**1.Giới thiệu.**

**\*Khái niệm SVM:**

SVM là một thuật toán học máy thuộc nhóm học có giám sát (suppervised learning)

SVM là phương pháp học máy mà dữ liệu huấn luyện đã có nhãn sẵn, mỗi điểm dữ liệu đi kèm với một nhãn(label) cho biết nó thuộc lớp nào.

SVM được sử dụng chủ yếu cho bài toán phân loại nhưng cũng có thể được mở rộng cho hồi quy.

**\*Mục tiêu của SVM là:**

- Tối đa hóa khoảng cách từ siêu phẳng đến các điểm gần nhất của mỗi lớp ( gọi là support vector)

- Khoảng cách này càng lớn thì mô hình càng được coi là “ tổng quát hóa tốt” và có khả năng phân loại chính xác dữ liệu mới.

- Siêu phẳng là một bề mặt phân chia không gian thành hai phần , đóng vai trò là ranh giới giữa các lớp dữ liệu trong bài toán SVM

* 1. **Khoảng cách từ 1 điểm tới 1 siêu mặt phẳng.**
* **Công thức tính khoảng cách theo Decart**

Tổng quát với không gian nhiều chiều:

Trong đó: mẫu = d: là số chiều của không gian.

**1.2.Bài toán phân chia hai classes .**

Giả sử có hai class khác nhau được mô tả bởi các điểm trong không gian nhiều chiều, hai classes này có thể tách rời tuyến tính tức là tồn tại một siêu mặt phẳng phân chia chính xác hai classes đó. Thuật toán PLA có thể làm được việc này nhưng sẽ cho ta vô số nghiệm.

Ảnh có chứa ảnh chụp màn hình, hàng, biểu đồ, Sơ đồ

Nội dung do AI tạo ra có thể không chính xác.

Vì vậy, câu hỏi đặt ra là đâu là mặt phân chia tốt nhất theo 1 tiêu chuẩn nào đó? Vì vậy ta có bài toán tối ưu trong Support Vector Machine(SVM) chính là bài toán đi tìm đường phân chia sao cho margin là lớn nhất( khoảng cách từ mặt phân tới 2 margin là lớn nhất).

**Mục tiêu của chúng ta là tìm siêu phẳng có lề hay khoảng cách tới các điểm của 2 lớp là lớn nhất**

Ảnh có chứa hàng, biểu đồ, ảnh chụp màn hình, Sơ đồ

Nội dung do AI tạo ra có thể không chính xác.

**2.Xây dựng bài toán tối ưu cho SVM.**

Giả sử rằng các cặp dữ liệu của tranning set là (x1, y1), (x2, y2) ,…., (xN, yN) với vector xi thuộc thể hiện đầu vào của một điểm dữ liệu và yi là nhãn của điểm dữ liệu đó. “d” là số chiều của dữ liệu và N là số điểm dữ liệu. Giả sử rằng nhãn của mỗi điểm dữ liệu được xác định bởi yi = 1( bên đỏ) hoặc yi = -1 (bên xanh) giống như PLA .

Ảnh có chứa ảnh chụp màn hình, văn bản, biểu đồ, hàng

Nội dung do AI tạo ra có thể không chính xác.

Mặt phẳng phân chia : + + b = 0 , chúng ta cần đi tìm hệ số w,b

Bài toán tối ưu trong SVM chính là bài toán tìm w,b sao cho margin đạt giá trị lớn nhất.

(w, b) = argmax = argmax

Ảnh có chứa văn bản, biểu đồ, hàng, ảnh chụp màn hình

Nội dung do AI tạo ra có thể không chính xác.

**Trong đó**:

- Đường ở giữa chính là ngưỡng phân tách dữ liệu.

- Từ giữa tới đường nét đứt gọi là lề.

- Chọn support vector là 2 điểm lân cận gần middle nhất và không tính điểm được cho phép phân loại sai.

