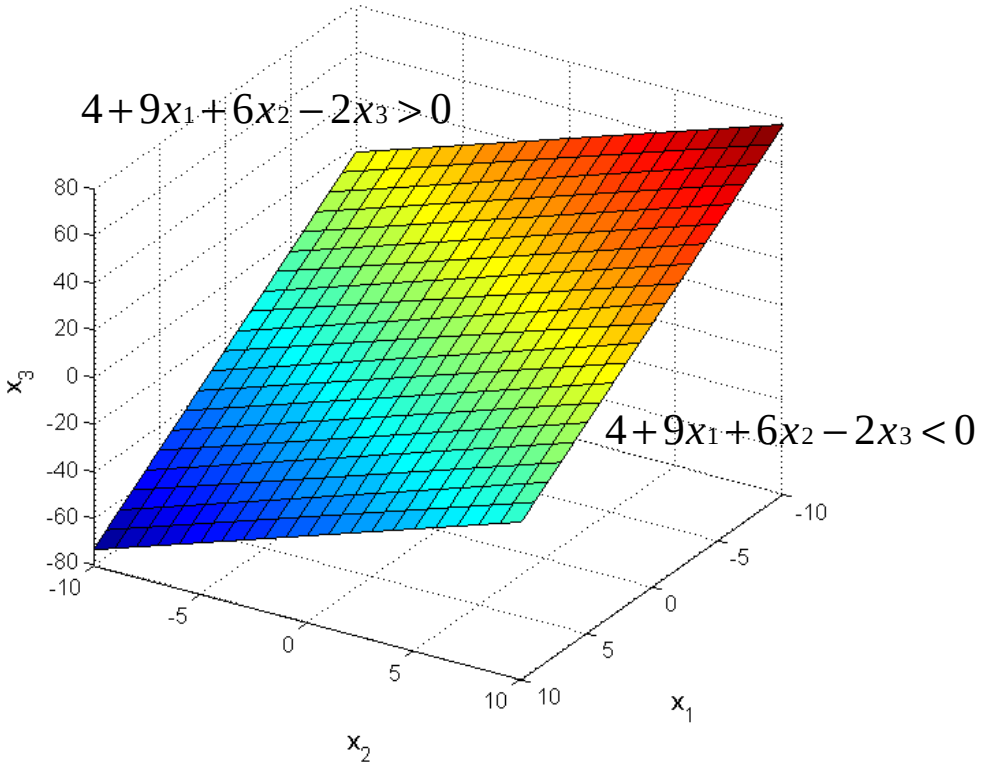
  - Tách không gian thành 2 nửa không gian con

Các siêu phẳng

$-7+2x_1-x_2=0$



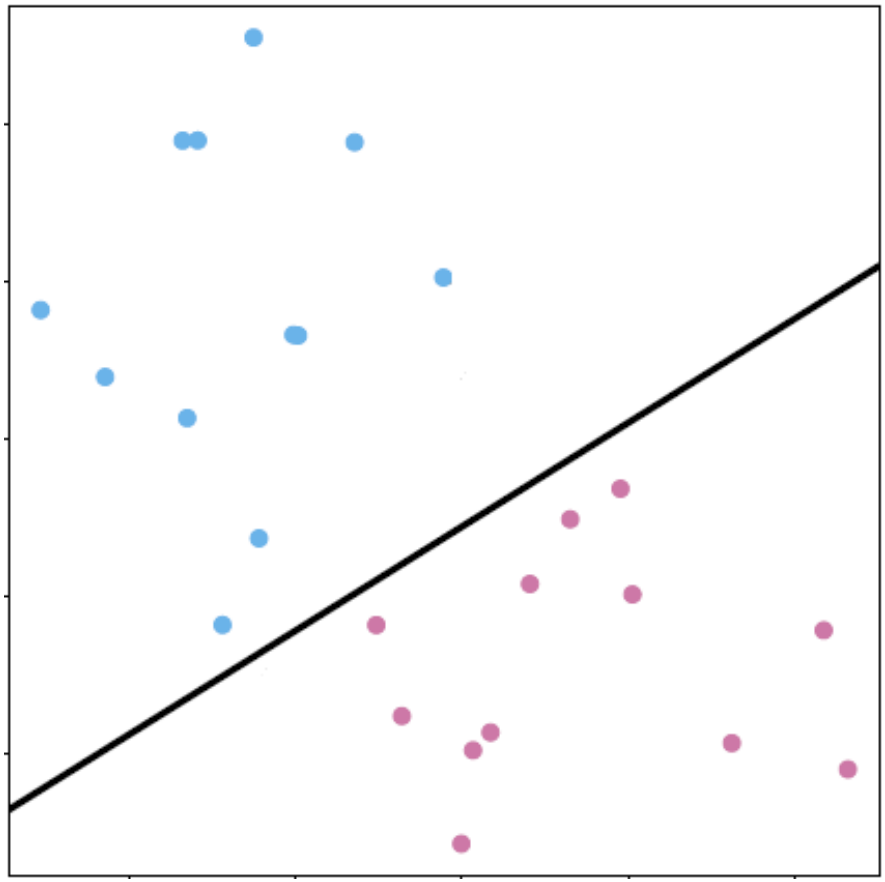
$4+9x_1+6x_2-2x_3=0$



Mặt siêu phẳng phân tách

Ý tưởng:
Dùng mặt siêu phẳng phân tách cho phân lớp nhị phân

Giả định:
Các lớp có thể tách được tuyến tính



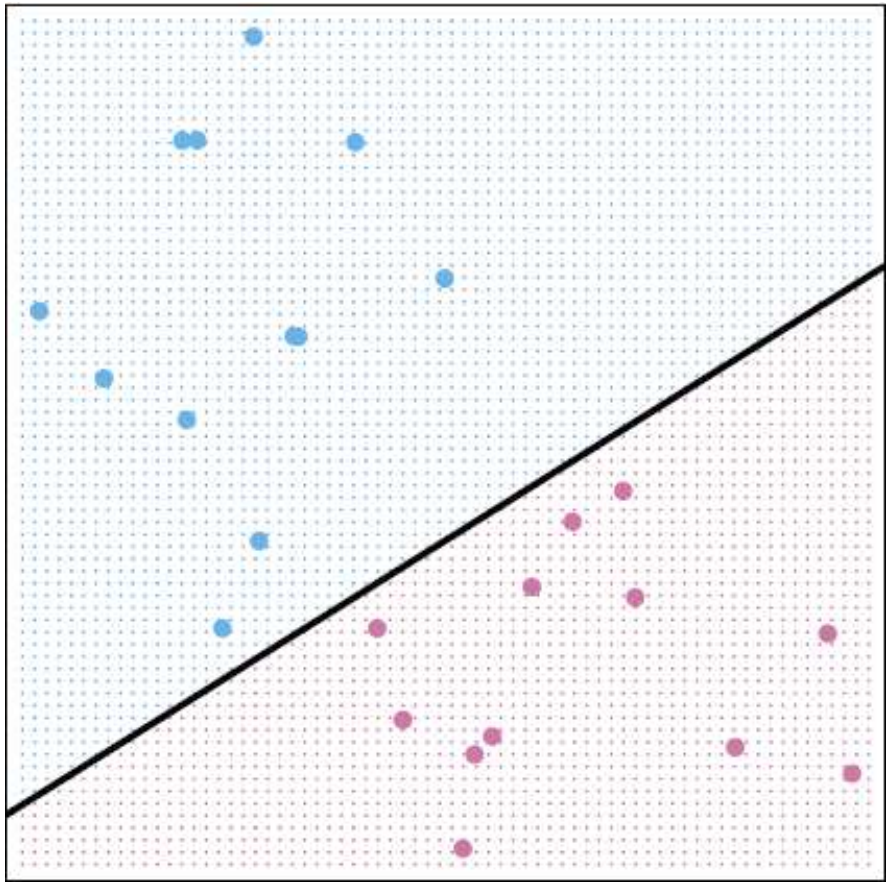
Hình 9.2 , ISL 2013

Mặt siêu phẳng phân tách

Phân lớp đối tượng mới:

Phân lớp dựa vào vị trí của đối tượng mới tương ứng với siêu phẳng:

$$\hat{Y} = \text{sign}\left(\beta_0 + \beta_1 X_1 + \cdots + \beta_d X_d\right)$$

