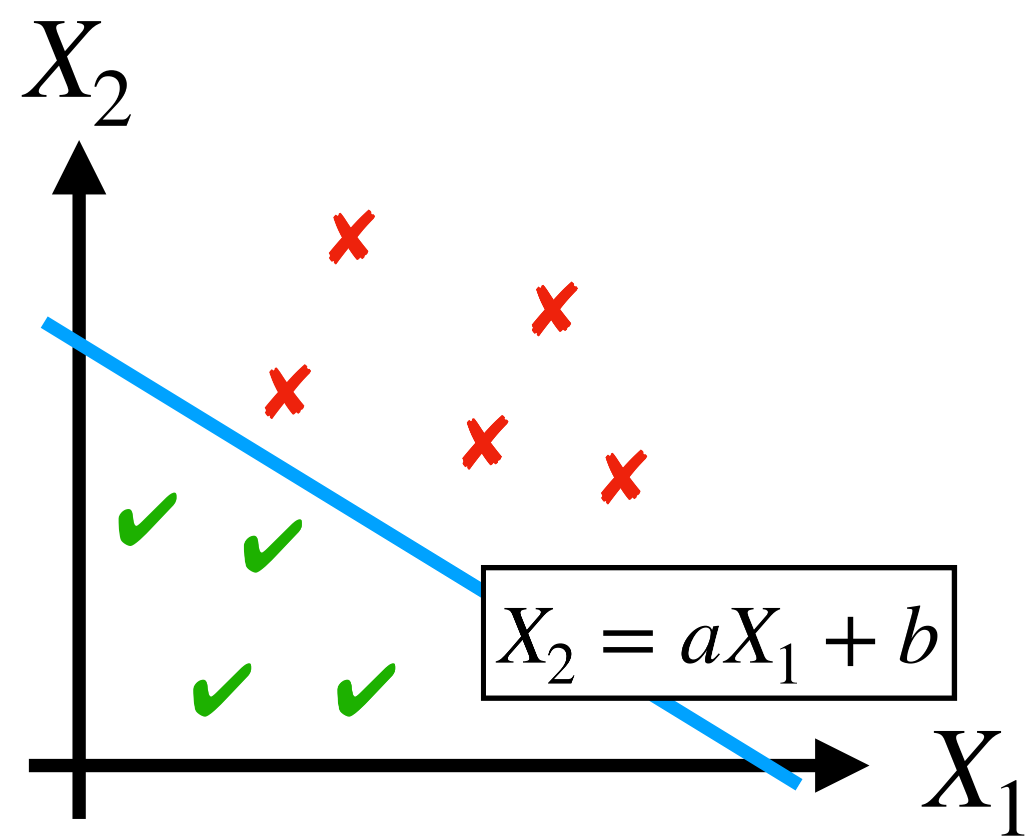
Artificial Intelligence

Taro Sekiyama

National Institute of Informatics (NII)
sekiyama@nii.ac.jp

Support vector machines (SVMs)

- A supervised learning algorithm applicable to both classification and regression problems
- Ex: classification
    - For input (x_1, x_2)
        - If $x_2 \leq ax_1 + b$, the label should be ✔
        - If $x_2 \geq ax_1 + b$, the label should be ✘



Pros and Cons

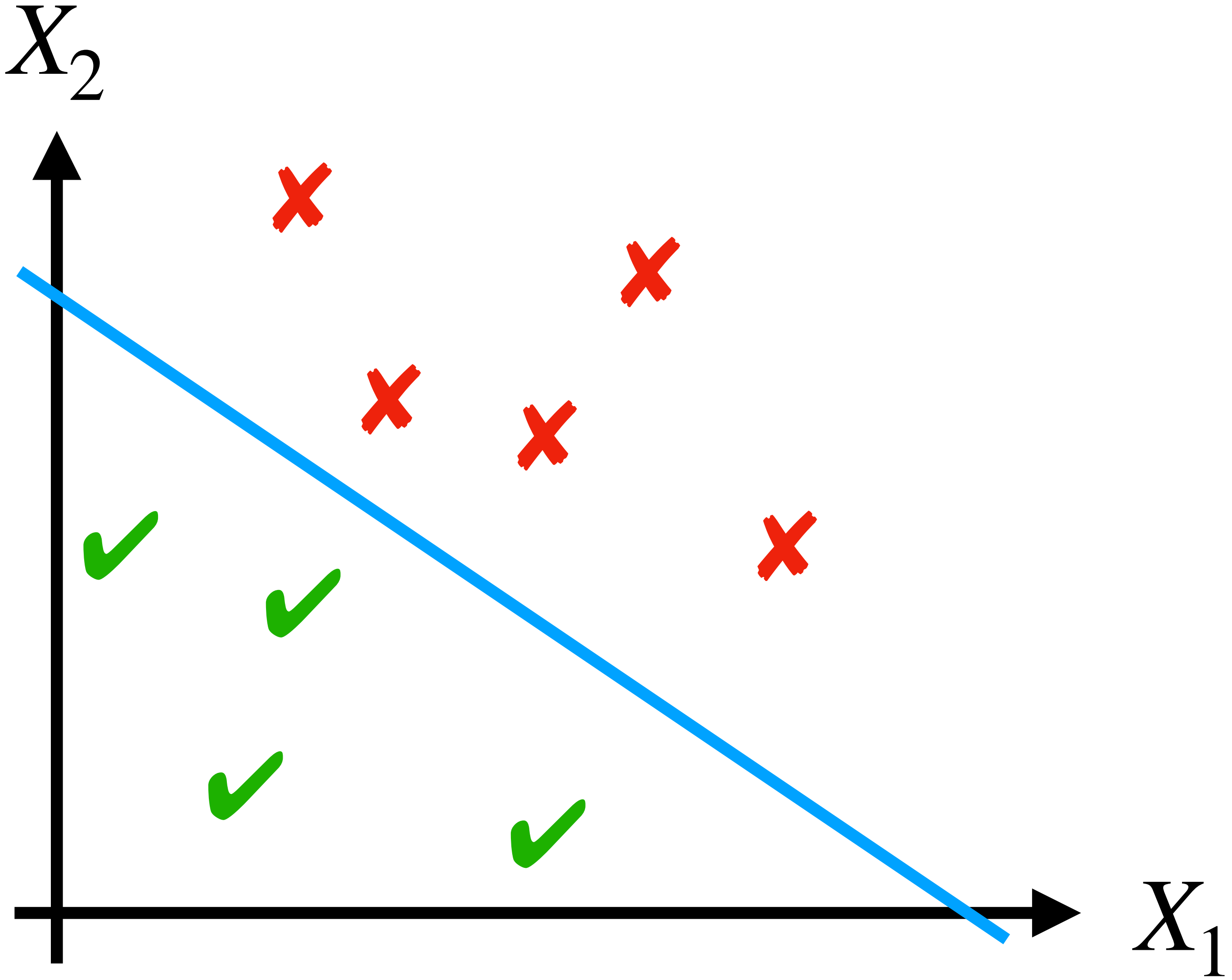
- Advantage
  - Works well for datasets with highly-dimensional features (i.e., the large # of features)
    - Ex: texts (1M~ features) and images ($2^8 \times 2^8 \times 2^8 \times 2^8$ features for RGBA images)
  - Able to handle non-linearly separable datasets
- Disadvantage
  - Computationally hard to handle a huge dataset

SVMs in this lecture

- Supposed to be used as ***binary classifiers***
  - Making it easier to investigate their mathematical aspects
  - Able to be extended to multi-class classifiers and regressors
- Dataset consists of data points (x_i, y_i) for $x_i \in \mathbb{R}^n$ and $y_i \in \{1, -1\}$

