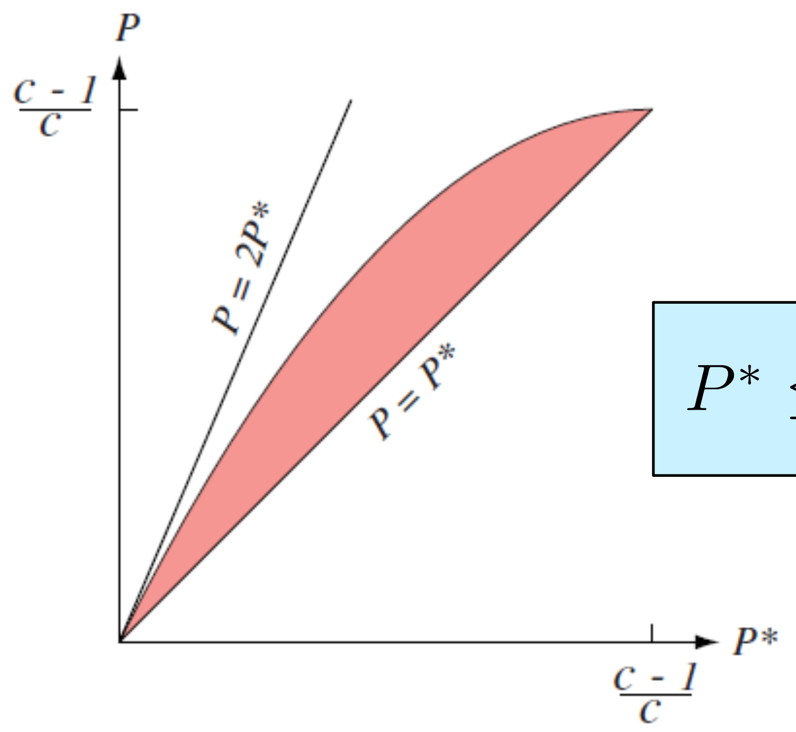
Review of Lecture 16

Nearest Neighbor Classifier

Assign $\mathbf{x}$ to same label as closest training point $\mathbf{x}_i$

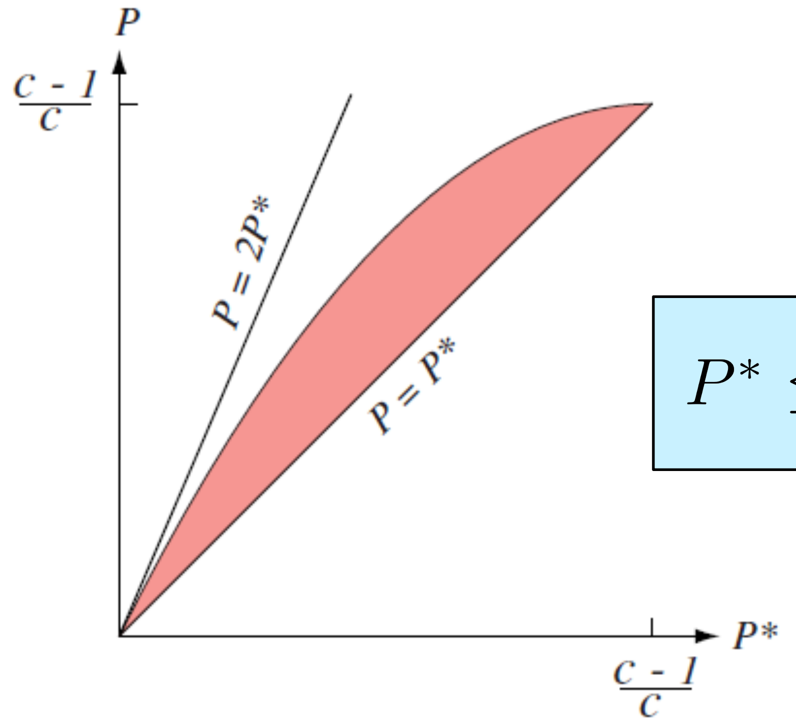


$$P^* \leq \; P \; \leq P^*(2 - \tfrac{c}{c-1}P^*)$$

Review of Lecture 16

Nearest Neighbor Classifier

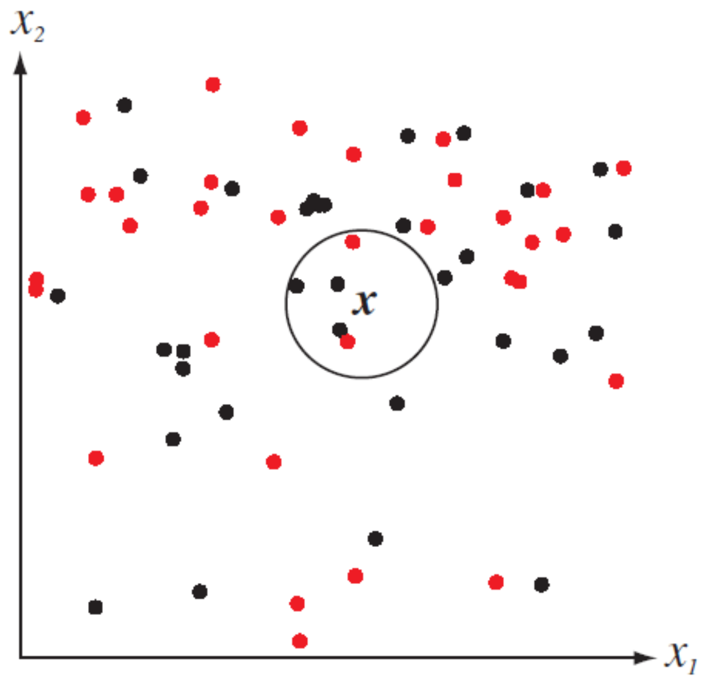
Assign $\mathbf{x}$ to same label as closest training point $\mathbf{x}_i$

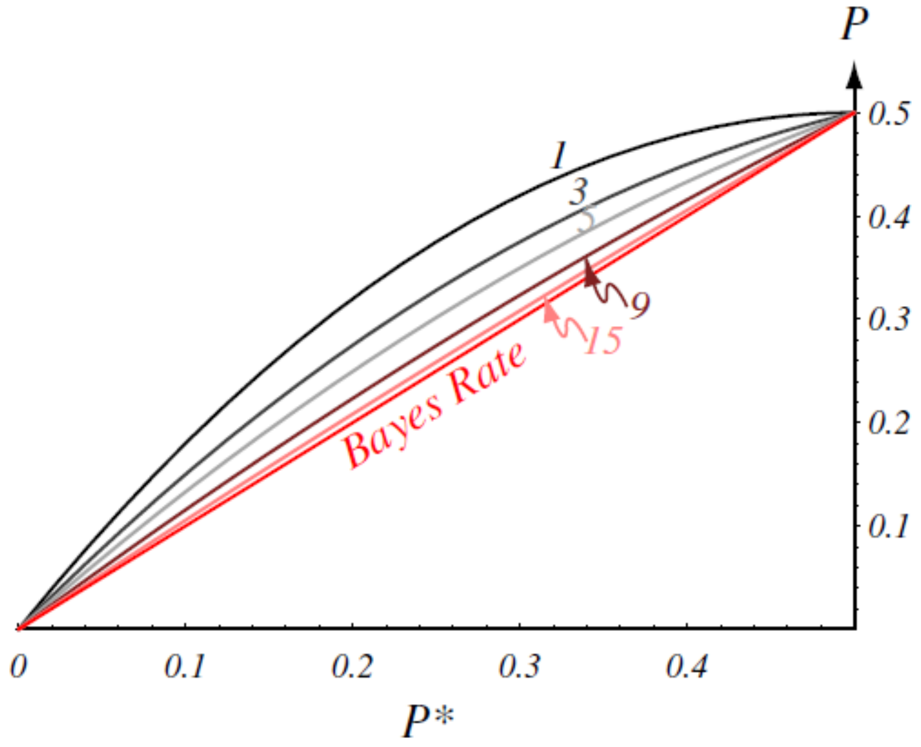


$$P^* \leq P \leq P^*(2 - \tfrac{c}{c-1}P^*)$$

K-Nearest Neighbor Classifier

Assign label of $\mathbf{x}$ by taking majority vote over *K* nearest neighbors





Given enough data, *K*-NN classifier will perform as well as any classifier

Catch

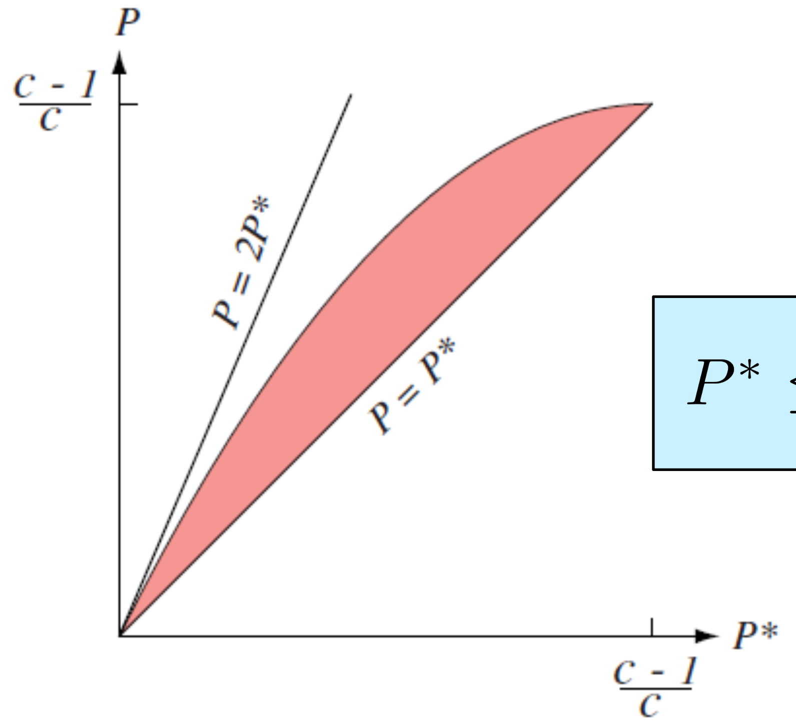
Huge amount of data, especially if feature space is high-dimensional

K-NN has slow inference vs. (most other classifiers) slow training

Review of Lecture 16

Nearest Neighbor Classifier

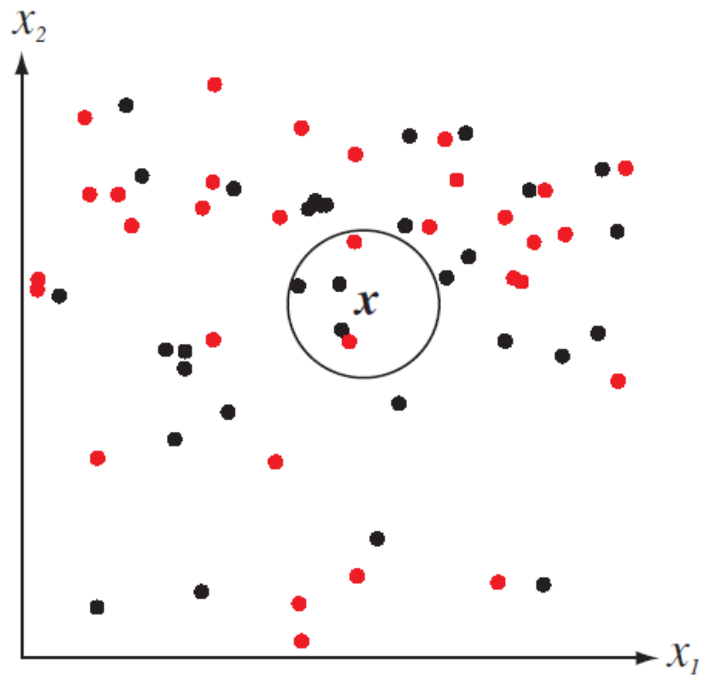
Assign $\mathbf{x}$ to same label as closest training point $\mathbf{x}_i$

