**Chương 8. Support Vector Machines (SVM - máy véc tơ hỗ trợ)**

**Outline**

[**I. CÁC CÂU HỎI CHƯƠNG 8 CÓ TRONG CÁC ĐỀ THI 2**](#_43e9l3bvebs3)

[1. Fin\_201.pdf\_HKI\_(2020-2021) 2](#_ghd6d3hz7wos)

[2. Fin\_202.pdf\_HKII\_(2020-2021) 2](#_3tl6br8wrpcf)

[3. Fin\_12\_2021.pdf\_HKI\_(2021-2022) 2](#_j9gq0wo6z7xt)

[4. Fin\_12\_2021.pdf\_HKII\_(2021-2022) 2](#_lxbo7ly6sudw)

[**II. CÂU HỎI CHI TIẾT TRONG ĐỀ 2**](#_redpzuq8njsd)

[1. Fin\_201.pdf\_HKI\_(2020-2021) 2](#_3xx3b76v0id3)

[2. Fin\_202.pdf\_HKII\_(2020-2021) 2](#_7ojfifuqfd31)

[3. Fin\_12\_2021.pdf\_HKI\_(2021-2022) 3](#_s1im3gd9yoi)

[4. Fin\_12\_2021.pdf\_HKII\_(2021-2022) 3](#_xr5cagvendmm)

[**II. GIẢI ĐỀ …To do … 3**](#_q2m3hf5vaw7i)

[**1. Fin\_201.pdf\_HKI\_(2020-2021) 3**](#_ze1dwaoantgw)

[Câu 1. (1 điểm) 3](#_mjksxwfmht7r)

[Đáp Án câu 1.1 4](#_mgi0mryne7k)

[Đáp án câu 1.3 5](#_uj0to419b7bg)

[Câu 5 (1.5 điểm) 5](#_3vudxdbjx1ij)

[Đáp án câu c 5](#_h5rjik8q1r7q)

[Đáp án câu d 6](#_uthpmg8m0lcx)

[**2. Fin\_202.pdf\_HKII\_(2020-2021) 7**](#_9drmw6p2olse)

[Câu 5. (1.75 điểm) 7](#_8rth6t247ebm)

[Đáp Án 7](#_byi9ohq64k81)

[**3. Fin\_12\_2021.pdf\_HKI\_(2021-2022) 9**](#_y8kr78wvet2w)

[**4. Fin\_12\_2021.pdf\_HKII\_(2021-2022) 9**](#_u45jliqsqi1u)

## **I. CÁC CÂU HỎI CHƯƠNG 8 CÓ TRONG CÁC ĐỀ THI**

##### [Fin\_201.pdf\_HKI\_(2020-2021)](https://docs.google.com/document/d/1O30YVMcYqIh3FvXyK30TFXqQB4cgPsMO/edit#heading=h.k859l5ku3md)

* + Câu 1.
    1. Phần 1.1, **- Tâm**
    2. Phần 1.3 **- Tâm**
  + Câu 5 **- Tâm (c, d)**

##### [Fin\_202.pdf\_HKII\_(2020-2021)](https://docs.google.com/document/d/1DkbvgkUJOOQcx9eNk-DsMT5MwyhNMvLE/edit#)

* + Câu 5 **- Tâm (a, b)**

##### [Fin\_12\_2021.pdf\_HKI\_(2021-2022)](https://docs.google.com/document/d/1FBqWzRhwBudjJDBuK2_z72pM98EndxA2/edit#heading=h.t37itj2wog4b)

* + Câu 5 **- Nghĩa**

##### [Fin\_12\_2021.pdf\_HKII\_(2021-2022)](https://docs.google.com/document/d/1BLSqsCee6LnqI5rh2J_ZeN3whI-TOJl-VinGDjto97A/edit#)

* + Câu 5 **(*Trùng* với câu 5.** [**Fin\_202.pdf\_HKII\_(2020-2021)**](https://docs.google.com/document/d/1DkbvgkUJOOQcx9eNk-DsMT5MwyhNMvLE/edit#)**)**

## **II. CÂU HỎI CHI TIẾT TRONG ĐỀ**

#### **1.** [**Fin\_201.pdf\_HKI\_(2020-2021)**](https://docs.google.com/document/d/1O30YVMcYqIh3FvXyK30TFXqQB4cgPsMO/edit#heading=h.k859l5ku3md)

**Câu 1. (1 điểm)** Trả lời các câu hỏi sau đây, mỗi câu 0.25 điểm

**1.1** Cho một tập dữ liệu không khả tách một cách tuyến tính, bộ phân lớp nhị phân nào sau đây có thể tách tập dữ liệu ra làm hai lớp một cách đúng đắn:

A. SVM tuyến tính B. Perceptron sử dụng perceptron training rule

C. SVM với RBF kernel D. Các ba bộ phân lớp trên đều không thể.  
**→ Chương 8**

**1.3** (Đúng/Sai) SVM và ANN có thể được dùng cho cả hai bài toán phân lớp và hồi

quy.

