

Support Vector Machine

Mahdi Roozbahani
Georgia Tech

Math moment

$$7 \times 0 = 0$$

$$3 \times 0 = 0$$

$$0 = 0$$

$$7 \times 0 = 3 \times 0$$

$$7 = 3$$



Outline

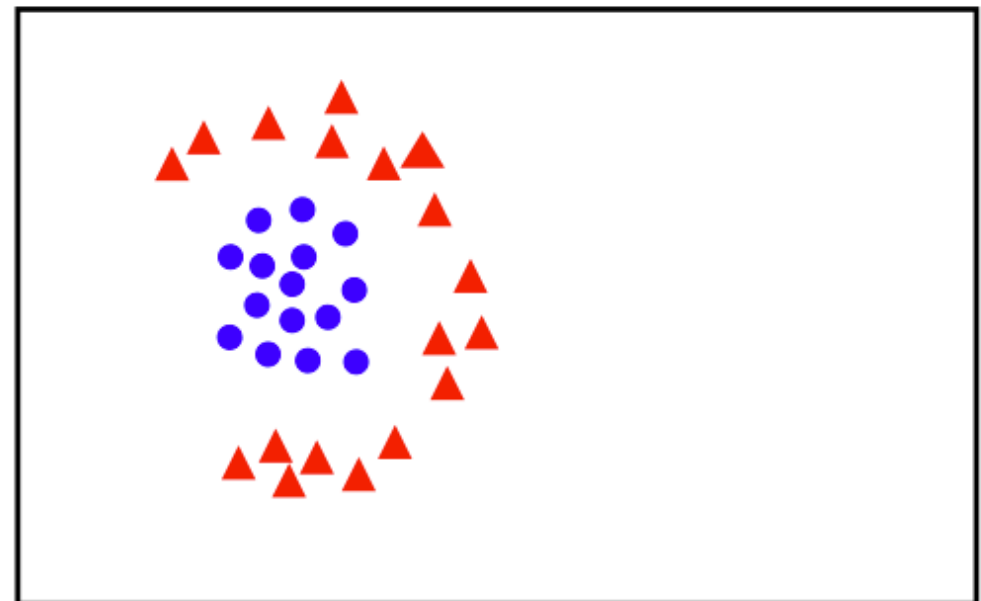
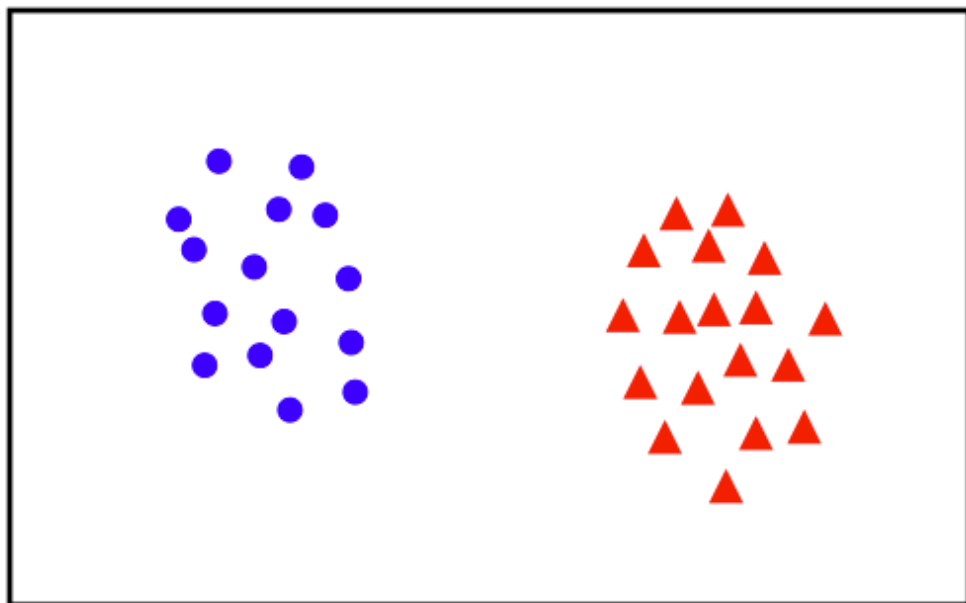
- Precursor: Linear Classifier and Perceptron 
- Support Vector Machine
- Parameter Learning

Binary Classification

Given training data $(x^{(i)}, y^{(i)})$ for $i = 1 \dots N$, with $x^{(i)} \in \mathbb{R}^d$ and $y^{(i)} \in \{-1, 1\}$, learn a classifier $f(\mathbf{x})$ such that

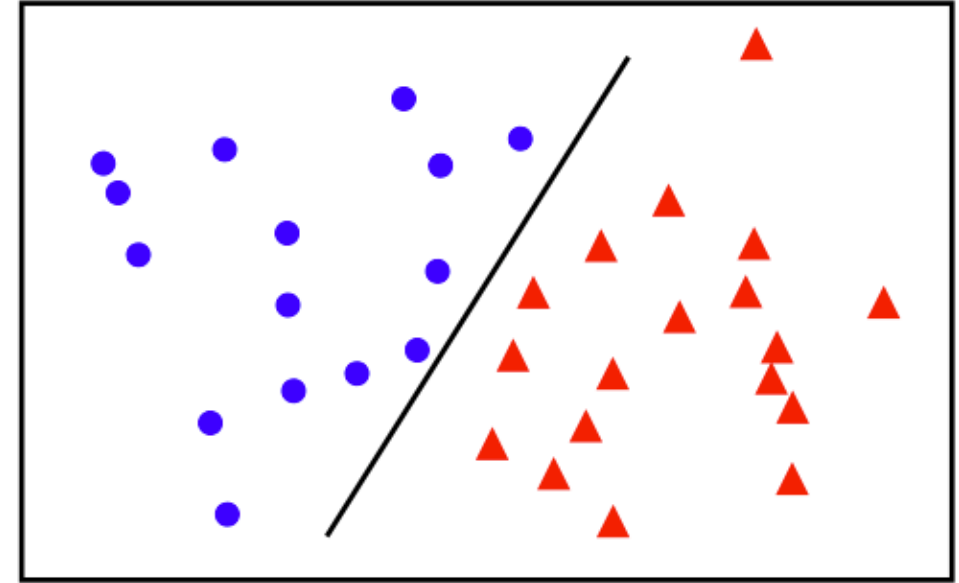
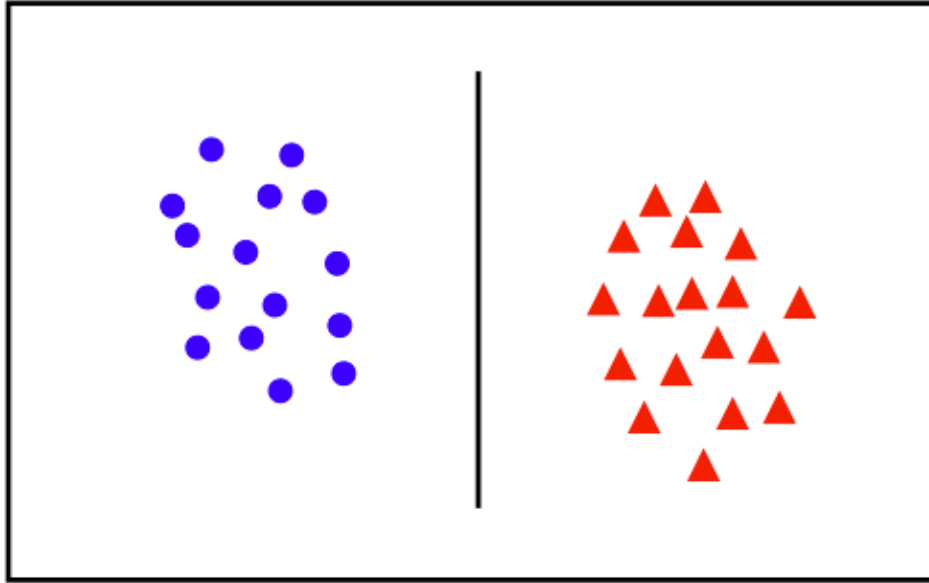
$$f(x^{(i)}) \begin{cases} \geq 0 & +1 \\ < 0 & -1 \end{cases}$$

i.e. $\overset{+}{-} \overset{+}{-} y^{(i)} f(x^{(i)}) > 0$ for a correct classification.

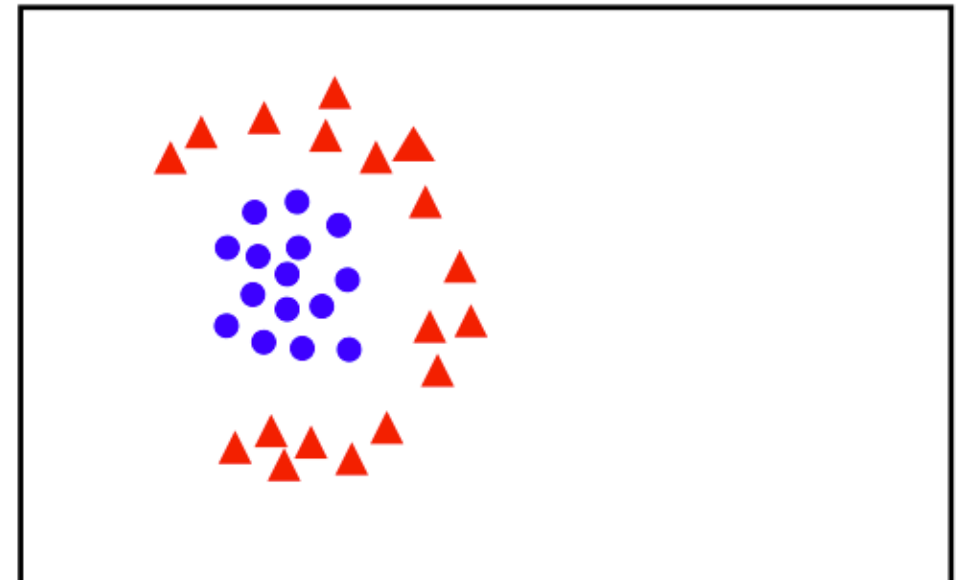
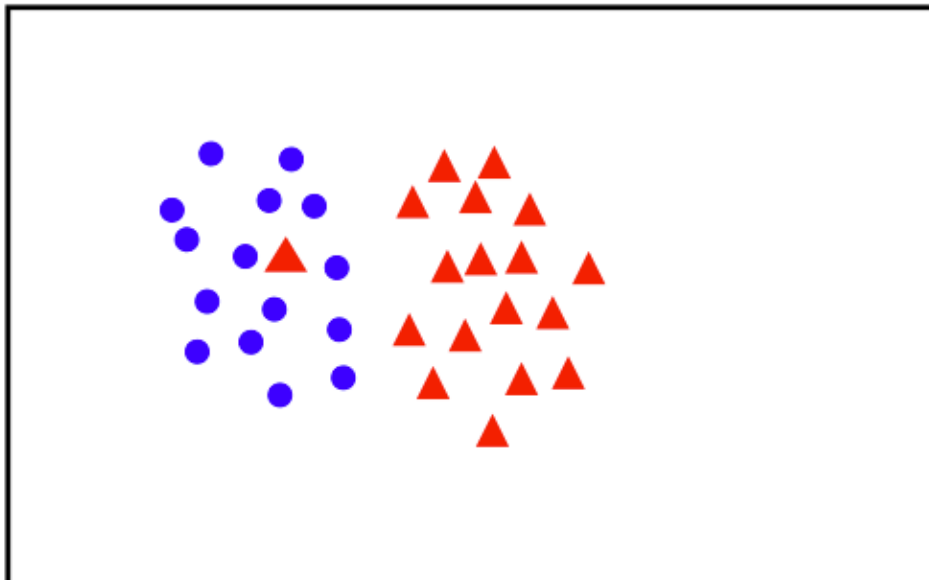


