Support Vector Machine
(Tạm dịch: Máy véc tơ hỗ trợ)

Trường Đại học Công nghệ Thông tin, Đại học Quốc gia Thành phố Hồ Chí Minh
Tài liệu nội bộ

Tháng 2 năm 2020

Tổng quan

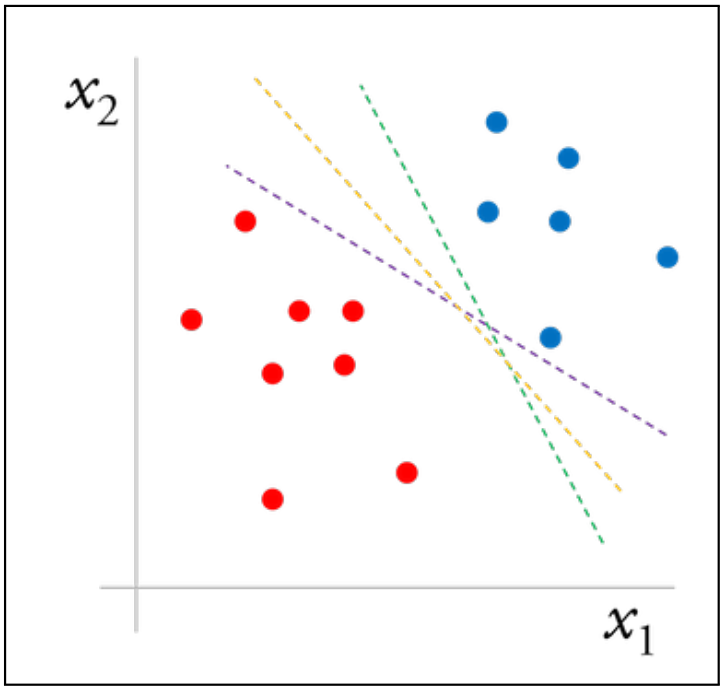
1. Giới thiệu về SVM
2. Cơ sở lý thuyết
3. Biên mềm (Soft margin) và Biên cứng (Hard margin)
4. Hồi quy sử dụng SVM (SVM Regression)
5. Thực hành với Python
6. Phương pháp Kernel SVM
7. SVM trực tuyến- online SVMs

Nội dung trình bày

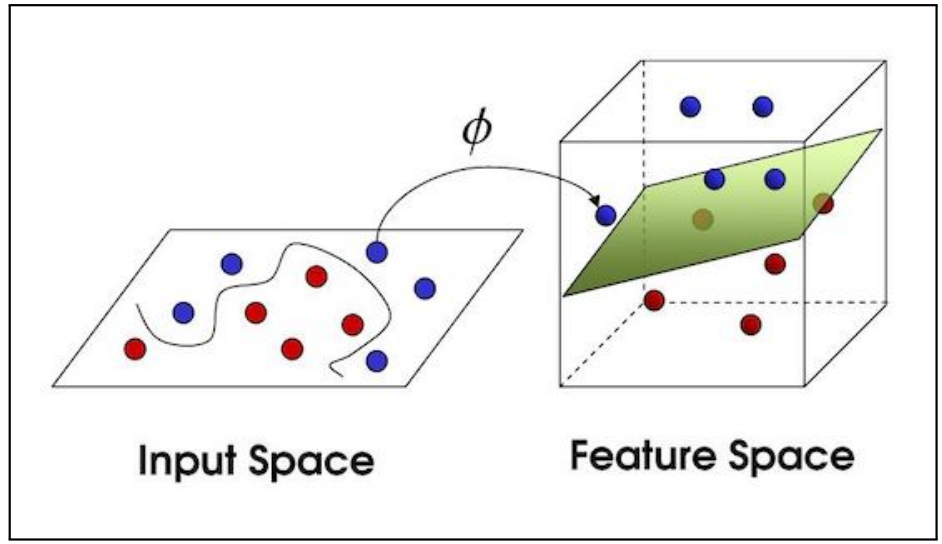
1. Giới thiệu về SVM

SVM?

Support Vector Machine (SVM) là một thuật toán thuộc nhóm Supervised Learning (học có giám sát) dùng để phân chia dữ liệu (classification) thành các nhóm riêng biệt (bài toán phân lớp).
Thuật toán được phát triển bởi nhà khoa học người Nga Vladimir Vapnik vào những năm 1960



Hình 1: Dữ liệu hai chiều với đường thẳng



Hình 2: tách dữ liệu 2 chiều bằng phép biến đổi

Vấn đề: Trong hình 1, đường thẳng phân tách lớp dữ liệu nào là tốt nhất

Nội dung trình bày

❷ Cơ sở lý thuyết

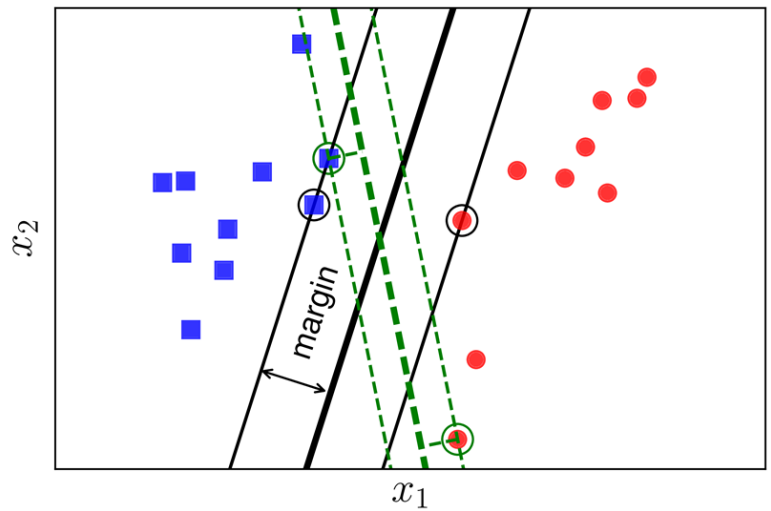
PP BIÊN CỰC ĐẠI
BÀI TOÁN CỰC TIỂU CÓ ĐIỀU KIỆN
PP NHÂN TỬ LAGRANGE

Để tìm hiểu cơ sở lý thuyết toán học của SVM, ta xét trường hợp số liệu hai chiều được tách rời tuyến tính.
Ghi chú:

- Đường thẳng: $ax + by + c = 0$ (trong không gian 2 chiều)
- Mặt phẳng: $ax + by + cz + d = 0$(trong không gian 3 chiều)
- Siêu phẳng (hyperplane): $w_1x_1 + w_2x_2 + ... + w_nx_n + b = 0$. Ta thường viết dạng véc tơ là $\mathbf{wx} + b = 0$

Biên (Margin)

Biên (hay lề - Margin) là khoảng cách giữa siêu phẳng đến 2 điểm dữ liệu gần nhất tương ứng với 2 phân lớp.



Hình 3: Độ rộng biên

Một cách trực quan, đường thẳng phân tách 2 tập nhãn là tốt nhất nếu như nó không quá gần một tập nào đó, nghĩa là độ rộng biên M là cực đại.Trong hình trên, phân lớp màu đen tốt hơn phân lớp màu xanh. Đây cũng là lý do vì sao SVM còn được gọi là pp phân loại biên cực đại (Maximum Margin Classifier).

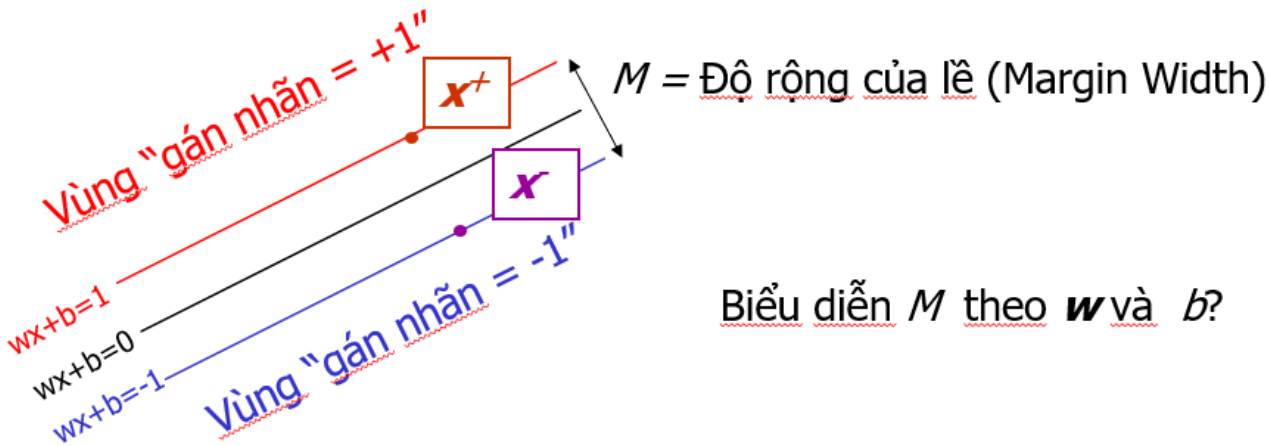
Tính M theo W và b

Từ 2 lớp nhãn thuộc tập huấn luyện, ta lấy một điểm bất kỳ và đặt (p+), (p-) là các ”mặt” đi qua mỗi điểm đó và song song với mặt phân lớp $Wx + b = 0$:

$(p+) = \{\mathbf{x}/\mathbf{wx} + b = +1\}$

$(p-) = \{\mathbf{x}/\mathbf{wx} + b = -1\}$

Để ý: **w** là ”pháp véc tơ” của mỗi ”mặt”.