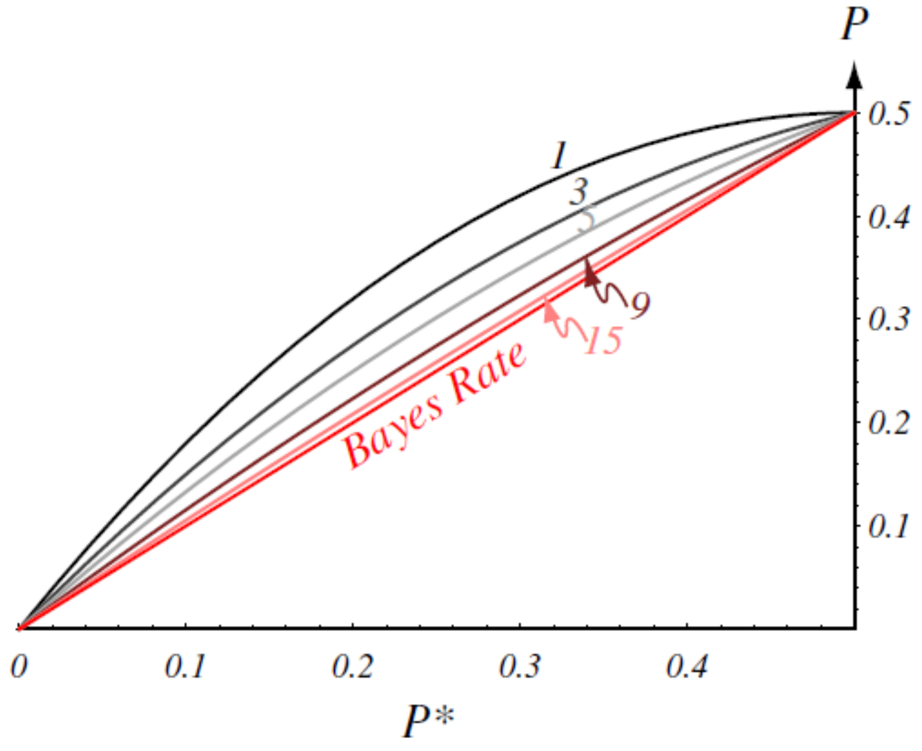


$$P^* \leq \; P \; \leq P^*(2 - \tfrac{c}{c-1}P^*)$$

K-Nearest Neighbor Classifier

Assign label of $\mathbf{x}$ by taking majority vote over *K* nearest neighbors





Given enough data, *K*-NN classifier will perform as well as any classifier

Catch

Huge amount of data, especially if feature space is high-dimensional

K-NN has <u>slow inference</u> vs. (most other classifiers) <u>slow training</u>

Perceptron Learning Algorithm

Given

- training data $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$
- a guess for $\boldsymbol{\theta}^0$

Pick any observations and update
one sample at a time

$$\boldsymbol{\theta}^{j+1} = \begin{cases} \boldsymbol{\theta}^j + y_i \widetilde{\mathbf{x}}_i & \text{if } y_i \neq \text{sign}((\boldsymbol{\theta}^j)^T \widetilde{\mathbf{x}}_i) \\ \boldsymbol{\theta}^j & \text{otherwise} \end{cases}$$

Linearly Separable

there exists $\mathbf{w} \in \mathbb{R}^d$ and $b \in \mathbb{R}$ such that

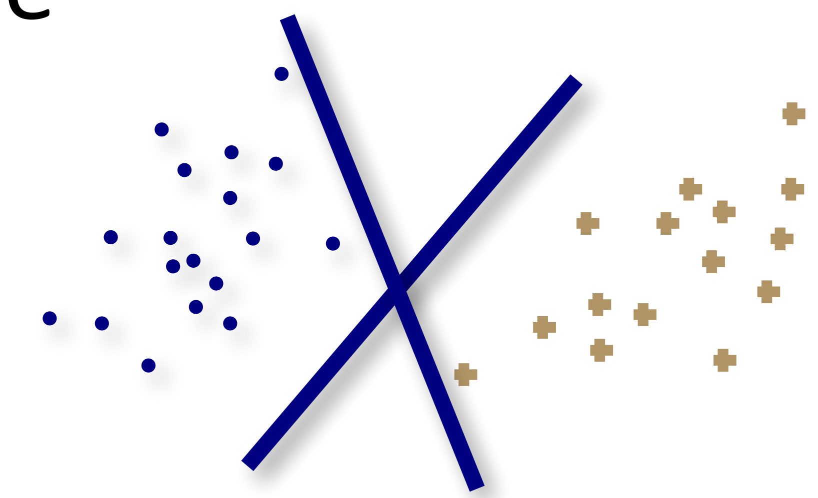
$$y_i = \text{sign}(\mathbf{w}^T \mathbf{x}_i + b)$$

$$\forall i = 1, \ldots, n$$

Maximum Margin separating plane

$$\rho(\mathbf{w}, b) = \min_i \frac{|\mathbf{w}^T \mathbf{x}_i + b|}{\|\mathbf{w}\|_2}$$

$$(\mathbf{w}^*, b^*) = \arg\max_{\mathbf{w}, b} \rho(\mathbf{w}, b)$$



Where are we with SVMs

- Introduced the concept of linear separability and margins

- Deep dive into SVMs (today)

- The kernel trick and Soft-Margin SVMs (next lecture)

