Học Máy 2

Support Vector Machine

Nguyễn Văn Minh Khánh - Trịnh Quốc Dân

School of Engineering and Technology Hue University

Ngày 1 tháng 6 năm 2024

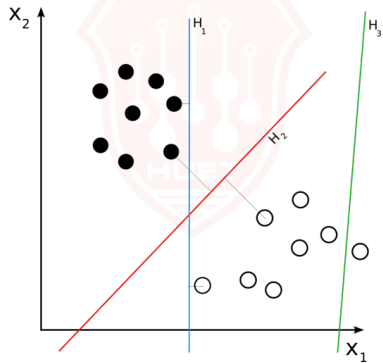
Tổng quan

1.1 Giới thiệu

- Mô hình Support Vector Machine - SVM là một mô hình máy học thuộc nhóm Supervised Learning được sử dụng cho các bài toán **Classification** (Phân lớp) và **Regression** (Hồi quy).
- Ta còn có thể phân loại mô hình này vào loại **Linear Model** (mô hình Tuyến tính).

1.1 Giới thiệu

Nền tảng của mô hình SVM nhằm giải quyết vấn đề cơ bản nhất là **phân loại hai lớp dữ liệu**. Đối với một tập dữ liệu ban đầu gồm hai lớp được tách biệt rõ ràng, câu hỏi đặt ra là nên chọn **đường nào để đảm bảo khả năng tổng quát tốt nhất**?

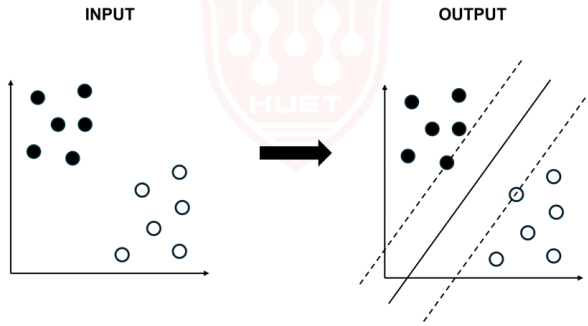


Phân chia 2 tập dữ liệu bằng svm - src:iostrean.co

1.2 Phát biểu bài toán cho mô hình SVM

Input: Bộ dữ liệu gồm 2 lớp, và mỗi điểm trong bộ dữ liệu được gán nhãn thuộc vào một trong hai lớp.

Output: Một đường thẳng (hoặc siêu phẳng trong không gian đa chiều) phân cách 2 lớp với khoảng cách từ đường thẳng đó tới điểm gần nhất của từng lớp, là lớn nhất có thể.



Đầu vào và đầu ra của tập dữ liệu - src:minhkhanh-coder

1.3 Pseudo Code cho thuật toán SVM

Algorithm Thuật toán SVM

Đầu vào: Dữ liệu huấn luyện $(x_1, y_1), (x_2, y_2), ..., (x_N, y_N)$.
Đầu ra: Trọng số w và tham số b
Khởi tạo $w \leftarrow 0$, $b \leftarrow 0$, tham số điều chỉnh $C > 0$
repeat
    for all (x_i, y_i) **do**
        Cập nhật trọng số w, b
    end for
until hội tụ (trọng số w, b không thay đổi qua nhiều lần lặp)
Tính toán các trọng số $w = \sum \alpha_i y_i x_i$
Chọn một vector hỗ trợ x_n và tính toán $b = y_n - w^T x_n$
return Trọng số w, tham số b

1.4.1 Tập Hypothesis của SVM

Tập Hypothesis của SVM là tập hợp các giả định được sử dụng để dự đoán lớp của các điểm dữ liệu.

$$y = h(x) = sign\left(w^T x + b\right) \tag{1}$$

Ta có:

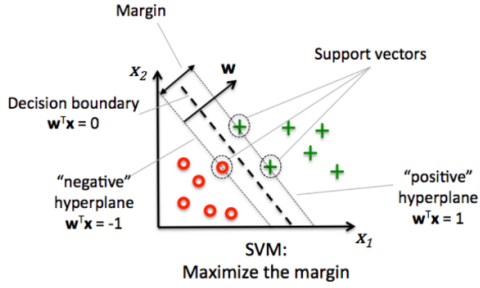
- x là bộ dữ liệu trong không gian d chiều
- w, b là các tham số về siêu phẳng trong không gian để phân tách.