=> Chọn support vector cho phép chúng ta cân bằng sai số,

+ bias (độ lệch – mức độ giả định đơn giản hóa mà mô hình tạo ra so với thực tế),

+ variance(phương sai – độ nhạy của mô hình với sự thay đổi của dữ liệu huấn luyện)

**3.Phương pháp SVM trong bài toán phân loại**

**a) hard margin**

**Khái niệm**: là phương pháp yêu cầu dữ liệu phải hoàn toàn phân tách tuyến tính, tức là sẽ không có điểm nào nằm sai phía siêu phẳng hoặc nằm trong vùng margin.( cx chỉ dùng cho nhị phân)

**Ưu điểm**: khi dữ liệu hoàn toàn tách được, hard margin sẽ cho mô hình đơn giản và tối ưu

**Mục tiêu**: Phân chia được hai lớp dữ liệu .

Khoảng cách giữa siêu phẳng và các điểm gần nhất là lớn nhất.

Ảnh có chứa hàng, biểu đồ, văn bản, Phông chữ

Nội dung do AI tạo ra có thể không chính xác.

**Điều kiện phân loại đúng:**

, y = 1, -1 là nhãn của dữ liệu

**Hàm mục tiêu:** mục tiêu tối đa hóa khoảng cách giữa hai biên(margin), tương đương với tối thiểu hóa

Bài toán tối ưu:

Với ràng buộc:

**Ta dùng phương pháp nhân tử Lagrange để đưa ràng buộc vào hàm tối ưu**:

Tại sao dùng phương pháp nhân tử lagrange:

Mục tiêu: đưa điều kiện ràng buộc vào bên trong hàm tối ưu, từ đó giải bài toán ko có ràng buộc nữa sẽ dễ tính đạo hàm và giải hơn.

Giải thích:

nhân tử lagrange ứng với ràng buộc thứ i.

N tổng số điểm trong tập huấn luyện.

W vecto trọng số của siêu phẳng phân loại

**Giải bài toán đối ngẫu :**

Từ hàm lagrange đó, ta:

+ Lấy đạo hàm theo w và b để tìm điểm yên ngựa

+ Thế ngược lại vào hàm để loại bỏ w và b => thu được bài toán chỉ theo

Khi đó thu được bài toán đối ngẫu như sau:

Ràng buộc

**b)Soft margin**

**Khái niệm:** soft margin chấp nhận một số lỗi phân loại(sai số) bằng cách cho phép một số điểm dữ liệu nằm giữa hai lớp hoặc trên đường biên. Những điểm này được gọi là các biến slack.

**Ưu điểm:** linh hoạt hơn và có thể hoạt động tốt hơn trên dữ liệu thực tế ( có thể dùng cho đa lớp)

**Nhược điểm:** cần phải điều chỉnh tham số để cân bằng giữa việc phân chia chính xác và khả năng chấp nhận lỗi.

**Sự cho phép vi phạm một số điểm dữ liệu giúp giảm thiểu overfitting (quá khớp) và làm cho mô hình SVM linh hoạt hơn trong việc xử lý các tập dữ liệu có độ phức tạp cao hơn.**

Một đại lượng quan trọng và để kiểm soát mức độ sai sót của Soft Margin SVM là mức độ chấp nhận lỗi (hay tham số phạt). Khi tham số phạt càng lớn nghĩa là SVM bị phạt càng nặng khi phân loại sai và kết quả sẽ cho ra 1 lề ( margin) hẹp và ngược lại.

Ảnh có chứa văn bản, biểu đồ, ảnh chụp màn hình, bản đồ

Nội dung do AI tạo ra có thể không chính xác.

Với những điểm nằm trong vùng an toàn

Với mỗi điểm trong tập dữ liệu huấn luyện, ta có thêm 1 biến đo sự hy sinh tương ứng.

Hàm mục tiêu cho soft margin:

Trong đó: C là 1 hằng số dương dùng để điều chỉnh tầm quan trọng giữa margin và điểm hy sinh.

* Điều kiện ràng buộc mềm:
* Từ đây áp dụng các phương pháp giải toán ta tìm đk w,b và đưa ra siêu phẳng tương ứng tối ưu.

Ảnh có chứa biểu đồ, ảnh chụp màn hình, hàng, văn bản

Nội dung do AI tạo ra có thể không chính xác.

Ảnh có chứa biểu đồ, vòng tròn

Nội dung do AI tạo ra có thể không chính xác.

