Machine Learning 2

Support Vector Machine (SVM)

Dept. SW and Communication Engineering

Prof. Giseop Noh (kafa46@hongik.ac.kr)

Contents

Do we need to study SVM?

- Data Efficiency
- Kernel Versatility
- Mathematical Interpretability

- Computational Efficiency
- Guaranteed Global Optimum
- High-Dimensional Effectiveness

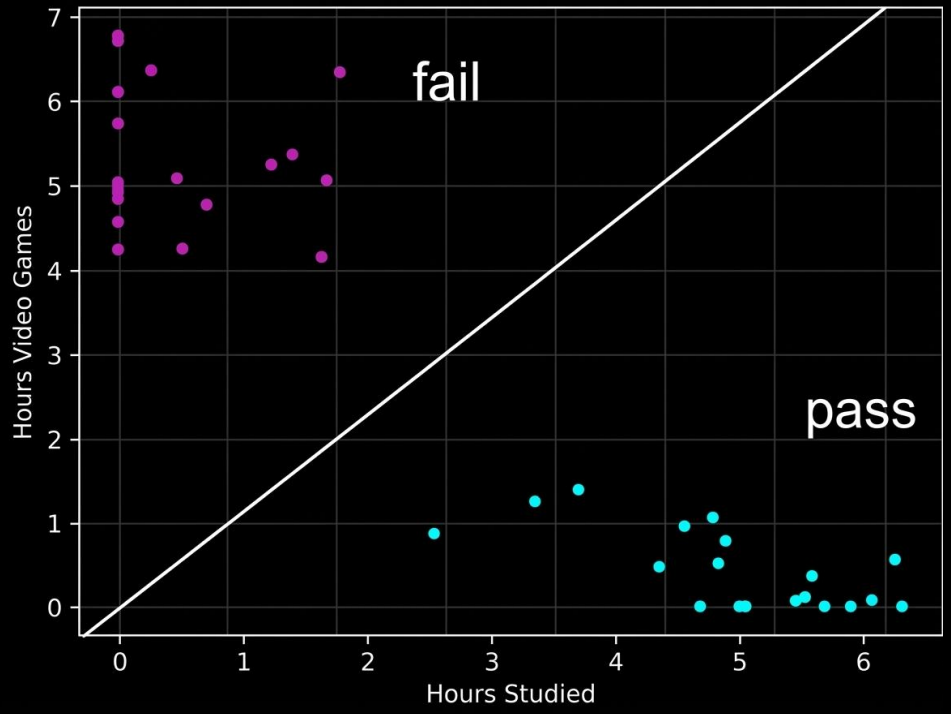
vs.

Recap: Linear Regression

Linear Classifiers

We want to predict whether a student passes or fails a ML exam.

Student	Hours Studied	Hours Video Games	Result
A	5.0	0.5	Pass
B	4.2	1.0	Pass
C	5.8	0.2	Pass
D	0.8	5.0	Fail
E	1.2	4.3	Fail
F	0.5	6.2	Fail



Different Problems, Same Geometry

Classification problems look very different in the real world.

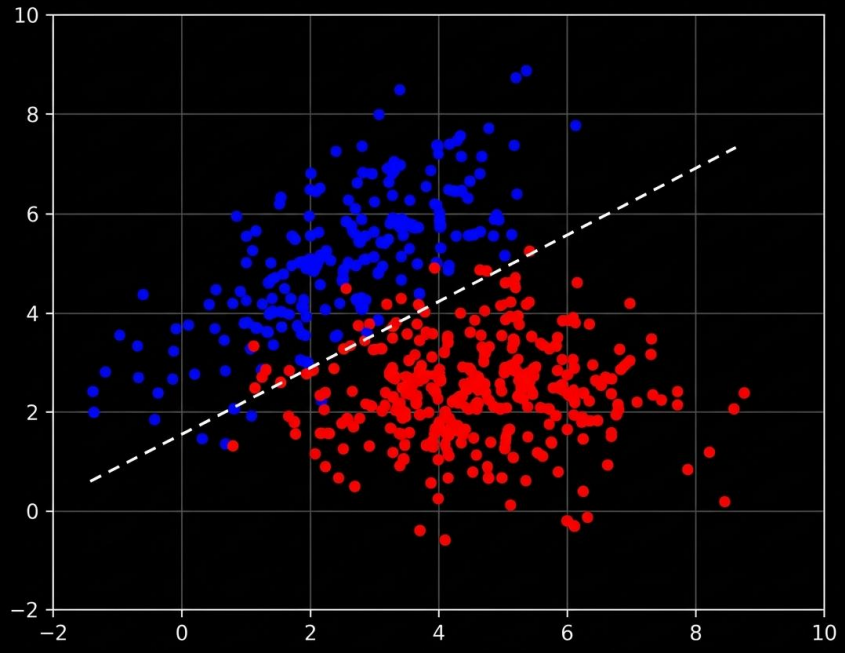
Examples:

- Predicting whether a student will **pass or fail**
- Detecting whether an email is **spam or not spam**
- Recognizing whether a handwritten digit is **1 or 2**

However, mathematically same structure.

Each data can be represented as a point in a space.

A linear classifier learns a decision boundary that separates the classes.

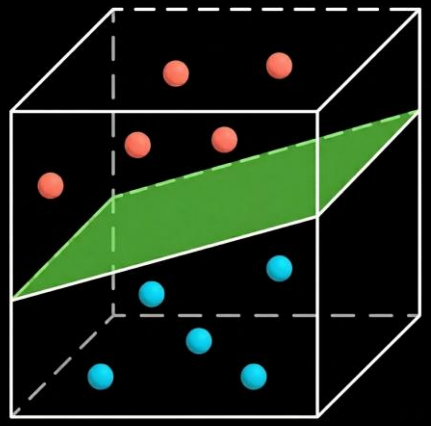


Linear Classification in Higher Dimensions

What happens when the data has more than two features?

Example features:

- Study hours
- Game hours
- Sleep hours
- Attendance rate



✓ Now each sample is a point in a high-dimensional space.

✓ A linear classifier separates the classes using a hyperplane.

Dimension	Decision Boundary
2D	Line
3D	Plane
d-D	Hyperplane

Recap: Geometry in ML

Plane Equation: How to represent a plane in Math?

If $\boldsymbol{n}$ is a normal vector to a plane, then the dot product between $\boldsymbol{n}$ and any vector lying in the plane is zero (i.e., $n \perp plane$).

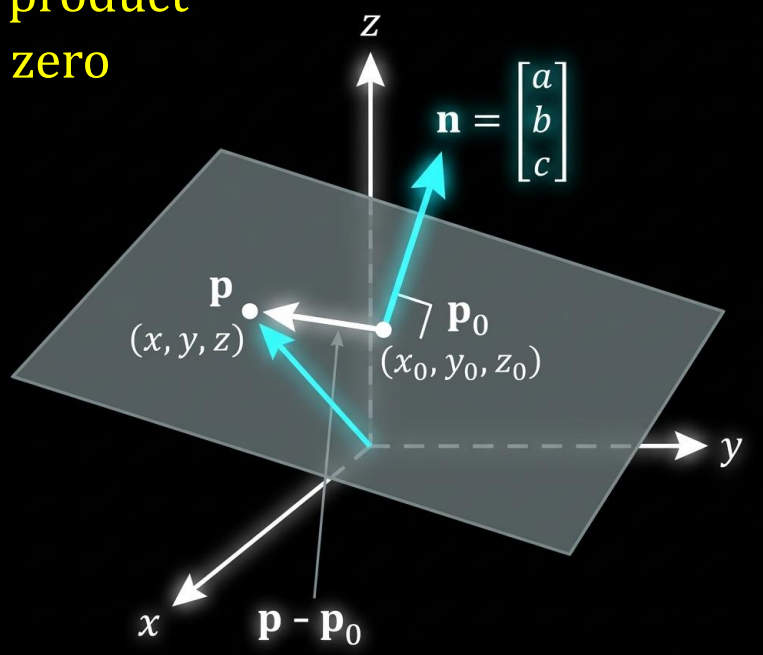
There are many points $p = (x, y, z)$ in a plane.

Let one specific point in the plane be

$$p_0 = (x_0, y_0, z_0)$$

$p - p_0$ is a vector in the plane.