**→ Chương 8**

**Câu 5 (1.5 điểm)**

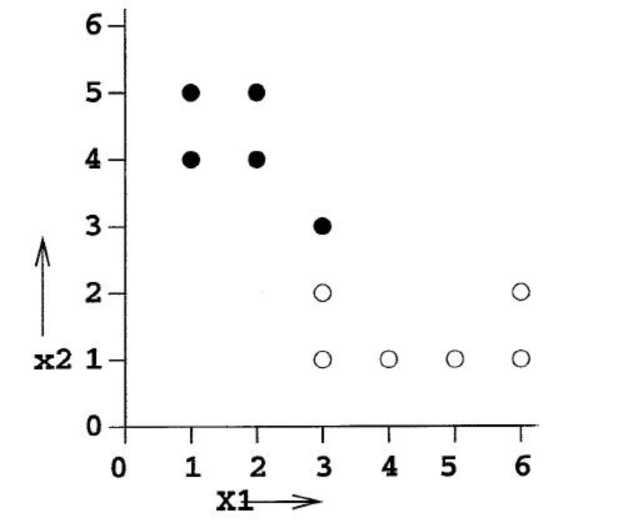
1. Giải thích các thuật ngữ khoảng biên (margin), siêu phẳng tách với khoảng biên tối đa (maximum margin hyperplane), siêu phẳng hỗ trợ (support hyperplane), véc tơ hỗ trợ (support vector). (*1 điểm*)
2. Giải thích ý nghĩa của các biến bù *xi* và thông số *C* trong máy vec tơ hỗ trợ với khoảng biên mềm (soft margin) (*0.5 điểm*)  
   **→ Chương 8**

#### **2.** [**Fin\_202.pdf\_HKII\_(2020-2021)**](https://docs.google.com/document/d/1DkbvgkUJOOQcx9eNk-DsMT5MwyhNMvLE/edit#)

**Câu 5. (1.75 điểm)**

1. Cho một tập dữ liệu mà mỗi điểm dữ liệu có hai thuộc tính (x1 and x2) và một nhãn lớp như trong hình vẽ sau. Các điểm thuộc lớp dương ký hiệu bằng dấu chấm đậm và các điểm thuộc lớp âm ký hiệu bằng những hình tròn nhỏ.

Giả sử chúng ta đang dùng SVM tuyến tính để phân lớp. Hãy vẽ 3 đường thẳng biểu diễn siêu phẳng tách có khoảng biên cực đại và hai siêu phẳng hỗ trợ. Tính khoảng biên cực đại. (*1 điểm*)



b. Hãy nêu công dụng của hàm kernel đối với máy véc tơ hỗ trợ trong trường hợp dữ liệu không khả tách một cách tuyến tính. (*0.25 điểm*)

c. Giải thích ý nghĩa của các biến bù *Ặ* và thông số *C* trong máy vec tơ hỗ trợ với khoảng biên mềm (soft margin) (*0.5 điểm*)

**→ Chương 8**

#### **3.** [**Fin\_12\_2021.pdf\_HKI\_(2021-2022)**](https://docs.google.com/document/d/1FBqWzRhwBudjJDBuK2_z72pM98EndxA2/edit#heading=h.t37itj2wog4b)

**Câu 5. (0.75 điểm)**

**b.** Giải thích ý nghĩa của các biến bù *Ặ* và thông số *C* trong máy vec tơ hỗ trợ với khoảng biên mềm (soft margin) (*0.5 điểm*)  
**→ Chương 8**

#### **4.** [**Fin\_12\_2021.pdf\_HKII\_(2021-2022)**](https://docs.google.com/document/d/1BLSqsCee6LnqI5rh2J_ZeN3whI-TOJl-VinGDjto97A/edit#)

Câu 5 **(*Trùng* với câu 5.** [**Fin\_202.pdf\_HKII\_(2020-2021)**](https://docs.google.com/document/d/1DkbvgkUJOOQcx9eNk-DsMT5MwyhNMvLE/edit#)**)**

## **II. GIẢI ĐỀ** …To do …

#### **1.** [**Fin\_201.pdf\_HKI\_(2020-2021)**](https://docs.google.com/document/d/1O30YVMcYqIh3FvXyK30TFXqQB4cgPsMO/edit#heading=h.k859l5ku3md)

##### **Câu 1. (1 điểm)**

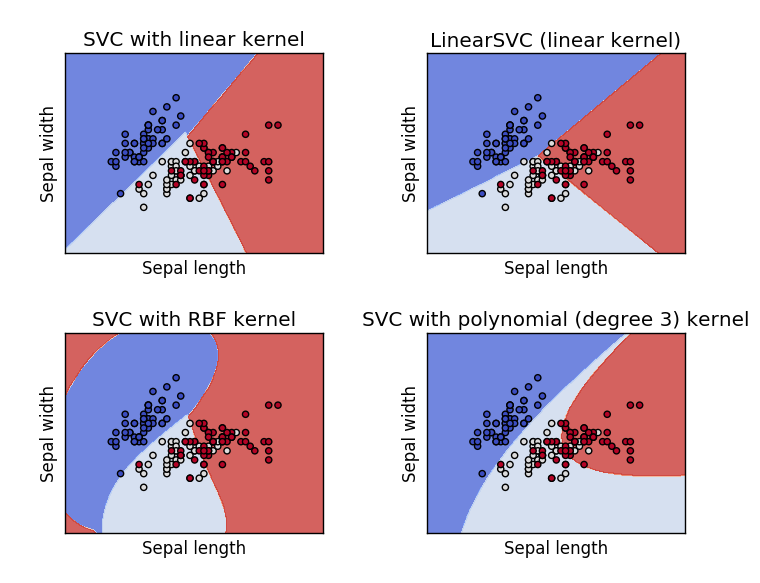
Trả lời các câu hỏi sau đây, mỗi câu 0.25 điểm

