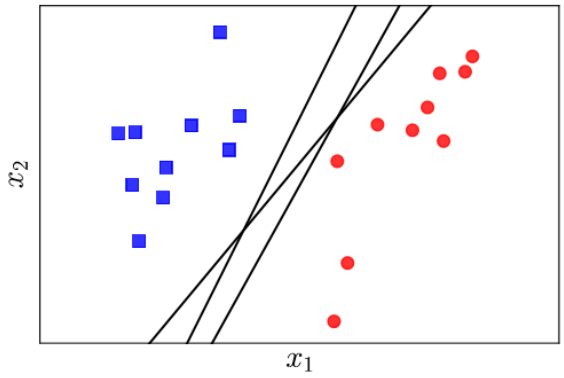
Support Vector Machines

1. Support vector machine (SVM)

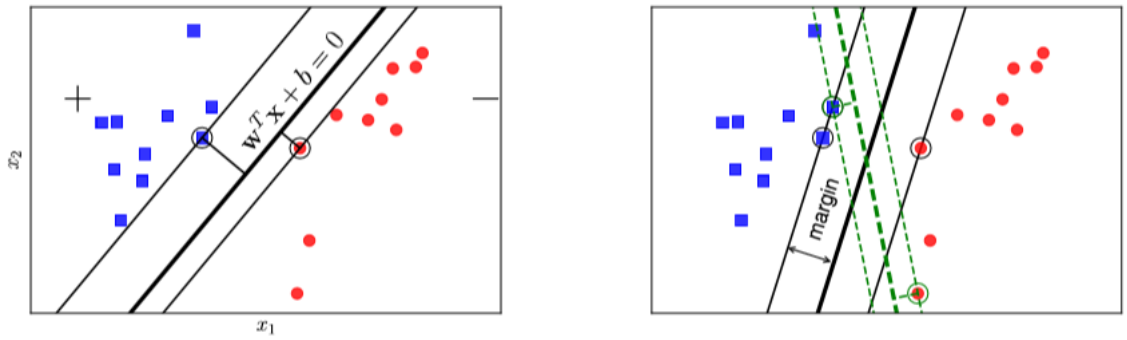
1.1 Ý tưởng

Quay lại bài toán phân lớp Perceptron Learning Algorithm, giả sử 2 lớp dữ liệu này là linearly separable, khi đó tồn tại siêu mặt phẳng phân chia chính xác 2 lớp đó. Nhưng không chỉ có 1 siêu mặt phẳng, ta có vô số mặt phẳng có thể phân chia 2 lớp đó, câu hỏi đặt ra là trong vô số các mặt phân chia đó, mặt nào là tốt nhất?



Ở hình trên, ta thấy có đường thẳng khá lệch về phía màu đỏ, khiến cho lớp này *không vui vì lãnh thổ bị lấn quá nhiều.* Việc này dẫn đến phân lớp trong tương lai có thể dẫn đến sai lệch do phân các điểm màu đỏ vào lớp nàu xanh.

Để giải quyết vấn đề này, ta cần một đại lượng để đo *mức độ hạnh phúc* của mỗi lớp, là khoảng cách gần nhất từ một điểm của lớp đó tới đường thẳng. Chúng ta phải tìm được một đường phân chia sao cho khoảng cách từ điểm gần nhất của mỗi lớp đến đường phân chia là như nhau thì mới *công bằng.* Khoảng cách như nhau này gọi là *biên* hoặc *lề* (margin).



Để việc phân chia mang lại hiệu ứng tốt, ta tìm đường phân lớp tạo margin lớn nhất có thể. Bài toán tối ưu trong SVM chính là đi tìm đường phân chia sao cho margin giữa 2 lớp là lớn nhất. (Svm còn có tên khác là *maximum margin classifier*)

1.2 Xây dựng bài toán tối ưu cho SVM

Giả sử các cặp dữ liệu trong tập huấn luyện là (x_1, y_1), (x_2, y_2), …, (x_N, y_N) với N là số điểm dữ liệu, x_i là dữ liệu đầu vào, y_i là nhãn tương ứng (1 or -1).

Đi từ bài toán tính khoảng cách từ một điểm (x_0, y_0) đến một đường thẳng $w_1x + w_2y + b = 0$:

$$\frac{|w_1 x_0 + w_2 y_0 + b|}{\sqrt{w_1^2 + w_2^2}}$$

Ta suy ra, với cặp dữ liệu (x_n, y_n) bất kì, khoảng cách từ điểm đó tới mặt phân cách là:

$$\frac{y_n(\mathbf{w}^T \mathbf{x}_n + b)}{\|\mathbf{w}\|_2}$$

mặt phân cách có pt là $\boldsymbol{w}^T x + b = 0$

Với mặt phân chia này, margin được tính là khoảng cách gần nhất từ một điểm tới mặt đó,

$$margin = \min_n \frac{y_n(\mathbf{w}^T \mathbf{x}_n + b)}{\|\mathbf{w}\|_2}$$

Bài toán tối ưu của SVM là tìm **w** và b sao cho margin này là lớn nhất:

$$(\mathbf{w}, b) = \arg\max_{\mathbf{w},b} \left\{ \min_n \frac{y_n(\mathbf{w}^T \mathbf{x}_n + b)}{\|\mathbf{w}\|_2} \right\} = \arg\max_{\mathbf{w},b} \left\{ \frac{1}{\|\mathbf{w}\|_2} \min_n y_n(\mathbf{w}^T \mathbf{x}_n + b) \right\}$$

