Học Máy

(Machine Learning)

Thân Quang Khoát

khoattq@soict.hust.edu.vn

Viện Công nghệ thông tin và Truyền thông
Trường Đại học Bách Khoa Hà Nội

Năm 2015

Nội dung môn học:

- Giới thiệu chung
- Các phương pháp học không giám sát
- **Các phương pháp học có giám sát**
  - **Máy vectơ hỗ trợ (Support vector machine)**
- Đánh giá hiệu năng hệ thống học máy

Máy vectơ hỗ trợ: Giới thiệu (1)

- Máy vectơ hỗ trợ (Support vector machine - SVM) được đề cử 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
- SVM là một phương pháp **phân lớp tuyến tính** (linear classifier), với mục đích xác định một siêu phẳng (hyperplane) để phân tách **hai lớp** của dữ liệu.
  Ví dụ: lớp có nhãn dương (positive) và lớp có nhãn âm (negative)
- **Các hàm nhân (kernel functions)**, cũng được gọi là các hàm biến đổi (transformation functions), được dùng cho các trường hợp phân lớp phi tuyến

Máy vectơ hỗ trợ: Giới thiệu (2)

- SVM có một nền tảng lý thuyết chặt chẽ
- SVM là một phương pháp tốt (phù hợp) đối với những bài toán phân lớp có không gian rất nhiều chiều (các đối tượng cần phân lớp được biểu diễn bởi một tập rất lớn các thuộc tính)
- SVM đã được biết đến là một trong số các phương pháp phân lớp tốt nhất đối với các bài toán phân lớp văn bản (text classification)

Máy vectơ hỗ trợ: Giới thiệu (3)

- *Các vectơ được ký hiệu bởi các chữ đậm nét!*
- Biểu diễn tập r các ví dụ huấn luyện (training examples)

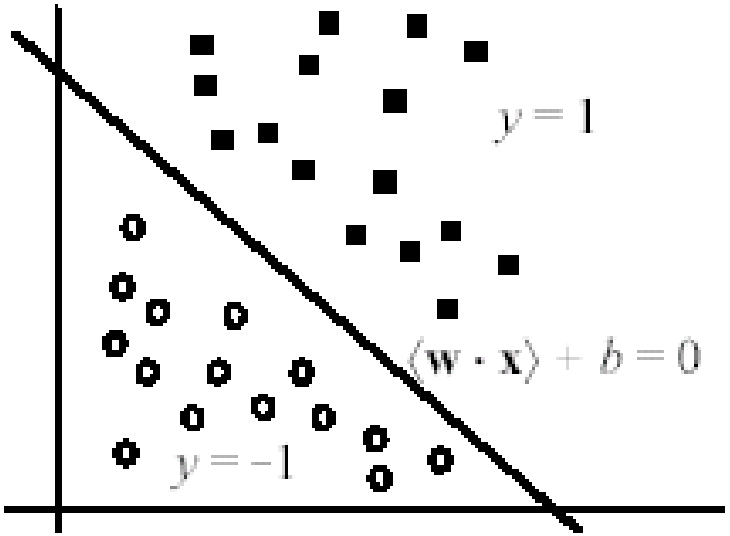
  $\{(\mathbf{x_1}, y_1), (\mathbf{x_2}, y_2), \ldots, (\mathbf{x_r}, y_r)\}$,
  - $\mathbf{x_i}$ là một **vectơ đầu vào** được biểu diễn trong không gian $X \subseteq R^n$
  - y_i là một **nhãn lớp** (giá trị đầu ra), $y_i \in \{1,-1\}$
  - y_i=1: lớp *dương* (positive); y_i=-1: lớp *âm* (negative)
- SVM xác định một hàm phân tách tuyến tính

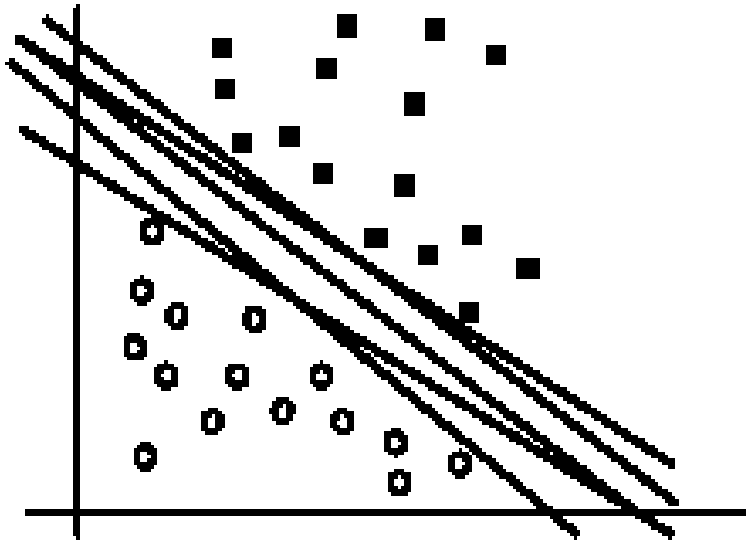
  $$f(\mathbf{x}) = \langle \mathbf{w} \cdot \mathbf{x} \rangle + b$$ [Eq.1]
  - **w** là vectơ trọng số các thuộc tính; b là một giá trị số thực
- Sao cho với mỗi $\mathbf{x_i}$:

  $$y_i = \begin{cases} 1 & if \quad \langle \mathbf{w} \cdot \mathbf{x}_i \rangle + b \geq 0 \\ -1 & if \quad \langle \mathbf{w} \cdot \mathbf{x}_i \rangle + b < 0 \end{cases}$$ [Eq.2]

Siêu phẳng phân tách

- Siêu phẳng phân tách các ví dụ huấn luyện lớp dương và các ví dụ huấn luyện lớp âm: $\langle \mathbf{w} \cdot \mathbf{x} \rangle + b = 0$
- Còn được gọi là ranh giới (bề mặt) quyết định
- Tồn tại nhiều siêu phẳng phân tách. Chọn cái nào?