**1.1** Cho một tập dữ liệu không khả tách một cách tuyến tính, bộ phân lớp nhị phân nào sau đây có thể tách tập dữ liệu ra làm hai lớp một cách đúng đắn:

SVM với RBF kernel

###### **Đáp Án câu 1.1**

**[Tâm]**

****

**C. SVM với RBF kernel.**

Trong bài toán phân lớp, **khả tách tuyến tính (linearly separable) ám chỉ trạng thái khi tập dữ liệu của hai lớp có thể được tách ra bằng một đường thẳng, một siêu phẳng hoặc một siêu mặt phẳng tuyến tính**. Trong trường hợp này, một bộ phân lớp tuyến tính như SVM tuyến tính hay perceptron có thể được sử dụng để tìm ra đường phân chia chính xác giữa các lớp.

Tuy nhiên, khi tập dữ liệu **không khả tách tuyến tính (linearly inseparable), không tồn tại một đường thẳng, một siêu phẳng hoặc một siêu mặt phẳng tuyến tính có thể tách các điểm dữ liệu của hai lớp**. Trong trường hợp này, cần sử dụng các bộ phân lớp phi tuyến (nonlinear classifiers) hoặc các phương pháp chuyển đổi dữ liệu sang không gian cao hơn để tìm một đường phân chia không tuyến tính.

Trong trường hợp tập dữ liệu không khả tách tuyến tính, SVM với RBF kernel (Radial Basis Function) thường cho phép phân lớp một cách chính xác hơn so với SVM tuyến tính hoặc perceptron. **Kernel RBF cho phép phân lớp các dữ liệu không khả tách tuyến tính bằng cách chuyển đổi chúng vào không gian cao hơn, nơi chúng có thể được phân lớp một cách tuyến tính**.

Perceptron sử dụng perceptron training rule không phải là một phương pháp hiệu quả để xử lý tập dữ liệu không khả tách tuyến tính. SVM tuyến tính cũng không thể tách một tập dữ liệu không khả tách tuyến tính một cách chính xác.

Vì vậy, đáp án đúng là C. SVM với RBF kernel.

**1.3** (Đúng/Sai)

SVM và ANN có thể được dùng cho cả hai bài toán phân lớp và hồi quy.

###### **Đáp án câu 1.3**

**[Tâm]**

Đúng. Cả SVM (Support Vector Machines) và ANN (Artificial Neural Networks) có thể được sử dụng cho cả bài toán phân lớp và hồi quy.

Mục tiêu chính trong bài toán hồi quy là tìm một mô hình tốt nhất để ước lượng giá trị đầu ra dựa trên các đặc trưng đầu vào.

Bài toán phân lớp (classification) mục tiêu là dự đoán lớp hoặc nhãn (label) của một mẫu dữ liệu dựa trên các đặc trưng (features) của nó. Mỗi mẫu dữ liệu thường được biểu diễn bằng một vectơ đặc trưng, và mục tiêu là phân loại mẫu dữ liệu vào một trong các lớp đã được xác định trước.

[Nam] Đây là lí do mạng nơ ron có thể xấp xỉ các hàm số tốt và không cần biết trước rằng hàm số đó có dạng gì, có là đa thức hay không. Và kết quả của quá trình huấn luyện sẽ là một quan hệ được biểu diễn bằng một hàm phức tạp thể hiện bằng các tham số của mạng nơ ron. Trong thực tế, các đơn vị tính toán cơ bản nhất trong mạng nơ ron được lấy cảm hứng từ các thuật toán học máy truyền thống như SVM hoặc hồi quy logistic.

=> Dong y phuong an tra loi la True

##### **Câu 5 (1.5 điểm)**

1. Giải thích các thuật ngữ khoảng biên (margin), siêu phẳng tách với khoảng biên tối đa (maximum margin hyperplane), siêu phẳng hỗ trợ (support hyperplane), véc tơ hỗ trợ (support vector). (*1 điểm*)

###### **Đáp án câu c**

**[Tâm]**

Trong SVM (Support Vector Machines), các thuật ngữ sau đây được sử dụng để mô tả các khái niệm quan trọng:

* ***Khoảng biên (Margin):*** Là khoảng cách giữa siêu phẳng tách và các mẫu dữ liệu gần nhất của các lớp khác nhau. Khoảng biên càng lớn, mô hình SVM càng tổng quát hóa tốt hơn và có khả năng phân loại tốt hơn các mẫu dữ liệu mới. Khoảng cách giữa support vector và siêu phẳng tách
* ***Siêu phẳng tách với khoảng biên tối đa (Maximum Margin Hyperplane):*** Là siêu phẳng tách lớn nhất mà có thể tách các mẫu dữ liệu của các lớp khác nhau. Siêu phẳng này được tìm kiếm trong quá trình huấn luyện SVM và được tối ưu hóa để có khoảng biên lớn nhất.
* ***Siêu phẳng hỗ trợ (Support Hyperplane):*** Là siêu phẳng tách gần nhất với các mẫu dữ liệu của các lớp khác nhau. Siêu phẳng hỗ trợ chứa các điểm dữ liệu gần nhất đến đường phân chia, và nó quyết định vị trí của đường phân chia và khoảng biên. Là 2 siêu phẳng cách đều siêu phẳng tách với khoảng cách là margin
* ***Véc tơ hỗ trợ (Support Vector):*** Là các điểm dữ liệu nằm trên siêu phẳng hỗ trợ. Các điểm dữ liệu này là quan trọng nhất trong việc xác định siêu phẳng tách và khoảng biên. Véc tơ hỗ trợ chỉ ra các mẫu dữ liệu quan trọng nhất để xác định đường phân chia.

