

ECE595 / STAT598: Machine Learning I

Lecture 19.2: Support Vector Machine (SVM)

Spring 2020

Stanley Chan

School of Electrical and Computer Engineering
Purdue University



Outline

Support Vector Machine

- **Lecture 19 SVM 1: The Concept of Max-Margin**
- Lecture 20 SVM 2: Dual SVM
- Lecture 21 SVM 3: Kernel SVM

This lecture: Support Vector Machine 1

- Concept of Margin
 - Distance from point to plane
 - Margin
 - Max Margin Classifier
- **SVM**
 - **SVM via Optimization**
 - **Programming SVM**
 - **Visualization**

Unfortunately ...

- If I can solve the optimization problem

$$\begin{aligned} & \underset{\mathbf{w}, w_0}{\text{maximize}} \quad \gamma \\ & \text{subject to} \quad y_j \left(\frac{\mathbf{w}^T \mathbf{x}_j + w_0}{\|\mathbf{w}\|_2} \right) \geq \gamma, \quad j = 1, \dots, N. \end{aligned}$$

Then I can obtain a good SVM.

- But solving the optimization is not easy!
- γ depends on (\mathbf{w}, w_0) . If you change (\mathbf{w}, w_0) , you also change γ
- There is a term $1/\|\mathbf{w}\|_2$. Nonlinear.

Trick 1: Scaling

- The optimization is

$$\begin{aligned} & \underset{\mathbf{w}, w_0}{\text{maximize}} \quad \gamma \\ & \text{subject to} \quad y_j \left(\frac{\mathbf{w}^T \mathbf{x}_j + w_0}{\|\mathbf{w}\|_2} \right) \geq \gamma, \quad j = 1, \dots, N. \end{aligned}$$

- Let \mathbf{x}_N be the point closest to the boundary
- Define the smallest **unsigned** distance

$$\tilde{\gamma} \stackrel{\text{def}}{=} |\mathbf{w}^T \mathbf{x}_N + w_0|$$

- Then, we can show that

$$\gamma \stackrel{\text{def}}{=} \frac{|\mathbf{w}^T \mathbf{x}_N + w_0|}{\|\mathbf{w}\|_2} = \frac{\tilde{\gamma}}{\|\mathbf{w}\|_2}.$$

Trick 1: Scaling

- So we can turn this optimization

$$\begin{aligned} & \underset{\mathbf{w}, w_0}{\text{maximize}} \quad \gamma \\ & \text{subject to} \quad y_j \left(\frac{\mathbf{w}^T \mathbf{x}_j + w_0}{\|\mathbf{w}\|_2} \right) \geq \gamma, \quad j = 1, \dots, N. \end{aligned}$$

- into this optimization

$$\begin{aligned} & \underset{\mathbf{w}, w_0}{\text{maximize}} \quad \frac{\tilde{\gamma}}{\|\mathbf{w}\|_2} \\ & \text{subject to} \quad y_j \left(\frac{\mathbf{w}^T \mathbf{x}_j + w_0}{\|\mathbf{w}\|_2} \right) \geq \frac{\tilde{\gamma}}{\|\mathbf{w}\|_2}, \quad j = 1, \dots, N. \end{aligned}$$

- $1/\|\mathbf{w}\|_2$ goes to objective function!

Eliminate $\tilde{\gamma}$

- How about we turn this optimization

$$\begin{aligned} & \underset{\mathbf{w}, w_0}{\text{maximize}} \quad \tilde{\gamma} \\ & \text{subject to} \quad y_j(\mathbf{w}^T \mathbf{x}_j + w_0) \geq \tilde{\gamma}, \quad j = 1, \dots, N. \end{aligned}$$

- into this optimization?

$$\begin{aligned} & \underset{\frac{\mathbf{w}}{\tilde{\gamma}}, \frac{w_0}{\tilde{\gamma}}}{\text{maximize}} \quad \frac{1}{\left\| \frac{\mathbf{w}}{\tilde{\gamma}} \right\|_2} \\ & \text{subject to} \quad y_j \left(\frac{\mathbf{w}}{\tilde{\gamma}}^T \mathbf{x}_j + \frac{w_0}{\tilde{\gamma}} \right) \geq 1, \quad j = 1, \dots, N. \end{aligned}$$