* Vấn đề của bài toán là phân loại nên đặt màu đỏ là 1 và màu xanh là -1 ( ko có ảnh hưởng gì), nó vẫn là 2 loại .
* Chúng ta cần tìm 1 đường tối ưu phân tách 2 nhóm dữ liệu.
* Có 1 đường hyperplane tối ưu có dạng ( wx + b = 0) thỏa mãn:

Giải thích:

1. Với tất cả những điểm y = 1 hay là các điểm màu đỏ thì pt thỏa mãn
2. Tương tự với

Ảnh có chứa hàng, biểu đồ, Sơ đồ, Phông chữ

Nội dung do AI tạo ra có thể không chính xác.

Hai đường thẳng đi qua 2 điểm gần hyperplane nhất sẽ có giá trị bằng 1 và -1

Ảnh có chứa hàng, biểu đồ, văn bản, Phông chữ

Nội dung do AI tạo ra có thể không chính xác.

* Khoảng cách để hyperplane nằm ở giữa.
* Tính khoảng cách từ và :

* Maximum margin Minimizing

**\* Dùng hinge loss: (hàm tổn thất bản lề)**

**-** Dùng cho bài toán phân loại nhị phân. ( chỉ có 2 lớp dự đoán)

**-** Dùng khi dl tương đối sạch (ít nhiễu)

**-** Có thể phân tách tuyến tính rõ ràng

* Hàm L : L(y, f(x)) = max(0, 1 – y\*f(x))
* Được tính bằng giá trị tối đa giữa 0 và phần còn lại của hàm này.
* “y” là giá trị thực output ( +1 or -1)
* “f(x)” là khoảng cách điểm dữ liệu đến hyperplane: f(x) = wx +b
* :
  + Nếu y\*f(x) < 1, then 1 – y\*f(x) > 0 and L = 1 – y\*f(x)
  + Nếu y\*f(x) >1, then 1 – y\*f(x) < 0 and L = 0
  + Nếu y\* f(x) = 1, then 1 – y\*f(x) = 0 and L = 0

Ảnh có chứa văn bản, ảnh chụp màn hình, hàng, Phông chữ

Nội dung do AI tạo ra có thể không chính xác.

Ảnh có chứa văn bản, hàng, biểu đồ, ảnh chụp màn hình

Nội dung do AI tạo ra có thể không chính xác.

Bên phải là phân loại đúng.

Bên trái là phân loại sai.

Ảnh có chứa hàng, biểu đồ, Sơ đồ, ảnh chụp màn hình

Nội dung do AI tạo ra có thể không chính xác.

Như vậy, có thể vẽ 1 đường đi qua các điểm và cắt trục tung 1 và điểm mà nằm về phía phân loại sai thì giá trị lại càng lớn(có thể dùng hinge loss để traning – tiêu chuẩn để chọn hinge tốt nhất)

L càng lớn , phân loại sai.

L càng nhỏ, phân loại đúng.

Vậy mục tiêu của SVM là giảm thiểu tổng số tổn thất bản lề trên tát cả điểm dữ liệu, đồng thời cố gắng tối đa hóa biên độ.

SVM sẽ giảm thiểu:

Luôn luôn tìm bề mặt hyperplane có tổng này nhỏ nhất

**Trong đó**: tổng đầu là cái lề của nó, w càng nhỏ càng tốt( lề càng lớn càng tốt)

Tổng thứ 2 là tổng tất cả các điểm dl.

C là thông số điều chỉnh giữa lề tối đa và hinge loss.

Tăng giá trị của lề thì C chọn nhỏ. ( nhiều khi bị phân loại sai khi lề lớn).

Giảm giá trị của lề thì C chọn lớn.

* Từ đây sẽ áp dụng các cách giải toán để tìm ra w và b:
  + Dùng gradient descent (code tay)
  + Dùng thư viện sẵn có.
* Cuối cùng sẽ vẽ được siêu phẳng cần tìm.

Hàm mục tiêu cho soft margin:

Trong đó: C là 1 hằng số dương dùng để điều chỉnh tầm quan trọng giữa margin và điểm hy sinh.

* Điều kiện ràng buộc mềm:
* Từ đây áp dụng các phương pháp giải toán ta tìm đk w,b và đưa ra siêu phẳng tương ứng tối ưu.