Linear Separability

linearly
separable



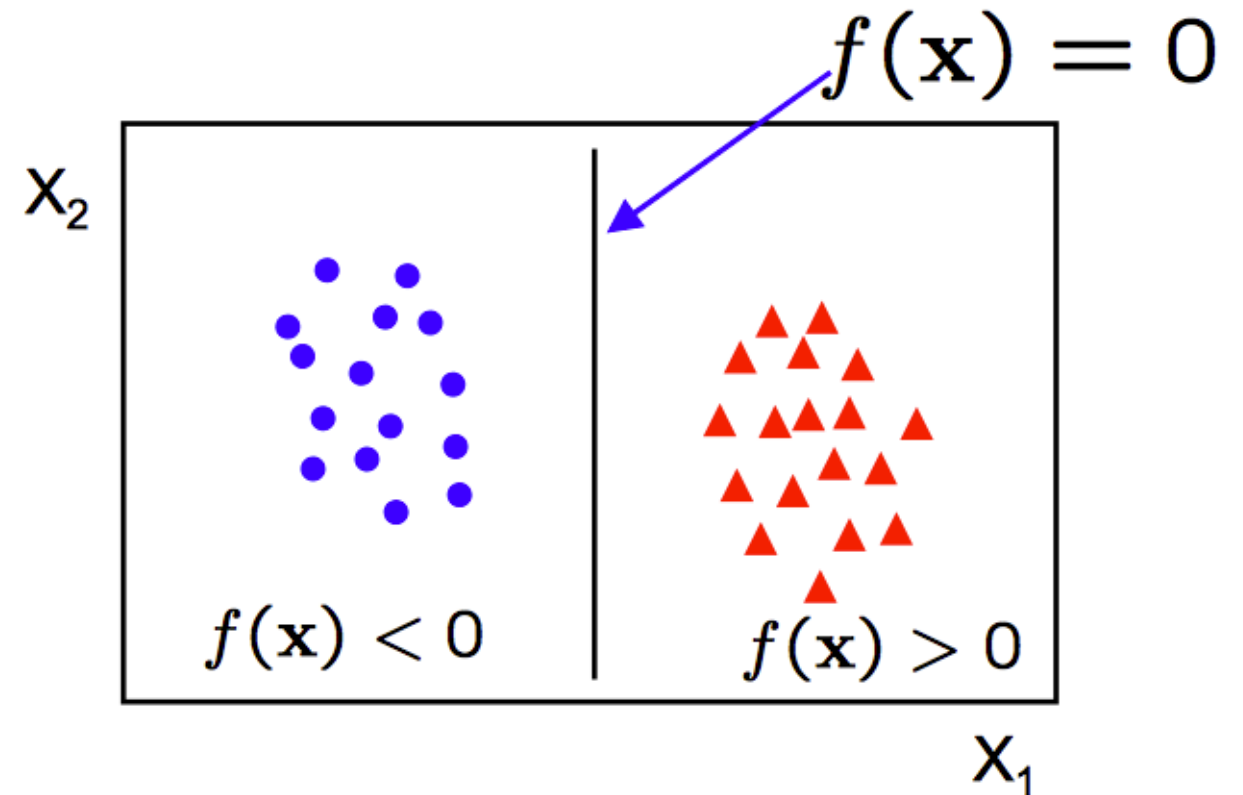
not
linearly
separable



Linear Classifier

A linear classifier has the form

$$f(x) = x\theta + \theta_0$$

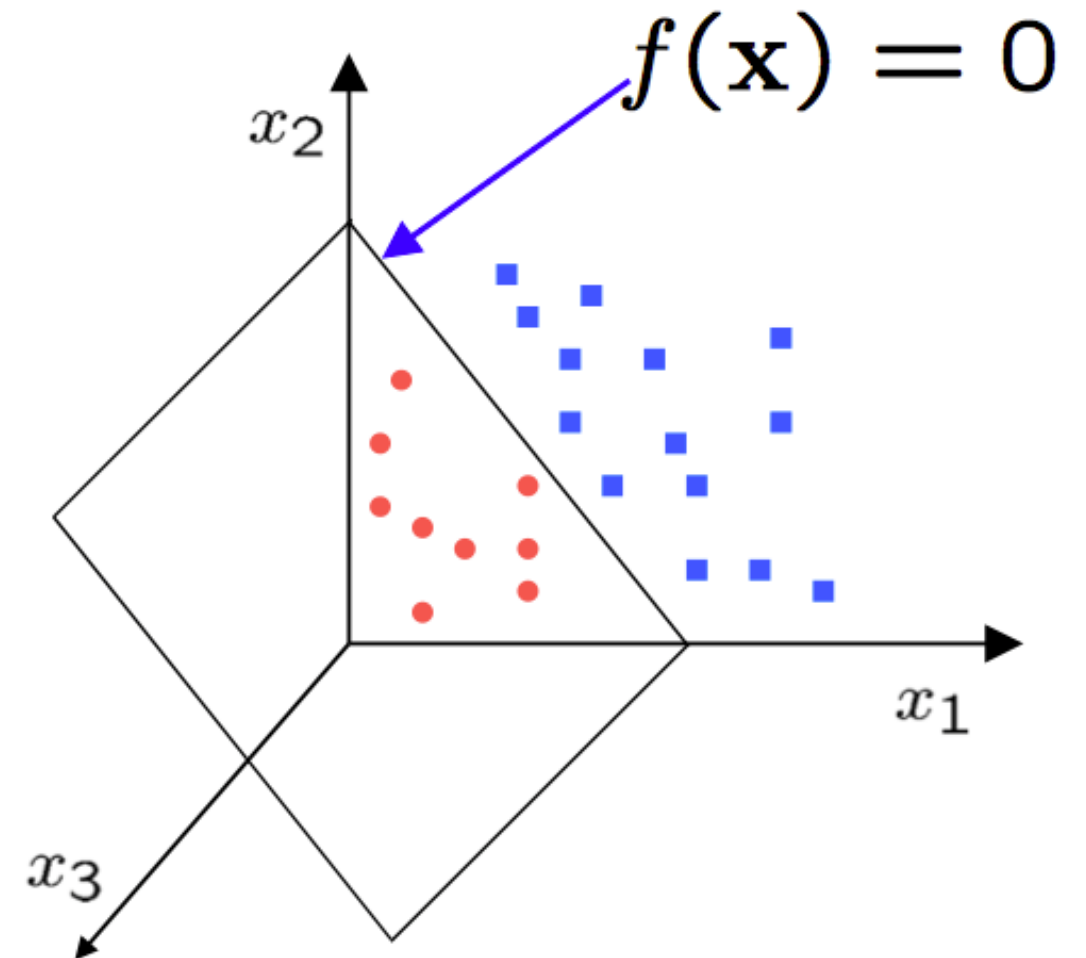


- in 2D the discriminant is a line
- θ is the **normal** to the line, and θ_0 the **bias**
- θ is known as the **weight vector**

Linear Classifier (higher dimension)

A linear classifier has the form

$$f(x) = x\theta + \theta_0$$



- in 3D the discriminant is a plane, and in nD it is a hyperplane

The Perceptron Classifier

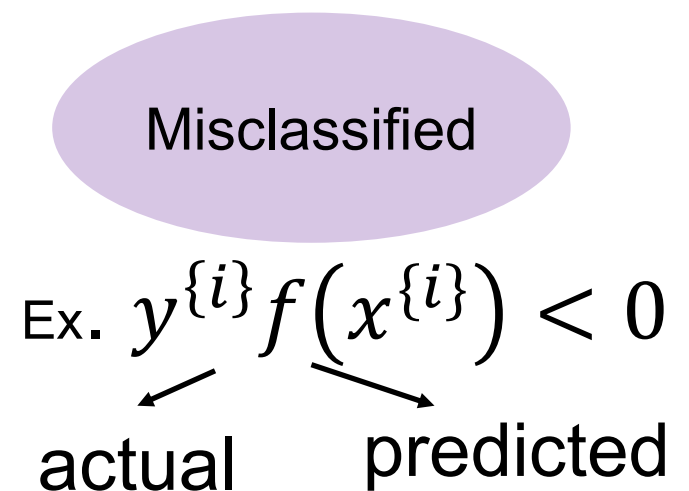
Considering $\mathbf{x}$ is linearly separable and y has two labels of $\{-1, 1\}$

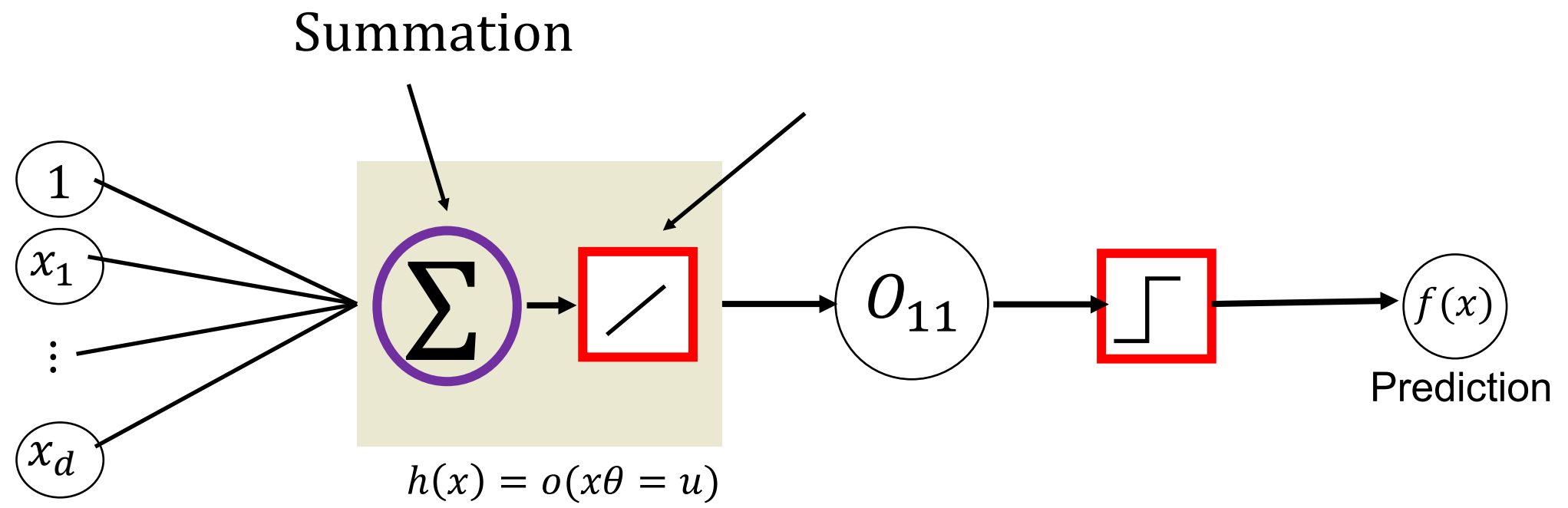
$$f(x^{(i)}) = x^{(i)} \theta \quad \text{Bias is inside } \theta \text{ now}$$

How can we separate datapoints with label 1 from datapoints with label -1 using a line?

Perceptron Algorithm:

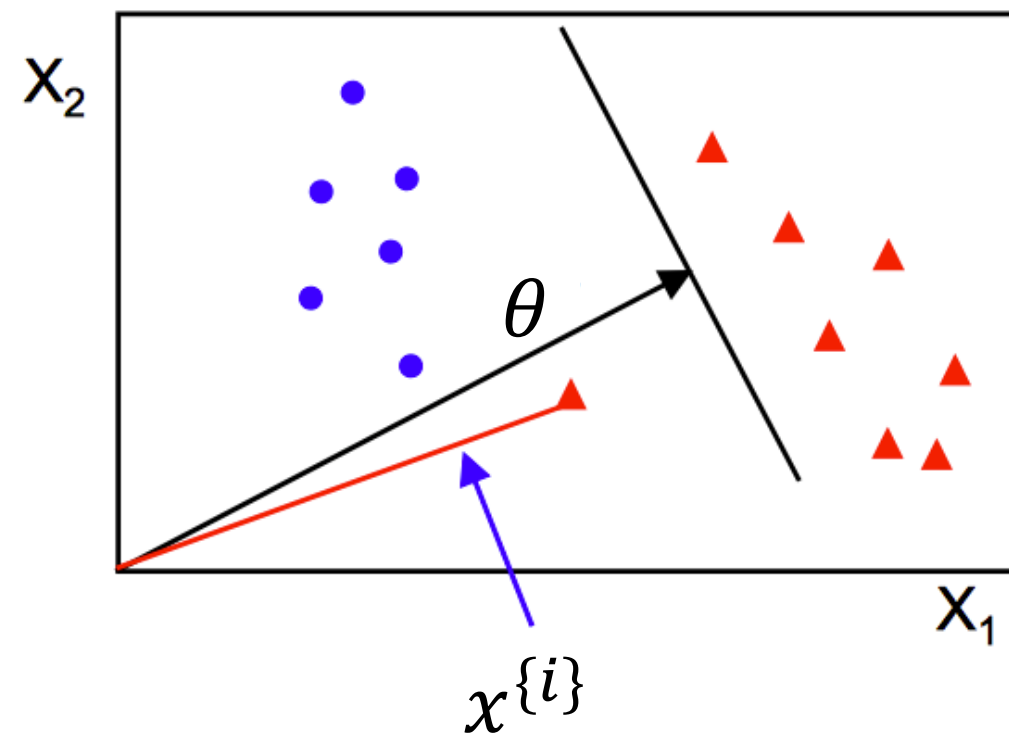
- Initialize $\theta = 0$
- Go through each datapoint $\{x^{(i)}, y^{(i)}\}$
 - If $x^{(i)}$ is misclassified then $\theta^{t+1} \leftarrow \theta^t + \alpha y^{(i)} x^{(i)}$
- Until all datapoints are correctly classified



