ECE595 / STAT598: Machine Learning I
Lecture 19 Support Vector Machine: Intro

Spring 2020

Stanley Chan

School of Electrical and Computer Engineering
Purdue University

Outline

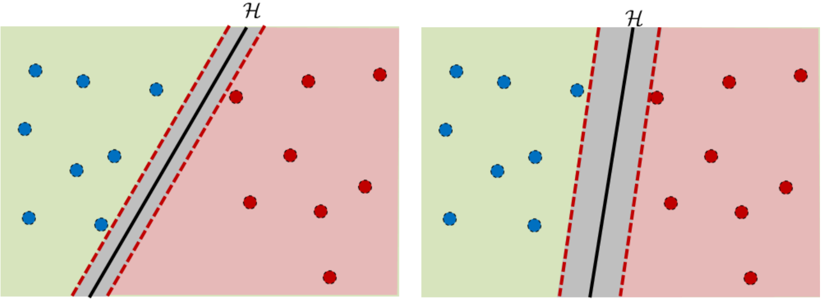
Support Vector Machine

- 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

Margin and Max-Margin Classifier



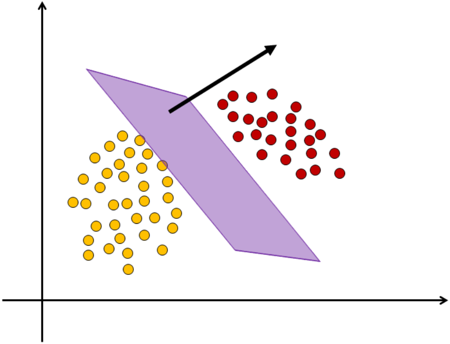
- **Margin**: Smallest gap between the two classes
- **Max-Margin Classifier**: A classifier that maximizes the margin
- **What do we need?**
  - How to measure the distance from a point to a plane?
  - How to formulate a max margin problem?
  - How to solve the max margin problem?

Recall: Linear Discriminant Function

- In high-dimension,

$$g(\boldsymbol{x}) = \boldsymbol{w}^T \boldsymbol{x} + w_0.$$

  is a hyperplane.

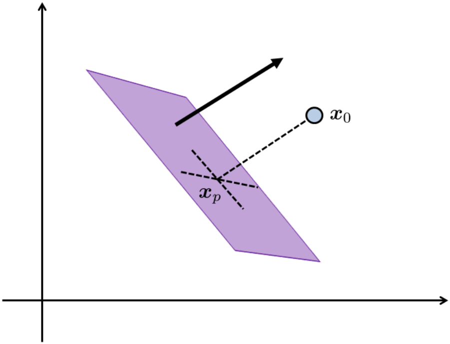


- **Separating Hyperplane**:

$$\begin{aligned}\mathcal{H} &= \{\boldsymbol{x} \mid g(\boldsymbol{x}) = 0\} \\ &= \{\boldsymbol{x} \mid \boldsymbol{w}^T \boldsymbol{x} + w_0 = 0\}\end{aligned}$$

- $\boldsymbol{x} \in \mathcal{H}$ means $\boldsymbol{x}$ is on the decision boundary.
- $\boldsymbol{w}/\|\boldsymbol{w}\|_2$ is the **normal vector** of $\mathcal{H}$.

Recall: Distance from $\boldsymbol{x}_0$ to $g(\boldsymbol{x}) = 0$



- Pick a point $\boldsymbol{x}_p$ on $\mathcal{H}$
- $\boldsymbol{x}_p$ is the closest point to $\boldsymbol{x}_0$
- $\boldsymbol{x}_0 - \boldsymbol{x}_p$ is the normal direction
- So, for some scalar $\eta > 0$,

$$\boldsymbol{x}_0 - \boldsymbol{x}_p = \eta \frac{\boldsymbol{w}}{\|\boldsymbol{w}\|_2}$$