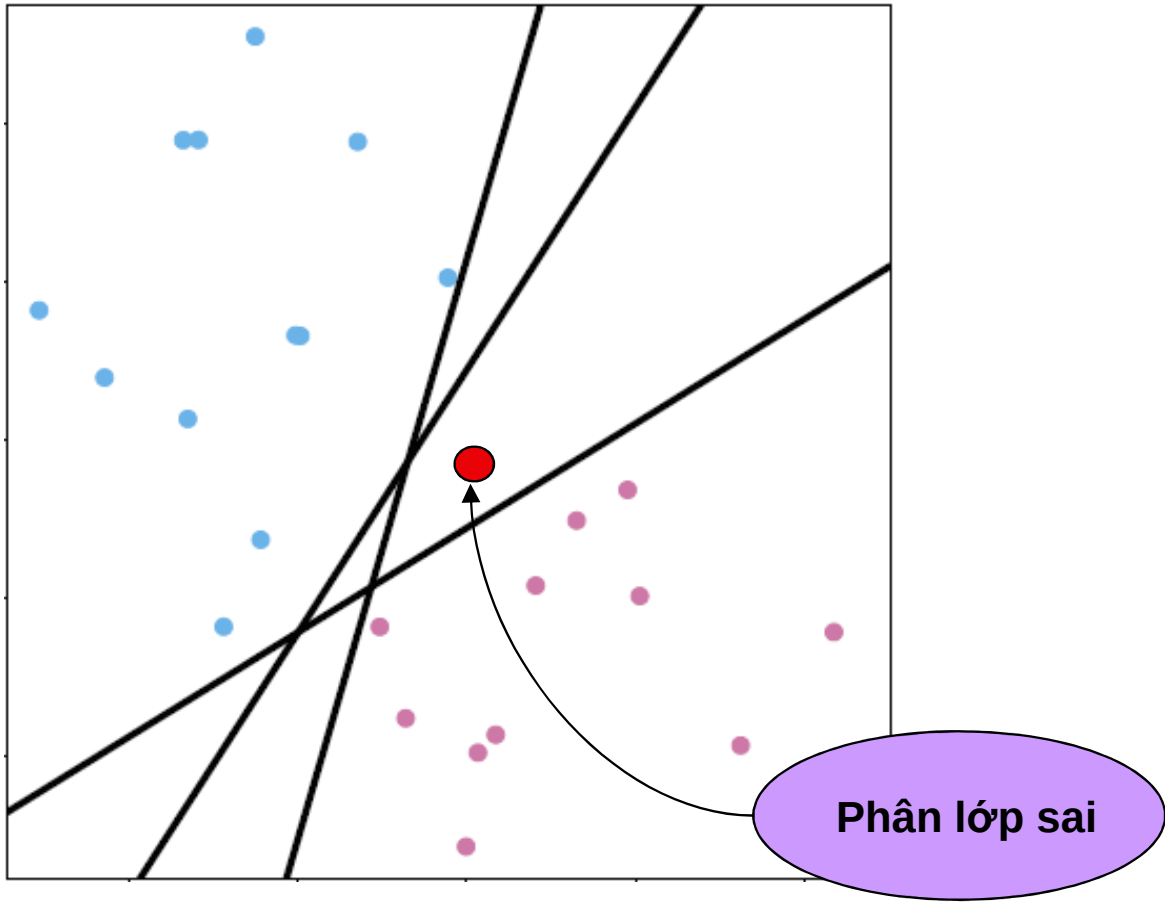


Hình 9.2 , ISL 2013

Mặt siêu phẳng phân tách

Giả định:
Các lớp có thể tách được tuyến tính

→ Tồn tại nhiều mặt phẳng tách…

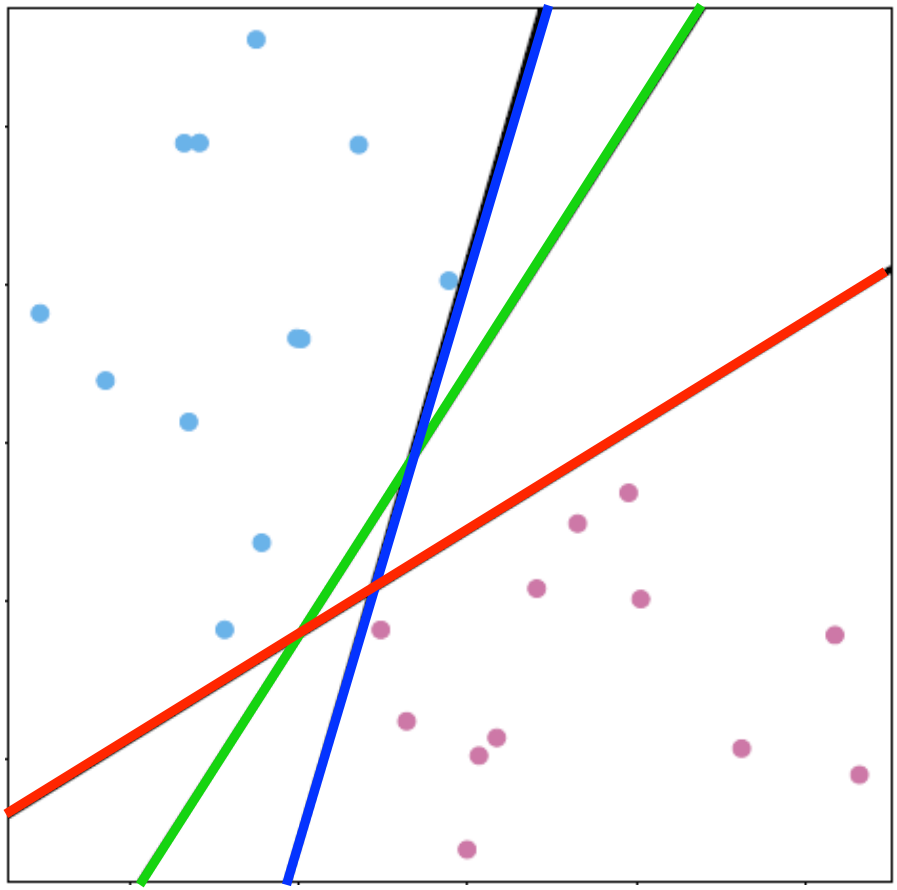


Hình 9.2 , ISL 2013

Câu hỏi:

Đường tách tuyến tính nào phù hợp?

Ta sử dụng tiêu chí gì để chọn?



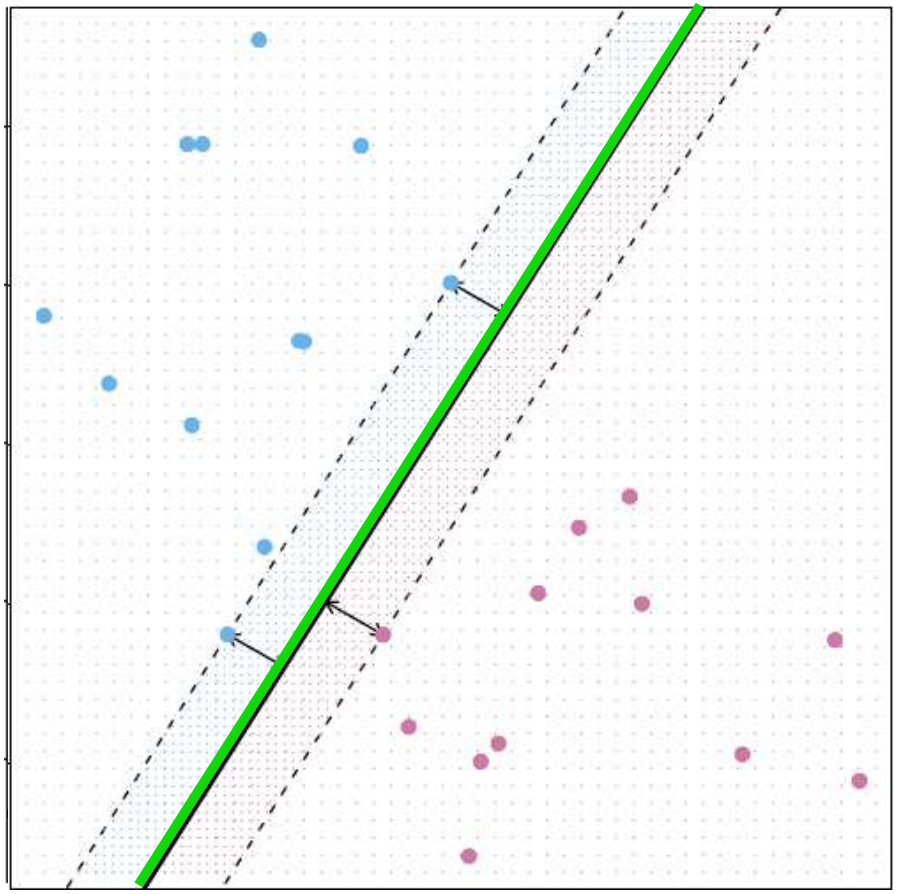
Hình 9.2 , ISL 2013

Mặt siêu phẳng phân tách

Đường tách tuyến tính nào phù hợp?

- Thay vì việc tìm 1 đường thẳng tách 2 lớp, ta tìm 2 đường thẳng tách các điểm này.
- Không có điểm DL nào nằm giữa 2 đường này.
- Đây là ý tưởng đơn giản nhất của SVM, gọi là SVM tuyến tính.

Mặt phẳng phân tách "xa nhất" từ tập dữ liệu huấn luyện
→ "Bộ phân lớp có lề cực đại"



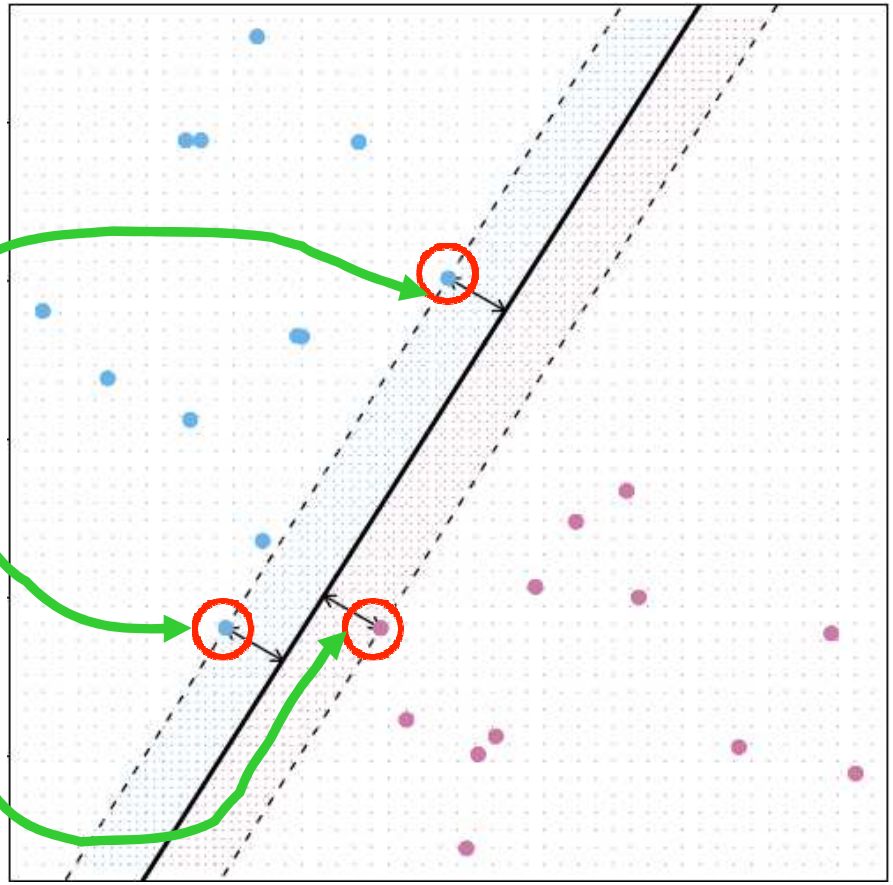
Hình 9.2, 9.3 , ISL 2013

Bộ phân lớp có lề cực đại

Siêu phẳng có lề cực đại

Siêu phẳng "xa nhất" từ tập huấn luyện→ Cực đại lề

Support Vectors là các điểm dữ liệu nằm trên 2 đường thẳng



Lề (Margin): khoảng cách nhỏ nhất giữa bất kỳ đối tượng nào trong tập huấn luyện và siêu phẳng

Véc-tơ hỗ trợ (Support vectors): Các đối tượng cách đều từ siêu phẳng

Hình 9.3 , ISL 2013

Bộ phân lớp có lề cực đại

- *Véc-tơ hỗ trợ*
  - Các đối tượng cách đều từ siêu phẳng có lề cực đại (maximal margin (MM) hyperplane)
  - "Hỗ trợ": siêu phẳng MM chỉ phụ thuộc vào các đối tượng (véc-tơ) này
    - Nếu các véc-tơ hỗ trợ bị nhiễu thì siêu phẳng MM sẽ thay đổi
    - Nếu bất kỳ một mẫu huấn luyện nào bị nhiễu, siêu phẳng MM không ảnh hưởng

