# 18-661 Introduction to Machine Learning

Support Vector Machines (SVM) – I

Spring 2024

# Announcements and Reminders

- Homework 2 due today!
- Homework 3 is posted and due Feb 23.

# This Week: Support Vector Machines (SVM)

Today:

- (Linear) Support Vector Machines
- Max-margin and hinge loss formulations

Next Class:

- Duality
- Kernel Machines and the “Kernel Trick”

# Outline

1. Why SVM?
2. Max-Margin Formulation
3. Hinge Loss Formulation
4. Summary

# Why SVM?

# Why Do We Need SVM?

Naïve Bayes (circa 1750):

- $\Pr(Y|x_1 \ldots x_n) = \Pr(Y) \prod_{i=1}^{n} \Pr(x_i|Y)$

# Why Do We Need SVM?

Naïve Bayes (circa 1750):

- $\Pr(Y|x_1 \ldots x_n) = \Pr(Y) \prod_{i=1}^{n} \Pr(x_i|Y)$

Logistic Regression (circa 1950):

- $\arg\min_{\mathbf{w}} - \sum_i \left(y_i \log \sigma(\mathbf{w}^T \mathbf{x}_n) + (1 - y_n \log(1 - \sigma(\mathbf{w}^T \mathbf{x}))\right)$

# Why Do We Need SVM?

Naïve Bayes (circa 1750):

- $\Pr(Y|x_1 \ldots x_n) = \Pr(Y) \prod_{i=1}^{n} \Pr(x_i|Y)$

Logistic Regression (circa 1950):

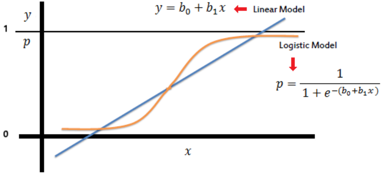
- $\arg\min_{\mathbf{w}} - \sum_i \left( y_i \log \sigma(\mathbf{w}^T \mathbf{x}_n) + (1 - y_n \log(1 - \sigma(\mathbf{w}^T \mathbf{x}))\right)$

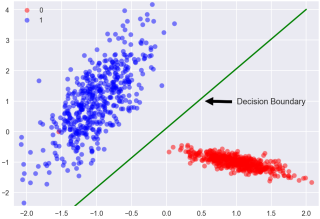
**The problem** (circa 1990):

- Logistic regression and Naïve Bayes train over the whole dataset.
- These can require a lot of memory in high-dimensional settings.
- Neither can be easily generalized to nonlinear settings.

Can we do better?

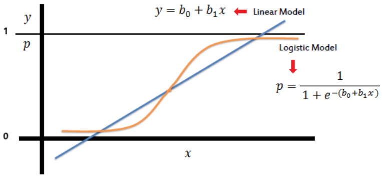
# Binary Logistic Regression

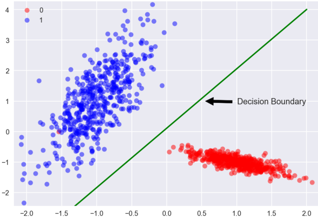



- We only need to know if $p(\boldsymbol{x}) > 0.5$ or $< 0.5$.
- We don't (always) need to know how far $\boldsymbol{x}$ is from this boundary.

# Binary Logistic Regression




- We only need to know if $p(\boldsymbol{x}) > 0.5$ or $< 0.5$.
- We don't (always) need to know how far $\boldsymbol{x}$ is from this boundary.

**How can we use this insight to improve the classification algorithm?**

- What if we just looked at the boundary?
- Maybe then we could ignore some of the samples?

# SVM: Support Vector Machines

Work done by researchers at AT&T Bell Labs in the 1990s.

- Becomes the classifier of choice for many applications until tree-based methods started gaining popularity in the early 2000s
- Still extremely popular today

# SVM: Support Vector Machines

Work done by researchers at AT&T Bell Labs in the 1990s.

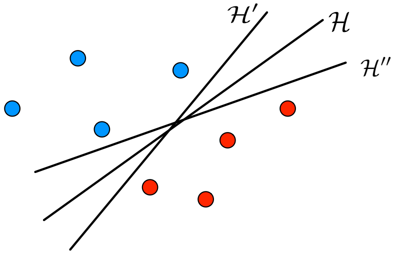
- Becomes the classifier of choice for many applications until tree-based methods started gaining popularity in the early 2000s
- Still extremely popular today

We will see that SVM:

- Is less sensitive to outliers.
- Maximizes distance of training data from the boundary.
- Only requires a subset of the training points.
- Generalizes well to many nonlinear models.
- Scales better with high-dimensional data.

# Max-Margin Formulation

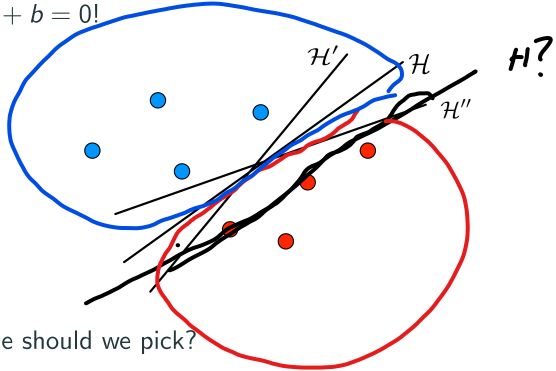
# Binary Classification: Finding a Linear Decision Boundary




- Input features $\boldsymbol{x}$.
- Decision boundary is a hyperplane $\mathcal{H} : \boldsymbol{w}^\top \boldsymbol{x} + b = 0$.

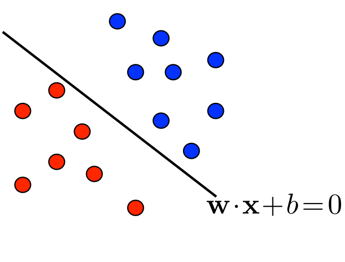
# Intuition: Where to Put the Decision Boundary?

- Consider a *separable* training dataset (e.g., with two features)
- There are an infinite number of decision boundaries $\mathcal{H} : \boldsymbol{w}^\top \boldsymbol{x} + b = 0$!

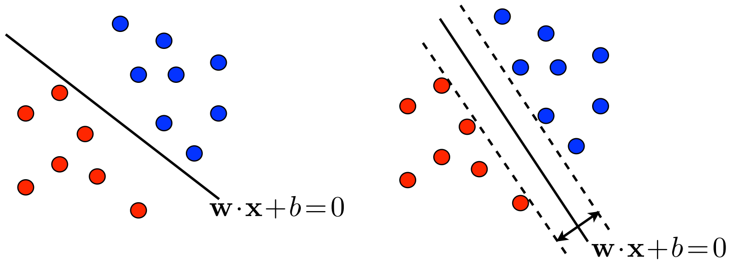



- Which one should we pick?

# Intuition: Where to Put the Decision Boundary?



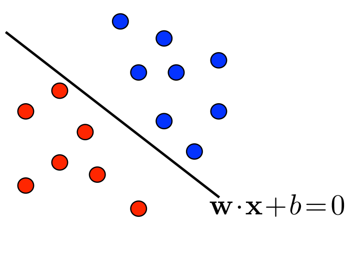
# Intuition: Where to Put the Decision Boundary?




Find a decision boundary in the '*middle*' of the two classes that:

- Perfectly classifies the training data
- Is as far away from every training point as possible

# Intuition: Where to Put the Decision Boundary?

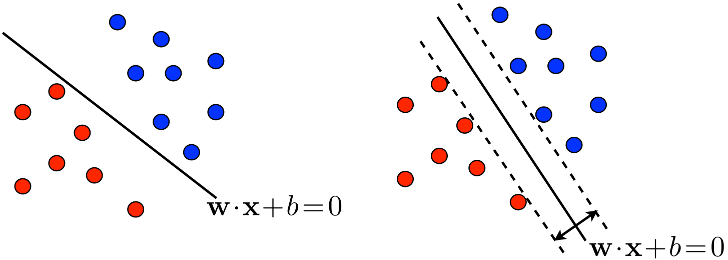



Find a decision boundary in the '*middle*' of the two classes that:

- Perfectly classifies the training data
- Is as far away from every training point as possible

Let us apply this intuition to build a classifier that maximizes the margin between training points and the decision boundary.

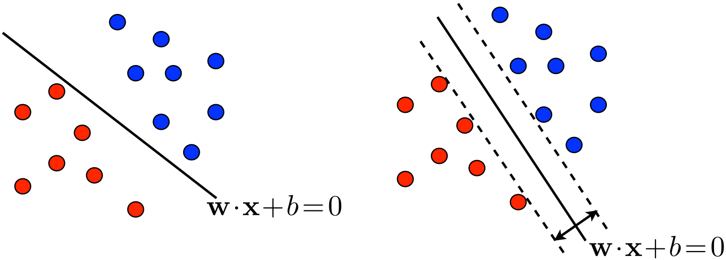
What is a hyperplane?




- General equation is $\boldsymbol{w}^\top \boldsymbol{x} + b = 0$
- Divides the space in half, i.e., $\boldsymbol{w}^\top \boldsymbol{x} + b > 0$ and $\boldsymbol{w}^\top \boldsymbol{x} + b < 0$

# First, Some Vector Geometry

## What is a hyperplane?




- General equation is $\boldsymbol{w}^\top \boldsymbol{x} + b = 0$
- Divides the space in half, i.e., $\boldsymbol{w}^\top \boldsymbol{x} + b > 0$ and $\boldsymbol{w}^\top \boldsymbol{x} + b < 0$
- A hyperplane is a line in 2D and a plane in 3D
- $\boldsymbol{w} \in \mathbb{R}^d$ is a non-zero normal vector

# Vector Norms and Inner Products

Given two vectors $\mathbf{w}$ and $\mathbf{x}$, what is their inner product?

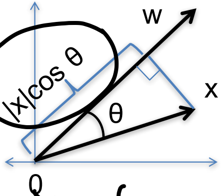
- Inner Product $\mathbf{w}^\top \mathbf{x} = w_1 x_1 + w_2 x_2 + \cdots + w_d x_d$

# Vector Norms and Inner Products

Given two vectors **w** and **x**, what is their inner product?

- Inner Product $\mathbf{w}^\top \mathbf{x} = w_1 x_1 + w_2 x_2 + \cdots + w_d x_d$

if $\|w\|_2 = 1$

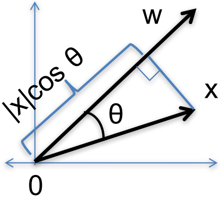



Inner Product $\mathbf{w}^\top \mathbf{x}$ is also equal to $\|\boldsymbol{w}\| \|\boldsymbol{x}\| \cos\theta$.

# Vector Norms and Inner Products

Given two vectors **w** and **x**, what is their inner product?

- Inner Product $\mathbf{w}^\top \mathbf{x} = w_1 x_1 + w_2 x_2 + \cdots + w_d x_d$



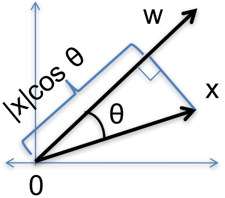

Inner Product $\mathbf{w}^\top \mathbf{x}$ is also equal to $\|\boldsymbol{w}\| \, \|\boldsymbol{x}\| \cos \theta$.