Khi ta nhân k (k > 0) vào pt đường phân cách thì nó không đổi, margin không đổi nên có thể giả sử $y_n(w^T x_n + b) = 1$ (chọn k phù hợp) với những điểm nằm gần mặt phân chia nhất, cho nên với mọi n ta luôn có: $y_n(w^T x_n + b) >= 1$

Biểu thức tối ưu có thể viết lại:

$$(\mathbf{w}, b) = \arg\max_{\mathbf{w},b} \frac{1}{\|\mathbf{w}\|_2}$$
$$\text{thoả mãn: } y_n(\mathbf{w}^T \mathbf{x}_n + b) \geq 1, \forall n = 1, 2, \ldots, N$$

Ta biến đổi để đưa về dang này:

$$(\mathbf{w}, b) = \arg\min_{\mathbf{w},b} \frac{1}{2}\|\mathbf{w}\|_2^2$$
$$\text{thoả mãn: } 1 - y_n(\mathbf{w}^T \mathbf{x}_n + b) \leq 0, \forall n = 1, 2, \ldots, N$$

Nhận xét: Trong bài toán trên, hàm mục tiêu là một norm nên là một hàm lồi, các bất đẳng thức ràng buộc là hàm tuyến tính nên chúng cũng là các hàm lồi. Suy ra bài toán tối ưu có hàm mục tiêu là lồi và hàm mục tiêu có strictly convex nên nghiệm SVM là duy nhất

Xác định lớp cho một điểm dữ liệu mới

Sau khi đã tìm được mặt phân cách $\mathbf{w}^T x + b = 0$, nhãn của một điểm mới lẽ là:

class(x) = sgn($\boldsymbol{w}^T x + b$*)*

1.3 Bài toán đối ngẫu SVM

$$(\mathbf{w}, b) = \arg\min_{\mathbf{w},b} \frac{1}{2}\|\mathbf{w}\|_2^2$$
$$\text{thoả mãn: } 1 - y_n(\mathbf{w}^T \mathbf{x}_n + b) \leq 0, \forall n = 1, 2, \ldots, N$$

Bài toán trên là một bài toán lồi, mà nếu một bài toán lồi thỏa mãn tiêu chuẩn Slater thì strong duality xảy ra, khi đó nghiệm của bài toán chính là nghiệm của hệ điều kiện KKT

Kiểm tra tiêu chuẩn Slater

Tiêu chuẩn Slater nói rằng nếu tồn tại **w** và b thỏa mãn điều kiện sau thì strong duality xảy ra.

$$1 - y_n(w^T x_n + b) < 0 \text{ với } n = 1, 2, ..., N$$

Ta đã biết, nếu 2 lớp là linearly separable thì bài toán luôn có nghiệm, khi đó feasible set của bài toán phải khác rỗng. Tức luôn tồn tại một cặp ($\mathbf{w}_0$, b_0) sao cho:

$$\begin{aligned} & 1 - y_n(\mathbf{w}_0^T \mathbf{x}_n + b_0) \leq 0, \quad \forall n = 1, 2, \ldots, N \\ \Leftrightarrow\ & 2 - y_n(2\mathbf{w}_0^T \mathbf{x}_n + 2b_0) \leq 0, \quad \forall n = 1, 2, \ldots, N \end{aligned}$$

Vậy ta chỉ cần chọn $\mathbf{w}_1 = 2\mathbf{w}_0$ và $b_1 = 2b_0$, ta sẽ có:

$$1 - y_n(\mathbf{w}_1^T \mathbf{x}_n + b_1) \leq -1 < 0, \quad \forall n = 1, 2, \ldots, N$$

Từ đó suy ra điều kiện Slater thoả mãn.

Lagrangian của bài toán SVM

Lagrangina của bài toán tối ưu là:

$$\mathcal{L}(\mathbf{w}, b, \boldsymbol{\lambda}) = \frac{1}{2}\|\mathbf{w}\|_2^2 + \sum_{n=1}^{N} \lambda_n (1 - y_n(\mathbf{w}^T \mathbf{x}_n + b))$$

với $\boldsymbol{\lambda} = [\lambda_1, \lambda_2, \ldots, \lambda_N]^T$ và $\lambda_n \geq 0, \ \forall n = 1, 2, \ldots, N$.