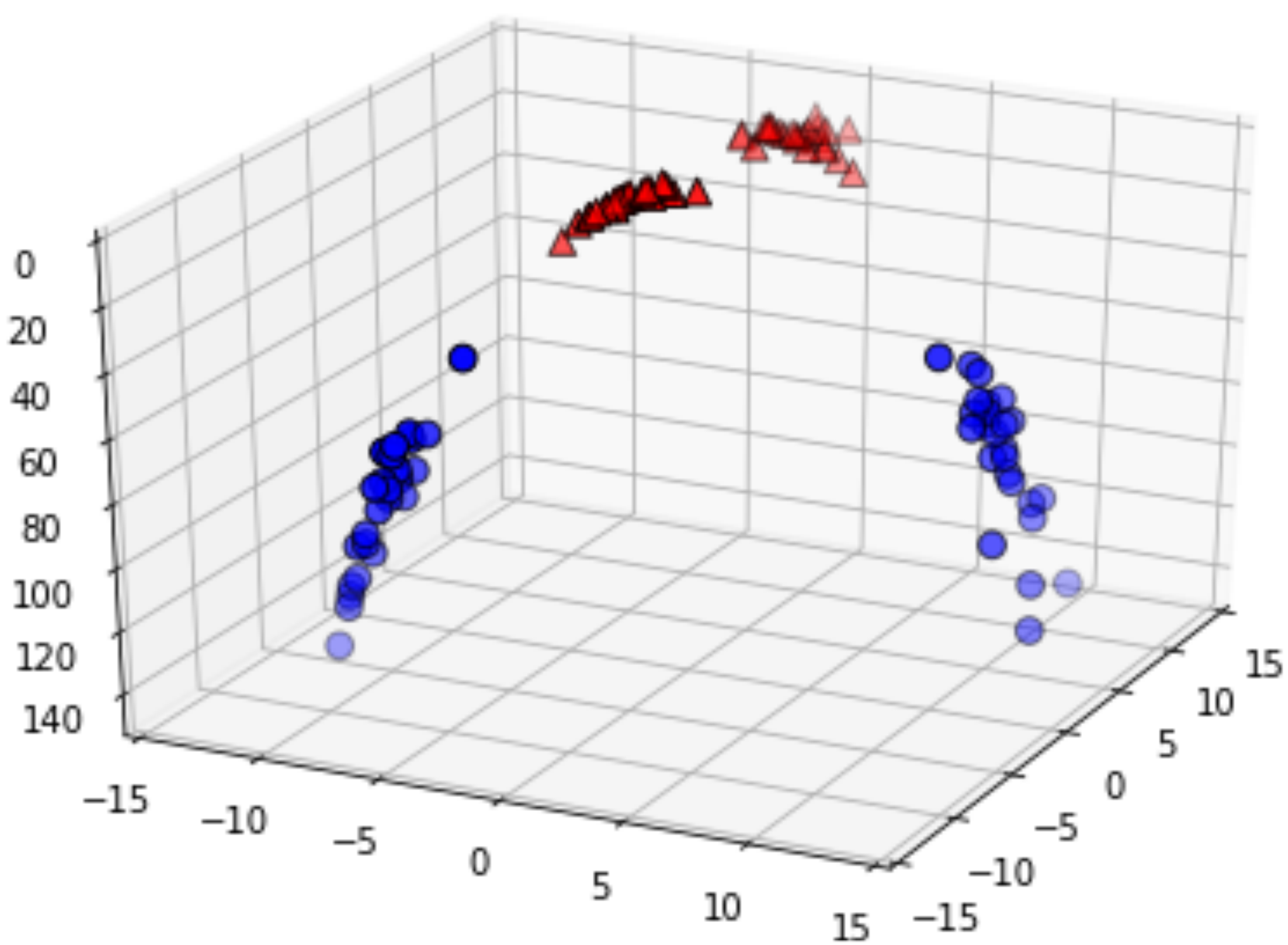
Idea

- Drawing a splitting line for 2-dim space



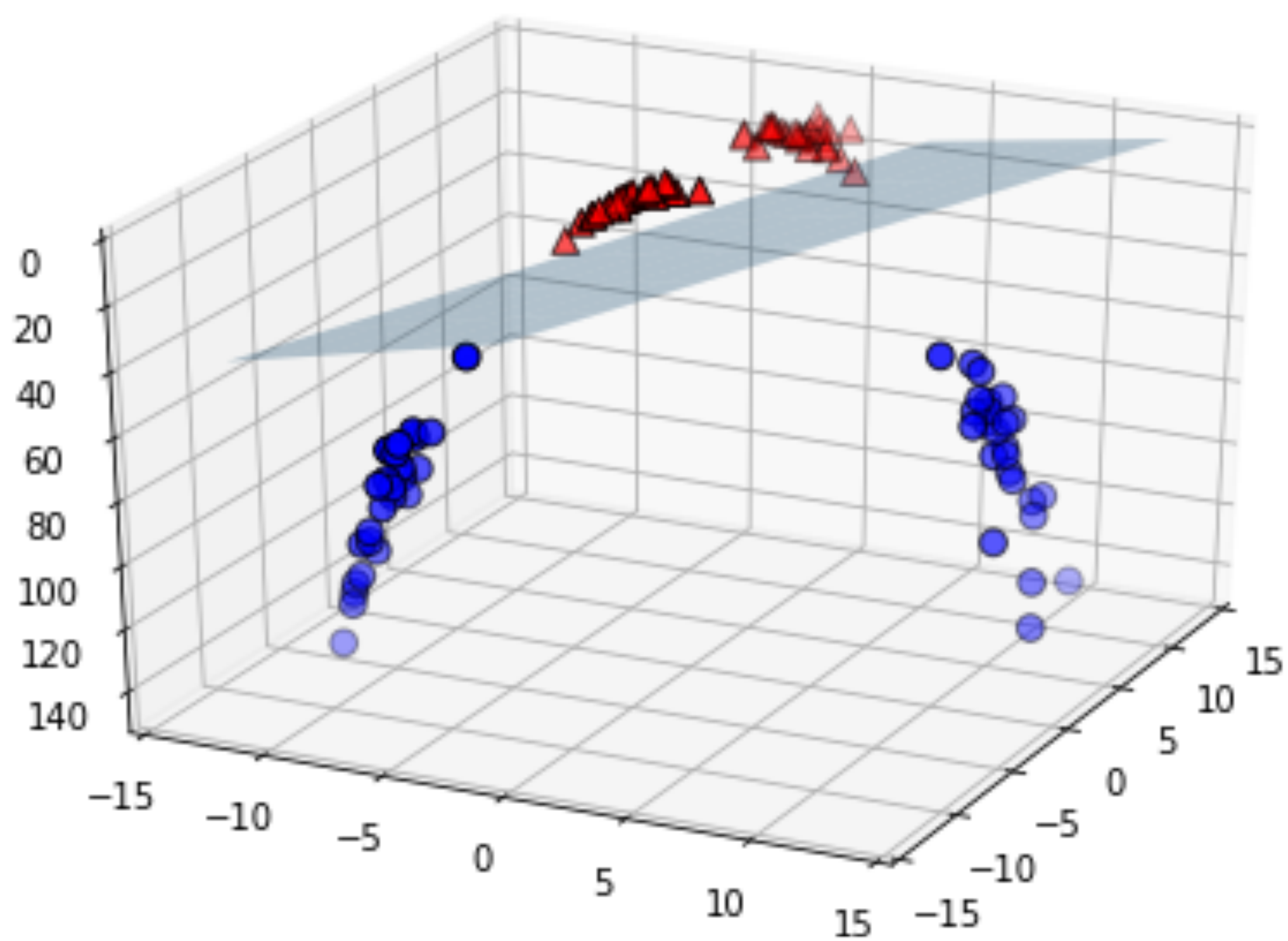
Idea

- Drawing a splitting plane for 3-dim space



Idea

- Drawing a splitting plane for 3-dim space



Idea

- Drawing a splitting ***hyperplane*** for n-dim space
    - A hyperplane of n-dim space is a (n-1)-dim space
        - Hyperplanes of ***3***-dim space are ***planes***
        - Hyperplanes of ***2***-dim space are ***lines***
        - Hyperplanes of ***1***-dim space are ***points***

Hyperplane

- A hyperplane of n-dim space is expressed by:

$$W \cdot X + b = 0$$

  - $X \in \mathbb{R}^n$: input features (with n-dimension)
  - $W \in \mathbb{R}^n$, $b \in \mathbb{R}$: parameters to determine the shape of the hyperplane
  - $W \cdot X$ is the inner product of X and W

- Ex: for 2-dim space

$$W \cdot X + b = 0$$

for $X \in R^2$ and $W \in R^2$

Hyperplane

- A hyperplane of n-dim space is expressed by:

$$W \cdot X + b = 0$$

  - $X \in \mathbb{R}^n$: input features (with n-dimension)
  - $W \in \mathbb{R}^n$, $b \in \mathbb{R}$: parameters to determine the shape of the hyperplane
  - $W \cdot X$ is the inner product of X and W

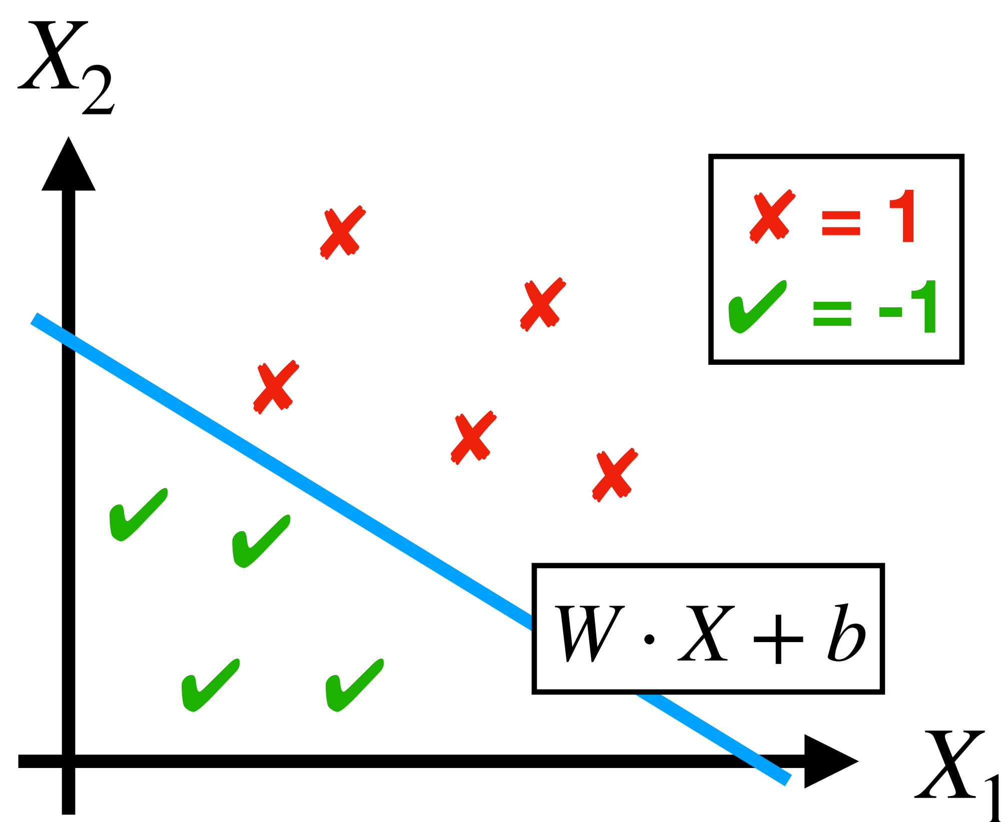
- Ex: for 2-dim space

$$w_1X_1 + w_2X_2 + b = 0$$

  - Another rep. of the standard form $X_2 = \alpha X_1 + \beta$

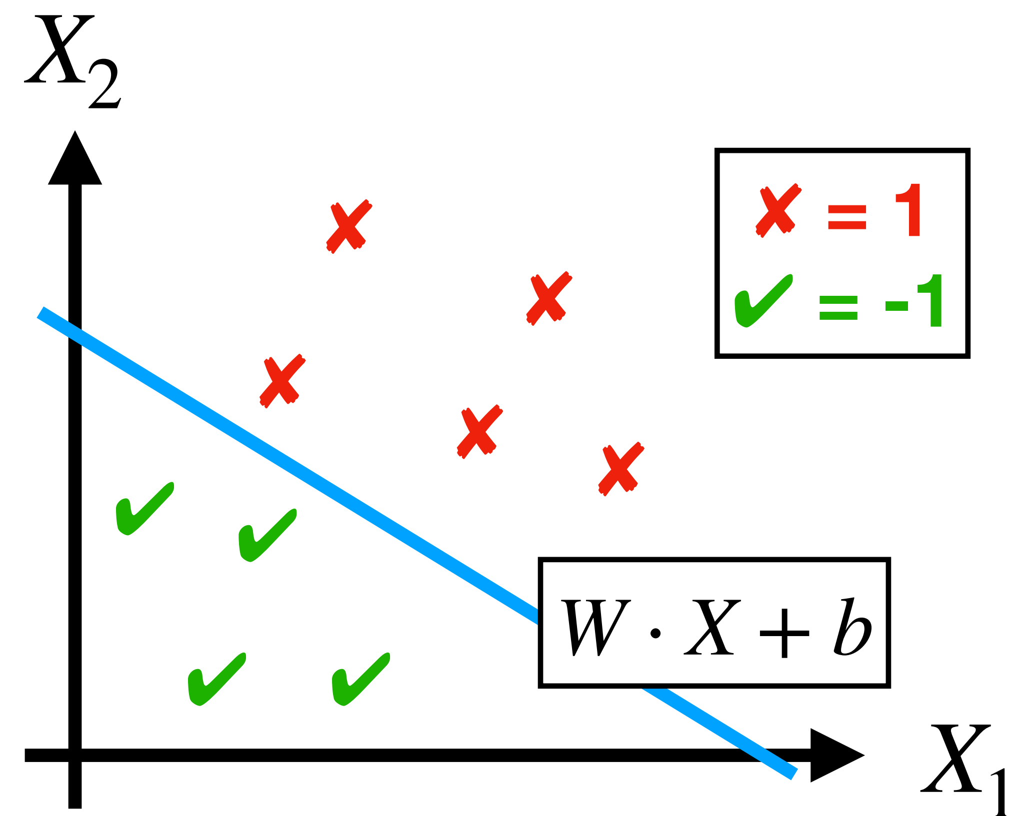
Classification by hyperplane

- Data points are classified by ***separating hyperplanes***
- A hyperplane is separating if and only if, for all the training data points (X, y),
  - $W \cdot X + b \geq 0$ if $y = 1$
  - $W \cdot X + b \leq 0$ if $y = -1$



Classification by hyperplane

- Data points are classified by ***separating hyperplanes***
- A hyperplane is separating if and only if, for all the training data points (X, y),
  - $y(W \cdot X + b) \geq 0$