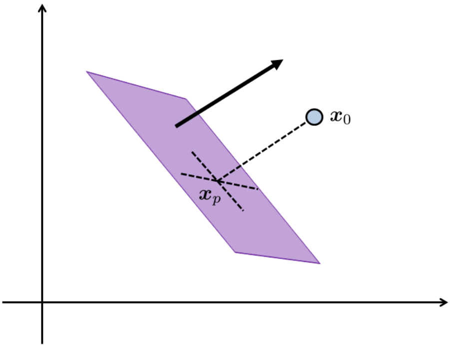
- $\boldsymbol{x}_p$ is on $\mathcal{H}$. So

$$g(\boldsymbol{x}_p) = \boldsymbol{w}^T \boldsymbol{x}_p + w_0 = 0$$

Therefore, we can show that

$$\begin{aligned} g(\boldsymbol{x}_0) &= \boldsymbol{w}^T \boldsymbol{x}_0 + w_0 \\ &= \boldsymbol{w}^T \left(\boldsymbol{x}_p + \eta \frac{\boldsymbol{w}}{\|\boldsymbol{w}\|_2} \right) + w_0 \\ &= g(\boldsymbol{x}_p) + \eta \|\boldsymbol{w}\|_2 = \eta \|\boldsymbol{w}\|_2. \end{aligned}$$

Recall: Distance from x_0 to $g(x) = 0$



- So distance is

$$\eta = \frac{g(\boldsymbol{x}_0)}{\|\boldsymbol{w}\|_2}$$

- The closest point $\boldsymbol{x}_p$ is

$$\begin{aligned} \boldsymbol{x}_p &= \boldsymbol{x}_0 - \eta \frac{\boldsymbol{w}}{\|\boldsymbol{w}\|_2} \\ &= \boldsymbol{x}_0 - \frac{g(\boldsymbol{x}_0)}{\|\boldsymbol{w}\|_2} \cdot \frac{\boldsymbol{w}}{\|\boldsymbol{w}\|_2}. \end{aligned}$$

Conclusion:

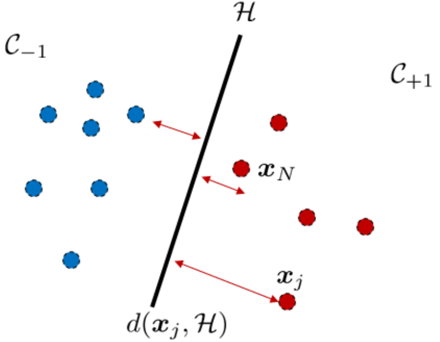
$$\boldsymbol{x}_p = \boldsymbol{x}_0 - \underbrace{\frac{g(\boldsymbol{x}_0)}{\|\boldsymbol{w}\|_2}}_{\text{distance}} \cdot \underbrace{\frac{\boldsymbol{w}}{\|\boldsymbol{w}\|_2}}_{\text{normal vector}}$$

Unsigned Distance

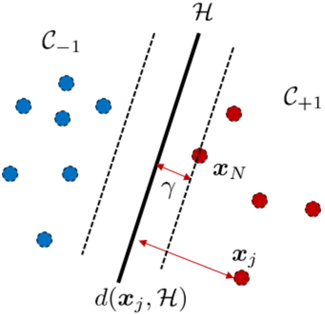
- We define the distance between a data point $\boldsymbol{x}_j$ and a separating hyperplane as

$$d(\boldsymbol{x}_j, \mathcal{H}) = \frac{|g(\boldsymbol{x}_j)|}{\|\boldsymbol{w}\|_2} = \frac{|\boldsymbol{w}^T\boldsymbol{x}_j + w_0|}{\|\boldsymbol{w}\|_2}.$$

- $d(\boldsymbol{x}_j, \mathcal{H})$ is called **unsigned** distance



Margin

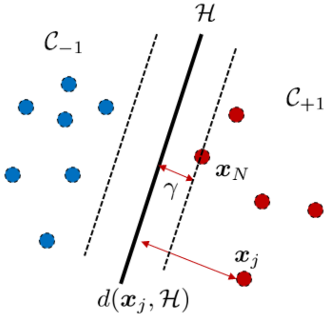


- Among all the unsigned distances, pick the smallest one.
- Margin: γ such that

$$\gamma = \min_{j=1,\dots,N} d(\boldsymbol{x}_j, \mathcal{H}).$$

- Without loss of generality, assume $\boldsymbol{x}_N$ is the closest point.

More about Margin



- Margin: γ such that

$$\gamma = \min_{j=1,\ldots,N} \; d(\boldsymbol{x}_j, \mathcal{H}).$$

- γ depends on $(\boldsymbol{w}, w_0)$.
- γ always exist because training set is finite.
- $\gamma \geq 0$, and is zero when $\boldsymbol{x}_N$ is on the boundary.