Bộ phân lớp có lề cực đại

- Để tìm siêu phẳng có lề cực đại, ta giải:

$$\underset{(\beta_0,\ldots,\beta_d)}{\text{maximize}} \quad M$$

Cực đại lề, M

$$\text{subject to} \quad \sum_{j=0}^{d} \beta_j^2 = 1$$

Bài toán tối ưu có ràng buộc

$$Y^{(i)}\left(\beta_0 + \beta_1 X_1^{(i)} + \cdots + \beta_d X_d^{(i)}\right) \geq M, \ \forall i$$

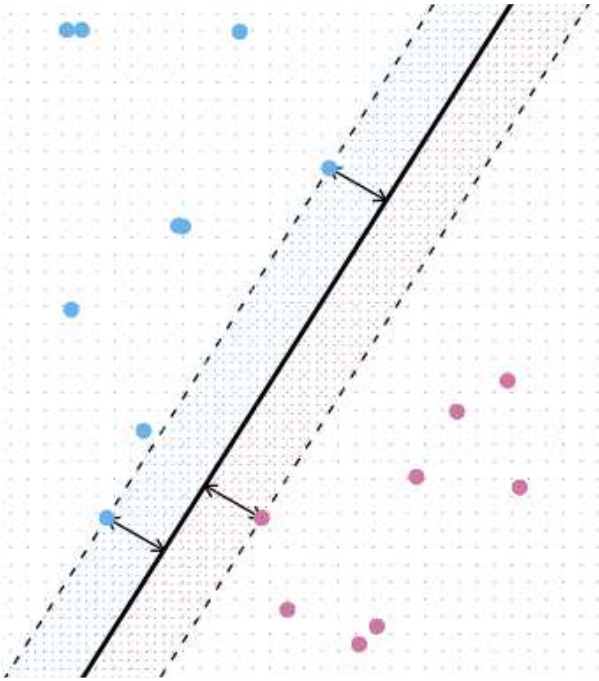
Tất cả các mẫu phải có khoảng cách tối thiểu M từ siêu phẳng

$Y^{(i)} \in \{-1, 1\}$ là các nhãn lớp

Tóm lược khái niệm cho SVM

- Xét bài toán dùng siêu phẳng tách 2 lớp
- Tịnh tiến song song siêu phẳng này về phía tập mẫu của mỗi lớp, quá trình này dừng khi có ít nhất 1 điểm thuộc siêu phẳng và không tiến thêm được nữa → *siêu phẳng lề*
- Hành lang nằm giữa 2 siêu phẳng gọi là *miền lề*
- Khoảng cách giữa 2 siêu phẳng gọi là *lề* của siêu phẳng tách
- Siêu phẳng tách tốt nhất có lề cực đại

→phương pháp tìm ra siêu phẳng tốt nhất này gọi là SVM

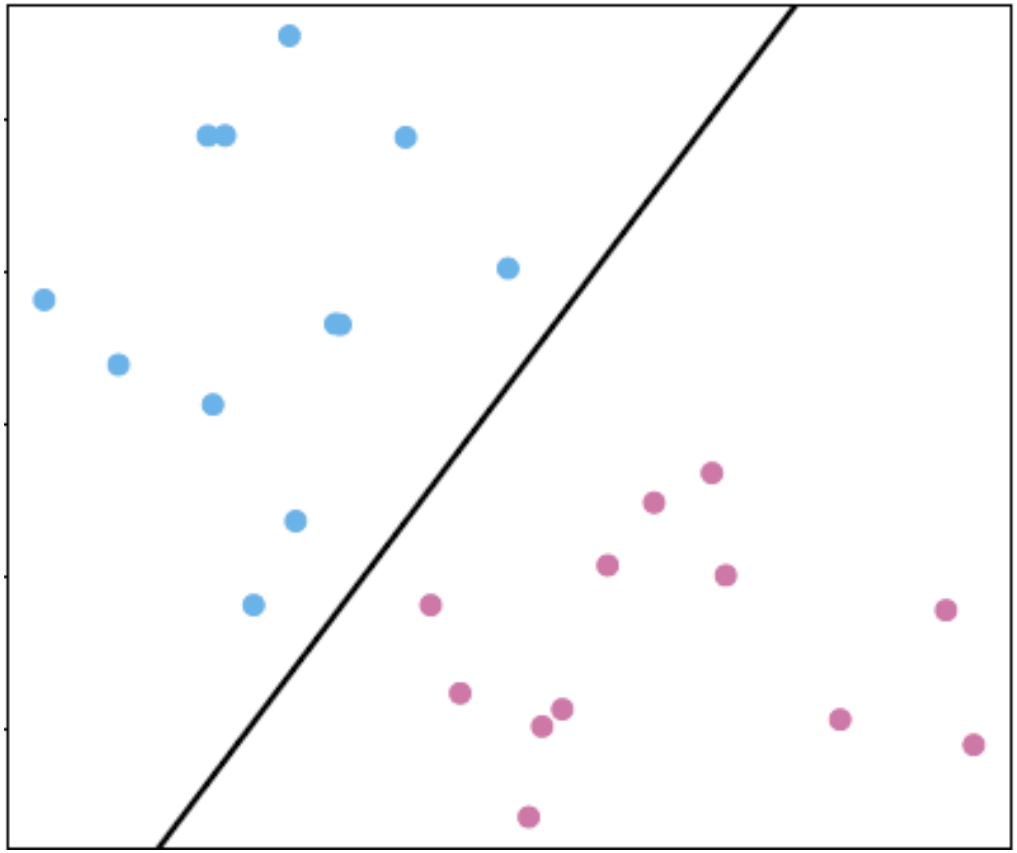


Hình 9.3 , ISL 2013

Bộ phân lớp có lề cực đại

Nhược điểm:

Có thể bị overfit trên dữ liệu huấn luyện



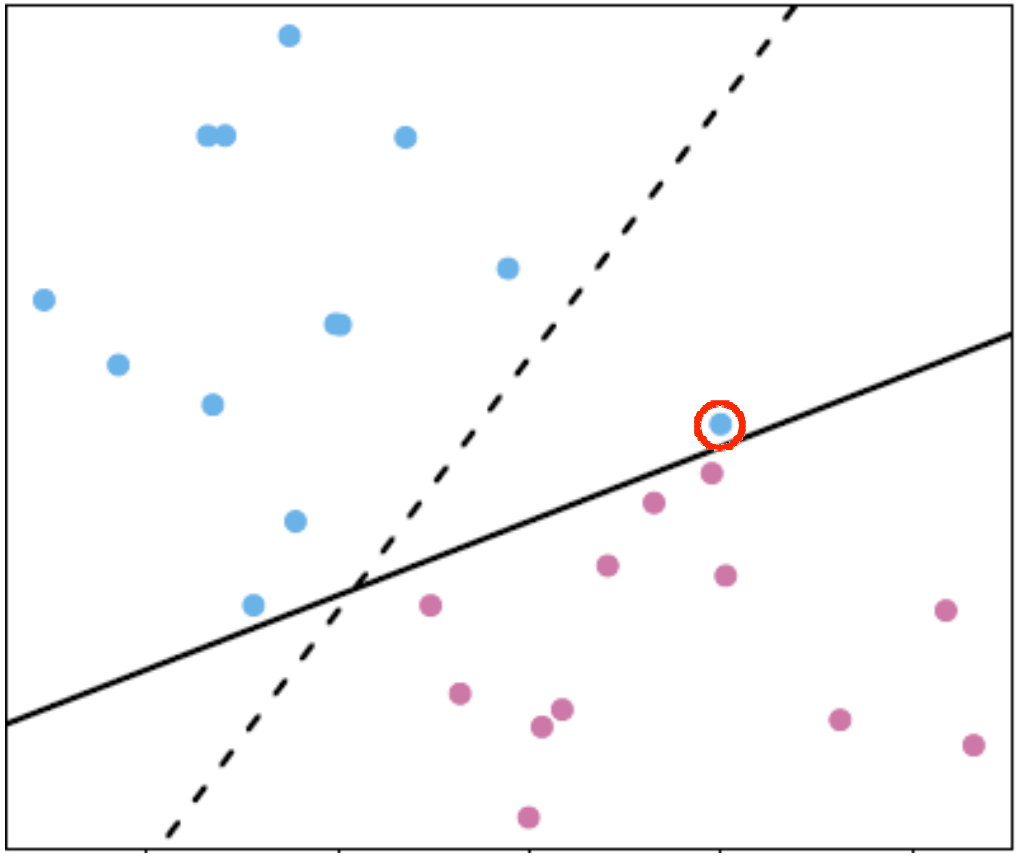
Hình 9.5 , ISL 2013

Bộ phân lớp có lề cực đại

Nhược điểm:

Có thể bị overfit trên dữ liệu huấn luyện

Nhạy cảm với các mẫu độc lập



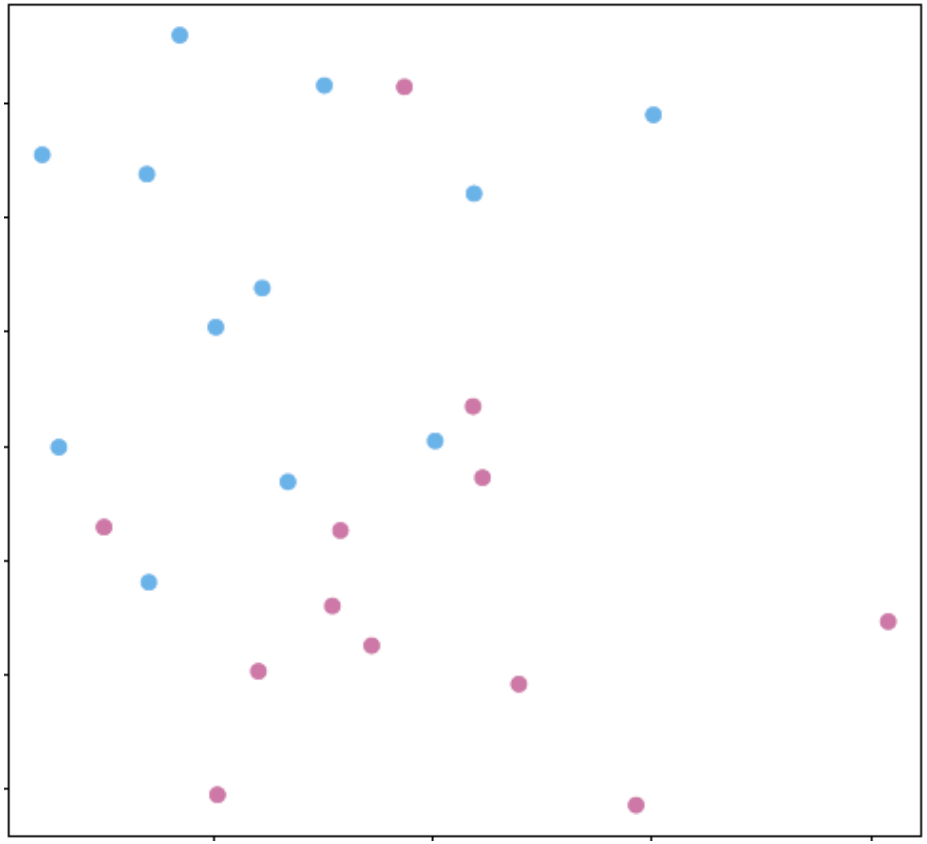
Hình 9.5 , ISL 2013

Bộ phân lớp có lề cực đại

Quay lại giả định trước:

Các lớp tách được bởi hàm tuyến tính

Điều gì xảy ra khi không tồn tại siêu phẳng tách?