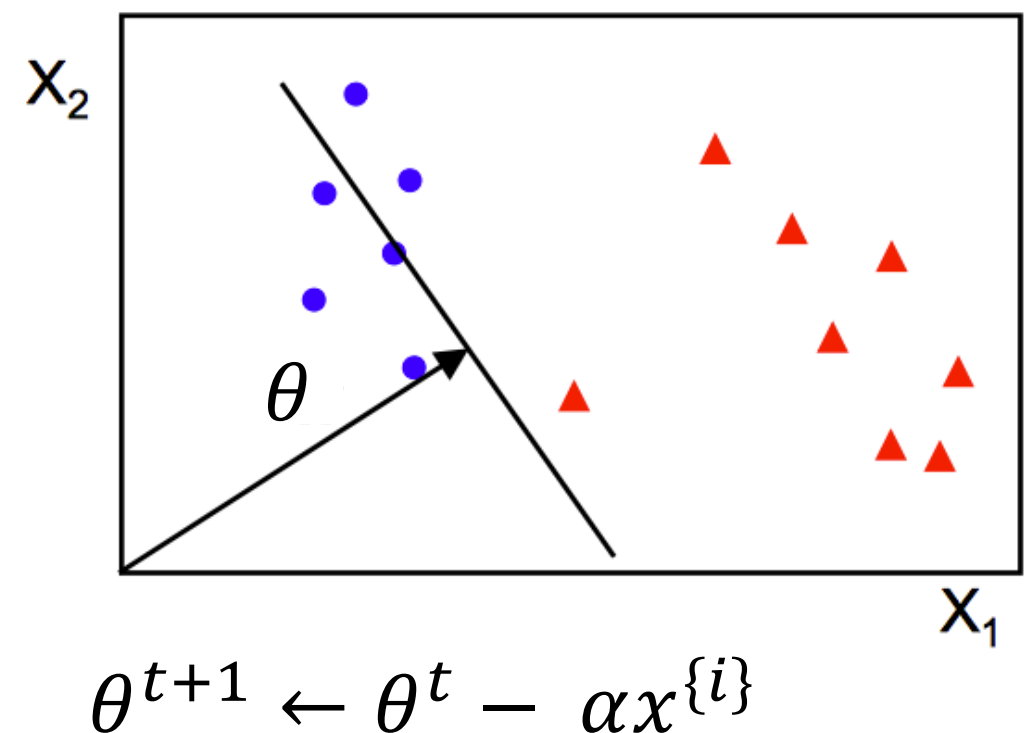


- Initialize $\theta = 0$
- Go through each datapoint $\{x^{(i)}, y^{(i)}\}$
 - If $x^{(i)}$ is misclassified then $\theta^{t+1} \leftarrow \theta^t + \alpha y^{(i)} x^{(i)}$
- Until all datapoints are correctly classified

before update

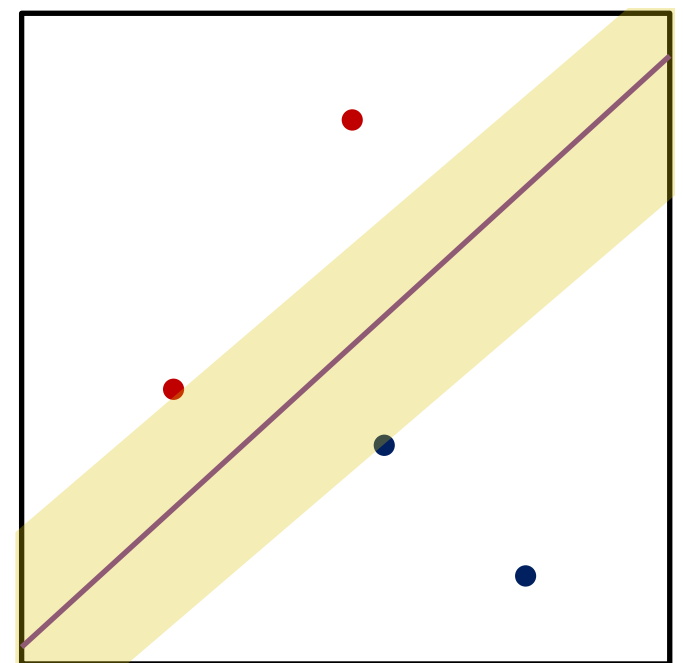
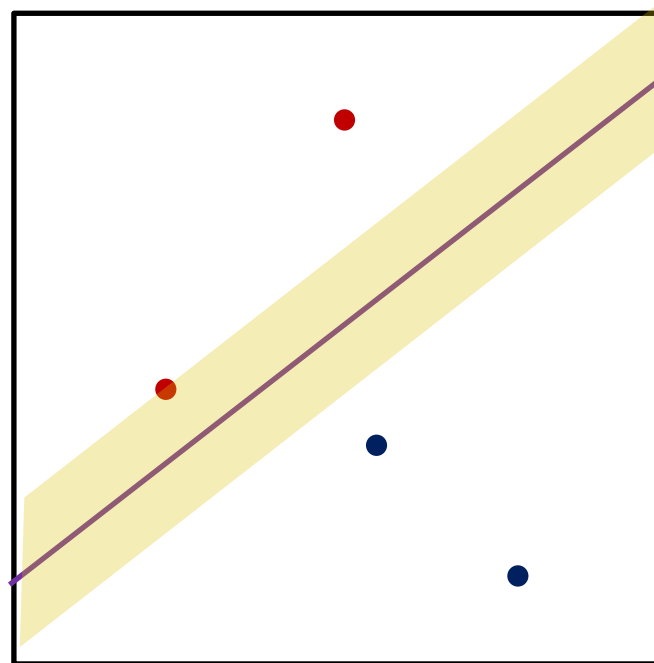
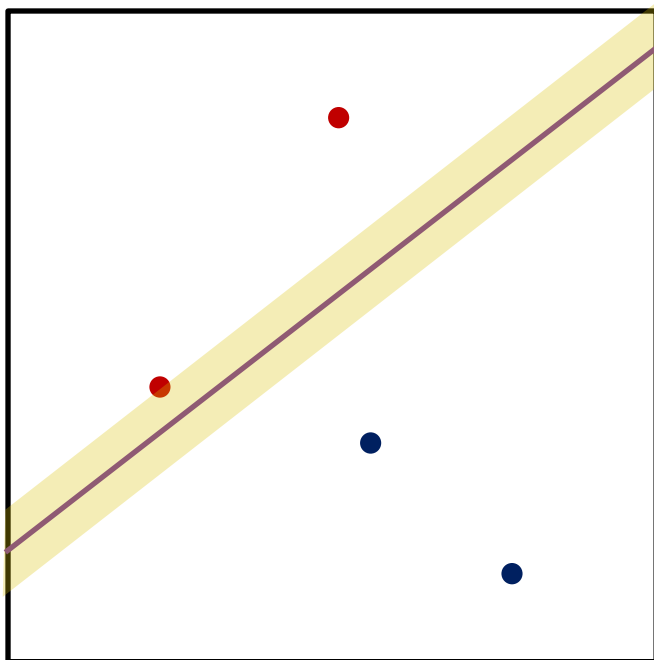


after update



Linear separation

We can have different separating lines



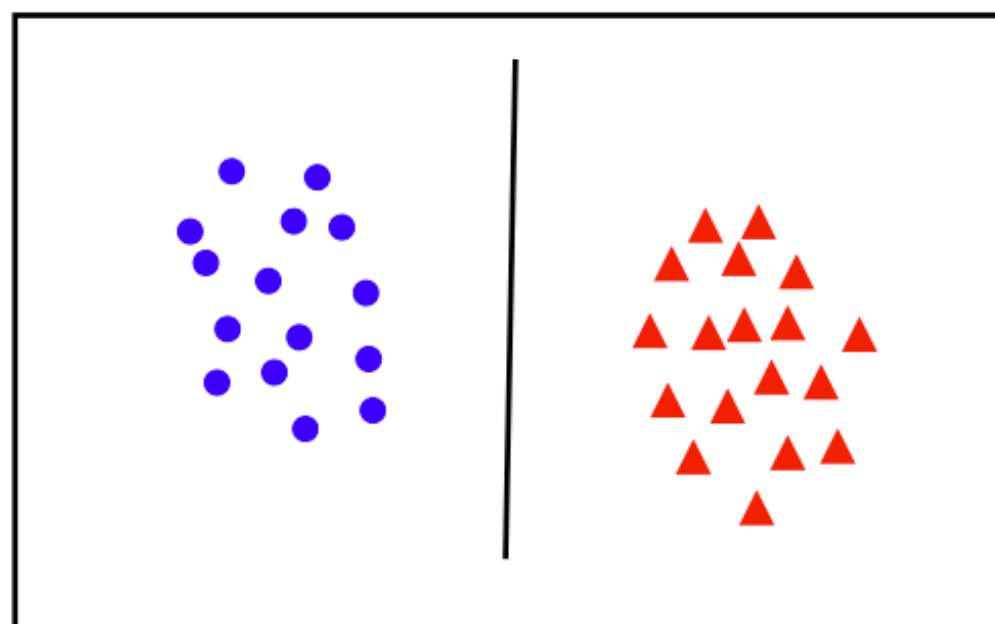
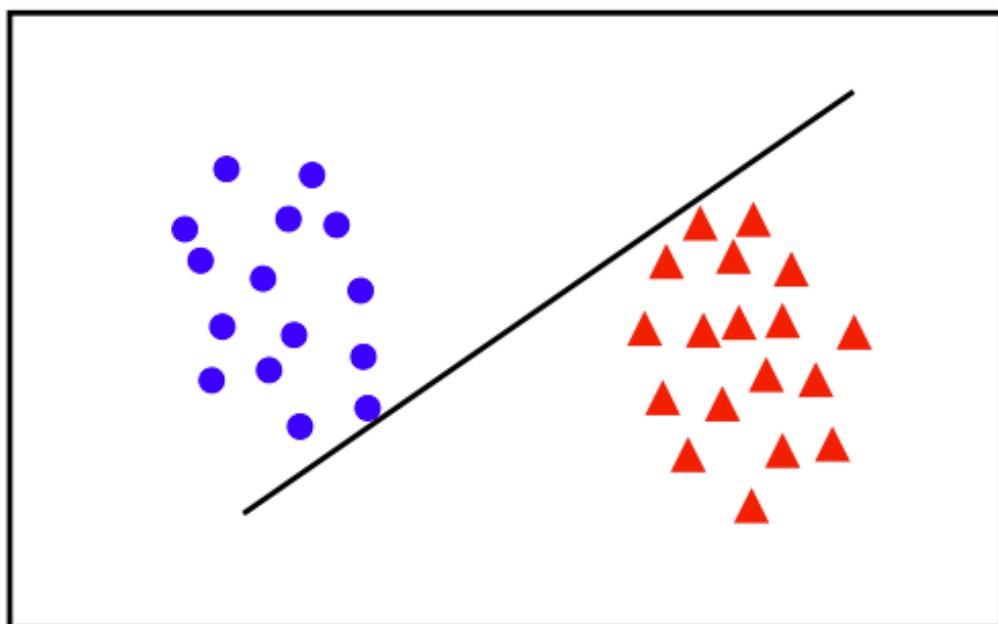
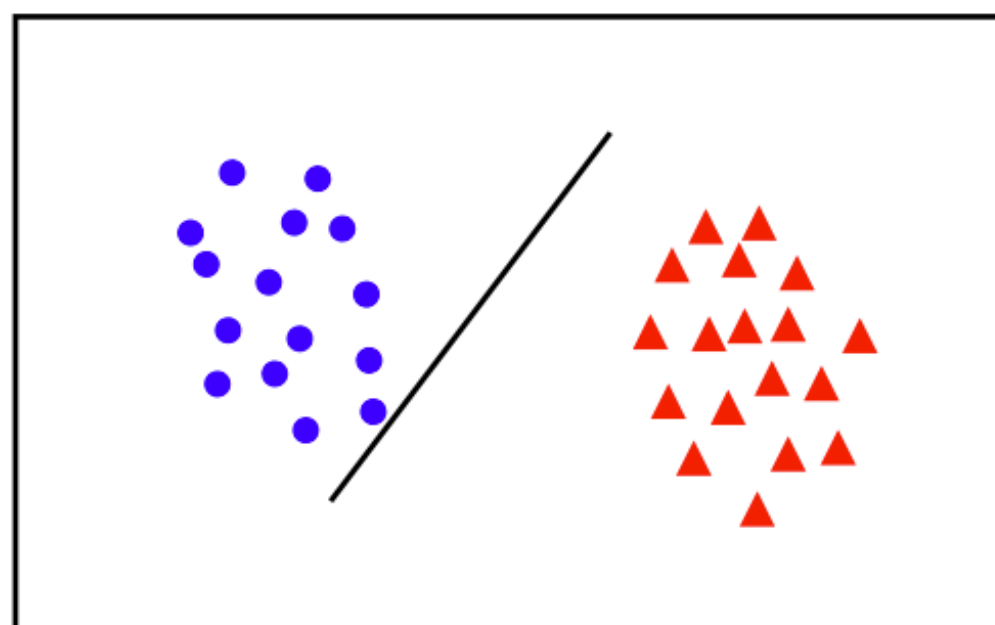
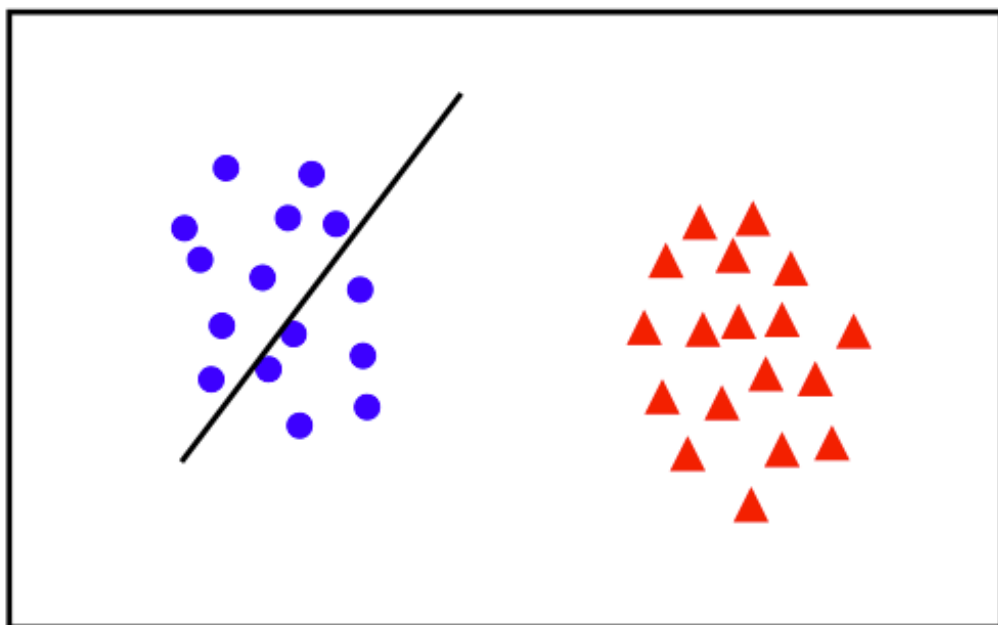
Which line is the best?