Signed Distance

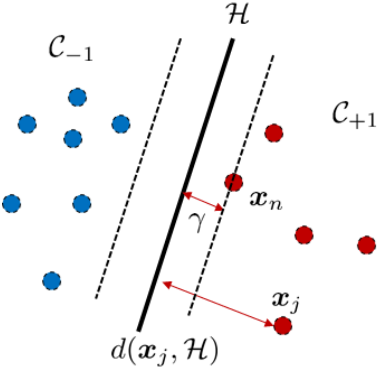
- $d(\boldsymbol{x}_j, \mathcal{H})$ is unsigned
- So γ does not tell whether a point $\boldsymbol{x}_j$ is correctly classified or not
- Assume that the labels are defined as $y_j \in \{-1, +1\}$
- Then define a **signed distance**

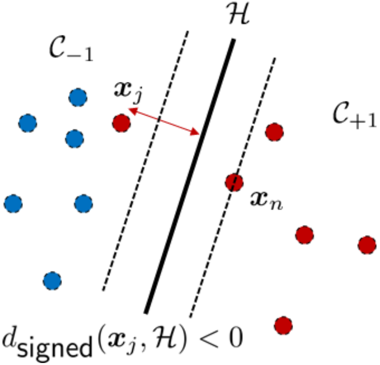
$$
\begin{aligned}
d_{\text{signed}}(\boldsymbol{x}_j, \mathcal{H}) &= y_j \left(\frac{\boldsymbol{w}^T \boldsymbol{x}_j + w_0}{\|\boldsymbol{w}\|_2} \right) \\
&= \begin{cases} \geq 0, & \text{correctly classify } \boldsymbol{x}_j \\ < 0, & \text{incorrectly classify } \boldsymbol{x}_j. \end{cases}
\end{aligned}
$$

- Recall perceptron loss:

$$
\mathcal{L}(\boldsymbol{x}_j) = \max\left\{ -y_j(\boldsymbol{w}^T \boldsymbol{x}_j + w_0), 0 \right\}
$$

Unsigned VS Signed Distance





- Unsigned distance: Just the distance
- Signed distance: Distance plus whether on the correct side

Max-Margin Objective

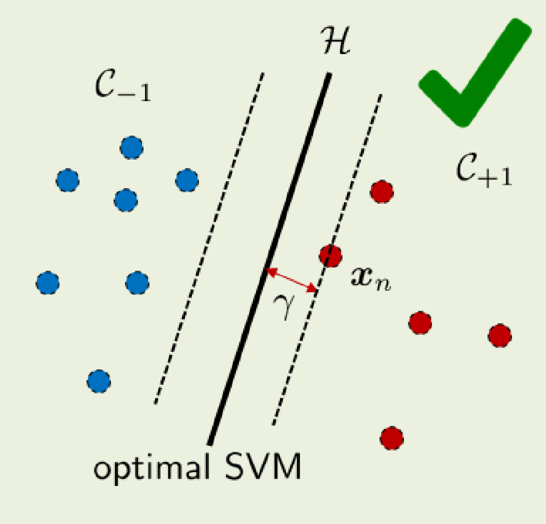
- **Assumptions**: Linearly separable.
- This means

$$y_j \left(\frac{\boldsymbol{w}^T \boldsymbol{x}_j + w_0}{\|\boldsymbol{w}\|_2} \right) \geq \gamma, \quad j = 1, \ldots, N.$$

- All training samples are correctly classified.
- All training samples are at lest γ from the boundary.
- So the max-margin classifier is

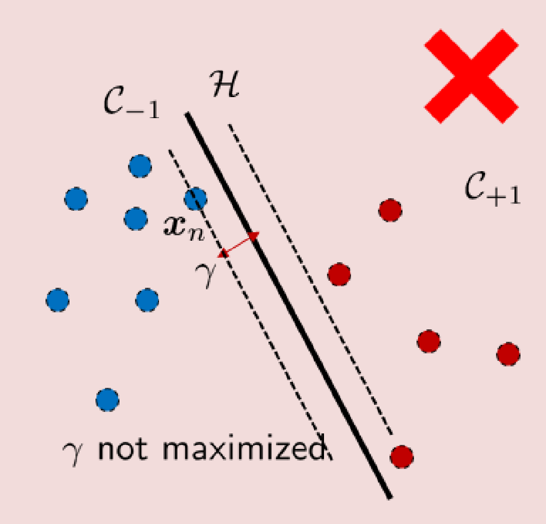
$$\begin{aligned} \underset{\boldsymbol{w}, w_0}{\text{maximize}} \quad & \gamma \\ \text{subject to} \quad & y_j \left(\frac{\boldsymbol{w}^T \boldsymbol{x}_j + w_0}{\|\boldsymbol{w}\|_2} \right) \geq \gamma, \quad j = 1, \ldots, N. \end{aligned}$$

Good or Bad?



