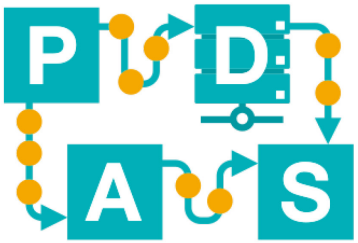
Support Vector Machines

Lecture 6

IDS-L6



Chair of Process and Data Science

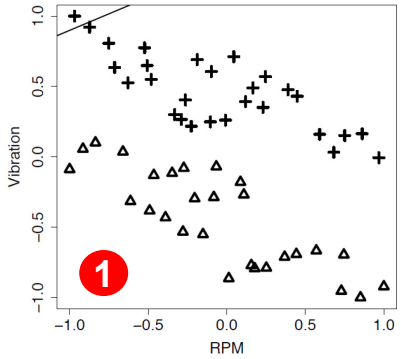
RWTH AACHEN UNIVERSITY

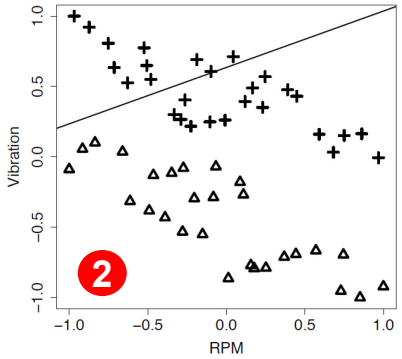
Outline of Today's Lecture

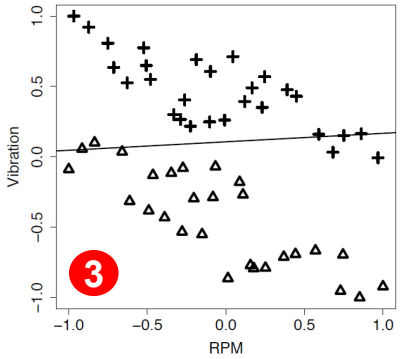
- SVMs: Basic Idea
- SVMs: A bit more technical
- Soft-margin SVMs
- Non-linear decision boundaries
- Using kernels

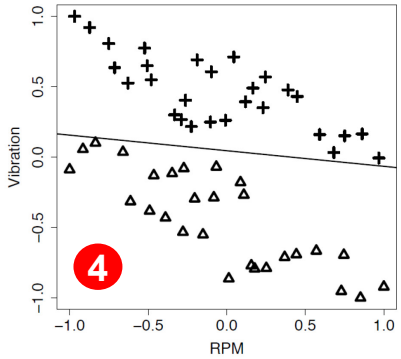
but first, let’s recapitulate

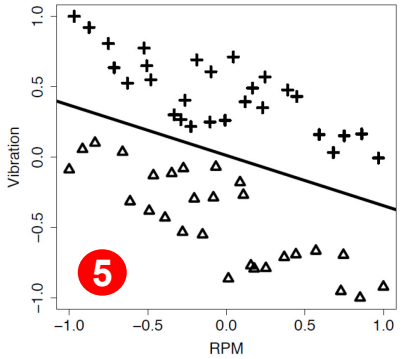
Logistic regression: Gradient descend process





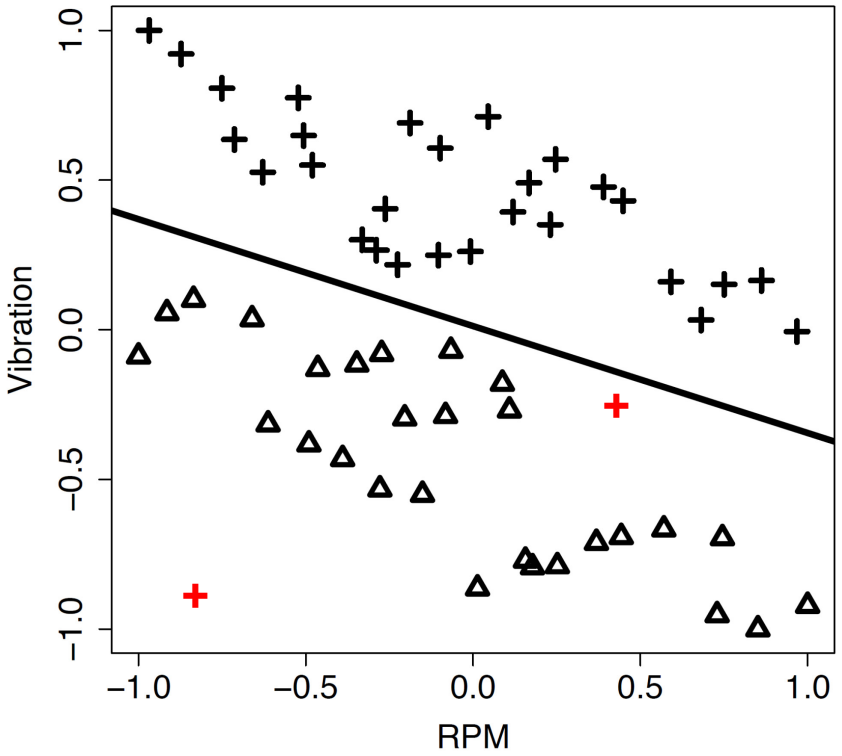




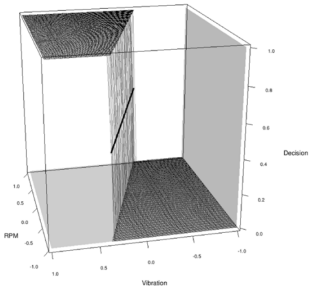


Chair of Process and Data Science

Measuring the error



$$\mathbb{M}_{\mathbf{w}}(\mathbf{d}) = \begin{cases} 1 & \text{if } \mathbf{w} \cdot \mathbf{d} \geq 0 \\ 0 & \textit{otherwise} \end{cases}$$

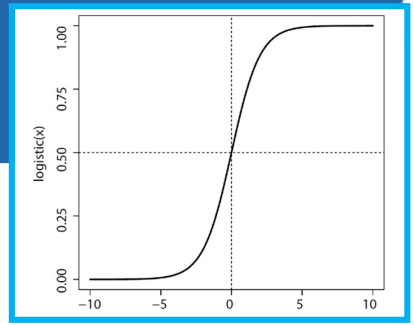


$$L_2(\mathbb{M}_{\mathbf{w}}, \mathcal{D}) = \frac{1}{2}\sum_{i=1}^{n}(t_i - \mathbb{M}_{\mathbf{w}}(\mathbf{d}_i))^2$$

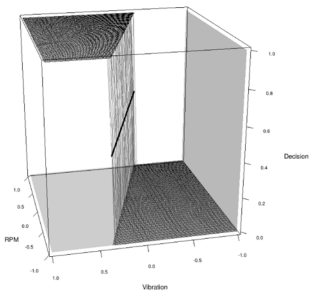
$$= \frac{1}{2}\sum_{i=1}^{n}(t_i - (\mathbf{w} \cdot \mathbf{d}_i))^2$$

Which way to go?

Using the logistic function

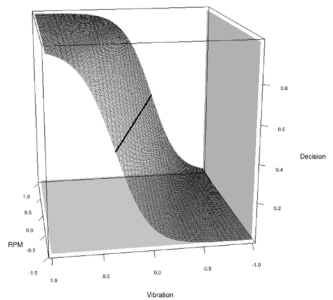


$$\mathbb{M}_{\mathbf{w}}(\mathbf{d}) = \begin{cases} 1 & \text{if } \mathbf{w} \cdot \mathbf{d} \geq 0 \\ 0 & \textit{otherwise} \end{cases}$$



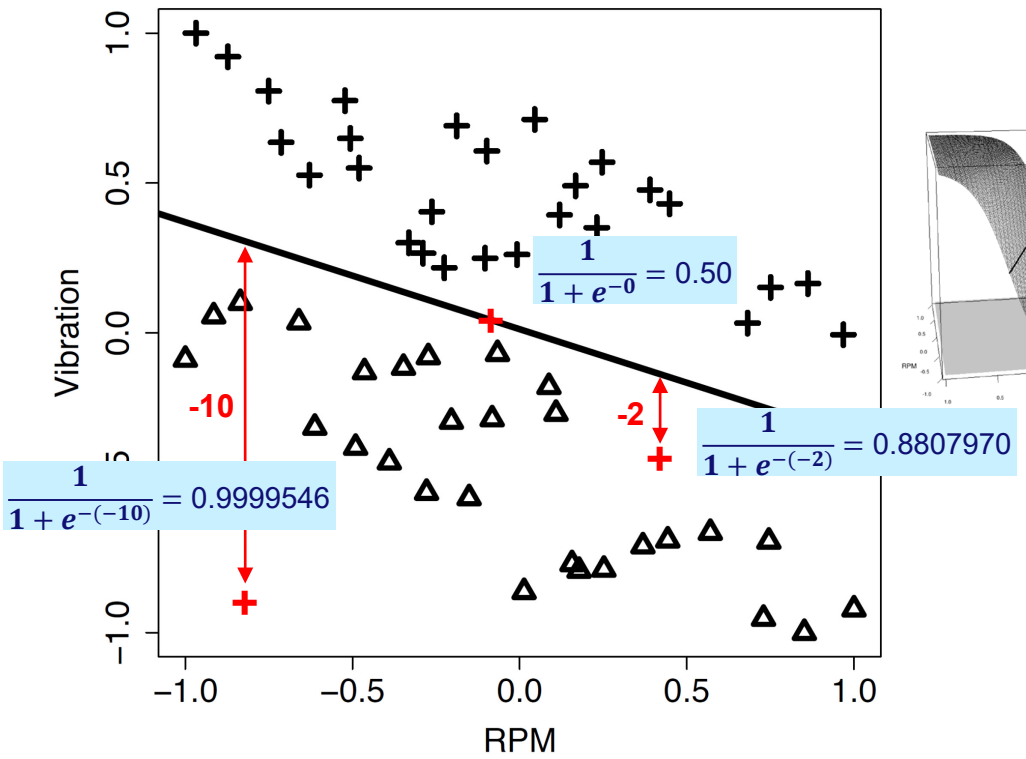
$$\mathbb{M}_{\mathbf{w}}(\mathbf{d}) = \textit{Logistic}(\mathbf{w} \cdot \mathbf{d})$$

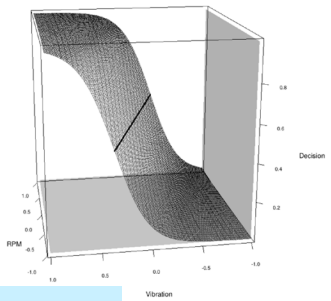
$$= \frac{1}{1 + e^{-\mathbf{w} \cdot \mathbf{d}}}$$



Chair of Process and Data Science

Measuring the error



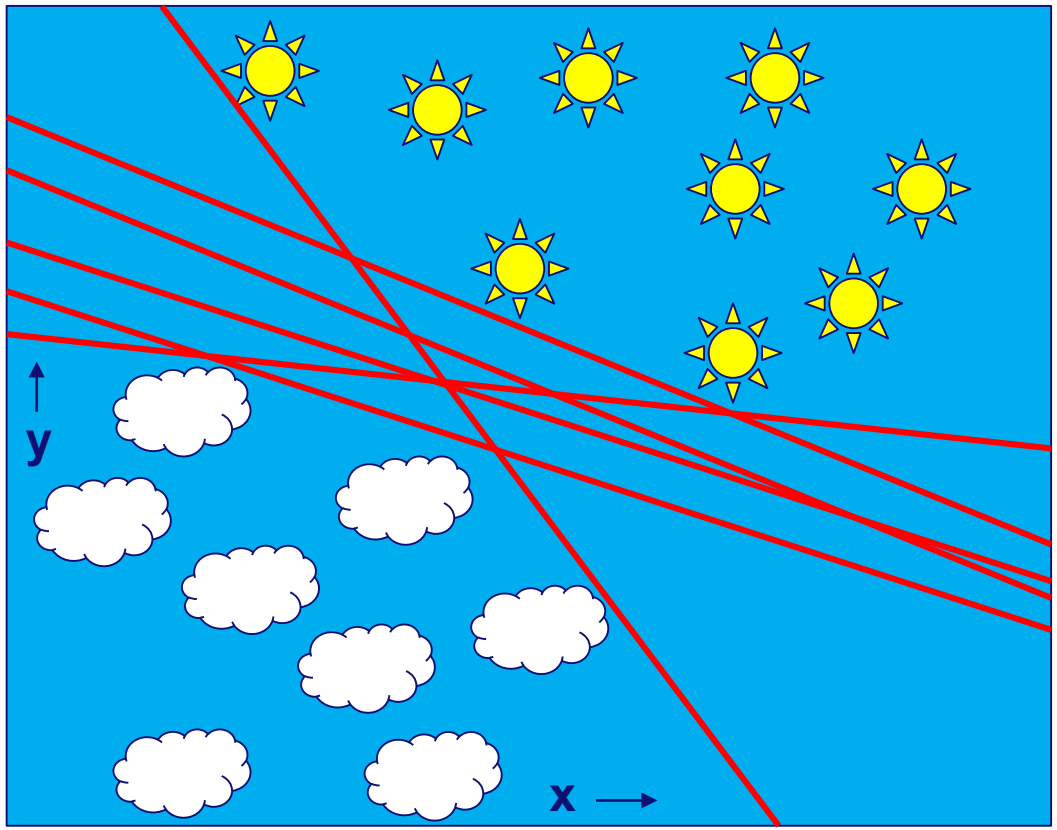


$$\mathbb{M}_{\mathbf{w}}(\mathbf{d}) = \mathit{Logistic}(\mathbf{w} \cdot \mathbf{d}) = \frac{1}{1 + e^{-\mathbf{w} \cdot \mathbf{d}}}$$

$$L_2(\mathbb{M}_{\mathbf{w}}, \mathcal{D}) = \frac{1}{2}\sum_{i=1}^{n}(t_i - \mathbb{M}_{\mathbf{w}}(\mathbf{d}_i))^2 = \frac{1}{2}\sum_{i=1}^{n}(t_i - (\mathbf{w} \cdot \mathbf{d}_i))^2$$