then $n \cdot (p - p_0) = 0$



$$(a, b, c) \cdot (x - x_0, y - y_0, z - z_0) = 0$$

$$a(x - x_0) + b(y - y_0) + c(z - z_0) = 0$$

$$ax + by + cz - (ax_0 + by_0 + cz_0) = 0$$

Let $d = -(ax_0 + by_0 + cz_0)$

$$\boldsymbol{ax + by + cz + d = 0}$$

Generalized Space Equation

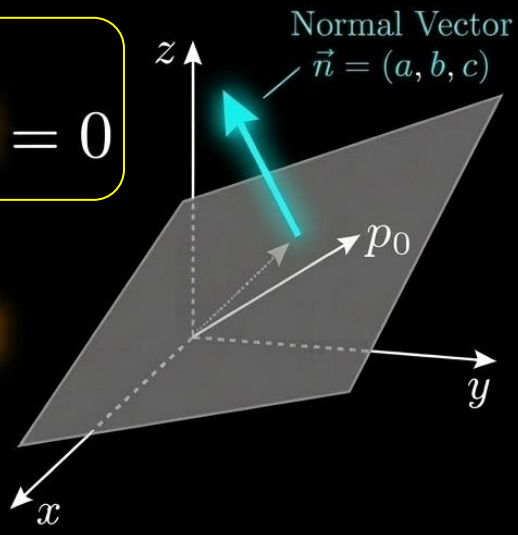
Plane Equation in ML

General form

$$ax + by + cz + d = 0$$

Standard form

$$ax + by + cz = d$$



a, b, c: Normal vector components

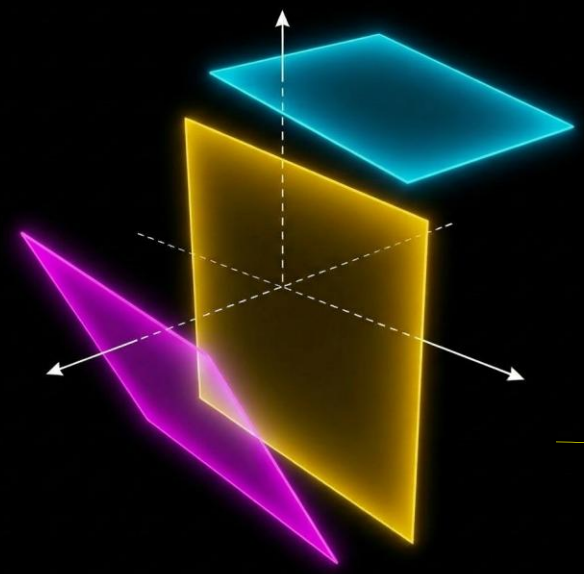
d: Distance parameter

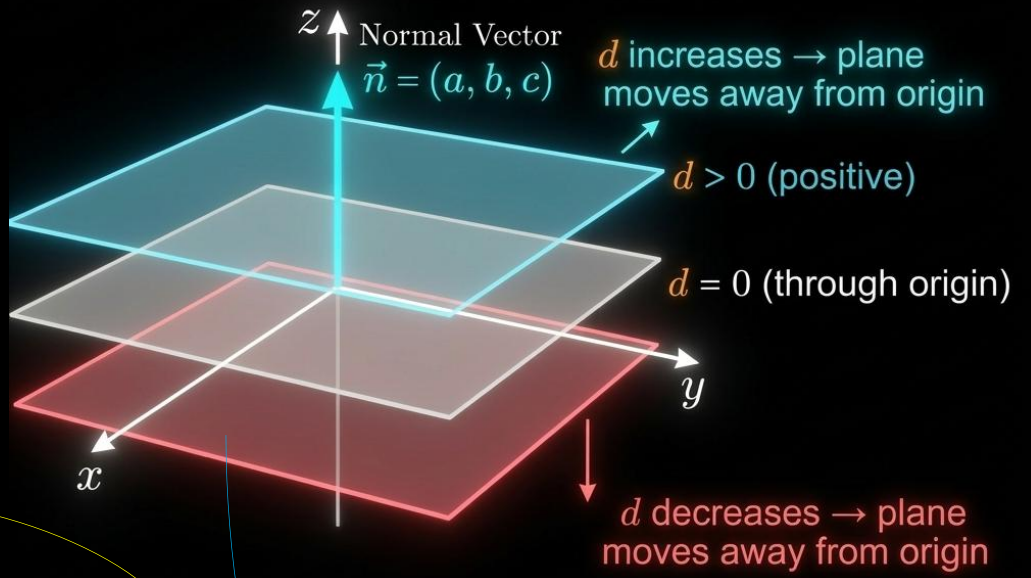
$\vec{n} \cdot (\vec{p} - \vec{p}_0) = 0$

Generalization

Dimension	Equation	Decision Boundary
1D	$ax + b = 0$	Point
2D	$ax + by + c = 0$	Line
3D	$ax + by + cz + d = 0$	Plane
nD	$w^T x + b = 0$	Hyperplane

Moving Decision Boundary





$$ax + by + cz + d = 0$$

In 3D space

If we finde a good shape of plan and → Find values of a, b, c

moving the plane accordingly, → Find values of d

we can do classification!

In nD space, find w^T and b!

Recap: Logistic Regression

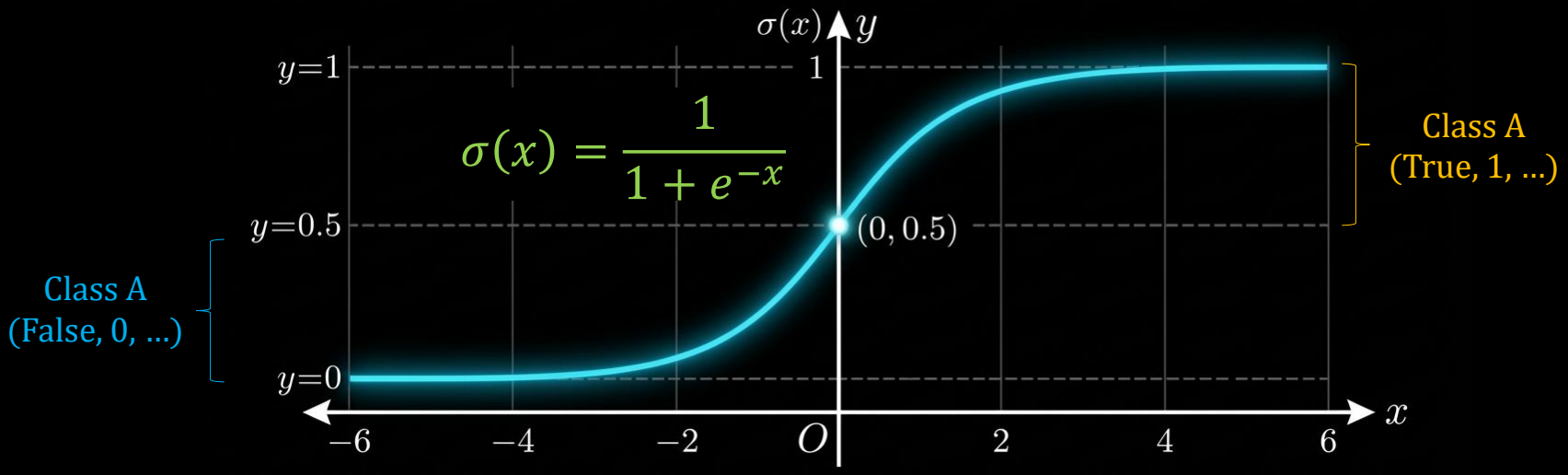
Logistic Regression

Logistic Regression: a linear classifier

$$\sigma(w^T x + b) \quad \begin{cases} p(y=1|x) = \sigma(w^T x + b) \\ p(y=0|x) = 1 - p(y=1|x) \end{cases}$$

Utilize sigmoid function instead of

checking greater or less than zero



Why do we use sigmoid?

Category	Generic Linear Classifier	Logistic Regression
Expression	$w^T x + b$	$\sigma(w^T x + b)$
Predict 1	output > 0	output > 0.5
Predict 0	output < 0	output < 0.5
On the decision boundary	output = 0	output = 0.5

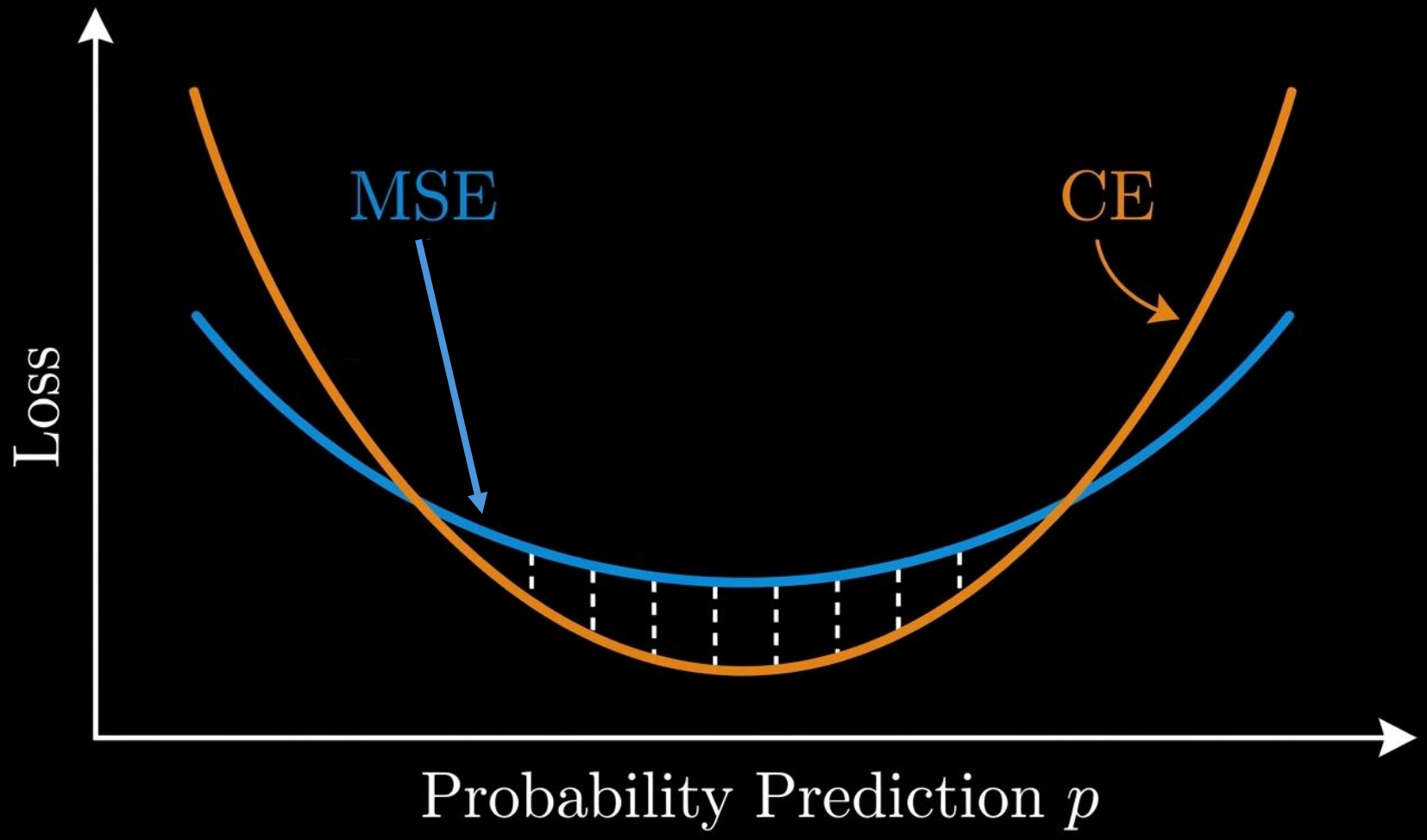
- ✓ Loss is convex: Opt. Min exists + can find optimal point
- ✓ Correct loss can be used w/ probability theory (MLE)

Let $\hat{y} = p(y = 1|x)$

BCE Loss

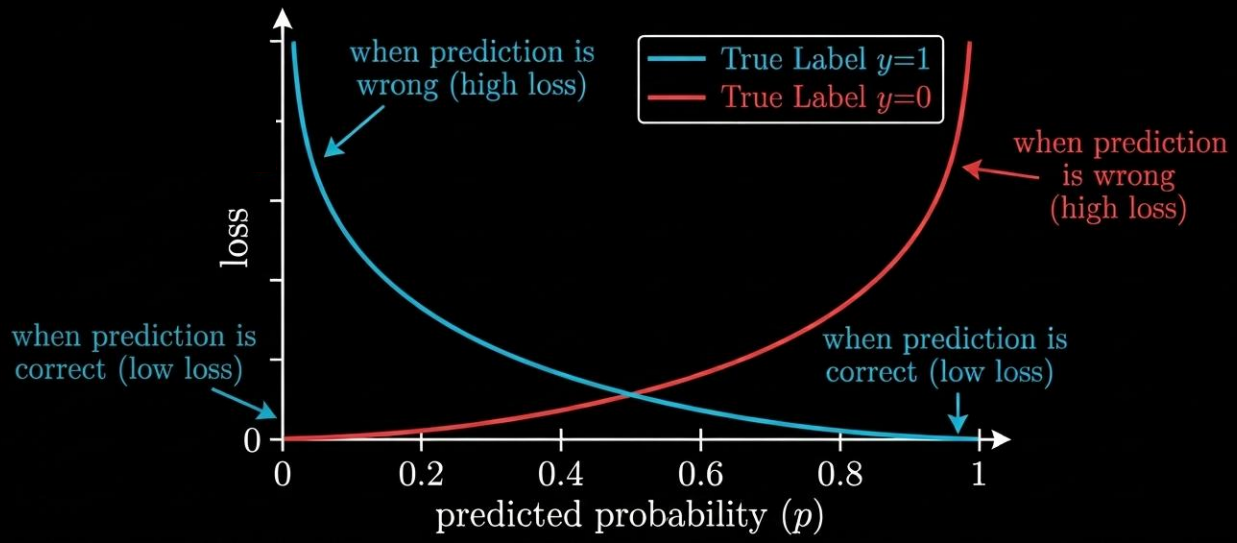
$$L = -\sum_{i=1}^{N} (y_i \log \hat{y}_i + (1 - y) \log(1 - \hat{y}_i))$$

Why Cross Entropy Loss is proper than MSE?