Trong SVM, mục tiêu là tìm một siêu phẳng tách tối ưu có khoảng biên lớn nhất và tối thiểu hóa sai số phân loại. Siêu phẳng tách và các véc tơ hỗ trợ chính là thành phần quan trọng để xác định đường phân chia và hỗ trợ quan trọng nhất cho quá trình phân loại.

1. Giải thích ý nghĩa của các biến bù *xi* và thông số *C* trong máy vec tơ hỗ trợ với khoảng biên mềm (soft margin) (*0.5 điểm*)

###### **Đáp án câu d**

**[Tâm]**

Biến bù xi (Slack Variables): Biến bù xi được sử dụng để đo lường mức độ vi phạm của các điểm dữ liệu đến khoảng biên. Mỗi điểm dữ liệu có thể có một giá trị bù xi tương ứng, cho biết mức độ vi phạm của nó đối với siêu phẳng tách.

* Nếu giá trị bù xi là 0, điểm dữ liệu nằm chính xác trên siêu phẳng tách.
* Nếu giá trị bù xi là dương và nhỏ hơn 1, điểm dữ liệu nằm trong vùng an toàn và gần siêu phẳng tách.
* Nếu giá trị bù xi là lớn hơn 1, điểm dữ liệu vi phạm quy tắc phân loại và nằm ở phía sai lớp.

Mục tiêu của quá trình tối ưu hóa là tìm một mô hình SVM tối ưu với sự phạm vi vi phạm của các điểm dữ liệu được kiểm soát thông qua việc điều chỉnh các giá trị bù xi.

Thông số C: Thông số C là một tham số quan trọng trong SVM với khoảng biên mềm. Nó kiểm soát mức độ mà mô hình cho phép vi phạm các điểm dữ liệu.

* Khi C càng lớn, mô hình SVM càng cứng và không chấp nhận sự phạm vi vi phạm cao hơn. Điều này có nghĩa là mô hình sẽ cố gắng tìm kiếm một siêu phẳng tách chặt chẽ và giảm thiểu vi phạm.
* Khi C nhỏ, mô hình SVM trở nên linh hoạt hơn và cho phép sự phạm vi vi phạm lớn hơn. Điều này có thể dẫn đến sự chấp nhận một số điểm dữ liệu vi phạm để tạo ra một đường phân chia tổng quát hơn và phù hợp với dữ liệu huấn luyện.

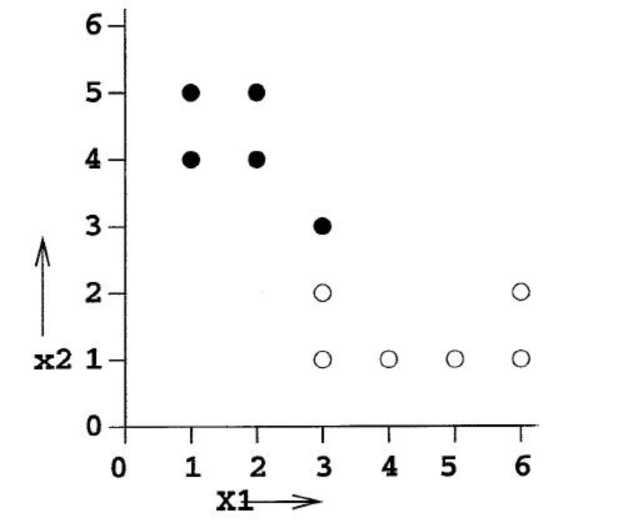
Lựa chọn giá trị C phù hợp là một quá trình cân nhắc giữa việc tối đa hóa sự phân loại chính xác và vi phạm dữ liệu. Một giá trị C lớn sẽ đánh đổi sự linh hoạt để giảm thiểu vi phạm, trong khi một giá trị C nhỏ sẽ tạo ra một siêu phẳng tách mềm hơn và cho phép vi phạm đến một mức độ nhất định.

#### **2.** [**Fin\_202.pdf\_HKII\_(2020-2021)**](https://docs.google.com/document/d/1DkbvgkUJOOQcx9eNk-DsMT5MwyhNMvLE/edit#)

##### **Câu 5. (1.75 điểm)**

1. Cho một tập dữ liệu mà mỗi điểm dữ liệu có hai thuộc tính (x1 and x2) và một nhãn lớp như trong hình vẽ sau. Các điểm thuộc lớp dương ký hiệu bằng dấu chấm đậm và các điểm thuộc lớp âm ký hiệu bằng những hình tròn nhỏ.