1.4.1 Tập Hypothesis của SVM

Hàm $sign(x)$ là một hàm xác định dấu

$$w^T x_{pos} + b = +1 \tag{2}$$

$$w^T x_{neg} + b = -1 \tag{3}$$



Tối đa hóa lề (margin) - src:ResearchGate

1.4.2 Bài toán tối ưu của SVM

Mục tiêu chính của SVM là **tìm ra một mặt phẳng, chia tách 2 phần dữ liệu với lề lớn nhất**.

- Khoảng cách giữa một điểm và một siêu phẳng:

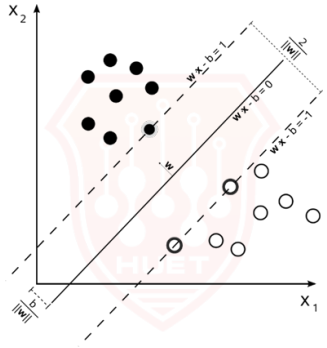
$$\frac{|w^T x_n + b|}{||w||_2} \tag{4}$$

- Trong đó độ dài của vector w được tính bằng:

$$||w||_2 = \sqrt{\sum_{i=1}^{d} w_i^2} \tag{5}$$

1.4.2 Bài toán tối ưu của SVM

Normalize vector w: $|w^T x_n + b| = 1$



Hyperlane - src:iostream.co

$\Rightarrow$ Với N điểm dữ liệu x_n, *ta có:* $y_n(w^T x_n + b) \geq 1 \ \forall n = 1, 2, 3...N$

Độ lớn của **lề lớn nhất** cho một mặt phẳng $w^T x + b$ là: $\frac{2}{||w||_2}$

1.4.2 Bài toán tối ưu của SVM

Ta phát biểu **"bài toán tối ưu cho SVM"** như sau:

$$maximize \frac{2}{||w||_2}$$

$$subject\ to\ y_n \left(w^T x_n + b \right) \geq 1\ \forall\ n = 1, 2, 3, \ldots, N$$

Hay ta có thể viết lại bài toán trên thành:

$$minimize \frac{1}{2} w^T w \tag{6}$$

$$subject\ to\ y_n \left(w^T x_n + b \right) \geq 1\ \forall\ n = 1, 2, 3, \ldots, N$$

1.4.3 Phương pháp tối ưu cho SVM

Để tối ưu cho bài toán SVM, ta dùng phương pháp nhân từ Lagrange[1] và dựng hàm (thêm biến α_n với điều kiện tương ứng).

$$\mathcal{L}(w, b, \alpha) = \frac{1}{2} w^T w - \sum_{n=1}^{N} \alpha_n (y_n \left(w^T x_n + b \right) - 1)$$

$$(6) \Longleftrightarrow \text{Minimize with } w, b; \text{ Maximize with } \alpha : \alpha_n \geq \mathcal{L}(w, b, \alpha) \qquad (*)$$

Chuyển bài toán tối ưu từ dạng primal form cho bài toán (6) thành dạng dual form (đối ngẫu với *min*, *max*) của bài toán $(*)$.

[1] Phương pháp nhân tử Lagrange

1.4.3 Phương pháp tối ưu cho SVM

Để giải bài toán $(*)$, ta có thể giải lần lượt từ *Minimize* cho w, b khi coi các α_n là hằng số. Tiếp theo là giải Maximize với $\alpha_n \geq 0$ coi w, b là hằng số.

Minimize with w, b

$$Minimize\ \mathcal{L}(w, b, \alpha) = \frac{1}{2} w^T w - \sum_{n=1}^{N} \alpha_n (y_n \left(w^T x_n + b \right) - 1)\ with\ w,\ b$$

1.4.3 Phương pháp tối ưu cho SVM

Lấy đạo hàm của w, b và đặt đạo hàm bằng 0, ta có:

$$\frac{\partial \mathcal{L}}{\partial w} = w - \sum_{n=1}^{N} \alpha_n y_n x_n = 0 \Longleftrightarrow w = \sum_{n=1}^{N} \alpha_n y_n x_n \tag{7}$$

$$\frac{\partial \mathcal{L}}{\partial b} = -\sum_{n=1}^{N} \alpha_n y_n = 0 \Longleftrightarrow \sum_{n=1}^{N} \alpha_n y_n = 0 \tag{8}$$