Hình 4: Cực đại độ rộng biên

Hai mặt phẳng (p+) và (p-) tựa trên một số điểm thuộc lớp phân loại nên những điểm (vectors) này gọi là các điểm hỗ trợ (support vector). Đó là lý do tên phương pháp này là SVM (support vector machine).

Lấy một điểm (x-) trong lớp nhãn ”-” thuộc tập huấn luyện và thuộc (p-). Lấy một điểm (x+) thuộc (p+) mà gần (x-) nhất. Khi đó (x+)-(x-) sẽ cùng phương với vector **w** nên

$$(x+) - (x-) = \lambda \mathbf{w} \tag{1}$$

Sử dụng phương trình mặt (p+), (p-) và thay vào phương trình trên ta được

$$\lambda = \frac{2}{\mathbf{w^T w}} \tag{2}$$

Để ý là độ rộng lề $M = |(x+) - (x-)| = \lambda |\mathbf{w}| = \lambda \sqrt{\mathbf{w^T w}}$ nên suy ra

$$M = \frac{2}{\sqrt{\mathbf{w^T w}}} = \frac{2}{||\mathbf{w}||} \tag{3}$$

Cực đại độ rộng biên

Cực đại độ rộng $M = \frac{2}{\sqrt{\mathbf{w^T w}}} = \frac{2}{||\mathbf{w}||}$ tương đương với bài toán cực tiểu có điều kiện như sau (Ta chọn bình phương để việc tính đạo hàm được thuận lợi):

$$(\mathbf{w}, b) = \arg\min_{\mathbf{w},b} \frac{1}{2}||\mathbf{w}||^2 \tag{4}$$

$$\text{với điều kiện:} \quad 1 - y^{(i)}(\mathbf{w}^T\mathbf{x}^{(i)} + b) \leq 0, \forall i = 1, 2, \ldots, m \tag{5}$$

với $y^{(i)} = +/-1$ tương ứng với nhãn của hai phân lớp.
Bài toán tối ưu trên thuộc loại tối ưu bậc 2 (Quadratic Programming-QP), được giải bằng cách giải bài toán đối ngẫu Lagrange (Lagrange dual problem).
Xác định lớp (gán nhãn) cho một điểm dữ liệu mới: Sau khi tìm được mặt phân cách $\mathbf{w}^T\mathbf{x} + b = 0$, phân lớp (gán nhãn) của một điểm bất kỳ được xác định như sau:

$$\hat{y} = \text{class}(\mathbf{x}) = \text{sgn}(\mathbf{w}^T\mathbf{x} + b) \tag{6}$$

Trong đó hàm $sgn(.)$ là hàm xác định dấu, nhận giá trị 1 nếu đối số là không âm và -1 nếu ngược lại.

Tối ưu có ràng buộc- Phương pháp nhân tử Lagrange

Xét bài toán tối ưu có ràng buộc:

$$\begin{aligned} (\mathbf{w}, b) &= \arg\min_{\mathbf{w},b} \frac{1}{2}||\mathbf{w}||^2 \\ \text{với điều kiện:} \quad & 1 - y^{(i)}(\mathbf{w}^T\mathbf{x}^{(i)} + b) \leq 0, \forall i = 1, 2, \ldots, m \end{aligned}$$

Việc giải bài toán này trở nên phức tạp khi số chiều (số thuộc tính) n và số điểm dữ liệu m lớn. Khi đó người ta xét bài toán đối ngẫu Lagrage của nó. Ta xây dựng hàm Lagrange với $\alpha_i \geq 0$ được gọi là các nhân tử, đưa bài toán đã cho về bài toán cực trị:

$$L(\mathbf{w}, b; \alpha_i) = \frac{1}{2}||\mathbf{w}||^2 + \sum_{i=1}^{m} \alpha_i[1 - y^{(i)}(\mathbf{w}^T\mathbf{x}^{(i)} + b)] \tag{7}$$

Giải phương trình đạo hàm riêng của L theo các tham số

$$\frac{\partial \mathcal{L}(\mathbf{w}, b, \alpha)}{\partial \mathbf{w}} = \mathbf{w} - \sum_{i=1}^{m} \alpha_i y^{(i)} \mathbf{x}^{(i)} = 0 \Rightarrow \mathbf{w} = \sum_{i=1}^{m} \alpha_i y^{(i)} \mathbf{x}^{(i)} \tag{8}$$

$$\frac{\partial \mathcal{L}(\mathbf{w}, b, \alpha)}{\partial b} = -\sum_{i=1}^{m} \alpha_i y^{(i)} = 0 \tag{9}$$

Giải phương trình đạo hàm riêng của L theo các tham số, ta được hàm đối ngẫu Lagrange và cần cực đại hàm này với $\alpha_i \geq 0$ (CT 5-6, tr 170)

$$L(\alpha) = -\sum_{i=1}^{m} \alpha_i + \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j y^{(i)} y^{(j)} \mathbf{x}^{(i)T} \mathbf{x}^{(j)} \tag{10}$$

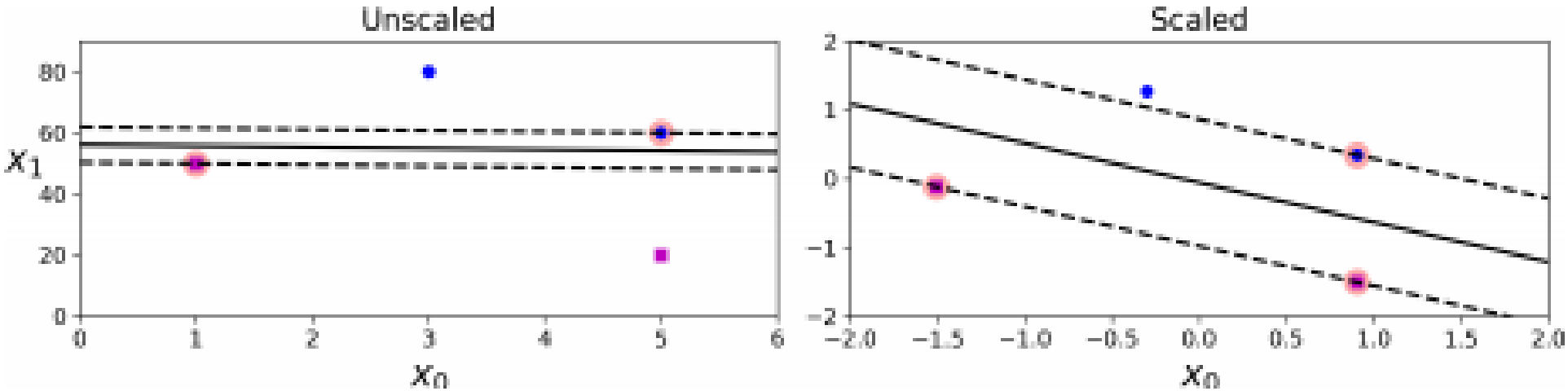
Dùng tool python để tính được α_i, khi đó ta tính các hệ số phương trình đường phân tách như sau (CT5-7 tr 171):

$$\mathbf{w} = \sum_{i \in \mathcal{S}} \alpha_i y^{(i)} \mathbf{x}^{(i)} \tag{11}$$

$$b = \frac{1}{n_{\mathcal{S}}} \sum_{i \in \mathcal{S}} (y^{(i)} - \mathbf{w}^T \mathbf{x}^{(i)}) = \frac{1}{n_{\mathcal{S}}} \sum_{i \in \mathcal{S}} \left(y^{(i)} - \sum_{j \in \mathcal{S}} \alpha_j y^{(j)} \mathbf{x}^{(j)T} \mathbf{x}^{(i)} \right) \tag{12}$$

với $n_{\mathcal{S}}$ là số điểm hỗ trợ, còn S là các điểm i ứng với $\alpha_i \neq 0$ (S= Support vector). Để ý là m và n có thể lớn, nhưng $n_{\mathcal{S}}$ lại khá nhỏ - Hãy kiểm chứng bằng CT python.

SVM nhạy cảm với phép biến đổi độ lớn (scale) thuộc tính. Hình dưới cho thấy hình bên phải cho phân lớp tốt hơn.