Questions

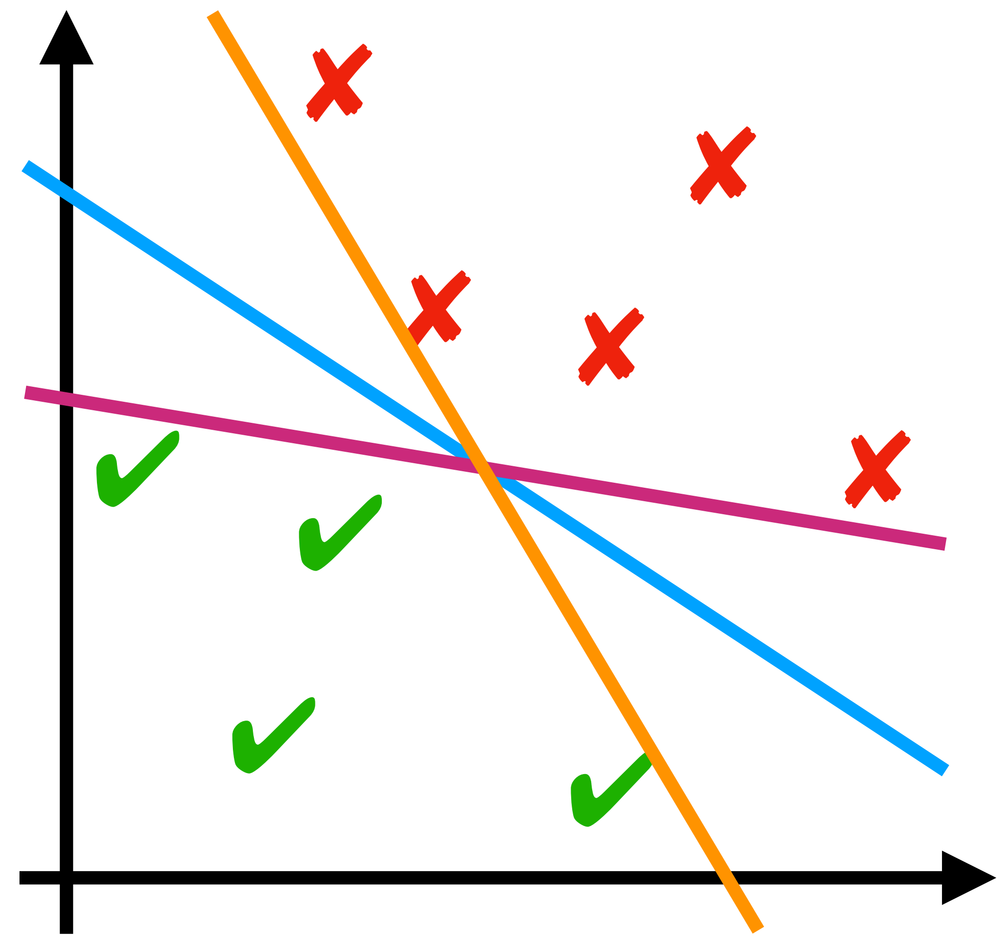
1. Which separating hyperplane is "optimal"?
2. How do we identify the optimal separating hyperplane?
3. How do we handle datasets for which there is no separating hyperplane?

Questions

1. **Which separating hyperplane is “optimal”?**
2. How do we identify the optimal separating hyperplane?
3. How do we handle datasets for which there is no separating hyperplane?

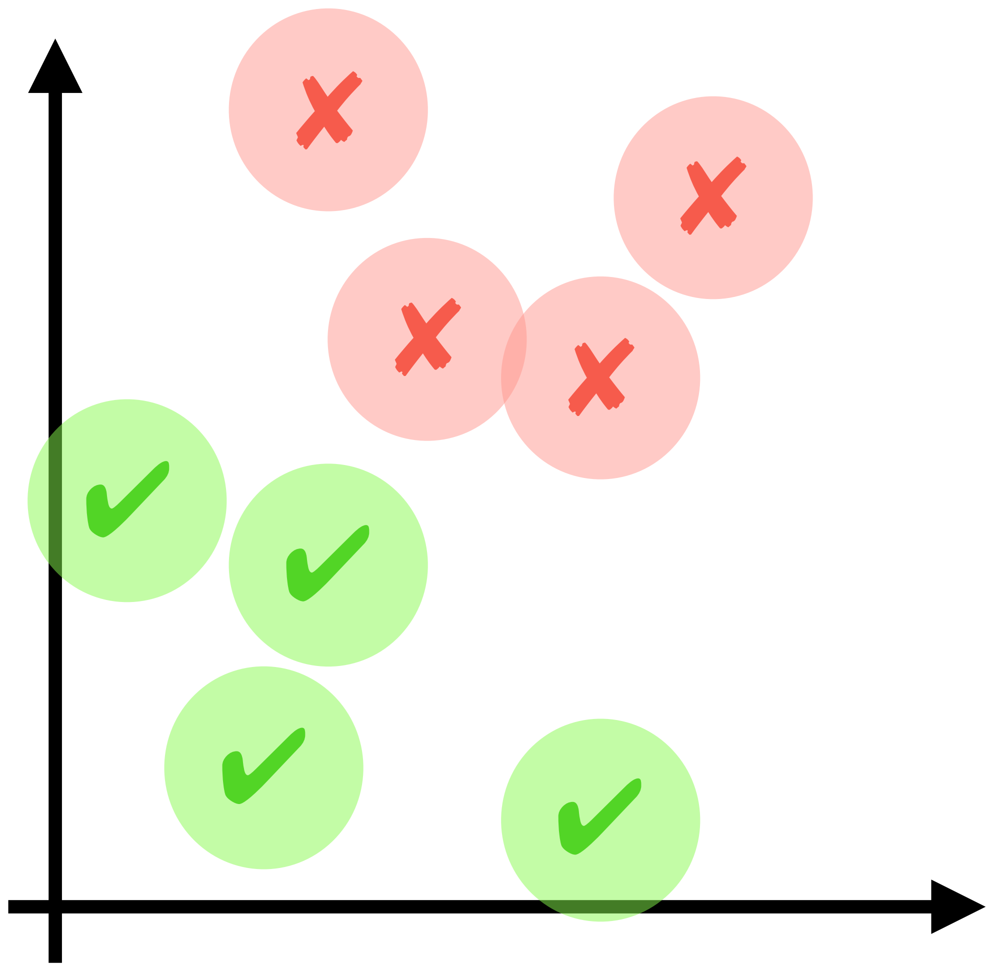
Motivation of Q1

- There can be multiple separating hyperplanes
- We need a metric to measure “goodness” of hyperplanes