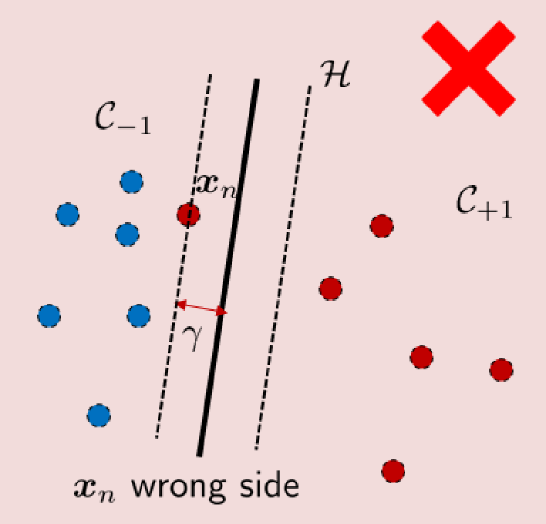
$\mathcal{H}$
$\mathcal{C}_{-1}$
$\mathcal{C}_{+1}$
$\boldsymbol{x}_n$
γ
optimal SVM

Good or Bad?



$\mathcal{H}$
$\mathcal{C}_{-1}$
$\mathcal{C}_{+1}$
$\boldsymbol{x}_n$
γ
γ not maximized

Good or Bad?



$\mathcal{H}$
$\mathcal{C}_{-1}$
$\boldsymbol{x}_n$
$\mathcal{C}_{+1}$
γ
$\boldsymbol{x}_n$ wrong side

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

$$\underset{\boldsymbol{w}, w_0}{\text{maximize}} \;\; \gamma$$
$$\text{subject to} \;\; y_j \left(\frac{\boldsymbol{w}^T \boldsymbol{x}_j + w_0}{\|\boldsymbol{w}\|_2} \right) \geq \gamma, \quad j = 1, \ldots, N.$$

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

Trick 1: Scaling

- The optimization is

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

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

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

- Then, we can show that

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

Trick 1: Scaling

- So we can turn this optimization

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

- into this optimization

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

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

Eliminate $\widetilde{\gamma}$

- How about we turn this optimization

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

- into this optimization?

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

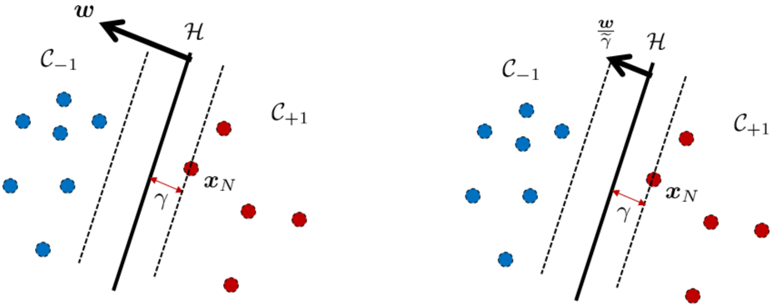
- You can refine the variables $\boldsymbol{w} \leftarrow \frac{\boldsymbol{w}}{\widetilde{\gamma}}$ and $w_0 \leftarrow \frac{w_0}{\widetilde{\gamma}}$

Eliminate $\tilde{\gamma}$

- This gives you

$$\underset{\boldsymbol{w}, w_0}{\text{maximize}} \quad \frac{1}{\|\boldsymbol{w}\|_2}$$
$$\text{subject to} \quad y_j(\boldsymbol{w}^T \boldsymbol{x}_j + w_0) \geq 1, \quad j = 1, \ldots, N.$$