ECE 6254
Statistical Machine Learning

Professor: Amirali Aghazadeh
Office: Coda S1209
Georgia Institute of Technology

Lecture 17: Support Vector Machines

Georgia Institute of Technology

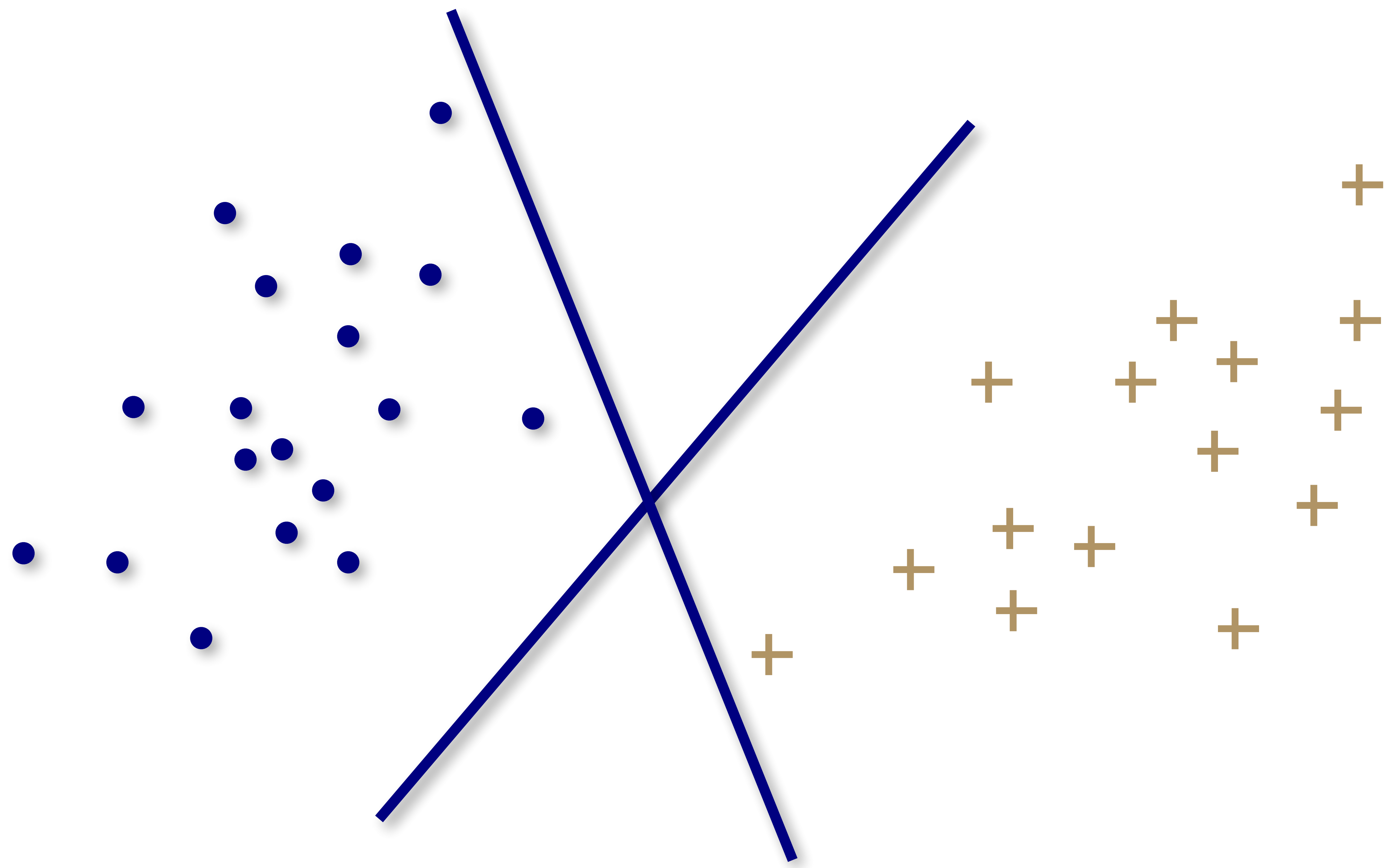
SVMs - Outline

- Maximizing the margin

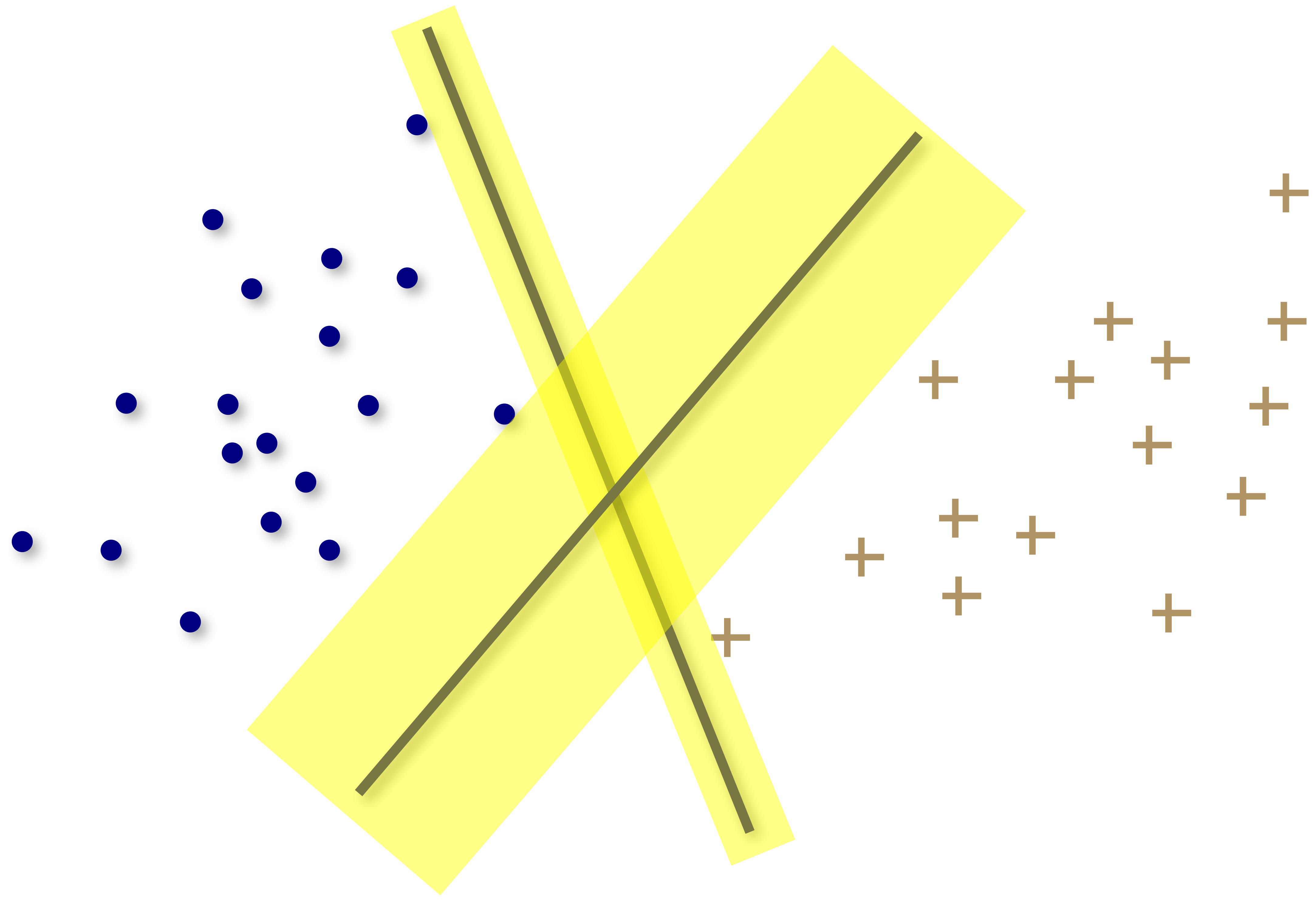
- The solution

- Nonlinear transforms

Are all separating hyperplanes equal?

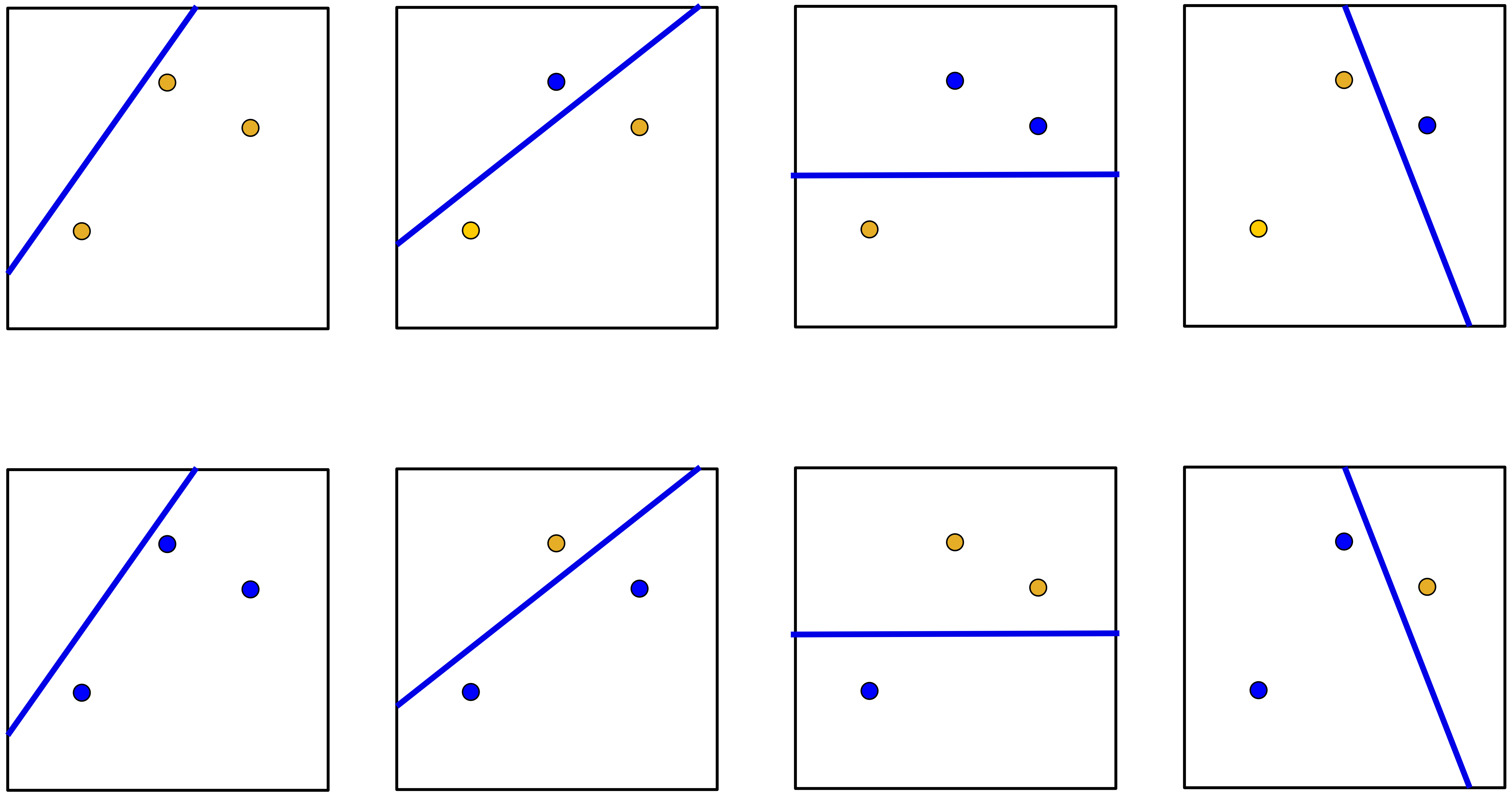


Are all separating hyperplanes equal?



Remember the growth function?

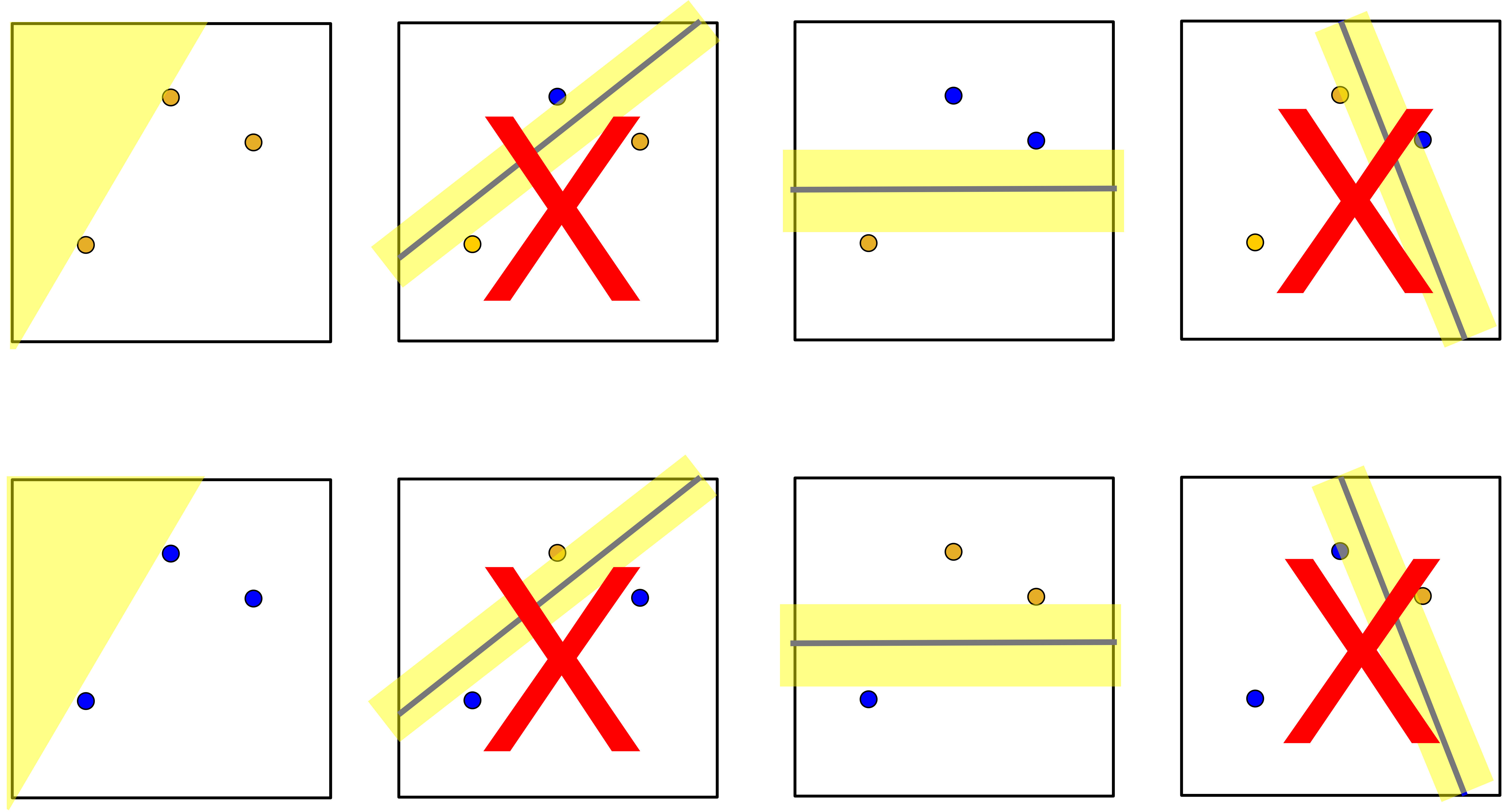
All dichotomies with any line in 2D (PLA)



$$2^3 = 8 \text{ dichotomies}$$

Remember the growth function?