All cases, error is zero and they are linear, so they are all good for generalization.

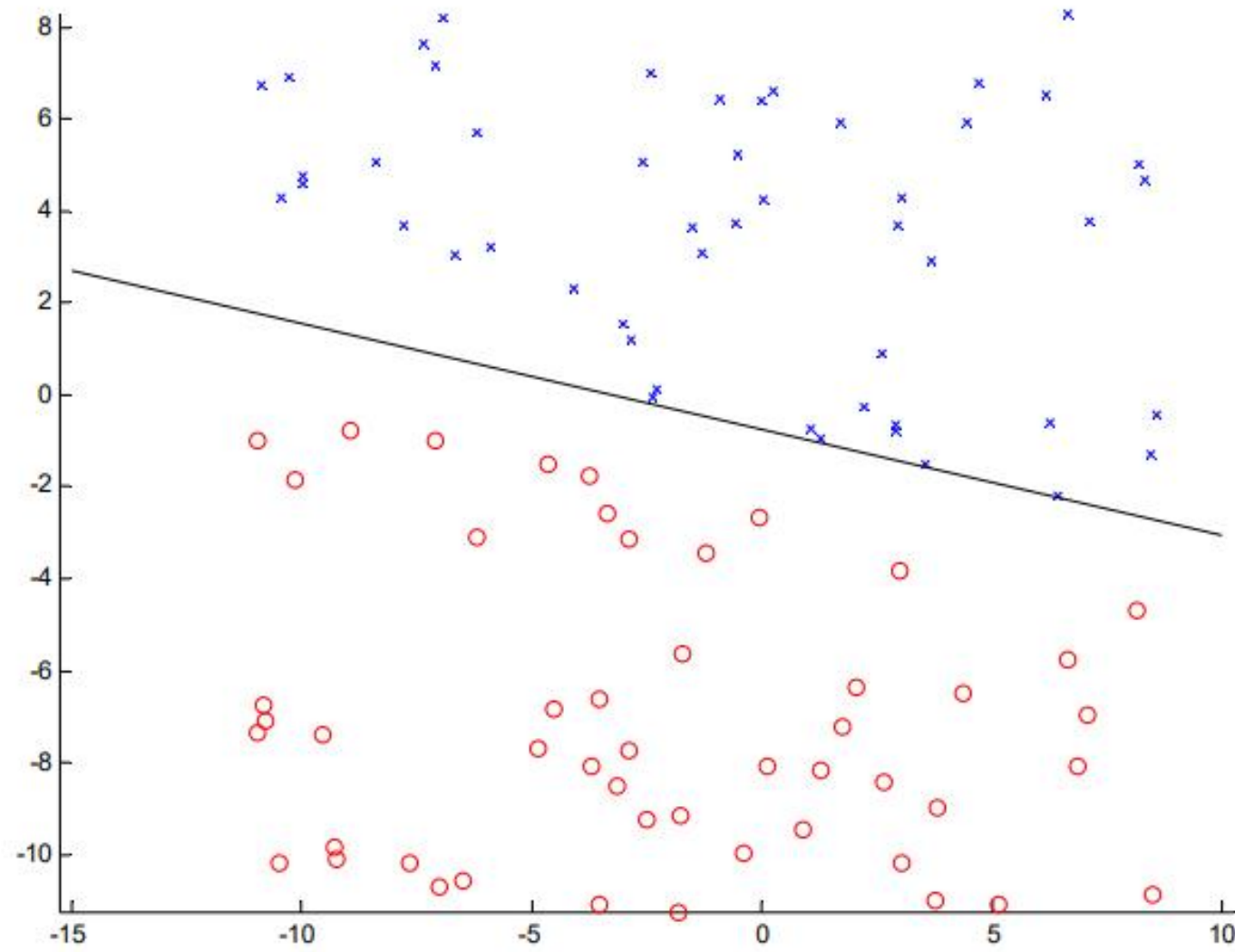
Why is the bigger margin better?

What θ maximizes the margin?

What is the Best θ ?



Perceptron example



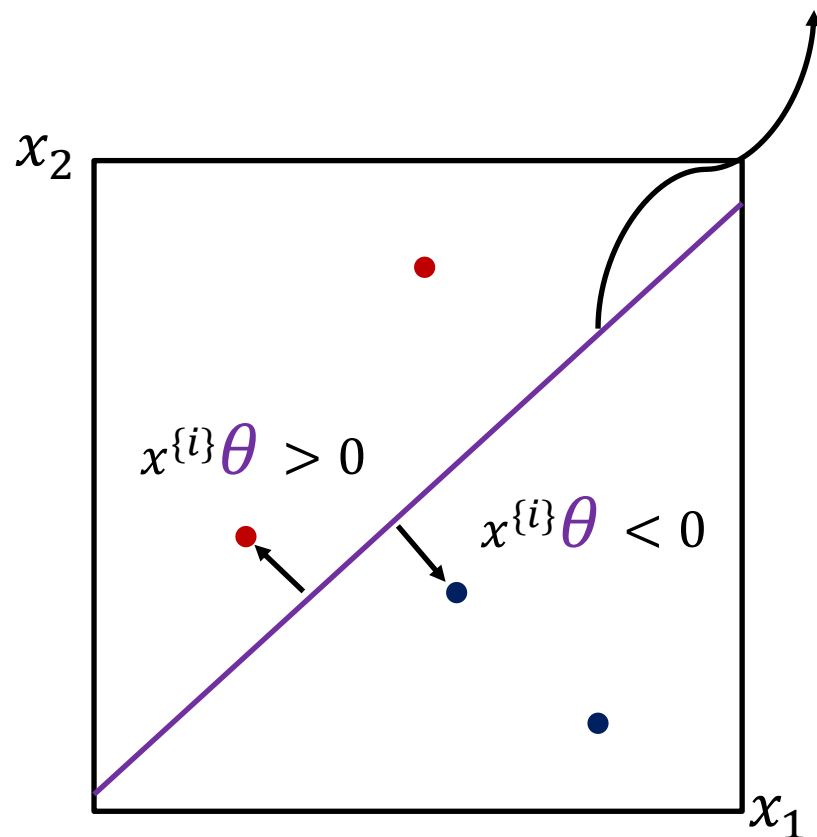
- if the data is linearly separable, then the algorithm will converge
- convergence can be slow ...
- separating line close to training data
- we would prefer a larger margin for generalization (better generalization)

Outline

- Precursor: Linear Classifier and Perceptron
- Support Vector Machine 
- Parameter Learning 

Finding θ with a **fat** margin

Solution (decision boundary) of the line: $x\theta = 0$



Let $x^{(i)}$ to be the nearest data point to the line (plane):

$$|x^{(i)}\theta| > 0$$

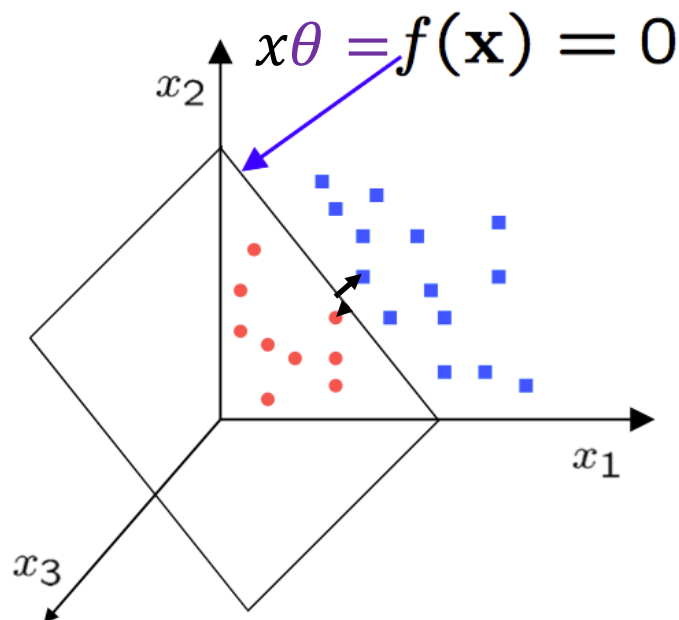
Our line solution is $x\theta = 0$

Does it matter if I scale up or down θ for the decision boundary?

$$|x^{(i)}\theta| = 1 \rightarrow \text{normalization}$$

Let's pull out θ_0 from $\theta = (\theta_1, \dots, \theta_d)$ and call it be b

Decision boundary would be: $x\theta + b = 0$



Computing the distance

The distance between $x^{\{i\}}$ and the line $x\theta + b = 0$ where $|x^{\{i\}}\theta + b| = 1$

The vector θ is perpendicular to the decision line.