Detail on BCE

$$BCE = -[y \log(p) + (1 - y) \log(1 - p)]$$



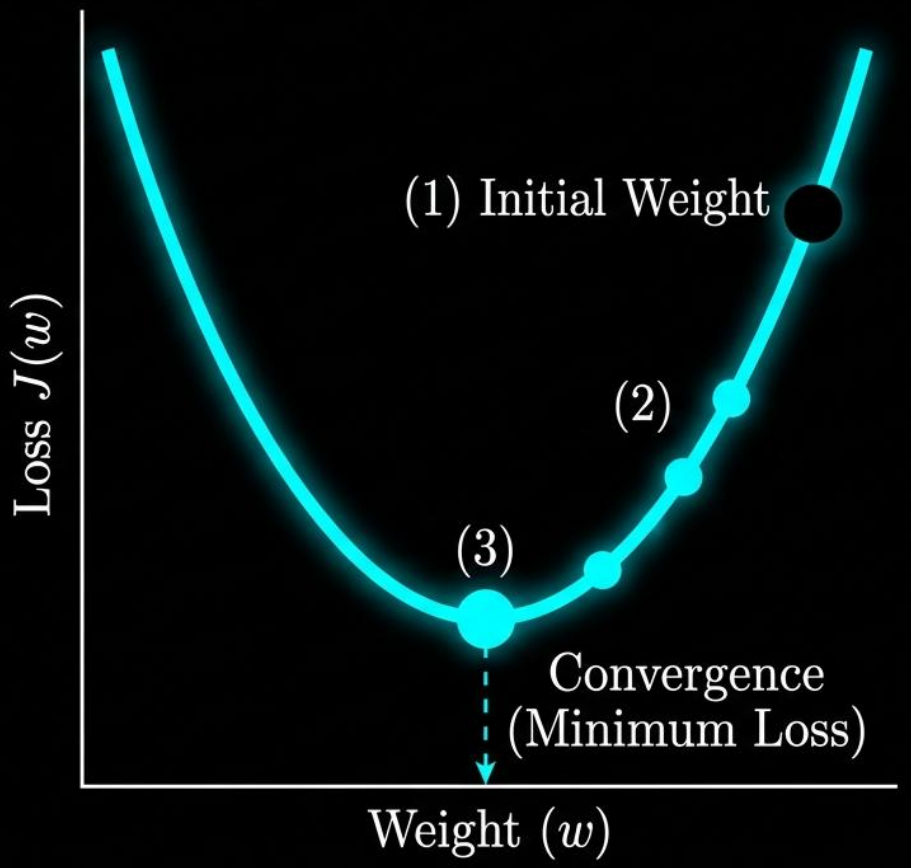
- ✓ Logistic regression outputs probabilities.
- ✓ (Binary) Cross entropy compares probability distributions.
- ✓ Provides stronger gradients → faster learning.
- ✓ Penalizes confident wrong predictions more.
- ✓ MSE assumes regression (Gaussian errors) → mismatch.

Loss Optimization: Find Optimal Minimum Loss

while *not converged*:

$$w \leftarrow w - \eta \frac{\partial L}{\partial w}$$

$$b \leftarrow b - \eta \frac{\partial L}{\partial b}$$



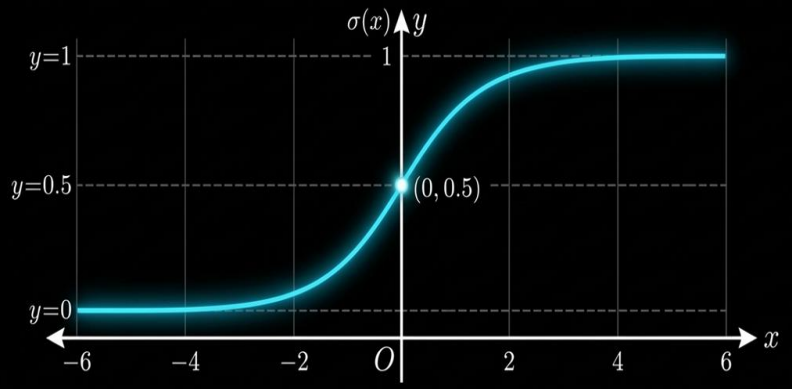
Problem in Logistic Regression

- ✓ If target is 1,

  ML pushes prediction toward predict 1

- ✓ If target is 0,

  ML pushes prediction toward predict 0



Sigmoid is only 1/0 when input is $+\infty/-\infty$

If we want to predict perfectly w/ prob. 100%

$$z \to +\infty/-\infty$$

Therefore, $z = w^T x + b \to +\infty/-\infty$, where x is given

$$|w| \to \infty$$

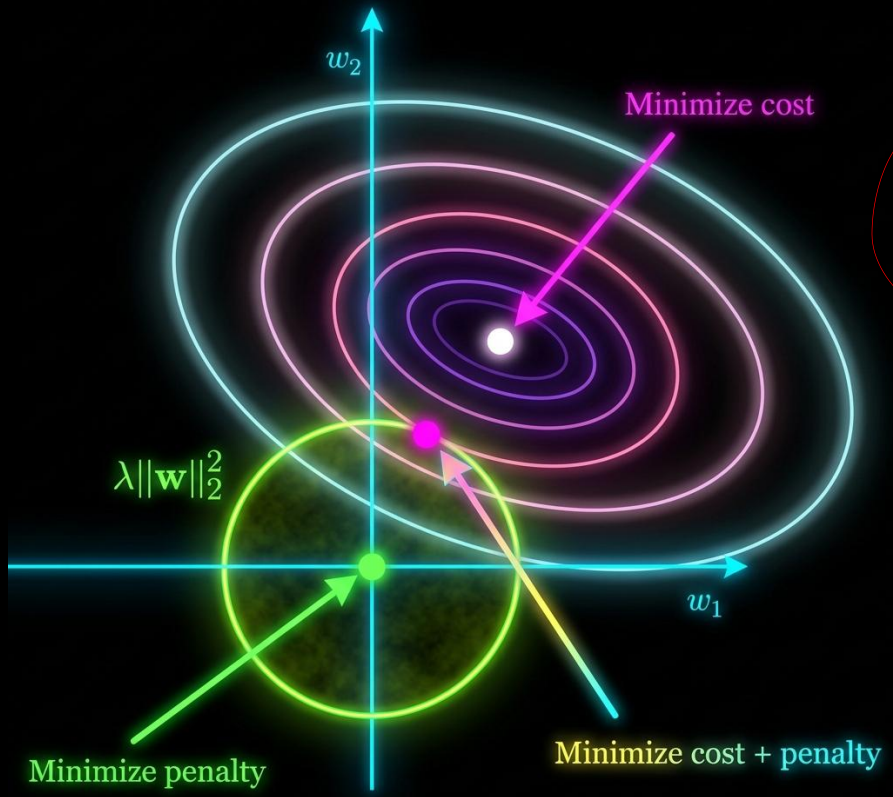
Decision boundary is still same

But probability (output) is extremely steep!

Instability + Practical Overfitting Problem

Very sensitive to noise, outlier, and small changes

Regularization: Penalizing Loss

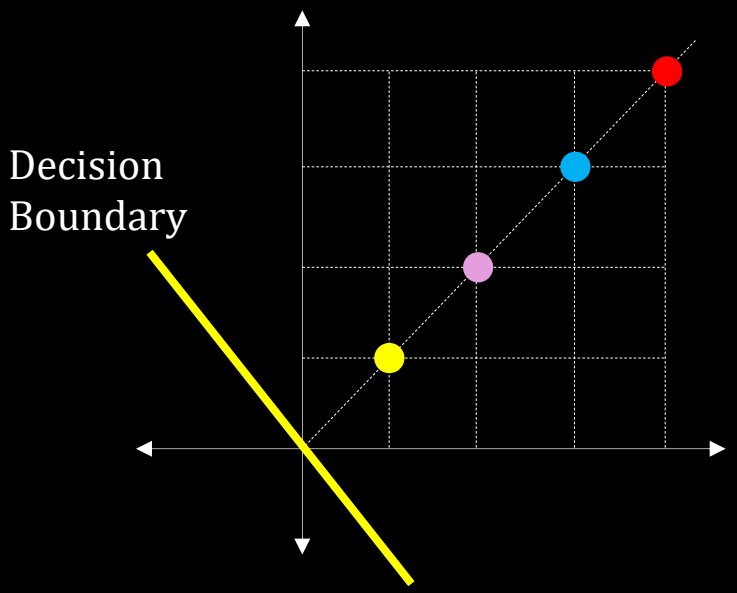


$$L_{reg} = L + \lambda\|w\|^2$$

This is the BASIC form of SVM!

Keeps weights small,
prevents explosion,
and encourages simple models
that generalize better.

Prediction Confidence



Moving farther from the decision boundary
increases confidence,
but the probability quickly saturates near 1.

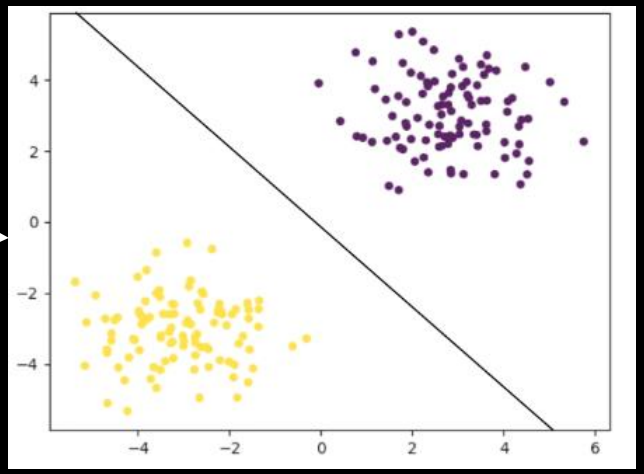
Logistic regression does NOT
explicitly maximize the margin.