Hàm đối ngẫu Lagrange

Theo định nghĩa, hàm đối ngẫu Lagrange là:

$$g(\boldsymbol{\lambda}) = \min_{\mathbf{w}, b} \mathcal{L}(\mathbf{w}, b, \boldsymbol{\lambda})$$

Với λ >= 0. Việc tìm giá trị nhỏ nhất của hàm này theo **w** và b có thể được tính bằng cách giải hệ phương trình đạo hàm của L(**w**, b, λ) theo **w** và b bằng 0:

$$\begin{aligned} \nabla_{\mathbf{w}} \mathcal{L}(\mathbf{w}, b, \boldsymbol{\lambda}) &= \mathbf{w} - \sum_{n=1}^{N} \lambda_n y_n \mathbf{x}_n = \mathbf{0} \Rightarrow \mathbf{w} = \sum_{n=1}^{N} \lambda_n y_n \mathbf{x}_n \\ \nabla_b \mathcal{L}(\mathbf{w}, b, \boldsymbol{\lambda}) &= \sum_{n=1}^{N} \lambda_n y_n = 0 \end{aligned}$$

Thay vào biểu thức Lagrangina, sau rút gọn ta được:

$$g(\boldsymbol{\lambda}) = \sum_{n=1}^{N} \lambda_n - \frac{1}{2} \sum_{n=1}^{N} \sum_{m=1}^{N} \lambda_n \lambda_m y_n y_m \mathbf{x}_n^T \mathbf{x}_m$$

*Đây là **hàm số quan trọng nhất** của SVM, cần lưu ý*

Bài toán đối ngẫu Lagrange

Kết hợp hàm đối ngẫu Lagrange và các điều kiện ràng buộc của **λ**, ta thu được bài toán đối ngẫu của biểu thức tối ưu có dạng:

$$\begin{aligned} & \boldsymbol{\lambda} = \arg\max_{\boldsymbol{\lambda}} g(\boldsymbol{\lambda}) \\ & \text{thoả mãn: } \boldsymbol{\lambda} \succeq 0 \\ & \sum_{n=1}^{N} \lambda_n y_n = 0 \end{aligned}$$

Điều kiện KKT

Vì đây là bài toán lồi và strong duality xảy ra, nghiệm của bài toán sẽ thỏa mãn hệ điều kiện Kkt sau với các biến số là **w**, b và **λ**

$$\begin{aligned} 1 - y_n(\mathbf{w}^T\mathbf{x}_n + b) &\leq 0, \ \forall n = 1, 2, \ldots, N \\ \lambda_n &\geq 0, \ \forall n = 1, 2, \ldots, N \\ \lambda_n(1 - y_n(\mathbf{w}^T\mathbf{x}_n + b)) &= 0, \ \forall n = 1, 2, \ldots, N \\ \mathbf{w} &= \sum_{n=1}^{N} \lambda_n y_n \mathbf{x}_n \\ \sum_{n=1}^{N} \lambda_n y_n &= 0 \end{aligned}$$

Từ điều kiện thứ 3, ta có thể suy ra ngay với bất kì n, hoặc $\lambda_n = 0$ hoặc $1 - y_n(\boldsymbol{w}^T\boldsymbol{x}_n + b) = 0$. Trường hợp thứ 2 tương đương với $\boldsymbol{w}^T\boldsymbol{x}_n + b = y_n$.

Sau khi tìm được λ từ bài toán đối ngẫu, ta có thể tính được **w** và b dựa vào hệ điều kiện KKT. Ta chỉ quan tâm tới λ != 0. Đặt S = {n : λ != 0} và N_S là số phần tử của tập S. Ta tính được:

$$\mathbf{w} = \sum_{m \in \mathcal{S}} \lambda_m y_m \mathbf{x}_m$$

Với mỗi n thuộc S, ta có:

$$1 = y_n(w^T x_n + b) \Leftrightarrow b = y_n - w^T x_n$$

Để ổn định hơn trong tính toán, ta tính trung bình cộng của các b tính được theo mỗi n thuộc S:

$$b = \frac{1}{N_S} \sum_{n \in \mathcal{S}} (y_n - \mathbf{w}^T\mathbf{x}_n) = \frac{1}{N_S} \sum_{n \in \mathcal{S}} \left(y_n - \sum_{m \in \mathcal{S}} \lambda_m y_m \mathbf{x}_m^T \mathbf{x}_n \right)$$

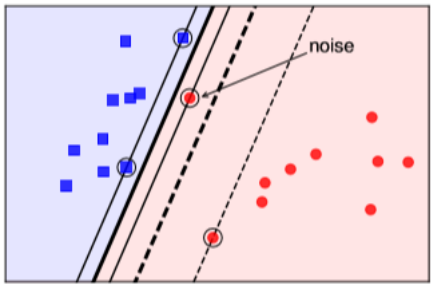
Để xác định một điểm **x** mới thuộc lớp nào, ta chỉ việc xác định dấu của biểu thức sau:

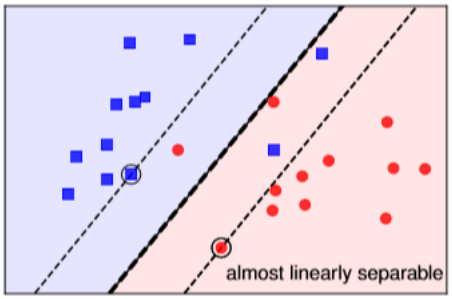
$$\mathbf{w}^T\mathbf{x} + b = \sum_{m \in \mathcal{S}} \lambda_m y_m \mathbf{x}_m^T \mathbf{x} + \frac{1}{N_S} \sum_{n \in \mathcal{S}} \left(y_n - \sum_{m \in \mathcal{S}} \lambda_m y_m \mathbf{x}_m^T \mathbf{x}_n \right)$$

2. Soft-margin support vector machine

2.1 Giới thiệu

Giống như *Perceptron Learning Algorithm (PLA)*, *Support Vector Machine (SVM)* chỉ cho ra nghiệm với các lớp dữ liệu *Linearly separable.* Chúng ta cũng mong muốn SVM có thể làm việc với dữ liệu gần Linearly separable.





Ở TH đầu, điểm nhiễu màu đỏ làm cho đường phân lớp quá gần lớp màu xanh, dẫn đến việc phân lớp sau này có nhầm lẫn, nếu ta *hi sinh* điểm này thì ta được một đường cho margin tốt hơn.

TH thứ 2, dữ liệu không Linearly separable mà gần Linearly separable, SVM trở nên vô nghiệm. Nếu ta chịu *hi sinh*một vài điểm ở khu vực biên giới của 2 lớp, ta vẫn tạo được đường phân chia tương đối tốt (đường nét đứt đậm). Các đường nét đứt mảnh giúp tạo margin cho lớp bộ phận này. Nếu các điểm nằm phía bên kia các đường support, ta nói những điểm đó rơi vào vùng k an toàn.

Trong cả 2 TH trên, margin được tạo bởi các đường support gọi là soft-margin. SVM chấp nhận mất một vài điểm trong tập huấn luyện được gọi là soft-margin SVM

Có 2 cách xây dựng và giải quyết bài toán tối ưu của soft-margin SVM:

- Giải bài toán tối ưu có ràng buộc bằng cách giải bài toán đối ngẫu như SVM

- Đưa về bài toán tối ưu không ràng buộc và dùng GD

2.2 Phân tích

$$(\mathbf{w}, b) = \arg\min_{\mathbf{w},b} \frac{1}{2}\|\mathbf{w}\|_2^2$$
$$\text{thoả mãn: } y_n(\mathbf{w}^T\mathbf{x}_n + b) \geq 1, \quad \forall n = 1, 2, \ldots, N$$

Soft-margin SVM là sự kết hợp của tối đa margin và tối thiểu sự hi sinh. Tương tự SVM, tối đa margin có thể đưa về tối thiểu $||\mathbf{w}||_2^2$. Để đong đếm sự hi sinh ta sử dụng một biến gọi là *slack variable* ξ_n. Với những điểm nằm trong vùng an toàn $\xi_n=0$, tức không có sự hi sinh nào, với những điểm nằm ngoài vùng an toàn: $\xi_n > 0$, đã có mất mát xảy ra. Đại lượng này tỉ lệ với khoảng cách từ điểm vi phạm tương ứng tới biên an toàn. Ta có thể định nghĩa ξ_i

$$\xi_n = |\boldsymbol{w}^T x_i + b - y_i|.$$

Hàm mục tiêu của soft-margin SVM sẽ là:

$$\frac{1}{2}\|\mathbf{w}\|_2^2 + C\sum_{n=1}^{N}\xi_n$$

C là hằng số dương, và điều kiện sử dụng ràng buộc mềm:

$$y_n(\mathbf{w}^T\mathbf{x}_n + b) \geq 1 - \xi_n \Leftrightarrow 1 - \xi_n - y_n(\mathbf{w}^T\mathbf{x}_n + b) \leq 0, \quad \forall n = 1, 2, \ldots, n$$

Và ràng buộc phụ $\xi_n \geq 0, \ \forall n = 1, 2, \ldots, N.$

Như vậy, ta sẽ có bài toán tối ưu primal cho soft-margin Svm như sau:

$$(\mathbf{w}, b, \xi) = \arg\min_{\mathbf{w},b,\xi} \frac{1}{2}\|\mathbf{w}\|_2^2 + C\sum_{n=1}^{N}\xi_n$$
$$\text{thoả mãn: } 1 - \xi_n - y_n(\mathbf{w}^T\mathbf{x}_n + b) \leq 0, \forall n = 1, 2, \ldots, N$$
$$-\xi_n \leq 0, \ \forall n = 1, 2, \ldots, N$$

Nhận xét:

- C là hằng số đánh giá mức độ quan trọng của sự hi sinh so với max margin. Nếu C nhỏ, thuật toán sẽ điều chỉnh sao cho margin lớn nhất, vùng an toàn sẽ nhỏ đi. Nếu C quá lớn thì bài toán sẽ tập trung đi vào giảm sự hi sinh.

- Hàm mục tiêu trong bài toán tối ưu là một hàm lồi vì nó là tổng của 2 hàm lồi.