Thay thế (7) và (8) vào $\mathcal{L}(w, b, \alpha)$ và rút gọn, ta có:

$$\mathcal{L}(\alpha) = \sum_{n=1}^{N} \alpha_n - \frac{1}{2} \sum_{n=1}^{N} \sum_{m=1}^{N} \alpha_n \alpha_m y_n y_m x_n^T x_m$$

1.4.3 Phương pháp tối ưu cho SVM

Vậy bài toán tối ưu chỉ còn lại:

$$\text{Maximize } \mathcal{L}(\alpha) = \sum_{n=1}^{N} \alpha_n - \frac{1}{2} \sum_{n=1}^{N} \sum_{m=1}^{N} \alpha_n \alpha_m y_n y_m x_n^T x_m \qquad (**)$$

Với các $\alpha_n \geq 0 \ \forall n = 1, 2, ..., N$ cộng thêm ràng buộc từ (10) $\sum_{n=1}^{N} \alpha_n y_n = 0$.

Vậy từ một dạng primal form (8), ta đã chuyển về dạng dual form như bài toán $(*)$, và cuối cùng ta chỉ cần tìm ra các nghiệm α_n để hoàn thành quá trình tối ưu cho SVM.

1.4.3 Phương pháp tối ưu cho SVM

Việc tìm ra các α_n sẽ sử dụng **Quadratic Programming** (QP - quy hoạch toàn phương) để giải ra. Vậy nên ta có thể tóm tắt lại các bước của việc tối ưu cho SVM như sau:

1. Giải các nghiệm α_n thỏa:

$$\alpha_n^* = argmin_\alpha \frac{1}{2} \sum_{n=1}^{N} \sum_{m=1}^{N} \alpha_n \alpha_m y_n y_m x_n^T x_m - \sum_{n=1}^{N} \alpha_n$$

Với các $\alpha_n \geq 0$ và $\sum_{n=1}^{N} \alpha_n y_n = 0$.

1.4.3 Phương pháp tối ưu cho SVM

2 Từ các α_n, ta tính được w theo (9):

$$w = \sum_{\alpha_n > 0} \alpha_n y_n x_n$$

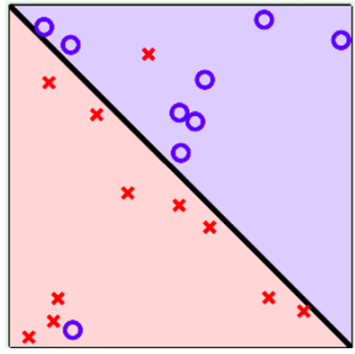
3 Tìm hệ số b. Ta có thể tìm hệ số b bằng cách dùng bất kỳ support vector nào vì $y_n \left(w^T x_n + b \right) = 1$.

Tổng quan

2.1 Dữ liệu không khả tách tuyến tính

Thực tế: Dữ liệu thường không tách được bằng đường thẳng.

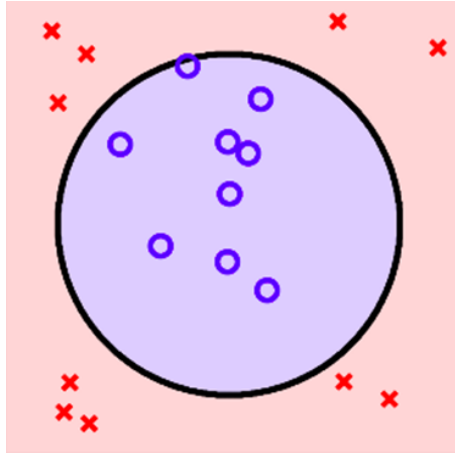
- **Trường hợp 1:** Dữ liệu không thể tách ra bằng 1 đường thẳng.



