*M* *ÔN KĨ THUẬT TRÍ TUỆ NHÂN TẠO*

**Support Vector Machine Algorithm**

***GIÁO VIÊN HƯỚNG DẪN:***

1. Nguyễn Ngọc Thảo

***NH ÓM SV TH ỰC HI ỆN:***

***HBGroup:***

1. 1512031 - Huỳnh Cao Biên
2. 1312005 - Nguyễn Hoàng Anh

LogoTruong

Khoa Công nghệ Thông tin

Đại học Khoa học Tự nhiên TP HCM

Tháng 5/2018

1. Khái niệm và các công thức liên quan
2. Động lực nghiên cứu
3. Công trình nghiên cứu liên quan
4. Phát biểu bái toán  
   Giả sử rằng các cặp dữ liệu của training set là (x1,y1), (x2,y2),…, (xN,yN) với vector xi ∈ Rd 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 NN là số điểm dữ liệu. Giả sử rằng nhãn của mỗi điểm dữ liệu được xác định bởi yi = 1 (class 1) hoặc yi = -1 (class 2).

Giả sử rằng các điểm vuông xanh thuộc class 1, các điểm tròn đỏ thuộc class -1 và mặt wTx+b=w1x1+ w2x2+b=0 là mặt   
Để giúp các bạn dễ hình dung, chúng ta cùng xét trường hợp trong không gian hai chiều dưới đây. Không gian hai chiều để các bạn dễ hình dung, các phép toán hoàn toàn có thể được tổng quát lên không gian nhiều chiều.

1. Hướng giải quyết
2. Coding demo

Trước tiên chúng ta gọi các *modules* cần dùng và tạo dữ liệu giả (dữ liệu này chính là dữ liệu tôi dùng trong các hình phía trên nên chúng ta biết chắc rằng hai classes là *linearly separable*):

**from** \_\_future\_\_ **import** print\_function

**import** numpy **as** np

**import** matplotlib.pyplot **as** plt

**from** scipy.spatial.distance **import** cdist

np**.**random**.**seed(22)

means **=** [[2, 2], [4, 2]]

cov **=** [[**.**3, **.**2], [**.**2, **.**3]]

N **=** 10

X0 **=** np**.**random**.**multivariate\_normal(means[0], cov, N) *# class 1*

X1 **=** np**.**random**.**multivariate\_normal(means[1], cov, N) *# class -1*

X **=** np**.**concatenate((X0**.**T, X1**.**T), axis **=** 1) *# all data*

y **=** np**.**concatenate((np**.**ones((1, N)), **-**1**\***np**.**ones((1, N))), axis **=** 1) *# labels*

Tiếp theo, chúng ta giải bài toán (9)(9) bằng CVXOPT:

**from** cvxopt **import** matrix, solvers

*# build K*

V **=** np**.**concatenate((X0**.**T, **-**X1**.**T), axis **=** 1)

K **=** matrix(V**.**T**.**dot(V)) *# see definition of V, K near eq (8)*

p **=** matrix(**-**np**.**ones((2**\***N, 1))) *# all-one vector*

*# build A, b, G, h*

G **=** matrix(**-**np**.**eye(2**\***N)) *# for all lambda\_n >= 0*

h **=** matrix(np**.**zeros((2**\***N, 1)))

A **=** matrix(y) *# the equality constrain is actually y^T lambda = 0*

b **=** matrix(np**.**zeros((1, 1)))

solvers**.**options['show\_progress'] **=** False

sol **=** solvers**.**qp(K, p, G, h, A, b)

l **=** np**.**array(sol['x'])

**print**('lambda = ')

**print**(l**.**T)

Kết quả:

lambda =

[[ 8.54018321e-01 2.89132533e-10 1.37095535e+00 6.36030818e-10

4.04317408e-10 8.82390106e-10 6.35001881e-10 5.49567576e-10

8.33359230e-10 1.20982928e-10 6.86678649e-10 1.25039745e-10

2.22497367e+00 4.05417905e-09 1.26763684e-10 1.99008949e-10

2.13742578e-10 1.51537487e-10 3.75329509e-10 3.56161975e-10]]

Ta nhận thấy rằng hầu hết các giá trị của lambda đều rất nhỏ, tới 10−910−9 hoặc 10−1010−10. Đây chính là các giá trị bằng 0 nhưng vì sai số tính toán nên nó khác 0 một chút. Chỉ có 3 giá trị khác 0, ta dự đoán là sẽ có 3 điểm là *support vectors*.