- If $\mathbf{w} = \mathbf{x}$?

# Vector Norms and Inner Products

Given two vectors $\mathbf{w}$ and $\mathbf{x}$, what is their inner product?

- Inner Product $\mathbf{w}^\top \mathbf{x} = w_1 x_1 + w_2 x_2 + \cdots + w_d x_d$



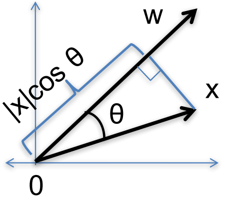

Inner Product $\mathbf{w}^\top \mathbf{x}$ is also equal to $\|\boldsymbol{w}\| \, \|\boldsymbol{x}\| \cos\theta$.

- If $\mathbf{w} = \mathbf{x}$? $\theta = 0$, so $\mathbf{w}^\top \mathbf{w} = \|\boldsymbol{w}\|^2$.
- If $\mathbf{w} \perp \mathbf{x}$?

# Vector Norms and Inner Products

Given two vectors **w** and **x**, what is their inner product?

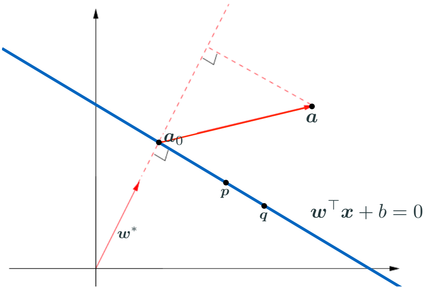
- Inner Product $\mathbf{w}^\top \mathbf{x} = w_1 x_1 + w_2 x_2 + \cdots + w_d x_d$




Inner Product $\mathbf{w}^\top \mathbf{x}$ is also equal to $\|\boldsymbol{w}\| \, \|\boldsymbol{x}\| \cos\theta$.

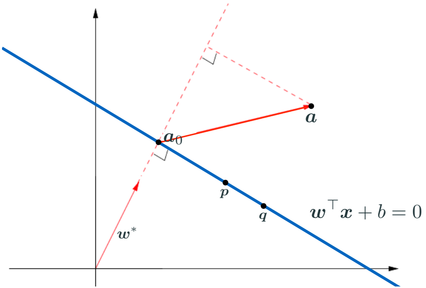
- If $\mathbf{w} = \mathbf{x}$? $\theta = 0$, so $\mathbf{w}^\top \mathbf{w} = \|\boldsymbol{w}\|^2$.
- If $\mathbf{w} \perp \mathbf{x}$? $\theta = \pi/2$, so $\mathbf{w}^T \mathbf{x} = 0$.

# Normal Vector of a Hyperplane



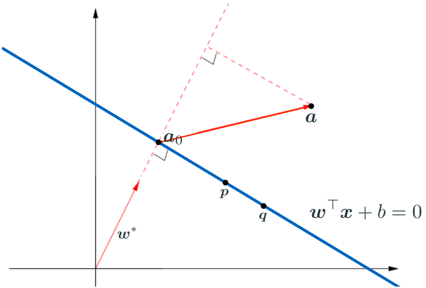

**What is the meaning of w in the hyperplane $\mathbf{w}^T\mathbf{x} + b = 0$?**

# Normal Vector of a Hyperplane




**Vector $\boldsymbol{w}$ is normal to the hyperplane. Why?**

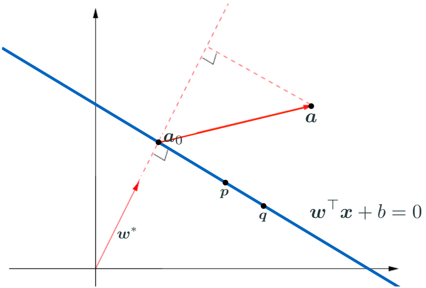
# Normal Vector of a Hyperplane




**Vector $\boldsymbol{w}$ is normal to the hyperplane. Why?**

- If $\boldsymbol{p}$ and $\boldsymbol{q}$ are both on the line, then $\boldsymbol{w}^\top \boldsymbol{p} + b = \boldsymbol{w}^\top \boldsymbol{q} + b = 0$.

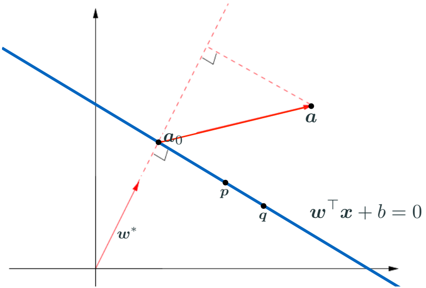
# Normal Vector of a Hyperplane




**Vector $\boldsymbol{w}$ is normal to the hyperplane. Why?**

- If $\boldsymbol{p}$ and $\boldsymbol{q}$ are both on the line, then $\boldsymbol{w}^\top \boldsymbol{p} + b = \boldsymbol{w}^\top \boldsymbol{q} + b = 0$.
- Then $\boldsymbol{w}^\top(\boldsymbol{p} - \boldsymbol{q}) = \boldsymbol{w}^\top \boldsymbol{p} - \boldsymbol{w}^\top \boldsymbol{q} = -b - (-b) = 0$

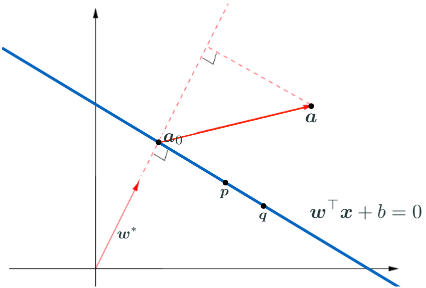
# Normal Vector of a Hyperplane




**Vector $\boldsymbol{w}$ is normal to the hyperplane. Why?**

- If $\boldsymbol{p}$ and $\boldsymbol{q}$ are both on the line, then $\boldsymbol{w}^\top \boldsymbol{p} + b = \boldsymbol{w}^\top \boldsymbol{q} + b = 0$.
- Then $\boldsymbol{w}^\top(\boldsymbol{p} - \boldsymbol{q}) = \boldsymbol{w}^\top \boldsymbol{p} - \boldsymbol{w}^\top \boldsymbol{q} = -b - (-b) = 0$
- $\boldsymbol{p} - \boldsymbol{q}$ is an arbitrary vector parallel to the line, thus $\boldsymbol{w}$ is orthogonal

# Normal Vector of a Hyperplane

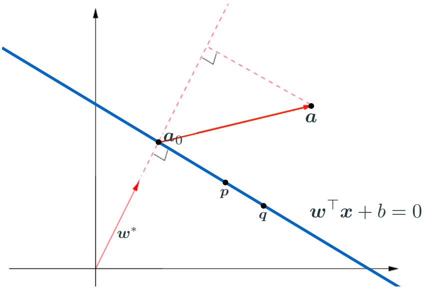



**Vector $\boldsymbol{w}$ is normal to the hyperplane. Why?**

- If $\boldsymbol{p}$ and $\boldsymbol{q}$ are both on the line, then $\boldsymbol{w}^\top \boldsymbol{p} + b = \boldsymbol{w}^\top \boldsymbol{q} + b = 0$.
- Then $\boldsymbol{w}^\top(\boldsymbol{p} - \boldsymbol{q}) = \boldsymbol{w}^\top \boldsymbol{p} - \boldsymbol{w}^\top \boldsymbol{q} = -b - (-b) = 0$
- $\boldsymbol{p} - \boldsymbol{q}$ is an arbitrary vector parallel to the line, thus $\boldsymbol{w}$ is orthogonal

Let $\boldsymbol{w}^* = \frac{\boldsymbol{w}}{\|\boldsymbol{w}\|_2}$ be the unit normal vector in the direction $\mathbf{w}$.

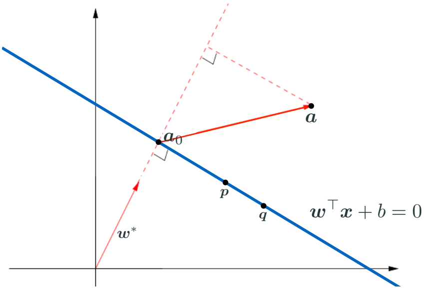
# Distance from a Hyperplane




**How to find the distance from $\boldsymbol{a}$ to the hyperplane?**

- We want to find distance between $\boldsymbol{a}$ and line in the direction of $\boldsymbol{w}^*$.

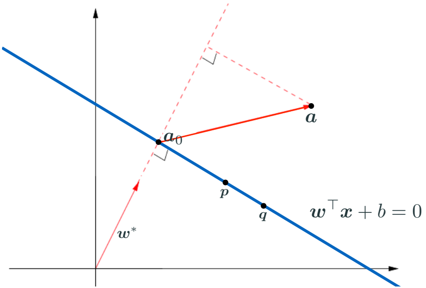
# Distance from a Hyperplane




**How to find the distance from $\boldsymbol{a}$ to the hyperplane?**

- We want to find distance between $\boldsymbol{a}$ and line in the direction of $\boldsymbol{w}^*$.
- If we define point $\boldsymbol{a}_0$ on the line, then this distance corresponds to length of $\boldsymbol{a} - \boldsymbol{a}_0$ in direction of $\boldsymbol{w}^*$, which equals ${\boldsymbol{w}^*}^\top(\boldsymbol{a} - \boldsymbol{a}_0)$.

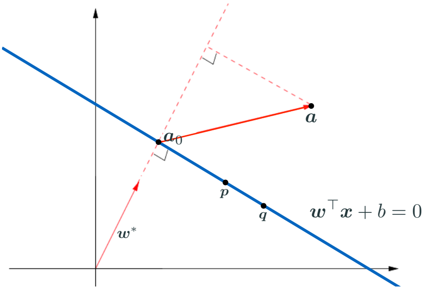
# Distance from a Hyperplane




**How to find the distance from $\boldsymbol{a}$ to the hyperplane?**

- We want to find distance between $\boldsymbol{a}$ and line in the direction of $\boldsymbol{w}^*$.
- If we define point $\boldsymbol{a}_0$ on the line, then this distance corresponds to length of $\boldsymbol{a} - \boldsymbol{a}_0$ in direction of $\boldsymbol{w}^*$, which equals $\boldsymbol{w}^{*\top}(\boldsymbol{a} - \boldsymbol{a}_0)$.
- We know $\boldsymbol{w}^\top \boldsymbol{a}_0 = -b$ since $\boldsymbol{w}^\top \boldsymbol{a}_0 + b = 0$.

# Distance from a Hyperplane




**How to find the distance from $\boldsymbol{a}$ to the hyperplane?**

- We want to find distance between $\boldsymbol{a}$ and line in the direction of $\boldsymbol{w}^*$.
- If we define point $\boldsymbol{a}_0$ on the line, then this distance corresponds to length of $\boldsymbol{a} - \boldsymbol{a}_0$ in direction of $\boldsymbol{w}^*$, which equals $\boldsymbol{w}^{*\top}(\boldsymbol{a} - \boldsymbol{a}_0)$.
- We know $\boldsymbol{w}^\top \boldsymbol{a}_0 = -b$ since $\boldsymbol{w}^\top \boldsymbol{a}_0 + b = 0$.
- Then the distance equals $\frac{1}{\|\boldsymbol{w}\|_2}(\boldsymbol{w}^\top \boldsymbol{a} + b)$ .