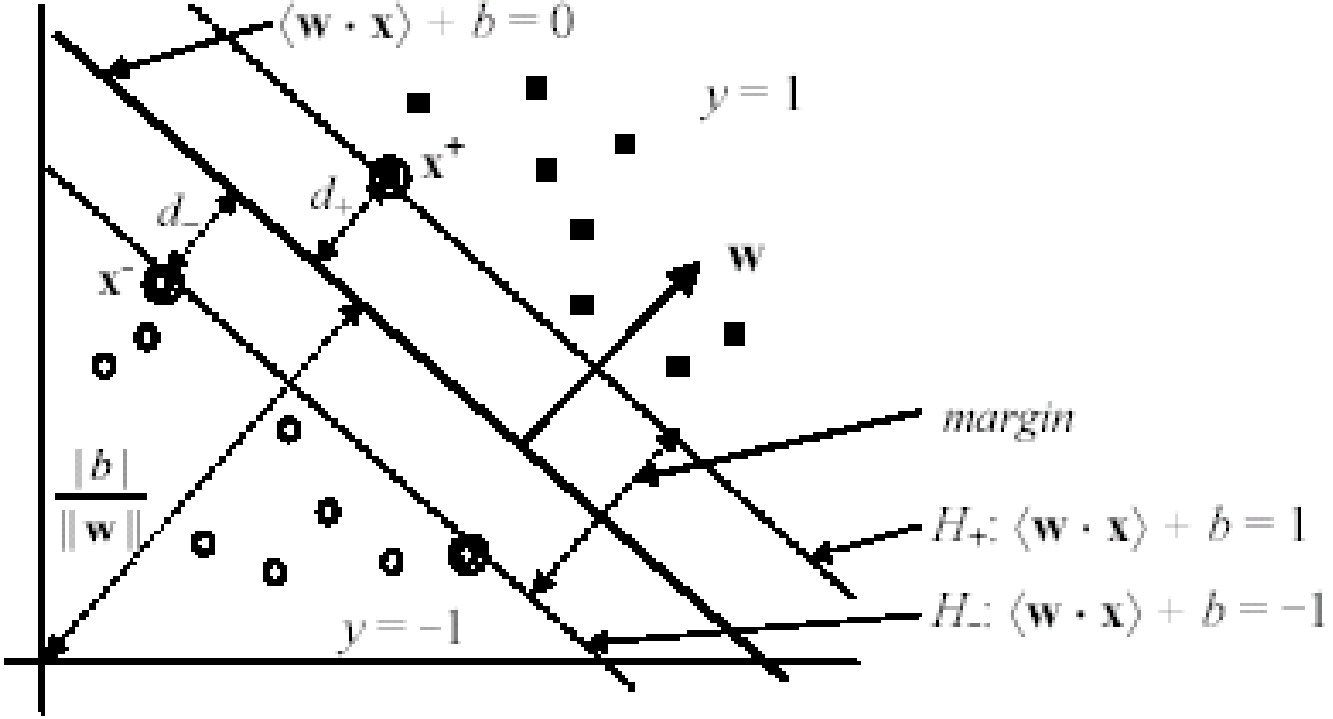




[Liu, 2006]

Mặt siêu phẳng có lề cực đại

- SVM lựa chọn mặt siêu phẳng phân tách có lề (margin) lớn nhất
- Lý thuyết học máy đã chỉ ra rằng *một mặt siêu phẳng phân tách như thế sẽ tối thiểu hóa giới hạn lỗi (phân lớp) mắc phải (so với mọi siêu phẳng khác)*



[Liu, 2006]

Phân tách tuyến tính (linear separability)

- Giả sử rằng tập dữ liệu (tập các ví dụ huấn luyện) có thể phân tách được một cách tuyến tính
- Xét một ví dụ của lớp dương ($\mathbf{x}^+$,1) và một ví dụ của lớp âm ($\mathbf{x}^-$,-1) *gần nhất* đối với siêu phẳng phân tách H_0 ($\langle \mathbf{w}\cdot\mathbf{x}\rangle + b = 0$)
- Định nghĩa 2 siêu phẳng lề song song với nhau
  - H_+ đi qua $\mathbf{x}^+$, và song song với H_0
  - H_- đi qua $\mathbf{x}^-$, và song song với H_0

H_+: $\langle \mathbf{w}\cdot\mathbf{x}^+\rangle + b = 1$

H_-: $\langle \mathbf{w}\cdot\mathbf{x}^-\rangle + b = -1$ [Eq.3]

sao cho: $\langle \mathbf{w}\cdot\mathbf{x}_i\rangle + b \geq 1$, nếu $y_i = 1$

$\langle \mathbf{w}\cdot\mathbf{x}_i\rangle + b \leq -1$, nếu $y_i = -1$

Tính toán mức lề (1)

- **Mức lề** (margin) là khoảng cách giữa 2 siêu phẳng lề H_+ và H_-. Trong hình vẽ nêu trên:
  - d_+ là khoảng cách giữa H_+ và H_0
  - d_- là khoảng cách giữa H_- và H_0
  - $(d_+ + d_-)$ là mức lề
- Trong không gian vectơ, **khoảng cách** từ một điểm $\mathbf{x}_i$ đến siêu phẳng ($\langle \mathbf{w} \cdot \mathbf{x} \rangle + b = 0$) là:

$$\frac{|\langle \mathbf{w} \cdot \mathbf{x}_i \rangle + b|}{\| \mathbf{w} \|} \qquad \text{[Eq.4]}$$

trong đó $\|\mathbf{w}\|$ là độ dài của $\mathbf{w}$:

$$\| \mathbf{w} \| = \sqrt{< \mathbf{w} \cdot \mathbf{w} >} = \sqrt{w_1^2 + w_1^2 + ... + w_1^2} \qquad \text{[Eq.5]}$$

Tính toán mức lề (2)

- Tính toán d_+: khoảng cách từ $\mathbf{x}^+$ đến ($\langle \mathbf{w} \cdot \mathbf{x} \rangle + b = 0$)
  - Áp dụng các biểu thức [Eq.3-4]:

$$d_+ = \frac{|\langle \mathbf{w} \cdot \mathbf{x}^+ \rangle + b|}{\|\mathbf{w}\|} = \frac{|1|}{\|\mathbf{w}\|} = \frac{1}{\|\mathbf{w}\|} \qquad \text{[Eq.6]}$$

- Tính toán d_-: khoảng cách từ $\mathbf{x}^-$ đến ($\langle \mathbf{w} \cdot \mathbf{x} \rangle + b = 0$)
  - Áp dụng các biểu thức [Eq.3-4]:

$$d_- = \frac{|\langle \mathbf{w} \cdot \mathbf{x}^- \rangle + b|}{\|\mathbf{w}\|} = \frac{|-1|}{\|\mathbf{w}\|} = \frac{1}{\|\mathbf{w}\|} \qquad \text{[Eq.7]}$$

- Tính toán mức lề

$$margin = d_+ + d_- = \frac{2}{\|\mathbf{w}\|} \qquad \text{[Eq.8]}$$

Học SVM: Cực đại hóa mức lề (1)

Định nghĩa (**Linear SVM –** Trường hợp **phân tách được**)

- Tập gồm r ví dụ huấn luyện có thể phân tách tuyến tính

  $D = \{(\mathbf{x_1}, y_1), (\mathbf{x_2}, y_2), \ldots, (\mathbf{x_r}, y_r)\}$

- SVM học một phân lớp (classifier) mà có mức lề cực đại
- Tương đương với việc giải quyết **bài toán tối ưu bậc hai** sau đây
  - Tìm ***w*** và *b* sao cho: $margin = \frac{2}{\|\mathbf{w}\|}$ đạt cực đại
  - Với điều kiện:

$$\begin{cases} \langle \mathbf{w} \cdot \mathbf{x_i} \rangle + b \geq 1, \text{ if } y_i = 1 \\ \langle \mathbf{w} \cdot \mathbf{x_i} \rangle + b \leq -1, \text{ if } y_i = -1 \end{cases}$$

  với mọi ví dụ huấn luyện $\mathbf{x_i}$ (i=1..r)