2.3 Bài toán đối ngẫu Lagrange

Kiểm tra tiêu chuẩn Slater

Với mọi n = 1, 2, …, N và mọi (**w**, b) ta luôn tìm được các số dương ξ_n đủ lớn sao cho

$$y_n(\boldsymbol{w}^T x_i + b) + \xi_n > 1$$

Vì vậy, bài toán thỏa mãn tiêu chuẩn Slater

Lagrangian của bài toán Soft-margin SVM

Lagrange cho bài toán tối ưu là

$$\mathcal{L}(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\lambda}, \boldsymbol{\mu}) = \frac{1}{2}\|\mathbf{w}\|_2^2 + C\sum_{n=1}^{N}\xi_n + \sum_{n=1}^{N}\lambda_n(1 - \xi_n - y_n(\mathbf{w}^T\mathbf{x}_n + b)) - \sum_{n=1}^{N}\mu_n\xi_n$$

với $\boldsymbol{\lambda} = [\lambda_1, \lambda_2, \ldots, \lambda_N]^T \succeq 0$ và $\boldsymbol{\mu} = [\mu_1, \mu_2, \ldots, \mu_N]^T \succeq 0$ là các biến đối ngẫu Lagrange.

Bài toán đối ngẫu

Hàm đối ngẫu của bài toán tối ưu là:

$$g(\boldsymbol{\lambda}, \boldsymbol{\mu}) = \min_{\mathbf{w}, b, \boldsymbol{\xi}} \mathcal{L}(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\lambda}, \boldsymbol{\mu})$$

Với mỗi cặp $(\boldsymbol{\lambda}, \boldsymbol{\mu})$, chúng ta sẽ quan tâm tới $(\mathbf{w}, b, \boldsymbol{\xi})$ thoả mãn điều kiện đạo hàm của Lagrangian bằng 0:

$$\nabla_{\mathbf{w}}\mathcal{L} = 0 \Leftrightarrow \mathbf{w} = \sum_{n=1}^{N}\lambda_n y_n \mathbf{x}_n$$

$$\nabla_b \mathcal{L} = 0 \Leftrightarrow \sum_{n=1}^{N}\lambda_n y_n = 0$$

$$\nabla_{\xi_n}\mathcal{L} = 0 \Leftrightarrow \lambda_n = C - \mu_n$$

Ta chỉ quan tâm tới những cặp **(λ, μ)** sao cho $\lambda_n = C - \mu_n$. Suy ra, $0 <= \lambda_n, \mu_n <= C$. Thay các biểu thức này vào Lagrangian, ta thu được hàm mục tiêu của bài toán đối ngẫu:

$$g(\boldsymbol{\lambda}, \boldsymbol{\mu}) = \sum_{n=1}^{N}\lambda_n - \frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N}\lambda_n\lambda_m y_n y_m \mathbf{x}_n^T\mathbf{x}_m$$

Hàm này không phụ thuộc vào **μ** nên bài toán đối ngẫu trở thành:

$$\boldsymbol{\lambda} = \arg\max_{\boldsymbol{\lambda}} g(\boldsymbol{\lambda})$$

$$\text{thoả mãn:} \quad \sum_{n=1}^{N}\lambda_n y_n = 0$$

$$0 \le \lambda_n \le C, \ \forall n = 1, 2, \ldots, N$$

Hệ điều kiện KKT

Hệ điều kiện KKT của bài toán tối ưu soft-margin SVM là:

$$\begin{aligned} 1 - \xi_n - y_n(\mathbf{w}^T\mathbf{x}_n + b) &\leq 0 \\ -\xi_n &\leq 0 \\ \lambda_n &\geq 0 \\ \mu_n &\geq 0 \\ \lambda_n(1 - \xi_n - y_n(\mathbf{w}^T\mathbf{x}_n + b)) &= 0 \\ \mu_n\xi_n &= 0 \\ \mathbf{w} &= \sum_{n=1}^{N} \lambda_n y_n \mathbf{x}_n \\ \sum_{n=1}^{N} \lambda_n y_n &= 0 \\ \lambda_n &= C - \mu_n \end{aligned}$$

Ta thấy, chỉ những n ứng với $\lambda_n > 0$ mới đóng góp vào nghiệm **w** của bài toán, khi đó

$$y_n(w^T x_n + b) = 1 - \xi_n$$

Thêm điều kiện $0 < \lambda_n < C$, $\mu_n = C - \lambda_n > 0$ và $\mu_n\xi_n = 0$ suy ra $\xi_n = 0$. ta được:

$$y_n(w^T x_n + b) = 1 \text{ hay } w^T x_n + b = y_n$$

Giá trị của b có thể được tính như sau

$$b = \frac{1}{N_{\mathcal{M}}} \sum_{m \in \mathcal{M}} \left(y_m - \mathbf{w}^T\mathbf{x}_m\right)$$

với $M = \{m : 0 < \lambda_m < C\}$ và N_M là số phần tử của M

Xác định nhãn cho một điểm dữ liệu mới bằng cách xét dấu biểu thức sau:

$$\mathbf{w}^T\mathbf{x} + b = \sum_{m \in \mathcal{S}} \lambda_m y_m \mathbf{x}_m^T\mathbf{x} + \frac{1}{N_{\mathcal{M}}} \sum_{n \in \mathcal{M}} \left(y_n - \sum_{m \in \mathcal{S}} \lambda_m y_m \mathbf{x}_m^T\mathbf{x}_n\right)$$