# Distance from a Point to Decision Boundary

The *unsigned* distance from a point $\boldsymbol{x}$ to the decision boundary (hyperplane) $\mathcal{H}$ is

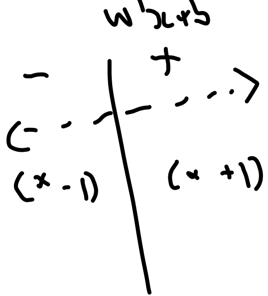
$$d_{\mathcal{H}}(\boldsymbol{x}) = \frac{|\boldsymbol{w}^\top \boldsymbol{x} + b|}{\|\boldsymbol{w}\|_2}$$

# Distance from a Point to Decision Boundary

The *unsigned* distance from a point $\boldsymbol{x}$ to the decision boundary (hyperplane) $\mathcal{H}$ is

$$d_{\mathcal{H}}(\boldsymbol{x}) = \frac{|\boldsymbol{w}^\top \boldsymbol{x} + b|}{\|\boldsymbol{w}\|_2}$$

How to remove the absolute value $|\cdot|$?



# Distance from a Point to Decision Boundary

The *unsigned* distance from a point $\boldsymbol{x}$ to the decision boundary (hyperplane) $\mathcal{H}$ is

$$d_{\mathcal{H}}(\boldsymbol{x}) = \frac{|\boldsymbol{w}^\top \boldsymbol{x} + b|}{\|\boldsymbol{w}\|_2}$$

How to remove the absolute value $|\cdot|$?

Notation changes from Logistic Regression: Use $y = +1$ to represent positive label and $y = -1$ for negative label.

# Distance from a Point to Decision Boundary

The *unsigned* distance from a point $\boldsymbol{x}$ to the decision boundary (hyperplane) $\mathcal{H}$ is

$$d_{\mathcal{H}}(\boldsymbol{x}) = \frac{|\boldsymbol{w}^\top \boldsymbol{x} + b|}{\|\boldsymbol{w}\|_2}$$

How to remove the absolute value $|\cdot|$?

Notation changes from Logistic Regression: Use $y = +1$ to represent positive label and $y = -1$ for negative label.

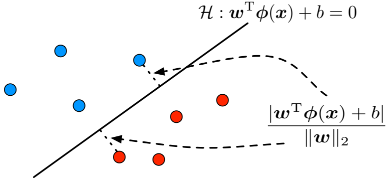
Then, exploiting the fact that the decision boundary classifies every point in the training dataset correctly, we have $(\boldsymbol{w}^\top \boldsymbol{x} + b)$ and $\boldsymbol{x}$'s label $y$ must have the same sign. So we get

$$d_{\mathcal{H}}(\boldsymbol{x}) = \frac{y[\boldsymbol{w}^\top \boldsymbol{x} + b]}{\|\boldsymbol{w}\|_2}$$

# Defining the Margin

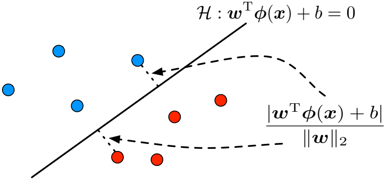
**Margin**: Smallest distance between the hyperplane and all training points

$$\text{MARGIN}(\boldsymbol{w}, b) = \min_n \frac{y_n[\boldsymbol{w}^\top \boldsymbol{x}_n + b]}{\|\boldsymbol{w}\|_2}$$




How can we use this to find the SVM solution?

# Optimizing the Margin

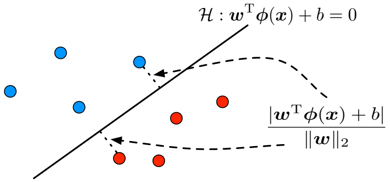



**How should we pick $(\boldsymbol{w}, b)$ based on its margin?**

We want a decision boundary that is as far away from all training points as possible, so we to *maximize* the margin!

$$\max_{\boldsymbol{w},b}\left(\min_{n}\frac{y_n[\boldsymbol{w}^\top\boldsymbol{x}_n + b]}{\|\boldsymbol{w}\|_2}\right) = \max_{\boldsymbol{w},b}\left(\frac{1}{\|\boldsymbol{w}\|_2}\min_{n} y_n[\boldsymbol{w}^\top\boldsymbol{x}_n + b]\right)$$

# Optimizing the Margin




**How should we pick $(\boldsymbol{w}, b)$ based on its margin?**

We want a decision boundary that is as far away from all training points as possible, so we to *maximize* the margin!

$$\max_{\boldsymbol{w},b}\left(\min_n \frac{y_n[\boldsymbol{w}^\top \boldsymbol{x}_n + b]}{\|\boldsymbol{w}\|_2}\right) = \max_{\boldsymbol{w},b}\left(\frac{1}{\|\boldsymbol{w}\|_2}\min_n y_n[\boldsymbol{w}^\top \boldsymbol{x}_n + b]\right)$$

Only involves points near the boundary (more on this later).

# Scale of $w$

**Margin**: Smallest distance between the hyperplane and all training points

$$\textsc{margin}(\boldsymbol{w}, b) = \min_n \frac{y_n[\boldsymbol{w}^\top \boldsymbol{x}_n + b]}{\|\boldsymbol{w}\|_2}$$

**Consider three hyperplanes**

$$(\boldsymbol{w}, b) \quad (2\boldsymbol{w}, 2b) \quad (.5\boldsymbol{w}, .5b)$$

**Which one has the largest margin?**

# Scale of $w$

**Margin**: Smallest distance between the hyperplane and all training points

$$\textsc{margin}(\boldsymbol{w}, b) = \min_n \frac{y_n[\boldsymbol{w}^\top \boldsymbol{x}_n + b]}{\|\boldsymbol{w}\|_2}$$

**Consider three hyperplanes**

$$(\boldsymbol{w}, b) \quad (2\boldsymbol{w}, 2b) \quad (.5\boldsymbol{w}, .5b)$$

**Which one has the largest margin?**

- The $\textsc{margin}$ doesn't change if we scale $(\boldsymbol{w}, b)$ by a constant $c$
- $\boldsymbol{w}^\top \boldsymbol{x} + b = 0$ and $(c\boldsymbol{w})^\top \boldsymbol{x} + (cb) = 0$: same decision boundary!

# Scale of $w$

**Margin**: Smallest distance between the hyperplane and all training points

$$\textsc{margin}(\boldsymbol{w}, b) = \min_n \frac{y_n[\boldsymbol{w}^\top \boldsymbol{x}_n + b]}{\|\boldsymbol{w}\|_2}$$

**Consider three hyperplanes**

$$(\boldsymbol{w}, b) \quad (2\boldsymbol{w}, 2b) \quad (.5\boldsymbol{w}, .5b)$$

**Which one has the largest margin?**

- The $\textsc{margin}$ doesn't change if we scale $(\boldsymbol{w}, b)$ by a constant $c$
- $\boldsymbol{w}^\top \boldsymbol{x} + b = 0$ and $(c\boldsymbol{w})^\top \boldsymbol{x} + (cb) = 0$: same decision boundary!
- Can we further constrain the problem so as to get a unique solution $(\boldsymbol{w}, b)$?

# Rescaled Margin

We can further constrain the problem by scaling $(\boldsymbol{w}, b)$ such that

$$\min_n y_n[\boldsymbol{w}^\top \boldsymbol{x}_n + b] = 1.$$

Note that there always exists a scaling for which this is true.

# Rescaled Margin

We can further constrain the problem by scaling $(\boldsymbol{w}, b)$ such that

$$\min_n y_n[\boldsymbol{w}^\top \boldsymbol{x}_n + b] = 1.$$

Note that there always exists a scaling for which this is true. We've fixed the numerator in the $\textsc{margin}(\boldsymbol{w}, b)$ equation, and we have:

$$\textsc{margin}(\boldsymbol{w}, b) = \frac{\min_n y_n[\boldsymbol{w}^\top \boldsymbol{x}_n + b]}{\|\boldsymbol{w}\|_2} = \frac{1}{\|\boldsymbol{w}\|_2}$$

## Rescaled Margin

We can further constrain the problem by scaling $(\boldsymbol{w}, b)$ such that

$$\min_n y_n[\boldsymbol{w}^\top \boldsymbol{x}_n + b] = 1.$$

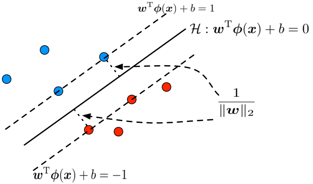
Note that there always exists a scaling for which this is true. We've fixed the numerator in the $\textsc{margin}(\boldsymbol{w}, b)$ equation, and we have:

$$\textsc{margin}(\boldsymbol{w}, b) = \frac{\min_n y_n[\boldsymbol{w}^\top \boldsymbol{x}_n + b]}{\|\boldsymbol{w}\|_2} = \frac{1}{\|\boldsymbol{w}\|_2}$$

Hence the points closest to the decision boundary are at distance $\frac{1}{\|\boldsymbol{w}\|_2}$.

$w^T x + b = 0$

$(wc)^T x + bc = 0$



# SVM: Max-margin Formulation for Separable Data

We thus want to solve:

$$\max_{\boldsymbol{w},b} \underbrace{\frac{1}{\|\boldsymbol{w}\|_2}}_{\text{margin}} \quad \text{such that} \quad \underbrace{\min_n y_n[\boldsymbol{w}^\top \boldsymbol{x}_n + b] = 1}_{\text{scaling of } \boldsymbol{w},\, b}$$

If $\min_n \beta_n = 1 \iff \forall n \ \beta_n \geq 1$

which is equivalent to

$$\max_{\boldsymbol{w},b} \frac{1}{\|\boldsymbol{w}\|_2} \quad \text{such that} \quad y_n[\boldsymbol{w}^\top \boldsymbol{x}_n + b] \geq 1, \quad \forall \quad n$$

# SVM: Max-margin Formulation for Separable Data

We thus want to solve:

$$\max_{\boldsymbol{w},b} \underbrace{\frac{1}{\|\boldsymbol{w}\|_2}}_{\text{margin}} \quad \text{such that} \quad \underbrace{\min_n y_n[\boldsymbol{w}^\top \boldsymbol{x}_n + b] = 1}_{\text{scaling of } \boldsymbol{w},\, b}$$

which is equivalent to

$$\max_{\boldsymbol{w},b} \frac{1}{\|\boldsymbol{w}\|_2} \quad \text{such that} \quad y_n[\boldsymbol{w}^\top \boldsymbol{x}_n + b] \geq 1, \quad \forall \quad n$$

This is further equivalent to

$$\begin{aligned} \min_{\boldsymbol{w},b} \quad & \frac{1}{2}\|\boldsymbol{w}\|_2^2 \\ \text{s.t.} \quad & y_n[\boldsymbol{w}^\top \boldsymbol{x}_n + b] \geq 1, \quad \forall \quad n \end{aligned}$$

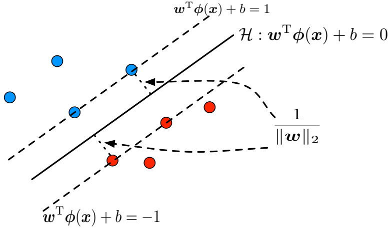
Given our geometric intuition, SVM is called a **max margin** (or large margin) classifier. The constraints are called **large margin constraints**.