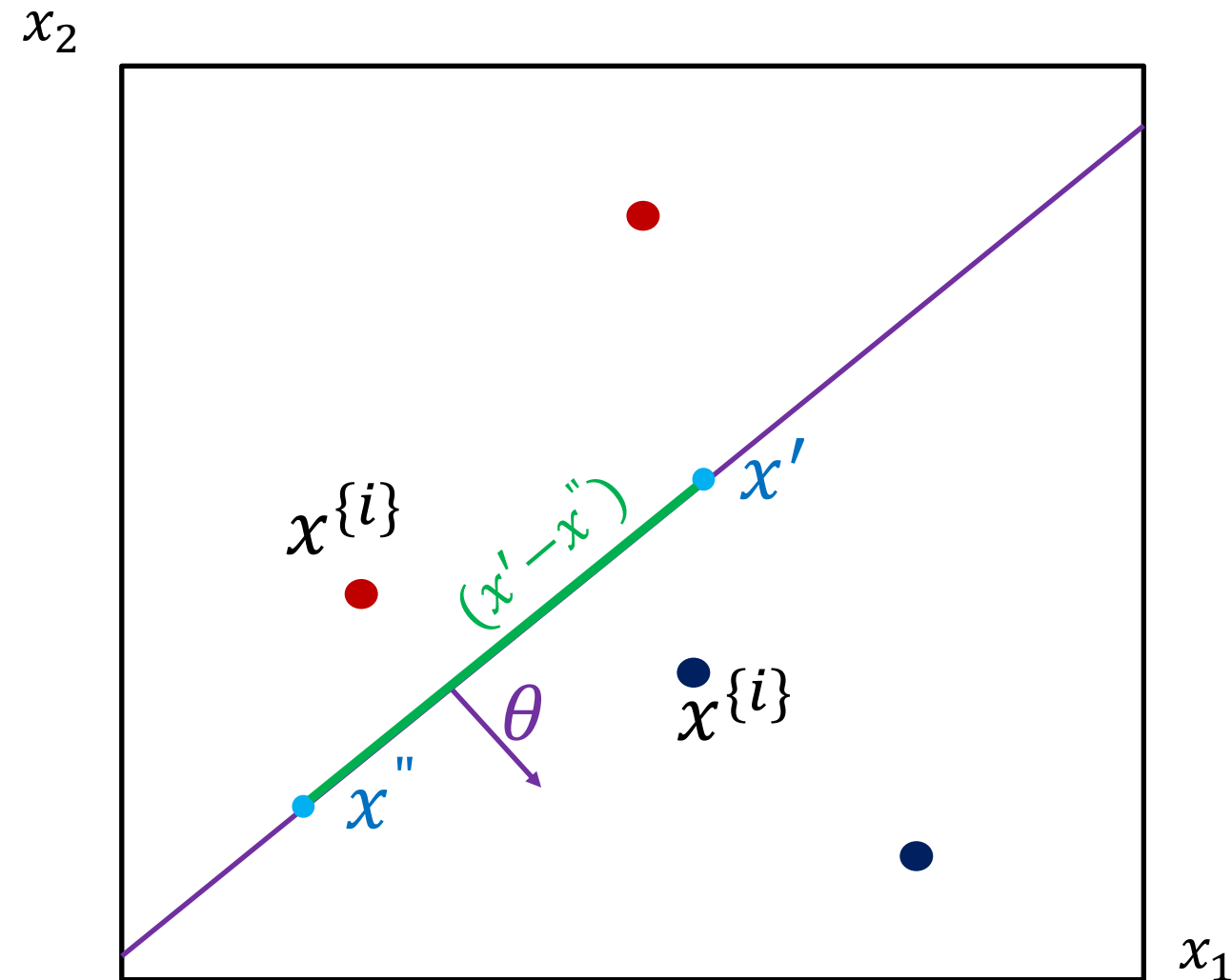
You should ask me why?

Consider x' and x'' on the plane

$$x'\theta + b = 0 \quad \text{and} \quad x''\theta + b = 0$$

$$\Downarrow$$
$$x'\theta + b = x''\theta + b$$

$$\Rightarrow (x' - x'')\theta = 0$$



What is the distance of my fat margin?

What is the distance between $x^{\{i\}}$ and the plane?

Let's take any point x on the line:

Distance would be projection of $(x^{\{i\}} - x)$ vector on θ .

To project the vector, we need to normalize θ to get the unit vector.

$\hat{\theta} = \frac{\theta}{\|\theta\|} \Rightarrow \text{distance} = |(x^{\{i\}} - x) \hat{\theta}|$ which is the dot product

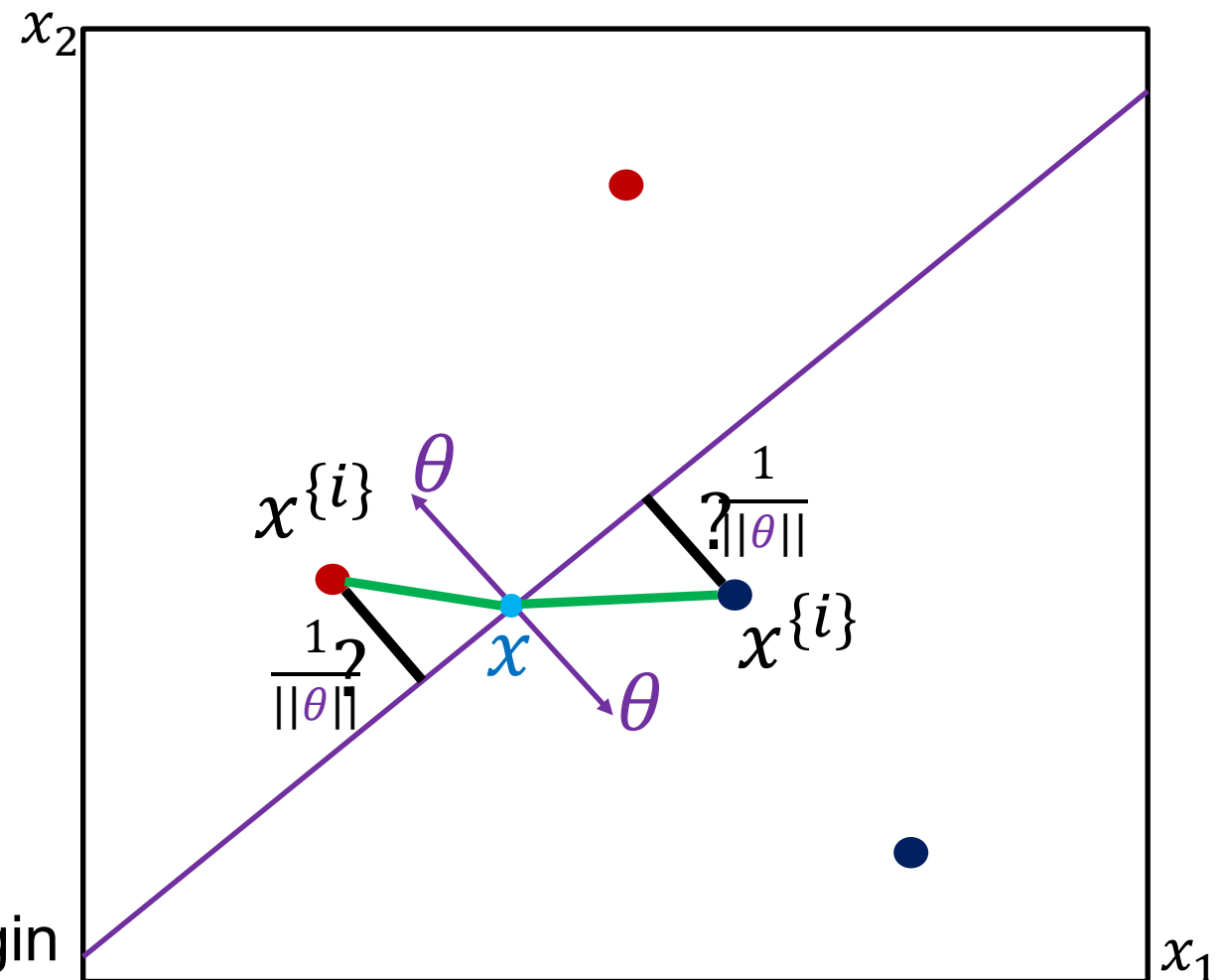
$$\text{distance} = \frac{1}{\|\theta\|} |(x^{\{i\}} \theta - x \theta)|$$

$$= \frac{1}{\|\theta\|} |(x^{\{i\}} \theta + b - x \theta - b)| = \frac{1}{\|\theta\|}$$

My constraint
 $|x^{\{i\}} \theta + b| = 1$

A point on the
 decision line
 $x \theta + b = 0$

The margin



Now we need to maximize the margin

Maximize $\frac{1}{||\theta||}$

Subject to Min value of $|x^{\{i\}}\theta + b| = 1 \Rightarrow \text{nearest neighbour}$
 $i = 1, 2, \dots, N$

There is a “min” in our constraining; it can be hard to optimize this problem(non-convex form)

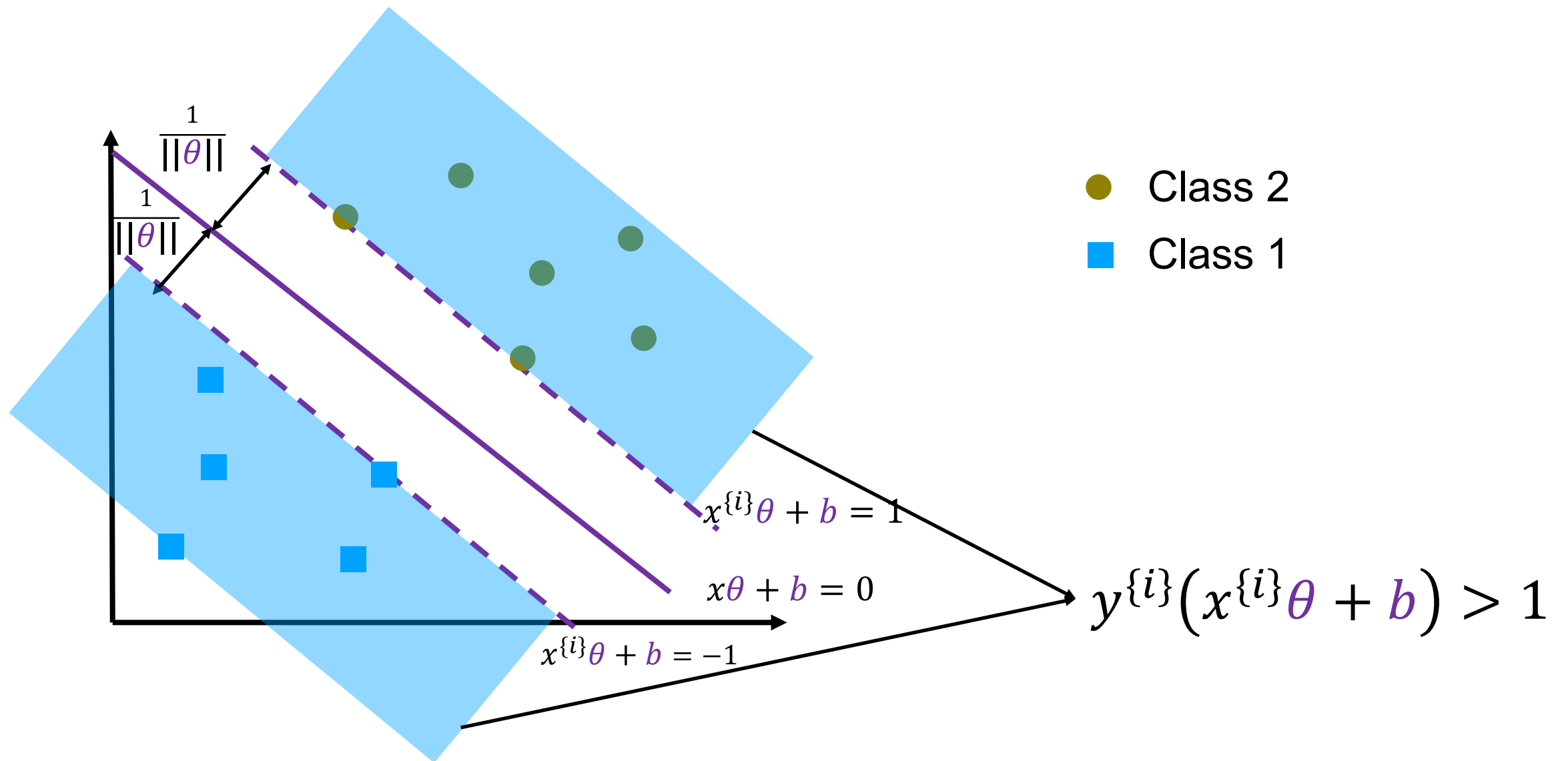
Can I write the following term to get rid of absolute value?

$$|x^{\{i\}}\theta + b| = y^{\{i\}}(x^{\{i\}}\theta + b) \Rightarrow \text{for a correct classification}$$

$$\text{If min } |x^{\{i\}}\theta + b| = 1 \Rightarrow \text{so it can be at least 1}$$