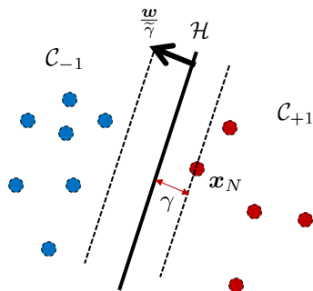
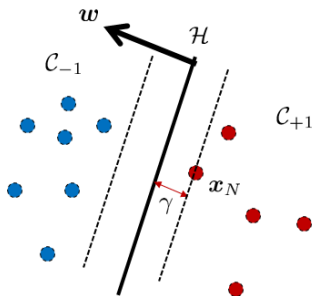
- You can refine the variables $\mathbf{w} \leftarrow \frac{\mathbf{w}}{\tilde{\gamma}}$ and $w_0 \leftarrow \frac{w_0}{\tilde{\gamma}}$

Eliminate $\tilde{\gamma}$

- This gives you

$$\begin{aligned} & \underset{\mathbf{w}, w_0}{\text{maximize}} && \frac{1}{\|\mathbf{w}\|_2} \\ & \text{subject to} && y_j(\mathbf{w}^T \mathbf{x}_j + w_0) \geq 1, \quad j = 1, \dots, N. \end{aligned}$$

- So $\tilde{\gamma}$ is eliminated!
- Geometrically: A scaling



Trick 2: Max to Min

- You want to solve

$$\begin{aligned} & \underset{\mathbf{w}, w_0}{\text{maximize}} && \frac{1}{\|\mathbf{w}\|_2} \\ & \text{subject to} && y_j(\mathbf{w}^T \mathbf{x}_j + w_0) \geq 1, \quad j = 1, \dots, N. \end{aligned}$$

- How about

$$\begin{aligned} & \underset{\mathbf{w}, w_0}{\text{minimize}} && \frac{1}{2} \|\mathbf{w}\|^2 \\ & \text{subject to} && y_j(\mathbf{w}^T \mathbf{x}_j + w_0) \geq 1, \quad j = 1, \dots, N. \end{aligned}$$

- This is a **quadratic minimization** with **linear constraint**.
- Convex.
- Solution is called a **support vector machine**.

Hand Crafted Example

- You have four data points

$$\mathbf{x}_1 = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad \mathbf{x}_2 = \begin{bmatrix} 2 \\ 2 \end{bmatrix} \quad \mathbf{x}_3 = \begin{bmatrix} 2 \\ 0 \end{bmatrix} \quad \mathbf{x}_4 = \begin{bmatrix} 3 \\ 0 \end{bmatrix}$$

- Labels are

$$y_1 = -1, y_2 = -1, y_3 = +1, y_4 = +1.$$

- Weight vector $\mathbf{w} = (w_1, w_2)$ and off-set w_0 .
- The constraints are $y_j(\mathbf{w}^T \mathbf{x}_j + w_0) \geq 1$:

$$-w_0 \geq 1 \quad (i)$$

$$-(2w_1 + 2w_2 + w_0) \geq 1 \quad (ii)$$

$$2w_1 + w_0 \geq 1 \quad (iii)$$

$$3w_1 + w_0 \geq 1 \quad (iv)$$

- Combine (i) and (iii): $w_1 \geq 1$
- Combine (ii) and (iii): $w_2 \leq -1$

Hand Crafted Example

- Combine (i) and (iii): $w_1 \geq 1$
- Combine (ii) and (iii): $w_2 \leq -1$
- Objective function is $\frac{1}{2} \|\mathbf{w}\|^2$.