# Support Vectors: A First Look

## SVM formulation for separable data

$$\min_{\boldsymbol{w}, b} \quad \frac{1}{2}\|\boldsymbol{w}\|_2^2$$

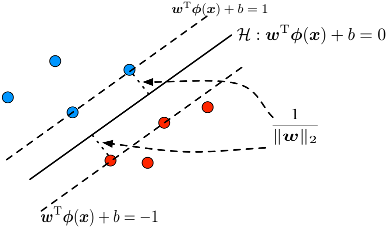
$$\text{s.t.} \quad y_n[\boldsymbol{w}^\top \boldsymbol{x}_n + b] \geq 1, \quad \forall \quad n$$



# Support Vectors: A First Look

**SVM formulation for separable data**

$$\min_{\boldsymbol{w},b} \quad \frac{1}{2}\|\boldsymbol{w}\|_2^2$$
$$\text{s.t.} \quad y_n[\boldsymbol{w}^\top \boldsymbol{x}_n + b] \geq 1 \quad \forall \quad n$$




Two types of training data, based on the situations of the constraint:

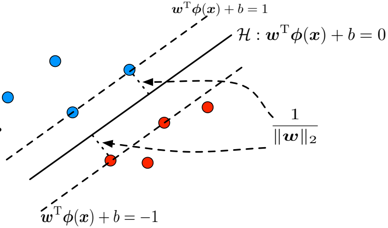
- "=": $y_n[\boldsymbol{w}^\top \boldsymbol{x}_n + b] = 1$. These training data points are called "support vectors", which have the minimum distance ($\frac{1}{\|\boldsymbol{w}\|}$) to the boundary.

### SVM formulation for separable data

$$\min_{\boldsymbol{w},b} \quad \frac{1}{2}\|\boldsymbol{w}\|_2^2 \quad \longleftrightarrow \quad \text{max margin}$$

$$\text{s.t.} \quad y_n[\boldsymbol{w}^\top \boldsymbol{x}_n + b] \geq 1, \quad \forall \quad n$$

Constraints




Two types of training data, based on the situations of the constraint:

- "$=$": $y_n[\boldsymbol{w}^\top \boldsymbol{x}_n + b] = 1$. These training data points are called "support vectors", which have the minimum distance ($\frac{1}{\|\boldsymbol{w}\|}$) to the boundary.
- "$>$": $y_n[\boldsymbol{w}^\top \boldsymbol{x}_n + b] > 1$. Distance to the boundary is larger than the minimum. Removing these data points does not affect the optimal solution (more on this next lecture).

# SVM for Non-separable Data

**SVM formulation for separable data**

$$\min_{\boldsymbol{w},b} \quad \frac{1}{2}\|\boldsymbol{w}\|_2^2$$
$$\text{s.t.} \quad y_n[\boldsymbol{w}^\top \boldsymbol{x}_n + b] \geq 1, \quad \forall \quad n$$

**Non-separable setting**
In practice our training data may not be separable. What issues arise with the optimization problem above when data is not separable?

# SVM for Non-separable Data

**SVM formulation for separable data**

$$\min_{\boldsymbol{w},b} \quad \frac{1}{2}\|\boldsymbol{w}\|_2^2$$
$$\text{s.t.} \quad y_n[\boldsymbol{w}^\top \boldsymbol{x}_n + b] \geq 1, \quad \forall \quad n$$

**Non-separable setting**
In practice our training data may not be separable. What issues arise with the optimization problem above when data is not separable?

- For every $\boldsymbol{w}$ there exists a training point $\boldsymbol{x}_i$ such that

$$y_i[\boldsymbol{w}^\top \boldsymbol{x}_i + b] \leq 0$$

# SVM for Non-separable Data

**SVM formulation for separable data**

$$\min_{\boldsymbol{w},b} \quad \frac{1}{2}\|\boldsymbol{w}\|_2^2$$
$$\text{s.t.} \quad y_n[\boldsymbol{w}^\top \boldsymbol{x}_n + b] \geq 1, \quad \forall \quad n$$

**Non-separable setting**
In practice our training data may not be separable. What issues arise with the optimization problem above when data is not separable?

- For every $\boldsymbol{w}$ there exists a training point $\boldsymbol{x}_i$ such that

$$y_i[\boldsymbol{w}^\top \boldsymbol{x}_i + b] \leq 0$$

- There is no feasible $(\boldsymbol{w}, b)$ as at least one of our constraints is violated!

# SVM for Non-separable Data

**Constraints in separable setting**

$$y_n[\boldsymbol{w}^\top \boldsymbol{x}_n + b] \geq 1, \quad \forall \quad n$$

**Constraints in non-separable setting**
Can we modify our constraints to account for non-separability?

# SVM for Non-separable Data

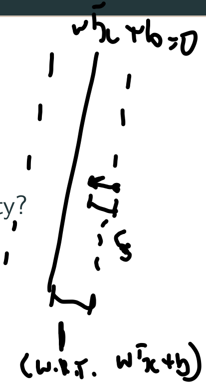
**Constraints in separable setting**

$$y_n[\boldsymbol{w}^\top \boldsymbol{x}_n + b] \geq 1, \quad \forall \quad n$$

**Constraints in non-separable setting**
Can we modify our constraints to account for non-separability?

Specifically, we introduce slack variables $\xi_n \geq 0$:

$$y_n[\boldsymbol{w}^\top \boldsymbol{x}_n + b] \geq 1 - \xi_n, \quad \forall \quad n$$



# SVM for Non-separable Data

**Constraints in separable setting**

$$y_n[\boldsymbol{w}^\top \boldsymbol{x}_n + b] \geq 1, \quad \forall \quad n$$

**Constraints in non-separable setting**
Can we modify our constraints to account for non-separability?

Specifically, we introduce slack variables $\xi_n \geq 0$:

$$y_n[\boldsymbol{w}^\top \boldsymbol{x}_n + b] \geq 1 - \xi_n, \quad \forall \quad n$$

- For "hard" training points, we can increase $\xi_n$ until the above inequalities are met.

# SVM for Non-separable Data

**Constraints in separable setting**

$$y_n[\boldsymbol{w}^\top \boldsymbol{x}_n + b] \geq 1, \quad \forall \quad n$$

**Constraints in non-separable setting**
Can we modify our constraints to account for non-separability?

Specifically, we introduce slack variables $\xi_n \geq 0$:

$$y_n[\boldsymbol{w}^\top \boldsymbol{x}_n + b] \geq 1 - \xi_n, \quad \forall \quad n$$

- For "hard" training points, we can increase $\xi_n$ until the above inequalities are met.
- What does it mean when $\xi_n = 0$?

# SVM for Non-separable Data

**Constraints in separable setting**

$$y_n[\boldsymbol{w}^\top \boldsymbol{x}_n + b] \geq 1, \quad \forall \quad n$$

**Constraints in non-separable setting**

Can we modify our constraints to account for non-separability?

Specifically, we introduce slack variables $\xi_n \geq 0$:

$$y_n[\boldsymbol{w}^\top \boldsymbol{x}_n + b] \geq 1 - \xi_n, \quad \forall \quad n$$

- For "hard" training points, we can increase $\xi_n$ until the above inequalities are met.
- What does it mean when $\xi_n = 0$? This data point is correctly classified.

# SVM for Non-separable Data

**Constraints in separable setting**

$$y_n[\boldsymbol{w}^\top \boldsymbol{x}_n + b] \geq 1, \quad \forall \quad n$$

**Constraints in non-separable setting**
Can we modify our constraints to account for non-separability?

Specifically, we introduce slack variables $\xi_n \geq 0$:

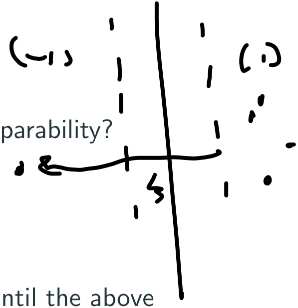
$$y_n[\boldsymbol{w}^\top \boldsymbol{x}_n + b] \geq 1 - \xi_n, \quad \forall \quad n$$

- For "hard" training points, we can increase $\xi_n$ until the above inequalities are met.
- What does it mean when $\xi_n = 0$? This data point is correctly classified.
- What does it mean when $\xi_n$ is very large?

# SVM for Non-separable Data

**Constraints in separable setting**

$$y_n[\boldsymbol{w}^\top \boldsymbol{x}_n + b] \geq 1, \quad \forall \quad n$$




**Constraints in non-separable setting**
Can we modify our constraints to account for non-separability?

Specifically, we introduce slack variables $\xi_n \geq 0$:

$$y_n[\boldsymbol{w}^\top \boldsymbol{x}_n + b] \geq 1 - \xi_n, \quad \forall \quad n$$

- For "hard" training points, we can increase $\xi_n$ until the above inequalities are met.
- What does it mean when $\xi_n = 0$? This data point is correctly classified.
- What does it mean when $\xi_n$ is very large? We have violated the original constraints "by a lot."

# Soft-margin SVM Formulation

We do not want $\xi_n$ to grow too large, and we can control their size by incorporating them into our optimization problem:

we pick

$$\min_{\boldsymbol{w},b,\boldsymbol{\xi}} \quad \frac{1}{2}\|\boldsymbol{w}\|_2^2 + C\sum_n \xi_n$$

$$\text{s.t.} \quad y_n[\boldsymbol{w}^\top \boldsymbol{x}_n + b] \geq 1 - \xi_n, \quad \forall \quad n$$

$$\xi_n \geq 0, \quad \forall \ n$$

# Soft-margin SVM Formulation

We do not want $\xi_n$ to grow too large, and we can control their size by incorporating them into our optimization problem:

$$\begin{aligned}\min_{\boldsymbol{w},b,\boldsymbol{\xi}} \quad & \frac{1}{2}\|\boldsymbol{w}\|_2^2 + C\sum_n \xi_n \\ \text{s.t.} \quad & y_n[\boldsymbol{w}^\top \boldsymbol{x}_n + b] \geq 1 - \xi_n, \quad \forall \;\; n \\ & \xi_n \geq 0, \quad \forall \; n\end{aligned}$$

What is the role of $C$?

# Soft-margin SVM Formulation

We do not want $\xi_n$ to grow too large, and we can control their size by incorporating them into our optimization problem:

$$\begin{aligned} \min_{\boldsymbol{w}, b, \boldsymbol{\xi}} \quad & \frac{1}{2}\|\boldsymbol{w}\|_2^2 + C \sum_n \xi_n \\ \text{s.t.} \quad & y_n[\boldsymbol{w}^\top \boldsymbol{x}_n + b] \geq 1 - \xi_n, \quad \forall \quad n \\ & \xi_n \geq 0, \quad \forall \ n \end{aligned}$$

What is the role of $C$?