Maximize $\frac{1}{||\theta||}$

Subject to $y^{\{i\}}(x^{\{i\}}\theta + b) \geq 1$ for $i = 1, 2, \dots, N$



Maximize $\frac{2}{\|\theta\|}$

If $\theta \neq 0$, there exists a max value

Subject to $y^{\{i\}}(x^{\{i\}}\theta + b) \geq 1$ for $i = 1, 2, \dots, N$

Minimize $\frac{1}{2}\theta\theta^T$

Subject to $y^{\{i\}}(x^{\{i\}}\theta + b) \geq 1$ for $i = 1, 2, \dots, N$

Constrained optimization

$$\text{Minimize } \frac{1}{2} \theta \theta^T$$

$$\text{Subject to } y^{\{i\}}(x^{\{i\}}\theta + b) \geq 1 \text{ for } i = 1, 2, \dots, N$$

$$\theta \in \mathbb{R}^d, b \in \mathbb{R}$$

Using Lagrange method: But wait, there is an **inequality** in our constraints

We use Karush-Kuhn-Tucker (KKT) condition to deal with this problem

$$g(x) = y^{\{i\}}(x^{\{i\}}\theta + b) - 1 \quad \alpha = \text{lagrange multiplier}$$

We need to optimize

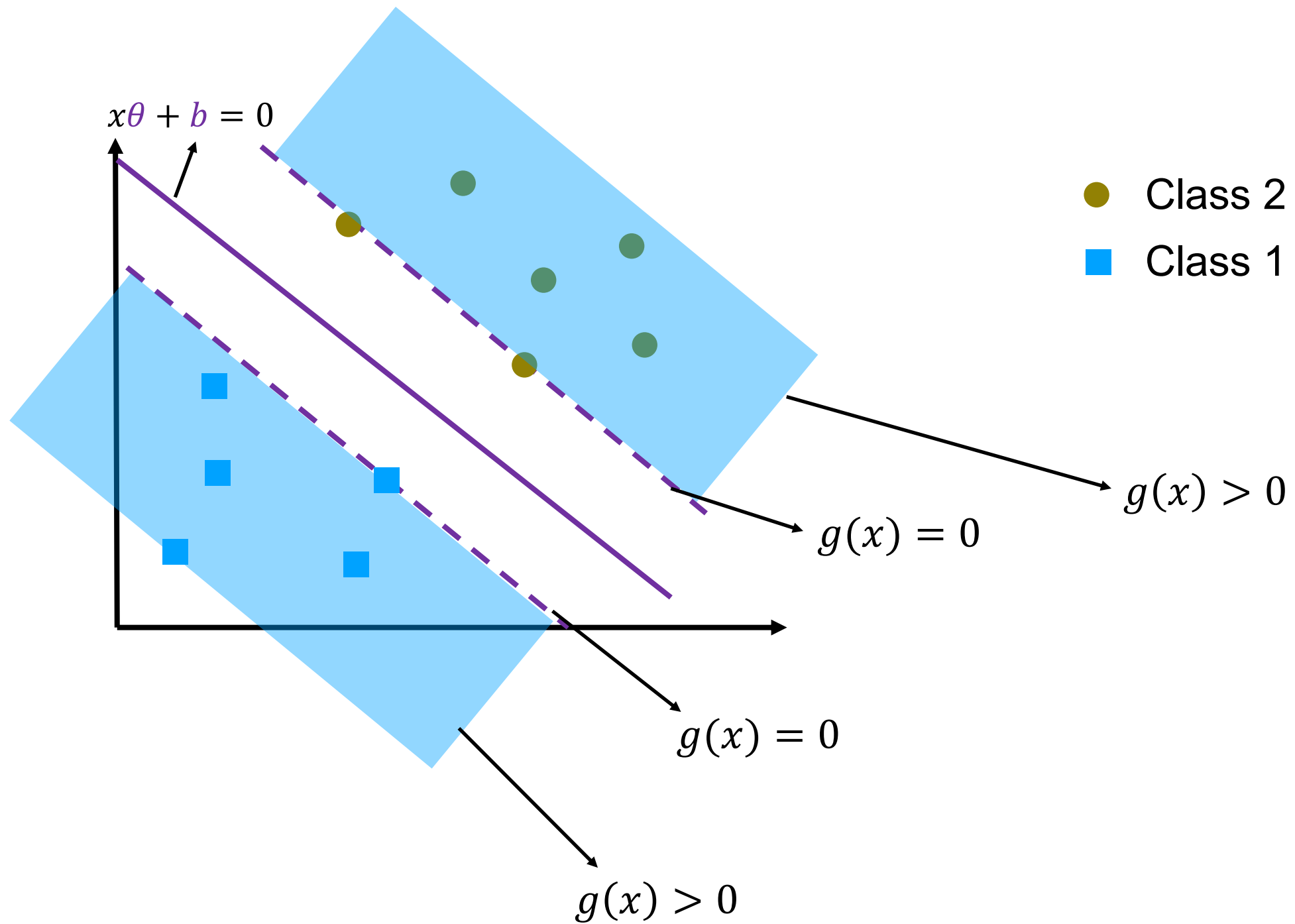
1) Stationary

θ, b , and α

2) $g(x) \geq 0$ Primal feasibility

3) $\alpha \geq 0$ Dual feasibility

4) $g(x)\alpha = 0$ Complementary slackness $\Rightarrow \begin{cases} g(x) > 0, & \alpha = 0 \\ \alpha > 0, & g(x) = 0 \end{cases}$



$$g(x) = y^{\{i\}}(x^{\{i\}}\theta + b) - 1$$

$$3) \quad g(x)\alpha = 0 \quad \text{Complementary slackness} \Rightarrow \begin{cases} g(x) > 0, & \alpha = 0 \\ \alpha > 0, & g(x) = 0 \end{cases}$$

Lagrange formulation

$$\text{Minimize} \quad \frac{1}{2} \theta \theta^T \quad \text{s.t.} \quad y^{\{i\}} (x^{\{i\}} \theta + b) - 1 \geq 0$$

$$\mathcal{L}(\theta, b, \alpha) = \frac{1}{2} \theta \theta^T - \sum_{i=1}^N \alpha^{\{i\}} (y^{\{i\}} (x^{\{i\}} \theta + b) - 1)$$

Minimize w.r.t θ and b and maximize w.r.t each $\alpha^{\{i\}} \geq 0$

$$\nabla_{\theta} \mathcal{L}(\theta, b, \alpha) = \theta - \sum_{i=1}^N \alpha^{\{i\}} y^{\{i\}} x^{\{i\}} = 0$$

$$\nabla_b \mathcal{L}(\theta, b, \alpha) = - \sum_{i=1}^N \alpha^{\{i\}} y^{\{i\}} = 0$$

$$\theta = \sum_{i=1}^N \alpha^{\{i\}} y^{\{i\}} x^{\{i\}}$$

$$\sum_{i=1}^N \alpha^{\{i\}} y^{\{i\}} = 0$$

Let's substitute these in the Lagrangian:

$$\mathcal{L}(\theta, b, \alpha) = \frac{1}{2} \theta \theta^T - \sum_{i=1}^N \alpha^{\{i\}} (y^{\{i\}} (x^{\{i\}} \theta + b) \cancel{-1})$$

$$\mathcal{L}(\theta, b, \alpha) = \sum_{i=1}^N \alpha^{\{i\}} + \frac{1}{2} \theta \theta^T - \sum_{i=1}^N \alpha^{\{i\}} (y^{\{i\}} (x^{\{i\}} \theta + \cancel{b}))$$

$$\begin{aligned} \mathcal{L}(\theta, b, \alpha) &= \sum_{i=1}^N \alpha^{\{i\}} + \frac{1}{2} \theta \theta^T - \sum_{i=1}^N \alpha^{\{i\}} (y^{\{i\}} (x^{\{i\}} \theta)) \cancel{- \sum_{i=1}^N \alpha^{\{i\}} b} + \frac{1}{2} \theta \theta^T - \theta \theta^T = \\ &= \sum_{i=1}^N \alpha^{\{i\}} - \frac{1}{2} \theta \theta^T \end{aligned}$$

$$\theta = \sum_{i=1}^N \alpha^{\{i\}} y^{\{i\}} x^{\{i\}}$$

$$\sum_{i=1}^N \alpha^{\{i\}} y^{\{i\}} = 0$$

$$\mathcal{L}(\theta, b, \alpha) = \sum_{i=1}^N \alpha^{\{i\}} - \frac{1}{2} \theta \theta^T$$

$$\mathcal{L}(\theta, b, \alpha) = \sum_{i=1}^N \alpha^{\{i\}} - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N y^{\{i\}} y^{\{j\}} \alpha^{\{i\}} \alpha^{\{j\}} x^{\{i\}} x^{\{j\}^T}$$

maximize *w.r.t* each $\alpha^{\{i\}} \geq 0$ for $i = 1, \dots, N$

and

$$\sum_{i=1}^N \alpha^{\{i\}} y^{\{i\}} = 0$$

The solution – quadratic programming

$$\max_{\alpha} \sum_{i=1}^N \alpha^{\{i\}} - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N y^{\{i\}} y^{\{j\}} \alpha^{\{i\}} \alpha^{\{j\}} x^{\{i\}} x^{\{j\}T}$$

Quadratic programming packages usually use “min”

$$\min_{\alpha} \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N y^{\{i\}} y^{\{j\}} \alpha^{\{i\}} \alpha^{\{j\}} x^{\{i\}} x^{\{j\}T} - \sum_{i=1}^N \alpha^{\{i\}}$$

$$\min_{\alpha} \frac{1}{2} \alpha^T \begin{bmatrix} y^{\{1\}} y^{\{1\}} x^{\{1\}} x^{\{1\}T} & \dots & y^{\{1\}} y^{\{N\}} x^{\{1\}} x^{\{N\}T} \\ \vdots & \ddots & \vdots \\ y^{\{N\}} y^{\{1\}} x^{\{N\}} x^{\{1\}T} & \dots & y^{\{N\}} y^{\{N\}} x^{\{N\}} x^{\{N\}T} \end{bmatrix} \alpha + (-I^T) \alpha$$

$$\min_{\alpha} \frac{1}{2} \alpha^T \begin{bmatrix} y^{\{1\}} y^{\{1\}} x^{\{1\}} x^{\{1\}T} & \dots & y^{\{1\}} y^{\{N\}} x^{\{1\}} x^{\{N\}T} \\ \vdots & \ddots & \vdots \\ y^{\{N\}} y^{\{1\}} x^{\{N\}} x^{\{1\}T} & \dots & y^{\{N\}} y^{\{N\}} x^{\{N\}} x^{\{N\}T} \end{bmatrix} \underbrace{\alpha + (-I^T) \alpha}_{\text{Linear term}}$$

Quadratic coefficients

Subject to $\sum_{i=1}^N \underbrace{\alpha^{\{i\}} y^{\{i\}}}_{\text{Linear equality constraint}} = y^T \alpha = 0$

Linear equality constraint

Pass these to a
quadratic programming
package

$$\text{lower bound}(0) \leq \alpha \leq \text{upper bound}(\infty)$$

$$\min_{\alpha} \frac{1}{2} \alpha^T Q \alpha - 1^T \alpha \quad \text{subject to} \quad y^T \alpha = 0; \alpha \geq 0$$