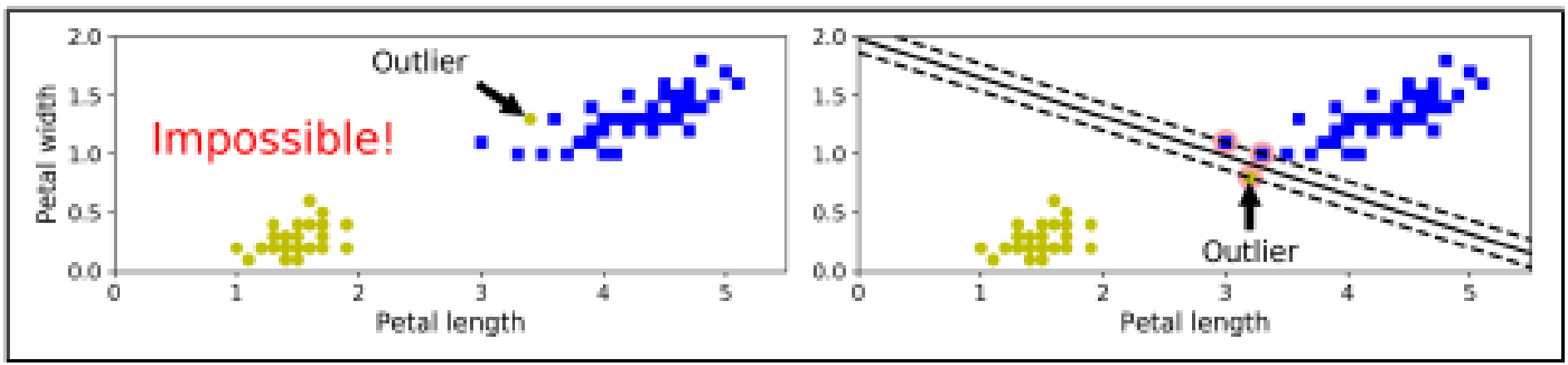
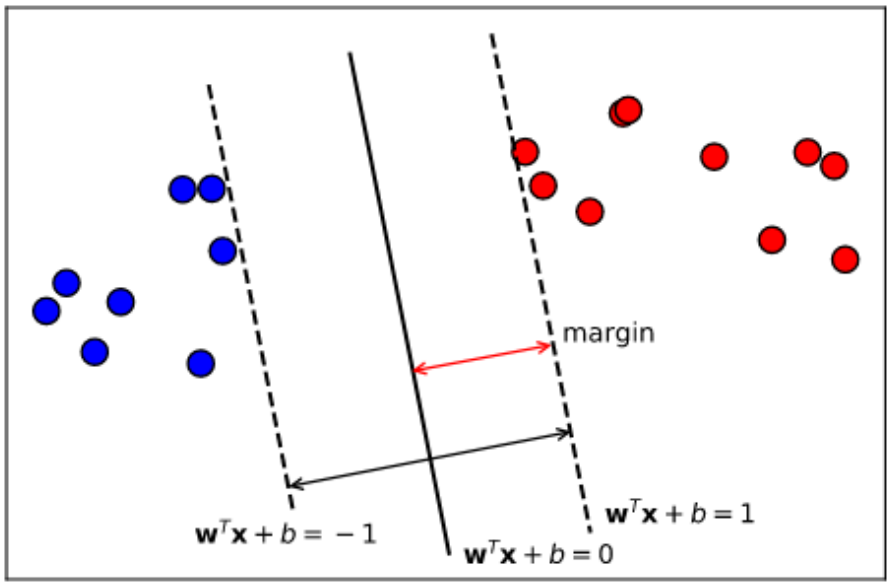


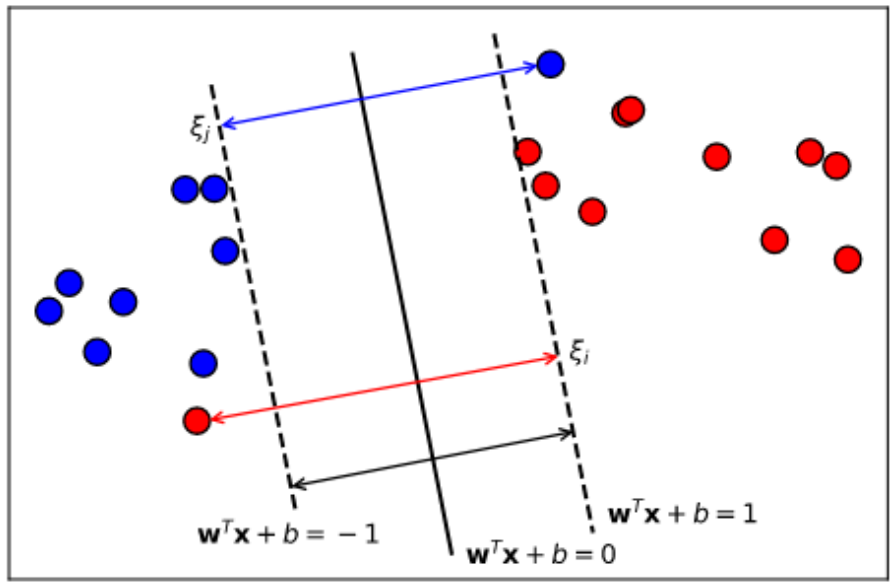
Figure 5-3. Hard margin sensitivity to outliers

Nội dung trình bày

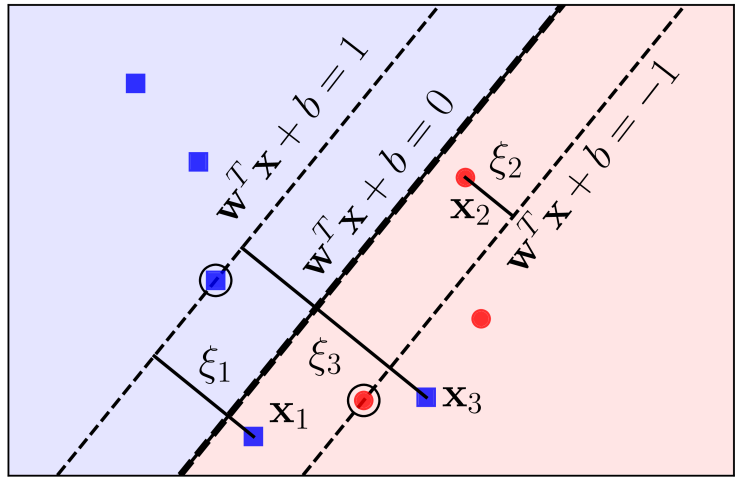
❸ Biên mềm (Soft margin) và Biên cứng (Hard margin)



Phương pháp biên cứng
(Hard margin)



Phương pháp biên mềm
(Soft margin)



Đưa thêm biến *Slack* để đánh giá các điểm "nhiễu"

Hình 5

Với trường hợp có nhiễu, điều kiện theo phương pháp biên cứng không thỏa được nên ta đưa thêm vào một biến phụ $\xi_i \geq 0$

$$1 - y^{(i)}(\mathbf{w}^T\mathbf{x}^{(i)} + b) \leq \xi_i, \forall i = 1, 2, \ldots, m \tag{13}$$

Bài toán tối ưu trở thành

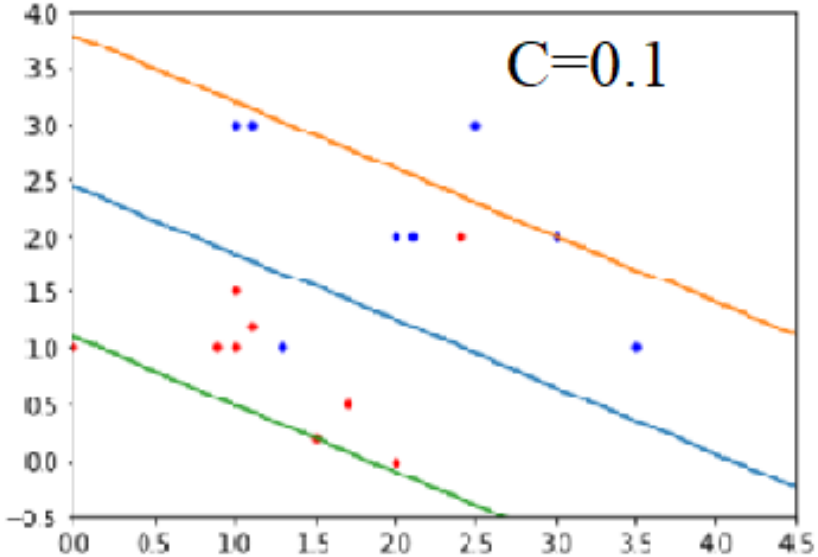
$$(\mathbf{w}, b) = \arg\min_{\mathbf{w},b} \left[\frac{1}{2}||\mathbf{w}||^2 + C\sum_{i=1}^{m} \xi_i \right] \quad (14)$$