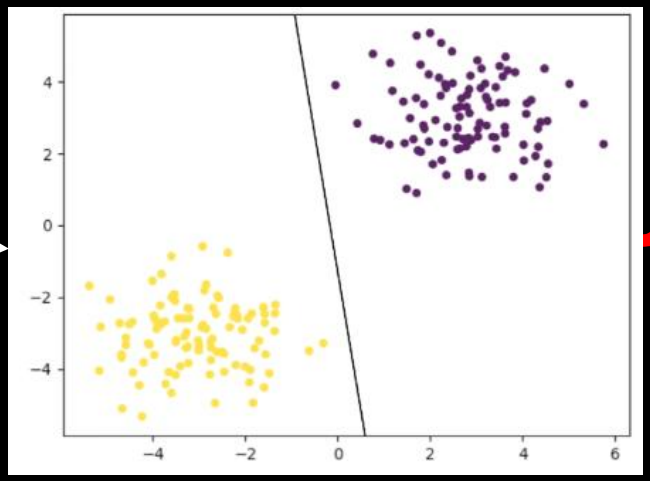
⇨ Why not? The Motivation of SVM!

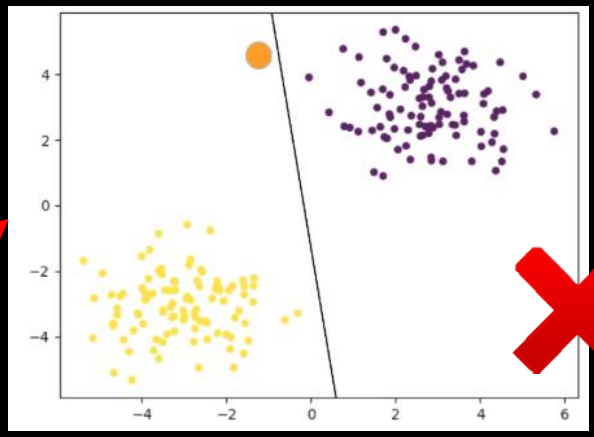
Data Point Movement	Logit $w_1x_1 + w_2x_2$	Logit Increment $z = w^Tx$	Probability from Sigmoid $p(y = 1\|x)$	Probability Increment (delta)
(1,1) → (2,2)	2 → 4	+2	0.8808 → 0.9820	**+10.1%**
(2,2) → (3,3)	4 → 6	+2	0.9820 → 0.9975	**+1.55%**
(3,3) → (4,4)	6 → 8	+2	0.9975 → 0.9996	**+0.22%**

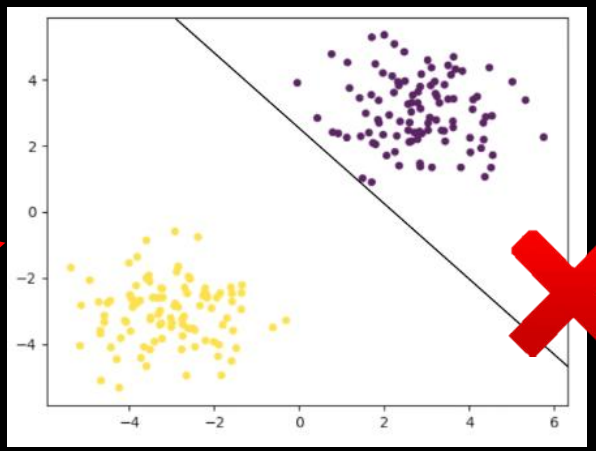
Good or Bad Boundaries w.r.t "Confidence"?

Perfect Prediction with 100%







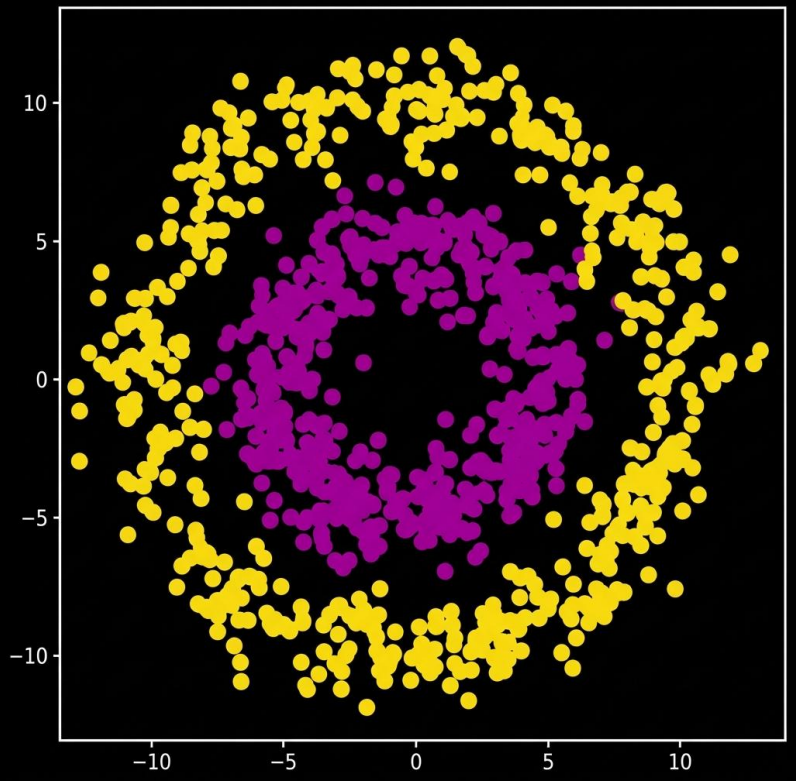


Far away == more "confident"

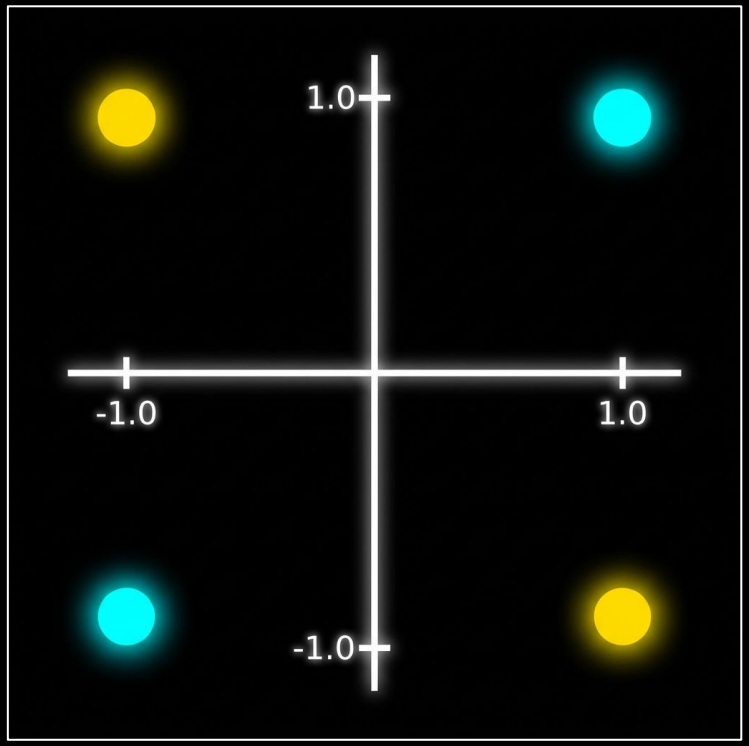
Far away from all training points as possible → SVM philosophy!

Non-linear Problem

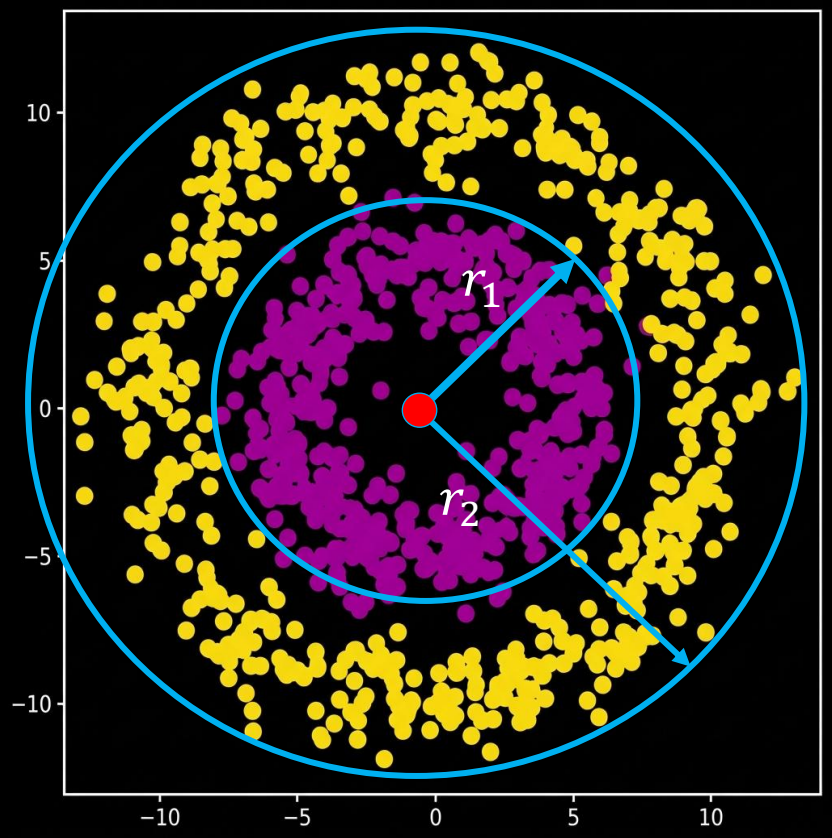
Donut Problem



XOR Problem



Solution to Donut (1/2)



Let Inner radius: $r_1 = 5$

Outer radius: $r_1 = 10$

Given $X = (x_1, x_2)$

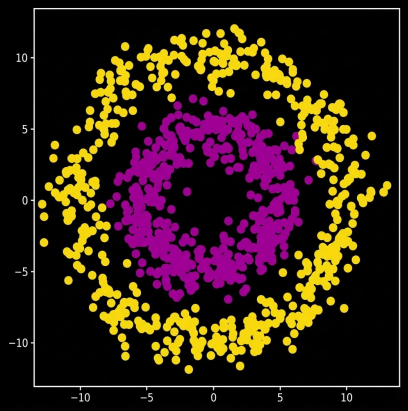
Decision Boundary

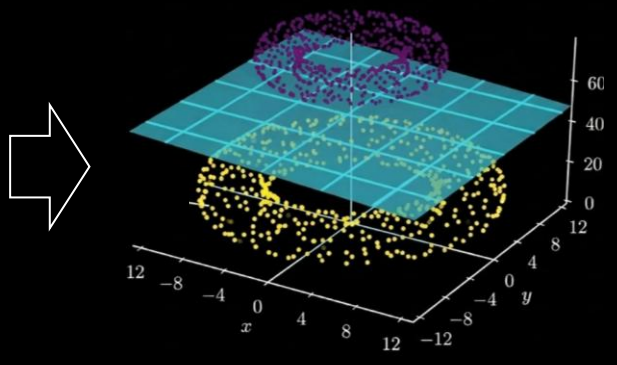
$$r = \sqrt{x_1^2 + x_2^2}$$

```
if  r > 0:
    Predict "Yellow"
else:
    Predict "Purple"
```

Solution to Donut (2/2)

We can use $r = \sqrt{x_1^2 + x_2^2}$





It is "over-engineered".
Can we make it simpler?

$$x_3 = x_1^2 + x_2^2$$

Can we make ML do it (more simpler)?

$$x_3 = x_1^2$$
$$x_4 = x_2^2$$

Add new features (2D → 4D)

$$x_3 + x_4 + 7.5^2 = 0$$

$$W = (w_1, w_2, w_3, w_4)$$

$$X = (x_1, x_2, x_3, x_4)$$

Assume the circle is centered at (p, q)

$$r^2 = (x_1 - p)^2 + (x_1 - q)^2$$

$$r^2 = x_1^2 - 2px_1 + p^2 + x_2^2 - 2qx_2 + q^2$$

Squared & Linear terms appear!