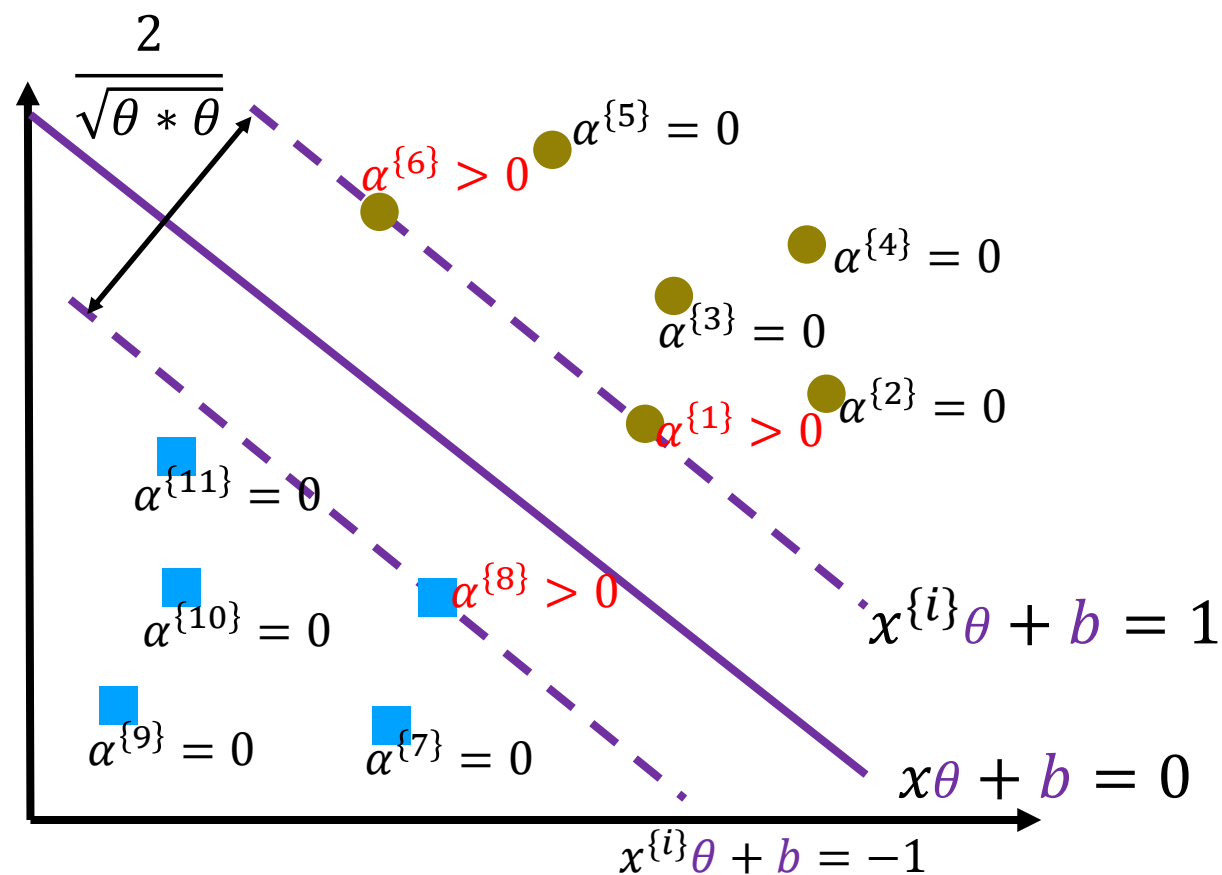
Quadratic programming will give us α

$$\text{Solution: } \alpha = \alpha^{\{1\}}, \dots, \alpha^{\{N\}}$$

$$\text{KKT condition } (\alpha_i g_i(\theta) = 0): \quad \alpha^{\{i\}} (y^{\{i\}} (x^{\{i\}} \theta + b) - 1) = 0$$

$$(y^{\{i\}} (x^{\{i\}} \theta + b) - 1) > 0 \quad \Rightarrow \quad \alpha^{\{i\}} = 0$$

$$(y^{\{i\}} (x^{\{i\}} \theta + b) - 1) = 0 \quad \Rightarrow \quad \alpha^{\{i\}} > 0 \Rightarrow x_i \text{ is a support vector}$$



Training

$$\theta = \sum_{i=1}^N \alpha^{(i)} y^{(i)} x^{(i)}$$

No need to go over all datapoints

$$\rightarrow \theta = \sum_{x^{(i)} \text{ in } SV} \alpha^{(i)} y^{(i)} x^{(i)}$$

and for b pick any support vector
and calculate:

$$y^{(i)} (x^{(i)} \theta + b) = 1$$

Testing

For a new test point s

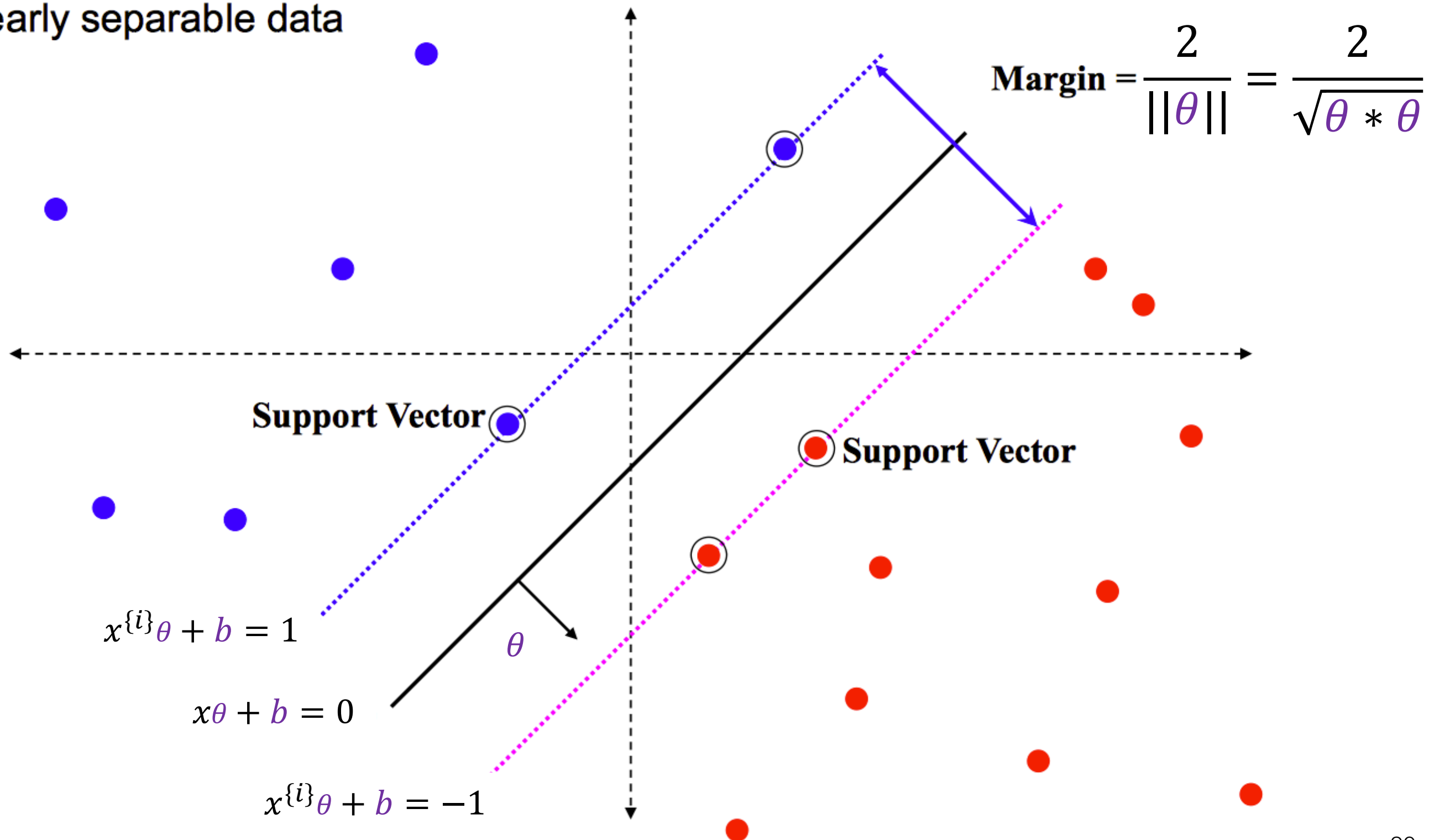
Compute:

$$s\theta + b = \sum_{x^{(i)} \text{ in } SV} \alpha^{(i)} y^{(i)} x^{(i)} s^T + b$$

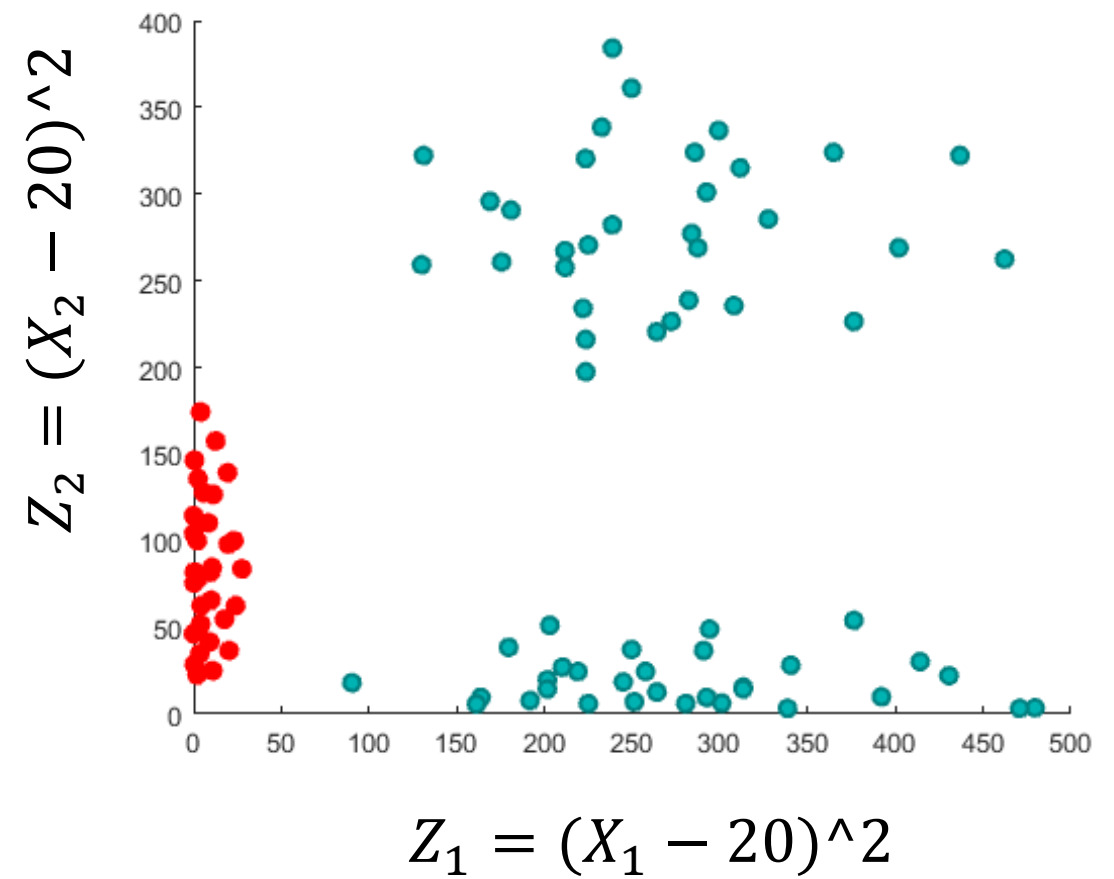
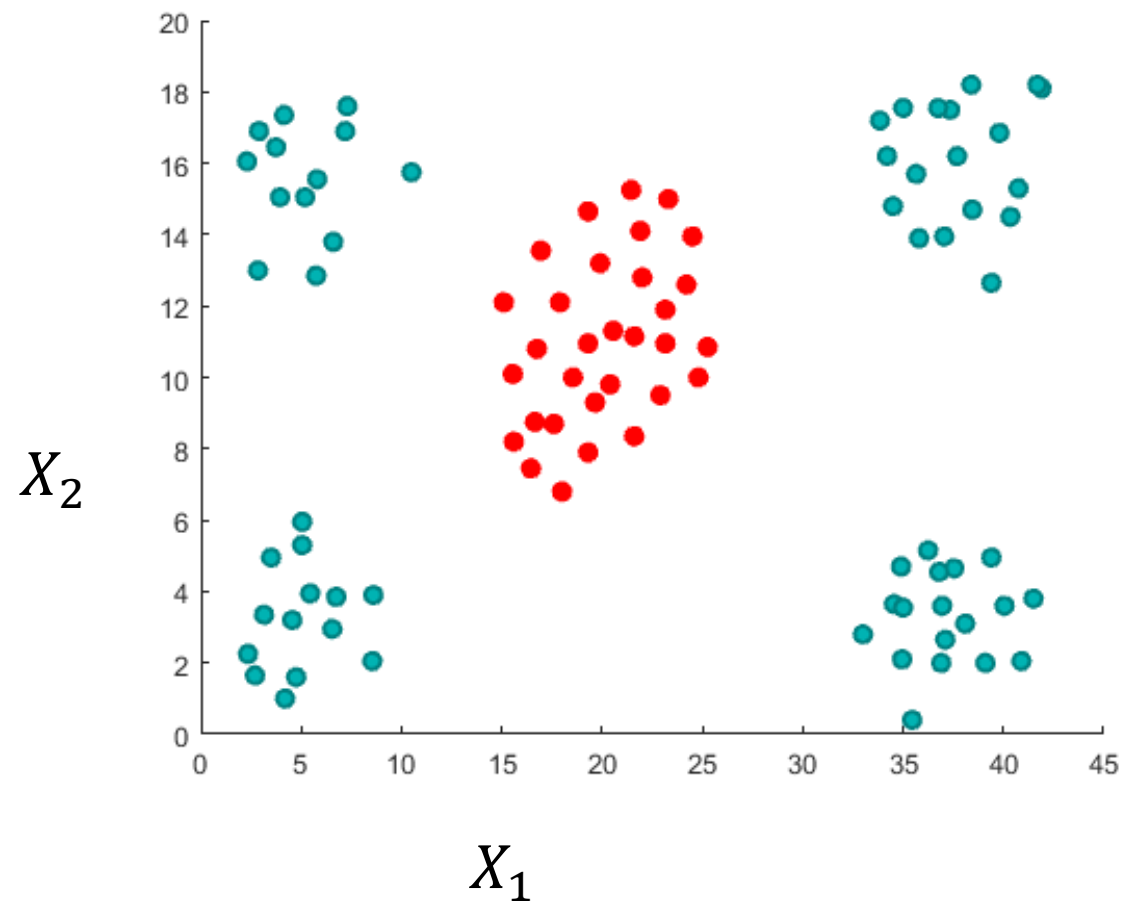
Classify s as class 1 if the result is
positive, and class 2 otherwise