2.4 Bài toán tối ưu không ràng buộc cho soft-margin SVM

Bài toán tối ưu không ràng buộc tương đương

Điều kiện ràng buộc

$$1 - \xi_n - y_n(\mathbf{w}^T\mathbf{x} + b) \leq 0 \Leftrightarrow \xi_n \geq 1 - y_n(\mathbf{w}^T\mathbf{x} + b)$$

khi kết hợp với điều kiện $\xi_n >= 0$, ta được bài toán ràng buộc tương đương

$$(\mathbf{w}, b, \xi) = \arg\min_{\mathbf{w},b,\xi} \frac{1}{2}\|\mathbf{w}\|_2^2 + C\sum_{n=1}^{N} \xi_n$$

$$\text{thoả mãn: } \xi_n \geq \max(0, 1 - y_n(\mathbf{w}^T\mathbf{x} + b)),\ \forall n = 1, 2, \ldots, N$$

Nếu (**w**, b, ξ) là nghiệm của bài toán tối ưu, tức tại đó hàm mục tiêu đạt giá trị nhỏ nhất thì

$$\xi_n = \max(0, 1 - y_n(\mathbf{w}^T\mathbf{x}_n + b)),\ \forall n = 1, 2, \ldots, N$$

Bằng cách thay các giá trị của ξ_n vào hàm mục tiêu, ta được bài toán tối ưu

$$(\mathbf{w}, b, \xi) = \arg\min_{\mathbf{w},b,\xi} \frac{1}{2}\|\mathbf{w}\|_2^2 + C\sum_{n=1}^{N}\max(0, 1 - y_n(\mathbf{w}^T\mathbf{x}_n + b))$$
$$\text{thoả mãn: } \xi_n = \max(0, 1 - y_n(\mathbf{w}^T\mathbf{x}_n + b)),\ \forall n = 1, 2, \ldots, N$$

Tương đương với (vì biến số ξ_n không còn quan trọng nữa)

$$(\mathbf{w}, b) = \arg\min_{\mathbf{w},b} \left\{ \frac{1}{2}\|\mathbf{w}\|_2^2 + C\sum_{n=1}^{N}\max(0, 1 - y_n(\mathbf{w}^T\mathbf{x}_n + b)) \triangleq J(\mathbf{w}, b) \right\}$$

Đây là bài toán tối ưu không ràng buộc với hàm mất mát *J(**w**, b)*

Hàm hinge loss

Hàm hinge loss có dạng:

$$J_n(\mathbf{w}, b) = max(0, 1 - y_n z_n)$$

trong đó $z_n = \mathbf{w}^T x_n + b$ có thể coi là score của x_n , y_n là đầu ra mong muốn.

Với những điểm nằm trong vùng an toàn, thì yz >= 1 sẽ không gây mất mát. Những điểm nằm ngoài vùng an toàn sẽ có yz < 1, do đó sẽ có mất mát. Với những điểm sai lệch càng xa mặt phẳng thì giá trị phạt càng lớn.

Xây dựng hàm mất mát

Xét bài toán soft-margin SVM bằng cách sử dụng *hinge loss*, với mỗi cặp (**w**, b) đặt

$$L_n(\mathbf{w}, b) = \max(0, 1 - y_n z_n) = \max(0, 1 - y_n(\mathbf{w}^T\mathbf{x}_n + b))$$

Lấy trung bình cộng của chúng, ta được:

$$L(\mathbf{w}, b) = \frac{1}{N}\sum_{n=1}^{N} L_n = \frac{1}{N}\sum_{n=1}^{N}\max(0, 1 - y_n(\mathbf{w}^T\mathbf{x}_n + b))$$

Ta thấy nếu (**w**, b) là nghiệm của bài toán thì (a**w**, ab) cũng là nghiệm, để tránh TH nghiệm quá lớn, ta sử dụng *regularization parameter l_2*

$$J(\mathbf{w}, b) = \frac{1}{N}\left(\underbrace{\sum_{n=1}^{N}\max(0, 1 - y_n(\mathbf{w}^T\mathbf{x}_n + b))}_{\text{hinge loss}} + \underbrace{\frac{\lambda}{2}\|\mathbf{w}\|_2^2}_{\text{regularization}} \right)$$