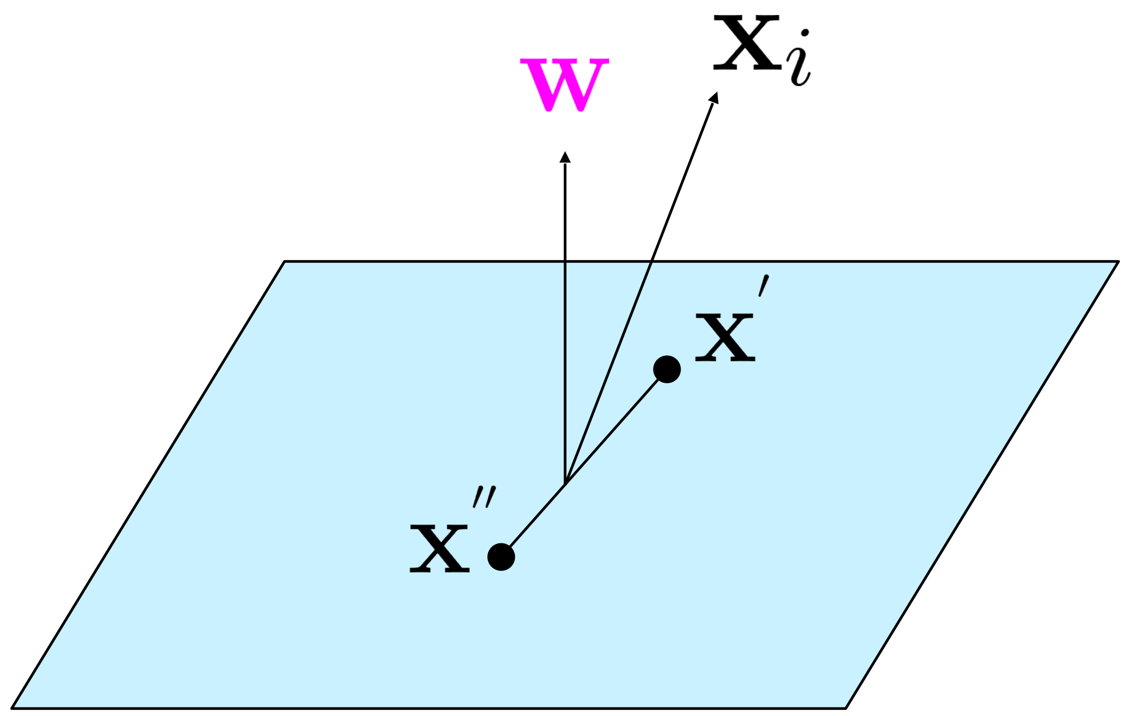
Fat margins imply fewer dichotomies



Finding $\mathbf{w}$ with largest margin

$\mathbf{x}_i$ is nearest point to hyperplane $\mathbf{w}^T\mathbf{x} + b = 0$ and $|\mathbf{w}^T x_i + b| = 1$

What is the distance?



Review:

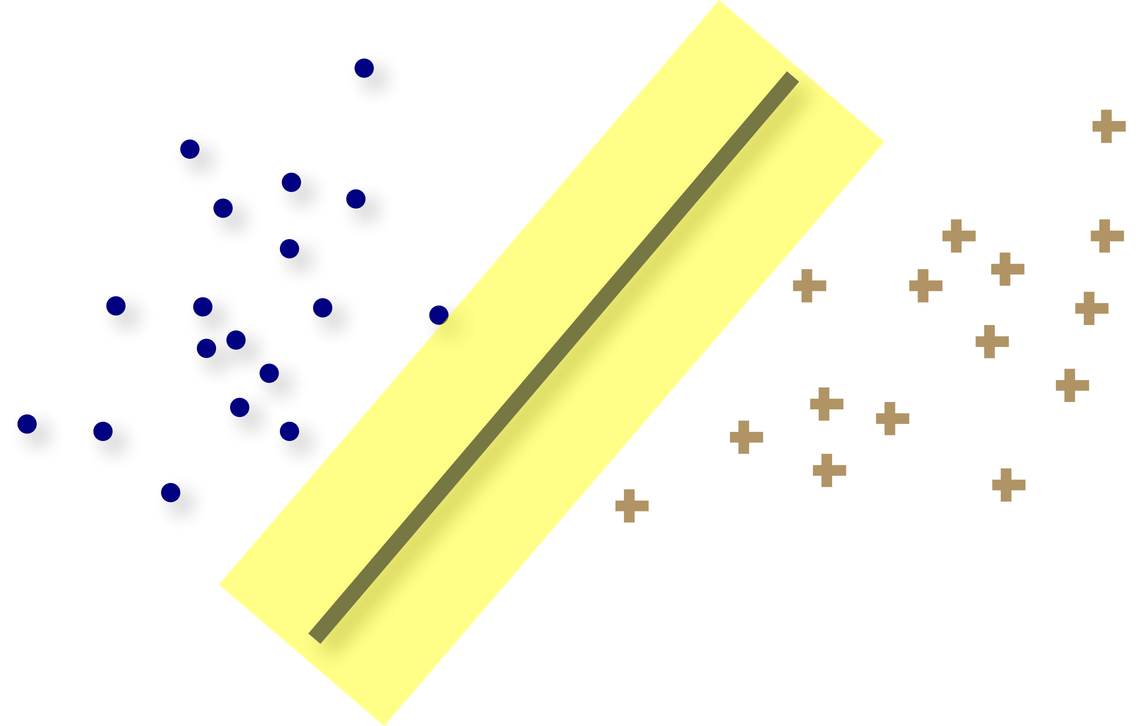
- The vector $\mathbf{w}$ is $\perp$ to the hyperplane:

  Take $\mathbf{x}'$ and $\mathbf{x}''$ on the plane

$$\mathbf{w}^T\mathbf{x}' + b = 0 \text{ and } \mathbf{w}^T\mathbf{x}'' + b = 0$$

$$\longrightarrow \mathbf{w}^T(\mathbf{x}' - \mathbf{x}'') = 0$$

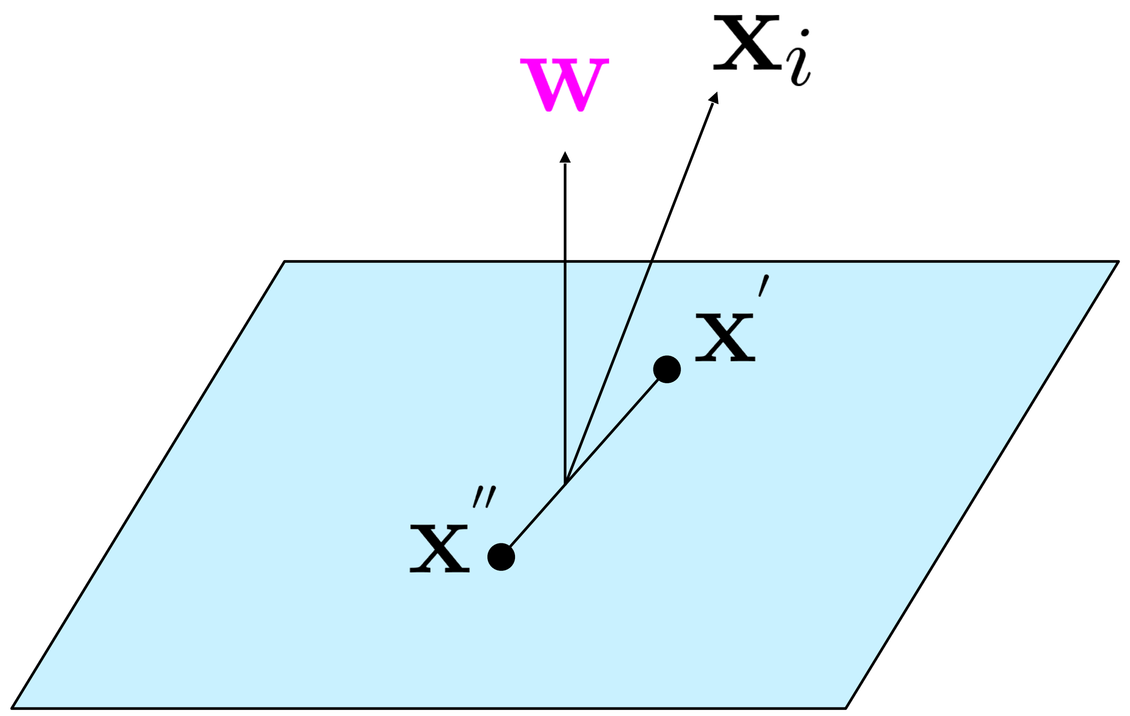
- Larger margin ➡ better generalization to new data



Finding $\mathbf{w}$ with largest margin

$\mathbf{x}_i$ is nearest point to hyperplane $\mathbf{w}^T\mathbf{x} + b = 0$, and $|\mathbf{w}^T x_i + b| = 1$

What is the distance?



Review:

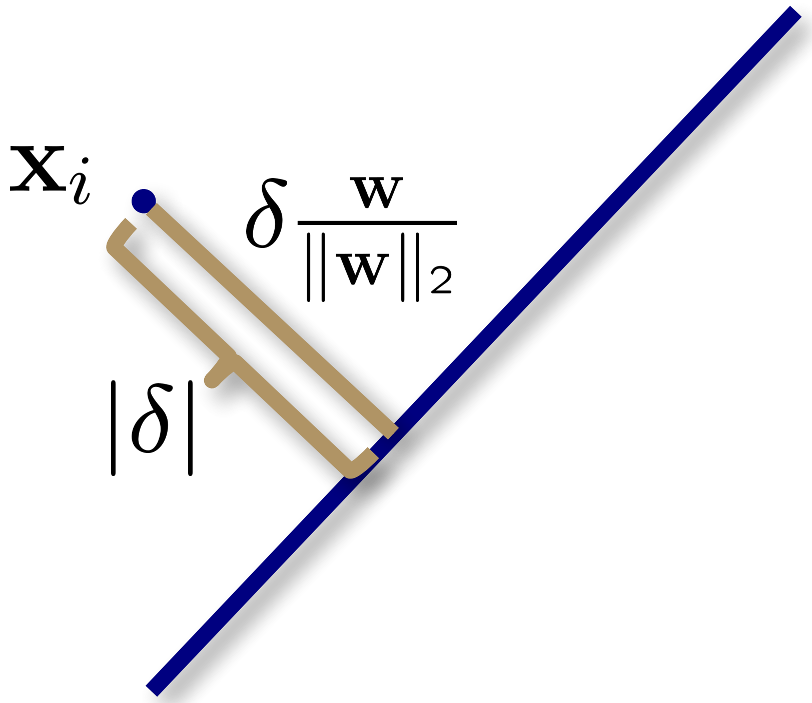
$$|\delta| = \frac{|\mathbf{w}^T\mathbf{x}_i + b|}{\|\mathbf{w}\|_2} = \frac{1}{\|\mathbf{w}\|_2}$$

- The vector $\mathbf{w}$ is $\perp$ to the hyperplane:

  Take $\mathbf{x}'$ and $\mathbf{x}''$ on the plane

  $\mathbf{w}^T\mathbf{x}' + b = 0$ and $\mathbf{w}^T\mathbf{x}'' + b = 0$

  $\longrightarrow \mathbf{w}^T(\mathbf{x}' - \mathbf{x}'') = 0$



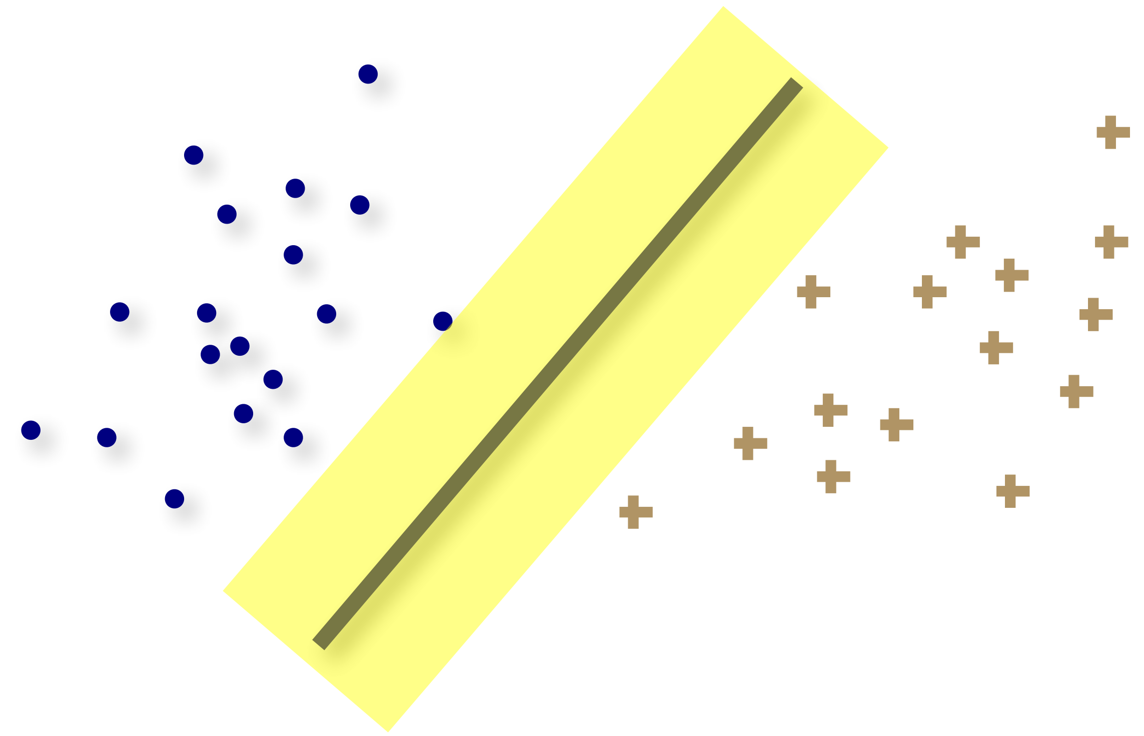
- Larger margin ➡ better generalization to new data

The optimization problem

Maximize $\frac{1}{\|\mathbf{w}\|_2}$

canonical form

subject to $\min_{n=1,2,\dots,N} |\mathbf{w}^T\mathbf{x}_n + b| = 1$



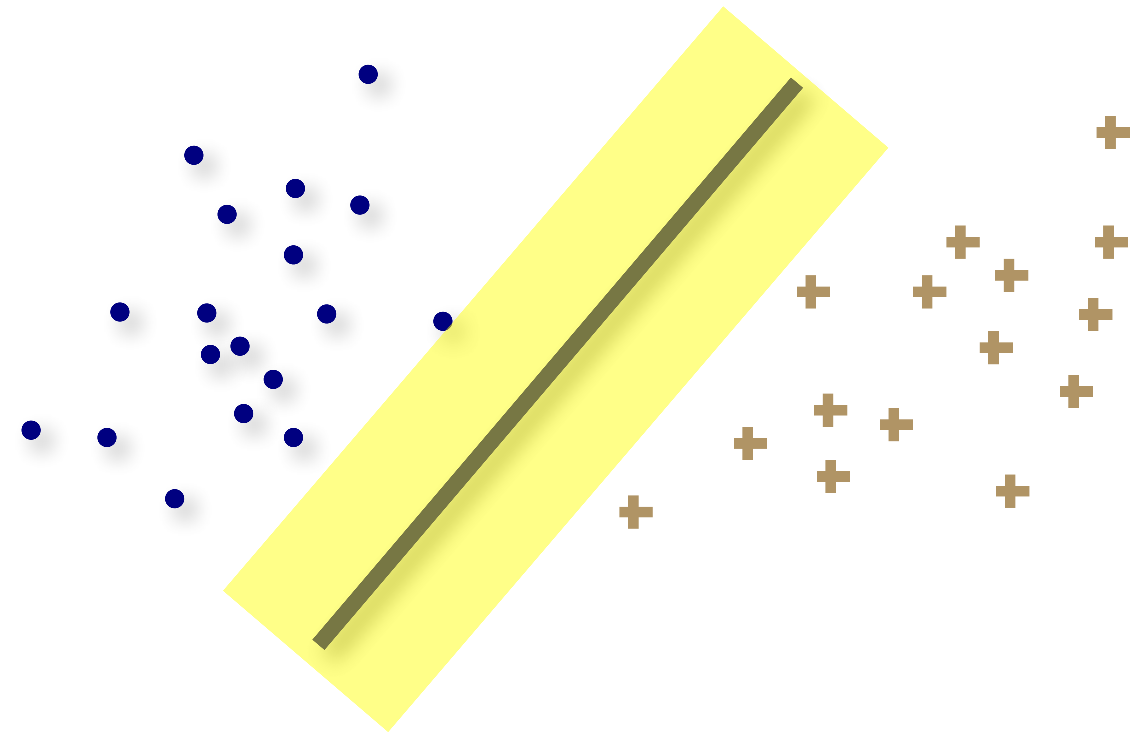
Notice: $|\mathbf{w}^T\mathbf{x}_n + b| = \mathbf{y}_n(\mathbf{w}^T\mathbf{x}_n + b)$

The optimization problem

Maximize $\frac{1}{\|\mathbf{w}\|_2}$

canonical form

subject to $\min_{n=1,2,\ldots,N} |\mathbf{w}^T\mathbf{x}_n + b| = 1$



Notice: $|\mathbf{w}^T\mathbf{x}_n + b| = \mathbf{y}_n(\mathbf{w}^T\mathbf{x}_n + b)$

Minimize $\frac{1}{2}\mathbf{w}^T\mathbf{w}$

subject to $\mathbf{y}_n(\mathbf{w}^T\mathbf{x}_n + b) \geq 1$ for $n = 1, 2, \ldots, N$

SVMs - Outline

- Maximizing the margin
- The solution
- Nonlinear transforms

Constrained optimization

Minimize $= \frac{1}{2}\mathbf{w}^T\mathbf{w}$

subject to $\mathbf{y}_n(\mathbf{w}^T\mathbf{x}_n + b) \geq 1$ for $n = 1, 2, \ldots, N$

$$\mathbf{w} \in \mathbb{R}^d, \; b \in \mathbb{R}$$

Lagrange? inequality constraints $\longrightarrow$ KKT

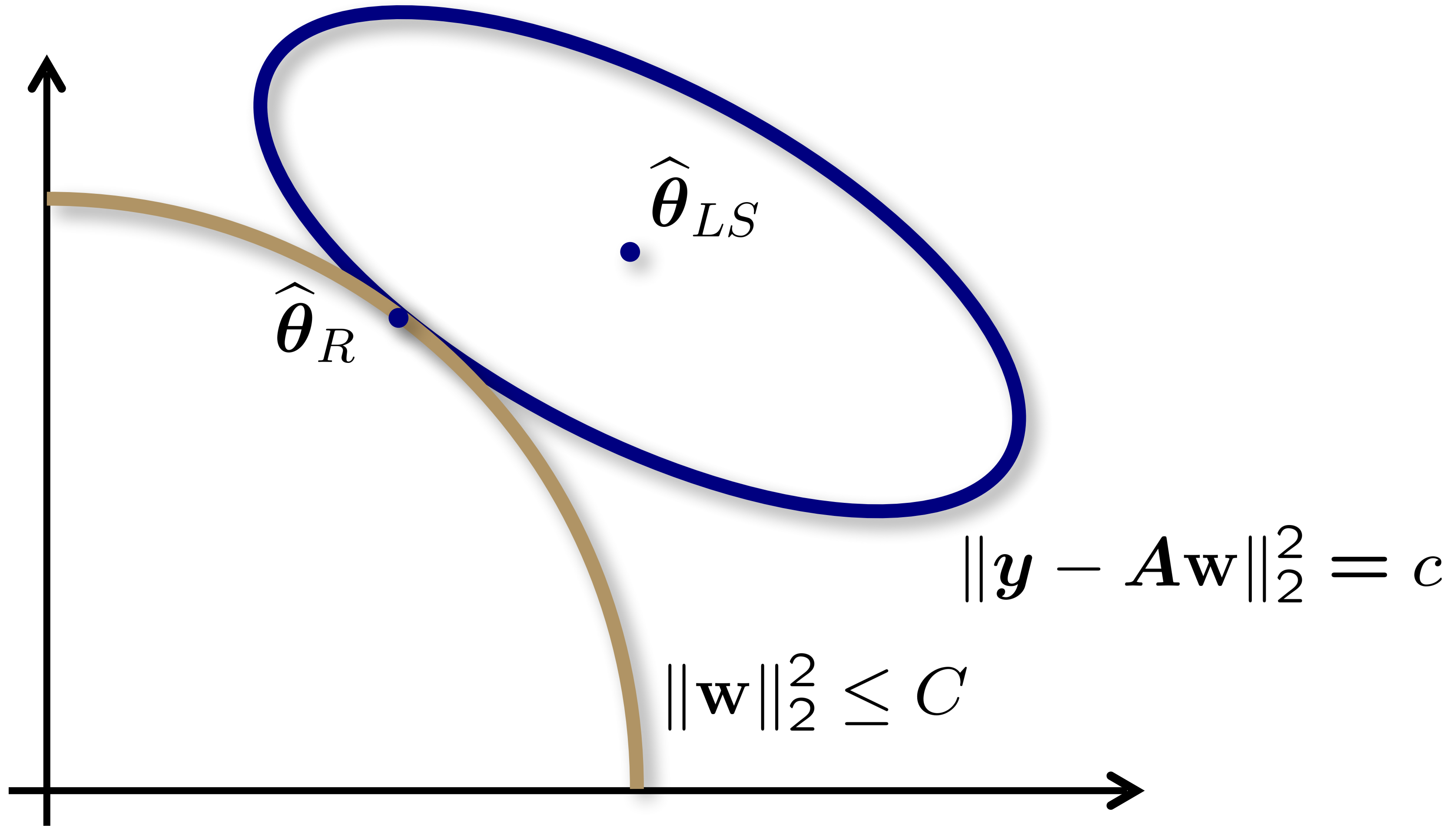
We saw this before

Remember regularization?