- User-defined hyperparameter

# Soft-margin SVM Formulation

We do not want $\xi_n$ to grow too large, and we can control their size by incorporating them into our optimization problem:

$$\begin{aligned} \min_{\boldsymbol{w}, b, \boldsymbol{\xi}} \quad & \frac{1}{2}\|\boldsymbol{w}\|_2^2 + C\sum_n \xi_n \\ \text{s.t.} \quad & y_n[\boldsymbol{w}^\top \boldsymbol{x}_n + b] \geq 1 - \xi_n, \quad \forall \quad n \\ & \xi_n \geq 0, \quad \forall\ n \end{aligned}$$

What is the role of $C$?

- User-defined hyperparameter
- Trades off between the two terms in our objective

# Soft-margin SVM Formulation

We do not want $\xi_n$ to grow too large, and we can control their size by incorporating them into our optimization problem:

$$\begin{aligned} \min_{\boldsymbol{w}, b, \boldsymbol{\xi}} \quad & \frac{1}{2}\|\boldsymbol{w}\|_2^2 + C\sum_n \xi_n \\ \text{s.t.} \quad & y_n[\boldsymbol{w}^\top \boldsymbol{x}_n + b] \geq 1 - \xi_n, \quad \forall \; n \\ & \xi_n \geq 0, \quad \forall \; n \end{aligned}$$

What is the role of $C$?

- User-defined hyperparameter
- Trades off between the two terms in our objective
- Same idea as the regularization term in ridge regression

# How to Solve this Problem?

$$
\begin{aligned}
\min_{\boldsymbol{w}, b, \boldsymbol{\xi}} \quad & \frac{1}{2}\|\boldsymbol{w}\|_2^2 + C \sum_n \xi_n \\
\text{s.t.} \quad & y_n[\boldsymbol{w}^\top \boldsymbol{x}_n + b] \geq 1 - \xi_n, \quad \forall \ n \\
& \xi_n \geq 0, \quad \forall \ n
\end{aligned}
$$

- This is a convex quadratic program: the objective function is quadratic in $\boldsymbol{w}$ and linear in $\boldsymbol{\xi}$ and the constraints are linear (inequality) constraints in $\boldsymbol{w}$, $b$ and $\xi_n$.

# How to Solve this Problem?

$$
\begin{aligned}
\min_{\boldsymbol{w}, b, \boldsymbol{\xi}} \quad & \frac{1}{2}\|\boldsymbol{w}\|_2^2 + C \sum_n \xi_n \\
\text{s.t.} \quad & y_n[\boldsymbol{w}^\top \boldsymbol{x}_n + b] \geq 1 - \xi_n, \quad \forall \ n \\
& \xi_n \geq 0, \quad \forall \ n
\end{aligned}
$$

- This is a convex quadratic program: the objective function is quadratic in $\boldsymbol{w}$ and linear in $\boldsymbol{\xi}$ and the constraints are linear (inequality) constraints in $\boldsymbol{w}$, $b$ and $\xi_n$.
- Early solvers were based on general-purpose quadratic program solvers (e.g. similar to `scipy.optimize` or Matlab's `quadprog()`, albeit in the 1990s)

# How to Solve this Problem?

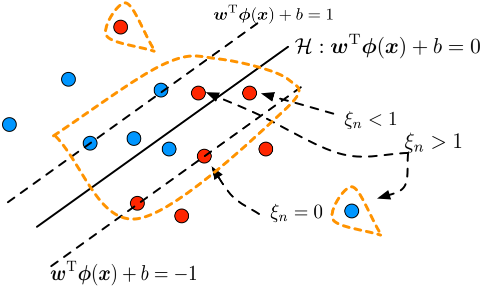
$$\min_{\boldsymbol{w},b,\boldsymbol{\xi}} \quad \frac{1}{2}\|\boldsymbol{w}\|_2^2 + C\sum_n \xi_n$$

$$\text{s.t.} \quad y_n[\boldsymbol{w}^\top \boldsymbol{x}_n + b] \geq 1 - \xi_n, \quad \forall \quad n$$

$$\xi_n \geq 0, \quad \forall \; n$$

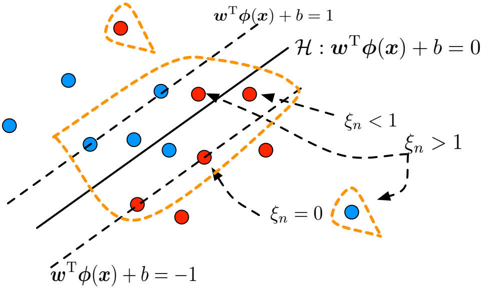
- This is a convex quadratic program: the objective function is quadratic in $\boldsymbol{w}$ and linear in $\boldsymbol{\xi}$ and the constraints are linear (inequality) constraints in $\boldsymbol{w}$, $b$ and $\xi_n$.
- Early solvers were based on general-purpose quadratic program solvers (e.g. similar to `scipy.optimize` or Matlab's `quadprog()`, albeit in the 1990s)
- SVM solvers today are based on highly optimized search algorithms that exploit SVM-specific structure.

# Support Vectors: Revisited

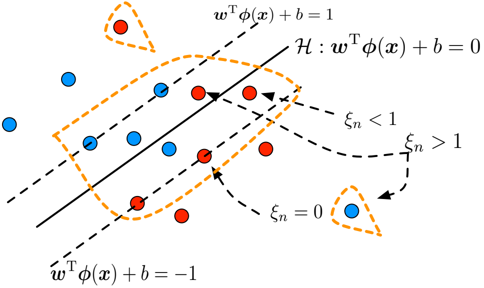



**Support vectors** are highlighted by the dotted orange lines. What does this mean mathematically?

# Support Vectors: Revisited

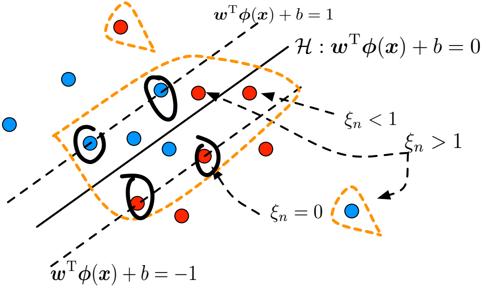



Recall the constraints $y_n[\boldsymbol{w}^\top \boldsymbol{x}_n + b] \geq 1 - \xi_n$ from the soft-margin formulation. All the training points $(\boldsymbol{x}_n, y_n)$ that satisfies the constraint with "$=$" are support vectors.



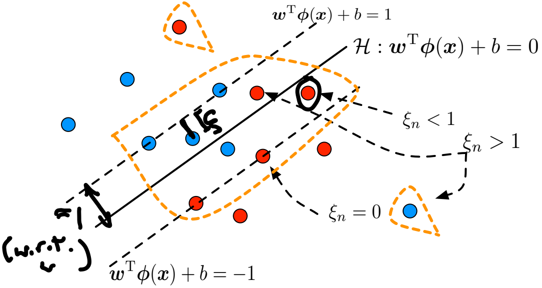

In other words, support vectors satisfy $y_n[\boldsymbol{w}^\top \boldsymbol{x}_n + b] = 1 - \xi_n$ , which can be further divided into several categories:

## Support Vectors: Revisited




In other words, support vectors satisfy $y_n[\boldsymbol{w}^\top \boldsymbol{x}_n + b] = 1 - \xi_n$ , which can be further divided into several categories:

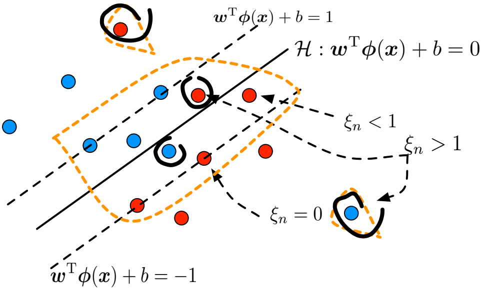
- $\xi_n = 0$: $y_n[\boldsymbol{w}^\top \boldsymbol{x}_n + b] = 1$, the point is on the correct side with distance $\frac{1}{\|\boldsymbol{w}\|}$.




In other words, support vectors satisfy $y_n[\boldsymbol{w}^\top \boldsymbol{x}_n + b] = 1 - \xi_n$, which can be further divided into several categories:

- $\xi_n = 0$: $y_n[\boldsymbol{w}^\top \boldsymbol{x}_n + b] = 1$, the point is on the correct side with distance $\frac{1}{\|\boldsymbol{w}\|}$.
- $0 < \xi_n \leq 1$: $y_n[\boldsymbol{w}^\top \boldsymbol{x}_n + b] \in [0, 1)$ on the correct side, but with distance less than $\frac{1}{\|\boldsymbol{w}\|}$.

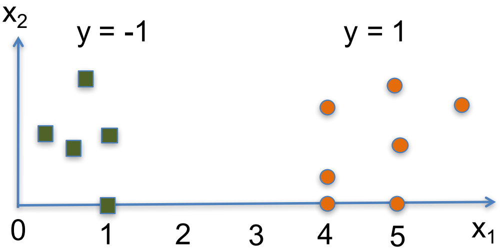
# Support Vectors: Revisited



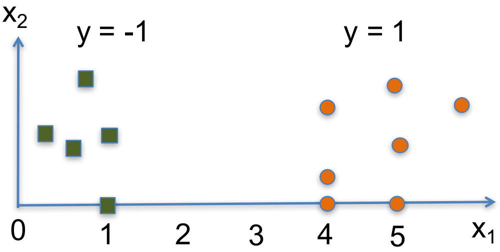

In other words, support vectors satisfy $y_n[\boldsymbol{w}^\top \boldsymbol{x}_n + b] = 1 - \xi_n$, which can be further divided into several categories:

- $\xi_n = 0$: $y_n[\boldsymbol{w}^\top \boldsymbol{x}_n + b] = 1$, the point is on the correct side with distance $\frac{1}{\|\boldsymbol{w}\|}$.
- $0 < \xi_n \leq 1$: $y_n[\boldsymbol{w}^\top \boldsymbol{x}_n + b] \in [0, 1)$ on the correct side, but with distance less than $\frac{1}{\|\boldsymbol{w}\|}$.
- $\xi_n > 1$: $y_n[\boldsymbol{w}^\top \boldsymbol{x}_n + b] < 0$, on the wrong side of the boundary.

# Example of SVM

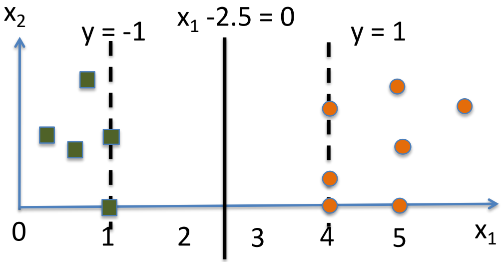


# Example of SVM



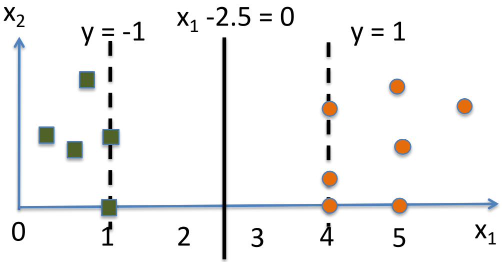

What will be the decision boundary learnt by solving the SVM optimization problem?