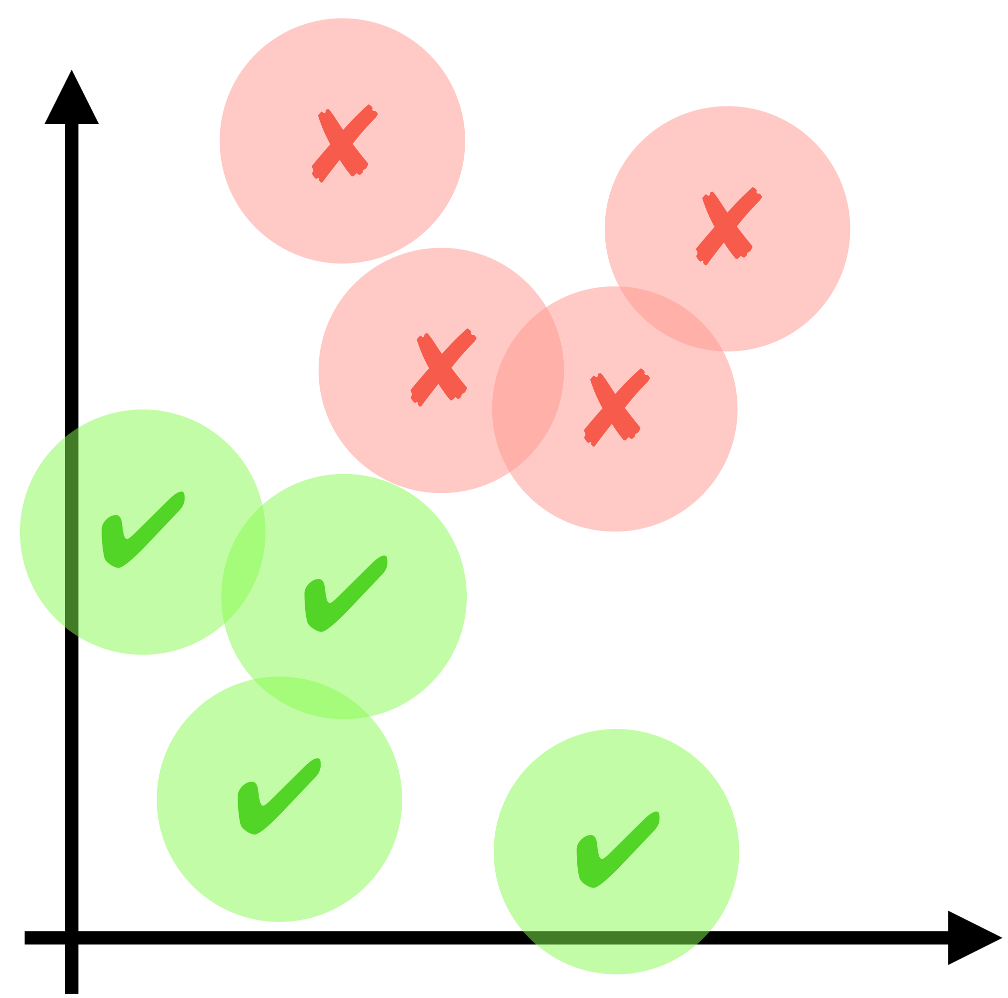
Hypothesis

- Data points around a y-labeled point are likely to be labeled with y



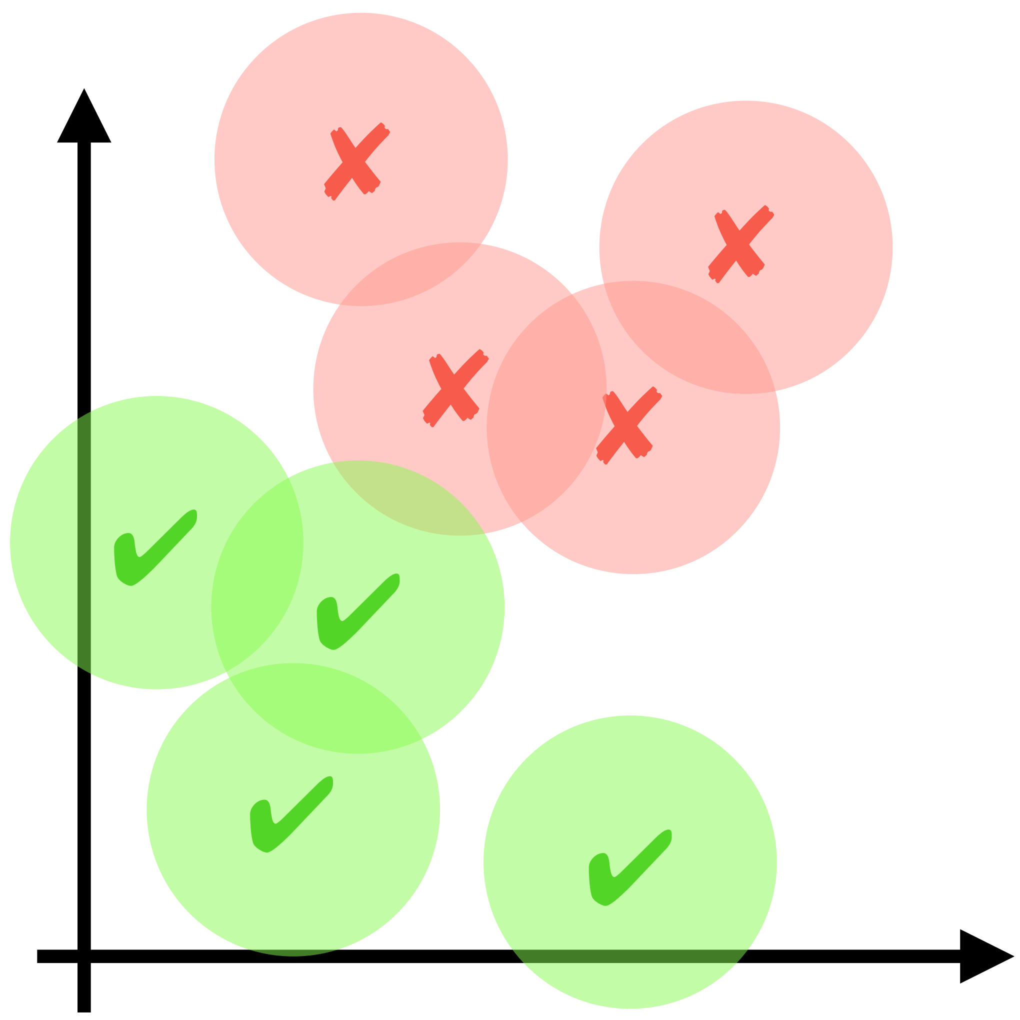
Hypothesis

- Data points around a y-labeled point are likely to be labeled with y



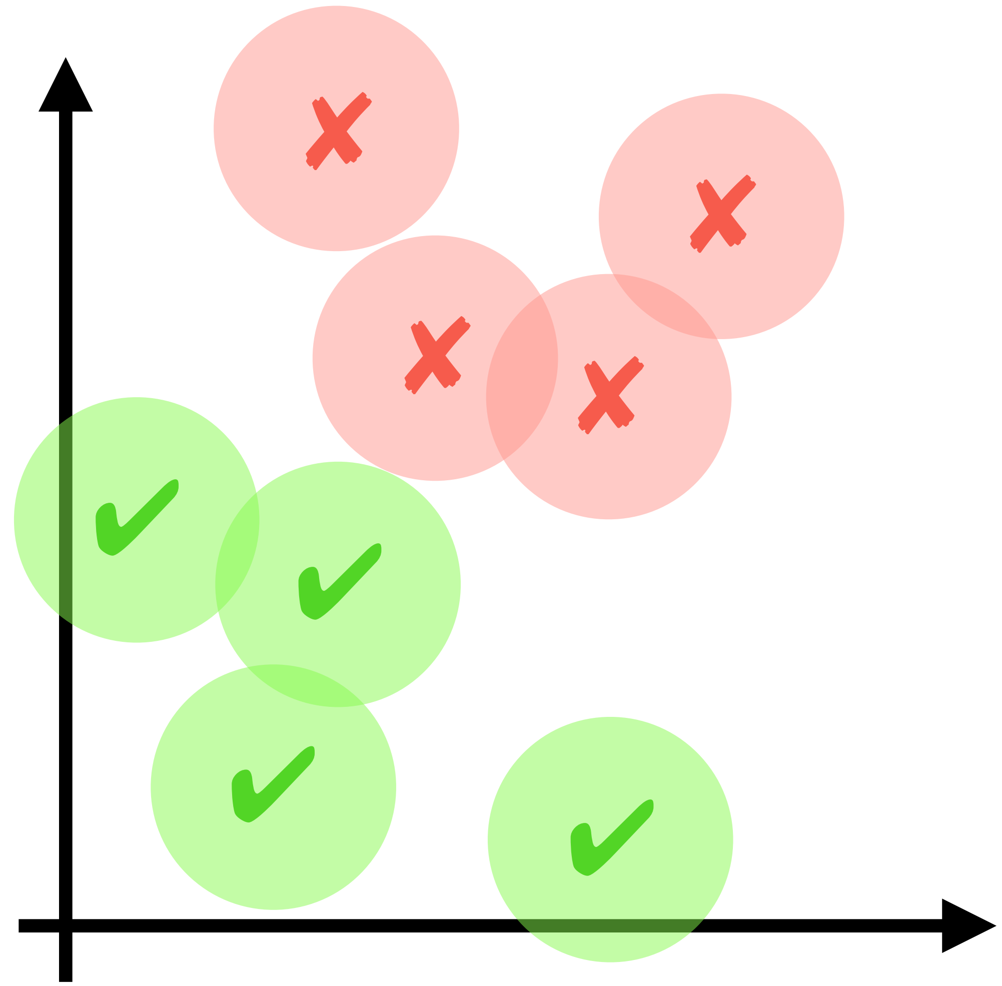
Hypothesis

- Data points around a y-labeled point are likely to be labeled with y



Goal

- To predict y for as many points around a training point with label y as possible



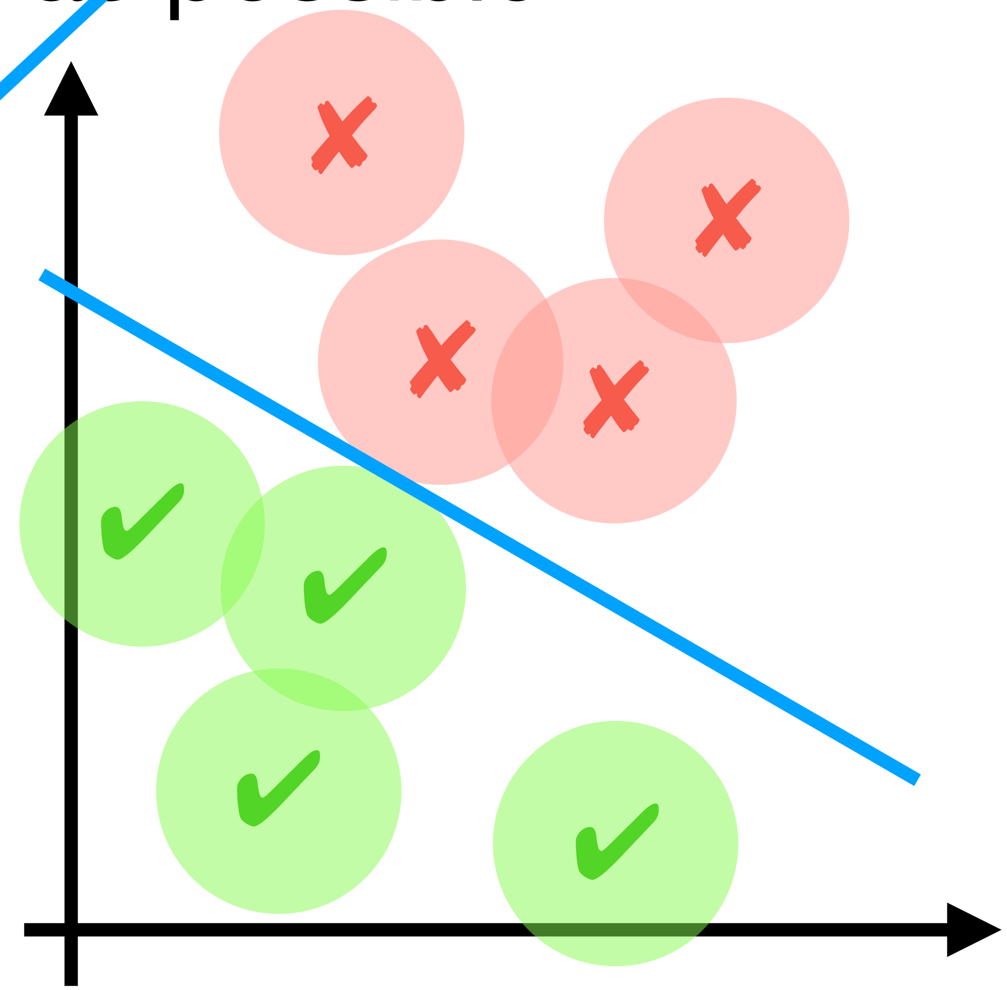
Goal

- To find a separating hyperplane s.t. $y(W \cdot x + b) \geq 0$ holds as many points with features x around a training point labeled with y as possible

$\approx$ Predicting y for point x

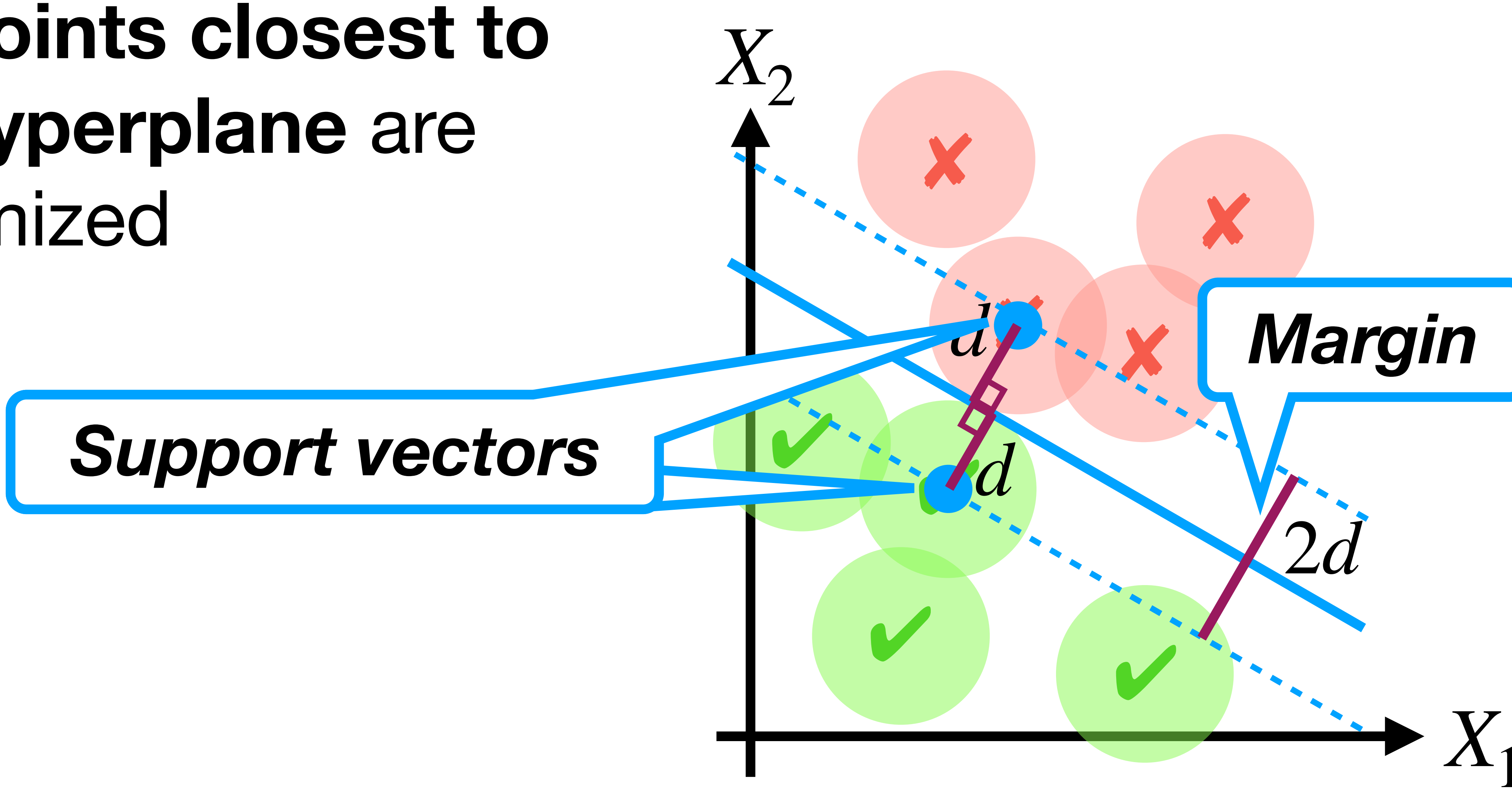
Q. When is the # of such points maximized?