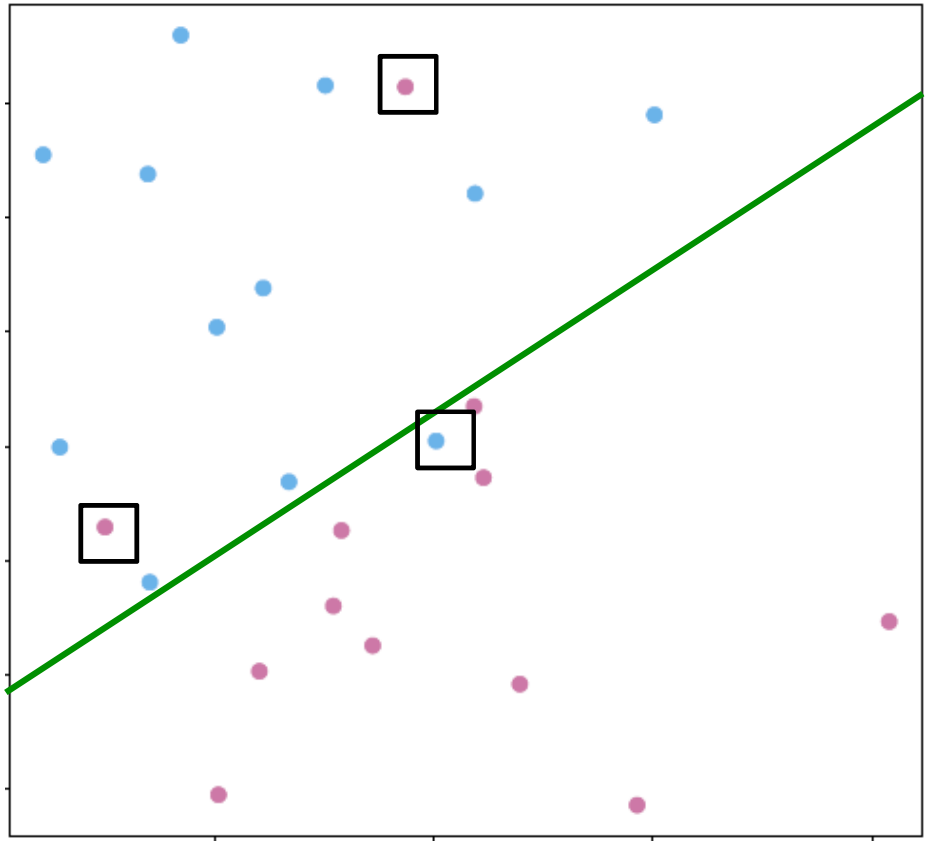


Hình 9.4 , ISL 2013

Bộ phân loại véc-tơ hỗ trợ

Điều gì xảy ra khi không tồn tại siêu phẳng tách?

Bộ phân loại véc-tơ hỗ trợ (Support Vector Classifier): cho phép các mẫu huấn luyện nằm phía phân loại sai "wrong side" của lề hoặc siêu phẳng

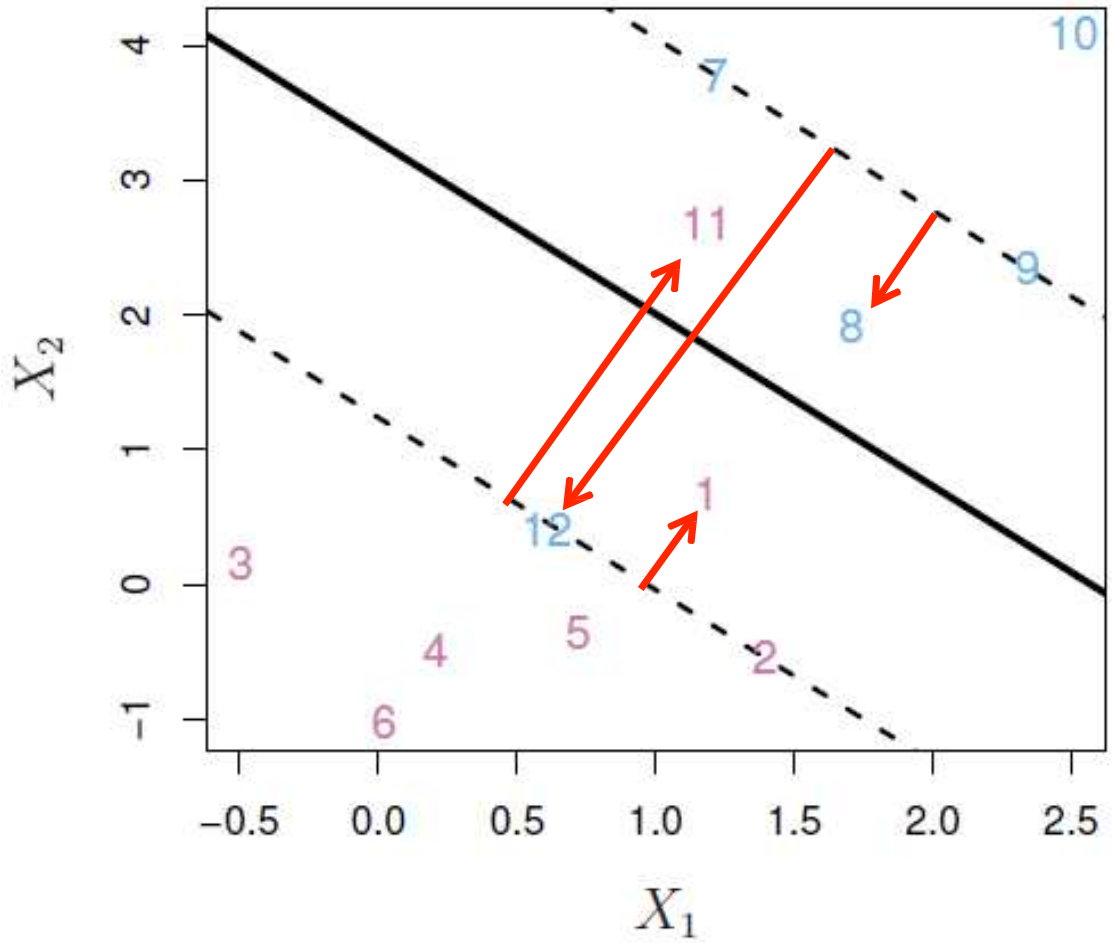


Hình 9.4 , ISL 2013

Bộ phân loại véc-tơ hỗ trợ

- *Support Vector Classifier*
  - Bộ phân loại dựa trên siêu phẳng
  - Cho phép một số mẫu trong tập huấn luyện nằm sai vị trí đối với lề/siêu phẳng
  - *Lề mềm (Soft margin)*: lề không cố định *ranh giới quyết định phân lớp* → Mục tiêu để mở rộng miền áp dụng, ta chấp nhận trong miền lề có lỗi nhưng ngoài miền lề phải phân lớp đúng

Bộ phân loại véc-tơ hỗ trợ



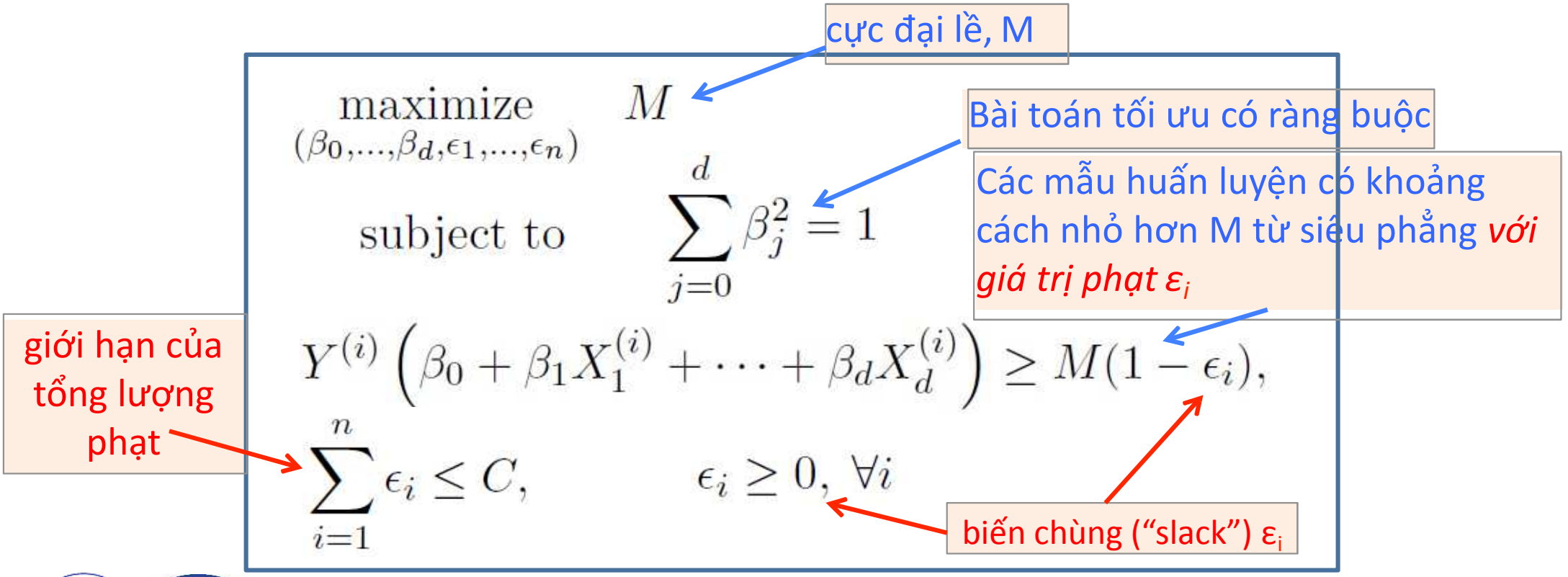
Hình 9.6 , ISL 2013

Bộ phân loại véc-tơ hỗ trợ

- *Support Vector Classifier*
  - Bộ phân loại dựa trên siêu phẳng
  - Cho phép một số mẫu trong tập huấn luyện nằm sai vị trí đối với lề/siêu phẳng
  - *Lề mềm (Soft margin)*: lề không có ranh giới cố định
- Ý tưởng: giải bài toán cực đại lề, nhưng cho phép một số lỗi (vi phạm) trong miền lề
  - Đưa thêm hệ số phạt để giới hạn số lượng/bậc của vi phạm