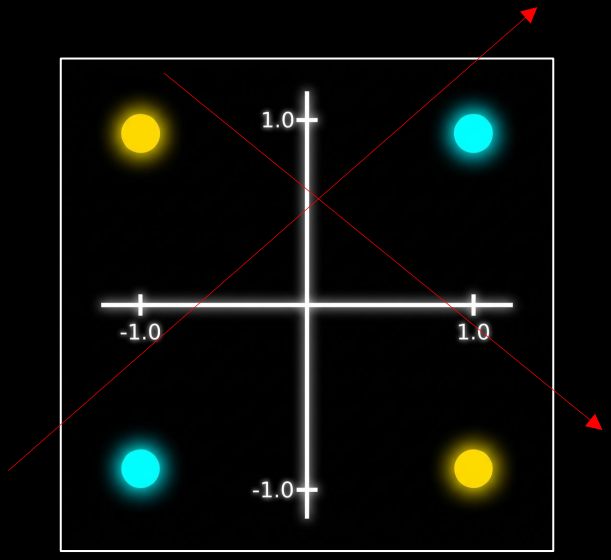
$$r^2 = -2px_1 - 2qx_2 + x_3 + x_4 + p^2 + q^2$$

Polynomial Expansion!

$$w_1x_1 + w_2x_2 + w_3x_3 + w_4x_4 + b = 0$$

Recap: $W^T X + b = 0$

Solution to XOR (1/2)



XOR Operations

$0 \oplus 0 = 0$

$0 \oplus 1 = 1$

$1 \oplus 0 = 1$

$1 \oplus 1 = 1$

Input: $X = (x_1, x_2)$

Params: $W = (w_1, w_2), b$

Decision Boundary:

$w_1 x_1 + w_2 x_2 + b = 0$

$W^T X + b = 0$

No way to find a perfect decision boundary!

How about add new features?

x_1^2 and x_2^2

Not Good:

$1^2 = 1$

$(-1)^2 = 1$

} All added features must have same value.

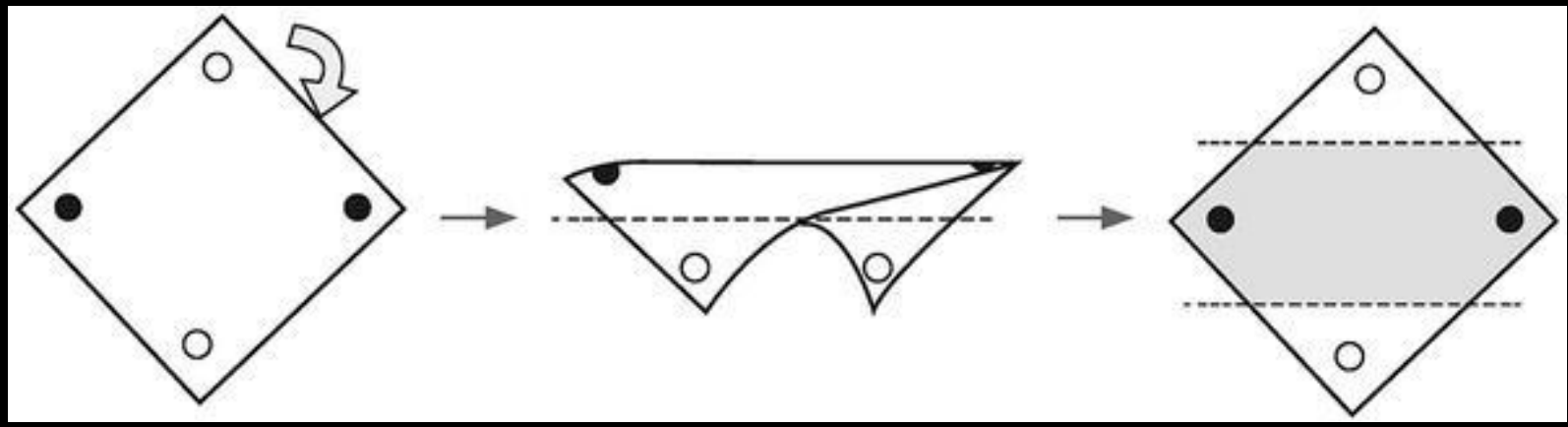
How about $x_1 x_2$?

Purple: $1 \times 1 = -1 \times -1 = 1$

Yellow: $1 \times -1 = -1 \times 1 = -1$

```
if  x1x2 > 0:
    Predict "Purple"
else:
    Predict "Yellow"
```

Solution to XOR (2/2)



No need to simplify x_1x_2 any more!

$$x_3 = x_1x_2$$

$$W = (w_1, w_2, w_3)$$

$$X = (x_1, x_2, x_3)$$

$$w_1x_1 + w_2x_2 + w_3x_3 + b = 0$$

Decision boundary is:

$W = (0,0,1)$

$b = 0$

} Perfectly predict with any input (x_1, x_2)

Recap: $W^T X + b = 0$

Feature Expansion Problem (1/2)

It is normal to include all possible terms.

$$X = (x_1, x_2) \rightarrow (x_1, x_2, x_1^2, x_2^2, x_1 x_2)$$

- ✓ Achieves "Automatic feature generation" in non-linear problem!
- ✓ Not try each feature manually!
- ✓ Ex. degree-2 expansion automatically generates these terms!

We can easily say that
"Hey, SVM package! Let's apply 'poly' feature expansion"

Are there any problems?

Feature Expansion Problem!

What if we have 3D (x_1, x_2, x_3) input?

$$X = (x_1, x_2, x_3) \rightarrow (x_1, x_2, x_3, x_1^2, x_2^2, x_3^2, x_1 x_2, x_1 x_3, x_2 x_3)$$

What if we have 4D (x_1, x_2, x_3, x_4) input?

$$X = (x_1, x_2, x_3, x_4)$$
$$\rightarrow (x_1, x_2, x_3, x_4, x_1^2, x_2^2, x_3^2, x_4^2, x_1 x_2, x_1 x_3, x_1 x_4, x_2 x_3, x_2 x_4, x_3 x_3)$$
$$\vdots$$

What if we have nD $(x_1, x_2, x_3, \cdots, x_n)$ input?

Exponential Increase

Feature Expansion Problem (2/2)

What if we increase polynomials?

If we have 2D (x_1, x_2) input?

Degree-1 terms: x_1, x_2

Degree-2 terms: x_1^2, x_2^2, x_1x_2

Degree-3 terms: $x_1^3, x_2^3, x_1^2x_2, x_1x_2^2$

If we have 3D (x_1, x_2, x_3) input?

Degree-1 terms: x_1, x_2, x_3

Degree-2 terms: $x_1^2, x_2^2, x_3^2, x_1x_2, x_1x_3, x_2x_3$

Degree-3 terms: $x_1^3, x_2^3, x_3^3, x_1^2x_2, x_1^2x_3, x_2^2x_1, x_2^2x_3, x_3^2x_1, x_3^2x_2, x_1x_2x_3$

⋮

What if we have nD $(x_1, x_2, x_3, \cdots, x_n)$ input?

Exponential Increase!

SVM has

- the ***capability*** of polynomial.
- beautiful technology, a.k.a. ***"Kernel Trick" (we will learn).***

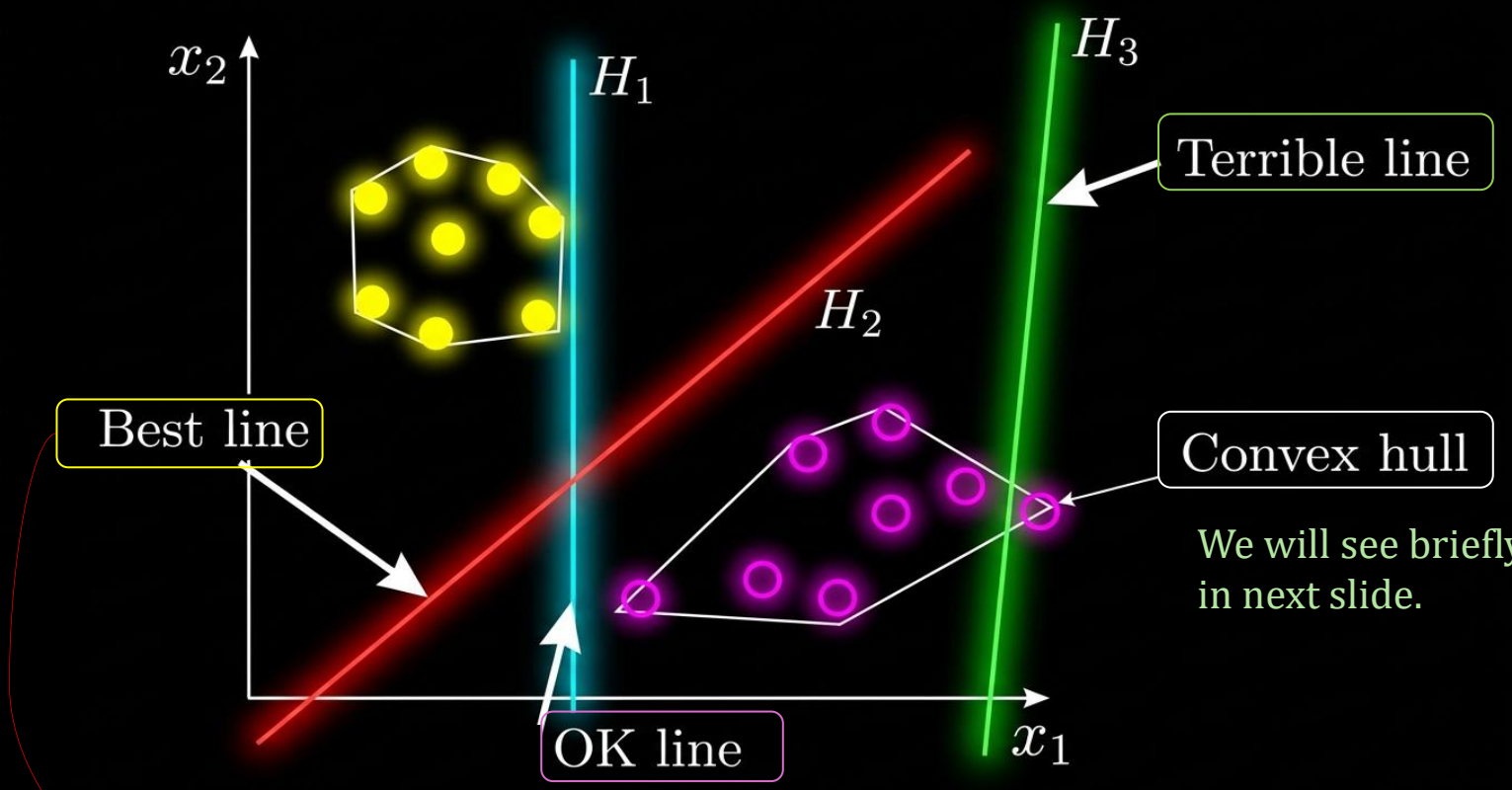
Also, Feature Expansion!