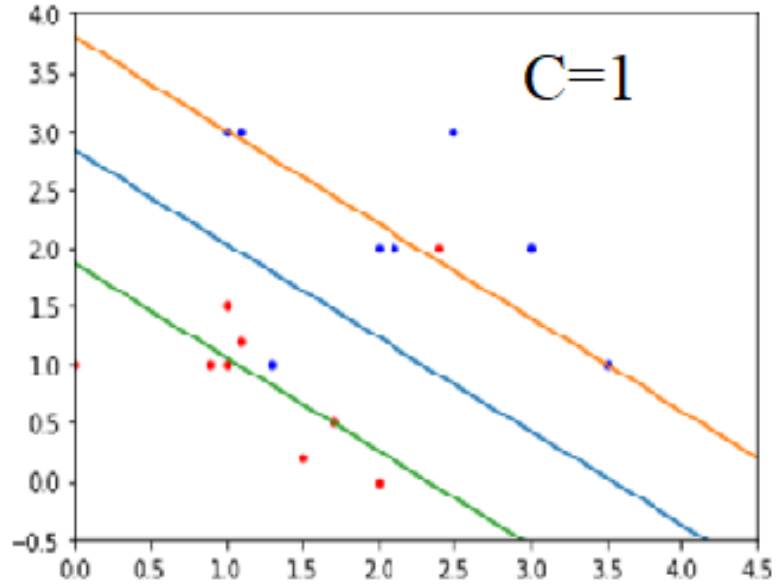
$$\text{với đk:} \quad 1 - y^{(i)}(\mathbf{w}^T\mathbf{x}^{(i)} + b) - \xi_i \leq 0, \xi_i \geq 0, \forall i = 1, 2, \ldots, m \quad (15)$$

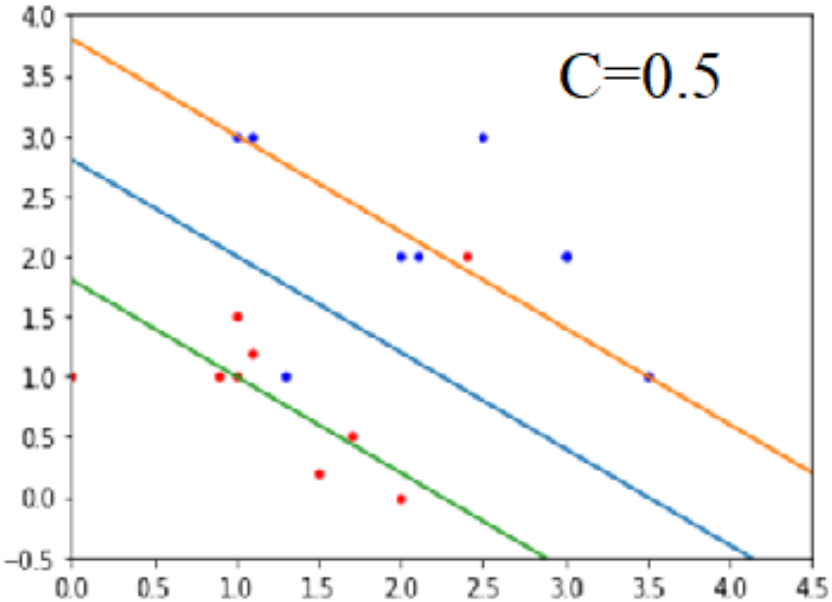
Tham số C được gọi là "siêu tham số" (hyperparameter). Độ lớn tham số C cho ta ưu tiên mức độ tối ưu giữa hai thành phần:

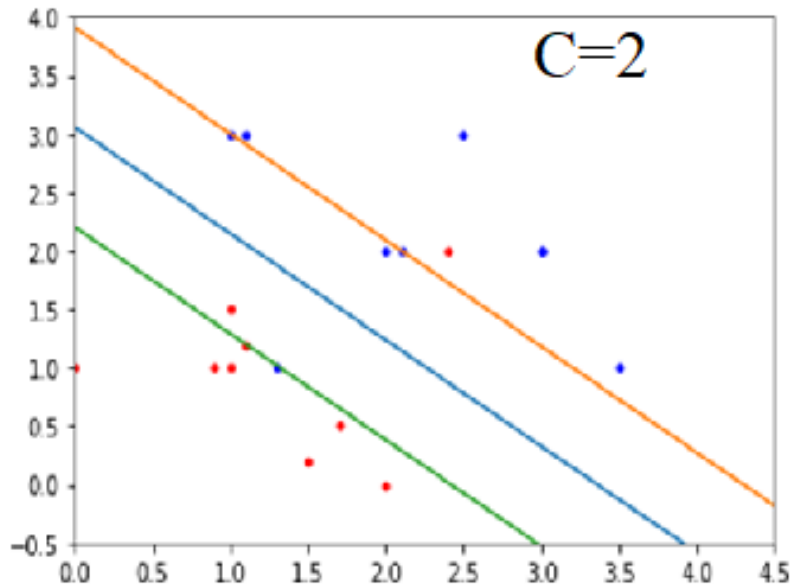
- C nhỏ: Ưu tiên cực tiểu thành phần thứ nhất
- C lớn: ưu tiên cực tiểu thành phần thứ 2. Nếu C quá lớn và hai tập tách rời tuyến tính thì hàm cực tiểu khi tổng $\xi_i = 0$, nghĩa là thuật toán trở thành thuật toán biên cứng SVM.
- Nếu mô hình SVM quá khớp (overfitting), ta có thể điều chỉnh giảm tham số C (khi đó độ rộng biên sẽ tăng lên).

Nhận xét: SVM không cho ta biết xác suất thuộc phân lớp của một điểm dữ liệu.









C càng lớn thì biên càng hẹp

Hình 6

Nội dung trình bày

4. Hồi quy sử dụng SVM (SVM Regression)

Thuật toán SVM có thể sử dụng xây dựng đường hồi quy. Thay vì cố gắng xác định ”đường đi rộng nhất” giữa 2 lớp, đường hồi quy SVM cố gắng càng gần các điểm càng tốt trong giới hạn của biên. Độ rộng của đường được kiểm soát bởi siêu tham số ε.
Để xác định đường hồi quy SVM, ta cực tiểu hàm mất mát với điều kiện như sau:

$$-\varepsilon \leq y_i - wx_i - b \leq \varepsilon \tag{16}$$

```
from sklearn.svm import LinearSVR

svm_reg = LinearSVR(epsilon=1.5)
svm_reg.fit(X, y)
```

Hình 7

Thực hành: Chạy kiểm tra các đoạn chương trình ở trang 165, 166.