Ta đi tìm *support set* SS rồi tìm nghiệm của bài toán:

epsilon **=** 1e-6 *# just a small number, greater than 1e-9*

S **=** np**.**where(l **>** epsilon)[0]

VS **=** V[:, S]

XS **=** X[:, S]

yS **=** y[:, S]

lS **=** l[S]

*# calculate w and b*

w **=** VS**.**dot(lS)

b **=** np**.**mean(yS**.**T **-** w**.**T**.**dot(XS))

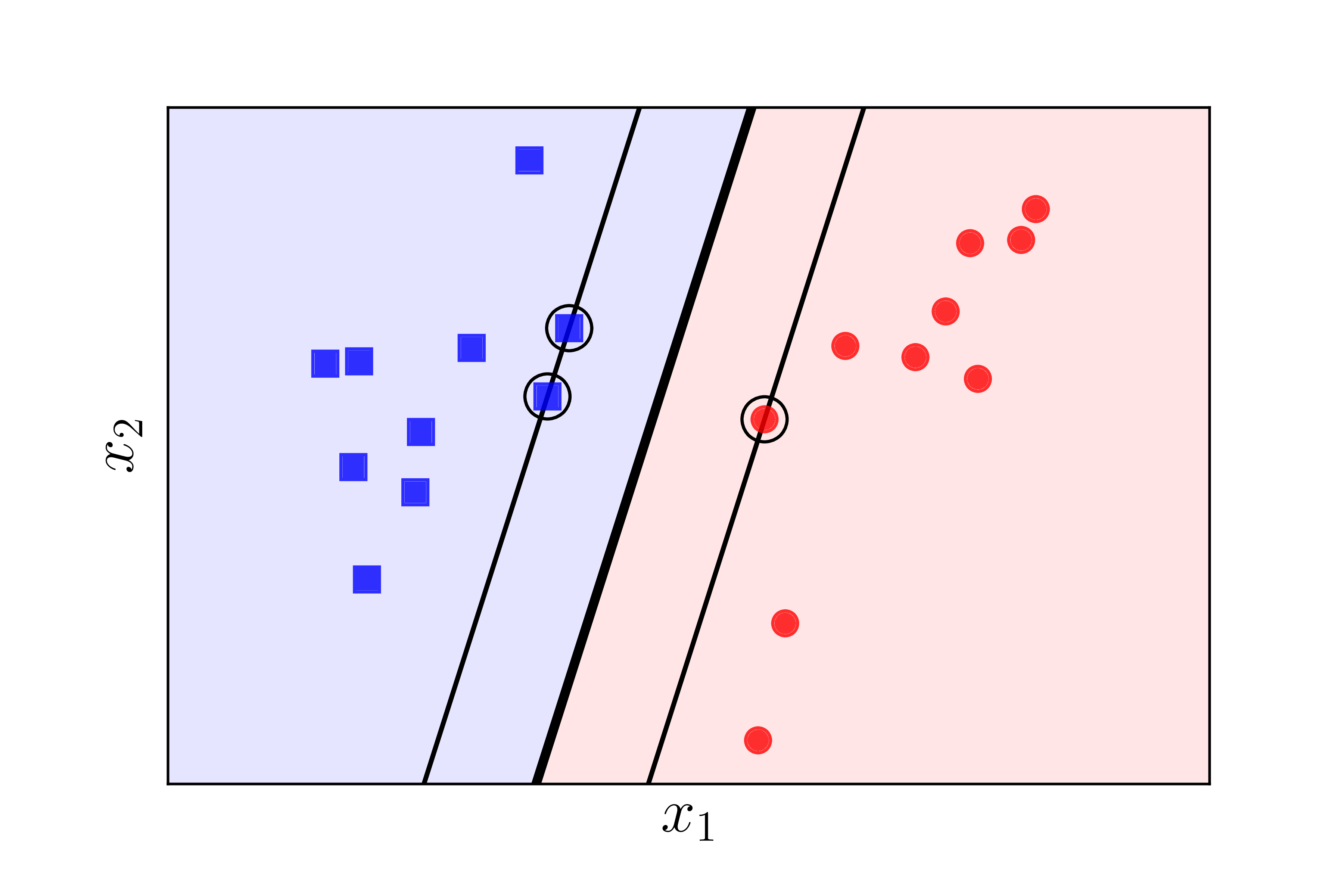
**print**('w = ', w**.**T)

**print**('b = ', b)

w = [[-2.00984381 0.64068336]]

b = 4.66856063387

Minh hoạ kết quả:



Hình 5: Minh hoạ nghiệm tìm được bởi SVM.

Đường màu đen đậm ở giữa chính là mặt phân cách tìm được bằng SVM. Từ đây có thể thấy *nhiều khả năng là các tính toán của ta là chính xác*. Để kiểm tra xem các tính toán phía trên có chính xác không, ta cần tìm nghiệm bằng các công cụ có sẵn, ví dụ như sklearn.

Source code có đính kèm trong thư mục demo hoặc link nguồn tham khảo: https://github.com/tiepvupsu/tiepvupsu.github.io/blob/master/assets/19\_svm/plt/SVM-example.ipynb

1. Ưu điểm và nhược điểm
2. Tài liệu tham khảo