Minimize $\| \mathbf{y} - A\mathbf{w} \|$

subject to: $\mathbf{w}^T\mathbf{w} \leq C$

$$\boldsymbol{A} = \begin{bmatrix} 1 & x_1(1) & \cdots & x_1(d) \\ 1 & x_2(1) & \cdots & x_2(d) \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_n(1) & \cdots & x_n(d) \end{bmatrix}$$



	optimize	constraint
Regularization:	$\widehat{R}_n$	$\mathbf{w}^T\mathbf{w}$
SVM:	$\mathbf{w}^T\mathbf{w}$	$\widehat{R}_n$

Lagrange formulation

Minimize $\frac{1}{2}\mathbf{w}^T\mathbf{w}$ $y_n(\mathbf{w}^T\mathbf{x}_n + b) \geq 1$

Lagrange formulation

Minimize $\frac{1}{2}\mathbf{w}^T\mathbf{w}$ $y_n(\mathbf{w}^T\mathbf{x}_n + b) - 1$

Lagrange formulation

Minimize $\frac{1}{2}\mathbf{w}^T\mathbf{w}$ $\alpha_n(y_n(\mathbf{w}^T\mathbf{x}_n + b) - 1)$

Lagrange formulation

Minimize $\frac{1}{2}\mathbf{w}^T\mathbf{w} - \sum_{n=1}^{N} \alpha_n(y_n(\mathbf{w}^T\mathbf{x}_n + b) - 1)$

Lagrange formulation

Minimize $\quad \mathcal{L}(\mathbf{w}, b, \alpha) = \frac{1}{2}\mathbf{w}^T\mathbf{w} - \sum_{n=1}^{N} \alpha_n (y_n(\mathbf{w}^T\mathbf{x}_n + b) - 1)$

w.r.t. $\mathbf{w}$ and b and maximize w.r.t. each $\alpha_n \geq 0$

$$\nabla_{\mathbf{w}}\mathcal{L} = \mathbf{w} - \sum_{n=1}^{N} \alpha_n y_n \mathbf{x}_n \; = \; 0$$

$$\frac{\partial \mathcal{L}}{\partial b} = -\sum_{n=1}^{N} \alpha_n y_n \; = \; 0$$

Substituting...

$$\mathbf{w} = \sum_{n=1}^{N} \alpha_n y_n \mathbf{x}_n \qquad \text{and} \qquad \sum_{n=1}^{N} \alpha_n y_n = 0$$

in the Lagrangian $\mathcal{L}(\mathbf{w}, b, \alpha) = \frac{1}{2}\mathbf{w}^T\mathbf{w} - \sum_{n=1}^{N} \alpha_n(y_n(\mathbf{w}^T\mathbf{x}_n + b) - 1)$

we get $\sum_{n=1}^{N} \alpha_n$

Substituting...

$$\mathbf{w} = \sum_{n=1}^{N} \alpha_n y_n \mathbf{x}_n \qquad \text{and} \qquad \sum_{n=1}^{N} \alpha_n y_n = 0$$

in the Lagrangian $\mathcal{L}(\mathbf{w}, b, \alpha) = \frac{1}{2}\mathbf{w}^T\mathbf{w} - \sum_{n=1}^{N} \alpha_n (y_n(\mathbf{w}^T\mathbf{x}_n + b) - 1)$

we get $\sum_{n=1}^{N} \alpha_n - \frac{1}{2} \sum_{n=1}^{N} \sum_{m=1}^{N} y_n y_m \alpha_n \alpha_m \mathbf{x_n}^T \mathbf{x_m}$

Substituting...

$$\mathbf{w} = \sum_{n=1}^{N} \alpha_n y_n \mathbf{x}_n \qquad \text{and} \qquad \sum_{n=1}^{N} \alpha_n y_n = 0$$

in the Lagrangian $\mathcal{L}(\mathbf{w}, b, \alpha) = \frac{1}{2}\mathbf{w}^T\mathbf{w} - \sum_{n=1}^{N} \alpha_n (y_n(\mathbf{w}^T\mathbf{x}_n + b) - 1)$

we get $\boxed{\mathcal{L}(\alpha) = \sum_{n=1}^{N} \alpha_n - \frac{1}{2} \sum_{n=1}^{N} \sum_{m=1}^{N} y_n y_m \alpha_n \alpha_m \mathbf{x_n}^T \mathbf{x_m}}$

Maximize w.r.t. to α <u>subject to</u> $\alpha_n \geq 0$ for $n = 1, \ldots, N$

and $\sum_{n=1}^{N} \alpha_n y_n = 0$

The solution

The solution – quadratic programming

$$\min_\alpha \quad \frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N} y_n y_m \alpha_n \alpha_m \mathbf{x_n}^T \mathbf{x_m} - \sum_{n=1}^{N} \alpha_n$$

The solution – quadratic programming

$$\min_\alpha \quad \frac{1}{2}\alpha^T \underbrace{\begin{bmatrix} y_1 y_1 \mathbf{x_1}^T\mathbf{x_1} & y_1 y_2 \mathbf{x_1}^T\mathbf{x_2} & \cdots & y_1 y_N \mathbf{x_1}^T\mathbf{x_N} \\ y_2 y_1 \mathbf{x_2}^T\mathbf{x_1} & y_2 y_2 \mathbf{x_2}^T\mathbf{x_2} & \cdots & y_2 y_N \mathbf{x_2}^T\mathbf{x_N} \\ \cdots & \cdots & \cdots & \cdots \\ y_N y_1 \mathbf{x_N}^T\mathbf{x_1} & y_N y_2 \mathbf{x_N}^T\mathbf{x_2} & \cdots & y_N y_N \mathbf{x_N}^T\mathbf{x_N} \end{bmatrix}}_{\text{quadratic coefficients}} \alpha + \underbrace{(-\mathbf{1}^T)}_{\text{linear}}\alpha$$

The solution – quadratic programming

$$\min_\alpha \quad \frac{1}{2}\alpha^T \underbrace{\begin{bmatrix} y_1 y_1 \mathbf{x_1}^T\mathbf{x_1} & y_1 y_2 \mathbf{x_1}^T\mathbf{x_2} & \cdots & y_1 y_N \mathbf{x_1}^T\mathbf{x_N} \\ y_2 y_1 \mathbf{x_2}^T\mathbf{x_1} & y_2 y_2 \mathbf{x_2}^T\mathbf{x_2} & \cdots & y_2 y_N \mathbf{x_2}^T\mathbf{x_N} \\ \cdots & \cdots & \cdots & \cdots \\ y_N y_1 \mathbf{x_N}^T\mathbf{x_1} & y_N y_2 \mathbf{x_N}^T\mathbf{x_2} & \cdots & y_N y_N \mathbf{x_N}^T\mathbf{x_N} \end{bmatrix}}_{\text{quadratic coefficients}} \alpha + (\underbrace{-\mathbf{1}^T}_{\text{linear}})\alpha$$

$$\text{subject to} \quad \underbrace{\mathbf{y}^T\alpha = 0}_{\text{linear constraint}}$$

$$\underbrace{0 \leq}_{\text{lower bounds}} \alpha \underbrace{\leq \infty}_{\text{upper bounds}}$$

The solution – quadratic programming

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

Quadratic programming output: α

Solution: $\alpha = \alpha_1, \ldots, \alpha_N$

$$\longrightarrow \mathbf{w} = \sum_{n=1}^{N} \alpha_n y_n \mathbf{x}_n$$

KKT condition: For $n = 1, \ldots, N$

$$\alpha_n(y_n(\mathbf{w}^T\mathbf{x}_n + b) - 1) = 0$$

Quadratic programming output: α

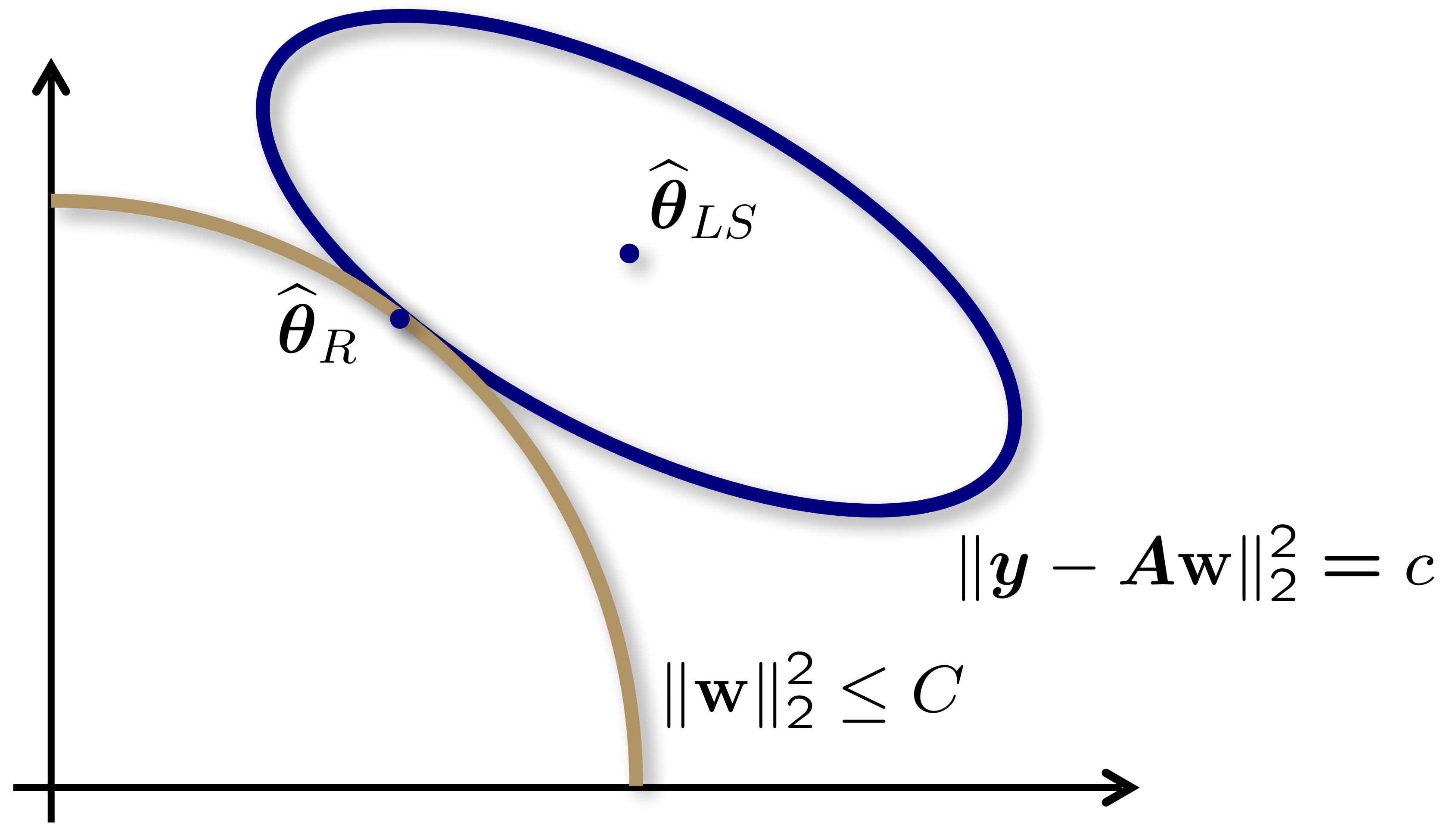
$$\mathbf{A} = \begin{bmatrix} 1 & x_1(1) & \cdots & x_1(d) \\ 1 & x_2(1) & \cdots & x_2(d) \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_n(1) & \cdots & x_n(d) \end{bmatrix}$$

Solution: $\alpha = \alpha_1, \ldots, \alpha_N$

$$\longrightarrow \mathbf{w} = \sum_{n=1}^{N} \alpha_n y_n \mathbf{x}_n$$

KKT condition: For $n = 1, \ldots, N$

$$\alpha_n(y_n(\mathbf{w}^T \mathbf{x}_n + b) - 1) = 0$$



We saw this before!