Ảnh có chứa biểu đồ, ảnh chụp màn hình, hàng, văn bản

Nội dung do AI tạo ra có thể không chính xác.

**4.Mở rộng SVM cho bài toán hồi quy:**

- Support vector regression là 1 phương pháp hồi quy dựa trên nguyên lý của SVM.

- SVR dự đoán giá trị các giá trị liên tục.

- SVR là 1 thuật toán học máy được sử dụng để phân tích hồi quy.

- SVR được thiết kế để **dự đoán các giá trị liên tục** bằng cách tìm một hàm hồi quy sao cho sai số của dự đoán nằm trong một khoảng chấp nhận được (gọi là **epsilon-tube**)

- Không giống như SVM cho bài toán phân loại, mô hình SVR tìm kiếm 1 siêu phẳng phù hợp nhất với các điểm dl trong 1 không gian liên tục. Điều này đạt được bằng cách ánh xạ các biến đầu vào thành một không gian tính năng chiều cao và tìm thấy siêu phẳng tối đa hóa lề( khoảng cách) giữa các điểm tăng áp và các điểm dữ liệu gần nhất, đồng thời giảm thiểu lỗi dự đoán.

Ảnh có chứa hàng, biểu đồ, văn bản, Sơ đồ

Nội dung do AI tạo ra có thể không chính xác.

2 dòng màu đỏ là 2 ranh giới quyết định, đường màu xanh lá là cây siêu phẳng, mục tiêu là xem xét các điểm trong dòng ranh giới quyết định.

Dòng phù hợp nhất là siêu phẳng với số điểm tối đa.

Trong SVM thì ranh giới quyết định là những đường nằm song song vs siêu phẳng và cách 1 khoảng nhất định.

* **Mục tiêu**:
  + Các sai số nằm trong khoảng epsilon sẽ **không bị phạt.**
  + Các sai số vượt ra ngoài epsilon sẽ bị phạt (đưa vào hàm mất mát)
* Trong SVR ranh giới quyết định là biên bên trên và biên bên dưới của dải sai số epsilon.
* Giả sử siêu phẳng có phương trình: y = wx + b
* Khi đó ranh giới có pt: wx + b = +a

wx + b = -a

* Do đó bất kì siêu phẳng nào thỏa mãn mô hình **hồi quy SVM** sẽ thỏa mãn:

1. Xây dựng ống epsilon – toán học của sai số chấp nhận được.
   * Mô hình hồi quy: f(x) = wx + b
   * Sai số:

Giải thích: nếu dự đoán của mô hình f(xi) và giá trị thực yi cashc nhau nhỏ hơn hoặc bằng epsion thì ko bị phạt.

* + Các điểm dữ liệu nằm trong khoảng :
    - và

1. Tối ưu hóa hàm mục tiêu.

Minimize:

* Giải thích:

C: là tham số điều chỉnh mức phạt cho các điểm vượt epsilon.

là sai số vượt ngoài epsilon ở hai phía.

* Ràng buộc:

( Nếu điểm dữ liệu nằm ngoài ống epsilon -> sai số được tính vào Ei or Ei\*. SVR cố gằng tối ưu cả độ phẳng của mô hình và số lượng sai số vượt epsilon)

Ảnh có chứa văn bản, ảnh chụp màn hình, biểu đồ, hàng

Nội dung do AI tạo ra có thể không chính xác.