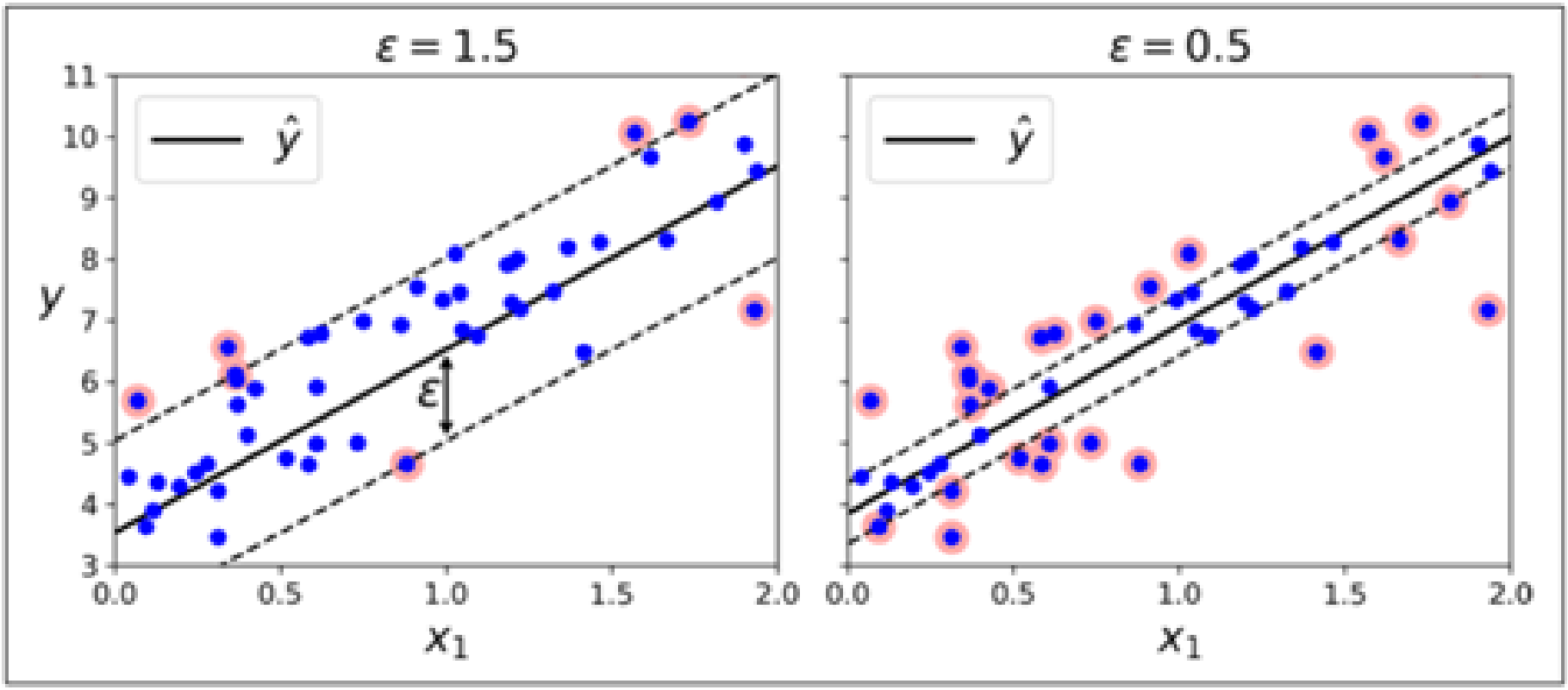
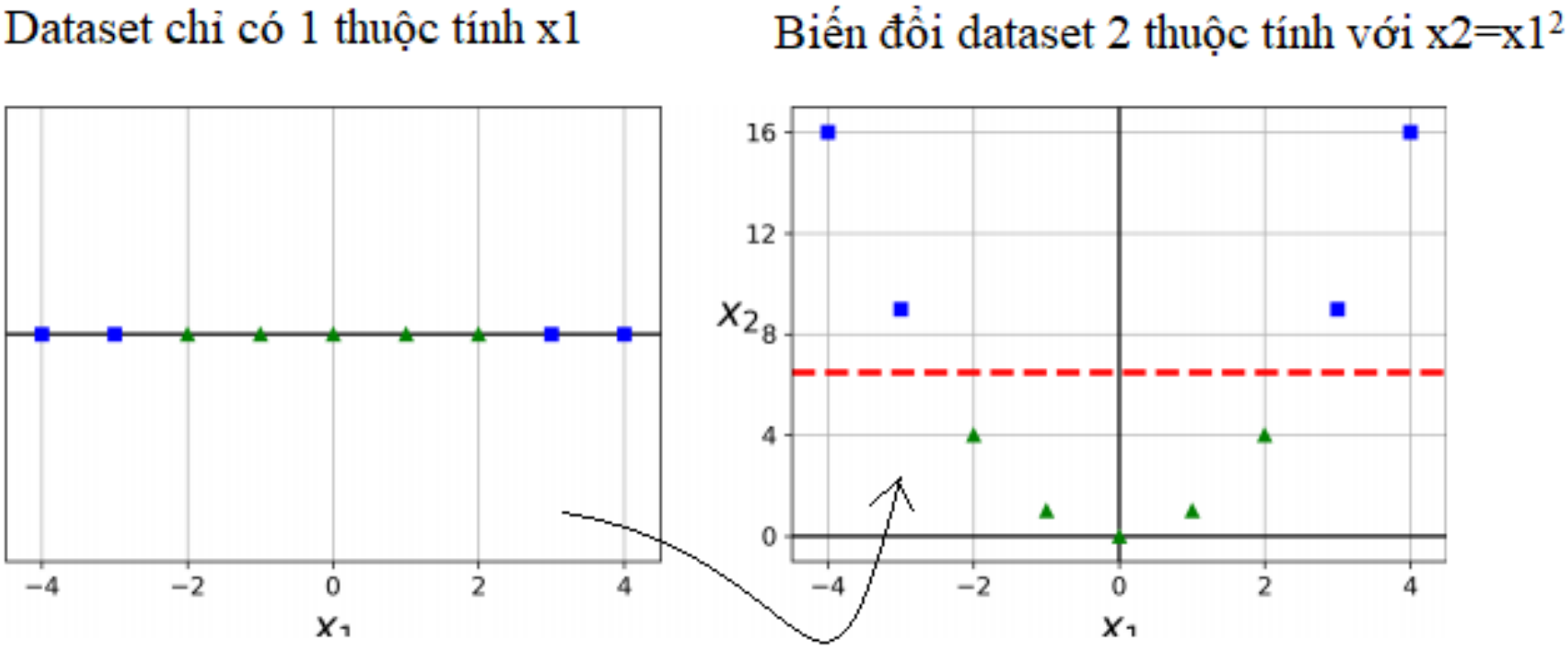


Figure 5-10. SVM Regression

Hình 8

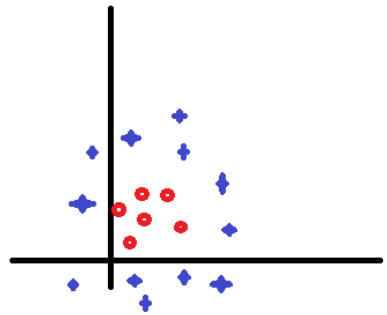
Phân loại SVM phi tuyến

Trong thực tế ta cũng gặp dữ liệu không ”gần với” tách rời tuyến tính, nghĩa là đường phân lớp có thể là một đường cong. Khi đó ta có thể thêm thuộc tính, chẳng hạn như thuộc tính đa thức, để có được tập tách rời tuyến tính (xem hình dưới), hoặc ta có thể sử dụng phép biến đổi đa thức như bài thực hành số 3 bên dưới.



Hình 9

Phân loại SVM phi tuyến



Hình 10

Ghi chú:

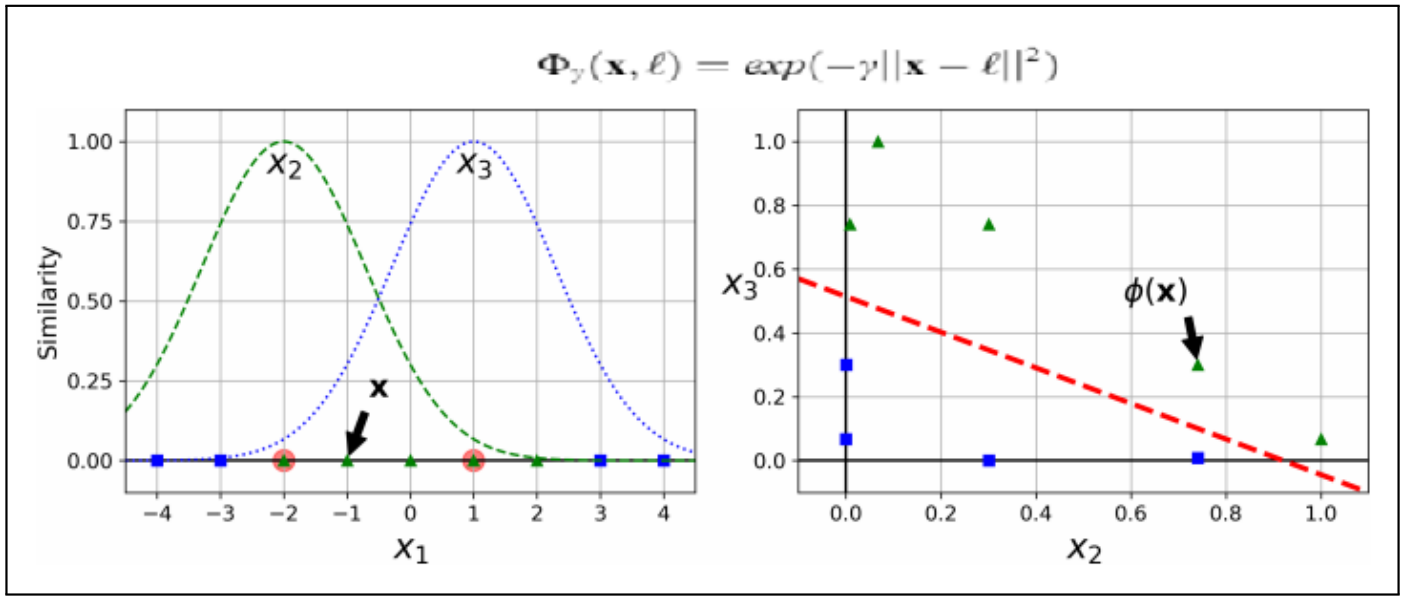
- Hàm đa thức bậc 2 $\Phi(x_1, x_2) = (x_1^2, \sqrt{2}x_1x_2, x_2^2)^T$
- Hàm biến đổi đa thức (kernel polynomial)

$$K(x, w; \gamma, r, d) = \Phi(x)^T\Phi(w) = (\gamma x^T w + r)^d$$

- Dữ liệu như trong hình trên có thể gặp trong điều trị bệnh: Lượng thuốc điều trị vừa đủ thì người bệnh sẽ được chữa khỏi, ngược lại, quá liều hoặc không đủ liều thì người bệnh không khỏi được.
- Trong lớp sklearn.svm.SVC, đặt kernel = 'poly', degree=?, C=?