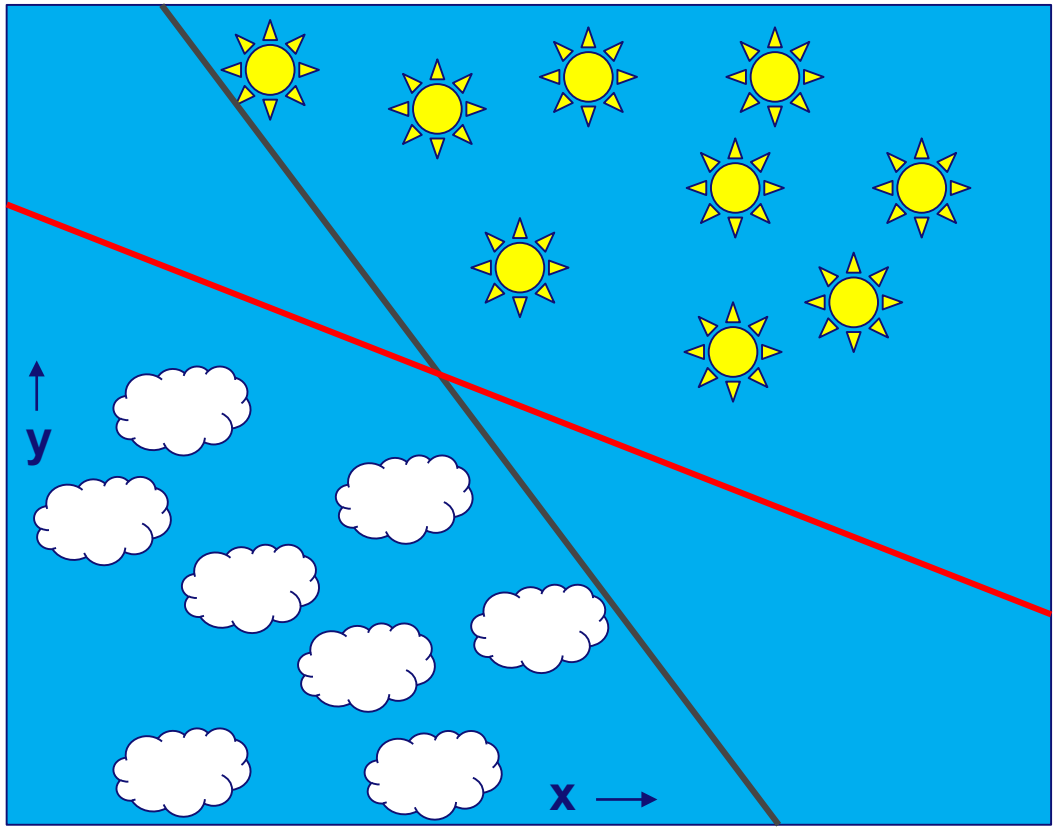
SVMs: Basic Idea

Basic idea



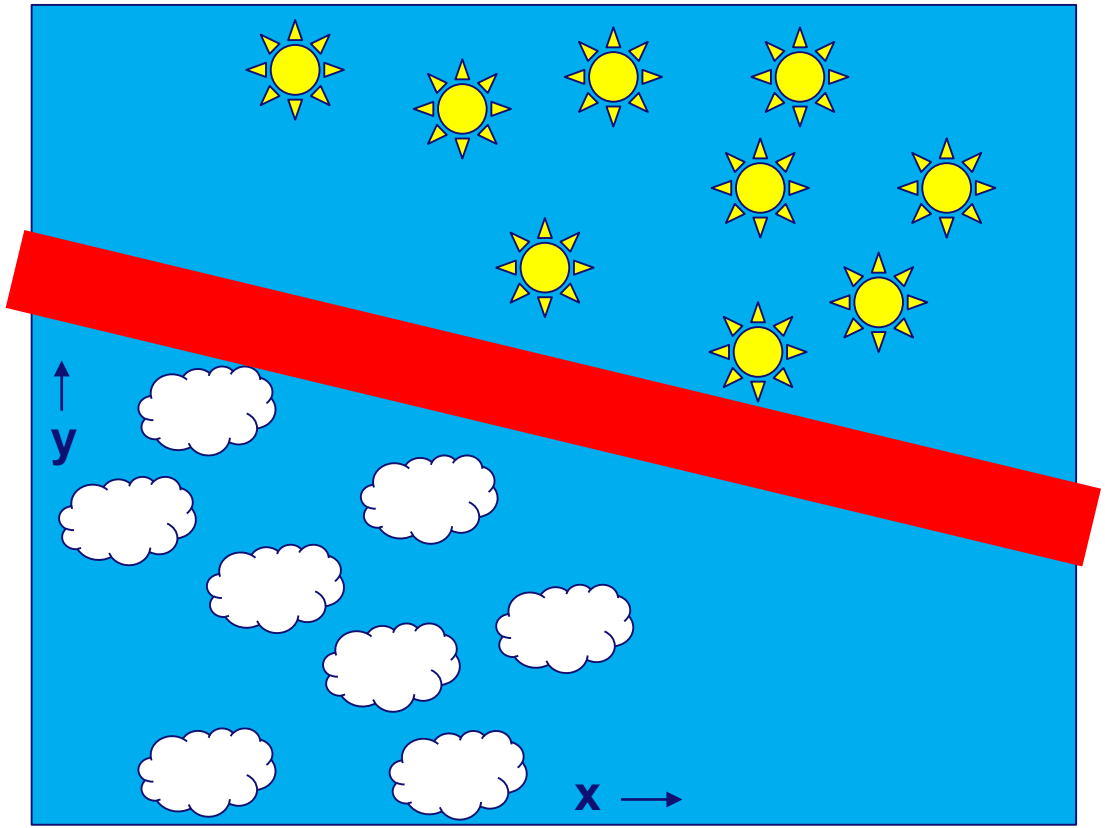
- Target features has two classes: sun and cloud.
- Two descriptive features: x and y.
- Separable data set.
- Find the best hyperplane (here line) separating both classes.

Basic idea



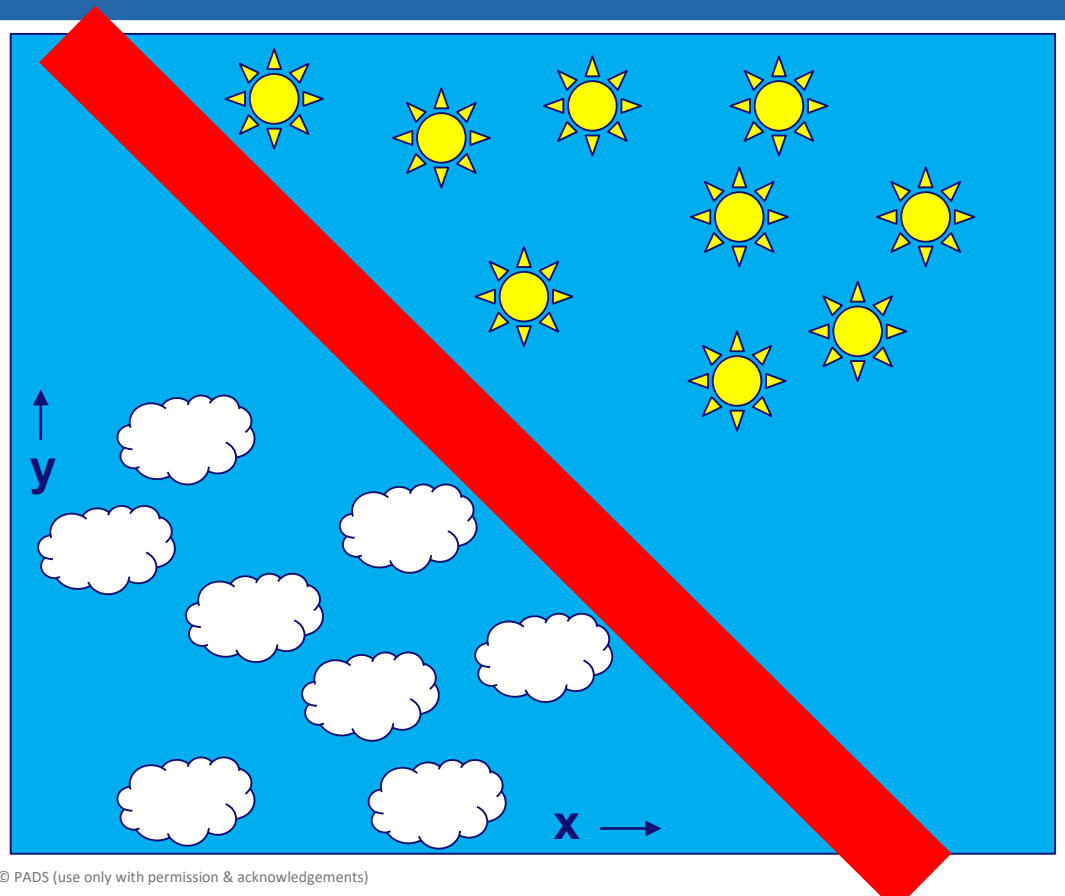
- **The red one seems better, but why?**

Basic idea



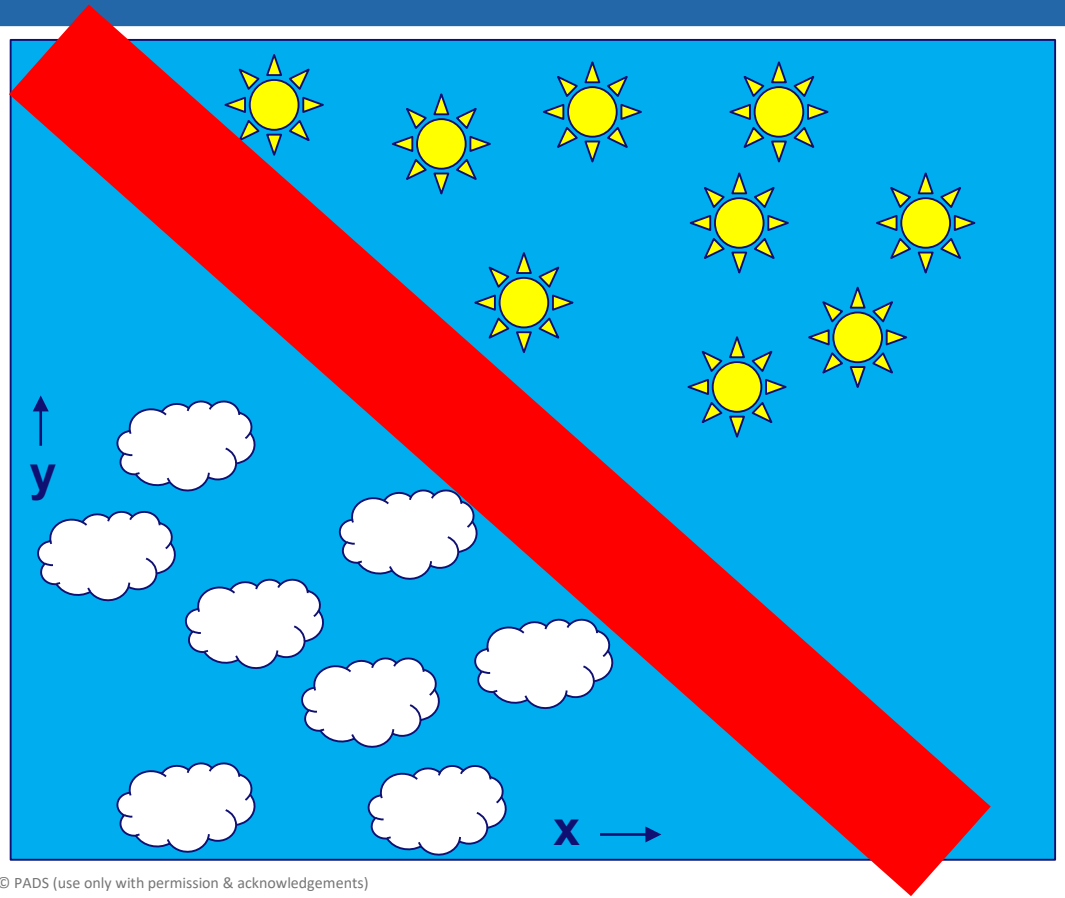
- **Degrees of freedom become less when the separating hyperplane gets thicker.**