- Captures **nonlinearity**
- Causes **feature explosion**
- Leads to **high computation**
- Risk of **overfitting**

→ Need a **better solution**

Linear SVM

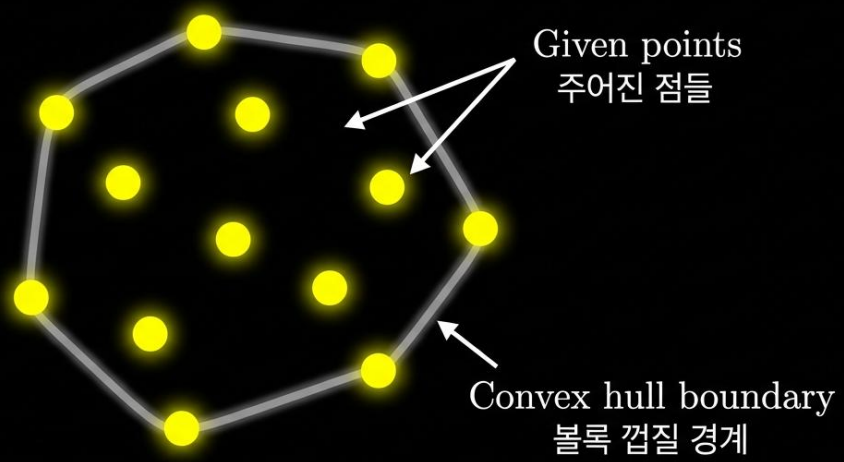
Basic Idea



Best line (decision boundary):
Furthest away form training data points

Convex Hull

The smallest convex shape that contains all given points



특징

- hull 내부의 어떤 두 점을 연결해도 선분이 항상 hull 내부에 존재
- 내부 점들은 hull에 영향을 주지 않고 바깥쪽 경계 점들만 hull을 결정한다.

직관적 이해

점들을 종이에 찍어 놓고 고무줄을 바깥에서 감싸면 고무줄이 만드는 경계가 바로 convex hull

Convex Hulls & Classification in ML:

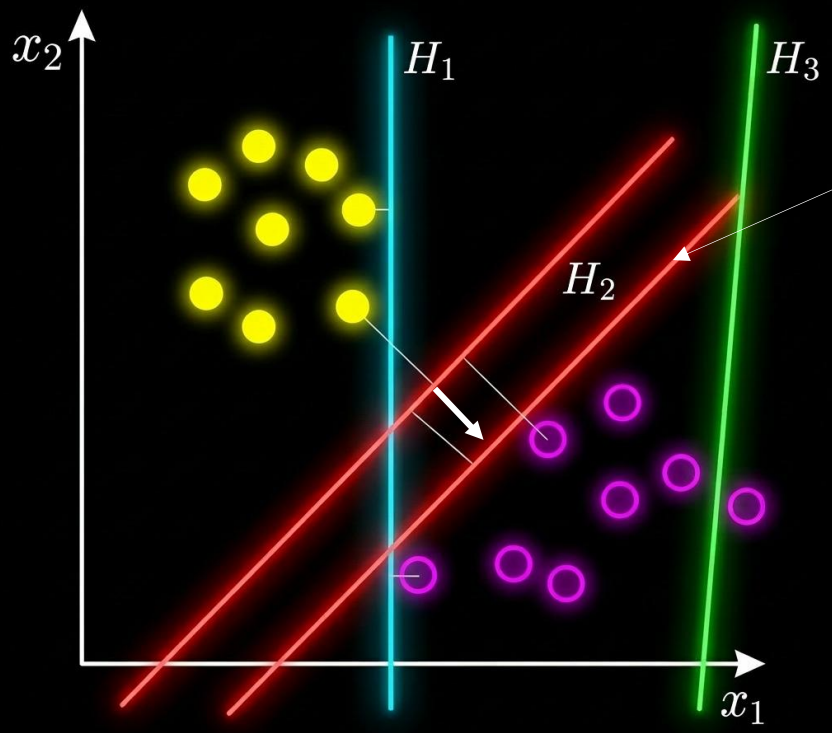
✓ If the convex hulls of two classes do not overlap,

✓ the data is linearly separable.

Support vectors lie on the boundary of convex hulls.

Even if convex hulls overlap, SVM can still separate the data using a soft margin.

SVM Classification



$$minDistance = \min_{i=1,\cdots,N} dist(i)$$

Red line shifted toward purple circles

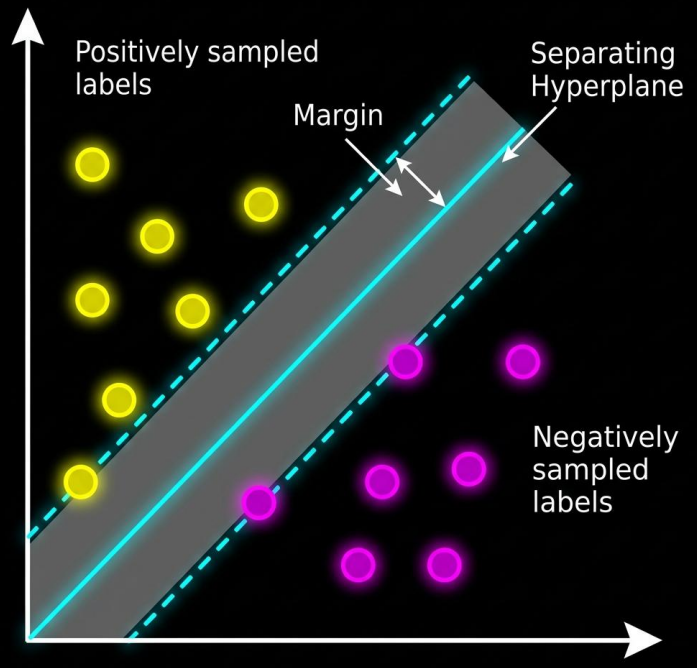
- Increased distance from yellow points
- Decreased distance from purple points

Just want maximize distance from all points

Maximize minimum distance between the line and all points

Objective: $\max_{w} minDistance$

The Concept of Margin



In SVM Theory

Margin: the closest point to the line

SVM is "Maximum Margin Classifier"

Let

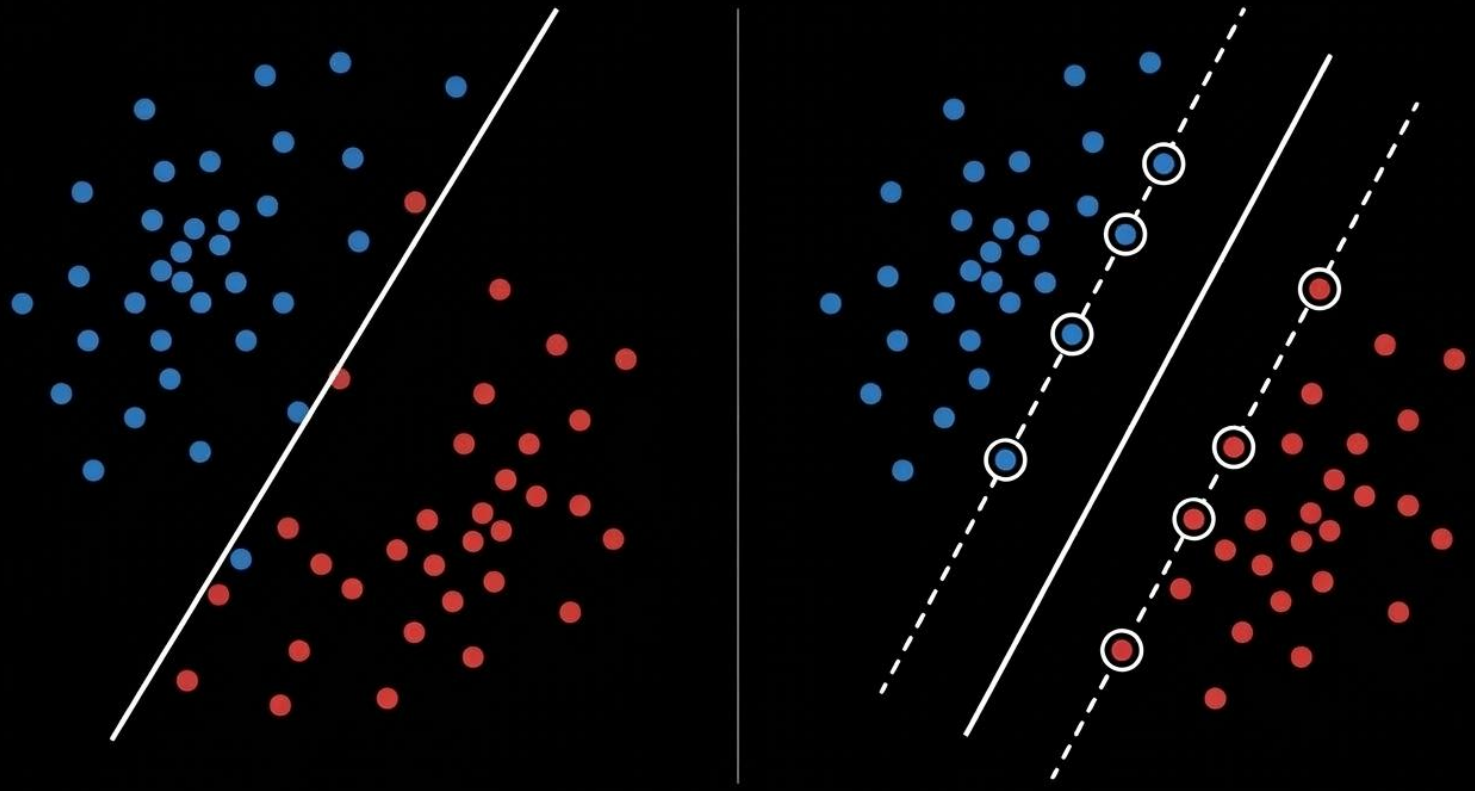
$J = Equation\ of\ Margin$

Goal:

$$\max_{w,b} J$$

- ✓ All that we need to do is finding J
- ✓ Then do some optimization using J

Logistic Regression vs. SVM



Margin

Functional Margin

For an data point, prediction is correct if:

$$w^T x_i + b > 0, \quad y_i = 1$$
$$w^T x_i + b < 0, \quad y_i = -1$$

If sign is reversed, prediction is incorrect.