Kernel Gauss RBF

Một kỹ thuật quan trọng để giải bài toán phân loại có đường phân loại phi tuyến là thêm thuộc tính được tính từ hàm đồng dạng (similarity function) dùng để đo độ tương đồng của mỗi mẫu dữ liệu với một điểm làm dấu (landmark). Chẳng hạn, với dữ liệu một chiều như hình dưới, chọn 2 điểm làm dấu là $\ell_1 = -2, \ell_2 = 1$, và chọn hàm đồng dạng RBF với $\gamma = 0.3$. Khi đó trong không gian thuộc tính mới, dữ liệu tách rời tuyến tính.



Hình 11: Thêm thuộc tính đồng dạng

Công thức hàm đồng dạng Gauss RBF có dạng:

$$\varphi_\gamma(x, \ell) = exp(-\gamma||x - \ell||^2), \ell = \text{điểm làm dấu} \quad (17)$$

Tính thuộc tính mới:

- Xét điểm $x = -1$. Ta có 2 điểm làm dấu là $\ell_1 = -1, \ell_2 = 1$. Thay vào công thức trên:

$$x_2 = exp(-0.3 * |-1-(-2)|^2 \approx 0.74$$

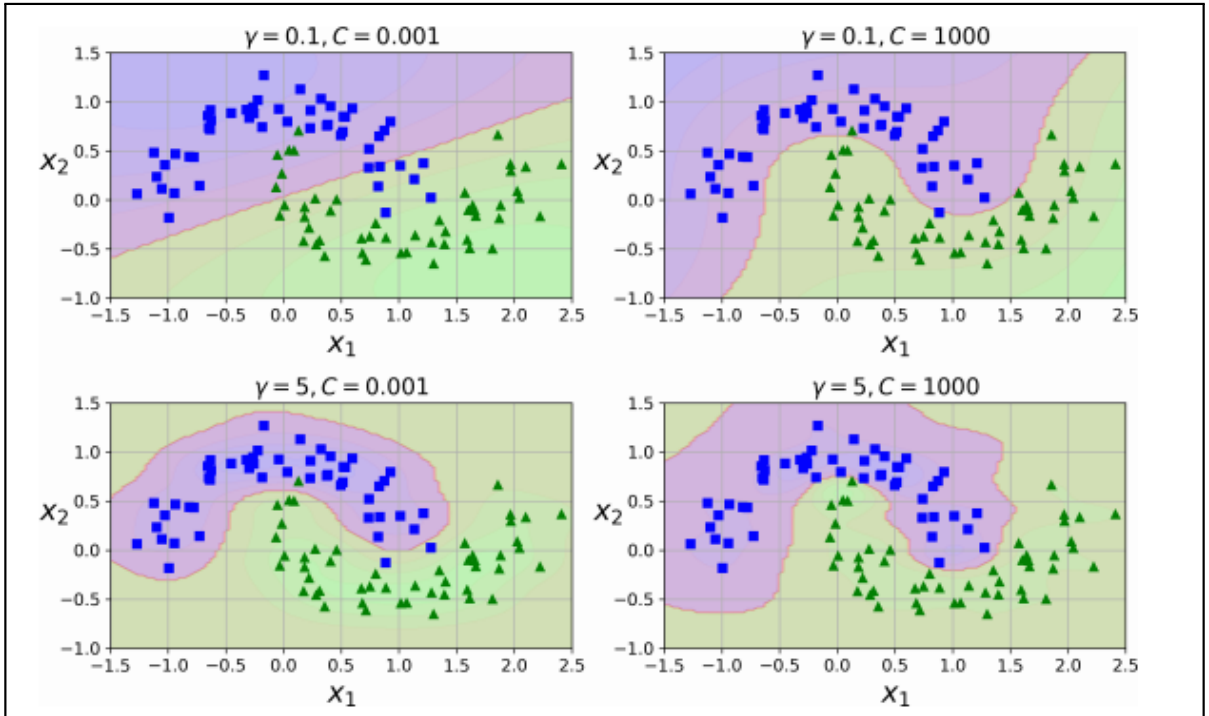
$$x_3 = exp(-0.3 * |-1-1|^2) \approx 0.3$$

- Làm tương tự cho các điểm dữ liệu còn lại ta được dữ liệu tách rời tuyến tính như hình bên phải.
- Trong sklearn.svm.SVC ta đặt kernel = 'rbf' γ=?, C=?

Để ý: γ càng lớn thì đỉnh chuông càng nhọn, khoảng ảnh hưởng của mỗi mẫu sẽ nhỏ, do đó biên phân lớp sẽ "gắt"(suy biến-irregular), nghĩa là mô hình quá khớp. Như vậy, nếu mô hình quá khớp thì ta điều chỉnh giảm tham số này (tương tự tham số C).

CHỌN ĐIỂM LÀM DẤU THẾ NÀO?

Ta chọn m mẫu trong tập huấn luyện là các điểm làm dấu. Như vậy với một mẫu $\mathbf{x}$ cho trước và hàm đồng dạng $\Phi_\gamma(.,.)$, ta có thuộc tính mới là
$[f_i(x) = \Phi_\gamma(\mathbf{x}, \mathbf{x}^{(i)}), i = 1...m]^T$
Cụ thể, giả sử tập huấn luyện $(x^{(i)}, y^{(i)}), i = 1...m$
Ta chọn các điểm làm dấu: $\ell^{(i)} = x^{(i)}, i = 1...m$
Biến đổi mỗi điểm dữ liệu $(x^{(i)}, y^{(i)})$: $x^{(i)} \rightarrow f = [f_1(x^{(i)}), ...f_m(x^{(i)})]^T$, để ý $f_i(x^{(i)}) = 1$ Như vậy, với dữ liệu $\mathbf{x}$, ta tính được thuộc tính
$f = [1, f_1, ..., f_m]^T \in R^{m+1}$ và gán nhãn "y=1" nếu như $\theta^T f \geq 0$, θ là tham số của mô hình, được xác định qua việc cực tiểu hàm mất mát tương ứng.



Nội dung trình bày

5 Thực hành với Python

Thực hành với lệnh pipeline trang 158

Bài thực hành 1.

- Chạy lại đoạn CT trang 158
- Hãy gán nhãn các các điểm (1.5, 1.7), (5.5, 1.6)

Sử dụng hàm sklearn.svm.SVC
TUYẾN TÍNH

Bài thực hành 2. Cho tập dữ liệu huấn luyện X1 = [[1,3], [3,2], [3.5,1], [3,2], [2, 2],[2.5,3]] được gán nhãn y1 = [1, 1, 1, 1, 1,1] và X2 = [[0,1], [1,1], [1,1.5], [2,0]] được gán nhãn y2 = [-1, -1, -1, -1]

i Hãy vẽ đồ thị phân tán các điểm X1 màu xanh và X2 màu đỏ và nhận xét về sự tách rời tuyến tính giữa 2 tập này.

ii Hãy sử dụng SVC trong sklearn.svm để xây dựng mô hình phân loại. HD: Sử dụng đoạn code sau:

```
from sklearn.svm import SVC
clf = SVC(kernel='linear', C=1E10)
clf.fit(X, y)
print(clf.support_vectors_) # cho tọa độ của các véc tơ hỗ trợ
```

iii Vẽ đường phân tách 2 tập và các đường thẳng biên wx+b=+-1

iv Gán tham số C=1, chạy lại toàn bộ chương trình python và nhận xét.

Sử dụng hàm sklearn.svm trong thư viện sklearn- ĐA THỨC

Bài thực hành 3. Sử dụng tập dữ liệu (dataset) moons trang 159

i Hãy vẽ đồ thị phân tán của tập dữ liệu (HD: sử dụng plt.scatter(X[:, 0], X[:, 1], c=y, s=50, cmap='brg')). Nhận xét về sự tách rời dữ liệu dựa vào tập nhãn.

ii Chạy đoạn code ở trang 158 (nếu có lỗi thì hãy hiệu chỉnh). Sử dụng mô hình tìm được, hãy gán nhãn cho các điểm A=(1.5, -0.7), B=(1,0.5).