Basic idea



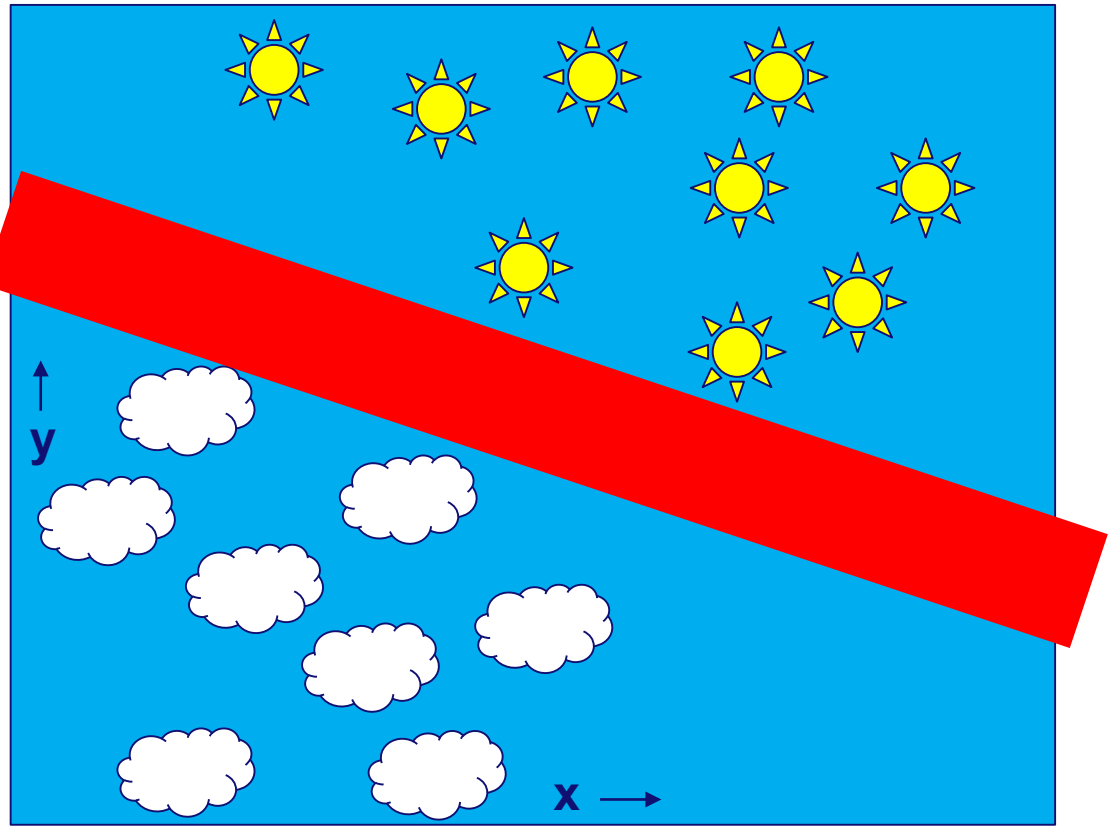
Chair of Process and Data Science

Basic idea



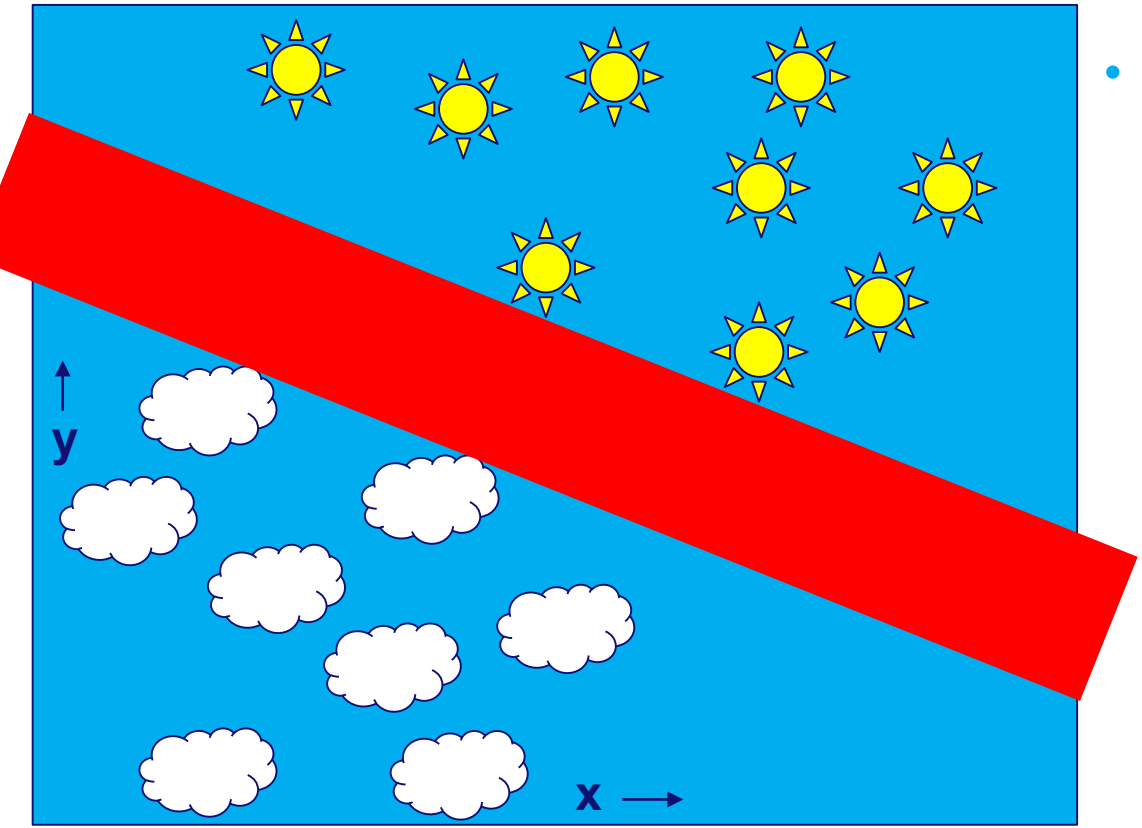
Chair of Process and Data Science

Basic idea



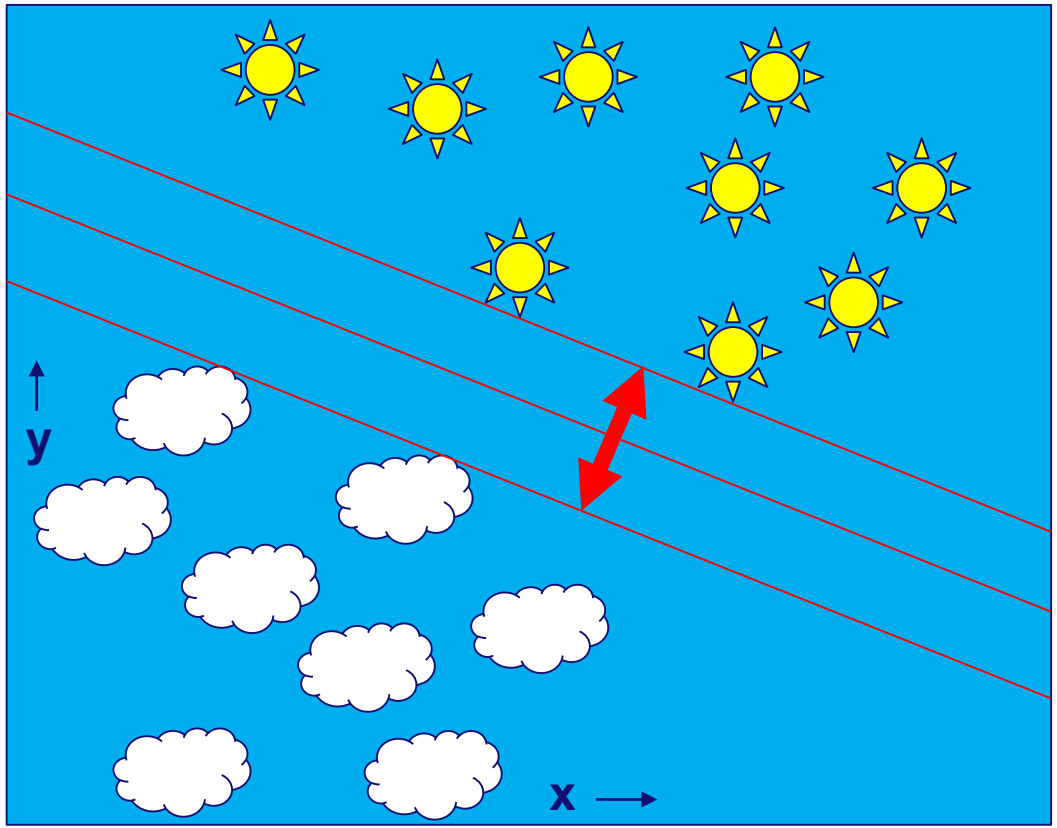
Chair of Process and Data Science

Basic idea



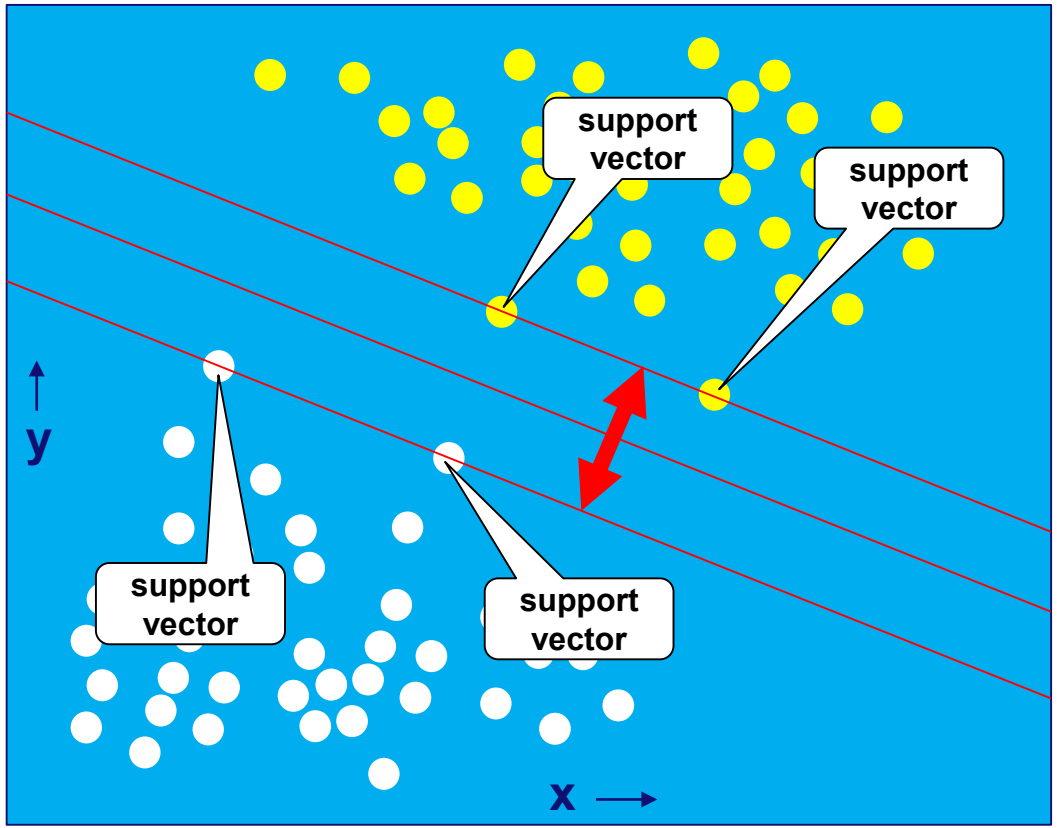
- **Separating hyperplane with the maximal thickness.**

Basic idea



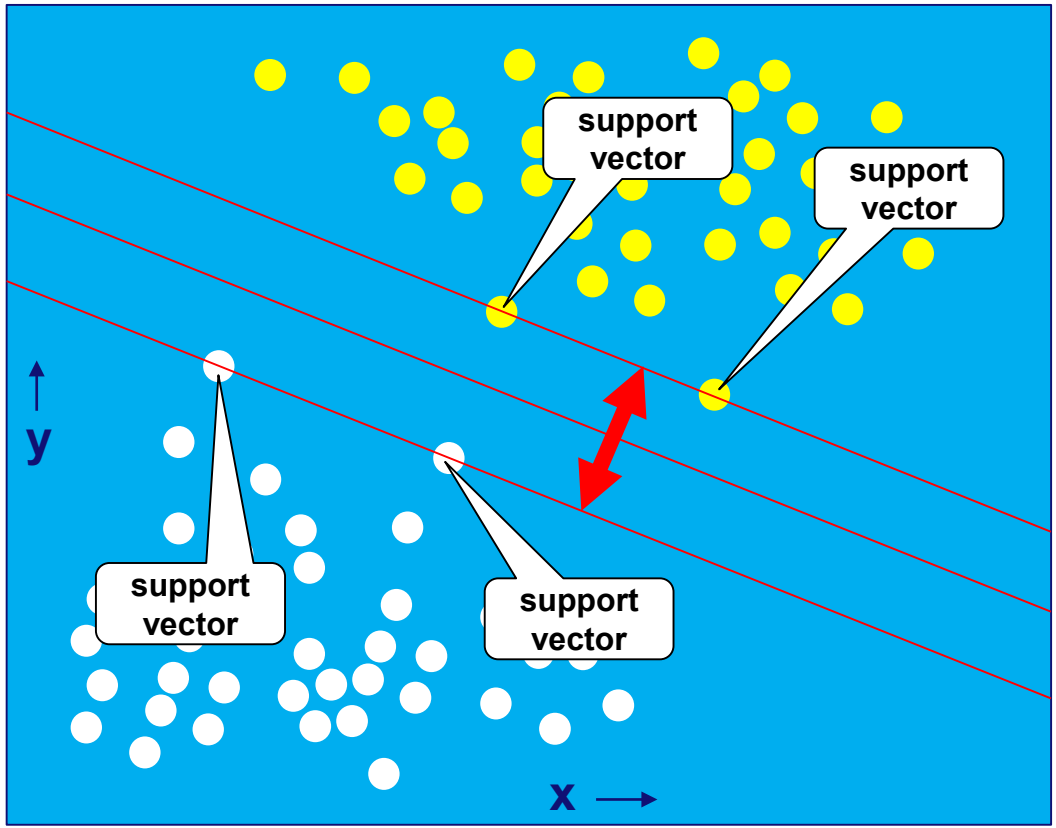
- **Thickness of the separating hyperplane is the safety margin.**

Basic idea



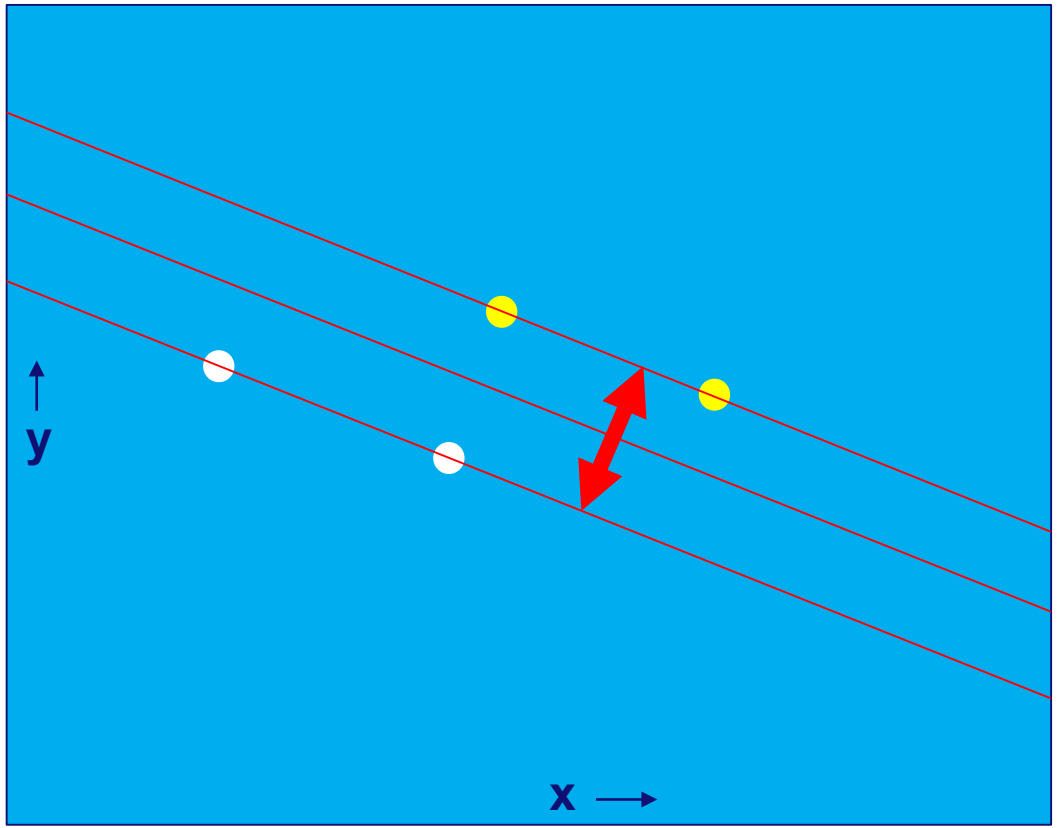
- Instances are actually (atomic) points in the N-dimensional space.
- The support vectors are the boundary points.

Key insight



- **Only the support vectors matter!**

Key insight



- **Only the support vectors matter!**
- **This makes things efficient.**

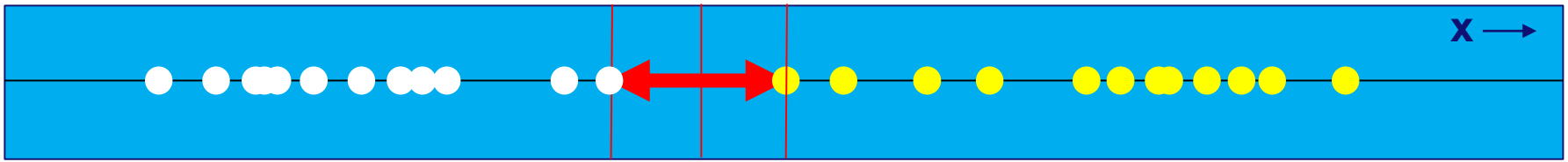
Support vectors

dog support vector

cat support vector

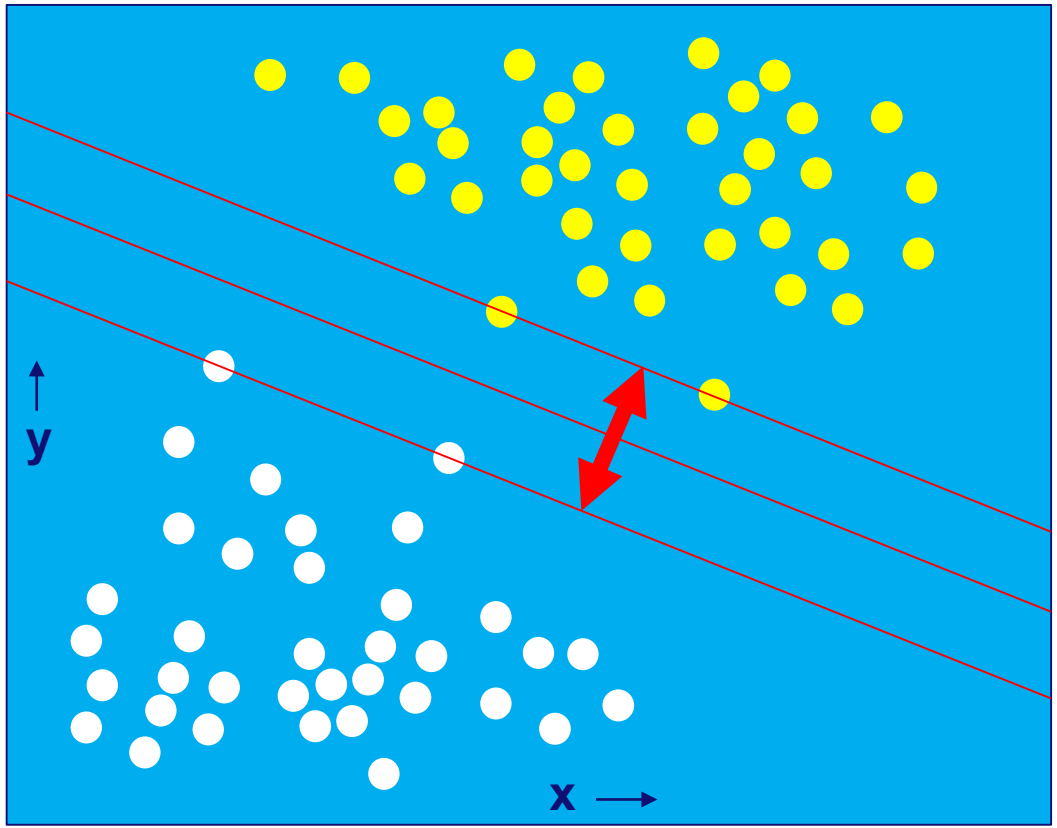
Chair of Process and Data Science

Multiple dimensions (n=1)