Giả sử chúng ta đang dùng SVM tuyến tính để phân lớp. Hãy vẽ 3 đường thẳng biểu diễn siêu phẳng tách có khoảng biên cực đại và hai siêu phẳng hỗ trợ. Tính khoảng biên cực đại. (*1 điểm*)



###### **Đáp Án**

Với bài toán phân loại SVM tuyến tính, chúng ta cần tìm một siêu phẳng tách để phân chia hai lớp dữ liệu. Siêu phẳng này sẽ có dạng w1\*x1 + w2\*x2 + b = 0

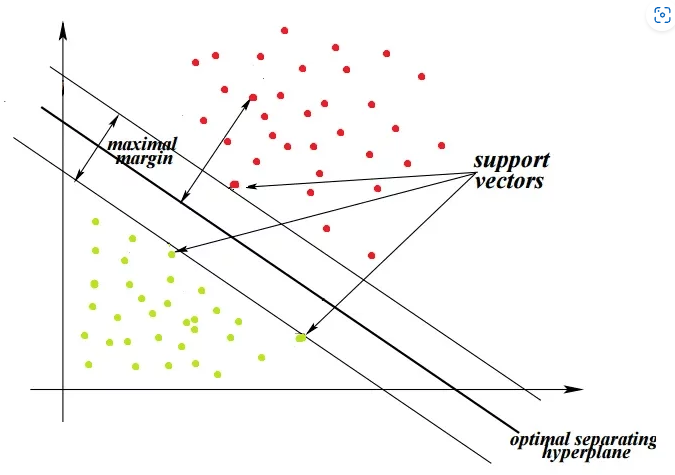
w1, w2 là tham số b là hệ số điều chỉnh

**Tìm đường biên có lề lớn nhất**

Siêu phẳng tách: w1\*x1 + w2\*x2 + b = 0 (Separating hyperplane)

Siêu phẳng hỗ trợ bên trái: w1\*x1 + w2\*x2 + b = -1 (Left support hyperplane)

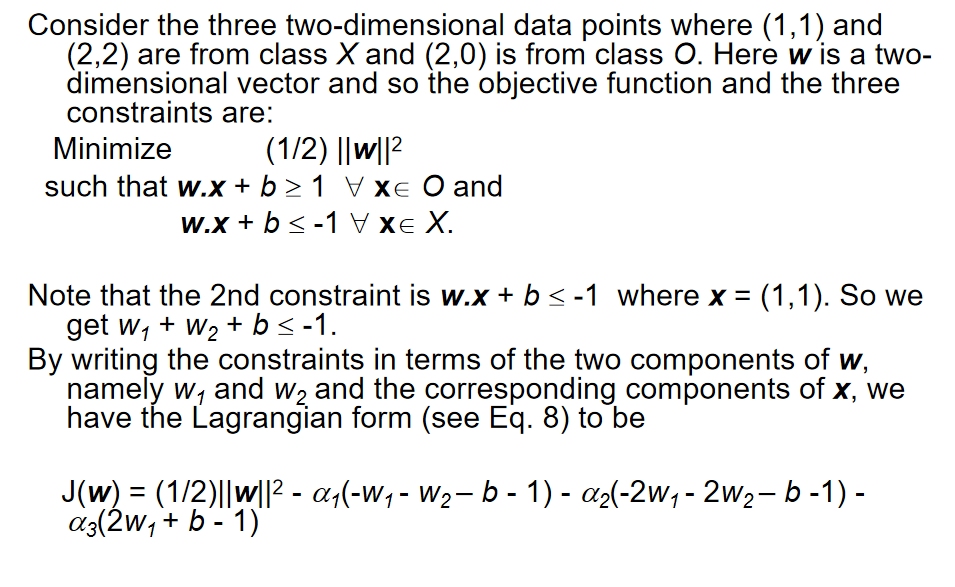
Siêu phẳng hỗ trợ bên phải: w1\*x1 + w2\*x2 + b = 1 (Right support hyperplane)