- Can show that

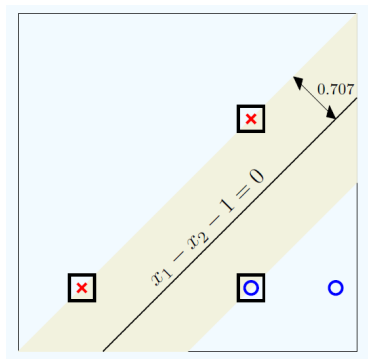
$$\frac{1}{2} \|\mathbf{w}\|^2 = \frac{1}{2}(w_1^2 + w_2^2) \geq 1.$$

- Equality holds when

$$w_1^* = 1, \quad w_2^* = -1.$$

- So $(w_1^*, w_2^*) = (1, -1)$ is a minimizer of the objective.
- Can further show that $w_0^* = -1$.
- All constraints are satisfied at $(w_1^*, w_2^*, w_0^*) = (1, -1, -1)$.

Hand Crafted Example



- Separating hyperplane:

$$h(\mathbf{x}) = \text{sign}(x_1 - x_2 - 1)$$

- Margin:

$$\frac{1}{\|\mathbf{w}^*\|_2} = \frac{1}{\sqrt{2}} = 0.707.$$

- Boxed data points: Meet the constraints (i), (ii) and (iii) with equality.
- These are the **support vectors**.

Writing Your SVM

- The problem is

$$\begin{aligned} & \underset{\mathbf{w}, w_0}{\text{minimize}} && \frac{1}{2} \|\mathbf{w}\|^2 \\ & \text{subject to} && y_j(\mathbf{w}^T \mathbf{x}_j + w_0) \geq 1, \quad j = 1, \dots, N. \end{aligned}$$

- This is a quadratic programming problem:

$$\begin{aligned} & \underset{\mathbf{u}}{\text{minimize}} && \frac{1}{2} \mathbf{u}^T \mathbf{Q} \mathbf{u} + \mathbf{p}^T \mathbf{u} \\ & \text{subject to} && \mathbf{A} \mathbf{u} \geq \mathbf{c} \end{aligned}$$

- Solution is $\mathbf{u}^* = \text{QP}(\mathbf{Q}, \mathbf{p}, \mathbf{A}, \mathbf{c})$. Solvable using any QP solver.
- To us: This is convex objective with convex constraint.
- Use CVX! Below is a 1D example.

Writing Your SVM

```
mu0    = 0;      mu1 = 10;      sigma0 = 1;      sigma1 = 1;
NO     = 20;     N1 = 20;      N = NO+N1;

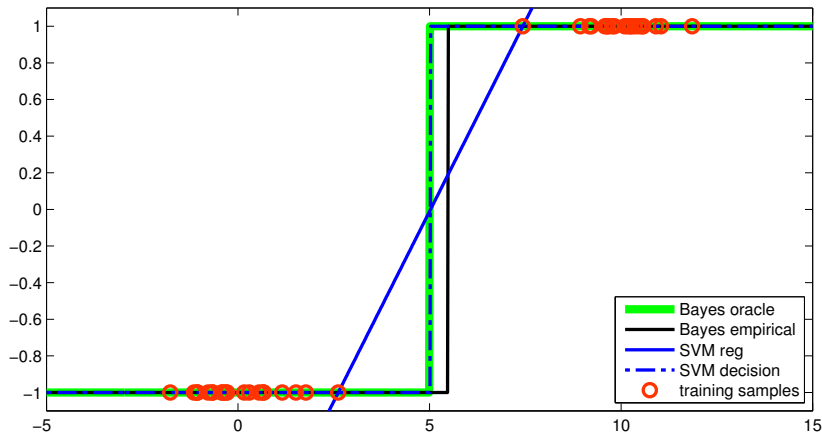
x0 = random('normal',mu0,sigma0,NO,1);
x1 = random('normal',mu1,sigma1,N1,1);

y0 = -ones(NO,1);  y1 = ones(N1,1);
x = [x0; x1];      y = [y0; y1];      b = ones(N,1);

% Solve CVX
cvx_expert true
cvx_begin
    variables w w0
    minimize( sum_square(w) )
    subject to
        y.*(w*x + w0*ones(N,1)) - b >= 0
cvx_end
```

Comparing SVM and Bayesian Oracle

- $\mathcal{N}(0, 1)$ with 20 samples and $\mathcal{N}(10, 1)$ with 20 samples.