Bộ phân loại véc-tơ hỗ trợ

- Để tìm siêu phẳng cho bộ phân loại véc-tơ hỗ trợ, ta giải:



Bộ phân loại véc-tơ hỗ trợ

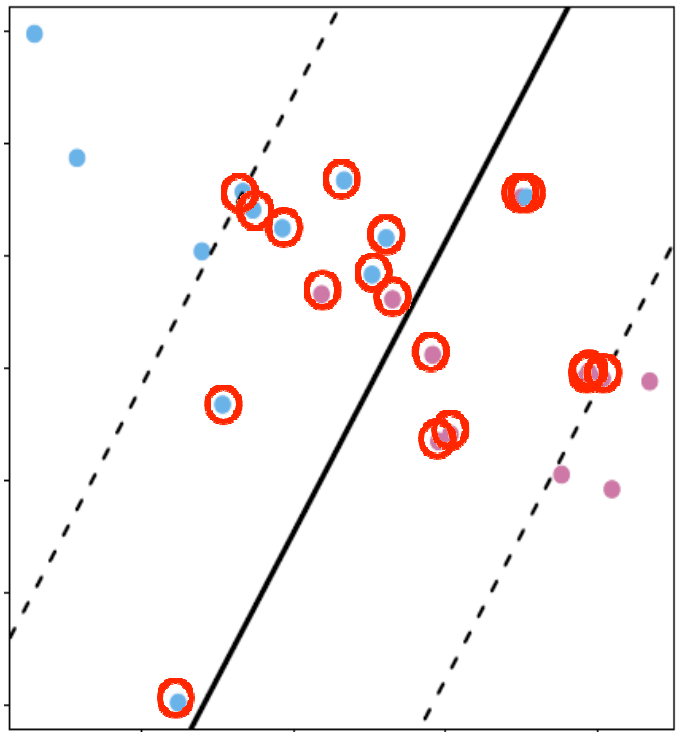
- Biến chùng (Slack) ε_i cho phép nới lỏng các vi phạm của lề
  - $\varepsilon_i = 0$: mẫu huấn luyện $X^{(i)}$ nằm đúng phía so với lề
  - $\varepsilon_i \in (0, 1)$: $X^{(i)}$ ở trong miền lề và phân lớp đúng
  - $\varepsilon_i > 1$: $X^{(i)}$ phân lớp sai (nằm sai vị trí so với siêu phẳng tách)
- Ta muốn tìm hàm quyết định có lề lớn nhất và số điểm có $\varepsilon_i > 0$ nhỏ nhất. Tham số phạt C (C>0) – biểu thị cho việc phạt các điểm phân lớp sai. **C càng lớn thì lề càng hẹp**, C$\rightarrow \infty$ ứng với trường hợp tách được tuyến tính.
  - Gán các giá trị chi phí C (cost) cho các lỗi. Cho phép nhiều nhất C phân lớp sai trên tập dữ liệu huấn luyện


Bộ phân loại véc-tơ hỗ trợ

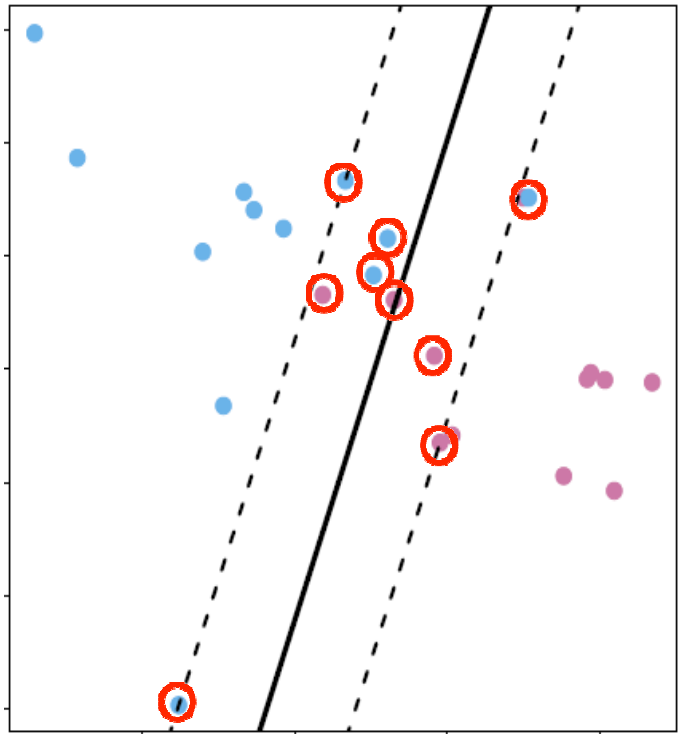
"Misclassification budget" tham số C được chọn bởi kỹ thuật *cross-validation*

điều khiển cân bằng bias-variance

Các véc-tơ hỗ trợ: các mẫu trên lề hoặc vi phạm lề



Large budget
many Support Vectors



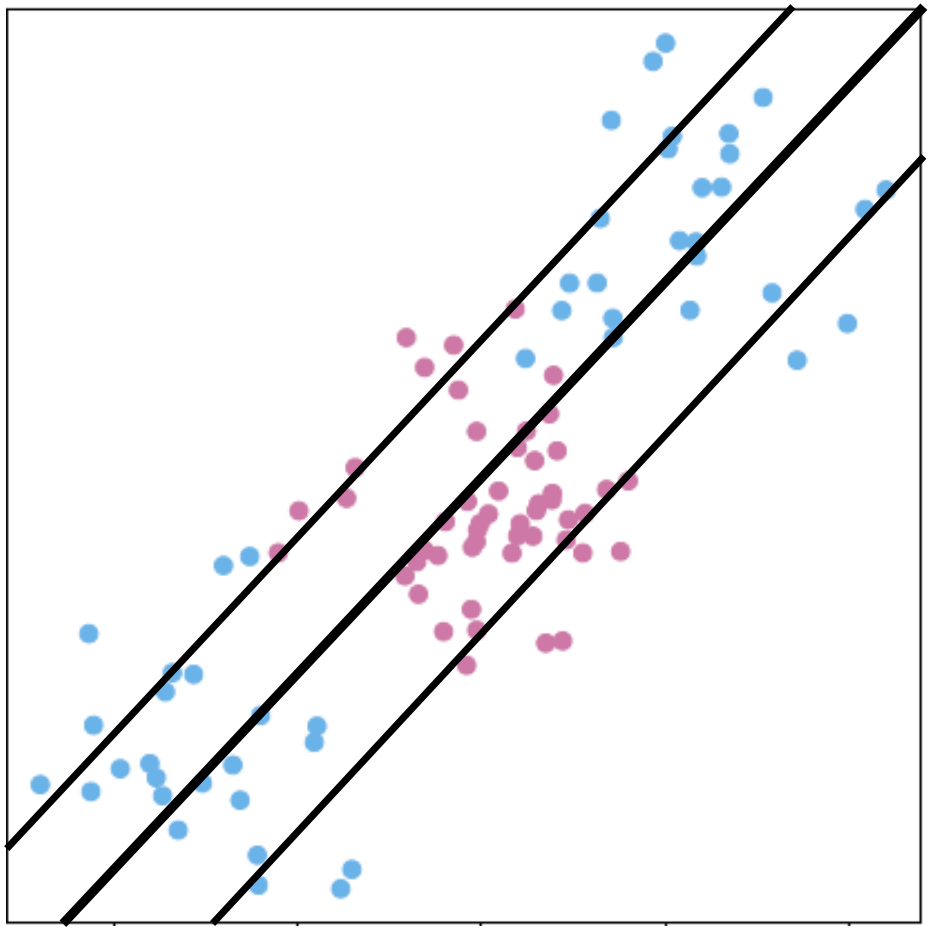
Small budget
fewer Support Vectors

Hình 9.7 , ISL 2013

Bộ phân loại véc-tơ hỗ trợ

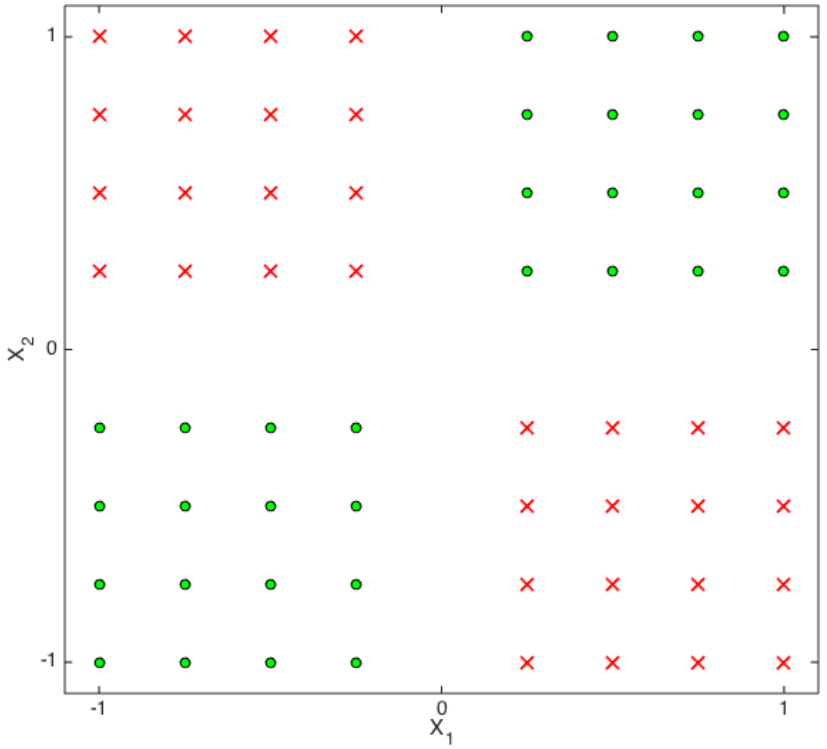
Nhược điểm:

Ranh giới quyết định tuyến tính

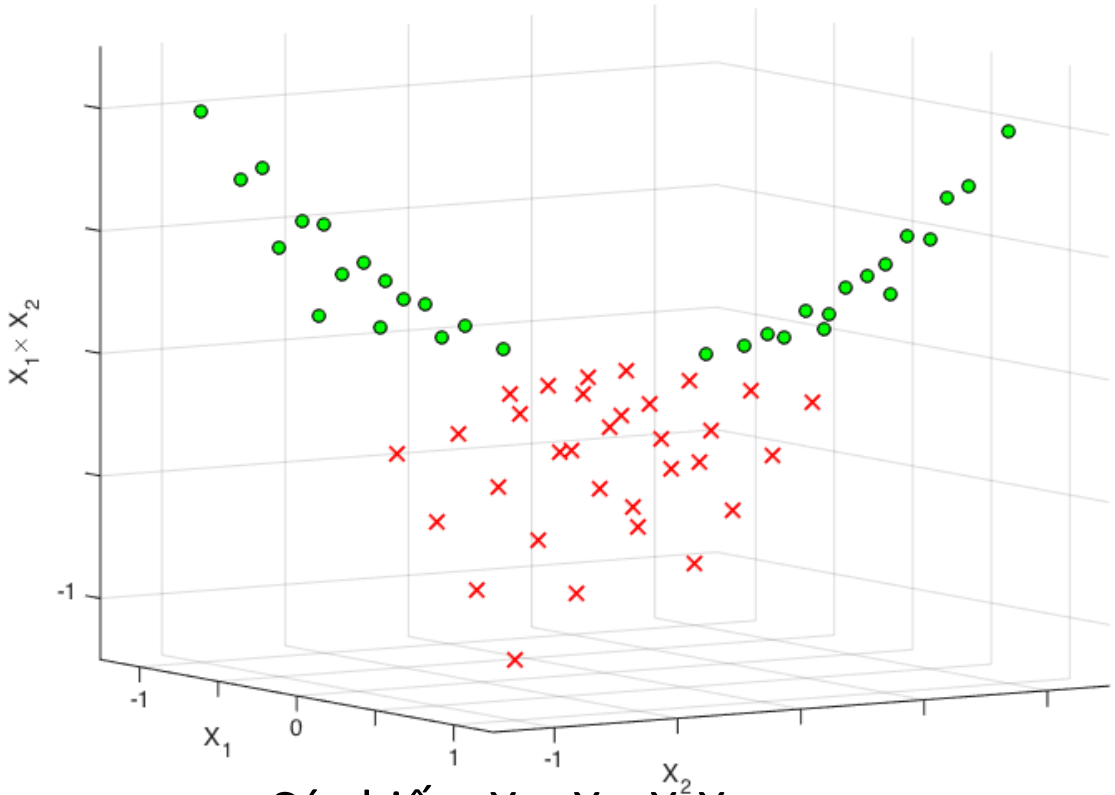


Hình 9.8 , ISL 2013

Mở rộng không gian biến



Các biến: X_1 , X_2



Các biến: X_1 , X_2, X_1X_2

Mở rộng không gian biến

- Hồi quy tuyến tính →Mô hình phi tuyến
  - Tạo ra các biến mà chúng là các hàm của biến đầu vào

- Áp dụng kỹ thuật tương tự này vào bộ phân lớp véc-tơ hỗ trợ
  - Xem xét các hàm đa thức của biến đầu vào

Phân tách phi tuyến

- Giả sử dữ liệu đầu vào của ta có d biến:

$$X = [X_1, X_2, \cdots, X_d]$$

- Mở rộng không gian biến gồm 2d biến:

$$\tilde{X} = [\underbrace{X_1}_{\tilde{X}_1}, \underbrace{(X_1)^2}_{\tilde{X}_2}, \underbrace{X_2}_{\tilde{X}_3}, \underbrace{(X_2)^2}_{\tilde{X}_4}, \cdots, \underbrace{X_d}_{\tilde{X}_{2d-1}}, \underbrace{(X_d)^2}_{\tilde{X}_{2d}}]$$

  - Ranh giới quyết định sẽ là phi tuyến trong không gian biến ban đầu

Phân tách phi tuyến

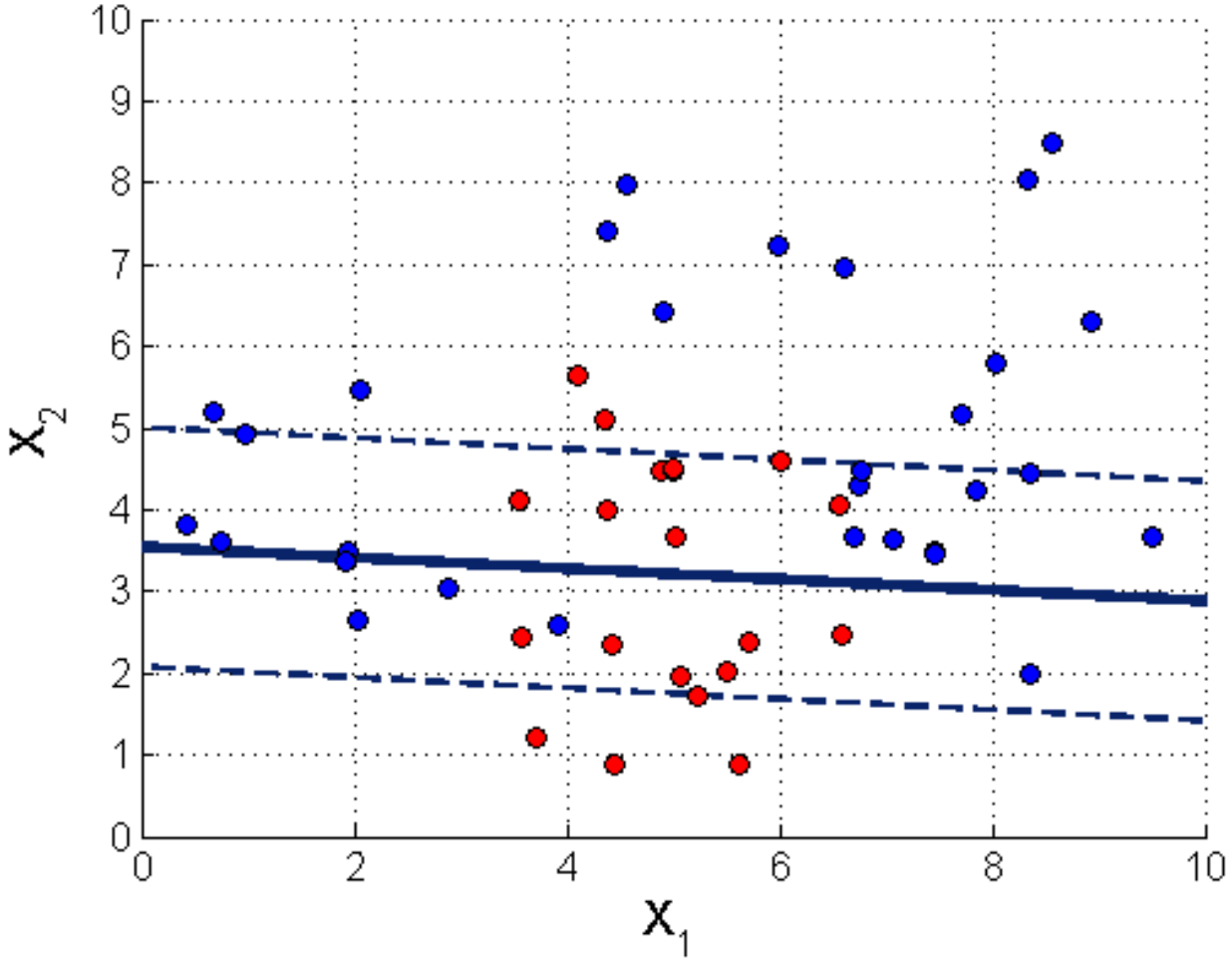
- Dữ liệu được biểu diễn trong không gian mới (đã chuyển đổi) có thể phân lớp tuyến tính:

$$\beta_0 + \beta_1 \tilde{X}_1 + \beta_2 \tilde{X}_2 + \cdots + \beta_{2d-1} \tilde{X}_{2d-1} + \beta_{2d} \tilde{X}_{2d} = 0$$

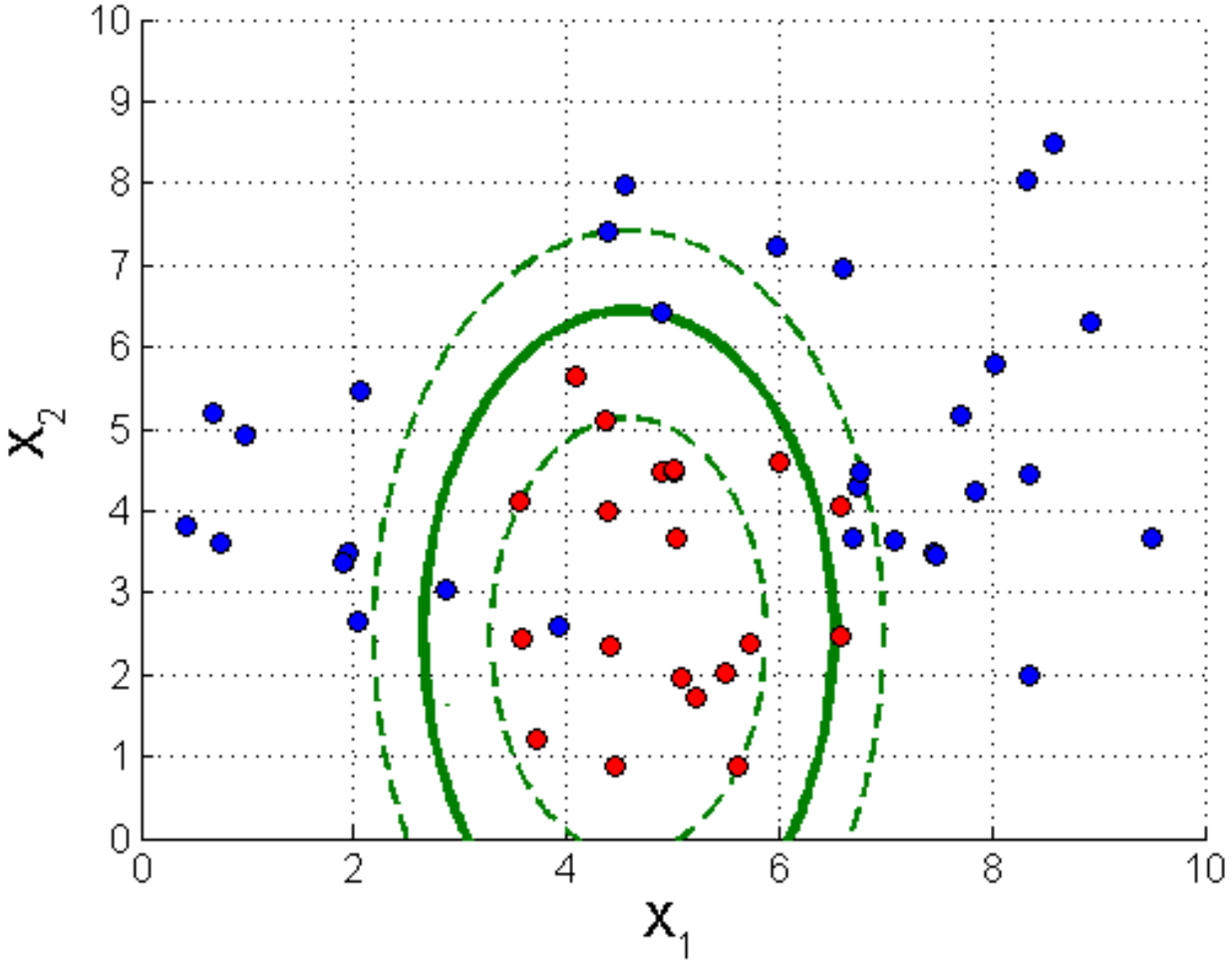
- Ranh giới quyết định trong không gian mở rộng là hình E-líp (ellipse) *trong không gian biến ban đầu*:

$$\beta_0 + \beta_1 X_1 + \beta_2 (X_1)^2 + \cdots + \beta_{2d-1} X_d + \beta_{2d} (X_d)^2 = 0$$