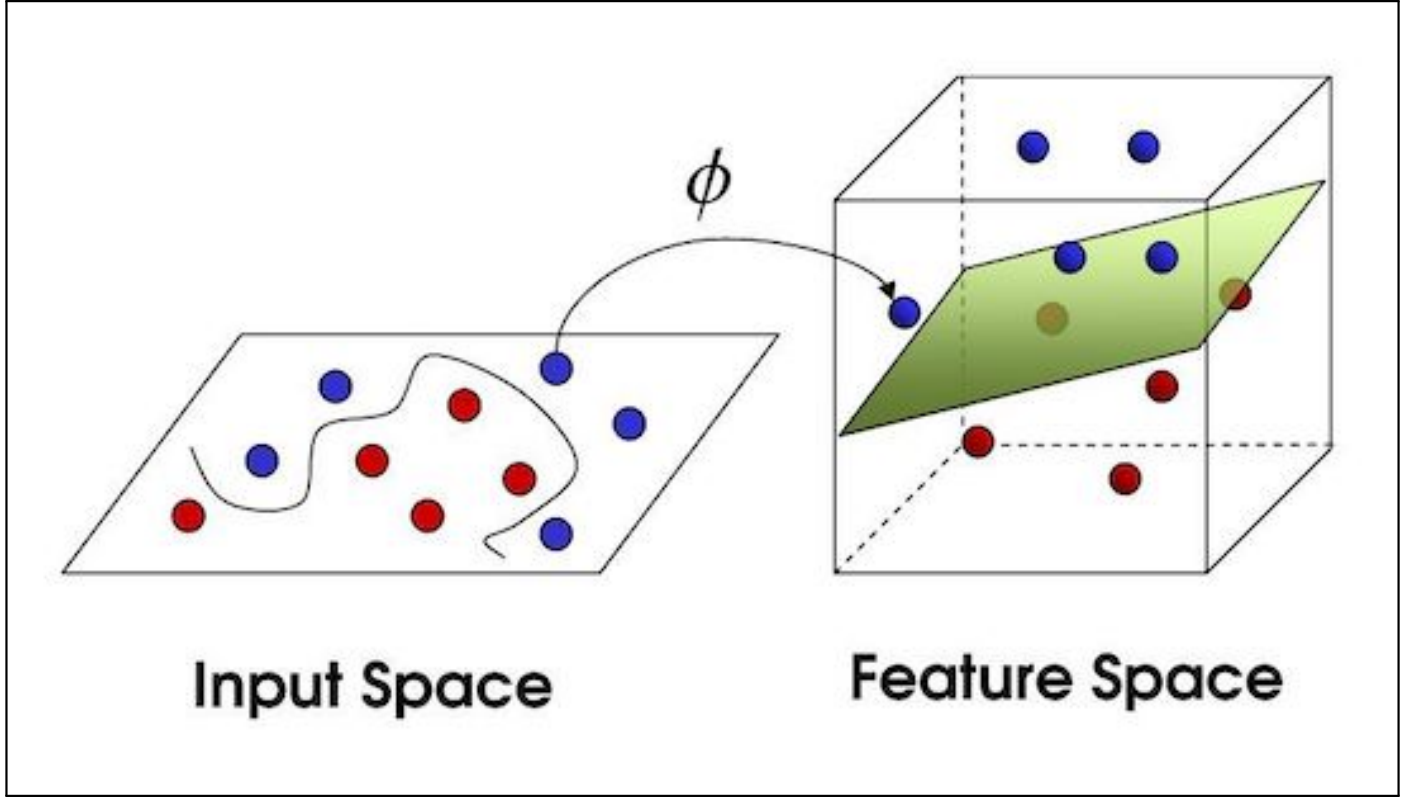
iii Sử dụng đoạn code sau để vẽ lại hình ở trang 160

```
axes=[-1.5, 2.5, -1, 1.5]
x0s = np.linspace(axes[0], axes[1], 100)
x1s = np.linspace(axes[2], axes[3], 100)
x0, x1 = np.meshgrid(x0s, x1s)
X = np.c_[x0.ravel(), x1.ravel()]
y_pred = polynomial_svm_clf.predict(X).reshape(x0.shape)
y_decision = polynomial_svm_clf.decision_function(X).reshape(x0.shape)
plt.contourf(x0, x1, y_pred, cmap=plt.cm.brg, alpha=0.2)
plt.contourf(x0, x1, y_decision, cmap=plt.cm.brg, alpha=0.1)
plt.show()
```

iv Sửa lại bậc của đa thức là 1, là 2 (degree=1; 2) và chạy lại toàn bộ chương

Nội dung trình bày

6 Phương pháp Kernel SVM



Hình 13: Tách phân lớp dữ liệu 2 chiều bằng phép biến đổi

PP Kernel SVM là việc đi tìm một hàm số biến đổi dữ liệu x từ không gian thuộc tính (feature) ban đầu thành dữ liệu trong một không gian mới bằng hàm số $\Phi(x)$

Equation 5-8. Second-degree polynomial mapping

$$\phi(\mathbf{x}) = \phi\left(\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}\right) = \begin{pmatrix} x_1^2 \\ \sqrt{2}\,x_1 x_2 \\ x_2^2 \end{pmatrix}$$

Hình 14

Equation 5-9. Kernel trick for a 2nd-degree polynomial mapping

$$K(\mathbf{a},\mathbf{b}) = \phi(\mathbf{a})^T\phi(\mathbf{b}) = \begin{pmatrix} a_1^2 \\ \sqrt{2}\,a_1 a_2 \\ a_2^2 \end{pmatrix}^T \begin{pmatrix} b_1^2 \\ \sqrt{2}\,b_1 b_2 \\ b_2^2 \end{pmatrix} = a_1^2 b_1^2 + 2a_1 b_1 a_2 b_2 + a_2^2 b_2^2$$

$$= (a_1 b_1 + a_2 b_2)^2 = \left(\begin{pmatrix} a_1 \\ a_2 \end{pmatrix}^T \begin{pmatrix} b_1 \\ b_2 \end{pmatrix}\right)^2 = \left(\mathbf{a}^T\mathbf{b}\right)^2$$

Hình 15

Định lý Mercer

Nếu hàm số thực hai biến $K(.,.)$ thỏa các điều kiện:

(i) K liên tục trên $\mathcal{R}^2$

(ii) K đối xứng: $K(a,b) = K(b,a), \forall a,b$

(iii) K nửa xác định dương:
$\sum_{i=1}^{N}\sum_{j=1}^{N} k(\mathbf{x}_j,\mathbf{x}_i)c_i c_j \geq 0, \quad \forall c_i \in \mathbb{R}, i = 1,2,\ldots,N$

Khi đó tồn tại hàm biến đổi $\Phi(x)$ biến x sang không gian khác và thỏa $K(a,b) = \Phi(a)^T\Phi(b)$.

Không gian đích có thể là không gian hữu hạn chiều hoặc là không gian vô hạn chiều (như hàm Gauss kernel- Radial Basic fuction (RBF)).

Trong thực hành, người ta thấy rằng ta vẫn có thể dùng hàm Kernel K ngay cả khi ta không biết hàm φ, hoặc dùng cả hàm K nhưng không thỏa định lý Mercer (như hàm sigmoid kernel)

Công thức ước lượng hệ số tự do b

$$
\begin{aligned}
b &= \frac{1}{N_{\mathcal{S}}} \sum_{i \in \mathcal{S}} \left(y^{(i)} - \sum_{j \in \mathcal{S}} \alpha_j y^{(j)} \Phi(\mathbf{x}^{(j)})^T \Phi(\mathbf{x}^{(i)}) \right) \\
&= \frac{1}{N_{\mathcal{S}}} \sum_{i \in \mathcal{S}} \left(y^{(i)} - \sum_{j \in \mathcal{S}} \alpha_j y^{(j)} K(x^{(j)}, x^{(i)}) \right)
\end{aligned}
\tag{18}
$$

Equation 5-10. Common kernels

$$\begin{aligned}
\text{Linear:} \quad & K(\mathbf{a}, \mathbf{b}) = \mathbf{a}^T \mathbf{b} \\
\text{Polynomial:} \quad & K(\mathbf{a}, \mathbf{b}) = \left(\gamma \mathbf{a}^T \mathbf{b} + r\right)^d \\
\text{Gaussian RBF:} \quad & K(\mathbf{a}, \mathbf{b}) = \exp\left(-\gamma \| \mathbf{a} - \mathbf{b} \|^2\right) \\
\text{Sigmoid:} \quad & K(\mathbf{a}, \mathbf{b}) = \tanh\left(\gamma \mathbf{a}^T \mathbf{b} + r\right)
\end{aligned}$$

Hình 16

Equation 5-11. Making predictions with a kernelized SVM

$$\begin{aligned}
h_{\hat{\mathbf{w}}, \hat{b}}\left(\phi\left(\mathbf{x}^{(n)}\right)\right) &= \hat{\mathbf{w}}^T \phi\left(\mathbf{x}^{(n)}\right) + \hat{b} = \left(\sum_{i=1}^{m} \hat{\alpha}^{(i)} t^{(i)} \phi\left(\mathbf{x}^{(i)}\right)\right)^T \phi\left(\mathbf{x}^{(n)}\right) + \hat{b} \\
&= \sum_{i=1}^{m} \hat{\alpha}^{(i)} t^{(i)} \left(\phi\left(\mathbf{x}^{(i)}\right)^T \phi\left(\mathbf{x}^{(n)}\right)\right) + \hat{b} \\
&= \sum_{\substack{i=1 \\ \hat{\alpha}^{(i)} > 0}}^{m} \hat{\alpha}^{(i)} t^{(i)} K\left(\mathbf{x}^{(i)}, \mathbf{x}^{(n)}\right) + \hat{b}
\end{aligned}$$