Tối ưu hàm mất mát

Để tối ưu hàm mất mát, ta tính đạo hàm theo **w** and b

Đạo hàm của phần *hinge loss* không quá phức tạp:

$$\nabla_{\mathbf{w}} \left(\max(0, 1 - y_n(\mathbf{w}^T\mathbf{x}_n + b))\right) = \begin{cases} -y_n\mathbf{x}_n & \text{nếu} \quad 1 - y_n(\mathbf{w}^T\mathbf{x}_n + b) \geq 0 \\ \mathbf{0} & \text{o.w.} \end{cases}$$

$$\nabla_b \left(\max(0, 1 - y_n(\mathbf{w}^T\mathbf{x}_n + b))\right) = \begin{cases} -y_n & \text{nếu} \quad 1 - y_n(\mathbf{w}^T\mathbf{x}_n + b) \geq 0 \\ 0 & \text{o.w.} \end{cases}$$

Phần *regularization* cũng có đạo hàm tương đối đơn giản:

$$\nabla_{\mathbf{w}} \left(\frac{\lambda}{2}\|\mathbf{w}\|_2^2\right) = \lambda\mathbf{w}; \quad \nabla_b \left(\frac{\lambda}{2}\|\mathbf{w}\|_2^2\right) = 0$$

Biểu thức cập nhật nghiệm:

$$\begin{array}{llll} \mathbf{w} \leftarrow \mathbf{w} - \eta(-y_n\mathbf{x}_n + \lambda\mathbf{w}); & b \leftarrow b + \eta y_n & \text{nếu} & 1 - y_n(\mathbf{w}^T\mathbf{x}_n + b) \geq 0) \\ \mathbf{w} \leftarrow \mathbf{w} - \eta\lambda\mathbf{w}; & b \leftarrow b & \text{o.w.} & \end{array}$$

với η là learning rate

3. Kernel Support Vector Machine

3.1 Giới thiệu

Ý tưởng cơ bản của Kernel SVM là biến đổi kiểu dữ liệu không linearly separable ở một không gian sang không gian mới mà ở đó dữ liệu này trở nên linearly separable or gần linearly separable, có thể giải được bằng SVM or soft-margin SVM

Nói theo cách khác Kernel SVM là phương pháp đi tìm một hàm số Φ() biến đổi dữ liệu x ban đầu thành dữ liệu trong không gian mới (thường là nhiều chiều hơn dữ liệu ban đầu, có thể là vô hạn)

3.2 Phân tích

Trong soft-margin SVM, ta có bài toán đối ngẫu:

$$\boldsymbol{\lambda} = \arg\max_{\boldsymbol{\lambda}} \sum_{n=1}^{N} \lambda_n - \frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N} \lambda_n\lambda_m y_n y_m \mathbf{x}_n^T\mathbf{x}_m$$

$$\text{thoả mãn: } \sum_{n=1}^{N} \lambda_n y_n = 0$$

$$0 \leq \lambda_n \leq C, \ \forall n = 1, 2, \ldots, N$$

Sau khi giải được, nhãn của dữ liệu mới sẽ là:

$$\text{class}(\mathbf{x}) = \text{sgn}\left\{\sum_{m\in\mathcal{S}} \lambda_m y_m \mathbf{x}_m^T\mathbf{x} + \frac{1}{N_{\mathcal{M}}}\sum_{n\in\mathcal{M}}\left(y_n - \sum_{m\in\mathcal{S}} \lambda_m y_m \mathbf{x}_m^T\mathbf{x}_n\right)\right\}$$

Khi áp dụng Kernel SVM, mỗi điểm dữ liệu **x** trở thành Φ(**x**), bài toán trở thành

$$\boldsymbol{\lambda} = \arg\max_{\boldsymbol{\lambda}} \sum_{n=1}^{N} \lambda_n - \frac{1}{2} \sum_{n=1}^{N} \sum_{m=1}^{N} \lambda_n \lambda_m y_n y_m \Phi(\mathbf{x}_n)^T \Phi(\mathbf{x}_m)$$

$$\text{thoả mãn: } \sum_{n=1}^{N} \lambda_n y_n = 0$$

$$0 \leq \lambda_n \leq C, \ \forall n = 1, 2, \ldots, N$$

và *nhãn* của một điểm dữ liệu mới được xác định bởi dấu của biểu thức

$$\mathbf{w}^T \Phi(\mathbf{x}) + b = \sum_{m \in \mathcal{S}} \lambda_m y_m \Phi(\mathbf{x}_m)^T \Phi(\mathbf{x}) + \frac{1}{N_{\mathcal{M}}} \sum_{n \in \mathcal{M}} \left(y_n - \sum_{m \in \mathcal{S}} \lambda_m y_m \Phi(\mathbf{x}_m)^T \Phi(\mathbf{x}_n) \right)$$