Quadratic programming output: α

Solution: $\alpha = \alpha_1, \dots, \alpha_N$

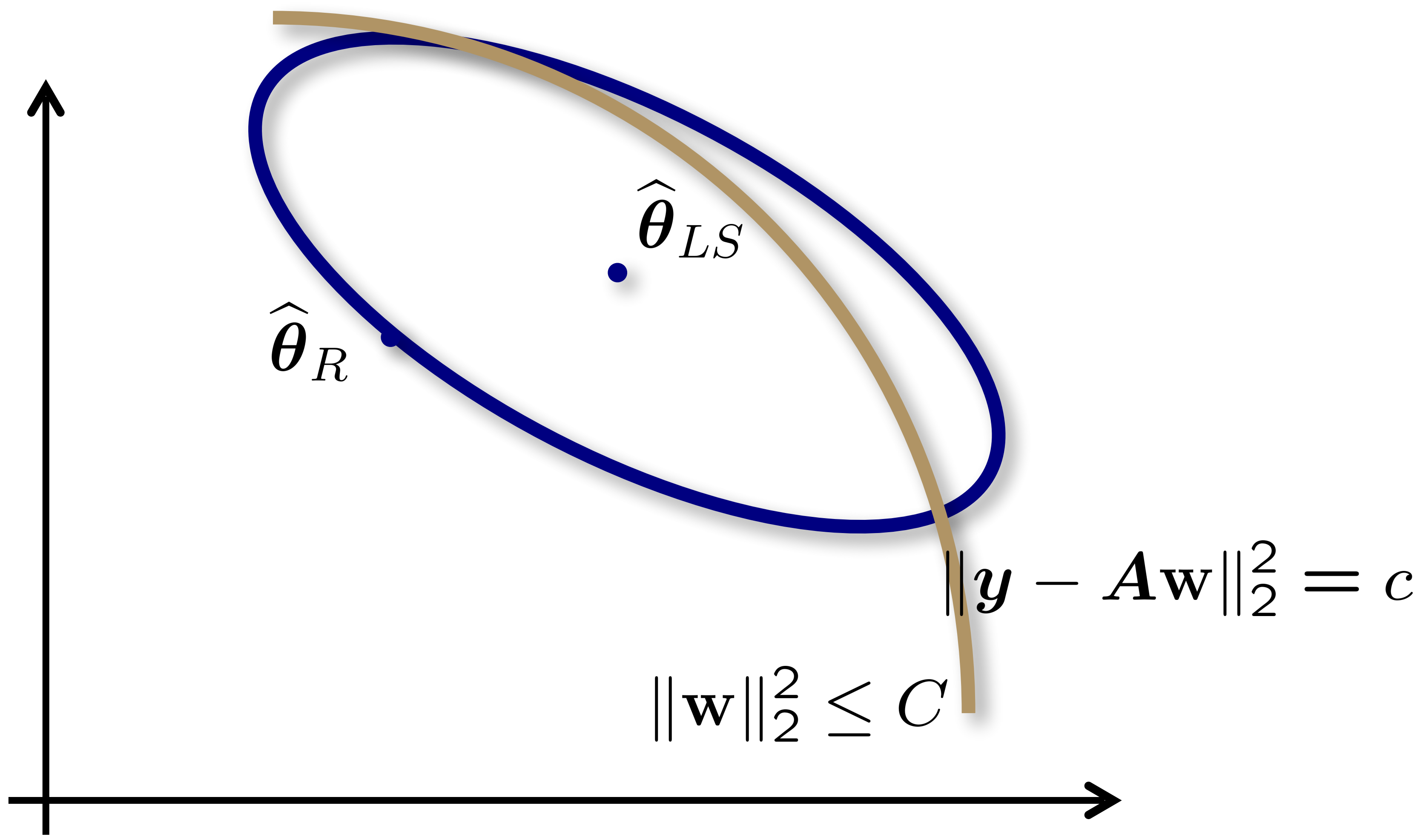
$$\longrightarrow \mathbf{w} = \sum_{n=1}^{N} \alpha_n y_n \mathbf{x}_n$$

KKT condition: For $n = 1, \dots, N$

$$\alpha_n(y_n(\mathbf{w}^T\mathbf{x}_n + b) - 1) = 0$$

We saw this before!

$$\boldsymbol{A} = \begin{bmatrix} 1 & x_1(1) & \cdots & x_1(d) \\ 1 & x_2(1) & \cdots & x_2(d) \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_n(1) & \cdots & x_n(d) \end{bmatrix}$$



Quadratic programming output: α

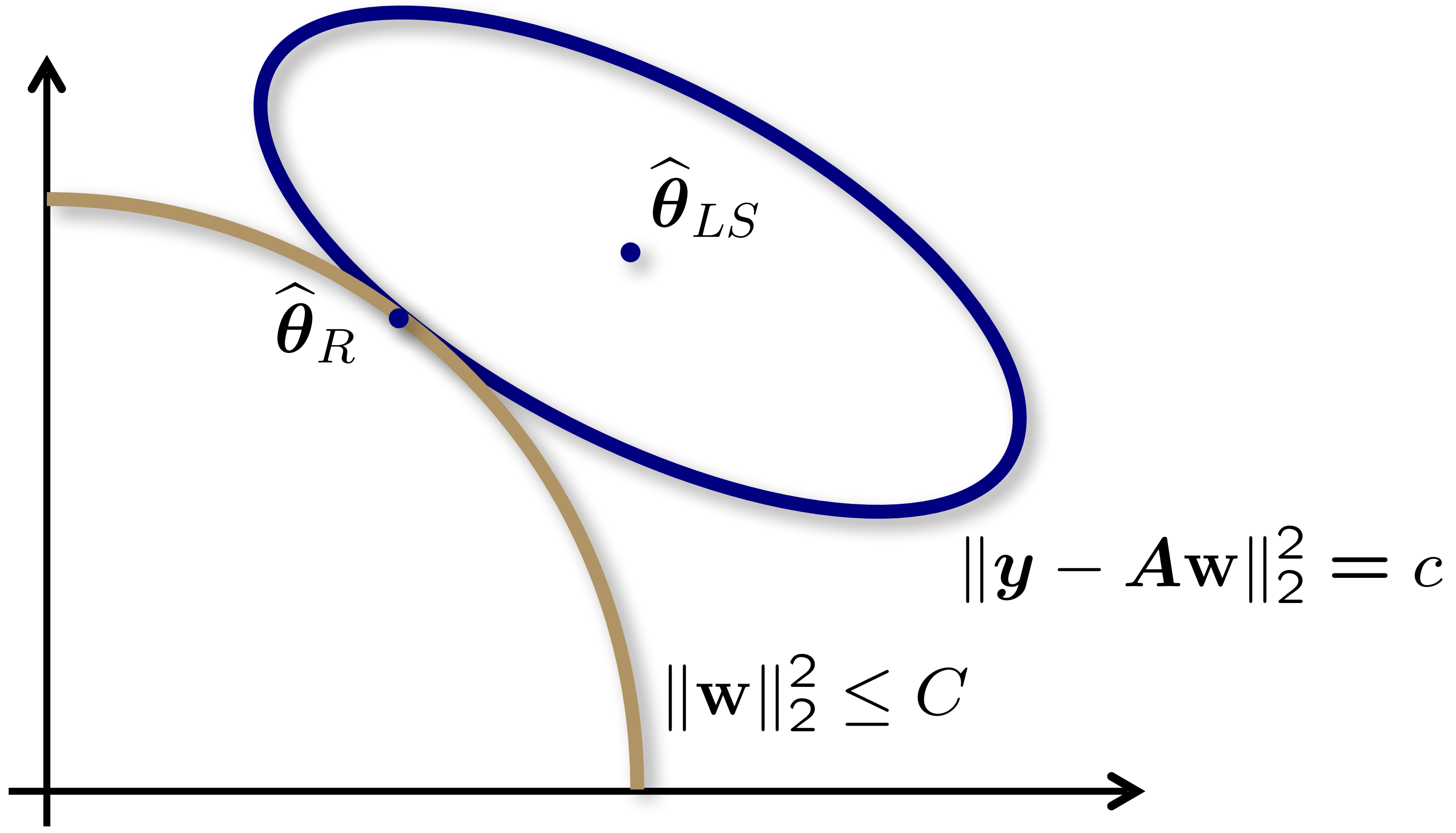
$$\boldsymbol{A} = \begin{bmatrix} 1 & x_1(1) & \cdots & x_1(d) \\ 1 & x_2(1) & \cdots & x_2(d) \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_n(1) & \cdots & x_n(d) \end{bmatrix}$$

Solution: $\alpha = \alpha_1, \ldots, \alpha_N$

$$\longrightarrow \mathbf{w} = \sum_{n=1}^{N} \alpha_n y_n \mathbf{x}_n$$

KKT condition: For $n = 1, \ldots, N$

$$\alpha_n(y_n(\mathbf{w}^T\mathbf{x}_n + b) - 1) = 0$$



We saw this before!