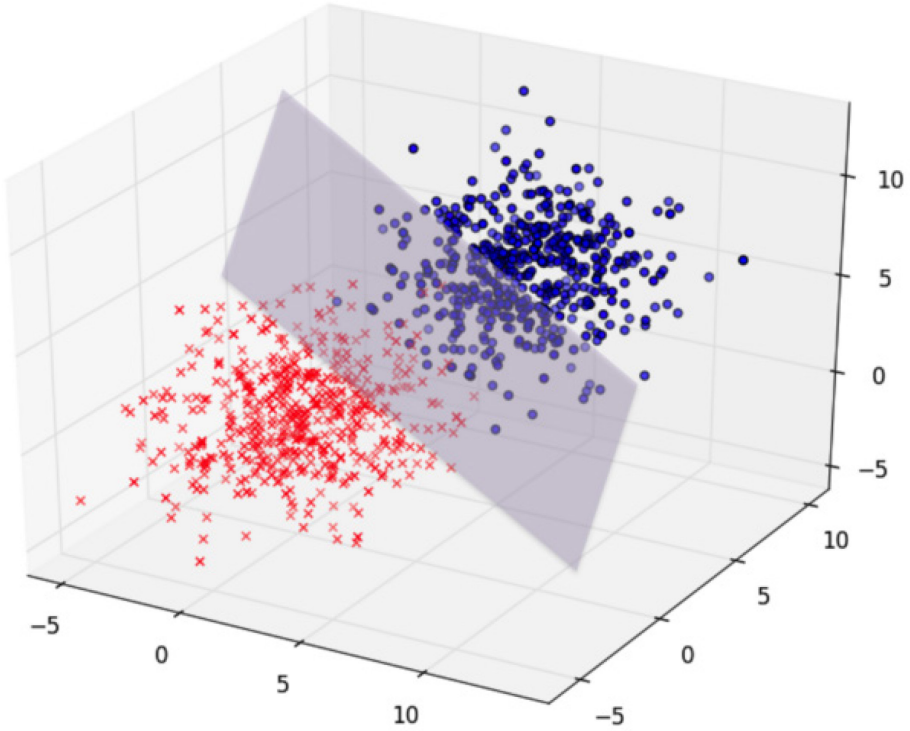
- Just one descriptive feature: x.
- Hyperplane has n-1=0 dimensions (a point).

Multiple dimensions (n=2)



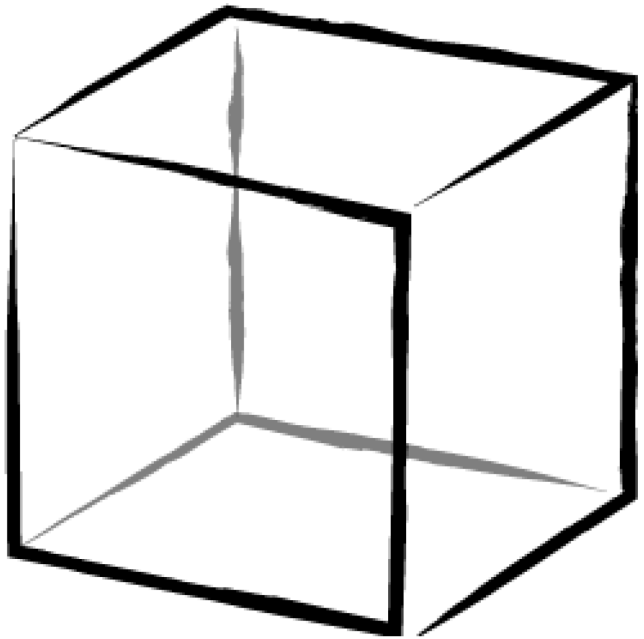
- Two descriptive features: x and y.
- Hyperplane has n-1=1 dimension (a line).

Multiple dimensions (n=3)



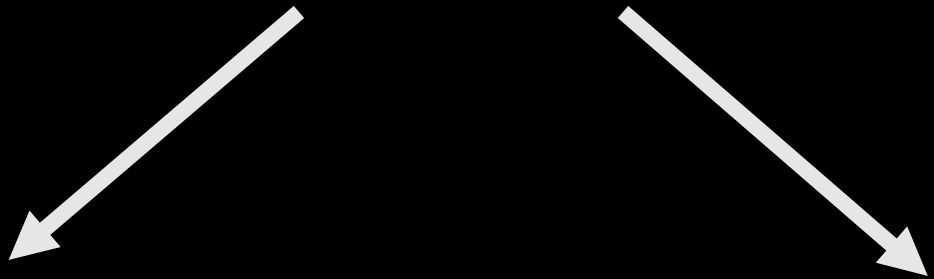
- Two descriptive features: x, y, and z.
- Hyperplane has n-1=2 dimensions.

Multiple dimensions



- *n* descriptive features: x_1, x_2, x_3, … x_n.
- Hyperplane has *n-1* dimensions.
- Problem: unable to visualize when *n* > *3*.
- Yet, SVMs work well for large n (e.g., compared to logistic regression).

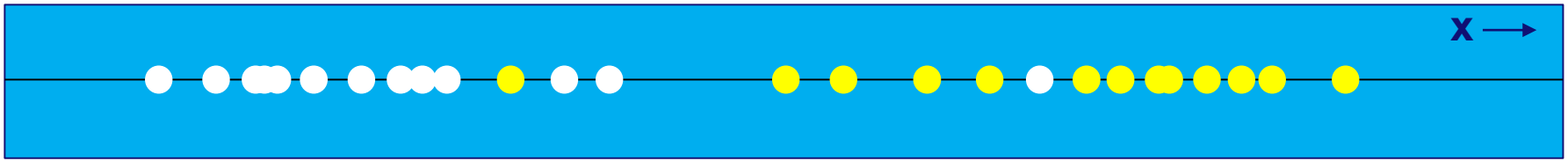
What if not separable?



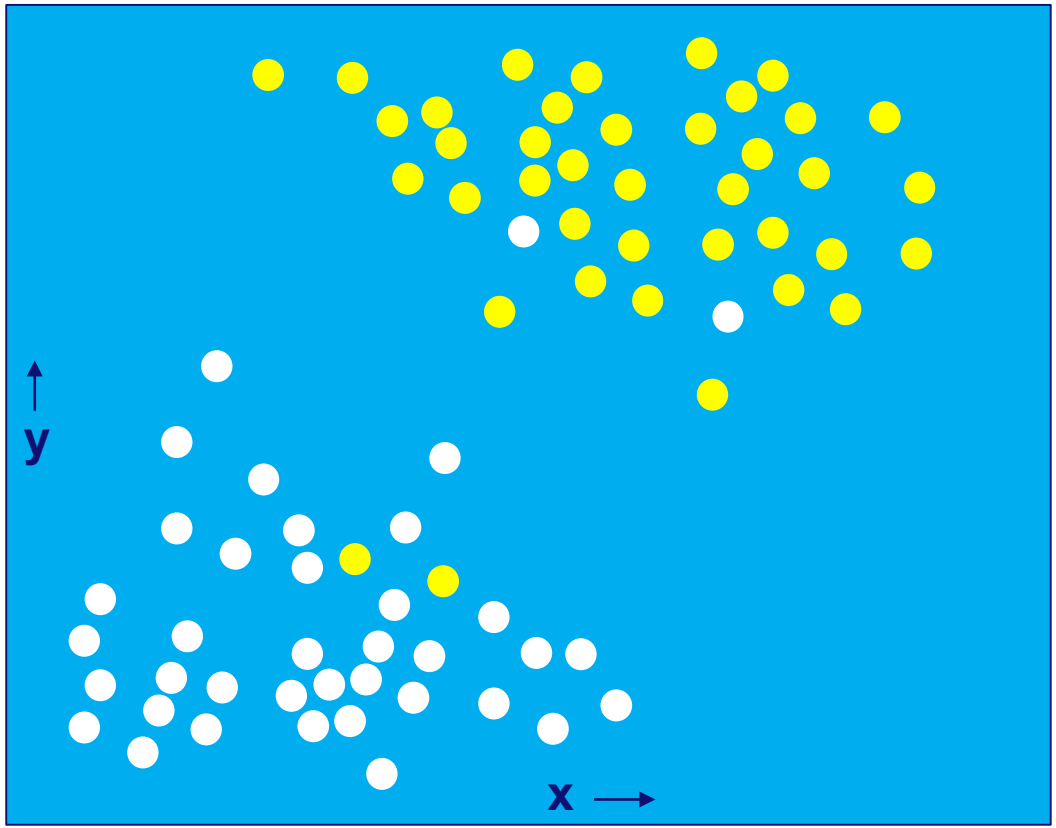
outliers

structural

What if not separable? (due to outliers)

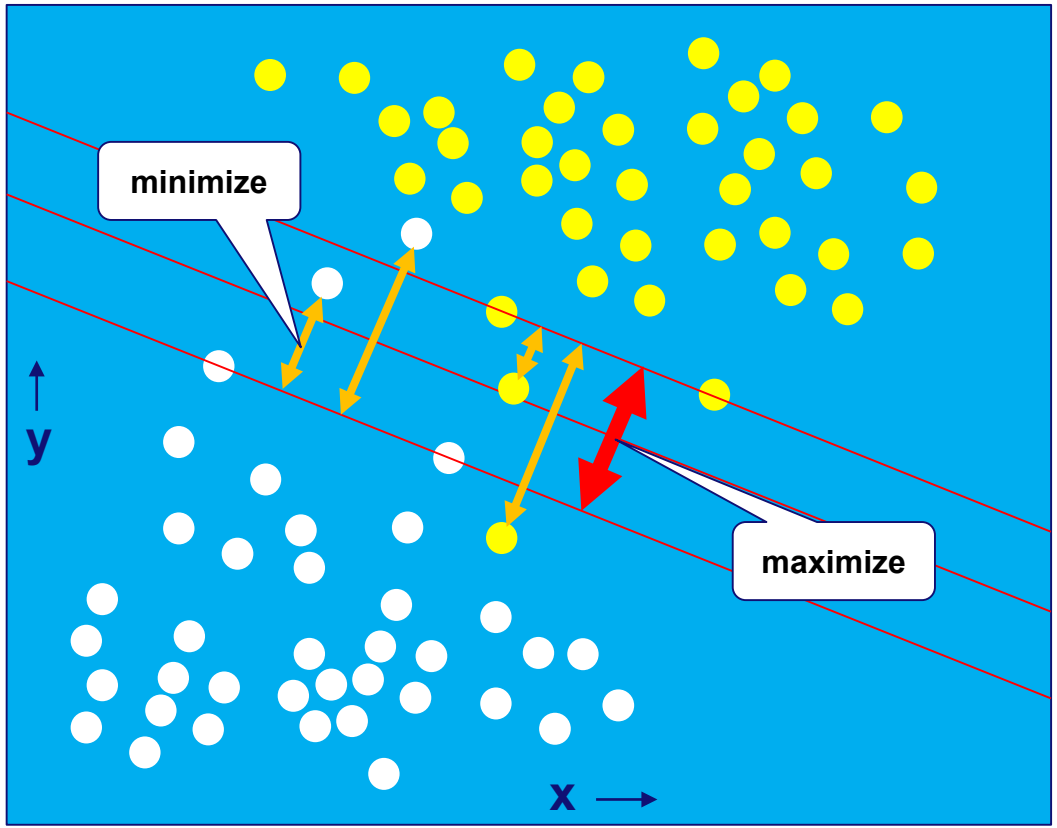


What if not separable? (due to outliers)



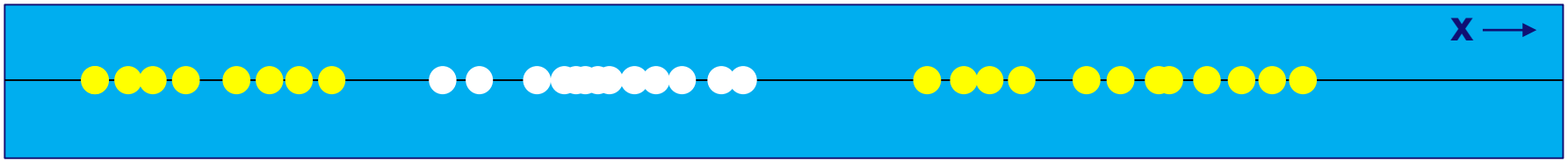
Chair of Process and Data Science

Idea



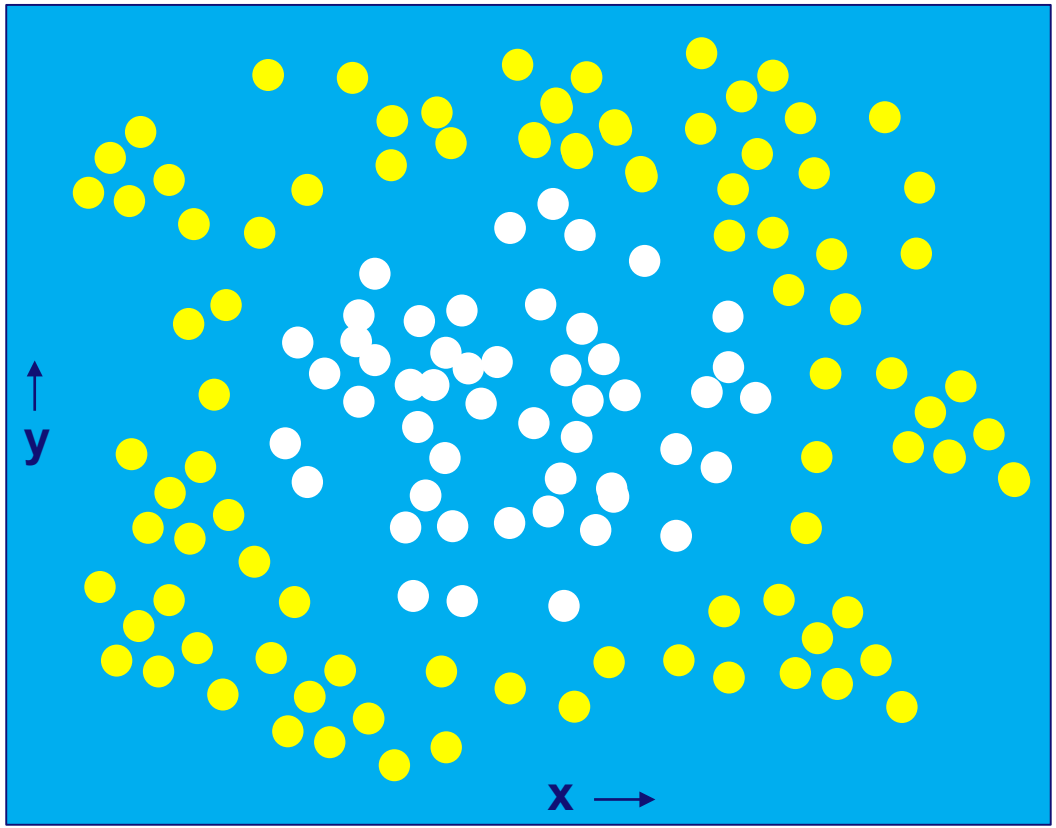
- Relax the problem.
- Allow for violating instances, but add penalty.