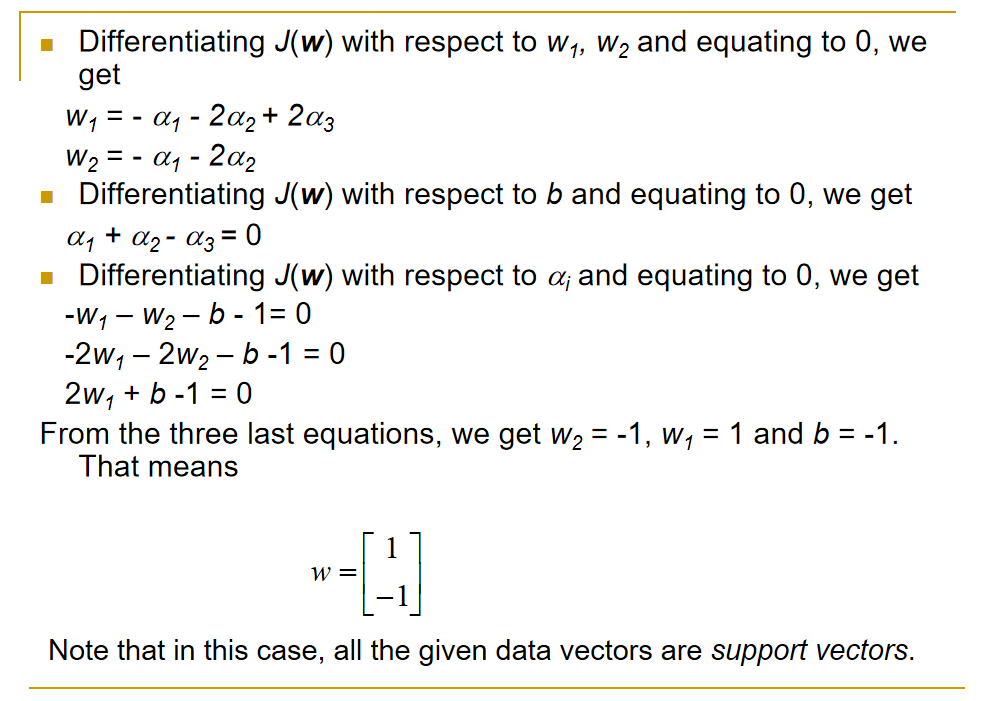
ảnh minh họa

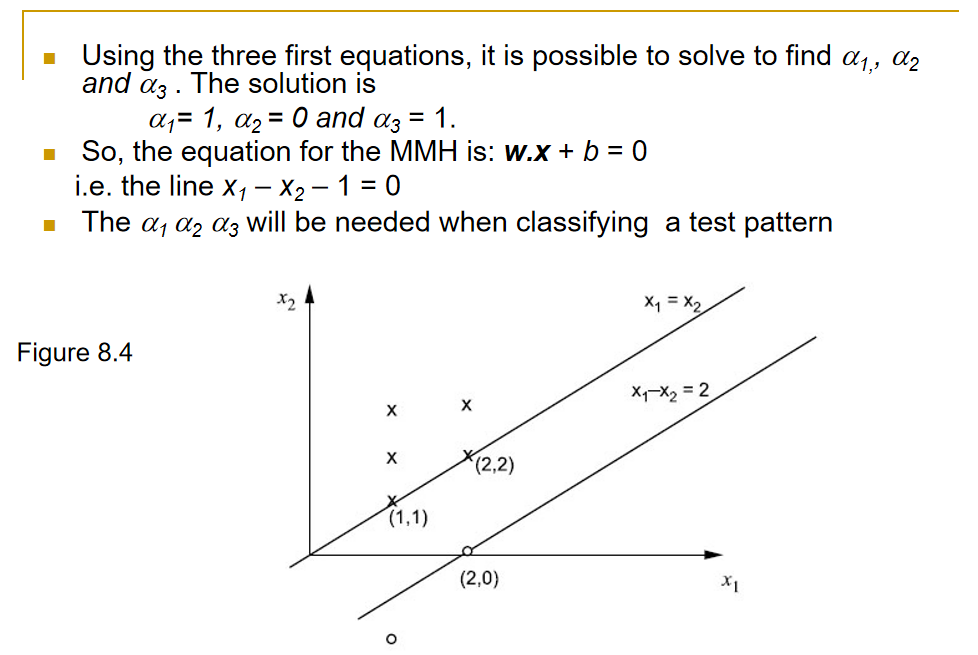
Ở bài này. Ta cũng có thể làm giống như slide 25 chương 8. Bài tập ví dụ như sau:

Làm giống như bài này, để tìm ra phương trình của biên quyết định. Cũng như độ lớn của Margin - khoảng biên.



Ở bước này, lấy đạo hàm riêng của từng biến một, sau đó lấy nghiệm từ hệ.





Sau khi tìm được được W ta sẽ có phương trình của biên quyết định

Sau đó sẽ tìm được margin - độ rộng của khoảng biên.

Cách tính dùng tối ưu cho nhiều điểm (GPT):

Để tính toán trực tiếp các giá trị w và b mà không sử dụng thư viện, chúng ta cần giải bài toán tối ưu hóa SVM bằng phương pháp Lagrange và các điều kiện KKT. Dưới đây là quy trình để tính toán w và b:

Xây dựng hàm Lagrange:

Xây dựng hàm Lagrange L(w, b, alpha) bằng cách thay thế hàm mục tiêu và các ràng buộc vào bài toán tối ưu hóa SVM. Hàm Lagrange có dạng:

L(w, b, alpha) = 0.5 \* ||w||^2 - sum(alpha\_i \* (y\_i \* (w\*x\_i + b) - 1))

Trong đó, alpha\_i là các biến Lagrange và y\_i là nhãn của từng điểm dữ liệu.

Tính đạo hàm của hàm Lagrange:

Tính đạo hàm riêng của hàm Lagrange theo w, b và alpha\_i và đặt bằng 0 để tìm điểm tối ưu:

dL/dw = w - sum(alpha\_i \* y\_i \* x\_i) = 0

dL/db = sum(alpha\_i \* y\_i) = 0

Sử dụng các điều kiện KKT:

Áp dụng các điều kiện KKT để xác định các giá trị của alpha\_i:

alpha\_i \* (y\_i \* (w\*x\_i + b) - 1) = 0

alpha\_i >= 0

y\_i \* (w\*x\_i + b) - 1 >= 0

Tìm giá trị của alpha\_i:

Giải hệ các phương trình KKT và bất đẳng thức để tìm các giá trị của alpha\_i. Chúng ta có thể sử dụng các phương pháp như phương pháp gradient descent hoặc phương pháp Lagrange dual để tìm các giá trị này.