A. When the distance between the hyperplane and the points closest to the hyperplane is maximized



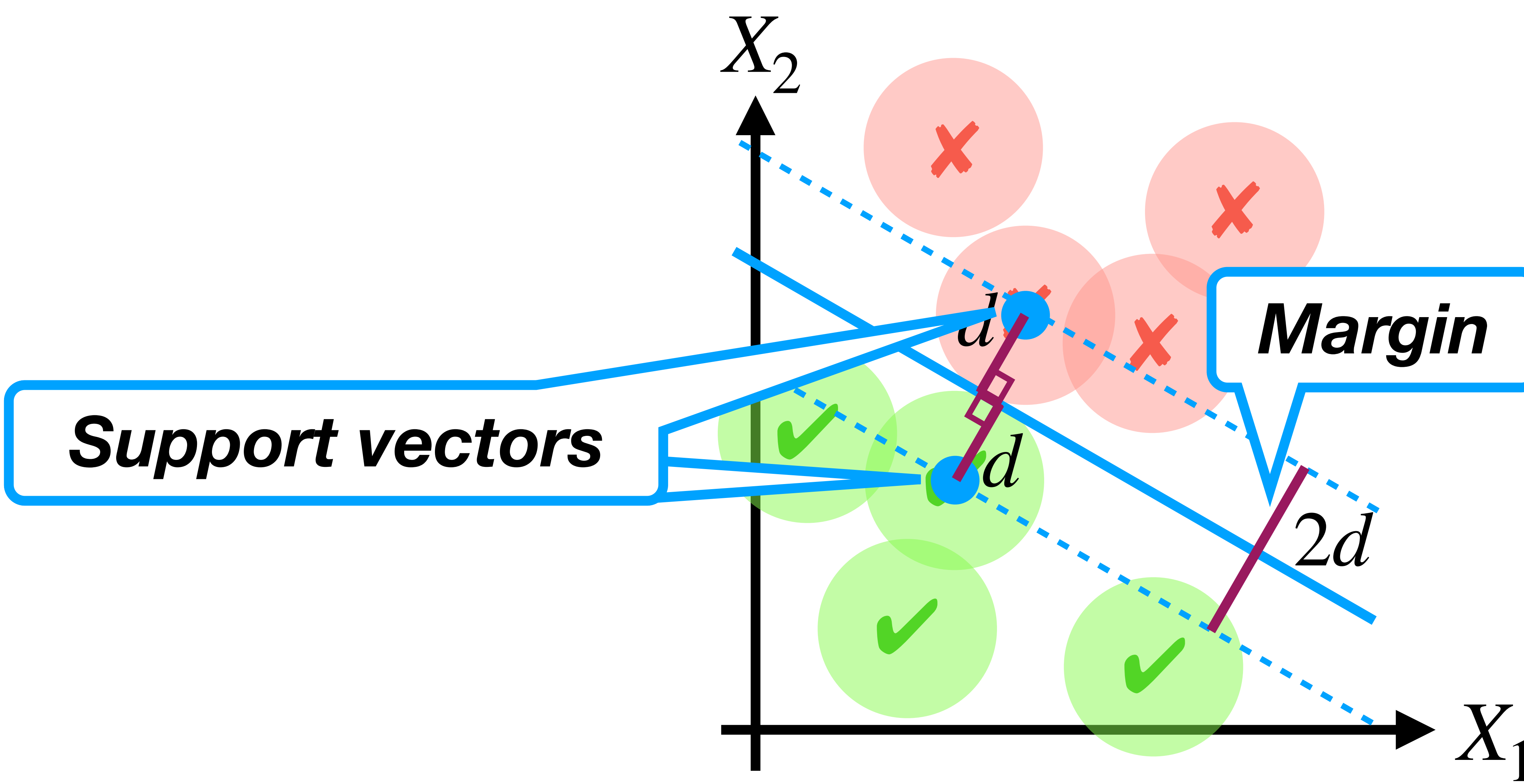
Goal

- To find a separating hyperplane s.t. the distance d between the hyperplane and **the points closest to the hyperplane** are maximized



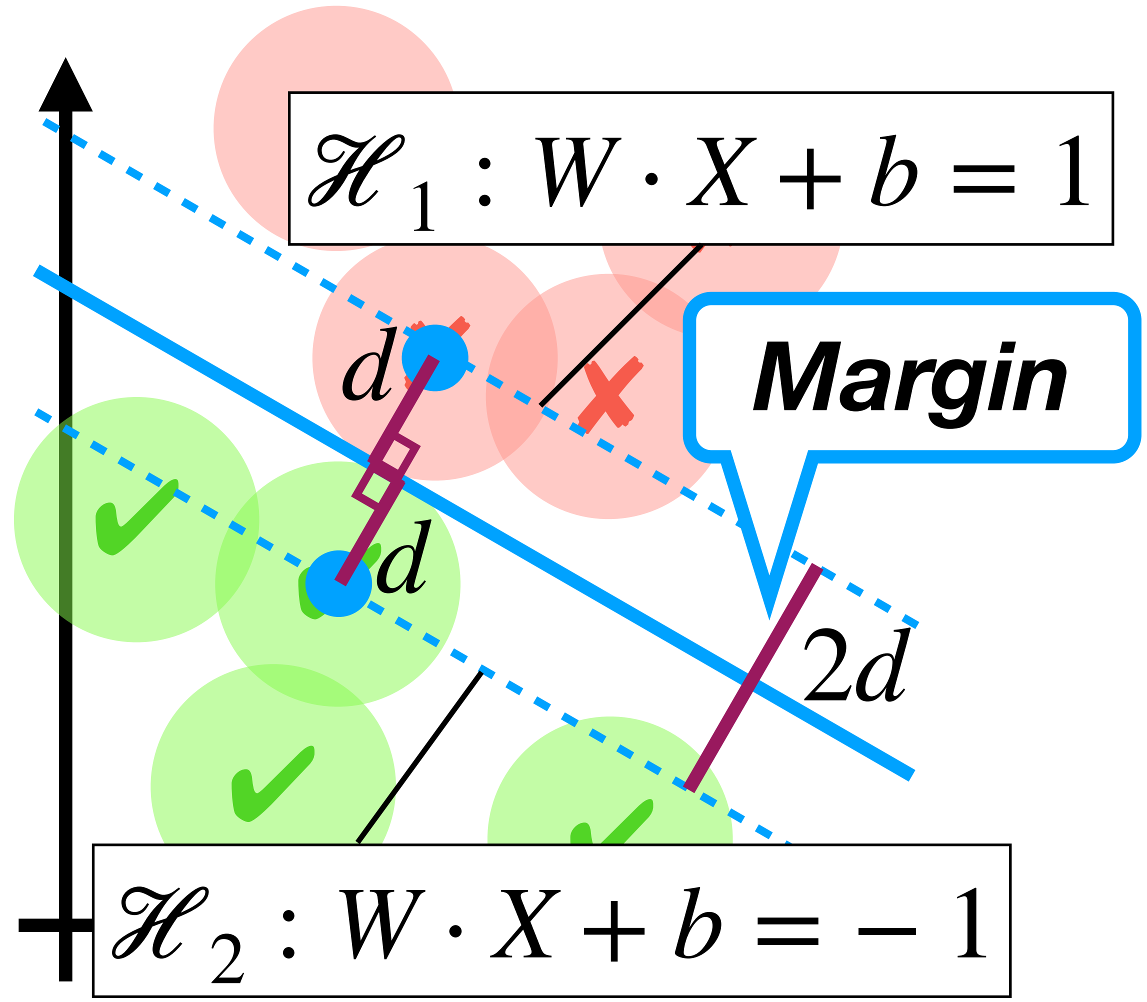
Goal

- To find a separating hyperplane s.t.
  its margin is maximized



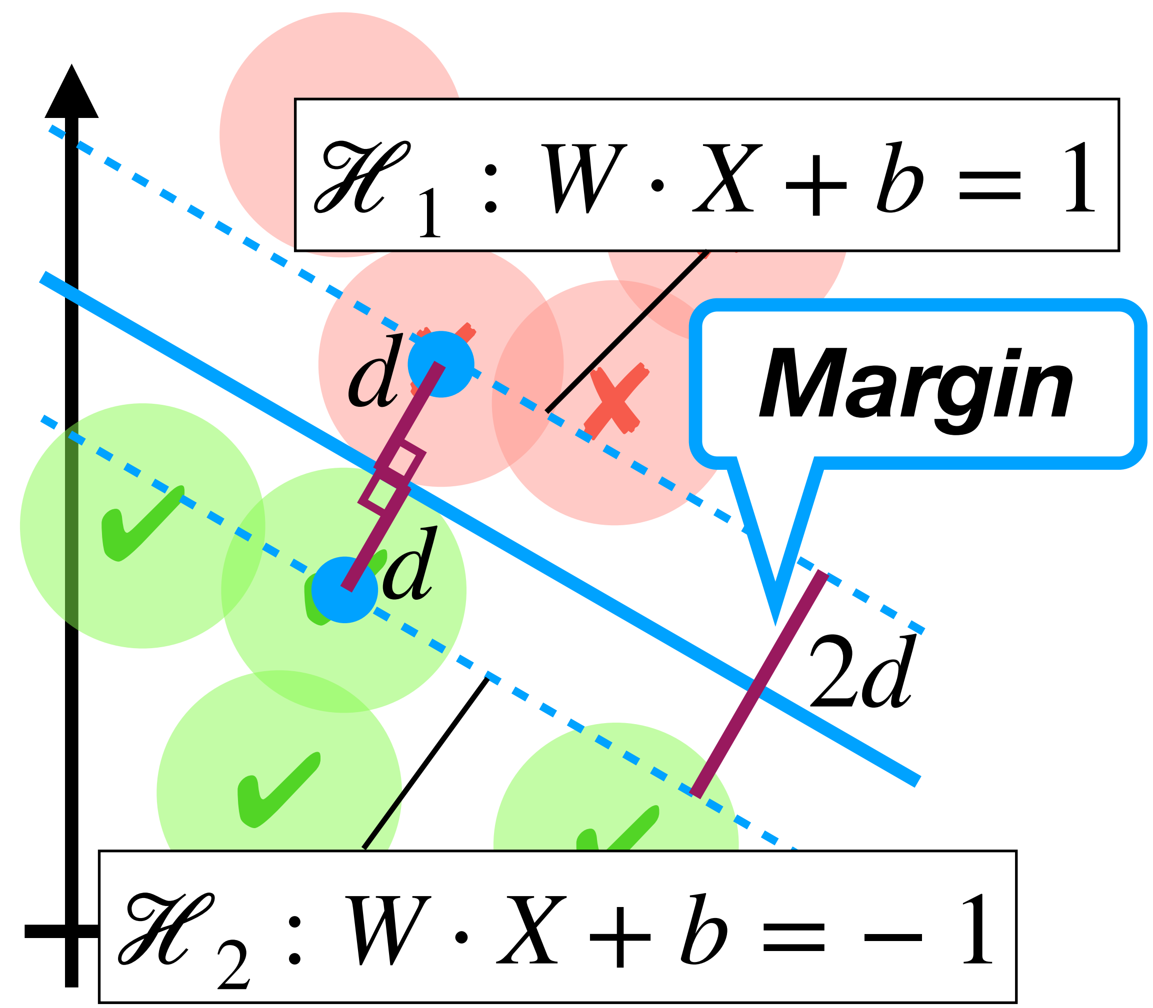
Calculation of margin

- How can the margin be calculated?