What if not separable? (structural problem)



Chair of Process and Data Science

What if not separable? (structural problem)

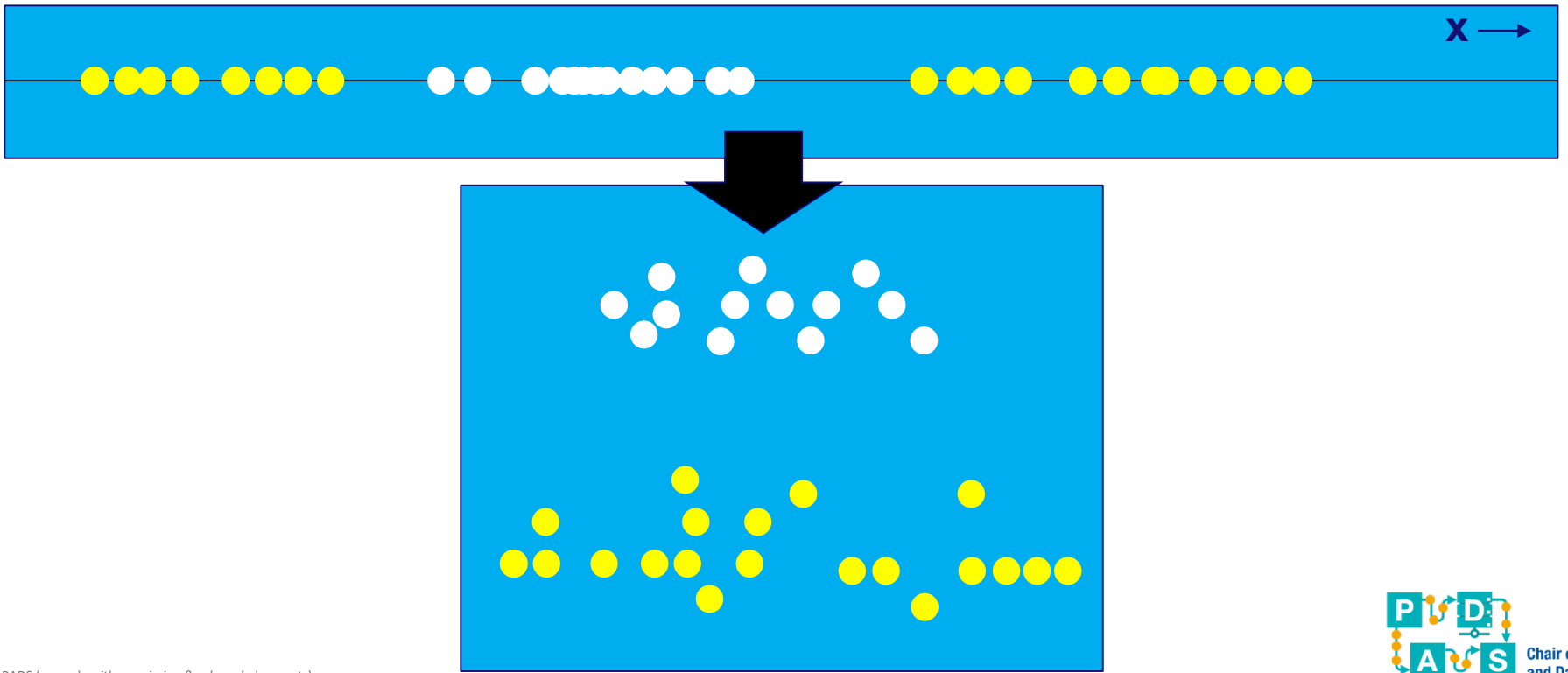


Chair of Process and Data Science

Idea

- **If data is not linearly separable in an n dimensional space, it may be linearly separable in a higher dimensional space.**

Separable in a higher dimension

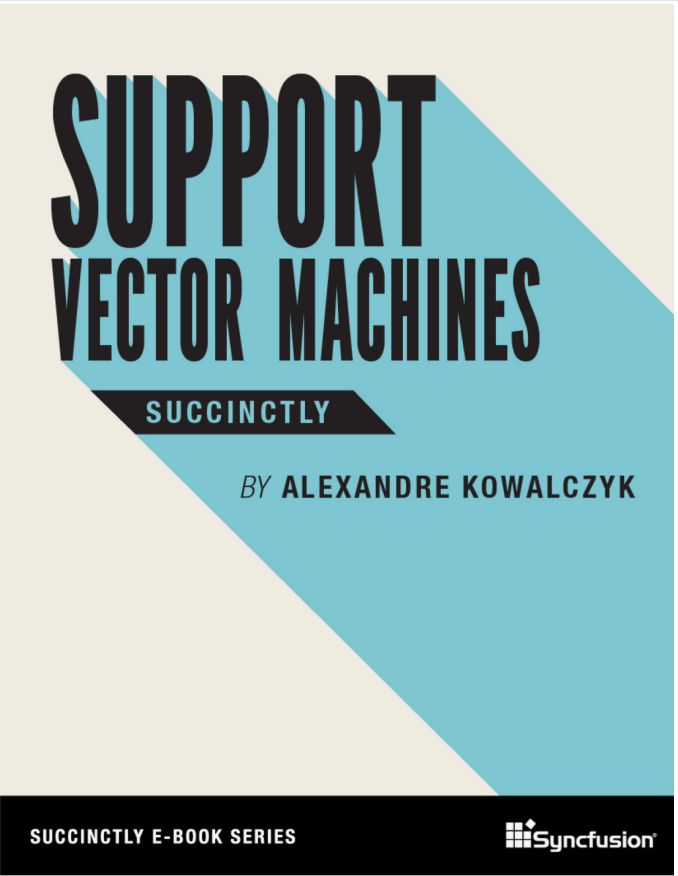


Simple right?

Not really …

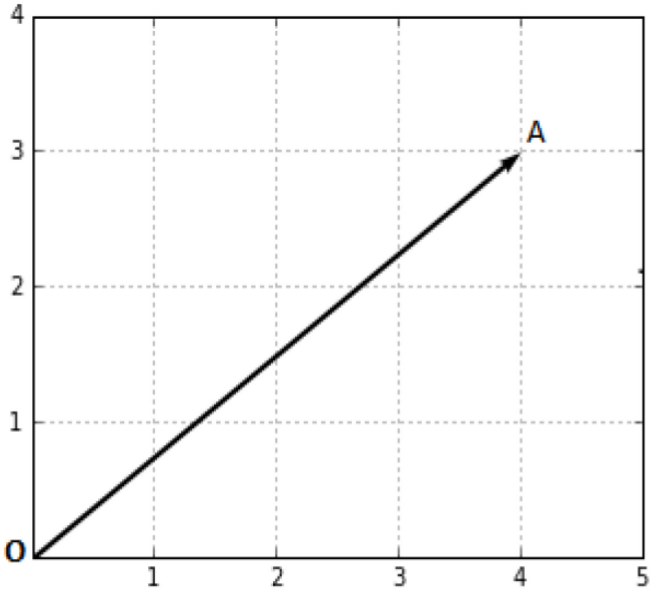
SVMs: A bit more technical

Figures on the next slides



Many figures on the next slides are taking from “Support Vector Machines Succinctly” by Alexandre Kowalczyk, Syncfusion, 2017 (with permission).

Vector length



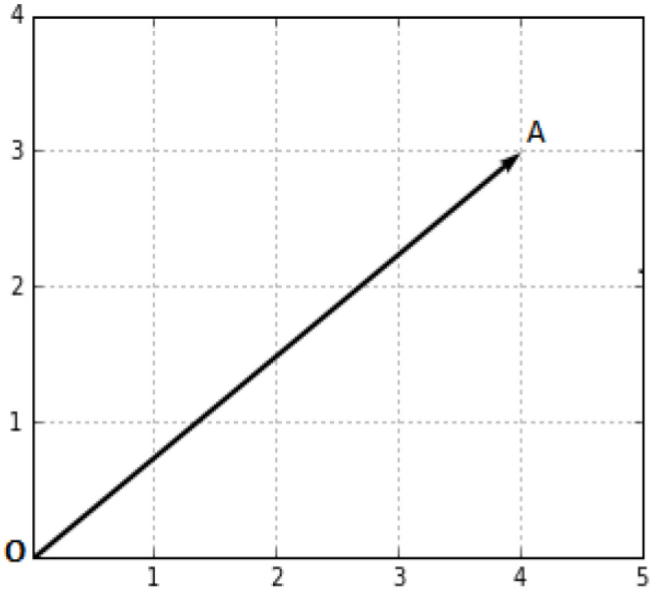
$$\vec{x} = (x_1, x_2, x_3, \dots, x_n)$$

$$\|\vec{x}\| = \sqrt{x_1^2 + x_2^2 + x_3^2 + \cdots + x_n^2}$$

$$\vec{x} = (4, 3)$$

$$\|\vec{x}\| = \sqrt{4^2 + 3^2} = \sqrt{25} = 5$$

Vector direction



$$\vec{x} = (x_1, x_2, x_3, \dots, x_n)$$

$$\vec{w} = \left(\frac{x_1}{\|\vec{x}\|}, \frac{x_2}{\|\vec{x}\|}, \frac{x_3}{\|\vec{x}\|}, \dots, \frac{x_n}{\|\vec{x}\|}\right)$$

$$\vec{x} = (4, 3)$$

$$\vec{w} = \left(\frac{4}{5}, \frac{3}{5}\right)$$

Extending length without changing direction

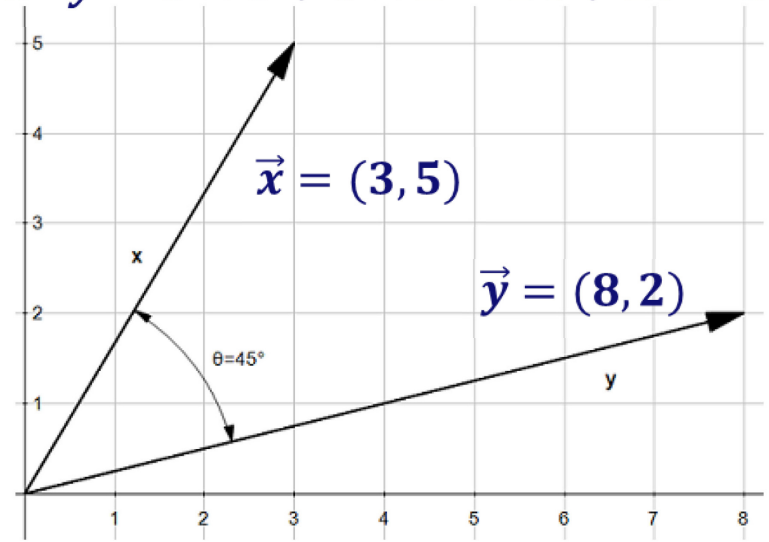
$$\vec{x} = (x_1, x_2, x_3, \dots, x_n)$$

$$q \in \mathbb{R}$$

$$q\vec{x} = (q \times x_1, q \times x_2, q \times x_3, \dots, q \times x_n)$$

Dot product

$$\vec{x} \cdot \vec{y} = 3 \times 8 + 5 \times 2 = 24 + 10 = 34$$



$$\vec{x} = (x_1, x_2, x_3, \dots, x_n)$$

$$\vec{y} = (y_1, y_2, y_3, \dots, y_n)$$

$$\vec{x} \cdot \vec{y} = \sum_{i=1}^{n} x_i y_i$$

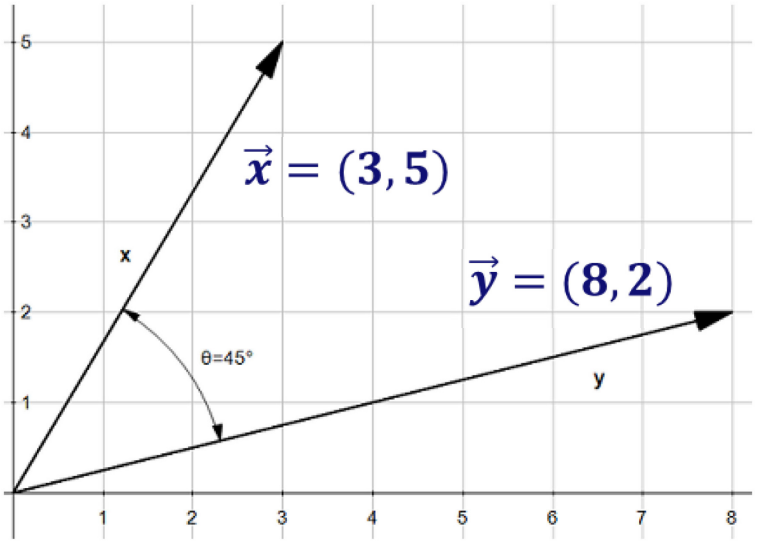
$$\vec{x} \cdot \vec{y} = x_1 y_1 + x_2 y_2 + x_3 y_3 + \dots + x_n y_n$$

Chair of Process and Data Science

Dot product (alternative formulation)

$$\vec{x} \cdot \vec{y} = 3 \times 8 + 5 \times 2 = 24 + 10 = 34$$

$$\vec{x} \cdot \vec{y} = \sqrt{3^2 + 5^2} \times \sqrt{8^2 + 2^2} \times \cos(45) = 34$$



$$\vec{x} = (x_1, x_2, x_3, \dots, x_n)$$

$$\vec{y} = (y_1, y_2, y_3, \dots, y_n)$$

$$\vec{x} \cdot \vec{y} = \|\vec{x}\| \|\vec{y}\| \cos\theta = \sum_{i=1}^{n} x_i y_i$$

Dot product (strongly influenced by angle)

$$\vec{x} = (x_1, x_2, x_3, \dots, x_n) \qquad \vec{y} = (y_1, y_2, y_3, \dots, y_n)$$

$$\vec{x} \cdot \vec{y} = \|\vec{x}\| \|\vec{y}\| \cos\theta = \sum_{i=1}^{n} x_i y_i$$

If $\theta = 0$, then $\cos\theta = 1$ and $\vec{x} \cdot \vec{y} = \|\vec{x}\| \|\vec{y}\|$

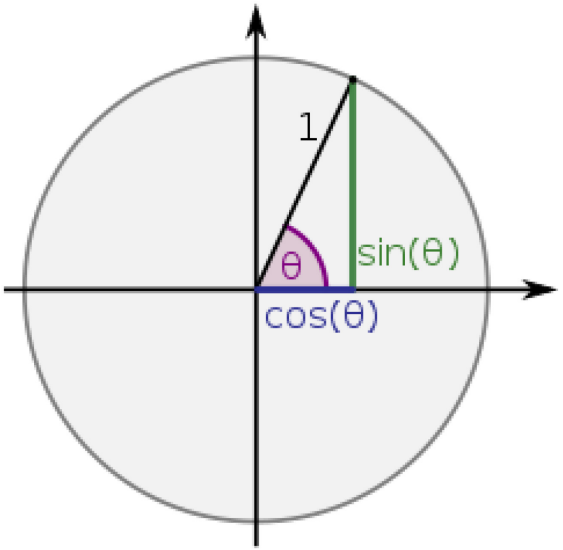
If $\theta = 45$, then $\cos\theta = 0.707$ and $\vec{x} \cdot \vec{y} = \|\vec{x}\| \|\vec{y}\|$ 0.707