Học SVM: Cực đại hóa mức lề (2)

- Học SVM tương đương với giải quyết **bài toán cực tiểu hóa có ràng buộc** sau đây

Cực tiểu hóa: $\frac{\langle \mathbf{w} \cdot \mathbf{w} \rangle}{2}$ [Eq.9]

Với điều kiện: $\begin{cases} \langle \mathbf{w} \cdot \mathbf{x}_i \rangle + b \geq 1, & if \ \ y_i = 1 \\ \langle \mathbf{w} \cdot \mathbf{x}_i \rangle + b \leq -1, & if \ \ y_i = -1 \end{cases}$

- tương đương với

Cực tiểu hóa: $\frac{\langle \mathbf{w} \cdot \mathbf{w} \rangle}{2}$ [Eq.10]

Với điều kiện: $y_i(\langle \mathbf{w} \cdot \mathbf{x}_i \rangle + b) \geq 1, \quad \forall i = 1..r$

Lý thuyết tối ưu có ràng buộc (1)

- Bài toán cực tiểu hóa có ràng buộc đẳng thức:

  Cực tiểu hóa $f(\mathbf{x})$, với điều kiện $g(\mathbf{x})=0$

- Điều kiện cần để $\mathbf{x_0}$ là một lời giải:

$$\begin{cases} \dfrac{\partial}{\partial \mathbf{x}}\left(f(\mathbf{x}) + \alpha g(\mathbf{x})\right)\Big|_{\mathbf{x}=\mathbf{x}_0} = 0 \\ g(\mathbf{x}) = 0 \end{cases};$$

với α là một hệ số nhân (multiplier) Lagrange

- Trong trường hợp có nhiều ràng buộc đẳng thức $g_i(\mathbf{x})=0$ (i=1..r), cần một hệ số nhân Lagrange cho mỗi ràng buộc:

$$\begin{cases} \dfrac{\partial}{\partial \mathbf{x}}\left(f(\mathbf{x}) + \sum_{i=1}^{r} \alpha_i g_i(\mathbf{x})\right)\Big|_{\mathbf{x}=\mathbf{x}_0} = 0 \\ g_i(\mathbf{x}) = 0 \end{cases};$$

Lý thuyết tối ưu có ràng buộc (2)

- Bài toán cực tiểu hóa có các ràng buộc bất đẳng thức:

  Cực tiểu hóa $f(\mathbf{x})$, với các điều kiện $g_i(\mathbf{x}) \leq 0$

- Điều kiện cần để $\mathbf{x_0}$ là một lời giải:

$$\begin{cases} \dfrac{\partial}{\partial \mathbf{x}} \left(f(\mathbf{x}) + \sum_{i=1}^{r} \alpha_i g_i(\mathbf{x}) \right) \Big|_{\mathbf{x}=\mathbf{x_0}} = 0; & \text{với } \alpha_i \geq 0 \\ g_i(\mathbf{x}) \leq 0 \end{cases}$$

- Hàm

$$L = f(\mathbf{x}) + \sum_{i=1}^{r} \alpha_i g_i(\mathbf{x})$$ được gọi là hàm Lagrange

Học SVM: giải bài toán cực tiểu hóa

- Biểu thức Lagrange

$$L_P(\mathbf{w}, b, \boldsymbol{\alpha}) = \frac{1}{2}\langle \mathbf{w} \cdot \mathbf{w} \rangle - \sum_{i=1}^{r} \alpha_i [y_i(\langle \mathbf{w} \cdot \mathbf{x}_i \rangle + b) - 1] \quad \text{[Eq.11]}$$

trong đó α_i (≥ 0) là các hệ số nhân Lagrange

- Lý thuyết tối ưu chỉ ra rằng một lời giải tối ưu cho [Eq.11] phải thỏa mãn các điều kiện nhất định, được gọi là **các điều kiện Karush-Kuhn-Tucker** (là các điều kiện cần, nhưng không phải là các điều kiện đủ)
- Các điều kiện Karush-Kuhn-Tucker đóng vai trò trung tâm trong cả lý thuyết và ứng dụng của lĩnh vực tối ưu có ràng buộc

Tập điều kiện Karush-Kuhn-Tucker

$$\frac{\partial L_P}{\partial \mathbf{w}} = \mathbf{w} - \sum_{i=1}^{r} \alpha_i y_i \mathbf{x_i} = 0 \qquad \text{[Eq.12]}$$

$$\frac{\partial L_P}{\partial b} = -\sum_{i=1}^{r} \alpha_i y_i = 0 \qquad \text{[Eq.13]}$$

$$y_i\left(\left\langle \mathbf{w} \cdot \mathbf{x_i} \right\rangle + b\right) - 1 \geq 0, \ \forall \mathbf{x_i} \ (i = 1..r) \qquad \text{[Eq.14]}$$

$$\alpha_i \geq 0 \qquad \text{[Eq.15]}$$

$$\alpha_i \left(y_i\left(\left\langle \mathbf{w} \cdot \mathbf{x_i} \right\rangle + b\right) - 1\right) = 0 \qquad \text{[Eq.16]}$$

- [Eq.14] chính là tập các ràng buộc ban đầu
- Điều kiện *bổ sung* [Eq.16] chỉ ra rằng chỉ những ví dụ (điểm dữ liệu) thuộc các mặt siêu phẳng lề (H_+ và H_-) mới có $\alpha_i > 0$ – bởi vì với những ví đụ đó thì $y_i(\langle \boldsymbol{w} \cdot \boldsymbol{x}_i \rangle + b) - 1 = 0$
  - →Những ví dụ (điểm dữ liệu) này được gọi là **các vectơ hỗ trợ**!
- Đối với các ví dụ khác (không phải là các vectơ hỗ trợ) thì $\alpha_i = 0$

Học SVM: giải bài toán cực tiểu hóa

- Trong trường hợp tổng quát, các điều kiện Karush-Kuhn-Tucker là *cần* đối với một lời giải tối ưu, nhưng *chưa đủ*
- Tuy nhiên đối với SVM, bài toán cực tiểu hóa có *hàm mục tiêu lồi* (convex) và *các ràng buộc tuyến tính*, thì các điều kiện Karush-Kuhn-Tucker là *cần và đủ* đối với một lời giải tối ưu
- Giải quyết bài toán tối ưu này vẫn là một nhiệm vụ khó khăn, do sự tồn tại của các ràng buộc bất đẳng thức!
- Phương pháp Lagrange giải quyết bài toán tối ưu hàm lồi dẫn đến một bài toán **đối ngẫu (dual)** của bài toán tối ưu