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



Trick 2: Max to Min

- You want to solve

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

- How about

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

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

Hand Crafted Example

- You have four data points

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

- Labels are

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

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

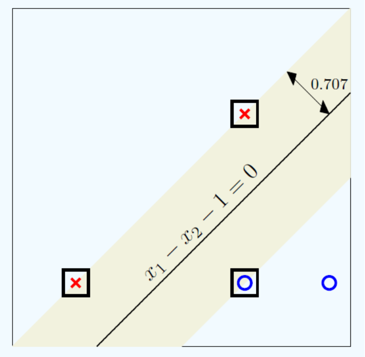
$$\begin{aligned} -w_0 &\geq 1 \quad (i) \\ -(2w_1 + 2w_2 + w_0) &\geq 1 \quad (ii) \\ 2w_1 + w_0 &\geq 1 \quad (iii) \\ 3w_1 + w_0 &\geq 1 \quad (iv) \end{aligned}$$

- 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 $\frac{1}{2}\|\boldsymbol{w}\|^2$.
- Can show that
$$\frac{1}{2}\|\boldsymbol{w}\|^2 = \frac{1}{2}(w_1^2 + w_2^2) \geq 1.$$
- 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:

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

- Margin:

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

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

Writing Your SVM

- The problem is

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

- This is a quadratic programming problem:

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

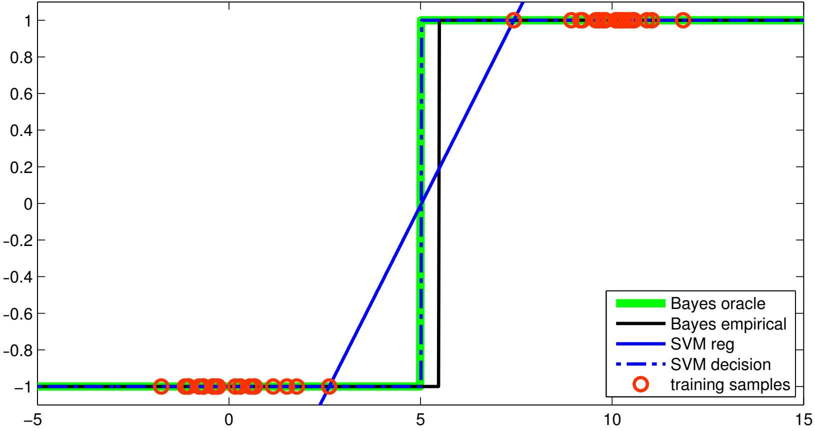
- Solution is $\boldsymbol{u}^* = \text{QP}(\boldsymbol{Q}, \boldsymbol{p}, \boldsymbol{A}, \boldsymbol{c})$. Solvable using any QP solver.
- To us: This is convex objective with convex constraint.
- Use CVX! Below is a 1D example.

Writing Your SVM

```
mu0    = 0;     mu1 = 10;      sigma0 = 1;     sigma1 = 1;
N0     = 20;     N1 = 20;           N = N0+N1;

x0 = random('normal',mu0,sigma0,N0,1);
x1 = random('normal',mu1,sigma1,N1,1);

y0 = -ones(N0,1);    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.



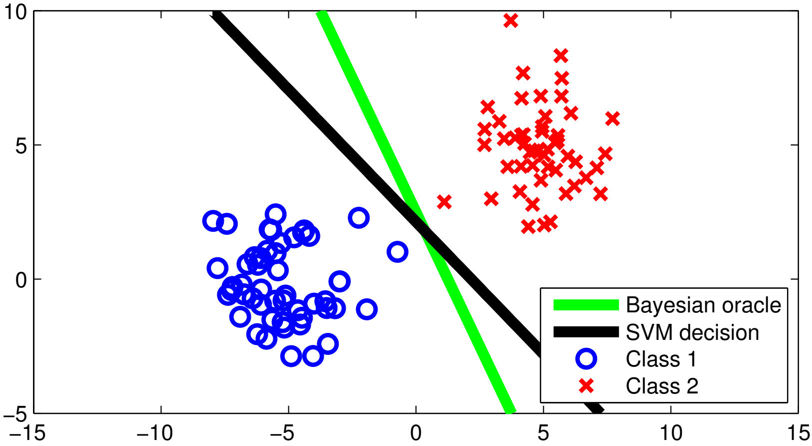
- Classifier is always defined by the support vectors!

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

- $\boldsymbol{\mu}_1 = [-5, 0]^T$, $\boldsymbol{\mu}_2 = [5, 5]^T$. $\boldsymbol{\Sigma} = 1.5^2 \boldsymbol{I}$.
- 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 $(\boldsymbol{w}, w_0)$?
- $\boldsymbol{w}^T\boldsymbol{x} + w_0 = 0$ implies $w_1x_1 + w_2x_2 + w_0 = 0$.
- 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/