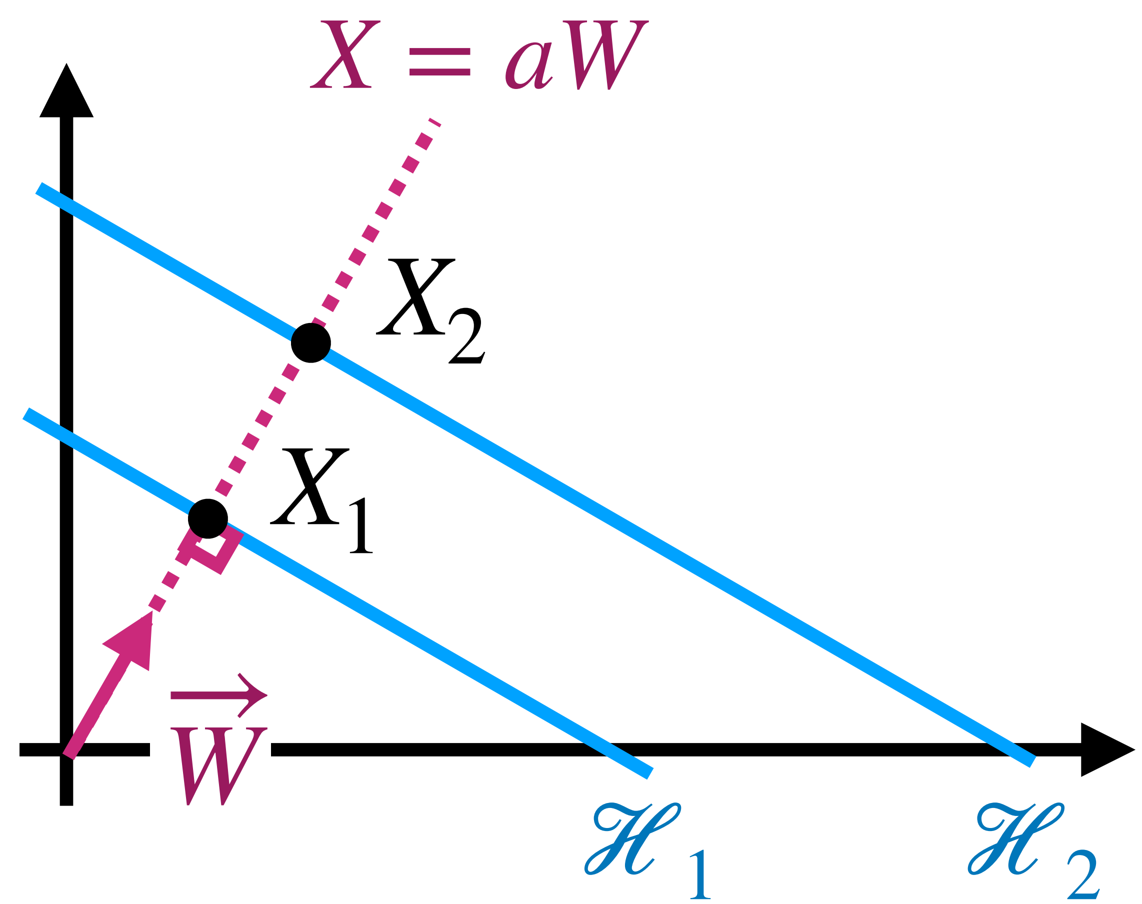
Calculation of margin

- How can the distance between hyperplanes be calculated?



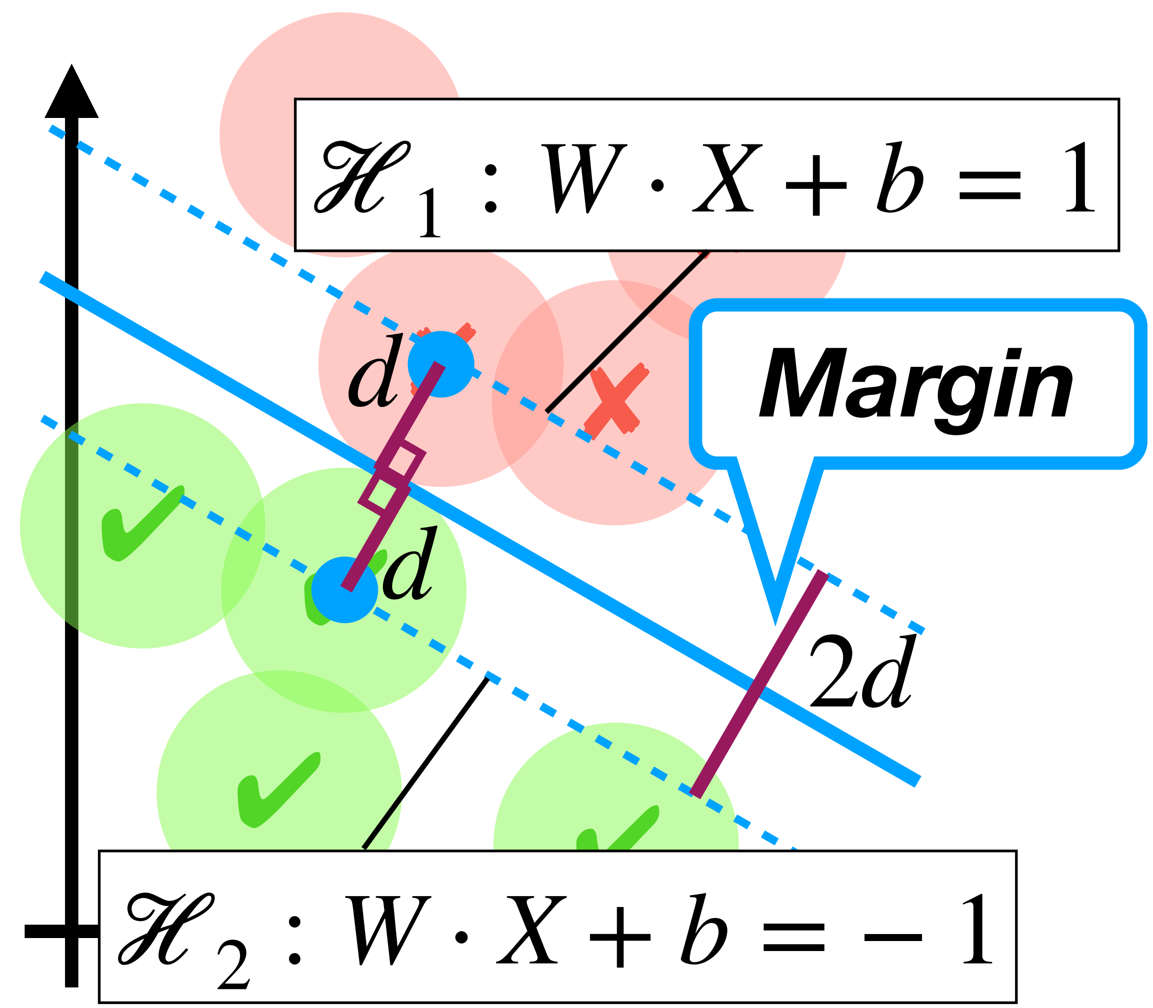
Distance between hyperplanes

- Give hyperplanes $\mathscr{H}_i \equiv W \cdot X + b_i = 0 \; (i \in \{1,2\})$
- Facts
  - W is orthogonal to $\mathscr{H}$
  - The distance is $|X_2 - X_1|$
  - Constraints on X_1, X_2
    - $X_i = a_i W$
    - $W \cdot X_i + b_i = 0$
- Distance = $\dfrac{|b_1 - b_2|}{||W||}$



Calculation of margin

- The margin is $\dfrac{|(b-1)-(b+1)|}{\|W\|} = \dfrac{2}{\|W\|}$



Goal

- To find a separating hyperplane s.t.

$$\text{the margin} = \frac{2}{||W||}$$

  is maximized

Goal

- To find a separating hyperplane s.t.

$$||W||$$

is ***minimized***

Goal

- To find a separating hyperplane s.t.

$$\frac{1}{2}||W||^2$$

is ***minimized***

$$\approx y_i(W \cdot x_i + b) \geq 1$$

for all the training points (x_i, y_i)

Goal

- To find W and b s.t.

$$\frac{1}{2}\|W\|^2$$

is minimized, ***subject to***

$$y_i(W \cdot x_i + b) \geq 1$$

for all the training points (x_i, y_i)

Questions

1. Which separating hyperplane is “optimal”?
2. **How do we identify the optimal separating hyperplane?**
3. How do we handle datasets for which there is no separating hyperplane?