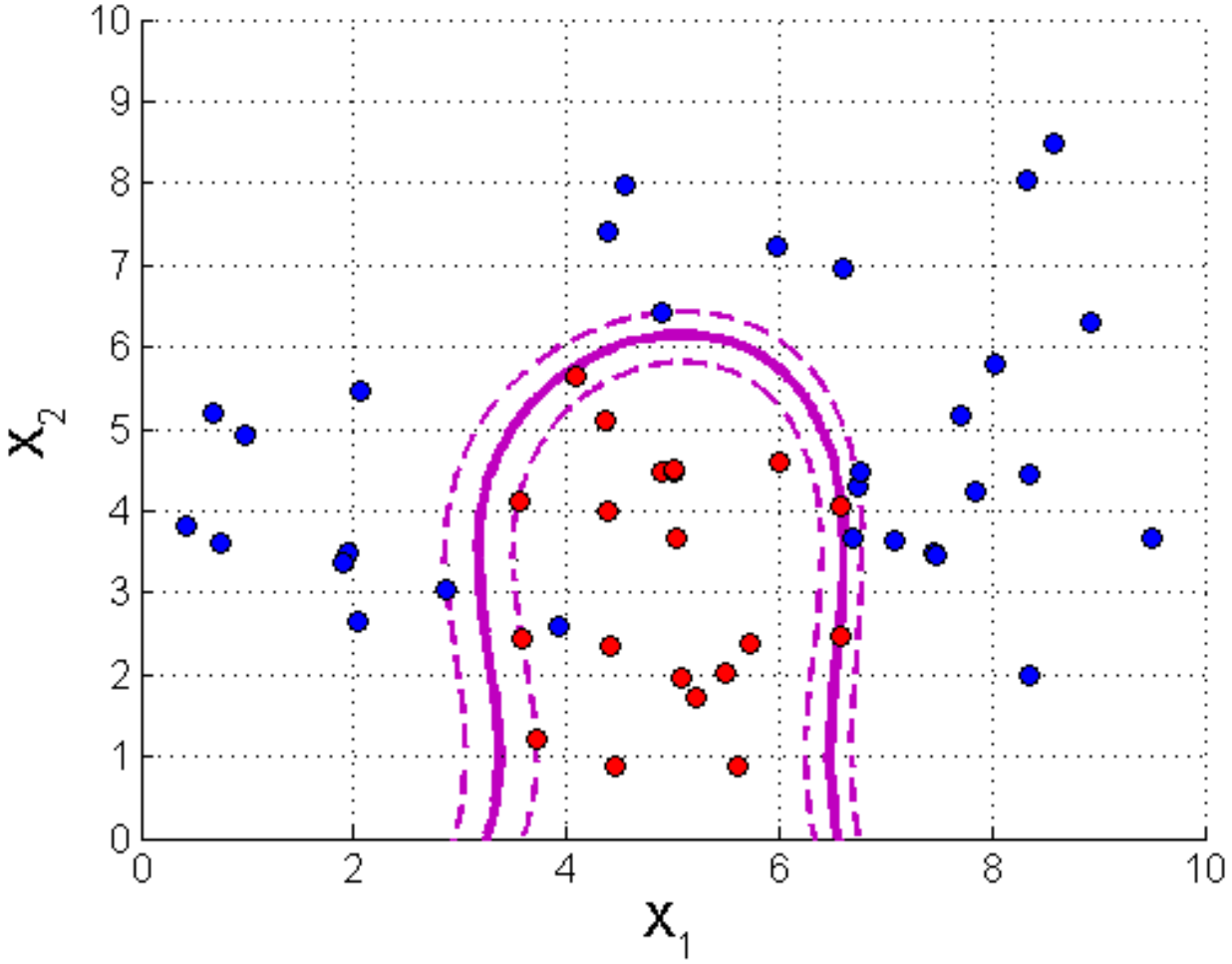
Phân tách phi tuyến



Phân tách phi tuyến



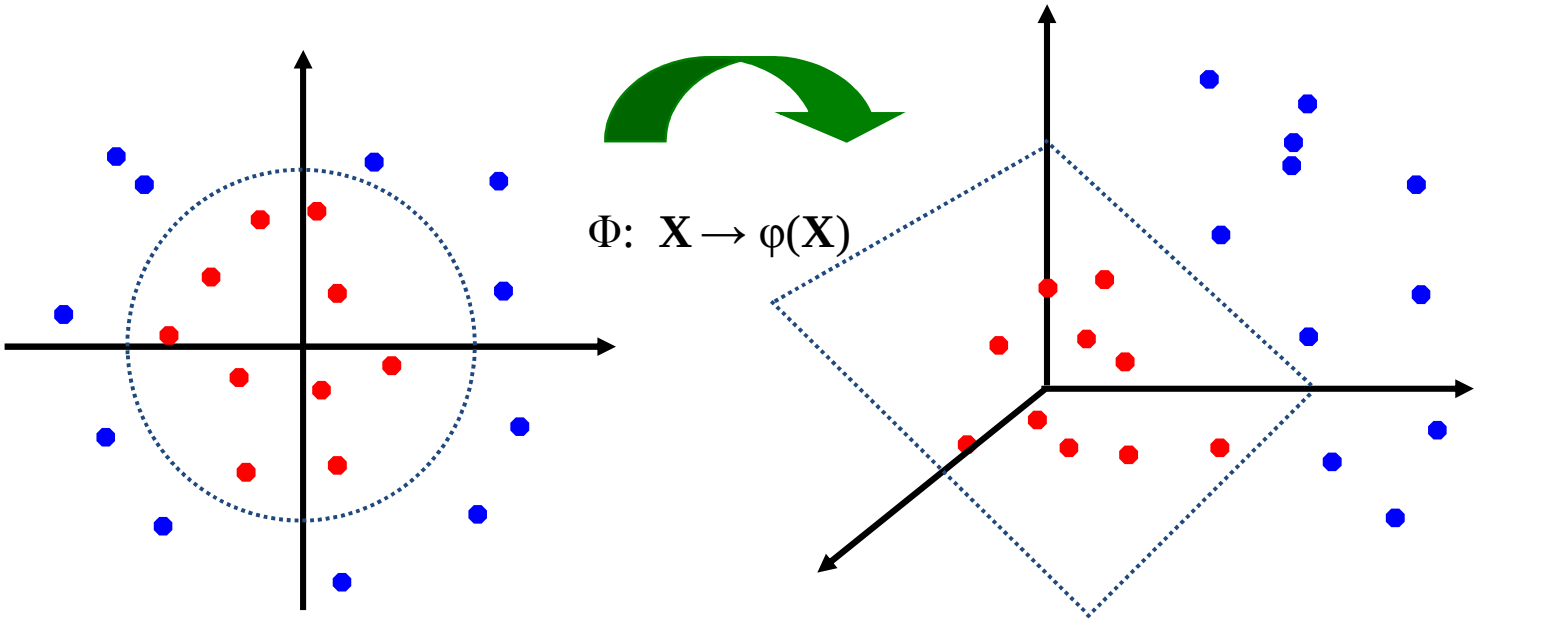
Phân tách phi tuyến



Phân tách phi tuyến

- Thêm đa thức bậc cao cho tập không gian biến mở rộng
  → số lượng biến tăng rất nhanh
  - Bài toán nhiều biến sẽ gặp trở ngại vì thời gian tính toán lâu
  - Ta cần 1 phương pháp hiệu quả để xử lý bài toán nhiều biến

Phân tách phi tuyến



Không gian đầu vào (2D)

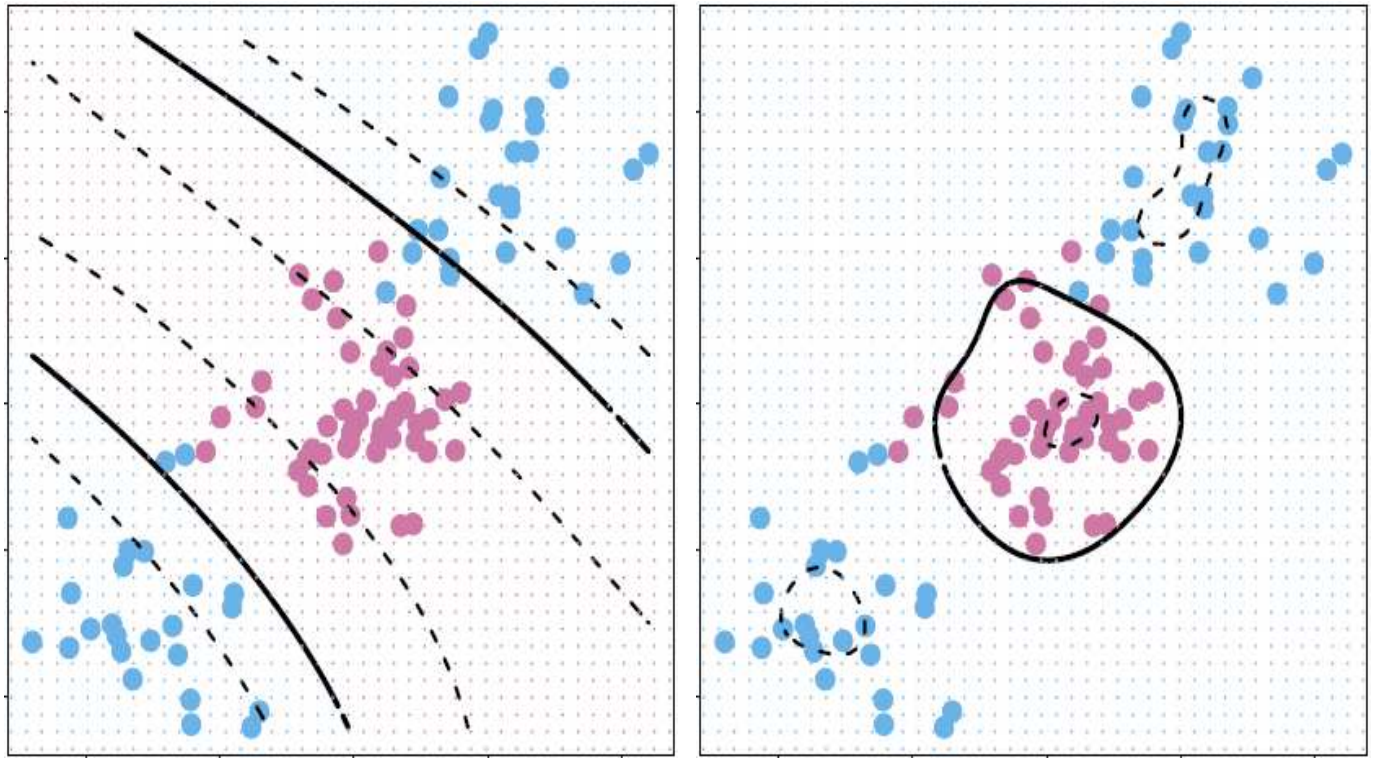
Không gian đặc trưng mới (3D)