  → Dễ giải quyết hơn so với bài toán tối ưu **ban đầu (primal)**

Học SVM: Biểu thức đối ngẫu

- Để thu được biểu thức **đối ngẫu** từ biểu thức **ban đầu**:
  - →Gán giá trị bằng 0 đối với các đạo hàm bộ phận của biểu thức Lagrange trong [Eq.11] đối với **các biến ban đầu** (**w** và *b*)
  - →Sau đó, áp dụng các quan hệ thu được đối với biểu thức Lagrange
    - Tức là: áp dụng các biểu thức [Eq.12-13] vào biểu thức Lagrange ban đầu ([Eq.11]) để loại bỏ các biến ban đầu (**w** và b)
- Biểu thức đối ngẫu L_D

$$L_D(\boldsymbol{\alpha}) = \sum_{i=1}^{r} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{r} \alpha_i \alpha_j y_i y_j \langle \mathbf{x}_i \cdot \mathbf{x}_j \rangle \qquad \text{[Eq.17]}$$

- Cả hai biểu thức L_P và L_D đều là các biểu thức Lagrange
  - Dựa trên cùng một hàm một tiêu – nhưng với các ràng buộc khác nhau
  - Lời giải tìm được, bằng cách cực tiểu hóa L_P hoặc cực đại hóa L_D

Bài toán tối ưu đối ngẫu

Cực đại hóa: $L_D(\boldsymbol{\alpha}) = \sum_{i=1}^{r} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{r} \alpha_i \alpha_j y_i y_j \langle \mathbf{x_i} \cdot \mathbf{x_j} \rangle$ [Eq.18]

Với điều kiện: $\begin{cases} \sum_{i=1}^{r} \alpha_i y_i = 0 \\ \alpha_i \geq 0, \forall i = 1..r \end{cases}$

- Đối với hàm mục tiêu là hàm lồi và các ràng buộc tuyến tính, giá trị cực đại của $\boldsymbol{L_D}$ xảy ra tại cùng các giá trị của **w**, *b* và α_i giúp đạt được giá trị cực tiểu của $\boldsymbol{L_P}$
- Giải bài toán [Eq.18], ta thu được các hệ số nhân Lagrange α_i (các hệ số α_i này sẽ được dùng để tính **w** và b)
- Giải bài toán [Eq.18] cần đến *các phương pháp lặp* (để giải quyết bài toán tối ưu hàm lồi bậc hai có các ràng buộc tuyến tính)

  → Chi tiết các phương pháp này nằm ngoài phạm vi của bài giảng!

Tính các giá trị $\boldsymbol{w^*}$ và b^*

- Gọi SV là tập các vectơ hỗ trợ
  - SV là tập con của tập r các ví dụ huấn luyện ban đầu
  - $\rightarrow \alpha_i > 0$ đối với các vectơ hỗ trợ $\boldsymbol{x_i}$
  - $\rightarrow \alpha_i = 0$ đối với các vectơ không phải là vectơ hỗ trợ $\boldsymbol{x_i}$
- Sử dụng biểu thức [Eq.12], ta có thể tính được giá trị $\boldsymbol{w}^*$

$$\mathbf{w^*} = \sum_{i=1}^{r} \alpha_i y_i \mathbf{x_i} = \sum_{\mathbf{x_i} \in SV} \alpha_i y_i \mathbf{x_i}; \qquad \text{bởi vì } \forall \boldsymbol{x_i} \notin SV: \ \alpha_i = 0$$

- Sử dụng biểu thức [Eq.16] và (bất kỳ) một vectơ hỗ trợ $\boldsymbol{x}_k$, ta có
  - $\alpha_k[y_k(\langle \boldsymbol{w}^*, \boldsymbol{x}_k \rangle + b^*) - 1] = 0$
  - Nhớ rằng $\alpha_k > 0$ đối với mọi vectơ hỗ trợ $\boldsymbol{x}_k$
  - Vì vậy: $y_k(\langle \boldsymbol{w}^*, \boldsymbol{x}_k \rangle + b^*) - 1 = 0$
  - Từ đây, ta tính được giá trị $b^* = y_k - \langle \boldsymbol{w}^*, \boldsymbol{x}_k \rangle$

Phân lớp cho ví dụ mới

- **Ranh giới quyết định phân lớp** được xác định bởi siêu phẳng:

$$f(\mathbf{x}) = \langle \mathbf{w}^* \cdot \mathbf{x} \rangle + b^* = \sum_{\mathbf{x_i} \in SV} \alpha_i y_i \langle \mathbf{x_i} \cdot \mathbf{x} \rangle + b^* = 0 \qquad \text{[Eq.19]}$$

- Đối với một ví dụ cần phân lớp **z**, cần tính giá trị:

$$sign(\ \langle \mathbf{w}^* \cdot \mathbf{z} \rangle + b^*) = sign\left(\sum_{\mathbf{x_i} \in SV} \alpha_i y_i \langle \mathbf{x_i} \cdot \mathbf{z} \rangle + b^* \right) \qquad \text{[Eq.20]}$$

→ Nếu biểu thức [Eq.20] trả về giá trị 1, thì ví dụ **z** được phân vào lớp có nhãn dương (positive); ngược lại, được phân vào lớp có nhãn âm (negative)

- Việc phân lớp này:
  - Chỉ phụ thuộc vào các vectơ hỗ trợ
  - Chỉ cần giá trị tích vô hướng (tích trong) của 2 vectơ (chứ không cần biết giá trị của 2 vectơ đấy)

Linear SVM: Không phân tách được (1)

- Phương pháp SVM trong trường hợp hai lớp không thể phân tách được bằng một siêu phẳng?
  - Trường hợp phân lớp tuyến tính và phân tách được là lý tưởng (ít xảy ra)
  - Tập dữ liệu có thể chứa nhiễu, lỗi (vd: một số ví dụ được gán nhãn lớp sai)
- Đối với trường hợp phân tách được, bài toán tối ưu:

Cực tiểu hóa: $\frac{\langle \mathbf{w} \cdot \mathbf{w} \rangle}{2}$

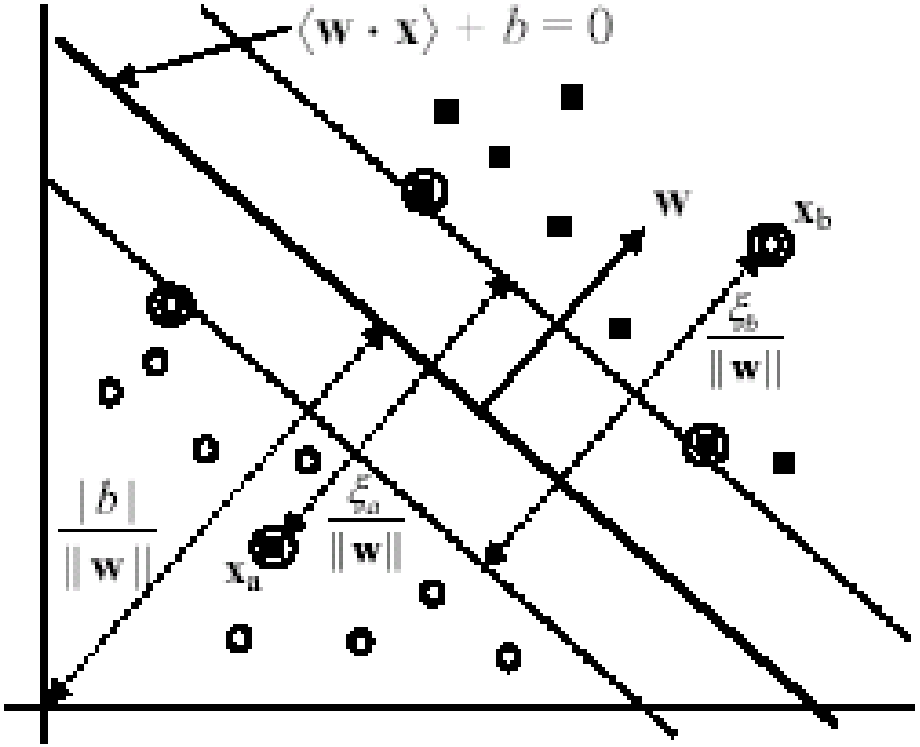
Với điều kiện: $y_i(\langle \mathbf{w} \cdot \mathbf{x}_i \rangle + b) \geq 1, \quad i = 1, 2, ..., r$

- Nếu tập dữ liệu chứa nhiễu, các điều kiện có thể không được thỏa mãn

  → Không tìm được lời giải (**w*** và b*)!

Linear SVM: Không phân tách được (2)

Hai ví dụ nhiễu $\mathbf{x_a}$ và $\mathbf{x_b}$ được gán nhãn lớp sai



[Liu, 2006]

Nới lỏng các điều kiện

- Để làm việc với các dữ liệu chứa nhiễu, cần nới lỏng các điều kiện lề (margin constraints) bằng cách sử dụng các biến **slack** ξ_i (≥ 0)

  $\langle \mathbf{w} \cdot \mathbf{x}_i \rangle + b \geq 1 - \xi_i$ đối với các ví dụ có giá trị $y_i = 1$

  $\langle \mathbf{w} \cdot \mathbf{x}_i \rangle + b \leq -1 + \xi_i$ đối với các ví dụ có giá trị $y_i = -1$

- Đối với một ví dụ nhiễu/lỗi: $\xi_i > 1$

- Các điều kiện mới đối với trường hợp (phân lớp tuyến tính) không thể phân tách được:

$$y_i \left(\langle \boldsymbol{w} \cdot \boldsymbol{x}_i \rangle + b \right) \geq 1 - \xi_i, \quad \forall i = 1..r$$
$$\xi_i \geq 0, \quad \forall i = 1..r$$

Tích hợp lỗi trong hàm mục tiêu

- Cần phải tích hợp lỗi trong hàm tối ưu mục tiêu
- Bằng cách gán giá trị chi phí (cost) cho các lỗi, và tích hợp chi phí này trong hàm mục tiêu mới:

Cực tiểu hóa:

$$\frac{\langle \mathbf{w} \cdot \mathbf{w} \rangle}{2} + C\left(\sum_{i=1}^{r} \xi_i\right)^k$$

trong đó C (>0) là tham số xác định ***mức độ phạt (penalty degree)*** đối với các lỗi

→ Giá trị C càng lớn, thì mức độ phạt càng cao đối với các lỗi

- k =1 thường được sử dụng
  - Lý do: Thu được biểu thức đối ngẫu (dual formulation) đơn giản hơn – không chứa ξ_i và các hệ số nhân Lagrange của chúng

Bài toán tối ưu mới

Cực tiểu hóa: $$\frac{\langle \mathbf{w} \cdot \mathbf{w} \rangle}{2} + C \sum_{i=1}^{r} \xi_i$$ [Eq.21]

Với điều kiện: $$\begin{cases} y_i(\langle \mathbf{w} \cdot \mathbf{x}_i \rangle + b) \geq 1 - \xi_i, & \forall i = 1..r \\ \xi_i \geq 0, & \forall i = 1..r \end{cases}$$

- Bài toán tối ưu mới này được gọi là **Soft-margin SVM**
- Biểu thức tối ưu Lagrange: [Eq.22]

$$L_P = \frac{1}{2}\langle \mathbf{w} \cdot \mathbf{w} \rangle + C \sum_{i=1}^{r} \xi_i - \sum_{i=1}^{r} \alpha_i [y_i(\langle \mathbf{w} \cdot \mathbf{x}_i \rangle + b) - 1 + \xi_i] - \sum_{i=1}^{r} \mu_i \xi_i$$

trong đó α_i (≥ 0) và μ_i (≥ 0) là các hệ số nhân Lagrange

Tập điều kiện Karush-Kuhn-Tucker (1)

$$\frac{\partial L_P}{\partial \mathbf{w}} = \mathbf{w} - \sum_{i=1}^{r} \alpha_i y_i \mathbf{x_i} = 0 \qquad \text{[Eq.23]}$$

$$\frac{\partial L_P}{\partial b} = -\sum_{i=1}^{r} \alpha_i y_i = 0 \qquad \text{[Eq.24]}$$

$$\frac{\partial L_P}{\partial \xi_i} = C - \alpha_i - \mu_i = 0, \quad \forall i = 1..r \qquad \text{[Eq.25]}$$

Tập điều kiện Karush-Kuhn-Tucker (2)

$$y_i\left(\langle \mathbf{w} \cdot \mathbf{x_i} \rangle + b\right) - 1 + \xi_i \geq 0, \quad \forall i = 1..r \qquad \text{[Eq.26]}$$

$$\xi_i \geq 0 \qquad \text{[Eq.27]}$$

$$\alpha_i \geq 0 \qquad \text{[Eq.28]}$$

$$\mu_i \geq 0 \qquad \text{[Eq.29]}$$

$$\alpha_i\left(y_i\left(\langle \mathbf{w} \cdot \mathbf{x_i} \rangle + b\right) - 1 + \xi_i\right) = 0 \qquad \text{[Eq.30]}$$

$$\mu_i \xi_i = 0 \qquad \text{[Eq.31]}$$

Chuyển về biểu thức đối ngẫu

- Giống như với trường hợp dữ liệu có thể phân tách được, chúng ta chuyển biểu thức Lagrange từ dạng ban đầu (primal formulation) về dạng đối ngẫu (dual formulation)
  - Gán giá trị bằng 0 cho các đạo hàm bộ phận của biểu thức Lagrange ([Eq.22]) đối với **các biến ban đầu** (**w**, b, ξ_i)
  - Thay thế các kết quả thu được vào biểu thức Lagrange ban đầu