Optimization

- Finding an optimal separating hyperplane is a quadratic optimization problem
- We can solve this problem by reducing it to a **Lagrangian problem**
  - This lecture skips the details of its mathematical development
  - References:
    - C. Bishop. ”Pattern Recognition and Machine Learning” Springer, 2011

Optimization problem…

- To find W and b s.t.

$$\frac{1}{2}\|W\|^2$$

is minimized, subject to

$$y_i(W \cdot x_i + b) \geq 1$$

for all the training points (x_i, y_i)

... Reduces to

- Finding a vector of ***Lagrangian multipliers*** $a \in \mathbb{R}^m$ s.t.

$$\sum_i a_i - \frac{1}{2}\sum_i \sum_j a_i a_j y_i y_j (x_i \cdot x_j)$$

  is maximized, subject to

$$a_i \geq 0 \quad \text{for any } i = 0, \cdots, m$$

$$\sum_i a_i y_i = 0$$

  - m: the number of training data points
- Once such a vector a is found:
  - $W = \sum_i a_i y_i x_i$
  - $b = y_i - \sum_{(x_j, y_j) \in \mathcal{S}} a_j y_j (x_i \cdot x_j)$ for $\mathcal{S} = \{(x_k, y_k) \mid a_k > 0\}$ and $(x_i, y_i) \in \mathcal{S}$)

Time complexity

- The order of the time complexity to find the optimal Lagrangian multipliers a is $O(m^3)$
  - The time complexity of the original optimization problem is $O(n^3)$
  - It works well for datasets where m (= the # of data points) $<$ n (= the # of features)

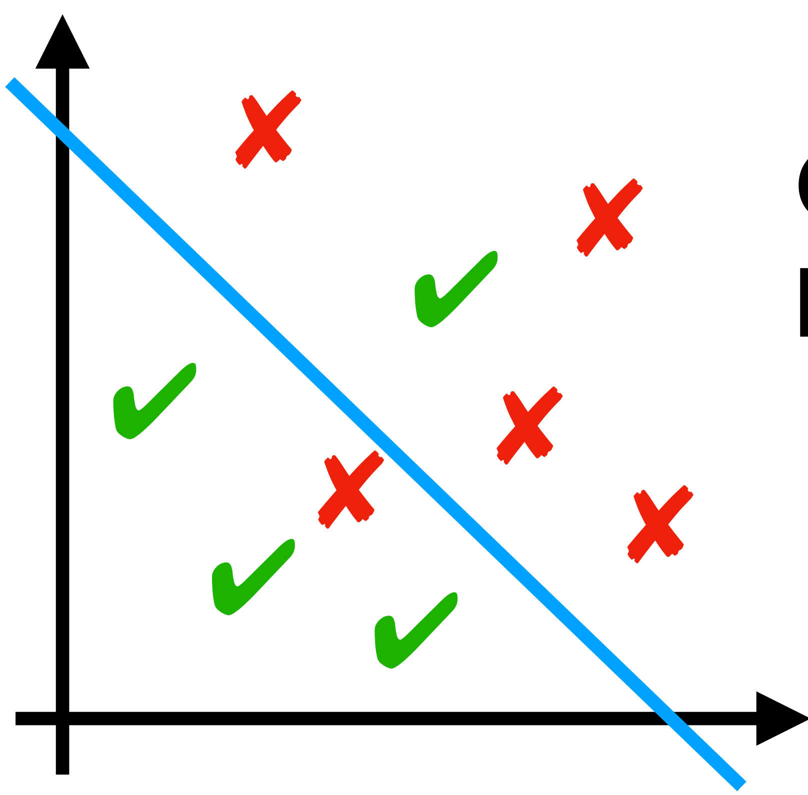
Questions

1. Which separating hyperplane is “optimal”?
2. How do we identify the optimal separating hyperplane?
3. **How do we handle datasets for which there is no separating hyperplane?**
   1. Soft margin
   2. Kernel tricks

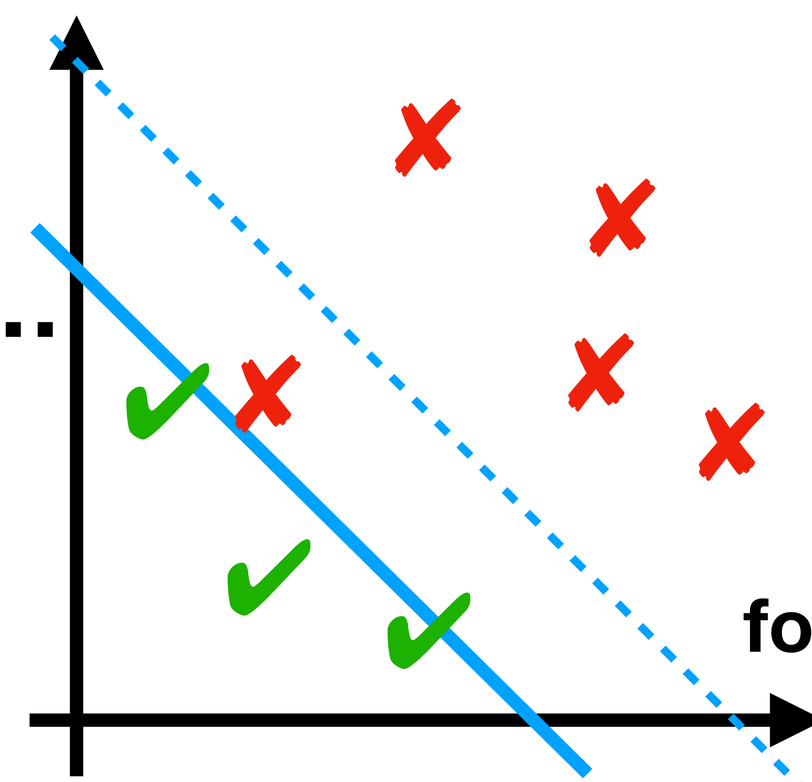
Problem

- In practice, data points may be unable to be clearly separated by any hyperplane
  - Data points involve noise
  - There may be outliers

NG



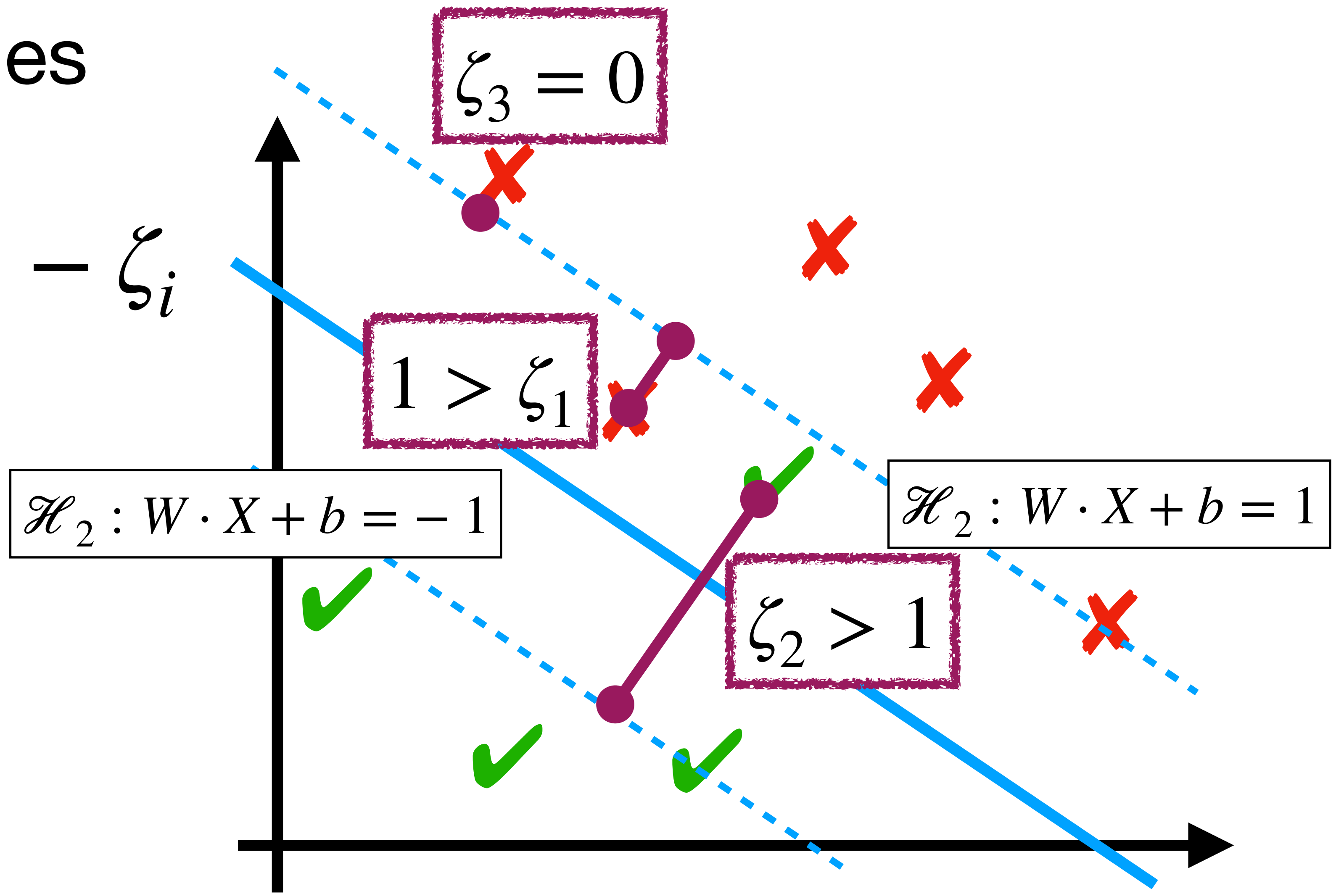
OK, but…



Looks better for unknown points

Solution: soft margin

- To introduce so-called slack variables $\zeta_i \geq 0$ $(i = 1, \cdots, m)$ that denote the errors of points (x_i, y_i)
- Optimal hyperplanes should satisfy $y_i(W \cdot x_i + b) = 1 - \zeta_i$



Formulation

- To find W and b, for given $C \geq 0$, s.t.

$$\frac{1}{2}\|W\|^2 + C\sum_i \zeta_i$$

  is minimized, subject to

$$y_i(W \cdot x_i + b) \geq 1 - \zeta_i$$

  for all the training points (x_i, y_i)

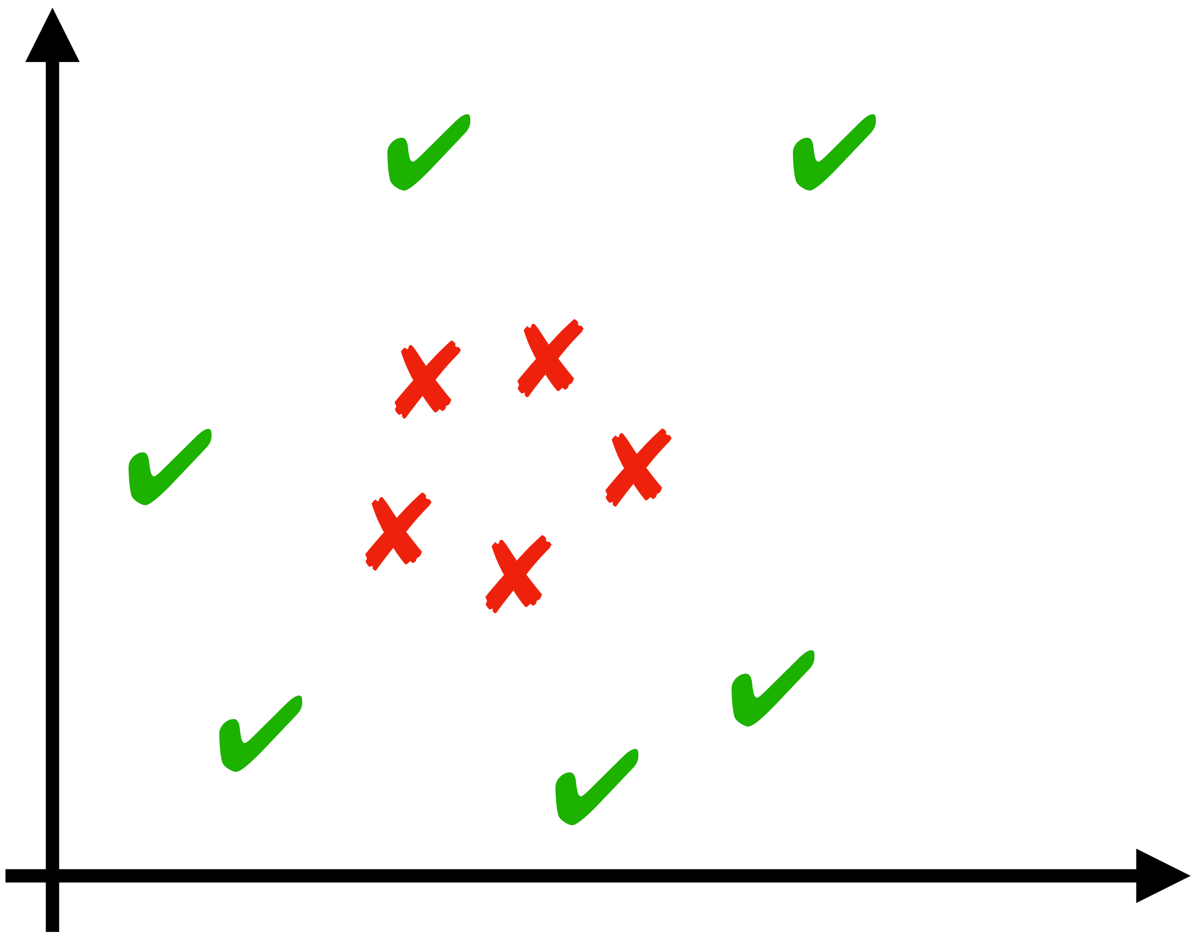
- C is a hyperparameter to control how erroneous data points are allowed
  - If $C = 0$, it is the same as the problem of finding a separating hyperplane