$\alpha_n > 0 \longrightarrow \mathbf{x}_n$ is a **support vector**

Support vectors

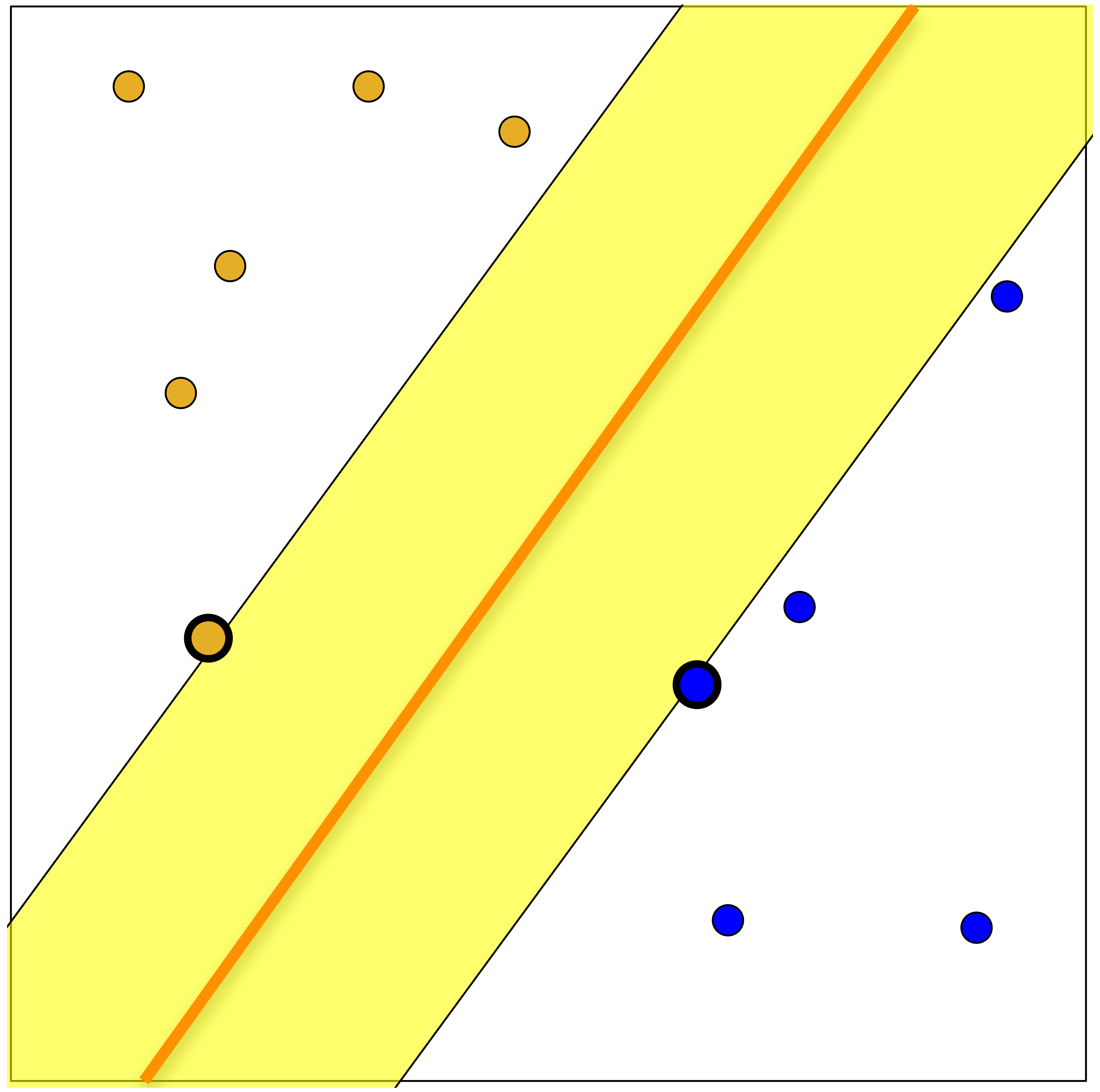
Closest $\mathbf{x}_n$'s to the plane: achieve the margin

$$\longrightarrow y_n(\mathbf{w}^T\mathbf{x}_n + b) = 1$$

$$\mathbf{w} = \sum_{\mathbf{x}_n \text{ is SV}} \alpha_n y_n \mathbf{x}_n$$

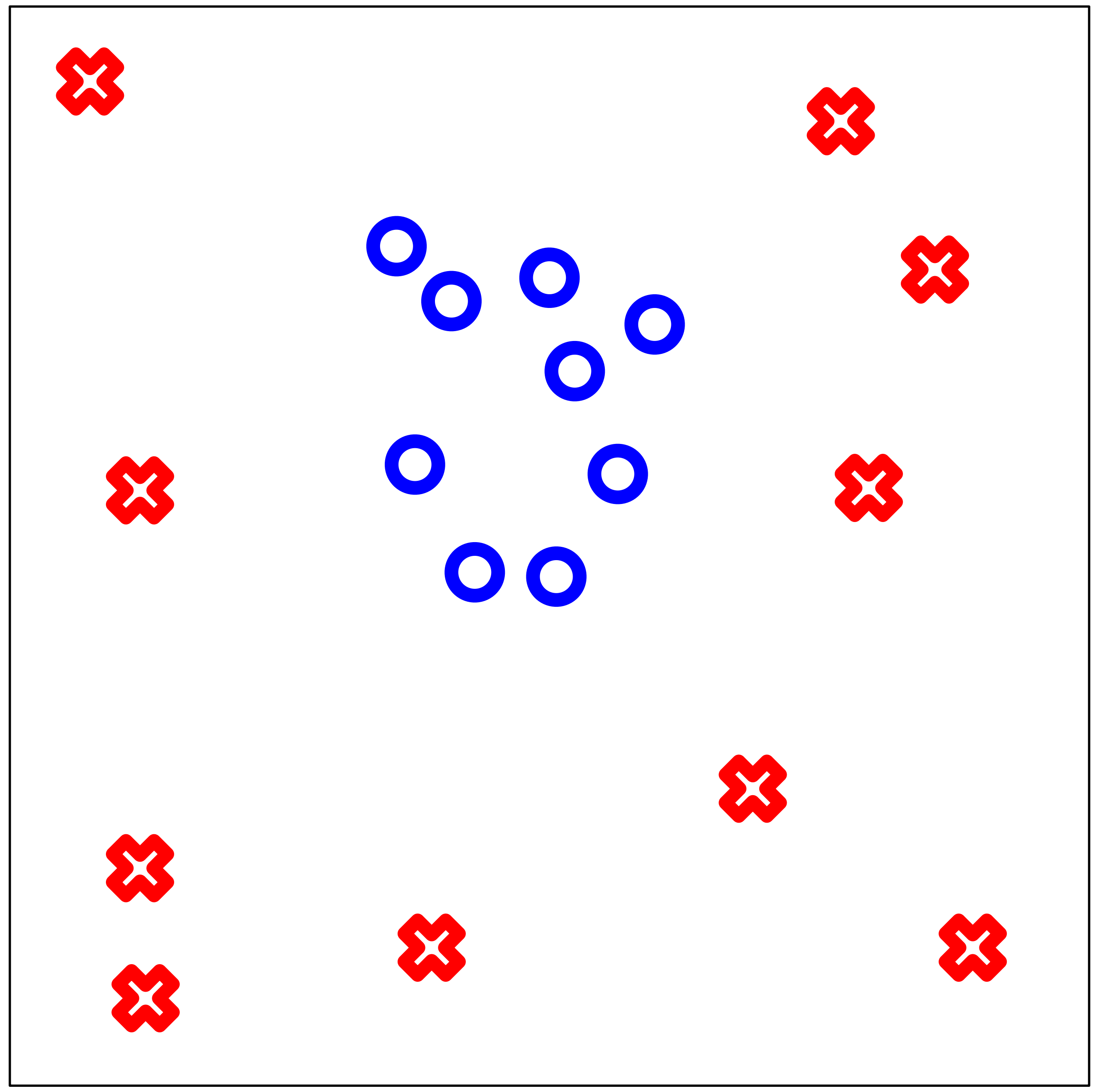
Solve for $\mathbf{b}$ using any SV:

$$y_n(\mathbf{w}^T\mathbf{x}_n + b) = 1$$

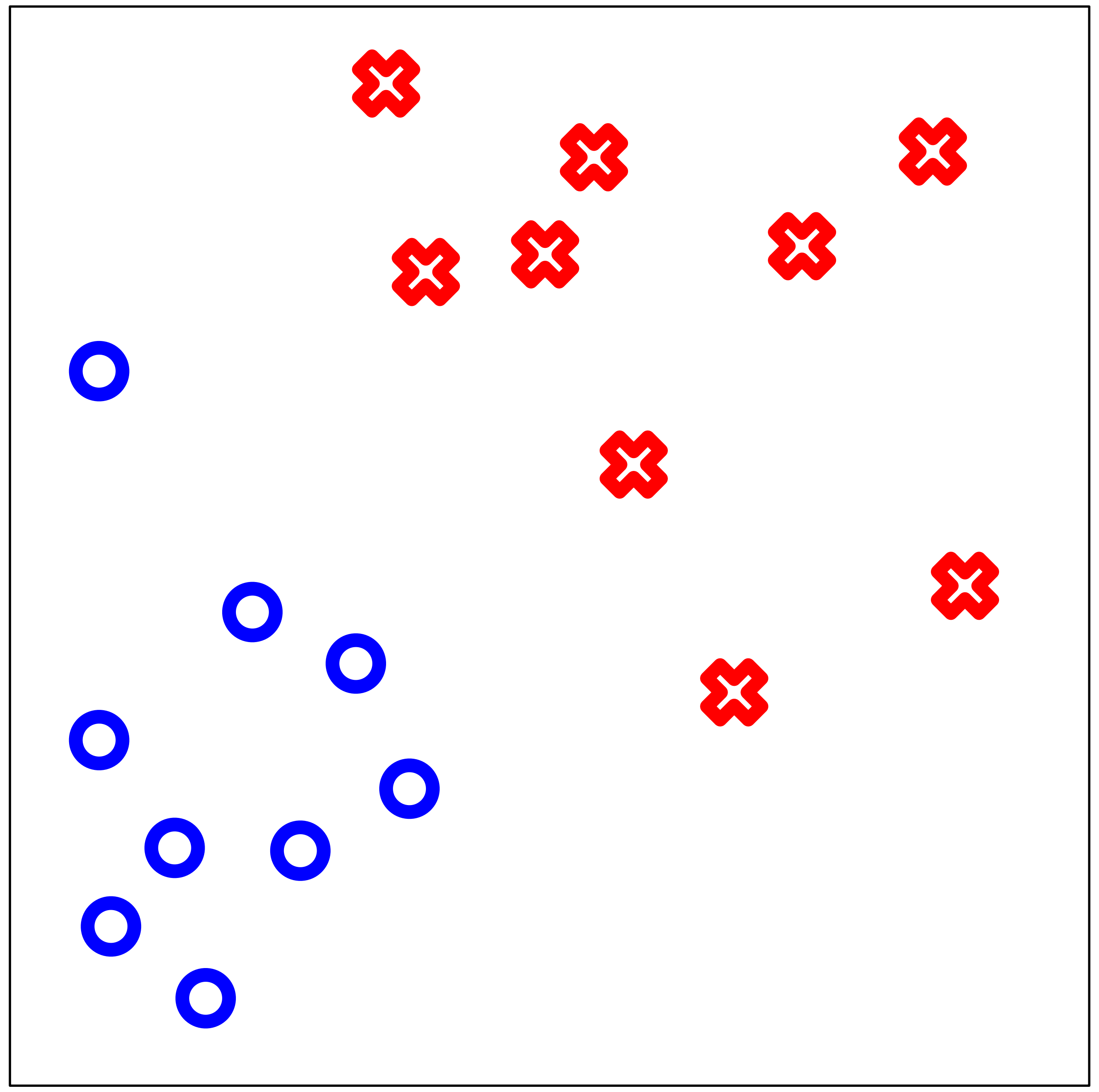


$\mathbf{z}$ instead of $\mathbf{x}$

$$\mathcal{L}(\alpha) = \sum_{n=1}^{N} \alpha_n - \frac{1}{2} \sum_{n=1}^{N} \sum_{m=1}^{N} y_n y_m \alpha_n \alpha_m \, \mathbf{z_n}^T \mathbf{z_m}$$



$$\mathcal{X} \longrightarrow \mathcal{Z}$$



“Support vectors” in $\mathcal{X}$ Space

Support vectors live in the $\mathcal{Z}$ space

In $\mathcal{X}$ space, “pre-images” of support vectors

The margin is maintained in $\mathcal{Z}$ space

Generalization result

$$\mathbb{E}\left[R(h)\right] \leq \frac{\mathbb{E}[\ |\text{support vectors}|\ ]}{N-1}$$

$$R(h) \leq \frac{|\text{support vectors}|}{N-1}$$