- Classifier is always defined by the support vectors!

MATLAB Code for 2D SVM

```
mu0 = [-5; 0]; mu1 = [5; 5];
s = 1.5; Sigma = (s^2)*[1 0; 0 1];

N0 = 50; N1 = 50; N = N0+N1;

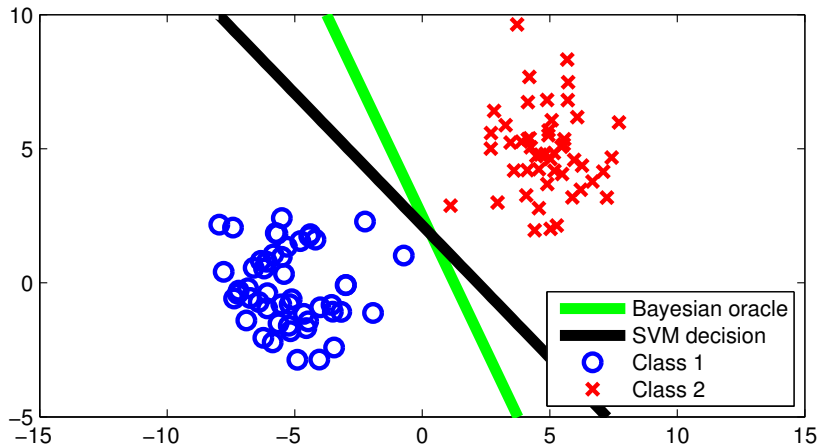
x0 = mvnrnd(mu0,Sigma,N0);
x1 = mvnrnd(mu1,Sigma,N1);
x = [x0; x1];

y0 = -ones(N0,1); y1 = ones(N1,1); y = [y0; y1];
b = ones(N,1);

cvx_expert true
cvx_begin
    variables w(2) w0
    minimize( sum_square(w) )
    subject to
        y.*(x*w + ones(N,1)*w0) - b >= 0
cvx_end
```

SVM in 2D

- $\mu_1 = [-5, 0]^T$, $\mu_2 = [5, 5]^T$. $\Sigma = 1.5^2 I$.
- Class 1: 50 samples. Class 2: 50 samples.



Displaying Results

```
wstar = (mu1-mu0)/s^2;
w0star = -wstar*((mu1+mu0)/2);

figure;
grid = linspace(-10,10,100);
hh{1} = plot( grid, (-w0star-wstar(1)*grid)/wstar(2), 'g', 'LineWidth', 5);
hh{2} = plot( grid, (-w0-w(1)*grid)/w(2), 'k', 'LineWidth', 5);
hh{3} = plot(x0(:,1),x0(:,2),'bo','LineWidth', 2, 'MarkerSize',8);
hh{4} = plot(x1(:,1),x1(:,2),'rx','LineWidth', 2, 'MarkerSize',8);
axis([-15 15 -5 10]);
legend([hh{1:4}], 'Bayesian oracle', 'SVM decision', 'Class 1', 'Class 2',
set(gcf, 'Position', [100, 100, 600, 300]));
```

- How to draw a line with (\mathbf{w}, w_0) ?
- $\mathbf{w}^T \mathbf{x} + w_0 = 0$ implies $w_1 x_1 + w_2 x_2 + w_0 = 0$.
- So $x_2 = -\frac{w_1}{w_2} x_1 - \frac{w_0}{w_2}$.
- Sweep a range of x_1 to get x_2 .
- Make sure to configure the aspect ratio of your plot!

Reading List

Support Vector Machine

- Mustafa, *Learning from Data*, e-Chapter
- Duda-Hart-Stork, *Pattern Classification*, Chapter 5.5
- Chris Bishop, *Pattern Recognition*, Chapter 7.1
- UCSD Statistical Learning
<http://www.svcl.ucsd.edu/courses/ece271B-F09/>