  → Sử dụng các kết quả của các biểu thức [Eq.23-25] để thay thế vào trong biểu thức Lagrange ban đầu [Eq.22]

- Từ biểu thức [Eq.25], ta có: $C - \alpha_i - \mu_i = 0$,
  - và bởi vì: $\mu_i \geq 0$,
  - nên ta suy ra điều kiện: $\alpha_i \leq C$

Biểu thức đối ngẫu

Cực đại hóa: $$L_D(\boldsymbol{\alpha}) = \sum_{i=1}^{r} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{r} \alpha_i \alpha_j y_i y_j \langle \mathbf{x_i} \cdot \mathbf{x_j} \rangle$$

Với điều kiện: $$\begin{cases} \sum_{i=1}^{r} \alpha_i y_i = 0 \\ 0 \le \alpha_i \le C, \quad \forall i = 1..r \end{cases}$$ [Eq.32]

- ξ_i và các hệ số nhân Lagrange của chúng (μ_i) không xuất hiện trong biểu thức đối ngẫu

  → Hàm mục tiêu giống hệt như đối với bài toán phân lớp tuyến tính phân tách được (separable linear classification)!

- Khác biệt duy nhất là tập các ràng buộc mới: $\alpha_i \le C$

Tìm lời giải cho các biến ban đầu

- Bài toán đối ngẫu [Eq.32] được giải quyết bằng *các phương pháp lặp* (để giải quyết bài toán tối ưu hàm lồi bậc hai có các ràng buộc tuyến tính)
- Các giá trị (hệ số nhân Lagrange) α_i lời giải được sử dụng để tính toán **w*** và *b**
  - **w*** được xác định sử dụng biểu thức [Eq.23]
  - *b** được xác định sử dụng các điều kiện bổ sung Karush-Kuhn-Tucker trong [Eq.30-31] …nhưng, có vấn đề: ξ_i chưa biết!
- Để tính được b*
  - Từ [Eq.25] và [Eq.31], ta suy ra được: ξ_i=0 nếu α_i<*C*
  - Vì vậy, ta có thể sử dụng một ví dụ học $\boldsymbol{x}_k$ thỏa mãn điều kiện (0<α_k<*C*) và [Eq.30] (với ξ_k =0) để tính toán *b**
  - Đến đây, việc tính toán *b** tương tự như với trường hợp phân lớp tuyến tính phân tách được!

Các đặc điểm quan trọng

- Từ các biểu thức [Eq.25-31], ta có thể suy ra các kết luận sau:

$$\begin{array}{llll} \text{If} & \alpha_i = 0 & \text{then} & y_i\left(\langle \mathbf{w}\cdot\mathbf{x_i}\rangle + b\right) \geq 1, \quad \text{and} \quad \xi_i = 0 \\ \text{If} & 0 < \alpha_i < C & \text{then} & y_i\left(\langle \mathbf{w}\cdot\mathbf{x_i}\rangle + b\right) = 1, \quad \text{and} \quad \xi_i = 0 \\ \text{If} & \alpha_i = C & \text{then} & y_i\left(\langle \mathbf{w}\cdot\mathbf{x_i}\rangle + b\right) < 1, \quad \text{and} \quad \xi_i > 0 \end{array} \qquad \text{[Eq.33]}$$

- Biểu thức [Eq.33] thể hiện một đặc điểm rất quan trọng của SVM
  - Lời giải được xác định dựa trên rất ít (**sparse)** các giá trị α_i
    - Rất nhiều ví dụ học nằm ngoài khoảng lề (margin area), và chúng có giá trị α_i bằng 0
    - Các ví dụ nằm trên lề ($y_i(\langle \boldsymbol{w}\cdot\boldsymbol{x}_i\rangle + b) = 1$ – chính là **các vectơ hỗ trợ**), thì có giá trị α_i khác không $(0 < \alpha_i < C)$
    - Các ví dụ nằm trong khoảng lề ($y_i(\langle \mathbf{w}\cdot\mathbf{x}_i\rangle + b) < 1$ – là các ví dụ nhiễu/lỗi), thì có giá trị $\alpha_i = C$
  - Nếu không có đặc điểm thưa thớt (sparsity) này, thì phương pháp SVM không thể hiệu quả đối với các tập dữ liệu lớn

Ranh giới quyết định phân lớp

- Ranh giới quyết định phân lớp chính là siêu phẳng:

$$\langle \mathbf{w}^* \cdot \mathbf{x} \rangle + b^* = \sum_{i=1}^{r} \alpha_i y_i \langle \mathbf{x}_i \cdot \mathbf{x} \rangle + b^* = 0$$

→ Rất nhiều ví dụ học $\boldsymbol{x_i}$ có giá trị α_i bằng 0! (chính là đặc điểm thưa thớt – sparsity – của phương pháp SVM)

- Đối với một ví dụ cần phân loại $\boldsymbol{z}$, nó được phân loại bởi:

$$sign(\langle \boldsymbol{w}^*, \boldsymbol{z} \rangle + b^*)$$

- Cần xác định giá trị phù hợp của tham số C (trong hàm tối ưu mục tiêu)

→ Thường được xác định bằng cách sử dụng một tập dữ liệu tối ưu (validation set)

Linear SVM: Tổng kết

- Sự phân lớp dựa vào siêu phẳng phân tách
- Siêu phẳng phân tách được xác định dựa trên tập **các vectơ hỗ trợ**
- Chỉ đối với các vectơ hỗ trợ, thì hệ số nhân Lagrange của chúng khác 0
  - Đối với các ví dụ huấn luyện khác (không phải là các vectơ hỗ trợ), thì hệ số nhân Lagrange của chúng bằng 0
- Việc xác định các vectơ hỗ trợ (trong số các ví dụ huấn luyện) đòi hỏi phải giải quyết bài toán tối ưu bậc hai
- Trong biểu thức đối ngẫu (L_D) và trong biểu thức biểu diễn siêu phẳng phân tách, các ví dụ huấn luyện chỉ xuất hiện bên trong các tích vô hướng (inner/dot-products) của các vectơ

Non-linear SVM

- Lưu ý: Các công thức trong phương pháp SVM đòi hỏi tập dữ liệu phải có thể phân lớp tuyến tính (có/không nhiễu)
- Trong nhiều bài toán thực tế, thì các tập dữ liệu có thể là phân lớp phi tuyến (non-linearly separable)
- Phương pháp phân loại SVM phi tuyến (Non-linear SVM):
  - Bước 1. **Chuyển đổi không gian biểu diễn đầu vào ban đầu sang một không gian khác** (thường có số chiều lớn hơn nhiều)
    → Dữ liệu được biểu diễn trong không gian mới (đã chuyển đổi) có thể phân lớp tuyến tính (linearly separable)
  - Bước 2. Áp dụng lại các công thức và các bước như trong phương pháp phân lớp SVM tuyến tính
- Không gian biểu diễn ban đầu: **Không gian đầu vào (input space)**
- Không gian biểu diễn sau khi chuyển đổi: **Không gian đặc trưng (feature space)**