# Example of SVM




Margin $= 1.5$; the decision boundary has $\mathbf{w} = [1, 0]^\top$, and $b = -2.5$.

# Example of SVM

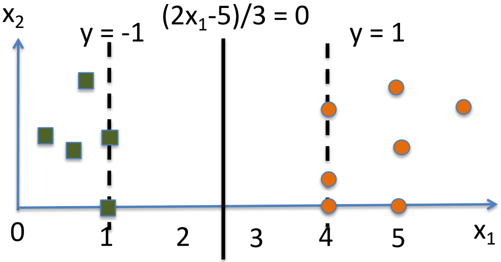



Margin $= 1.5$; the decision boundary has $\mathbf{w} = [1, 0]^\top$, and $b = -2.5$.

Not quite: we need the support vectors to satisfy to $y_n(\mathbf{w}^\top \mathbf{x}_n + b) = 1$. For example, for $\mathbf{x}_n = [1, 0]^\top$, we have

$$y_n(\mathbf{w}^\top \mathbf{x}_n + b) = (-1)[1 - 2.5] = 1.5.$$

# Example of SVM: Scaling



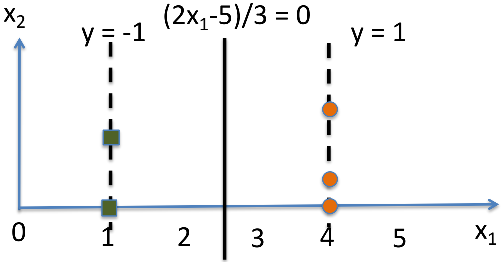

Thus, our optimization problem will re-scale **w** and $b$ to get this equation for the same decision boundary.

Margin $= 1.5$; the decision boundary has $\mathbf{w} = [2/3, 0]^\top$, and $b = -5/3$.

For example, for $\mathbf{x}_n = [1, 0]^\top$, we have

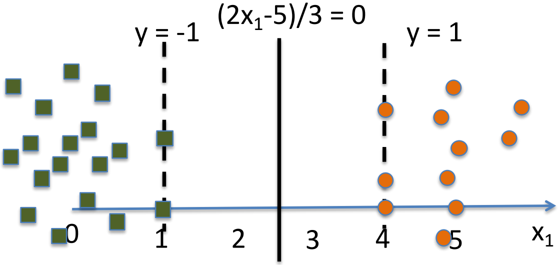
$$y_n(\mathbf{w}^\top \mathbf{x}_n + b) = (-1)[2/3 - 5/3] = 1.$$

# Example of SVM: Support Vectors



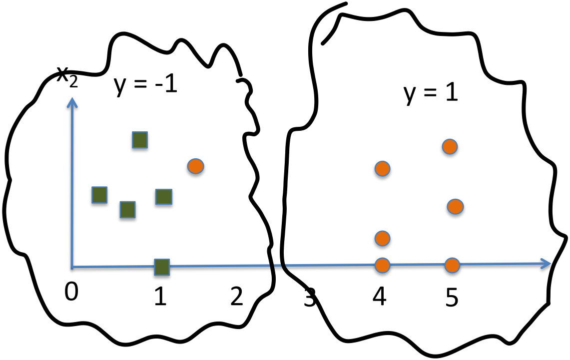

The solution to our optimization problem will be the **same** to the *reduced* dataset containing all the support vectors.

# Example of SVM: Support Vectors



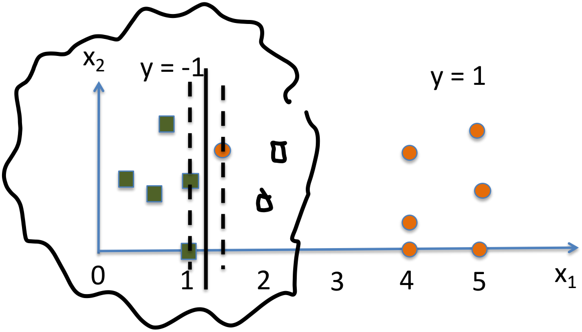

There can be many more data than the number of support vectors (so we can train on a smaller dataset).

# Example of SVM: Resilience to Outliers



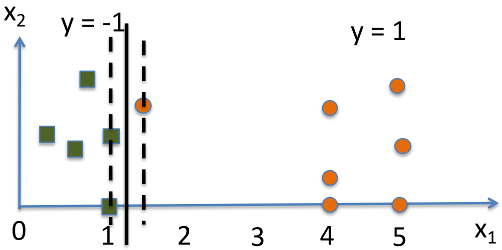

- Still linearly separable, but one of the orange dots is an "outlier".

# Example of SVM: Resilience to Outliers



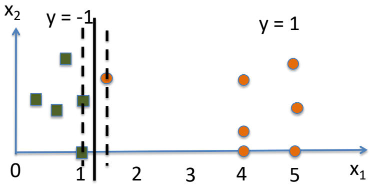

- Naively applying the hard-margin SVM will result in a classifier with small margin.

# Example of SVM: Resilience to Outliers




- Naively applying the hard-margin SVM will result in a classifier with small margin.
- So, better to use the soft-margin (or equivalently, hinge loss) formulation.

# Example of SVM: Resilience to Outliers



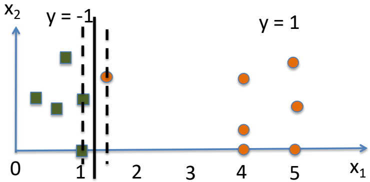

$$\min_{\boldsymbol{w},b,\boldsymbol{\xi}} \quad \frac{1}{2}\|\boldsymbol{w}\|_2^2 + C\sum_n \xi_n$$

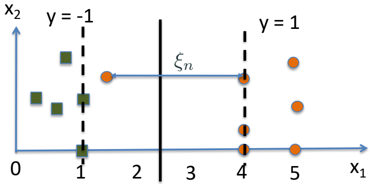
$$\text{s.t.} \quad y_n[\boldsymbol{w}^\top \boldsymbol{x}_n + b] \geq 1 - \xi_n, \quad \forall \quad n$$

$$\xi_n \geq 0, \quad \forall \ n$$

Due to the flexibility provided by $C$, (properly tuned) SVM is less sensitive to outliers.

# Example of SVM: Resilience to Outliers

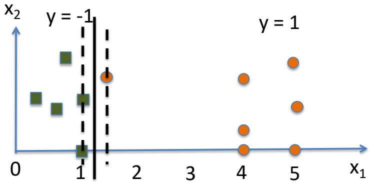


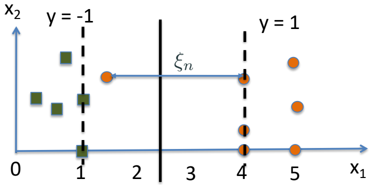




$$\begin{aligned}\min_{\boldsymbol{w},b,\boldsymbol{\xi}} \quad & \frac{1}{2}\|\boldsymbol{w}\|_2^2 + C\sum_n \xi_n \\ \text{s.t.} \quad & y_n[\boldsymbol{w}^\top \boldsymbol{x}_n + b] \geq 1 - \xi_n, \quad \forall \; n \\ & \xi_n \geq 0, \quad \forall \; n\end{aligned}$$

- What happens if $C$ is very small?

# Example of SVM: Resilience to Outliers



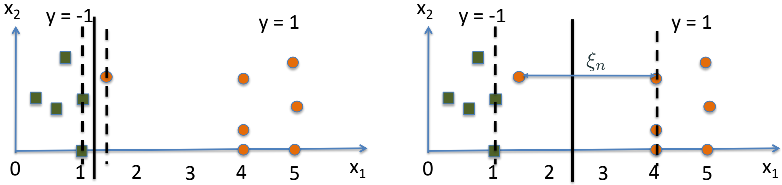

$$\min_{\boldsymbol{w}, b, \boldsymbol{\xi}} \quad \frac{1}{2}\|\boldsymbol{w}\|_2^2 + C \sum_n \xi_n$$

$$\text{s.t.} \quad y_n[\boldsymbol{w}^\top \boldsymbol{x}_n + b] \geq 1 - \xi_n, \quad \forall \quad n$$

$$\xi_n \geq 0, \quad \forall \quad n$$

- What happens if $C$ is very small? More data points near the boundary are disregarded.

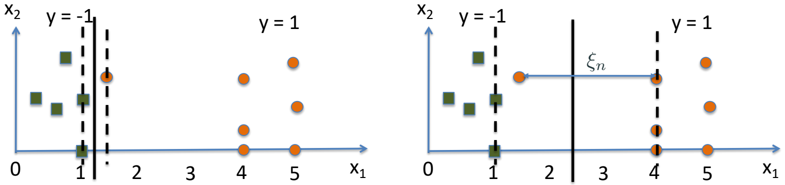
# Example of SVM: Resilience to Outliers




$$\begin{aligned} \min_{\boldsymbol{w},b,\boldsymbol{\xi}} \quad & \frac{1}{2}\|\boldsymbol{w}\|_2^2 + C\cancel{\sum_n}\xi_n \\ \text{s.t.} \quad & y_n[\boldsymbol{w}^\top \boldsymbol{x}_n + b] \geq 1 - \xi_n, \quad \forall \quad n \\ & \xi_n \geq 0, \quad \forall \quad n \end{aligned}$$

- What happens if $C$ is very small? More data points near the boundary are disregarded.
- What happens if $C$ is 0?

# Example of SVM: Resilience to Outliers



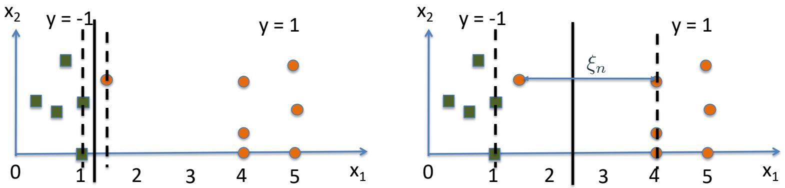

$$\min_{\boldsymbol{w},b,\boldsymbol{\xi}} \quad \frac{1}{2}\|\boldsymbol{w}\|_2^2 + C\sum_n \xi_n$$

$$\text{s.t.} \quad y_n[\boldsymbol{w}^\top \boldsymbol{x}_n + b] \geq 1 - \xi_n, \quad \forall \quad n$$

$$\xi_n \geq 0, \quad \forall \; n$$

- What happens if $C$ is very small? More data points near the boundary are disregarded.
- What happens if $C$ is 0? All data points will be ignored.