Dữ liệu không thể tách ra bằng 1 đường thẳng - src: jip.dev

2.1 Dữ liệu không khả tách tuyến tính

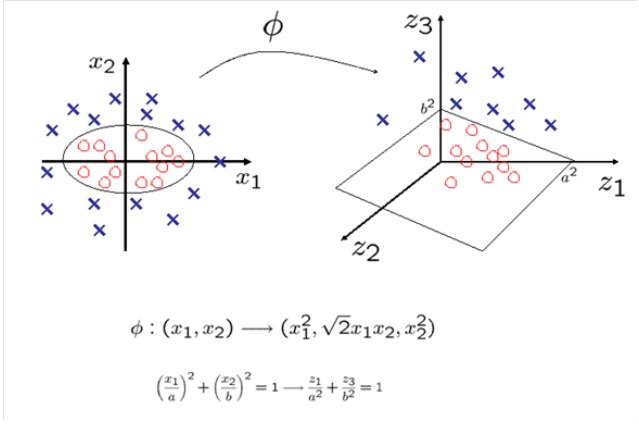
- **Trường hợp 2:** Dữ liệu là phi tuyến tính.



Dữ liệu là phi tuyến tính - src: jip.dev

2.2 Biến đổi phi tuyến với các Kernel

Phương pháp này cho phép chúng ta ánh xạ dữ liệu từ không gian ban đầu sang một không gian khác có số chiều cao hơn, nơi mà việc tách các lớp trở nên dễ dàng hơn.



Biến đổi phi tuyến từ không gian X sang Z - src: omega0.xyz

2.2 Biến đổi phi tuyến với các Kernel

Ý tưởng: thay x_n^T, x_m bằng $K(x_n, x_m)$
Bài toán tối ưu:

$$argmin_\alpha \frac{1}{2} \sum_{n=1}^{N} \sum_{m=1}^{N} \alpha_n \alpha_m y_n y_m K(x_n x_m) - \sum_{n=1}^{N} \alpha_n$$

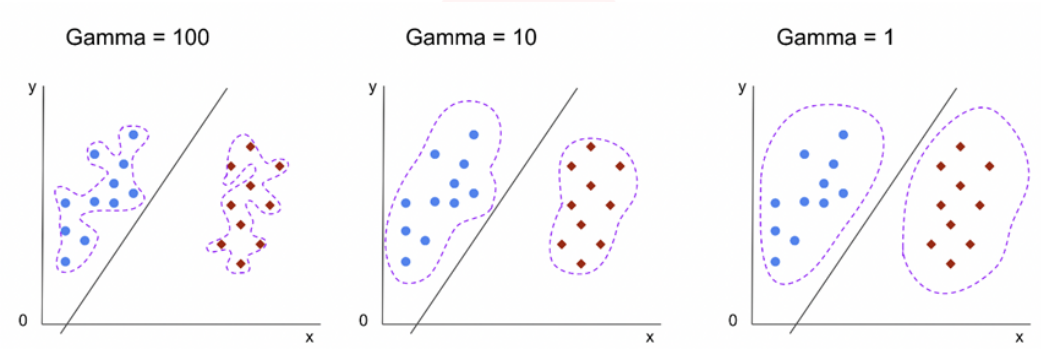
Hàm Kernel:

1. Tuyến tính: $K(x, x') = x^T x'$
2. Đa thức: $K(x, x') = (a(x^T x') + b)^Q$
3. RBF - Radial Basis Function[1]: $K(x, x') = exp(\gamma |x - x\prime|^2)$

[1]Chứng minh phép biến đổi thành không gian vô hạn chiều

2.2 Biến đổi phi tuyến với các Kernel

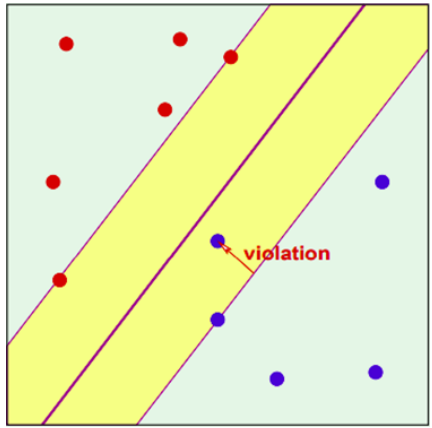
Giá trị γ càng lớn, đường biên càng phức tạp và uốn cong nhiều hơn để vừa khớp với dữ liệu. Trong khi đó, giá trị γ nhỏ sẽ dẫn đến một đường biên mềm hơn và ít uốn cong hơn.