Chuyển đổi không gian biểu diễn (1)

- Ý tưởng cơ bản là việc ánh xạ (chuyển đổi) biểu diễn dữ liệu từ không gian ban đầu *X* sang một không gian khác *F* bằng cách áp dụng một hàm ánh xạ phi tuyến ϕ

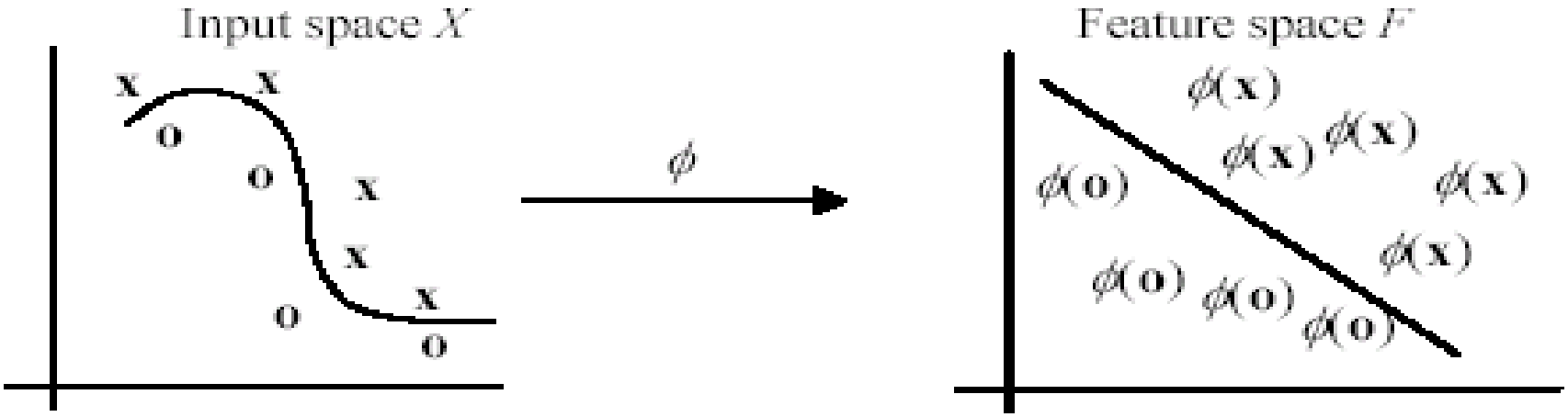
$$\phi : X \rightarrow F$$

$$\mathbf{x} \mapsto \phi(\mathbf{x})$$

- Trong không gian đã chuyển đổi, tập các ví dụ học ban đầu $\{(\boldsymbol{x_1}, y_1), (\boldsymbol{x_2}, y_2), \ldots, (\boldsymbol{x_r}, yr)\}$ được biểu diễn (ánh xạ) tương ứng:

$$\{(\phi(\mathbf{x_1}), y_1), (\phi(\mathbf{x_2}), y_2), \ldots, (\phi(\mathbf{x_r}), y_r)\}$$

Chuyển đổi không gian biểu diễn (2)



[Liu, 2006]

- Trong ví dụ này, không gian sau chuyển đổi vẫn là có số chiều bằng không gian ban đầu (2 chiều)
- Nhưng thông thường, số chiều của không gian sau chuyển đổi (feature space) lớn hơn (nhiều) số chiều của không gian ban đầu (input space)

Non-linear SVM: Bài toán tối ưu

- Sau quá trình chuyển đổi không gian biểu diễn, bài toán tối ưu:

Cực tiểu hóa: $$L_P = \frac{\langle \mathbf{w} \cdot \mathbf{w} \rangle}{2} + C \sum_{i=1}^{r} \xi_i$$ [Eq.34]

Với điều kiện: $$\begin{cases} y_i \left(\langle \mathbf{w} \cdot \phi(\mathbf{x_i}) \rangle + b \right) \geq 1 - \xi_i, \quad \forall i = 1..r \\ \xi_i \geq 0, \quad \forall i = 1..r \end{cases}$$

- Bài toán (tối ưu) đối ngẫu:

Cực đại hóa: $$L_D = \sum_{i=1}^{r} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{r} \alpha_i \alpha_j y_i y_j \langle \phi(\mathbf{x_i}) \cdot \phi(\mathbf{x_j}) \rangle$$ [Eq.35]

Với điều kiện: $$\begin{cases} \sum_{i=1}^{r} \alpha_i y_i = 0 \\ 0 \leq \alpha_i \leq C, \quad \forall i = 1..r \end{cases}$$

- Ranh giới quyết định phân lớp là siêu phẳng phân tách:

$$f(\mathbf{z}) = \langle \mathbf{w}^* \cdot \phi(\mathbf{z}) \rangle + b^* = \sum_{i=1}^{r} \alpha_i y_i \langle \phi(\mathbf{x_i}) \cdot \phi(\mathbf{z}) \rangle + b^*) = 0$$ [Eq.36]

Chuyển đổi không gian: Ví dụ

- Xét không gian biểu diễn ban đầu có 2 chiều, và chúng ta chọn hàm ánh xạ từ không gian ban đầu (2-D) sang không gian mới (3-D) như sau:

$$(x_1, x_2) \mapsto (x_1, x_2, \sqrt{2x_1x_2})$$

- Xét ví dụ học $(\boldsymbol{x} = (2, 3), y = -1)$ trong không gian ban đầu (2-D)

- Trong không gian sau chuyển đổi (3-D), thì ví dụ học này được biểu diễn như sau:

$$(\phi(\boldsymbol{x}) = (4, 9, 8.49), y = -1)$$

Chuyển đổi không gian: Trở ngại

- Việc chuyển đổi không gian một cách trực tiếp có thể gặp vấn đề về số chiều không gian quá lớn (curse of dimensionality)
- Ngay cả với một không gian ban đầu có số chiều không lớn, một hàm chuyển đổi (ánh xạ) thích hợp có thể trả về một không gian mới có số chiều rất lớn

  → "thích hợp" ở đây mang ý nghĩa là hàm chuyển đổi cho phép xác định không gian mới mà trong đó tập dữ liệu có thể phân lớp tuyến tính
- Vấn đề: Chi phí tính toán quá lớn đối với việc chuyển đổi không gian trực tiếp
- Rất may, việc chuyển đổi không gian trực tiếp là không cần thiết…

Các hàm nhân (Kernel functions)