# Example of SVM: Resilience to Outliers



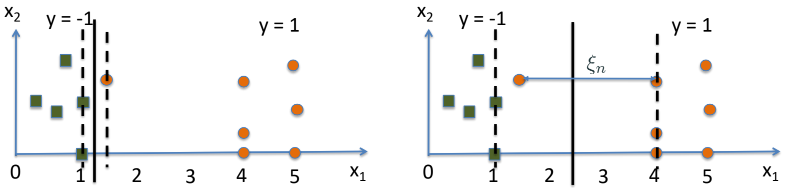

$$\min_{\boldsymbol{w},b,\boldsymbol{\xi}} \quad \frac{1}{2}\|\boldsymbol{w}\|_2^2 + C\sum_n \xi_n$$

$$\text{s.t.} \quad y_n[\boldsymbol{w}^\top \boldsymbol{x}_n + b] \geq 1 - \xi_n, \quad \forall \quad n$$

$$\xi_n \geq 0, \quad \forall \ n$$

- What happens if $C$ is very large?

# Example of SVM: Resilience to Outliers



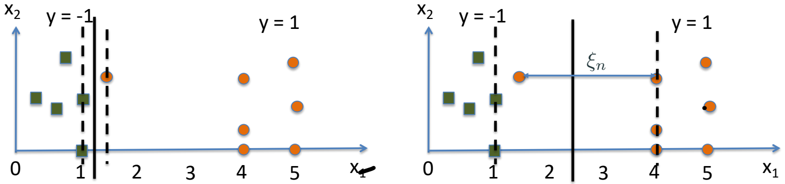

$$\min_{\boldsymbol{w},b,\boldsymbol{\xi}} \quad \frac{1}{2}\|\boldsymbol{w}\|_2^2 + C\sum_n \xi_n$$

$$\text{s.t.} \quad y_n[\boldsymbol{w}^\top \boldsymbol{x}_n + b] \geq 1 - \xi_n, \quad \forall \quad n$$

$$\xi_n \geq 0, \quad \forall \; n$$

- What happens if $C$ is very large? Outliers near the decision boundary will have a greater impact.

# Example of SVM: Resilience to Outliers



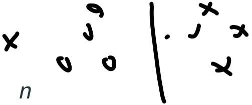

$\infty$

"unfeasible"

$$\min_{\boldsymbol{w},b,\boldsymbol{\xi}} \quad \frac{1}{2}\|\boldsymbol{w}\|_2^2 + C\sum_n \xi_n$$

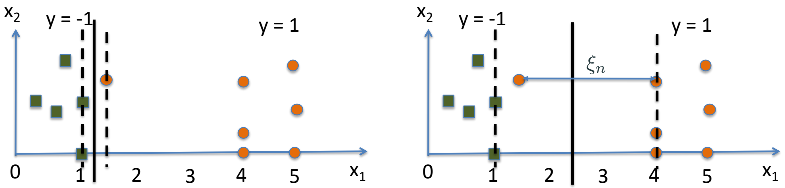
$$\text{s.t.} \quad y_n[\boldsymbol{w}^\top \boldsymbol{x}_n + b] \geq 1 - \xi_n, \quad \forall \; n$$

$$\xi_n \geq 0, \quad \forall \; n$$



- What happens if $C$ is very large? Outliers near the decision boundary will have a greater impact.
- What happens if $C$ is $\infty$?

# Example of SVM: Resilience to Outliers




$$\min_{\boldsymbol{w}, b, \boldsymbol{\xi}} \quad \frac{1}{2}\|\boldsymbol{w}\|_2^2 + C\sum_n \xi_n$$

$$\text{s.t.} \quad y_n[\boldsymbol{w}^\top \boldsymbol{x}_n + b] \geq 1 - \xi_n, \quad \forall \quad n$$

$$\xi_n \geq 0, \quad \forall \; n$$

- What happens if $C$ is very large? Outliers near the decision boundary will have a greater impact.
- What happens if $C$ is $\infty$? We get hard margin SVM.

# Hinge Loss Formulation

# SVM vs. Logistic Regression

**SVM soft-margin formulation**

$$\min_{\boldsymbol{w},b,\boldsymbol{\xi}} \quad \frac{1}{2}\|\boldsymbol{w}\|_2^2 + C\sum_n \xi_n$$

$$\text{s.t.} \quad y_n[\boldsymbol{w}^\top \boldsymbol{x}_n + b] \geq 1 - \xi_n, \ \forall \ n$$

$$\xi_n \geq 0, \quad \forall \ n$$

**Logistic regression formulation**

$$\min_{\boldsymbol{w}} - \sum_n \{y_n \log \sigma(\boldsymbol{w}^\top \boldsymbol{x}_n) + (1 - y_n)\log[1 - \sigma(\boldsymbol{w}^\top \boldsymbol{x}_n)]\} + \frac{\lambda}{2}\|\boldsymbol{w}\|_2^2$$

# SVM vs. Logistic Regression

**SVM soft-margin formulation**

$$\min_{\boldsymbol{w},b,\boldsymbol{\xi}} \quad \frac{1}{2}\|\boldsymbol{w}\|_2^2 + C\sum_n \xi_n$$
$$\text{s.t.} \quad y_n[\boldsymbol{w}^\top \boldsymbol{x}_n + b] \geq 1 - \xi_n, \ \forall\ n$$
$$\xi_n \geq 0, \quad \forall\ n$$

**Logistic regression formulation**

$$\min_{\boldsymbol{w}} - \sum_n \{y_n \log \sigma(\boldsymbol{w}^\top \boldsymbol{x}_n) + (1-y_n)\log[1-\sigma(\boldsymbol{w}^\top \boldsymbol{x}_n)]\} + \frac{\lambda}{2}\|\boldsymbol{w}\|_2^2$$

- Logistic regression defines a loss for each data point and minimizes the total loss plus a regularization term.

# SVM vs. Logistic Regression

**SVM soft-margin formulation**

$$\begin{aligned}\min_{\boldsymbol{w},b,\boldsymbol{\xi}} \quad & \frac{1}{2}\|\boldsymbol{w}\|_2^2 + C\sum_n \xi_n \\ \text{s.t.} \quad & y_n[\boldsymbol{w}^\top \boldsymbol{x}_n + b] \geq 1 - \xi_n, \ \forall\ n \\ & \xi_n \geq 0, \quad \forall\ n\end{aligned}$$

**Logistic regression formulation**

$$\begin{aligned}\min_{\boldsymbol{w}} - \sum_n \{ & y_n \log \sigma(\boldsymbol{w}^\top \boldsymbol{x}_n) \\ & + (1 - y_n) \log[1 - \sigma(\boldsymbol{w}^\top \boldsymbol{x}_n)]\} \\ & + \frac{\lambda}{2}\|\boldsymbol{w}\|_2^2\end{aligned}$$

- Logistic regression defines a loss for each data point and minimizes the total loss plus a regularization term.
- This is convenient for assessing the “goodness” of the model on each data point.

# SVM vs. Logistic Regression

**SVM soft-margin formulation**

$$\begin{aligned}\min_{\boldsymbol{w},b,\boldsymbol{\xi}} \quad & \frac{1}{2}\|\boldsymbol{w}\|_2^2 + C\sum_n \xi_n \\ \text{s.t.} \quad & y_n[\boldsymbol{w}^\top \boldsymbol{x}_n + b] \geq 1 - \xi_n, \ \forall\ n \\ & \xi_n \geq 0, \quad \forall\ n\end{aligned}$$

**Logistic regression formulation**

$$\begin{aligned}\min_{\boldsymbol{w}} -\sum_n \{ & y_n \log \sigma(\boldsymbol{w}^\top \boldsymbol{x}_n) \\ & + (1 - y_n)\log[1 - \sigma(\boldsymbol{w}^\top \boldsymbol{x}_n)]\} \\ & + \frac{\lambda}{2}\|\boldsymbol{w}\|_2^2\end{aligned}$$

- Logistic regression defines a loss for each data point and minimizes the total loss plus a regularization term.
- This is convenient for assessing the “goodness” of the model on each data point.
- Can we write SVMs in this form as well?

# SVM vs. Logistic Regression

**SVM soft-margin formulation**

$$\min_{\boldsymbol{w}, b, \boldsymbol{\xi}} \quad \frac{1}{2}\|\boldsymbol{w}\|_2^2 + C \sum_n \xi_n$$

$$\text{s.t.} \quad y_n[\boldsymbol{w}^\top \boldsymbol{x}_n + b] \geq 1 - \xi_n, \ \forall\ n$$

$$\xi_n \geq 0, \quad \forall\ n$$

**Logistic regression formulation**

$$\min_{\boldsymbol{w}} - \sum_n \{y_n \log \sigma(\boldsymbol{w}^\top \boldsymbol{x}_n) + (1 - y_n) \log[1 - \sigma(\boldsymbol{w}^\top \boldsymbol{x}_n)]\} + \frac{\lambda}{2} \|\boldsymbol{w}\|_2^2$$

- Logistic regression defines a loss for each data point and minimizes the total loss plus a regularization term.
- This is convenient for assessing the "goodness" of the model on each data point.
- Can we write SVMs in this form as well? The Hinge Loss formulation!

# Derive the Hinge Loss Formulation

**Here's the soft-margin formulation again:**

$$\min_{\boldsymbol{w},b,\boldsymbol{\xi}} \frac{1}{2}\|\boldsymbol{w}\|_2^2 + C\sum_n \xi_n \quad \text{s.t. } y_n[\boldsymbol{w}^\top \boldsymbol{x}_n + b] \geq 1 - \xi_n,\ \xi_n \geq 0,\ \forall\ n$$

# Derive the Hinge Loss Formulation

**Here's the soft-margin formulation again:**

$$\min_{\boldsymbol{w},b,\boldsymbol{\xi}} \frac{1}{2}\|\boldsymbol{w}\|_2^2 + C\sum_n \xi_n \quad \text{s.t. } y_n[\boldsymbol{w}^\top \boldsymbol{x}_n + b] \geq 1 - \xi_n, \ \xi_n \geq 0, \ \forall\ n$$

Now since $y_n[\boldsymbol{w}^\top \boldsymbol{x}_n + b] \geq 1 - \xi_n \iff \xi_n \geq 1 - y_n[\boldsymbol{w}^\top \boldsymbol{x}_n + b]$:

# Derive the Hinge Loss Formulation

**Here's the soft-margin formulation again:**

$$\min_{\boldsymbol{w},b,\boldsymbol{\xi}} \frac{1}{2}\|\boldsymbol{w}\|_2^2 + C\sum_n \xi_n \quad \text{s.t. } y_n[\boldsymbol{w}^\top \boldsymbol{x}_n + b] \geq 1 - \xi_n,\ \xi_n \geq 0,\ \forall\ n$$

Now since $y_n[\boldsymbol{w}^\top \boldsymbol{x}_n + b] \geq 1 - \xi_n \Longleftrightarrow \xi_n \geq 1 - y_n[\boldsymbol{w}^\top \boldsymbol{x}_n + b]$:

$$\min_{\boldsymbol{w},b,\boldsymbol{\xi}} C\sum_n \xi_n + \frac{1}{2}\|\boldsymbol{w}\|_2^2 \text{ s.t. } \xi_n \geq \max(0, 1 - y_n[\boldsymbol{w}^\top \boldsymbol{x}_n + b]),\ \forall\ n$$