Questions

1. Which separating hyperplane is "optimal"?
2. How do we identify the optimal separating hyperplane?
3. **How do we handle datasets for which there is no separating hyperplane?**
   1. Soft margin
   2. **Kernel tricks**

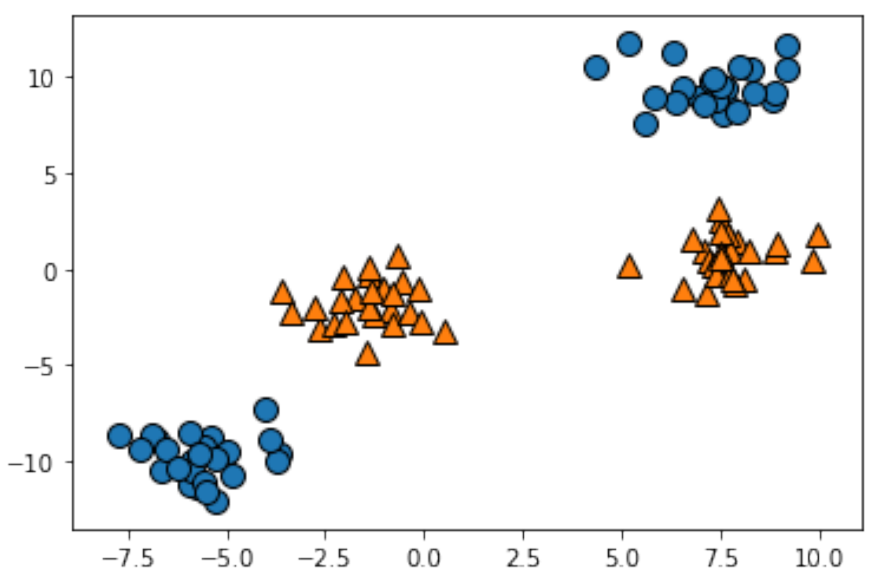
Problem

- The pattern of data points cannot be represented by linear functions

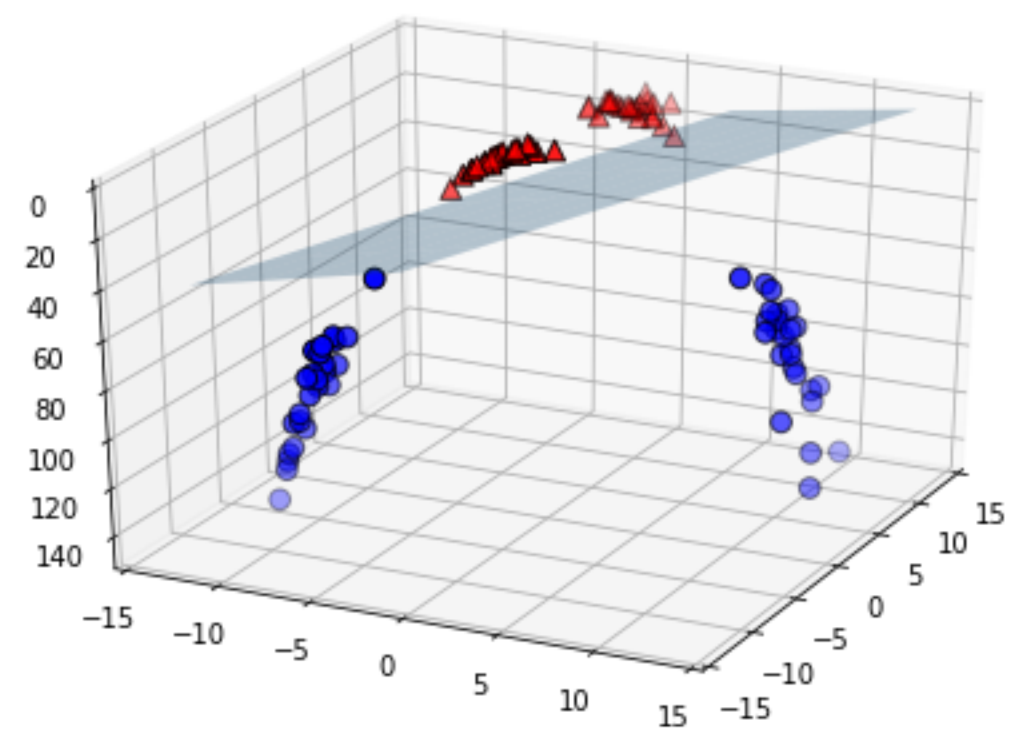


General solution

- Transforming a feature vector into more highly-dimensional space



$$\phi(x_1, x_2) = (x_1, x_2, x_1^2)$$



Kernel trick

- Embedding the transformation of vectors into the optimization problem
- Benefits
  - Efficiency: it is enough to transform only support vectors
  - Enable transformation into the infinitely-dimensional space

Kernel

- The kernel K_ϕ of $\phi : \mathbb{R}^n \to \mathbb{R}^m$ is:

$$K_\phi(x, x') = \phi(x) \cdot \phi(x')$$

  for $x, x' \in \mathbb{R}^n$
- Exampels:
  - Linear kernel: $\phi = x \mapsto x, K_\phi(x, x') = x \cdot x'$
  - Polynominal kernel: $K^{d,c}(x, x') = (x \cdot x + c)^d$

Off-the-shelf Kernels

- Linear kernel

$$\phi = x \mapsto x$$

$$K_\phi(x, x') = x \cdot x'$$

- Polynomial kernel

$$K^{d,c}(x, x') = (x \cdot x + c)^d$$

  - $\phi(x)$ returns a vector involving all the polynomial terms w.r.t. $x_1, \cdots, x_n$ (degree d and constant term c)

Off-the-shelf Kernels

- RBF (radial basis function) kernel

$$K_\phi(x, x') = e^{-\gamma \|x - x'\|}$$

  - $\phi(x)$ returns a vector in the infinitely-dimensional space $\mathbb{R}^\infty$

x is replaced with $\phi(x)$

Optimization problem

- To find W and b s.t.

$$\frac{1}{2}\|W\|^2$$

is minimized, subject to

$$y_i(W \cdot \phi(x_i) + b) \geq 1$$

for all the training points $(\phi(x_i), y_i)$

x is replaced with $\phi(x)$

... Reduces to

- Finding a vector of Lagrangian multipliers $a \in \mathbb{R}^m$ s.t.

$$\sum_i a_i - \frac{1}{2} \sum_i \sum_j a_i a_j y_i y_j (\phi(x_i) \cdot \phi(x_j))$$

is maximized, subject to

$$a_i \geq 0 \text{ for any } i = 0, \cdots, m$$

$$\sum_i a_i y_i = 0$$

  - m: the number of training data points

- Once such a vector a is found:
  - $W = \sum_i a_i y_i \phi(x_i)$
  - $b = y_i - \sum_{(x_j, y_j) \in \mathcal{S}} a_j y_j (\phi(x_i) \cdot \phi(x_j))$ for $\mathcal{S} = \{(\phi(x_k), y_k) \mid a_k > 0\}$ and $(\phi(x_i), y_i) \in \mathcal{S})$

$x_i \cdot x_j$ is replaced with $K_\phi(x_i, x_2)$

… Reduces to

- Finding a vector of Lagrangian multipliers $a \in \mathbb{R}^m$ s.t.

$$\sum_i a_i - \frac{1}{2} \sum_i \sum_j a_i a_j y_i y_j K_\phi(x_i, x_j)$$

  is maximized, subject to

$$a_i \geq 0 \text{ for any } i = 0, \cdots, m$$

$$\sum_i a_i y_i = 0$$

  - m: the number of training data points
- Once such a vector a is found:
  - $W = \sum_i a_i y_i \phi(x_i)$
  - $b = y_i - \sum_{(x_j, y_j) \in \mathcal{S}} a_j y_j K_\phi(x_i, x_j)$ for $\mathcal{S} = \{(\phi(x_k), y_k) \mid a_k > 0\}$ and $(\phi(x_i), y_i) \in \mathcal{S})$

Kernel examples

- Linear kernel

$$\phi = x \mapsto x$$

$$K_\phi(x, x') = x \cdot x'$$

- Polynomial kernel

$$K^{d,c}(x, x') = (x \cdot x + c)^d$$

  - $\phi(x)$ returns a vector involving all the polynomial terms w.r.t. $x_1, \cdots, x_n$ (degree d and constant term c)

... Reduces to

- Finding a vector of ***Lagrangian multipliers*** $a \in \mathbb{R}^m$ s.t.

$$\sum_i a_i - \frac{1}{2} \sum_i \sum_j a_i a_j y_i y_j (x_i \cdot x_j)$$

  is maximized, subject to

$$a_i \geq 0 \text{ for any } i = 0, \cdots, m$$

$$\sum_i a_i y_i = 0$$

  - m: the number of training data points
- Once such a vector a is found:
  - $W = \sum_i a_i y_i x_i$
  - $b = y_i - \sum_{(x_j, y_j) \in \mathcal{S}} a_j y_j (x_i \cdot x_j)$ for $\mathcal{S} = \{(x_k, y_k) \mid a_k > 0\}$ and $(x_i, y_i) \in \mathcal{S}$)

Questions

1. Which separating hyperplane is “optimal”?
2. How do we identify the optimal separating hyperplane?
3. **How do we handle datasets for which there is no separating hyperplane?**

Notice

- Next week is for programming
  - Assignments will be notified via Course N@vi by next week
  - The room will be E-room, 3rd floor, Building 63
  - You need to bring your laptop
- (Common) Programming weeks are for making opportunities to receive questions
  - No new information will be provided
  - No attendance will be taken