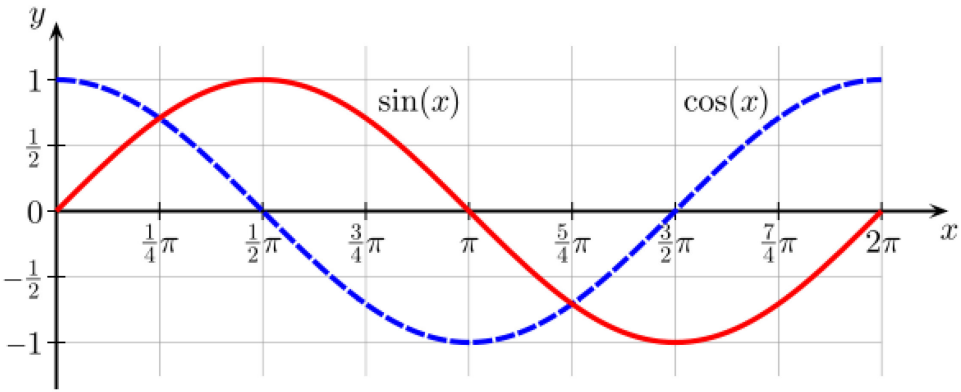
If $\theta = 90$, then $\cos\theta = 0$ and $\vec{x} \cdot \vec{y} = 0$

If $\theta = 180$, then $\cos\theta = -1$ and $\vec{x} \cdot \vec{y} = -\|\vec{x}\| \|\vec{y}\|$

If $\theta = 270$, then $\cos\theta = 0$ and $\vec{x} \cdot \vec{y} = 0$

Cosine (Latin: Cosinus)





Taken from https://en.wikipedia.org/wiki/Trigonometric_functions

Chair of Process and Data Science

Hyperplane

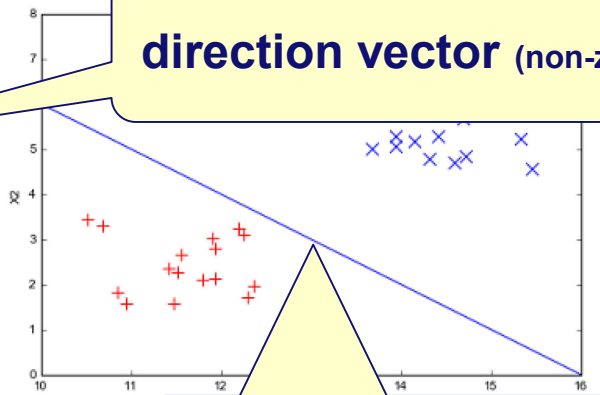
free variables

$$\vec{x} = (x_1, x_2, x_3, \dots, x_n)$$

$$\vec{w} = (w_1, w_2, w_3, \dots, w_n)$$

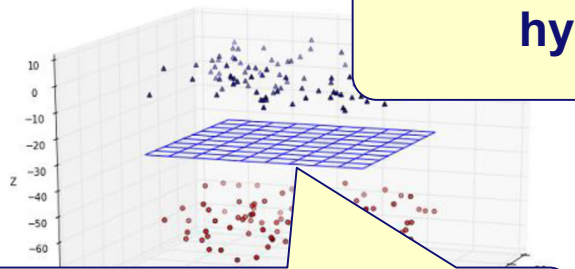
direction vector (non-zero)

$$\vec{w} \cdot \vec{x} + b = 0$$



hyperplane

defines hyperplane



hyperplane = set of points

Chair of Process and Data Science

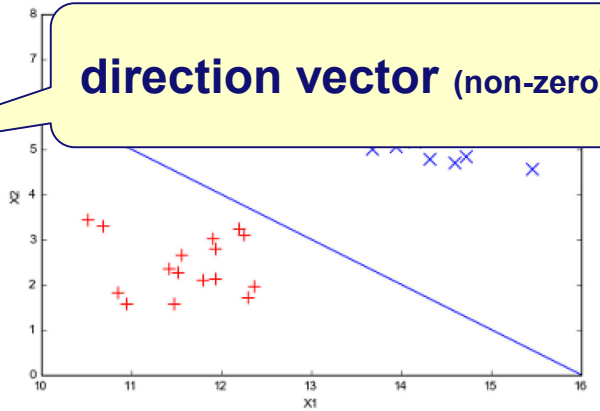
Hyperplane

free variables

$$\vec{x} = (x_1, x_2, x_3, \dots, x_n)$$

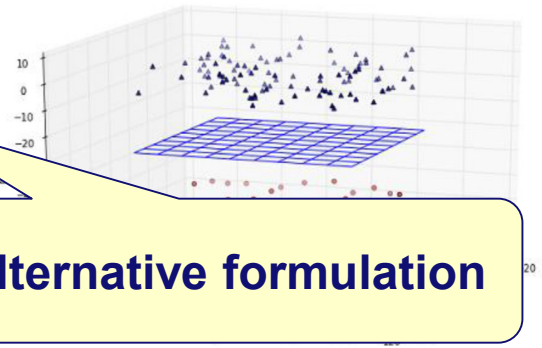
$$\vec{w} = (w_1, w_2, w_3, \dots, w_n)$$

direction vector (non-zero)



$$\vec{w} \cdot \vec{x} + b = 0$$

$$\sum_{i=1}^{n} w_i x_i + b = 0$$



alternative formulation

Chair of Process and Data Science

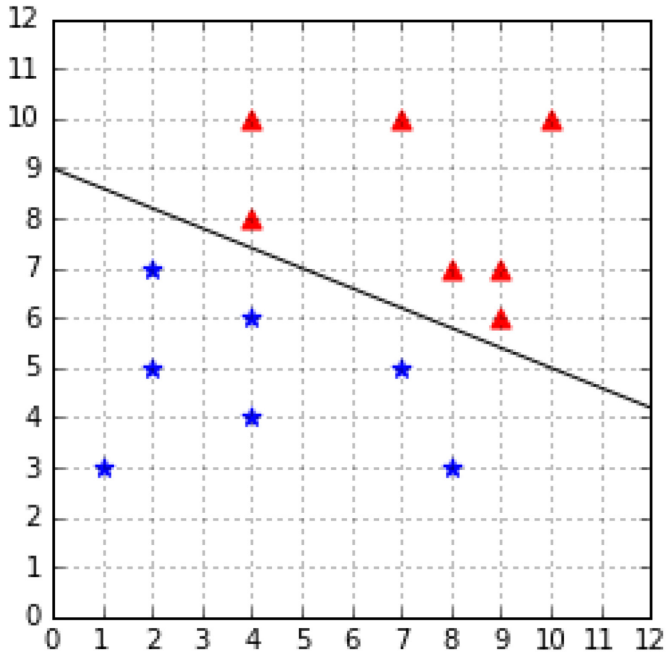
Hyperplane is fully defined by w and b

$$\vec{x} = (x_1, x_2, x_3, \dots, x_n)$$

$$\vec{w} = (w_1, w_2, w_3, \dots, w_n)$$

$$\vec{w} \cdot \vec{x} + b = 0$$

$$\sum_{i=1}^{n} w_i x_i + b = 0$$

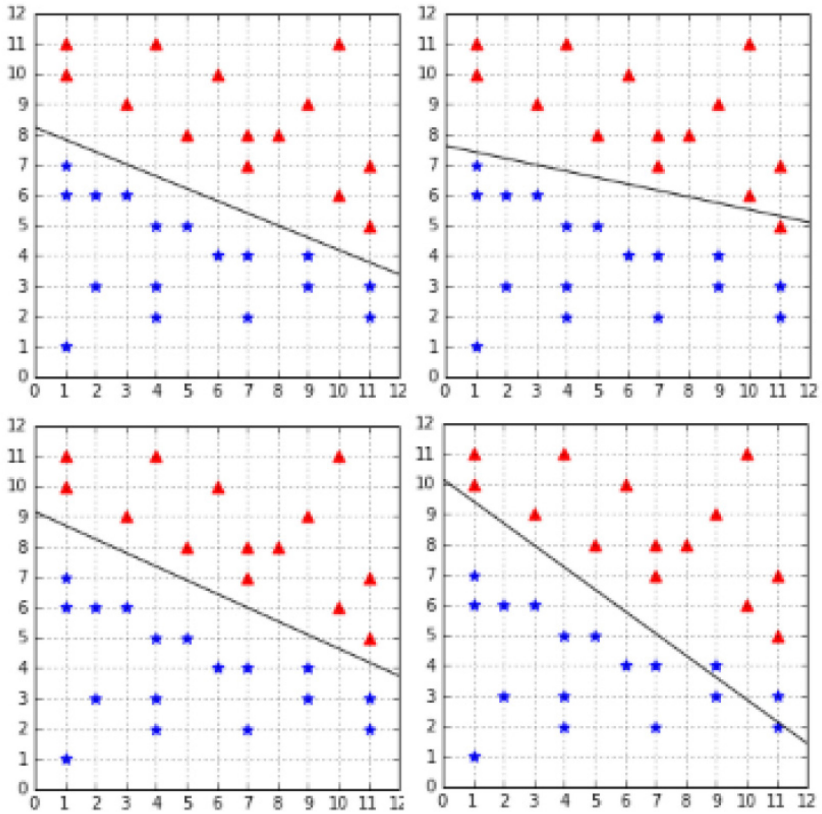


$$\vec{w} = (0.4, 1.0)$$

$$b = -9$$

Chair of Process and Data Science

Different hyperplanes



A hyperplane is fully defined by w and b, but there are infinitely many ways to define the same hyperplane

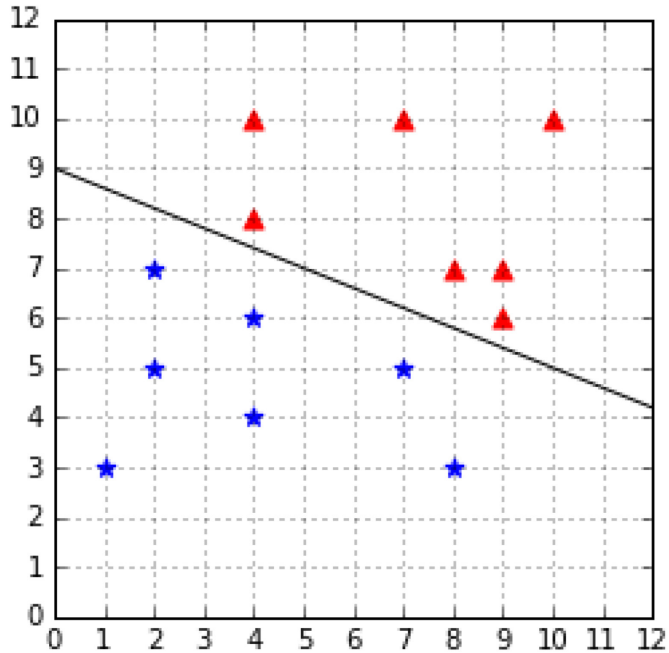
$$\vec{x} = (x_1, x_2, x_3, \dots, x_n)$$

$$\vec{w} = (w_1, w_2, w_3, \dots, w_n)$$

$$\vec{w} \cdot \vec{x} + b = 0$$

$$q\vec{w} \cdot \vec{x} + qb = 0$$

If $q \in \mathbb{R}$ and $q \neq 0$, then both are equivalent !



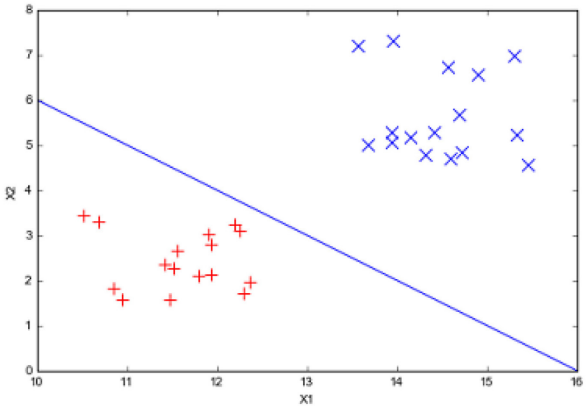
$$\vec{w} = (4, 10)$$

$$b = -90$$

Chair of Process and Data Science

Hyperplane splits the set of data points

$$\vec{x}_i = (x_{i,1}, x_{i,2}, x_{i,3}, \dots, x_{i,n})$$
$$\vec{w} = (w_1, w_2, w_3, \dots, w_n)$$



$$+ \quad \vec{w} \cdot \vec{x}_i + b \geq 0$$
$$- \quad \vec{w} \cdot \vec{x}_i + b < 0$$

$$\sum_{j=1}^{n} w_i x_{i,j,} + b = 0$$

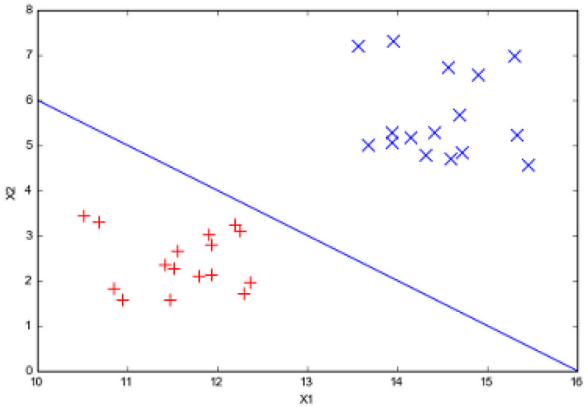
Chair of Process and Data Science

Hyperplane splits the set of data points

$$\vec{x}_i = (x_{i,1}, x_{i,2}, x_{i,3}, \dots, x_{i,n})$$

$$\vec{w} = (w_1, w_2, w_3, \dots, w_n)$$

- \+ $\vec{w} \cdot \vec{x}_i + b \geq 0$
- \- $\vec{w} \cdot \vec{x}_i + b < 0$



$$+ \sum_{j=1}^{n} w_i x_{i,j} + b \geq 0 \qquad - \sum_{j=1}^{n} w_i x_{i,j} + b < 0$$

Chair of Process and Data Science

Hyperplane splits the set of data points

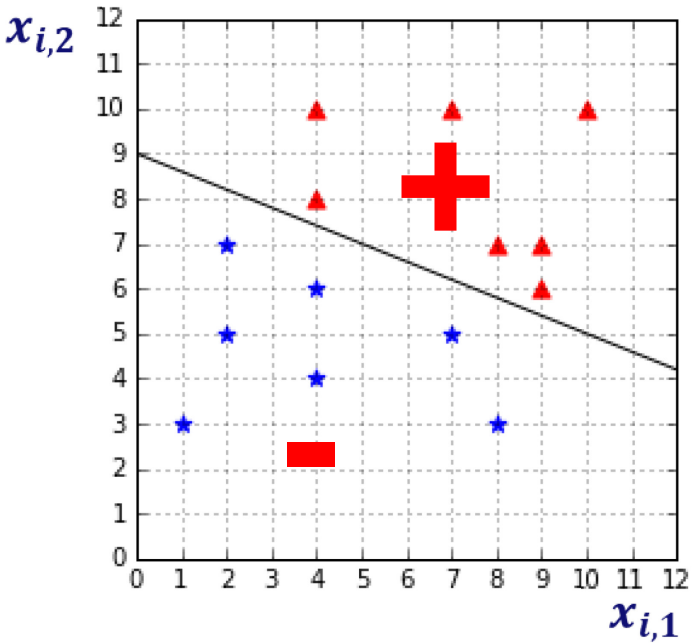
$$\vec{x}_i = (x_{i,1}, x_{i,2})$$
$$\vec{w} = (4, 10)$$

\+ $\vec{w} \cdot \vec{x}_i + b = 4 \times x_{i,1} + 10 \times x_{i,2} - 90 \geq 0$

\- $\vec{w} \cdot \vec{x}_i + b = 4 \times x_{i,1} + 10 \times x_{i,2} - 90 < 0$

\+ $x_{i,2} \geq 9 - 0.4 \times x_{i,1}$

\- $x_{i,2} < 9 - 0.4 \times x_{i,1}$

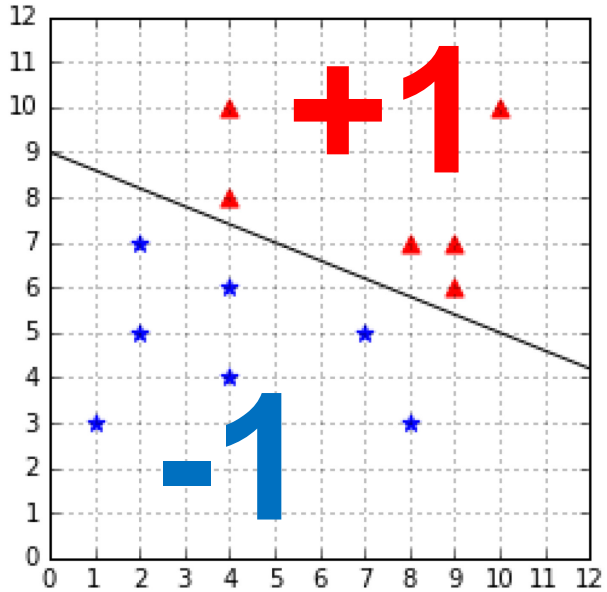


Chair of Process and Data Science

Introduce two classes +1 and -1

$$\vec{x}_i = (x_{i,1}, x_{i,2}, x_{i,3}, \dots, x_{i,n})$$

$$y_i \in \{-1, 1\}$$



Let's find a hyperplane that separates the -1 instances from the +1 instances!