Low gamma – High gamma – src: stackabuse.com

2.3 Mô hình SVM với Lề mềm (Soft-margin)

Trong nhiều trường hợp, mô hình Hard-margin quá khó để đáp ứng $\Rightarrow$ mô hình SVM Soft-margin được tạo ra.



Vi phạm các ràng buộc - src: Caltech

2.3 Mô hình SVM với Lề mềm (Soft-margin)

Cho phép vi phạm lề: $y_n(w^T x_n + b) \geq 1 - \xi_n$
Bài toán tối ưu thành:

$$minimize \frac{1}{2} w^T w + C \sum_{n=1}^{N} \xi_n \tag{11}$$

$$subject\ to\ y_n \left(w^T x_n + b \right) \geq 1\ - \xi_n, \xi_n \geq 0\ \forall\ n = 1, 2, 3, \ldots, N$$

với $\xi_n \geq 0$
C: tham số thể hiện tương quan giữa 2 mục tiêu tối ưu (điều chỉnh giữa lề và lỗi)

2.3 Mô hình SVM với Lề mềm (Soft-margin)

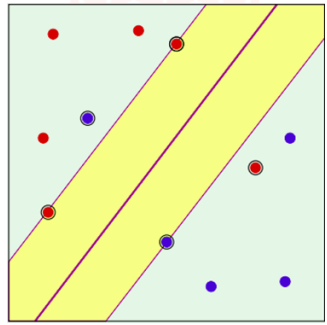
Nhận xét về vai trò của tham số C trong Soft-margin:

- Quyết định cân bằng giữa tối ưu hóa lề và giảm thiểu lỗi phân loại.
- C lớn hơn, mô hình tập trung vào giảm lỗi và lề nhỏ hơn.
- C nhỏ hơn, mô hình chấp nhận lề lớn hơn và số lỗi cao hơn.
- C cũng tương tự như chính quy hóa, ảnh hưởng đến chính xác và độ lớn của lề.

2.3 Mô hình SVM với Lề mềm (Soft-margin)

Các support vector được phân loại thành hai loại:

- Margin support vector: Các vector này có $\alpha_n < C$, nằm trên lề của đường phẳng. Thỏa mãn ràng buộc $y_n \left(w^T x_n + b\right) = 1$
- Non-margin support vector: Các vector này có $\alpha_n = C$, không nằm trên lề và có $\xi_n > 0$. Thỏa mãn ràng buộc $y_n \left(w^T x_n + b\right) < 1$



Hai loại super vector - src: ashwanibhardwajcodevita16

Tổng quan

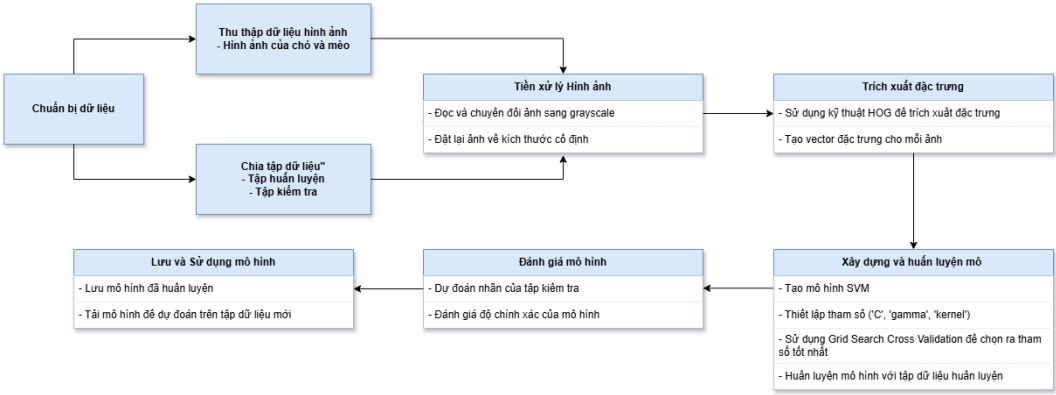
3.1 Đề xuất ứng dụng

Một số ứng dụng của thuật toán SVM đó là:

- **Phân loại hình ảnh:** phân loại các đối tượng trong hình ảnh, như nhận dạng khuôn mặt, phân loại đối tượng trong ảnh y khoa, hay phân loại sản phẩm trong ảnh bán hàng.
- **Phát hiện gian lận:** sử dụng trong lĩnh vực tài chính để phát hiện gian lận trong giao dịch tín dụng hoặc giao dịch tài chính.
- **Nhận diện văn bản:** sử dụng để nhận diện văn bản, ví dụ như phân loại email vào hộp thư rác và hộp thư đến.
- ...