Máy véc-tơ hỗ trợ (SVM)

Support Vector Machine: mở rộng để sử dụng các nhân *(kernels)* đạt được *ranh giới quyết định phi tuyến*

- Các hàm nhân ánh xạ dữ liệu vào không gian có số chiều cao hơn

– Áp dụng bộ phân lớp véc-tơ hỗ trợ vào không gian chiều cao với siêu phẳng (tuyến tính) ranh giới quyết định



Máy véc-tơ hỗ trợ

- Việc tính toán trong bộ phân lớp véc-tơ hỗ trợ chỉ yêu cầu xử lý nội tại (inner product) của dữ liệu huấn luyện, không cần thiết phải chuyển đổi không gian trực tiếp

$$f(X) = \beta + \sum_{i \in S} \alpha_i \langle X^{(i)}, X \rangle$$

- Trong phương pháp SVM, ta sử dụng các hàm nhân (kernel functions), được ký hiệu là K

$$f(X) = \beta + \sum_{i \in S} \alpha_i K\left(X^{(i)}, X\right)$$

Ví dụ về ϕ(.) và K(.,.)

- Giả sử $\phi(.)$ được cho bởi

$$\phi\left(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}\right) = (1, \sqrt{2}x_1, \sqrt{2}x_2, x_1^2, x_2^2, \sqrt{2}x_1x_2)$$

- Inner product trong không gian đặc trưng mới là

$$\left\langle \phi\left(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}\right), \phi\left(\begin{bmatrix} y_1 \\ y_2 \end{bmatrix}\right) \right\rangle = (1 + x_1y_1 + x_2y_2)^2$$

- Do đó, nếu ta định nghĩa hàm nhân (kernel function) như dưới đây, ta không cần phải thực hiện $\phi(.)$ một cách tường minh

$$K(\boldsymbol{x}, \boldsymbol{y}) = (1 + x_1y_1 + x_2y_2)^2$$

- Việc sử dụng hàm nhân để tránh thực hiện $\phi(.)$ tường minh được gọi là kernel trick

Máy véc-tơ hỗ trợ

- Tính chất của hàm nhân $K(X, X')$:
  - Không cần chuyển đổi trực tiếp không gian biến

$$K(X, X') = \langle \phi(X), \phi(X') \rangle, \quad \phi \text{ feature mapping}$$

  - Đối xứng: $K(X, X') = K(X', X)$
  - Cho ta tính tương tự của X và X'
    - Nếu X và X' gần nhau thì $K(X, X')$ lớn
    - Nếu X và X' xa nhau từng phần thì $K(X, X')$ nhỏ

Máy véc-tơ hỗ trợ

- Nhân tuyến tính (Linear kernel)

$$K(X, X') = \langle X, X' \rangle$$

- Nhân đa thức bậc p (Polynomial kernel (degree p))

$$K(X, X') = (1 + \langle X, X' \rangle)^p$$

- Nhân Radial (Radial basis kernel)

$$K(X, X') = \exp\left(-\gamma \|X - X'\|^2\right)$$

Máy véc-tơ hỗ trợ

- Tại sao sử dụng nhân thay thế cho xây dựng trực tiếp không gian biến chiều cao?

– Ưu điểm tính toán nhanh hơn

$$\phi : \mathbb{R}^d \rightarrow \mathbb{R}^D, \quad d << D$$

$$K(X, X') = \langle \phi(X), \phi(X') \rangle \quad \text{in } O(d)$$

- Các phương pháp học máy khác sử dụng nhân
  - Vd: kernel PCA

- Ví dụ: nhân đa thức (polynomial kernel), p = 2, d = 2:

$$K(X,Y) = (1 + \langle X, Y \rangle)^2 \qquad X = \begin{bmatrix} X_1 \\ X_2 \end{bmatrix}$$

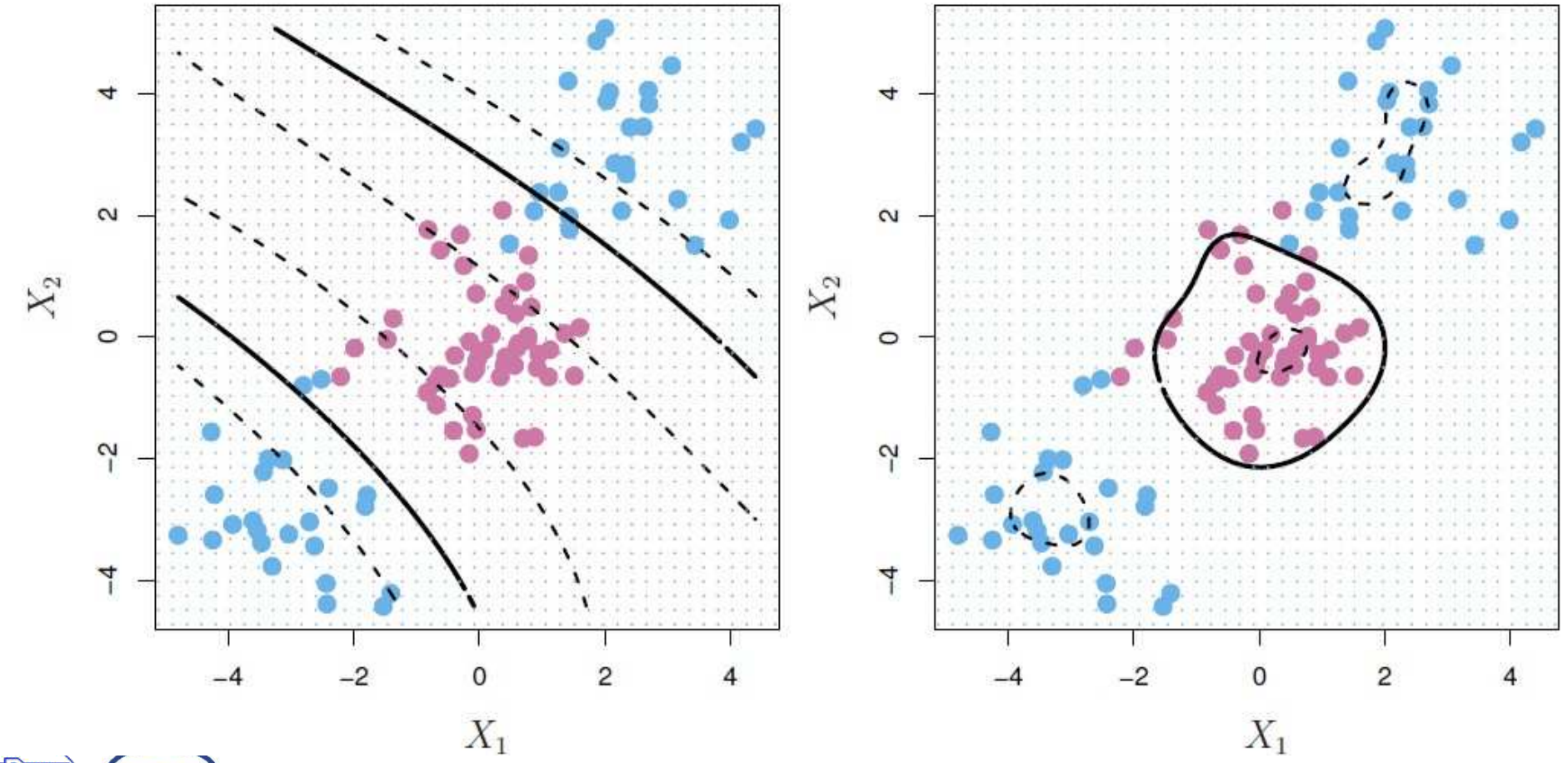
Ta có

$$K(X,Y) = 1 + 2X_1Y_1 + 2X_2Y_2 + X_1^2Y_1^2 + X_2^2Y_2^2 + 2X_1Y_1X_2Y_2$$

$$= \langle \phi(X), \phi(Y) \rangle$$

$$\text{where} \quad \phi(X) = \begin{bmatrix} 1 \\ \sqrt{2}X_1 \\ \sqrt{2}X_2 \\ \sqrt{2}X_1X_2 \\ X_1^2 \\ X_2^2 \end{bmatrix}$$

Máy véc-tơ hỗ trợ



Máy véc-tơ hỗ trợ

- Ưu điểm
  - Điều chỉnh tham số C để tránh overfitting
  - Sử dụng nhân cung cấp độ linh hoạt dưới hình thức ranh giới quyết định
  - Tối ưu hóa hàm lồi – cho lời giải duy nhất

- Nhược điểm
  - Phải thử nhiều siêu tham số (vd: C, kernel function)
    - Đạt hiệu suất kém nếu chọn sai
  - Phải đưa được về bài toán phân lớp nhị phân
  - Khó diễn giải


Câu hỏi?

SVM với 3+ lớp

- SVMs được thiết kế cho phân lớp nhị phân
  - Siêu phẳng tách phân dữ liệu thành 2 lớp
- Làm sao để xử lý dữ liệu khi số lớp nhiều hơn 2?
- Có 2 cách tiếp cận thông dụng:
  1. One–versus–one
  2. One–versus–all

SVM với 3+ lớp

- Phân lớp *One-versus-one*
  - Xây dựng SVM cho từng cặp
  - Với K lớp sẽ yêu cầu huấn luyện $\frac{K(K-1)}{2}$ SVMs
  - Để phân lớp đối tượng mới, áp dụng tất cả $\frac{K(K-1)}{2}$SVMs cho từng mẫu – chọn lớp có tần suất nhiều nhất trong từng cặp để lấy giá trị phân lớp cuối cùng
  - Nhược điểm: thời gian tính toán lâu khi giá trị của K lớn

SVM với 3+ lớp

- Phân lớp *One–versus–all*
  - Xấp xỉ K SVMs, ở đó lớp *K* biểu thị một phân lớp, và các lớp K−1 còn lại được gộp vào thành lớp thứ 2
  - Khoảng cách đến siêu phẳng tách sẽ đại diện cho độ tin cậy của phân lớp
  - Với đối tượng mới, chọn lớp có "độ tin cậy cao nhất" để dự đoán

Câu hỏi?