Let's assume we have a hyperplane defined by $\vec{w}$ and b that separates the -1 instances from the +1 instances!

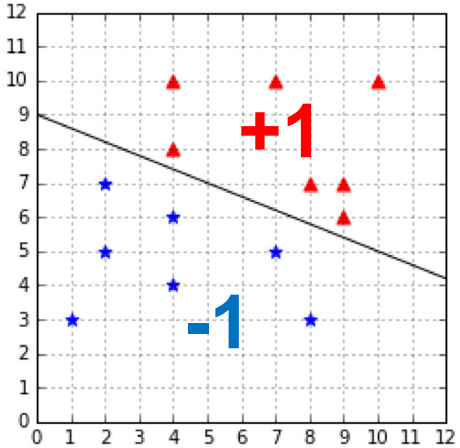
Chair of Process and Data Science

Hyperplane $(\vec{w}, b)$ perfectly separates if

If for all instances $(\vec{x}_i, y_i)$:

$\vec{x}_i = (x_{i,1}, x_{i,2}, x_{i,3}, \ldots, x_{i,n})$

$y_i \in \{-1, 1\}$

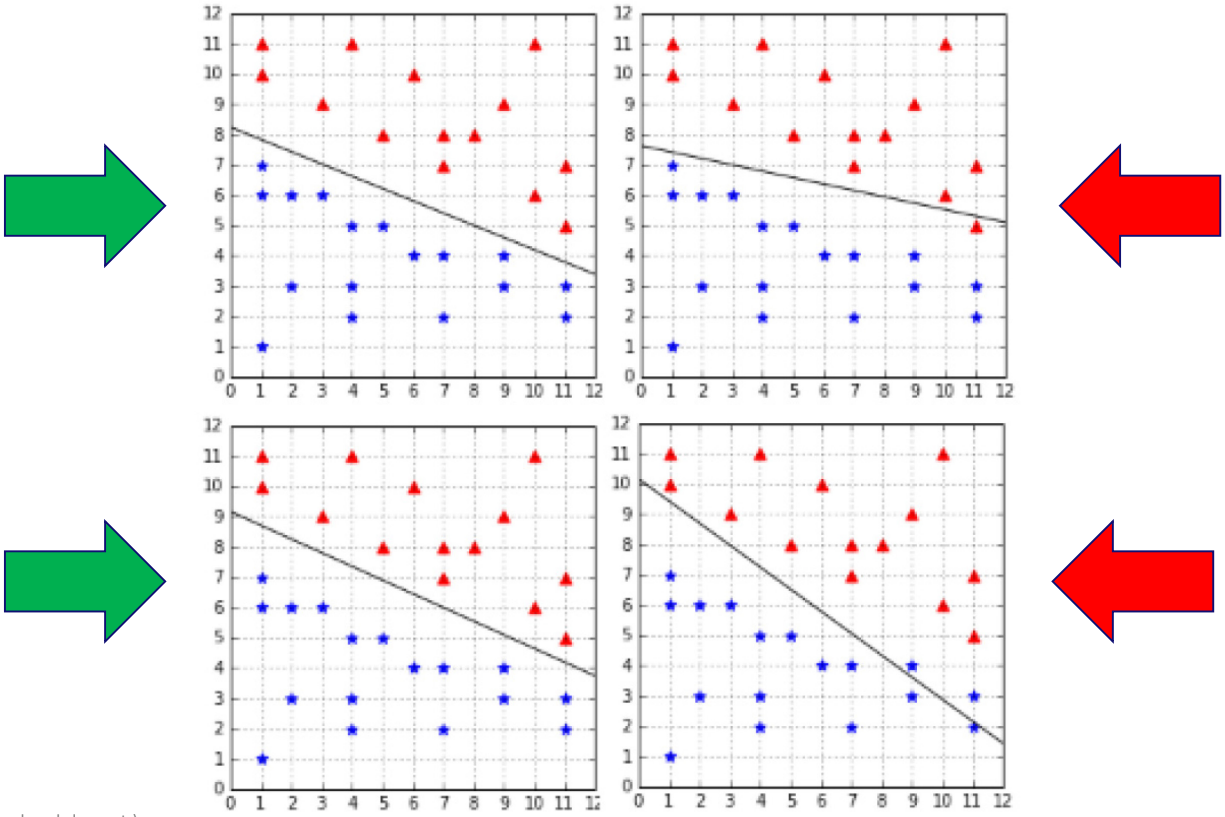


If $y_i = +1$, then $\vec{w} \cdot \vec{x}_i + b \geq 0$.

If $y_i = -1$, then $\vec{w} \cdot \vec{x}_i + b < 0$.

Chair of Process and Data Science

Which hyperplanes perfectly separate the instances?



Chair of Process and Data Science

Initial task: find $(\vec{w}, b)$ such that …

Given a set of m instances

$$\{(\vec{x}_i, y_i) \in \mathbb{R}^n \times \{-1, 1\} | 1 \leq i \leq m\}$$

Find, $(\vec{w}, b)$ such that for any $1 \leq i \leq m$:

If $y_i = +1$, then $\vec{w} \cdot \vec{x}_i + b \geq 0$.

If $y_i = -1$, then $\vec{w} \cdot \vec{x}_i + b < 0$.

There may be many solutions

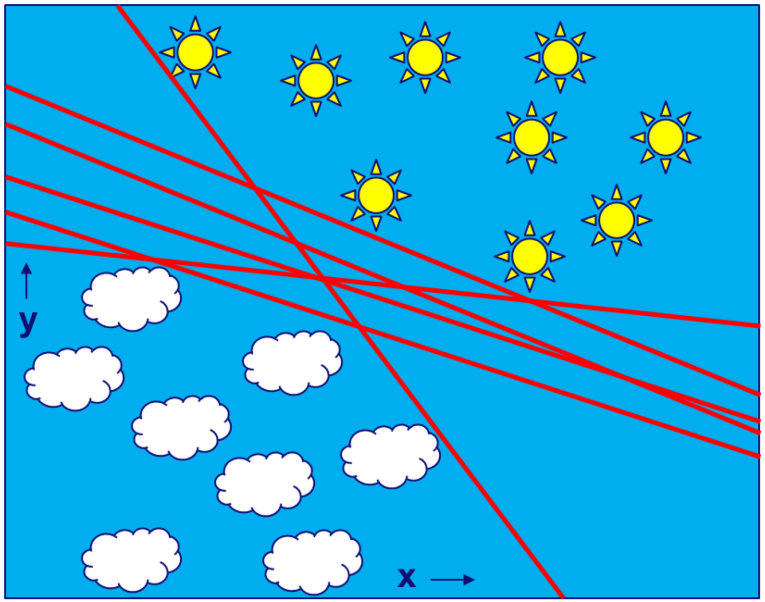
Given a set of m instances

$$\{(\vec{x}_i, y_i) \in \mathbb{R}^n \times \{-1, 1\} | 1 \leq i \leq m\}$$

Find, $(\vec{w}, b)$ such that for any $1 \leq i \leq m$:

If $y_i = +1$, then $\vec{w} \cdot \vec{x}_i + b \geq 0$.

If $y_i = -1$, then $\vec{w} \cdot \vec{x}_i + b < 0$.



Which one to pick?

Th

We would like to maximize the distance of the hyperplane to the nearest point

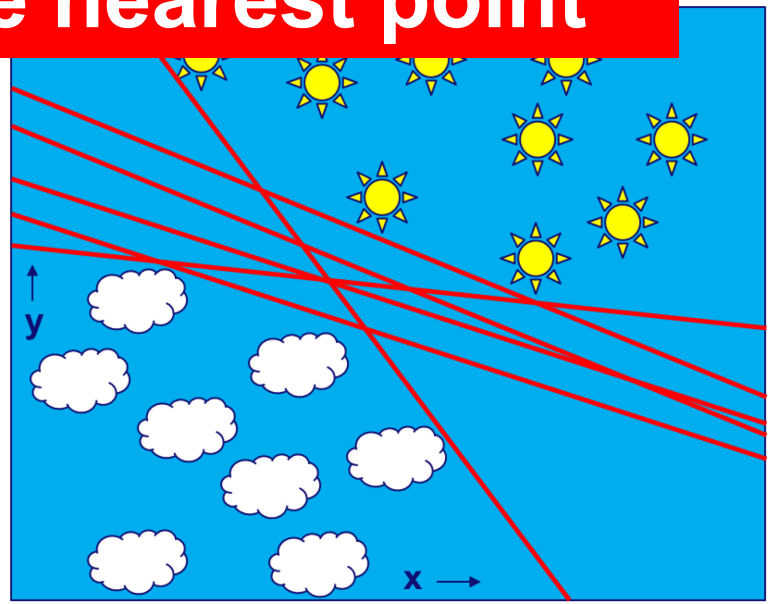
Given a set of m instances

$$\{(\vec{x}_i, y_i) \in \mathbb{R}^n \times \{-1, 1\} | 1 \leq i \leq m\}$$

Find, $(\vec{w}, b)$ such that for any $1 \leq i \leq m$:

If $y_i = +1$, then $\vec{w} \cdot \vec{x}_i + b \geq 0$.

If $y_i = -1$, then $\vec{w} \cdot \vec{x}_i + b < 0$.



Give a set of

Which one to pick?

Distance between a point and the hyperplane

$$\vec{x}_1 = \vec{x}_2 + \lambda\vec{w}$$

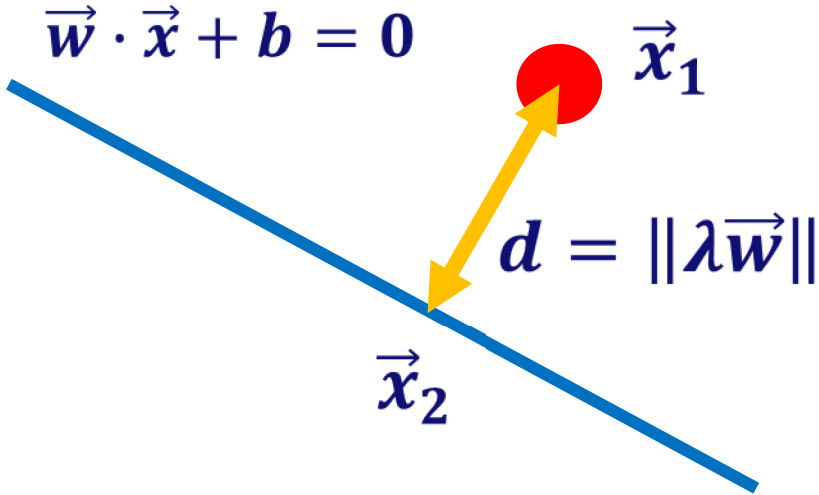
closest point on hyperplane $\vec{x}_2$ can be found by following an orthogonal path

$$\vec{w} \cdot \vec{x}_2 + b = 0$$

closest point $\vec{x}_2$ on hyperplane is a solution

hence

$$\vec{w} \cdot (\vec{x}_1 - \lambda\vec{w}) + b = 0$$



$$d = \|\lambda\vec{w}\|$$

Distance between a point and the hyperplane

$$d = \|\lambda \vec{w}\|$$

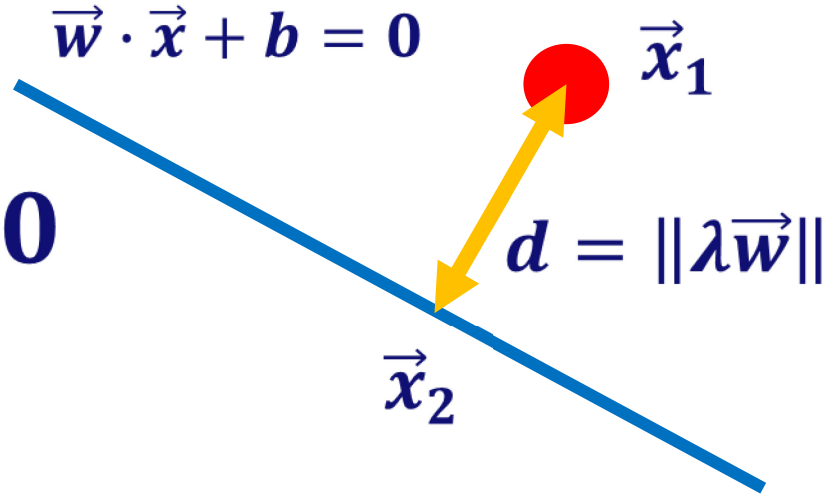
$$\vec{w} \cdot (\vec{x}_1 - \lambda \vec{w}) + b = 0$$

hence

$$\vec{w} \cdot \vec{x}_1 + b = \lambda \vec{w} \cdot \vec{w}$$

hence

$$\lambda \|\vec{w}\|^2 = \vec{w} \cdot \vec{x}_1 + b$$

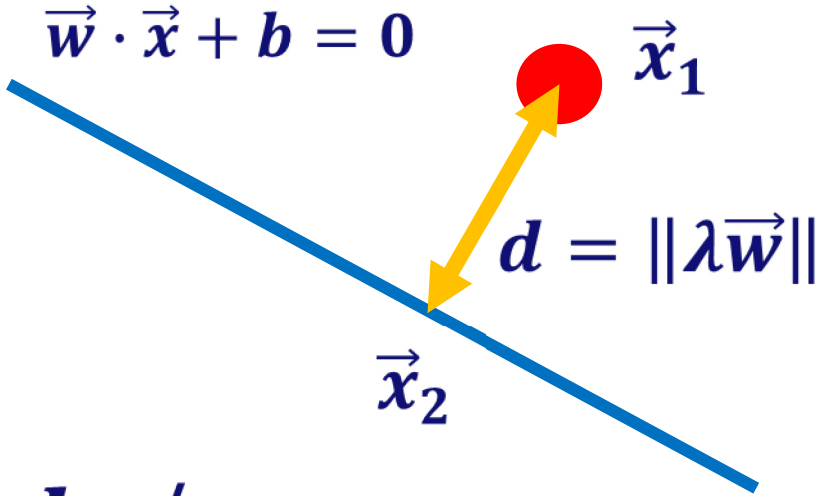


Chair of Process and Data Science

Distance between a point and the hyperplane

$$d = \|\lambda \vec{w}\|$$

$$\lambda \|\vec{w}\|^2 = \vec{w} \cdot \vec{x}_1 + b$$



hence

$$d = \|\lambda \vec{w}\| = \vec{w} \cdot \vec{x}_1 + b \Big/ \|\vec{w}\|$$

Goal find $(\vec{w}, b)$ such that …

Given $\{(\vec{x}_i, y_i) \in \mathbb{R}^n \times \{-1, 1\} | 1 \leq i \leq m\}$

Find, $(\vec{w}, b)$ such that for any $1 \leq i \leq m$:

If $y_i = +1$, then $\vec{w} \cdot \vec{x}_i + b \geq 0$.