3.2.1 Phân loại hình ảnh

Source code: Click here.



Quy trình phân loại hình ảnh - src: Khánh và Dân

3.2.1 Phân loại hình ảnh

Điểm đánh giá mô hình phân loại hình ảnh chó và mèo.

```
Best parameters found: {'C': 1, 'gamma': 0.1, 'kernel': 'rbf'}
Accuracy: 0.7071428571428572
Model saved to svm_dog_cat_model.pkl
```

Điểm đánh giá mô hình phân loại hình ảnh - src: Khánh và Dân

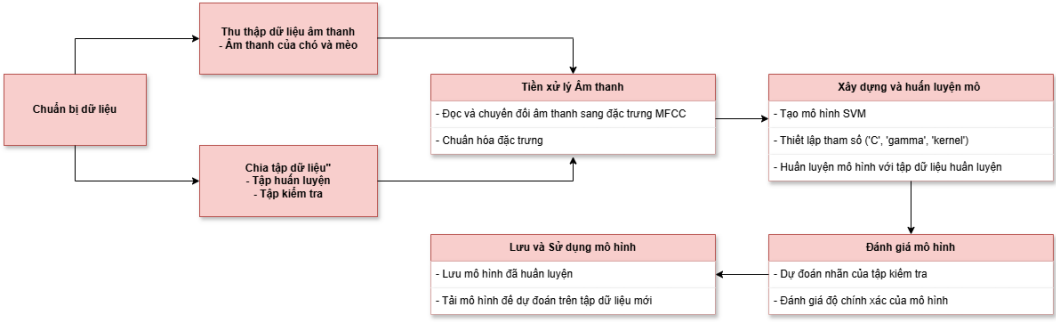
3.2.1 Phân loại hình ảnh

Kết quả sau khi dự đoán với ảnh mới.

Kết quả dự đoán nhãn với ảnh mới - src: Khánh và Dân

3.2.2 Phân loại âm thanh

Link source code: Click here.



Quy trình phân loại âm thanh - src: Khánh và Dân

3.2.2 Phân loại âm thanh

Điểm đánh giá mô hình phân loại âm thanh của chó và mèo.

Accuracy: 0.9104477611940298

	precision	recall	f1-score	support
cat	0.88	0.97	0.93	39
dog	0.96	0.82	0.88	28
accuracy			0.91	67
macro avg	0.92	0.90	0.91	67
weighted avg	0.91	0.91	0.91	67

Điểm đánh giá mô hình phân loại âm thanh - src: Khánh và Dân

Kết quả sau khi dự đoán với âm thanh mới.

```
File: dataset/test\cat\cat_85.wav => dự đoán Label: cat
File: dataset/test\cat\cat_86.wav => dự đoán Label: cat
File: dataset/test\cat\cat_88.wav => dự đoán Label: cat
File: dataset/test\cat\cat_90.wav => dự đoán Label: cat
File: dataset/test\dog\dog_barking_112.wav => dự đoán Label: dog
File: dataset/test\dog\dog_barking_12.wav => dự đoán Label: dog
File: dataset/test\dog\dog_barking_15.wav => dự đoán Label: dog
File: dataset/test\dog\dog_barking_19.wav => dự đoán Label: dog
File: dataset/test\dog\dog_barking_24.wav => dự đoán Label: cat
File: dataset/test\dog\dog_barking_3.wav => dự đoán Label: cat
File: dataset/test\dog\dog_barking_34.wav => dự đoán Label: dog
```

Điểm đánh giá mô hình phân loại âm thanh - src: Khánh và Dân

Tổng quan

Tài liệu tham khảo

[1] Learning From Data - Caltech.

[2] SVM Introduction - Bùi Văn Hợp.

[3] Python Machine Learning Packt Publishing 2015 - Sebastian Raschka

[4] ChatGPT