Combine into 1 equation

$$\underbrace{y_i(w^T x_i + b)}_{} > 0$$

Let this quantity be $\hat{\gamma}_i$

Higher value: more confident

Lower value: less confident

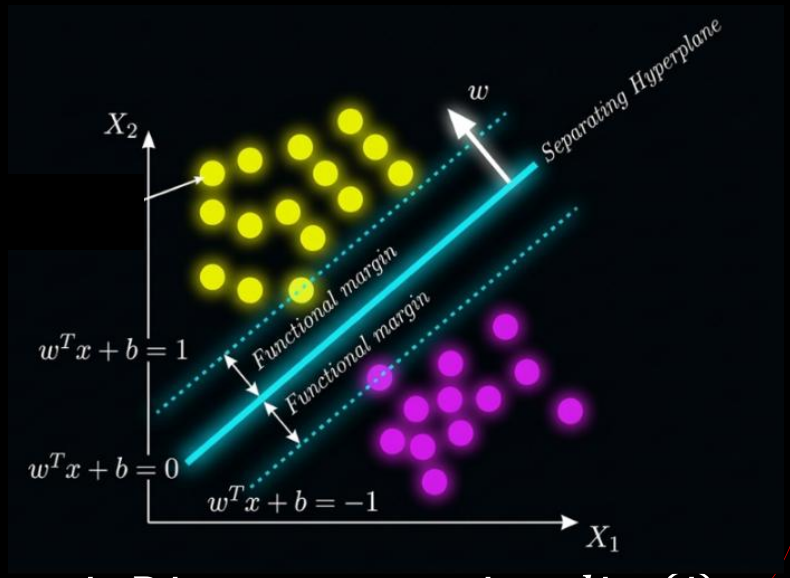
$$\hat{\gamma}_i = y_i(w^T x_i + b)$$

Call this "Functional Margin"

Recap: Prediction Confidence

Problem in Margin

Functional margin of params (w, b) w.r.t.



$$Dataset\ D = \{(x_i, y_i)\}_{i=1}^{N}$$

$$\hat{\gamma}_i = y_i(w^T x_i + b)$$

$$\hat{\gamma} = \min_{i=1,\cdots,N} \hat{\gamma}_i$$

$$minDistance = \min_{i=1,\cdots,N} dist(i)$$

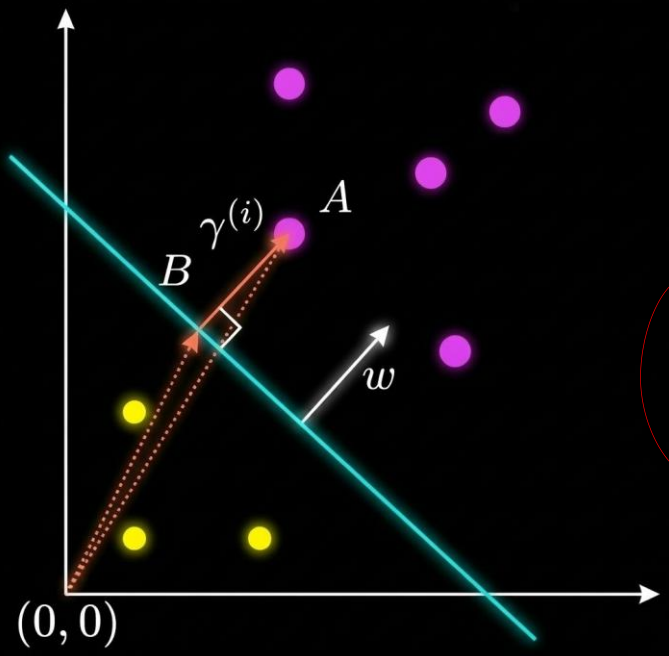
Params (w, b) can be scaled arbitrarily, Functional margin $\hat{\gamma}_i$ do **NOT** represent any distance in space.

Just scaled value of $w^T x + b$

How can we know the distance between line and a point?

⇨ **Geometric Margin!**

Geometric Margin (1/4)



Functional margin: $\hat{\gamma}_i = y_i(w^T x_i + b)$

$$\hat{\gamma} = \min_{i=1,\cdots,N} \hat{\gamma}_i$$

Geometric margin: $\gamma_i = \frac{1}{\|w\|}\hat{\gamma}_i$

Geometric Margin is the true distance from a point to the hyperplane.

We need to Find: γ_i

Fact 1: $\overrightarrow{BA}$ is normal to the line

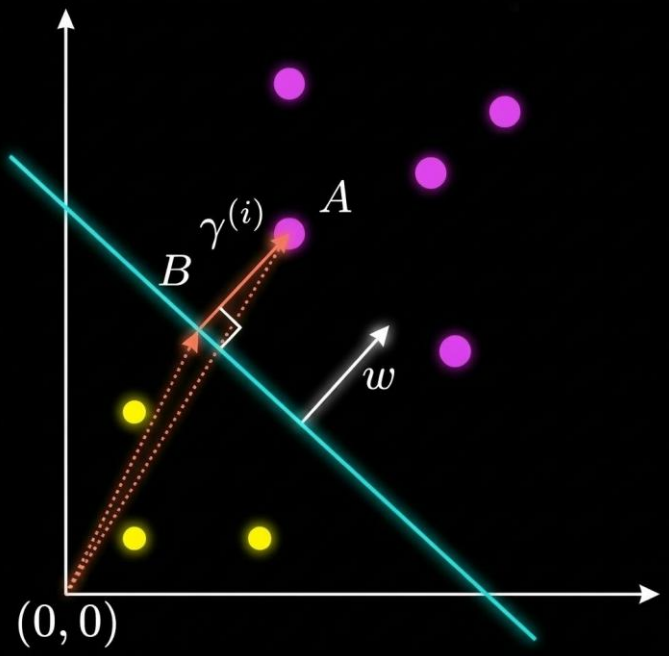
Fact 2: w is also normal to the line

Fact 3: $\frac{w}{\|w\|}$ is a unit vector parallel to $\overrightarrow{BA}$

Therefore,

$$\overrightarrow{BA} = \frac{w}{\|w\|}\hat{\gamma}$$

Geometric Margin (2/4)



$$\overrightarrow{OA} = x_i$$

$$\overrightarrow{OB} = \overrightarrow{OA} - \overrightarrow{BA} = x_i - \frac{w}{\|w\|}\hat{\gamma}_i$$

$\overrightarrow{OB}$ is on the line,
it MUST satisfiy the equation of the line.

$$w^T\overrightarrow{OB} + b = 0$$

$$w^T\left(x_i - \frac{w}{\|w\|}\hat{\gamma}_i\right) + b = 0$$

Can we isolate γ_i?

Geometric Margin (3/4)

$$w^T\left(x_i - \frac{w}{\|w\|}\gamma_i\right) + b = 0$$

We can isolate γ_i

$$w^T x_i - \frac{w^T w}{\|w\|}\gamma_i + b = 0$$

$$w^T w = \sum_i w_i^2$$

$$\|w\| = \sqrt{\sum_i w_i^2}$$

$$w^T w = \|w\|^2$$

$$w^T x_i - \|w\|\gamma_i + b = 0$$

$$\gamma_i = \left(\frac{w}{\|w\|}\right)^T x_i + \frac{b}{\|w\|}$$

$$y_i(w^T x_i + b) > 0, \qquad y_i = 1$$

If sign is reversed, prediction is incorrect.

$$y_i(w^T x_i + b) < 0, \qquad y_i = -1$$

Final Form of Margin:

$$\gamma_i = y_i\left(\left(\frac{w}{\|w\|}\right)^T x_i + \frac{b}{\|w\|}\right)$$

Geometric Margin (4/4)

Geometric margin represents the true distance to the decision boundary and is invariant to scaling of (w, b).

This always holds because geometric margin is an actual distance.

$$\gamma_i = y_i \left(\left(\frac{w}{\|w\|} \right)^T x_i + \frac{b}{\|w\|} \right) = \frac{y_i (w^T x_i + b)}{\|w\|} = y_i \frac{\hat{\gamma}_i}{\|w\|}$$

Simple Relationship between Geometric & Functional Margin

(y_i is 1 or -1, which is sign)

$$\gamma_i = \frac{1}{\|w\|} \hat{\gamma}_i$$

Therefore, maximizing geometric margin is equivalent to minimizing $\|w\|$.

SVM Objective

Derivation of Objective in Linear SVM

Set of Constraints:

$$w^T x_i + b > 0, \quad y_i = 1$$

$$w^T x_i + b < 0, \quad y_i = -1$$

} If sign is reversed, prediction is incorrect.