Geometric Interpretation

linearly separable data



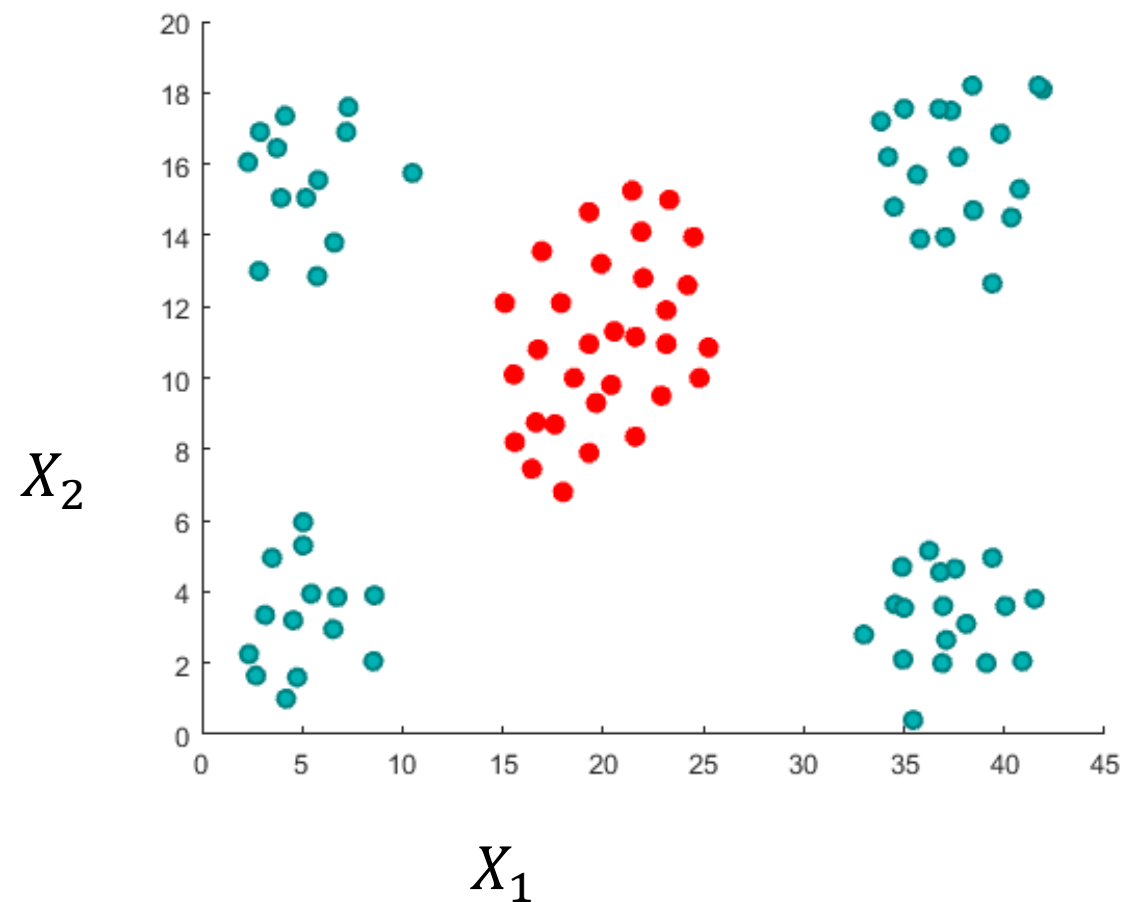
From x to z space



$$X \xrightarrow{(X_1^2, X_2^2)} Z$$

In x space

$$\max_{\alpha} \sum_{i=1}^N \alpha^{\{i\}} - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N y^{\{i\}} y^{\{j\}} \alpha^{\{i\}} \alpha^{\{j\}} x^{\{i\}} x^{\{j\}T}$$

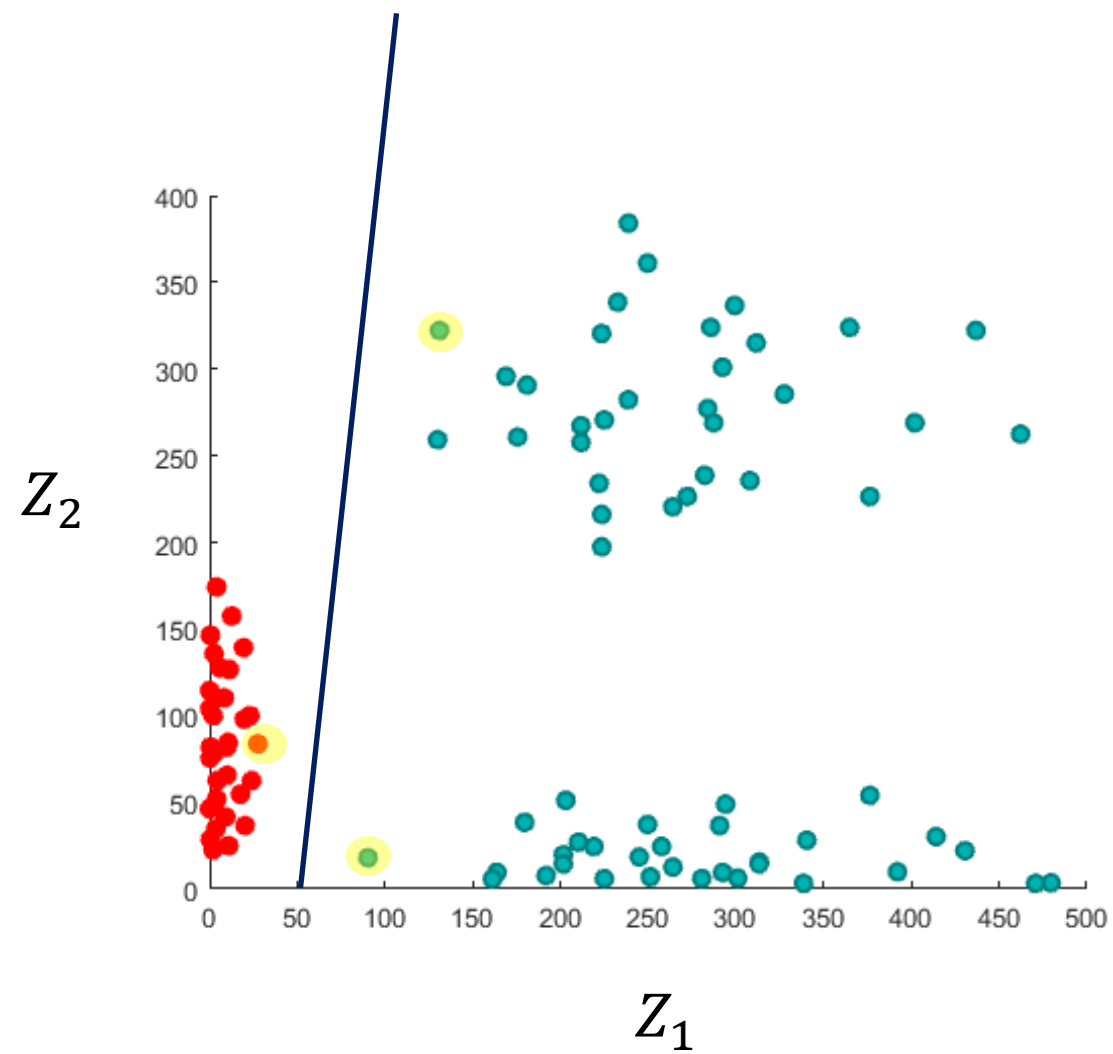


let's say x is $n \times d$
 xx^T will be $n \times n$

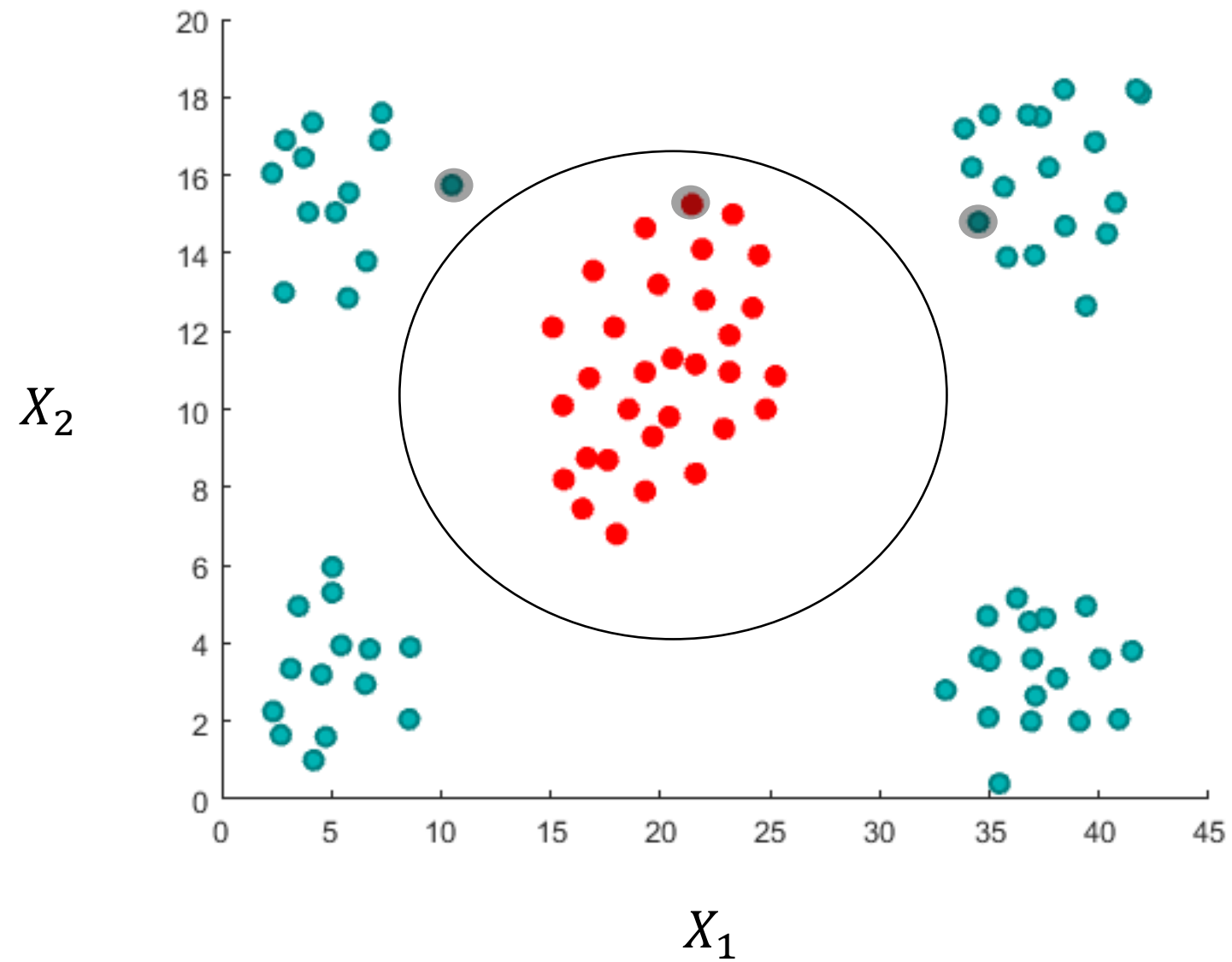
If I add millions of
 dimensions to x , would
 it affect the final size of
 xx^T ?

In z space

$$\max_{\alpha} \sum_{i=1}^N \alpha^{\{i\}} - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N y^{\{i\}} y^{\{j\}} \alpha^{\{i\}} \alpha^{\{j\}} z^{\{i\}} z^{\{j\}T}$$



In x space, they are called pre-images of support vectors



Take-Home Messages

- Linear Separability
- Perceptron
- SVM: Geometric Intuition and Formulation