Ảnh có chứa hàng, biểu đồ, ảnh chụp màn hình, bản đồ

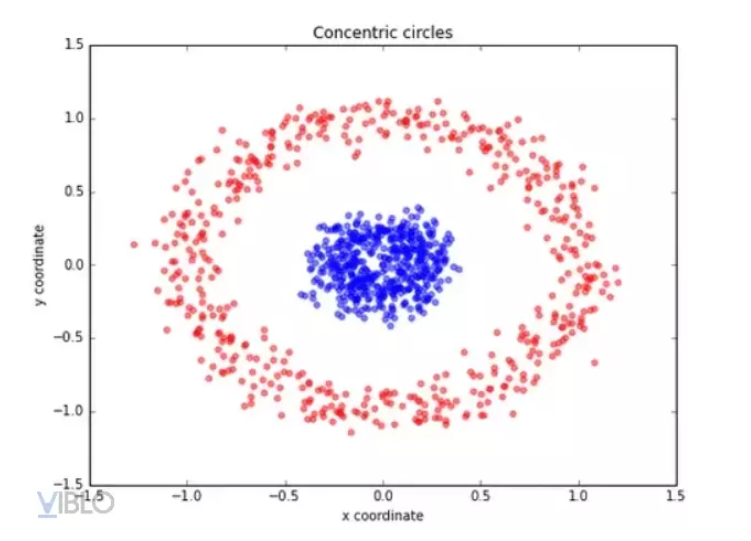
Nội dung do AI tạo ra có thể không chính xác.

\*so sánh hồi quy và phân loại

Ảnh có chứa biểu đồ, văn bản, ảnh chụp màn hình, hàng

Nội dung do AI tạo ra có thể không chính xác.

**5. Kernel trong SVM**

****

là một phiên bản mở rộng của SVM cho phép xử lý các tập dữ liệu không tuyến tính bằng cách sử dụng hàm nhân (kernel function).

Thông qua quá trình **feature mapping ( biến đổi bộ dữ liệu ban đầu thông qua một hàm nào đó)**, chúng ta sử dụng bộ dữ liệu đã qua xử lí( ko dùng bộ dữ liệu ban đầu) để tìm ra **hyperplane**

Thay vì tìm một siêu phẳng trong không gian ban đầu, Kernel SVM ánh xạ dữ liệu vào không gian nhiều chiều hơn thông qua hàm nhân và sau đó tìm một siêu phẳng phân chia tuyến tính trong không gian mới.

Ảnh có chứa biểu đồ, hàng, nghệ thuật gấp giấy origami, thiết kế

Nội dung do AI tạo ra có thể không chính xác.

1. **Polynomial kernel**

**Ví dụ:** input a,b sau khi bình phương có tọa độ không gian 2 chiều là (

Ảnh có chứa văn bản, ảnh chụp màn hình, hàng, Sơ đồ

Nội dung do AI tạo ra có thể không chính xác.

Muốn tìm mối quan hệ giữa 2 điểm thì ta xét tích vô hường của cặp dữ liệu sau khi đã biến đổi.

=

- Dùng hàm Polynomial Kernel để tính, hàm có dạng:

Trong đó: a nhân vô hướng với b.

+ d là bậc của hàm

+ c là tham số cta tự định nghĩa.

Chọn c = ½, d = 2 thì hàm sẽ là:

=

Nhận xét: kết quả là tích vô hướng của điểm dữ liệu 3 chiều nhưng nằm trên cùng 1 bề mặt z = ½

Kết luận: tìm được siêu phẳng phân tách dữ liệu có z = ½

**b)RBF kernel ( radial basic function kernel)**

Hay còn gọi là gaussian kernel là 1 loại hàm phổ biến trong svm.

Nó cho phép ánh xạ dữ liệu từ không gian gốc sang không gian đặc trưng có chiều cao hơn, giúp phân tách dữ liệu phi tuyến.

* Khi nào nên dùng rbf:
  + Dữ liệu ko thể phân tách tuyến tính.( biên cong phức tạp)
  + Ko bt trc hình dạng phân tách
  + Muốn độ tối ưu chính xác cao nhất
  + Sẵn sàng chấp nhận chi phí tính toán cao
* Công thức:

Trong đó: xi, xj là 2 vector đặc trưng

|| ….|| là khoảng cách eulidean bình phương giữa 2 điểm

Gamma: tham số quan trọng của kernel RBF, quyết định độ ảnh hưởng của 1 điểm dữ liệu.

+ Nếu thì ảnh hưởng của từng điểm nhỏ, mô hình rất phức tạp, dề overfitting.

+ Nếu thì ảnh hưởng lan rộng có thể underfitting

* Hàm mục tiêu của RBF
* Ảnh có chứa ảnh chụp màn hình

  Nội dung do AI tạo ra có thể không chính xác.

Hình cho thấy sự ảnh hưởng của gamma và mô hình rbf