Bằng cách định nghĩa hàm kernel $k(\mathbf{x}, \mathbf{z}) = \Phi(\mathbf{x})^T \Phi(\mathbf{z})$, ta có thể viết lại bài toán:

$$\boldsymbol{\lambda} = \arg\max_{\boldsymbol{\lambda}} \sum_{n=1}^{N} \lambda_n - \frac{1}{2} \sum_{n=1}^{N} \sum_{m=1}^{N} \lambda_n \lambda_m y_n y_m k(\mathbf{x}_n, \mathbf{x}_m)$$

$$\text{thoả mãn: } \sum_{n=1}^{N} \lambda_n y_n = 0$$

$$0 \leq \lambda_n \leq C, \ \forall n = 1, 2, \ldots, N$$

$$\sum_{m \in \mathcal{S}} \lambda_m y_m k(\mathbf{x}_m, \mathbf{x}) + \frac{1}{N_{\mathcal{M}}} \sum_{n \in \mathcal{M}} \left(y_n - \sum_{m \in \mathcal{S}} \lambda_m y_m k(\mathbf{x}_m, \mathbf{x}_n) \right)$$

Note: Hàm kernel có tính chất đối xứng $k(\mathbf{x}, \mathbf{z}) = k(\mathbf{z}, \mathbf{x})$ vì tích vô hướng 2 vector có tính đối xứng

3.3 Một số hàm số kernel thông dụng

- Linear
- Polynomial
- Radial basic function
- Sigmod

Tên kernel	Công thức	Thiết lập hệ số
'linear'	$\mathbf{x}^T\mathbf{z}$	không có hệ số
'poly'	$(r + \gamma \mathbf{x}^T \mathbf{z})^d$	d: degree, γ: gamma, r: coef0
'sigmoid'	$\tanh(\gamma \mathbf{x}^T \mathbf{z} + r)$	γ: gamma, r: coef0
'rbf'	$\exp(-\gamma \&\mathbf{x} - \mathbf{z}\&_2^2)$	$\gamma > 0$: gamma

Ngoài ra, người dùng có thể tự tạo customize kernel

4. Multi-class Support Vector Machine

4.1 Ý tưởng

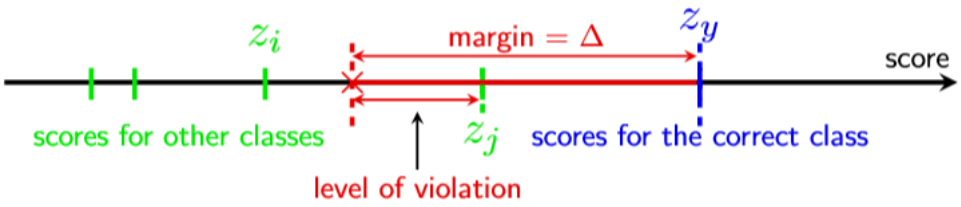
Trong Multi-class SVM, ta đi tìm hệ số **W** bà vector bias **b** sao cho với mỗi điểm dữ liệu **x**, vector $\mathbf{W}^T\mathbf{x} + \mathbf{b}$ có thành phần cao nhất tại chỉ số tương ứng với nhãn của **x**. Hàm hinge loss được sử dụng là hàm mất mát. Thuật toán tối ưu cũng dựa trên GD

4.2 Xây dựng hàm mất mát

Hinge loss tổng quát cho multi-class SVM

Multi-class SVM xây dựng hàm mất mát trên định nghĩa biên an toàn, ép thành phần ứng với corect class của score vector lớn hơn các phần tử khác, và lớn hơn một đại lượng $\Delta > 0$ gọi là biên an toàn.

Nếu correct class lớn hơn các score khác một khoảng Δ thì k có mất mát nào xảy ra, nếu khoảng lớn hơn k đủ Δ thì điểm đó sẽ bị phạt, vi phạm càng nhiều, giá trị phạt càng lớn.



Tóm lại, với một score z_j, j != y, loss gây ra là:

$$\max(0, \Delta - z_y + z_j) = \max(0, \Delta - \mathbf{w}_y^T\mathbf{x} + \mathbf{w}_j^T\mathbf{x})$$

Với một điểm dữ liệu $\mathbf{x}_n$, n = 1, 2, …, N. Tổng loss sẽ là

$$\mathcal{L}_n = \sum_{j \neq y_n} \max(0, \Delta - z_{y_n}^n + z_j^n)$$

Vậy, với toàn bộ các điểm dữ liệu $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N]$, loss được định nghĩa là:

$$\mathcal{L}(\mathbf{X}, \mathbf{y}, \mathbf{W}) = \frac{1}{N}\sum_{n=1}^{N}\sum_{j \neq y_n} \max(0, \Delta - z_{y_n}^n + z_j^n)$$

Regularization

Dùng *weight decay*, sử dụng l_2

$$\mathcal{L}(\mathbf{X}, \mathbf{y}, \mathbf{W}) = \underbrace{\frac{1}{N}\sum_{n=1}^{N}\sum_{j \neq y_n} \max(0, \Delta - \mathbf{w}_{y_n}^T\mathbf{x}_n + \mathbf{w}_j^T\mathbf{x}_n)}_{\text{data loss}} + \underbrace{\frac{\lambda}{2}\|\mathbf{W}\|_F^2}_{\text{regularization loss}}$$

Trên thực tế, giá trị $\Delta = 1$ không ảnh hưởng nhiều tới chất lượng của nghiệm, nên thường được chọn