Tính toán w và b:

Sau khi có các giá trị alpha\_i, tính toán w và b bằng công thức sau:

w = sum(alpha\_i \* y\_i \* x\_i)

b = y\_i - sum(alpha\_i \* y\_i \* (x\_i \* x\_j))

Trong đó, x\_j là một điểm dữ liệu trên biên (có giá trị alpha\_i > 0 và 0 < y\_i \* (w \* x\_j + b) < 1).

b. Hãy nêu công dụng của hàm kernel đối với máy véc tơ hỗ trợ trong trường hợp dữ liệu không khả tách một cách tuyến tính. (*0.25 điểm*)

Hàm kernel là một phương pháp được sử dụng trong máy học để giải quyết vấn đề phân loại dữ liệu không khả tách một cách tuyến tính. Trong SVM, hàm kernel được sử dụng để ánh xạ dữ liệu từ không gian dữ liệu ban đầu sang một không gian mới nơi dữ liệu có thể được phân loại tốt hơn bằng một siêu phẳng tách tuyến tính.

Các hàm kernel thường được sử dụng trong SVM bao gồm:

Hàm đa thức (Polynomial Kernel): Đây là hàm kernel đơn giản nhất và được sử dụng phổ biến nhất trong SVM. Hàm đa thức ánh xạ dữ liệu từ không gian ban đầu sang một không gian mới bằng cách sử dụng một đa thức bậc cao hơn để giúp phân loại dữ liệu tốt hơn.

Hàm Gauss (Gaussian Kernel): Đây là hàm kernel phổ biến khác được sử dụng để xử lý dữ liệu không khả tách một cách tuyến tính trong SVM. Hàm Gauss ánh xạ dữ liệu từ không gian ban đầu sang một không gian mới bằng cách sử dụng một phân phối Gauss để tính toán độ tương đồng giữa các điểm dữ liệu. Hàm này cho phép phân loại các điểm dữ liệu theo các siêu phẳng không tuyến tính.

Hàm Sigmoid (Sigmoid Kernel): Hàm kernel này ánh xạ dữ liệu từ không gian ban đầu sang một không gian mới bằng cách sử dụng một hàm sigmoid tuyến tính. Hàm này thường được sử dụng trong bài toán phân loại nhị phân.

**Các hàm kernel này giúp máy véc tơ hỗ trợ có thể phân loại dữ liệu không khả tách một cách tuyến tính bằng cách ánh xạ dữ liệu từ không gian ban đầu sang một không gian mới có số chiều lớn hơn. Khi dữ liệu được ánh xạ sang không gian mới, nó có thể được phân loại tốt hơn bằng một siêu phẳng tách tuyến tính. Hàm kernel không chỉ giúp giải quyết vấn đề phân loại dữ liệu không khả tách một cách tuyến tính mà còn giúp giảm chi phí tính toán và không gian lưu trữ dữ liệu khi áp dụng SVM cho các tập dữ liệu lớn.**

c. Giải thích ý nghĩa của các biến bù *Epsilon* và thông số *C* trong máy vec tơ hỗ trợ với khoảng biên mềm (soft margin)

 là một siêu tham số, cho phép có bao nhiêu điểm dữ liệu được phép vi phạm ràng buộc khoảng biên.

C is a parameter chosen by the user that controls the tradeoff between the margin and the misclassification errors. A larger C means that a higher penalty to misclassification errors is assigned.

C điều tiết sự đánh đổi giữa độ rộng của khoảng biên và số lỗi do phân lớp sai. C Lớn Có nghĩa là penalty cao hơn cho lỗi phân lớp.

Đôi với Epsilon:

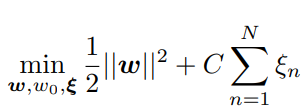
Nếu dữ liệu không khả tách tuyến tính, sẽ không có nghiệm khả thi cho ràng buộc:

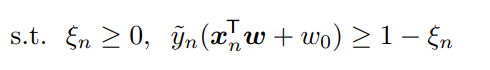


Vi vậy, Chúng ta giới thiệu biến dư - Slack Variables -

Sau đó thay thế ràng buộc cứng  bằng ràng buộc độ lớn khoảng biên mềm, 

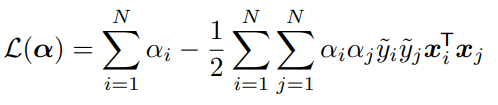
Vậy hàm mục tiêu mới là:



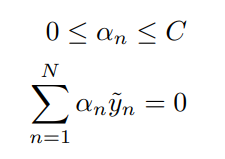


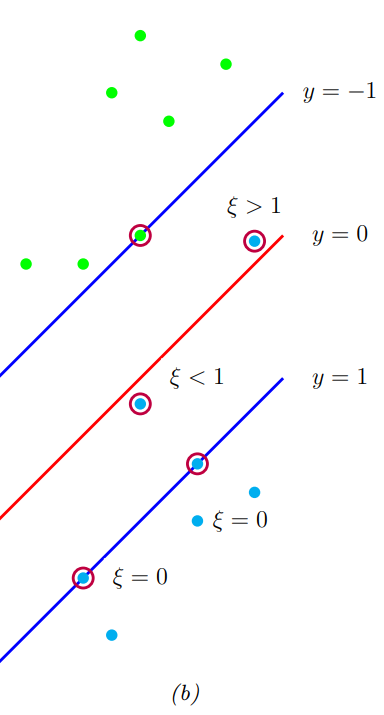


Tối ưu hóa w, w0, Epsilon ta bằng Lagrange đối ngẫu:



Mặc dù đối ngẫu giống hệt với trường hợp độ rộng khoảng biên cứng - Hard Margin. Nhưng khác nhau về KKT condition:





Nếu:

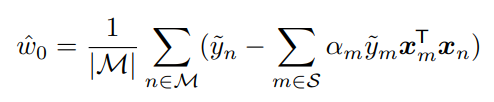
 điểm dữ liệu sẽ bị bỏ qua.

 thì,  vậy điểm dữ liệu sẽ nằm trên khoảng biên.

 thì điểm dữ liệu sẽ nằm bên trong khoảng biên. Lúc này

1.  có thể, có khả năng điểm được phân lớp đúng
2.  điểm được phân lớp sai

 là biên trên - upper bound của của số lượng điểm bị phân lớp sai



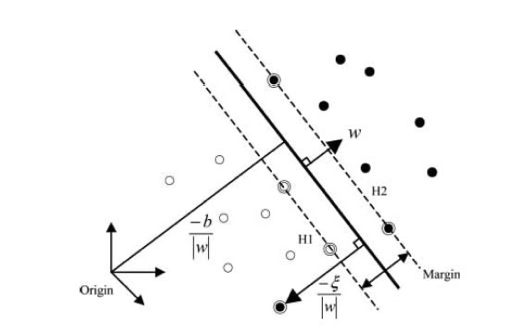
với:  là tập hợp các điểm có 

#### **3.** [**Fin\_12\_2021.pdf\_HKI\_(2021-2022)**](https://docs.google.com/document/d/1FBqWzRhwBudjJDBuK2_z72pM98EndxA2/edit#heading=h.t37itj2wog4b)

**Câu 5. (0.75 điểm)**

**b.** Giải thích ý nghĩa của các biến bù *Epsilon* và thông số *C* trong máy vec tơ hỗ trợ với khoảng biên mềm (soft margin) (*0.5 điểm*)  
**→ Chương 8**

[Nghia]

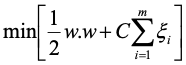


1. Biến bù (Slack variables):

- Các biến bù được sử dụng để đo lường mức độ vi phạm của mỗi điểm dữ liệu đến siêu phẳng tối ưu.

- Mỗi điểm dữ liệu có thể có một biến bù tương ứng. Giá trị của biến bù cho biết mức độ vi phạm của điểm dữ liệu đó đến siêu phẳng. Nếu biến bù có giá trị lớn, điểm dữ liệu đó được coi là vi phạm mức độ lớn hơn và có khả năng bị phân loại sai. Nếu biến bù có giá trị nhỏ hoặc bằng 0, điểm dữ liệu đó nằm đúng phía của siêu phẳng hoặc nằm trong vùng an toàn.

- Mục tiêu là tìm một giải pháp tối ưu (minimize hàm dưới) mà tổng các biến bù là nhỏ nhất để đạt được sự cân bằng giữa việc tìm siêu phẳng tối ưu và vi phạm của các điểm dữ liệu.



2. Thông số C:

- Thông số C được sử dụng để kiểm soát mức độ phân loại sai sót mà chúng ta chấp nhận trong SVM với khoảng biên mềm.

- Thông số C quyết định sự cân nhắc giữa việc tìm một siêu phẳng tối ưu và vi phạm của các điểm dữ liệu. Nếu C lớn, mức độ vi phạm sẽ nhỏ, điều này dẫn đến việc SVM sẽ cố gắng tìm một siêu phẳng chặt chẽ hơn, nhưng có khả năng overfitting.

- Ngược lại, nếu C nhỏ, SVM sẽ chấp nhận một mức độ vi phạm lớn hơn và tìm một siêu phẳng có độ rộng lớn hơn, dẫn đến tính tổng quát hóa tốt hơn trên dữ liệu mới.

- Lựa chọn thông số C phụ thuộc vào đặc tính của dữ liệu và mục tiêu của bài toán. Nếu dữ liệu nhiễu nhiều hoặc có sự chồng lấn mạnh giữa các lớp, ta có thể sử dụng C lớn để giảm thiểu phân loại sai sót. Ngược lại, nếu dữ liệu ít nhiễu và phân tách rõ ràng, ta có thể sử dụng C nhỏ để đạt được tính tổng quát hóa tốt hơn.

#### **4.** [**Fin\_12\_2021.pdf\_HKII\_(2021-2022)**](https://docs.google.com/document/d/1BLSqsCee6LnqI5rh2J_ZeN3whI-TOJl-VinGDjto97A/edit#)

Câu 5 **(*Trùng* với câu 5.** [**Fin\_202.pdf\_HKII\_(2020-2021)**](https://docs.google.com/document/d/1DkbvgkUJOOQcx9eNk-DsMT5MwyhNMvLE/edit#)**)**