Hình 17

Nội dung trình bày

7. SVM trực tuyến- online SVMs

SVM trực tuyến

- Thuật ngữ "học trực tuyến-online learning" hiểu theo nghĩa mô hình được cập nhật tức thì khi ngay khi có mẫu mới.
- Xét lại ràng buộc trong bài toán SVM tuyến tính:

$$1 - y^{(i)}(\mathbf{w}^T\mathbf{x}^{(i)} + b) - \xi_i \leq 0, \xi_i \geq 0, \forall i = 1, 2, \ldots, m \tag{19}$$

được viết thành

$$\xi_i \geq max[0, (1 - y^{(i)}(\mathbf{w}^T\mathbf{x}^{(i)} + b), \forall i = 1, 2, \ldots, m \tag{20}$$

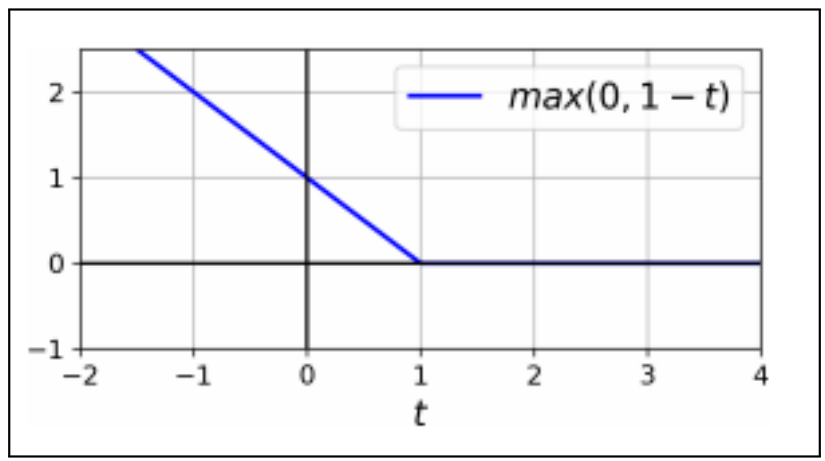
Có thể thấy nếu $(\mathbf{w}, b, \boldsymbol{\xi})$ là nghiệm của bài toán tối ưu

$$(\mathbf{w}, b) = \arg\min_{\mathbf{w},b} \left[\frac{1}{2}||\mathbf{w}||^2 + C\sum_{i=1}^{m}\xi_i\right] \tag{21}$$

thì $\xi_i = \max[0, (1 - y^{(i)}(\mathbf{w}^T\mathbf{x}^{(i)} + b), \forall i = 1, 2, \ldots, m$ Vì vậy hàm mất mát của thuật toán SVM tuyến tính trở thành (CT 5-13, tr174)

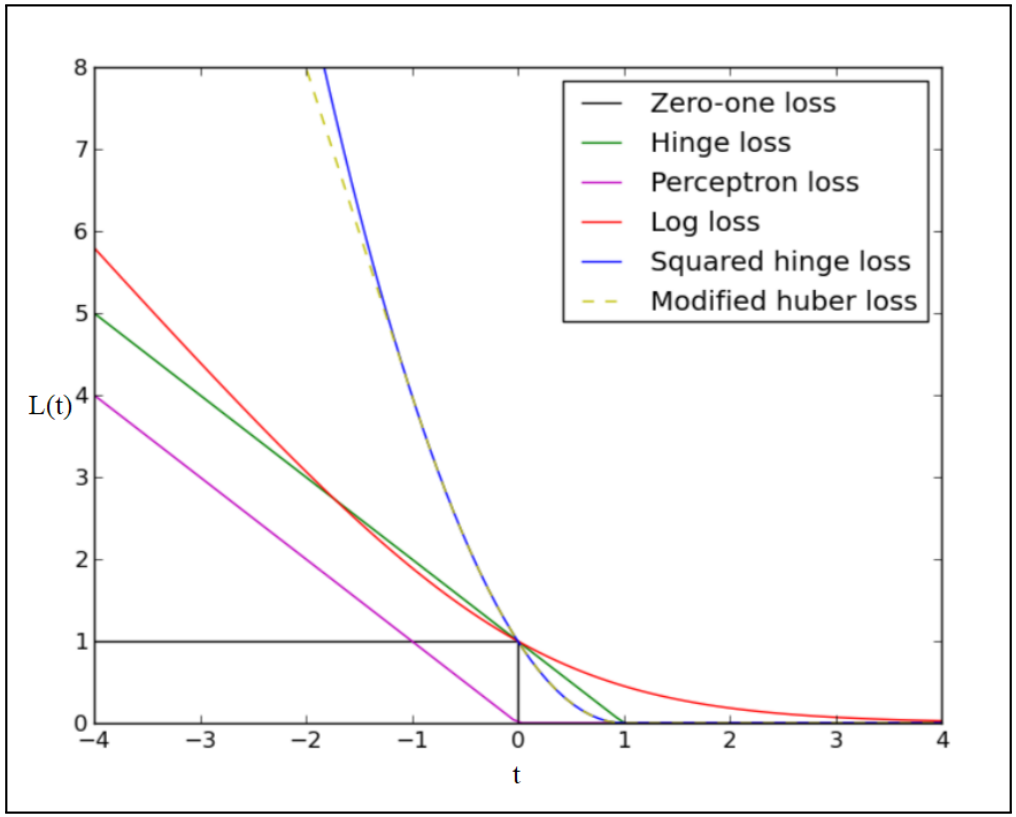
$$(\mathbf{w}, b) = \arg\min_{\mathbf{w},b} \left[\frac{1}{2}||\mathbf{w}||^2 + C\sum_{i=1}^{m} max[0, (1 - y^{(i)}(\mathbf{w}^T\mathbf{x}^{(i)} + b)\right] \tag{22}$$

- Người ta thấy rằng bộ phân loại SVM tuyến tính với thuật toán tối ưu SGD (GD ngẫu nhiên) áp dụng cho hàm mất mát hội tụ khá chậm so với thuật toán sử dụng quy hoạch bậc 2 (QP)
- Hàm mất mát Hinge: được định nghĩa là $h(t) = max(0, 1 - t)$. Hàm này không khả vi tại $t = 1$, $h'(t) = -1$ nếu $t < 1$ và $= 0$ nếu $(t > 1)$. Do vậy, khi dùng GD, ta gán đạo hàm của $h(t)$ tại $t = 1$ một giá trị bất kỳ trong khoảng $(-1, 0)$.



Hình 18: Đồ thị hàm mất mát Hinge

Các hàm mất mát cơ bản dành cho bài toán phân lớp nhị phân



Hình 19: Đồ thị một số hàm mất mát hay dùng

Bài tập

1. Ý tưởng cơ bản của SVM là gì? Ý tưởng của thuật toán biên mềm SVM. Nêu ý nghĩa siêu tham số C trong bải toán cực tiểu hàm mất mát.
2. Sử dụng SVM thư viện sklearn để xây dựng mô hình phân loại cho tập huấn luyện có dữ liệu cho trong data-demo-svm.xls. Thay đổi độ lớn tham số C và nhận xét.
3. Ý tưởng của hàm kernel $K(.,.)$ là gì? Khi nào ta áp dụng hàm kernel? Ta có cần biết biểu thức của hàm $\Phi(x)$ không?
4. Cho điểm dữ liệu trong không gian hai chiều $\mathbf{x} = [x_1, x_2]^T$ và hàm biến đổi sang không gian 5 chiều: $\Phi(\mathbf{x}) = [1, \sqrt{2}x_1, \sqrt{2}x_2, x_1^2, \sqrt{2}x_1x_2, x_2^2]^T$. Hãy tính hàm kernel $K(a, b)$
5. Giả sử bạn dùng bộ phân loại SVM với hàm kernel RBF cho tập huấn luyện và thấy mô hình phân loại chưa tốt. Để cải thiện, bạn sẽ giảm hay tăng tham số γ trong công thức hàm kernel, tham số C trong hàm mất mát.
6. Làm bài tập 9, 10 trang 175, 176 (Phần đánh giá độ chính xác có thể để sau)

Nhà khoa học người Nga Vladimir Vapnik

Hình 20

- 1936/12/6- được sinh ra ở Liên bang Soviet
- 1958- TN đại học, 1960's Bảo vệ TS về SVM - nhưng không ai để ý KQ này.
- 1990- Di cư đến Mỹ, 1992- gửi 3 bài báo cho tạp chí NIPS (Neural Information Processing Systems) về SVM nhưng đều không vượt qua vòng phản biện.
- 1992-1994 PTN Bell quan tâm đến các nhận dạng chữ viết tay, thuật toán của Vapnik đã được sử dụng và cho kết quả tốt.
- Cuốn sách ông cầm trên tay được trích dẫn hơn 60.000 lần -wow

Tài liệu tham khảo

- Hands-on Machine Learning with Scikit-Learn, Keras & TensorFlow, 2nd Edition của tác giả Aurélien Géron.
- https://www.kaggle.com/bbose71/svm-non-linear-classification
- https://machinelearningcoban.com/