# Derive the Hinge Loss Formulation

**Here's the soft-margin formulation again:**

$$\min_{\boldsymbol{w},b,\boldsymbol{\xi}} \frac{1}{2}\|\boldsymbol{w}\|_2^2 + C\sum_n \xi_n \text{ s.t. } y_n[\boldsymbol{w}^\top \boldsymbol{x}_n + b] \geq 1 - \xi_n,\ \xi_n \geq 0,\ \forall\ n$$

Now since $y_n[\boldsymbol{w}^\top \boldsymbol{x}_n + b] \geq 1 - \xi_n \iff \xi_n \geq 1 - y_n[\boldsymbol{w}^\top \boldsymbol{x}_n + b]$:

$$\min_{\boldsymbol{w},b,\boldsymbol{\xi}} C\sum_n \xi_n + \frac{1}{2}\|\boldsymbol{w}\|_2^2 \text{ s.t. } \xi_n \geq \boxed{\max(0, 1 - y_n[\boldsymbol{w}^\top \boldsymbol{x}_n + b]),}\ \forall\ n$$

Now since the $\xi_n$ should always be as small as possible, we obtain:

$$\min_{\boldsymbol{w},b} C\sum_n \max(0, 1 - y_n[\boldsymbol{w}^\top \boldsymbol{x}_n + b]) + \frac{1}{2}\|\boldsymbol{w}\|_2^2$$

# Derive the Hinge Loss Formulation

**Here's the soft-margin formulation again:**

$$\min_{\boldsymbol{w},b,\boldsymbol{\xi}} \frac{1}{2}\|\boldsymbol{w}\|_2^2 + C\sum_n \xi_n \quad \text{s.t. } y_n[\boldsymbol{w}^\top \boldsymbol{x}_n + b] \geq 1 - \xi_n, \xi_n \geq 0, \ \forall\ n$$

Now since $y_n[\boldsymbol{w}^\top \boldsymbol{x}_n + b] \geq 1 - \xi_n \iff \xi_n \geq 1 - y_n[\boldsymbol{w}^\top \boldsymbol{x}_n + b]$:

$$\min_{\boldsymbol{w},b,\boldsymbol{\xi}} C\sum_n \xi_n + \frac{1}{2}\|\boldsymbol{w}\|_2^2 \text{ s.t. } \xi_n \geq \max(0, 1 - y_n[\boldsymbol{w}^\top \boldsymbol{x}_n + b]), \ \forall\ n$$

Now since the $\xi_n$ should always be as small as possible, we obtain:

$$\min_{\boldsymbol{w},b} C\sum_n \max(0, 1 - y_n[\boldsymbol{w}^\top \boldsymbol{x}_n + b]) + \frac{1}{2}\|\boldsymbol{w}\|_2^2$$

Divide by $C$ and set $\lambda = \frac{1}{C}$, we get get Hinge Loss formulation:

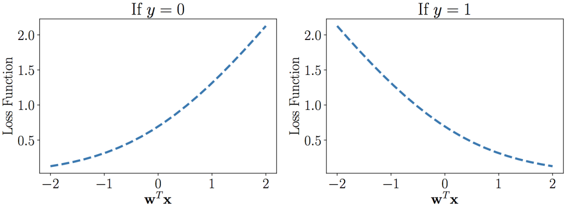
$$\min_{\boldsymbol{w},b} \sum_n \underbrace{\max(0, 1 - y_n[\boldsymbol{w}^\top \boldsymbol{x}_n + b])}_{\text{Hinge Loss for } x_n, y_n} + \frac{\lambda}{2}\|\boldsymbol{w}\|_2^2$$

# Logistic Regression Loss vs Hinge Loss

Given training data $(x_n, y_n)$, the cross entropy loss was

$$-\{y_n \log \sigma(\boldsymbol{w}^\top \boldsymbol{x}_n) + (1 - y_n) \log[1 - \sigma(\boldsymbol{w}^\top \boldsymbol{x}_n)]\}$$

$$-\log\left(\frac{e^{-\mathbf{w}^T\mathbf{x_n}}}{1+e^{-\mathbf{w}^T\mathbf{x_n}}}\right) \qquad -\log\left(\frac{1}{1+e^{-\mathbf{w}^T\mathbf{x_n}}}\right)$$



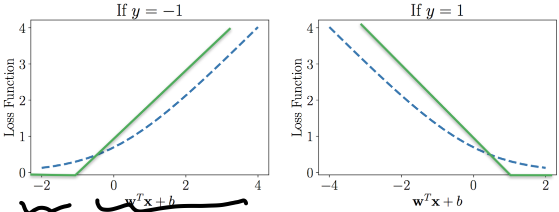

- What does the Hinge Loss Function look like?

# Logistic Regression Loss vs Hinge Loss

Given training data $(x_n, y_n)$, the Hinge loss is

$$\max(0, 1 - y_n[\boldsymbol{w}^\top \boldsymbol{x}_n + b])$$

$\max\left(\mathbf{w}^T\mathbf{x_n} + b + 1, 0\right)$ $\qquad$ $\max\left(1 - \mathbf{w}^T\mathbf{x_n} - b, 0\right)$




not violating $\qquad$ violating

i.e. not a support vector

- Loss grows linearly as we move away from the boundary.
- No penalty if a point is more than 1 unit from the boundary.

# Hinge Loss SVM Formulation

**Minimizing the total hinge loss on all the training data**

$$\min_{\boldsymbol{w},b} \sum_n \underbrace{\max(0, 1 - y_n[\boldsymbol{w}^\top \boldsymbol{x}_n + b])}_{\text{hinge loss for sample } n} + \underbrace{\frac{\lambda}{2}\|\boldsymbol{w}\|_2^2}_{\text{regularizer}}$$

Analogous to regularized least squares or logistic regression, as we balance between two terms (the loss and the regularizer).

# Hinge Loss SVM Formulation

**Minimizing the total hinge loss on all the training data**

$$\min_{\boldsymbol{w},b} \sum_n \underbrace{\max(0, 1 - y_n[\boldsymbol{w}^\top \boldsymbol{x}_n + b])}_{\text{hinge loss for sample } n} + \underbrace{\frac{\lambda}{2}\|\boldsymbol{w}\|_2^2}_{\text{regularizer}}$$

Analogous to regularized least squares or logistic regression, as we balance between two terms (the loss and the regularizer).

- Can solve using gradient descent to get the optimal **w** and $b$

## Hinge Loss SVM Formulation

**Minimizing the total hinge loss on all the training data**

$$\min_{\boldsymbol{w},b} \sum_n \underbrace{\max(0, 1 - y_n[\boldsymbol{w}^\top \boldsymbol{x}_n + b])}_{\text{hinge loss for sample } n} + \underbrace{\frac{\lambda}{2}\|\boldsymbol{w}\|_2^2}_{\text{regularizer}}$$

Analogous to regularized least squares or logistic regression, as we balance between two terms (the loss and the regularizer).

- Can solve using gradient descent to get the optimal $\mathbf{w}$ and $b$
- Gradient of the first term will be either 0, $\mathbf{x}_n$ or $-\mathbf{x}_n$ depending on $y_n$ and $\boldsymbol{w}^\top \boldsymbol{x}_n + b$.

# Hinge Loss SVM Formulation

**Minimizing the total hinge loss on all the training data**

$$\min_{\boldsymbol{w},b} \sum_n \underbrace{\max(0, 1 - y_n[\boldsymbol{w}^\top \boldsymbol{x}_n + b])}_{\text{hinge loss for sample } n} + \underbrace{\frac{\lambda}{2}\|\boldsymbol{w}\|_2^2}_{\text{regularizer}}$$

Analogous to regularized least squares or logistic regression, as we balance between two terms (the loss and the regularizer).

- Can solve using gradient descent to get the optimal $\mathbf{w}$ and $b$
- Gradient of the first term will be either 0, $\mathbf{x}_n$ or $-\mathbf{x}_n$ depending on $y_n$ and $\boldsymbol{w}^\top \boldsymbol{x}_n + b$.
- Much easier to compute than in logistic regression, where we need to compute the sigmoid function $\sigma(\boldsymbol{w}^\top \boldsymbol{x}_n + b)$ in each iteration.

# Summary

# Three SVM Formulations

**Hard-margin (for separable data)**

$$\min_{\boldsymbol{w}, b, \boldsymbol{\xi}} \ \frac{1}{2}\|\boldsymbol{w}\|_2^2 \text{ s.t. } y_n[\boldsymbol{w}^\top \boldsymbol{x}_n + b] \geq 1, \ \xi_n \geq 0, \ \forall \ n$$

# Three SVM Formulations

**Hard-margin (for separable data)**

$$\min_{\boldsymbol{w},b,\boldsymbol{\xi}} \; \frac{1}{2}\|\boldsymbol{w}\|_2^2 \text{ s.t. } y_n[\boldsymbol{w}^\top \boldsymbol{x}_n + b] \geq 1, \; \xi_n \geq 0, \; \forall \; n$$

**Soft-margin (add slack variables)**

$$\min_{\boldsymbol{w},b,\boldsymbol{\xi}} \; \frac{1}{2}\|\boldsymbol{w}\|_2^2 + C\sum_n \xi_n \;\; \text{s.t. } y_n[\boldsymbol{w}^\top \boldsymbol{x}_n + b] \geq 1 - \xi_n, \; \xi_n \geq 0, \; \forall \; n$$

# Three SVM Formulations

**Hard-margin (for separable data)**

$$\min_{\boldsymbol{w},b,\boldsymbol{\xi}} \ \frac{1}{2}\|\boldsymbol{w}\|_2^2 \text{ s.t. } y_n[\boldsymbol{w}^\top \boldsymbol{x}_n + b] \geq 1, \ \xi_n \geq 0, \ \forall \ n$$

**Soft-margin (add slack variables)**

$$\min_{\boldsymbol{w},b,\boldsymbol{\xi}} \ \frac{1}{2}\|\boldsymbol{w}\|_2^2 + C\sum_n \xi_n \ \text{ s.t. } y_n[\boldsymbol{w}^\top \boldsymbol{x}_n + b] \geq 1 - \xi_n, \ \xi_n \geq 0, \ \forall \ n$$

**Hinge loss (define a loss function for each data point)**

$$\min_{\boldsymbol{w},b} \ \sum_n \max(0, 1 - y_n[\boldsymbol{w}^\top \boldsymbol{x}_n + b]) + \frac{\lambda}{2}\|\boldsymbol{w}\|_2^2$$

# Advantages of SVM

We've seen the geometric formulation of SVM and the equivalent formulation of minimizing the empirical hinge loss.

This explains why SVM:

1. Is less sensitive to outliers.
2. Maximizes distance of training data from the boundary.
3. Only requires a subset of the training points.
4. Generalizes well to many nonlinear models.
5. Scales better with high-dimensional data.

We will need to use duality to show the remaining properties.

# Summary

You should know:

- Max-margin formulation for separable and non-separable SVMs.
- Definition and importance of support vectors.
- Hinge loss formulation of SVMs.
- Equivalence of the max-margin and hinge loss formulations.

Next class:

- Duality
- Nonlinear SVM