- Trong biểu thức đối ngẫu ([Eq.35]) và trong biểu thức siêu phẳng phân tách ([Eq.36]):
  - Việc xác định trực tiếp (cụ thể) giá trị $\phi(\mathbf{x})$ và $\phi(\mathbf{z})$ là không cần thiết
  - Chỉ cần tính giá trị tích vô hướng vectơ $\langle \phi(\mathbf{x}) \cdot \phi(\mathbf{z}) \rangle$

  $\rightarrow$ *Việc chuyển đổi không gian trực tiếp là không cần thiết*!
- Nếu có thể *tính được tích vô hướng vectơ* $\langle \phi(\mathbf{x}) \cdot \phi(\mathbf{z}) \rangle$ *trực tiếp từ các vectơ* ***x*** *và* ***z***, thì không cần phải xác định (không cần biết):
  - vectơ đặc trưng (trong không gian sau chuyển đổi) $\phi(\mathbf{x})$, và
  - hàm chuyển đổi (ánh xạ) ϕ
- Trong phương pháp SVM, mục tiêu này đạt được thông qua việc sử dụng **các hàm nhân (kernel functions)**, được ký hiệu là *K*

$$K(\mathbf{x},\mathbf{z}) = \langle \phi(\mathbf{x}) \cdot \phi(\mathbf{z}) \rangle \qquad \text{[Eq.37]}$$

Hàm nhân: Ví dụ

- Hàm nhân đa thức

$$K(\mathbf{x},\mathbf{z}) = \langle \mathbf{x}\cdot\mathbf{z}\rangle^d \qquad \text{[Eq.38]}$$

- Xét hàm nhân đa thức với bậc d=2, đối với 2 vectơ được biểu diễn trong không gian 2 chiều: $\mathbf{x}=(x_1,x_2)$ và $\mathbf{z}=(z_1,z_2)$

$$\langle \mathbf{x}\cdot\mathbf{z}\rangle^2 = (x_1z_1 + x_2z_2)^2$$

$$= x_1^2 z_1^2 + 2x_1z_1x_2z_2 + x_2^2 z_2^2$$

$$= \langle (x_1^2, x_2^2, \sqrt{2}x_1x_2)\cdot(z_1^2, z_2^2, \sqrt{2}z_1z_2)\rangle$$

$$= \langle \phi(\mathbf{x})\cdot\phi(\mathbf{z})\rangle = K(\mathbf{x},\mathbf{z})$$

- Ví dụ trên thể hiện hàm nhân $\langle \mathbf{x}\cdot\mathbf{z}\rangle^2$ là một tích vô hướng của 2 vectơ $\phi(\mathbf{x})$ và $\phi(\mathbf{z})$ trong không gian sau chuyển đổi

Kernel trick

- Diễn giải chi tiết của các bước tính toán trong ví dụ trên chỉ mang mục đích giải thích (minh họa)
- Trong thực tế, ta không cần phải tìm (xác định) hàm ánh xạ ϕ
- Bởi vì: Ta có thể áp dụng hàm nhân *một cách trực tiếp*

  → Thay thế tất cả các giá trị tích vô hướng vectơ $\langle \phi(\mathbf{x}) \cdot \phi(\mathbf{z}) \rangle$ trong [Eq.35-36] bằng một hàm nhân được chọn $K(\mathbf{x},\mathbf{z})$ (ví dụ: hàm nhân đa thức $\langle \mathbf{x} \cdot \mathbf{z} \rangle^d$ trong [Eq.38])
- Chiến lược này được gọi là **kernel trick**!

Kernel function: How to know?

- Làm sao để biết một hàm là hàm nhân hay không – mà không cần thực hiện các bước suy diễn (phân tích) cụ thể như trong ví dụ minh họa?

  → Làm sao để biết một hàm có phải là một tích vô hướng vectơ trong một không gian nào đó?

- Câu hỏi này được trả lời bằng **định lý Mercer (Mercer's theorem)**

  →Nằm ngoài phạm vi của bài giảng này!

Các hàm nhân thường dùng

- Đa thức:

$$K(\mathbf{x},\mathbf{z}) = \left(\langle \mathbf{x} \cdot \mathbf{z} \rangle + \theta\right)^d; \quad \text{trong đó : } \theta \in R, d \in N$$

- Gaussian RBF (Gaussian radial basis function)

$$K(\mathbf{x},\mathbf{z}) = e^{-\frac{\|\mathbf{x}-\mathbf{z}\|^2}{2\sigma}}; \quad \text{trong đó : } \sigma > 0$$

- Xích-ma (Sigmoidal)

$$K(\mathbf{x},\mathbf{z}) = \tanh\left(\beta\langle \mathbf{x} \cdot \mathbf{z} \rangle - \lambda\right) = \frac{1}{1 + e^{-(\beta\langle \mathbf{x}\cdot\mathbf{z}\rangle - \lambda)}}; \quad \text{trong đó : } \beta, \lambda \in R$$

Phân lớp bằng SVM: Các vấn đề

- SVM chỉ làm việc với không gian đầu vào là các số thực
  - → Đối với các thuộc tính định danh (nominal), cần chuyển các giá trị định danh thành các giá trị số
- SVM chỉ làm việc (thực hiện phân lớp) với 2 lớp
  - → Đối với các bài toán phân lớp gồm nhiều lớp, cần chuyển thành một tập các bài toán phân lớp gồm 2 lớp, và sau đó giải quyết riêng rẽ từng bài toán 2 lớp này
  - → Ví dụ: chiến lược "one-against-rest"
- Siêu phẳng phân tách (ranh giới quyết định phân lớp) xác định được bởi SVM thường khó hiểu đối với người dùng
  - Vấn đề (khó giải thích quyết định phân lớp) này càng nghiêm trọng, nếu các hàm nhân (kernel functions) được sử dụng
  - SVM thường được dùng trong các bài toán ứng dụng mà trong đó việc giải thích hoạt động (quyết định) của hệ thống cho người dùng không phải là một yêu cầu quan trọng

SVM: thư viện mở

- LibSVM:
  - http://www.csie.ntu.edu.tw/~cjlin/libsvm/
- Linear SVM for large datasets:
  - http://www.csie.ntu.edu.tw/~cjlin/liblinear/
  - http://www.cs.cornell.edu/people/tj/svm_light/svm_perf.html
- Scikit-learn in python:
  - http://scikit-learn.org/stable/modules/svm.html
- SVM*light*:
  - http://www.cs.cornell.edu/people/tj/svm_light/index.html

SVM: bài tập

- SVM khác gì so với k-NN?
- Số lượng support vectors trong trường hợp xấu nhất là bao nhiêu? Tại sao?
- Ý nghĩa của hệ số C trong SVM? So sánh vai trò của C trong SVM với vai trò của λ trong Ridge regression.

Tài liệu tham khảo

- B. Liu. *Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data*. Springer, 2006.
- C. J. C. Burges. *A Tutorial on Support Vector Machines for Pattern Recognition*. Data Mining and Knowledge Discovery, 2(2): 121-167, 1998.