Combine Constraints:

$$y_i(w^T x_i + b) > 0$$

⇒

Enforce Stronger Constraint:

$$y_i(w^T x_i + b) > \hat{\gamma}_i \,, \forall i = 1,2,\cdots,N$$

$$\hat{\gamma} = \min_i y_i(w^T x_i + b)$$

Objective of Linear SVM

$$\max_{\gamma,w,b} \gamma \quad \text{, where } \gamma = \frac{\hat{\gamma}}{\|w\|}$$

⇒

$$\max_{w,b} \frac{\hat{\gamma}}{\|w\|}$$

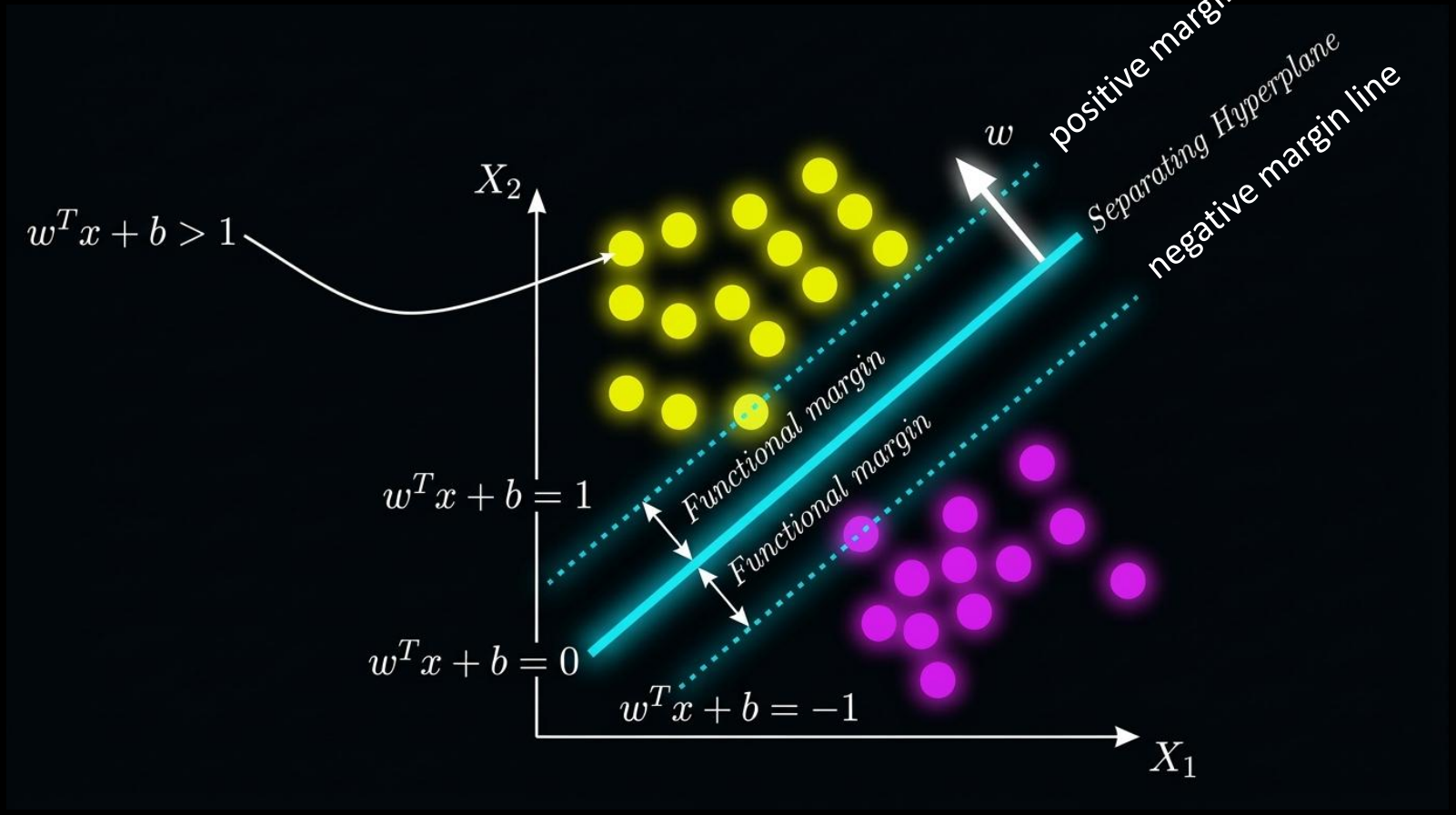
subject to

$$y_i(w^T x_i + b) > \hat{\gamma}_i$$

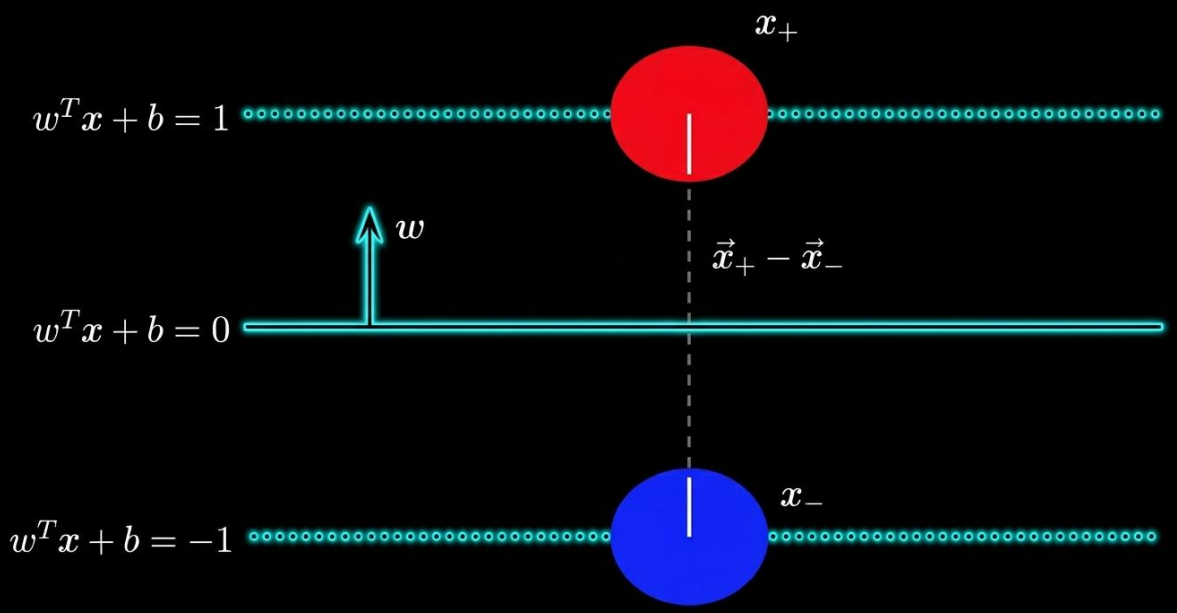
$$\forall i = 1,2,\cdots,N$$

Begin with Standard Margin (size of 1)

Functional Margin is arbitrary, let's fix it to 1



Find SVM Objective (1/2)



$$(x_+ - x_-) \parallel w$$

$$w^T x_+ + b = 1$$

$$w^T x_- + b = -1$$

$$\begin{array}{rl} & w^T x_+ + b = 1 \\ - & w^T x_- + b = -1 \\ \hline & w^T (x_+ - x_-) = 2 \end{array}$$

벡터 내적 공식: $a^T b = \|a\|\|b\| \cos\theta$

$$\|w\|\|x_+ - x_-\| \cos\theta = 2$$

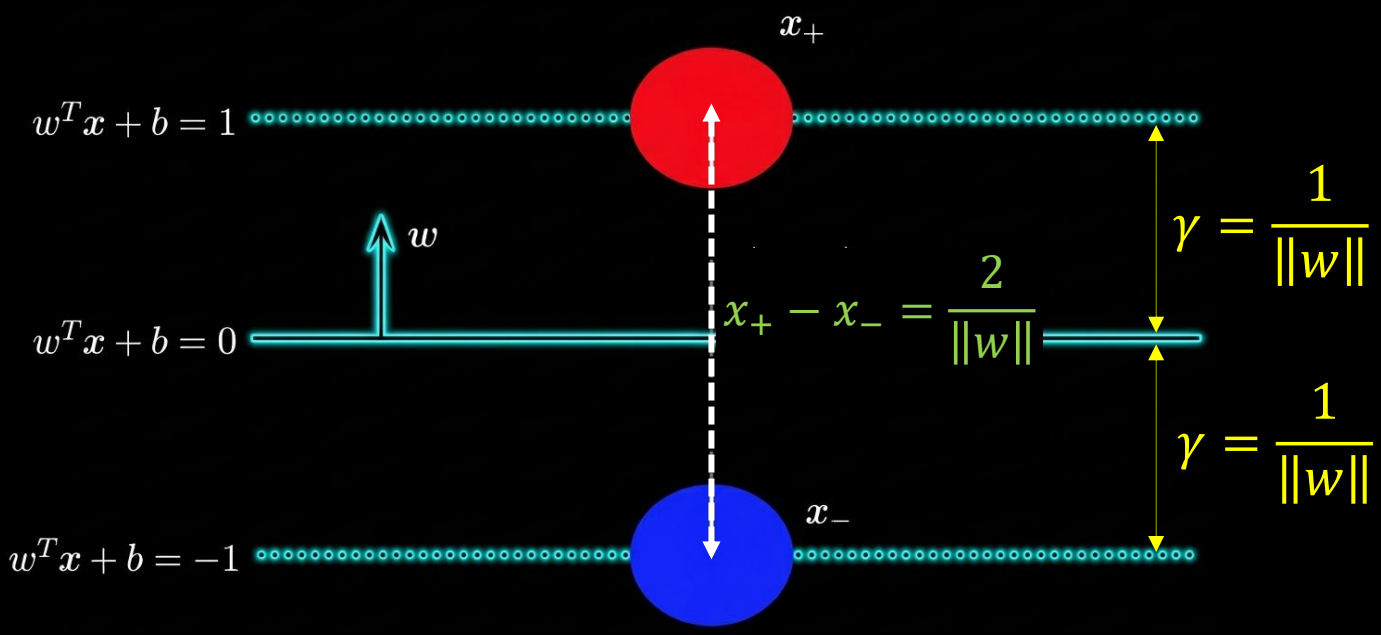
$(x_+ - x_-) \parallel w$ 이므로
$\theta = 90,\ \cos\theta = 1$

$$\|w\|\|x_+ - x_-\| = 2$$

Therefore,

$$\|x_+ - x_-\| = \frac{2}{\|w\|}$$

Find SVM Objective (2/2)



$$\|x_+ - x_-\| = \frac{2}{\|w\|}$$

$$\max_{w,b} \frac{\hat{\gamma}}{\|w\|} \Rightarrow \max_{w,b} \frac{1}{\|w\|}$$

Because we fixed functional margin to 1

$$\gamma = \frac{1}{\|w\|} = \frac{\hat{\gamma}}{\|w\|}$$

subject to

$$y_i(w^T x_i + b) > \hat{\gamma}_i$$

$$\forall i = 1,2,\cdots,N$$

Final Objective of Linear SVM

$$\max_{w,b} \frac{\hat{\gamma}}{\|w\|} \Rightarrow \max_{w,b} \frac{1}{\|w\|}$$

$$\Rightarrow \min_{w,b} \|w\|$$

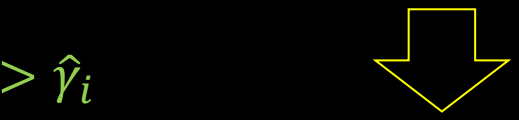
$$\Rightarrow \min_{w,b} \frac{1}{2}\|w\|^2$$

Quadratic Form

$$\|w\|^2 = w_1^2 + w_2^2 + \cdots + w_n^2$$

Optimizing this form is very strange to Computer Science.

Out of ML's Responsibility!



It can be solved by a softwawre package for quadratic programming (Google OR tools, Matlab, Python cvxopt, etc.)

subject to

$$y_i(w^T x_i + b) > \hat{\gamma}_i$$

$$\forall i = 1, 2, \cdots, N$$

Thank you!