If $y_i = -1$, then $\vec{w} \cdot \vec{x}_i + b < 0$.

And maximize $\min_{1 \leq i \leq m} d_i$

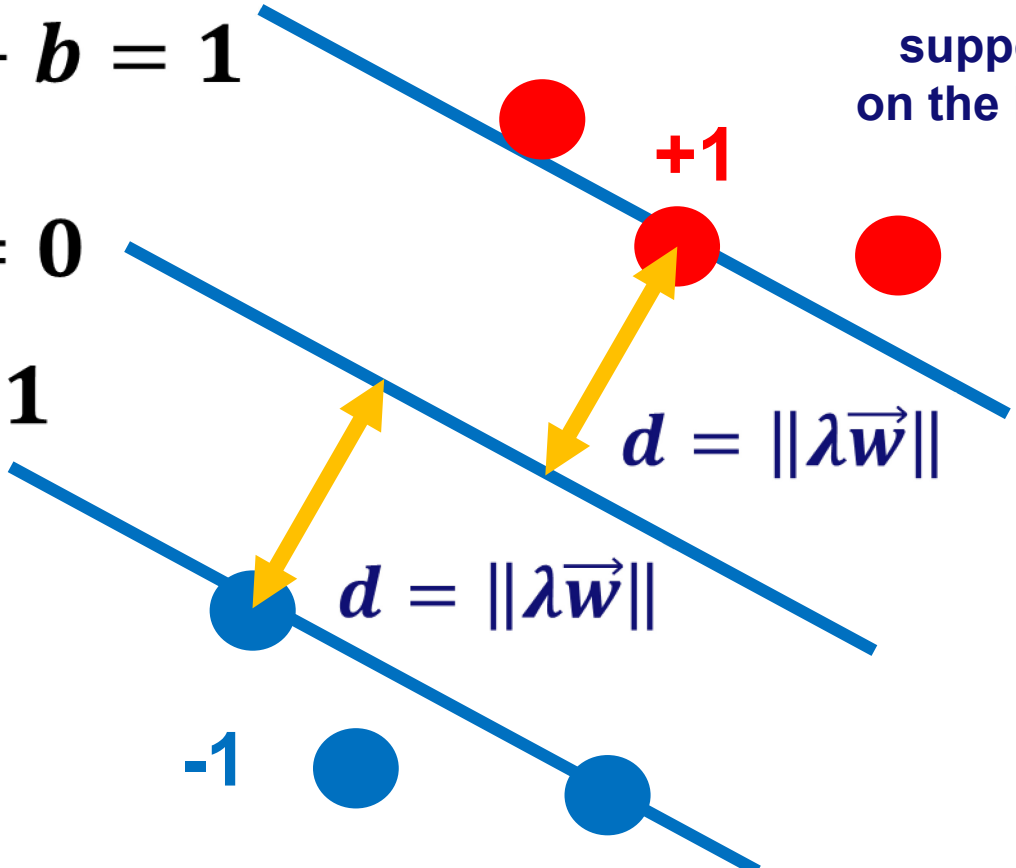
$$d_i = \|\lambda \vec{w}\| = \frac{\vec{w} \cdot \vec{x}_i + b}{\|\vec{w}\|}$$

An alternative formulation

$$\vec{w} \cdot \vec{x} + b = 1$$

$$\vec{w} \cdot \vec{x} + b = 0$$

$$\vec{w} \cdot \vec{x} + b = -1$$



support vectors are on the lines indicated (by rescaling)

This can be done because:

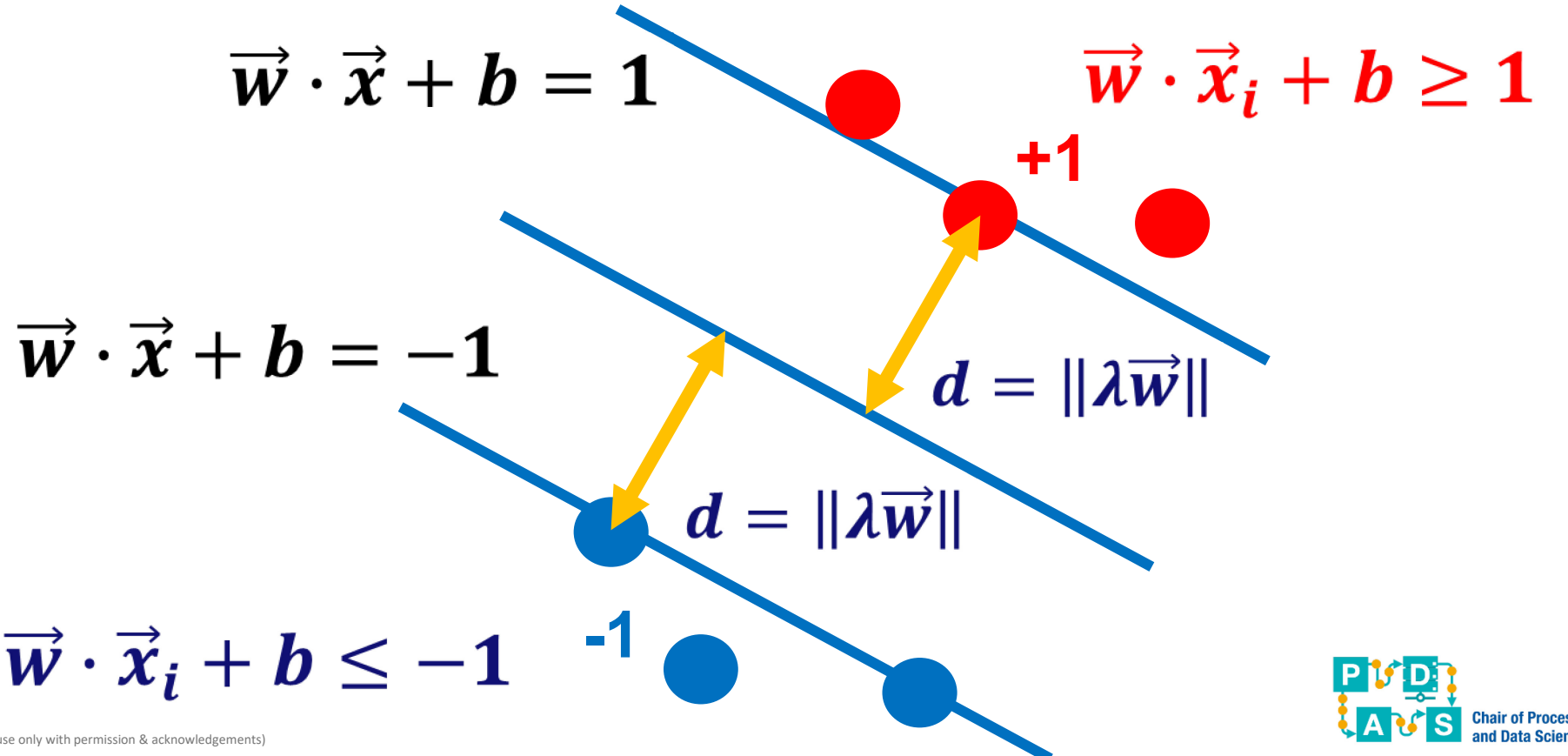
$$\vec{w} \cdot \vec{x} + b = 0$$

$$q\vec{w} \cdot \vec{x} + qb = 0$$

If $q \in \mathbb{R}$ and $q \neq 0$, then both are equivalent !

Chair of Process and Data Science

An alternative formulation



Chair of Process and Data Science

An alternative formulation

$$\vec{w} \cdot \vec{x}_i + b \geq 1 \quad \text{if } y_i = +1$$

$$\vec{w} \cdot \vec{x}_i + b \leq -1 \text{ if } y_i = -1$$

Can be combined into:

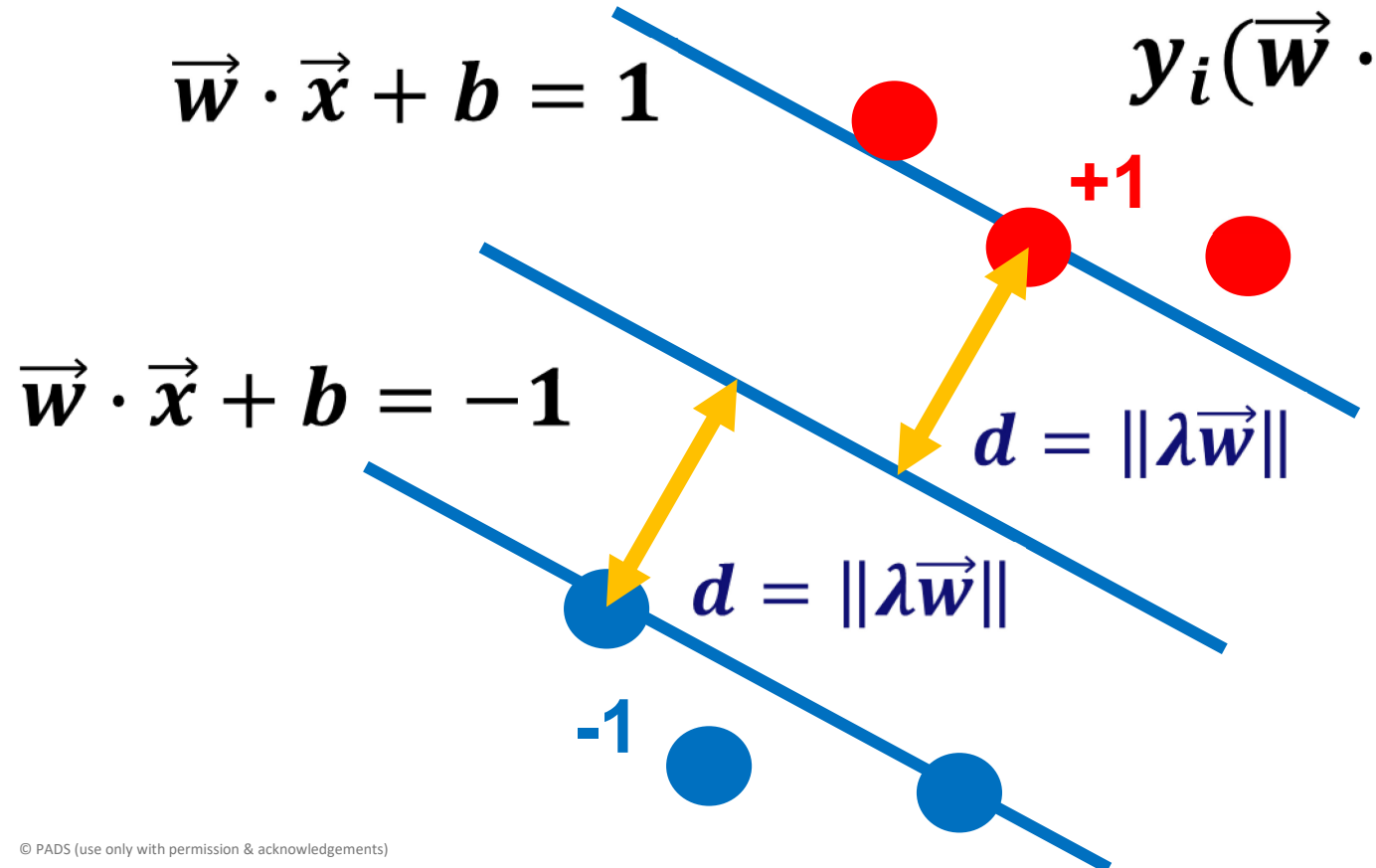
$$y_i(\vec{w} \cdot \vec{x}_i + b) \geq 1 \quad \text{for any } i$$

Goal maximize $d = \|\lambda\vec{w}\|$ such that:

$$\vec{w} \cdot \vec{x} + b = 1$$

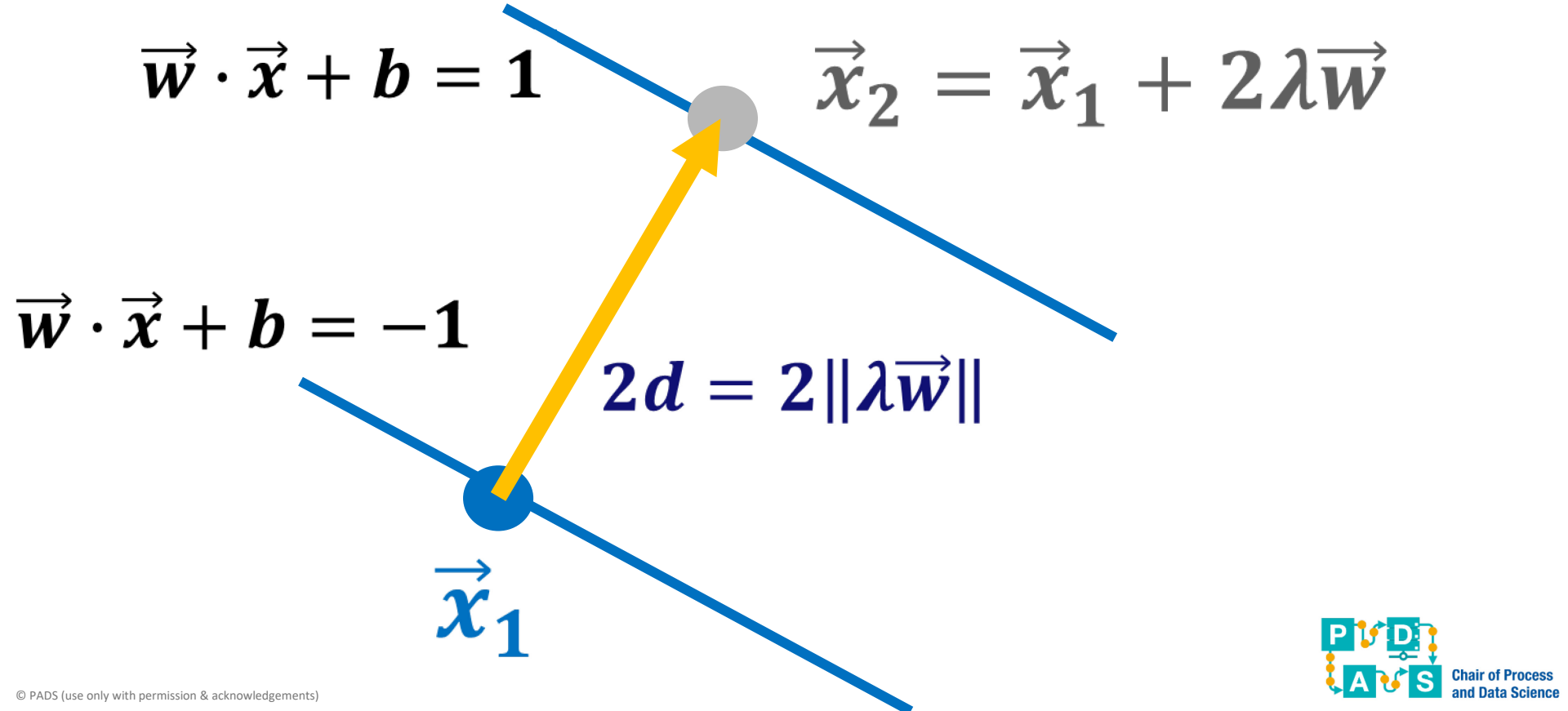
$$y_i(\vec{w} \cdot \vec{x}_i + b) \geq 1$$

for any i



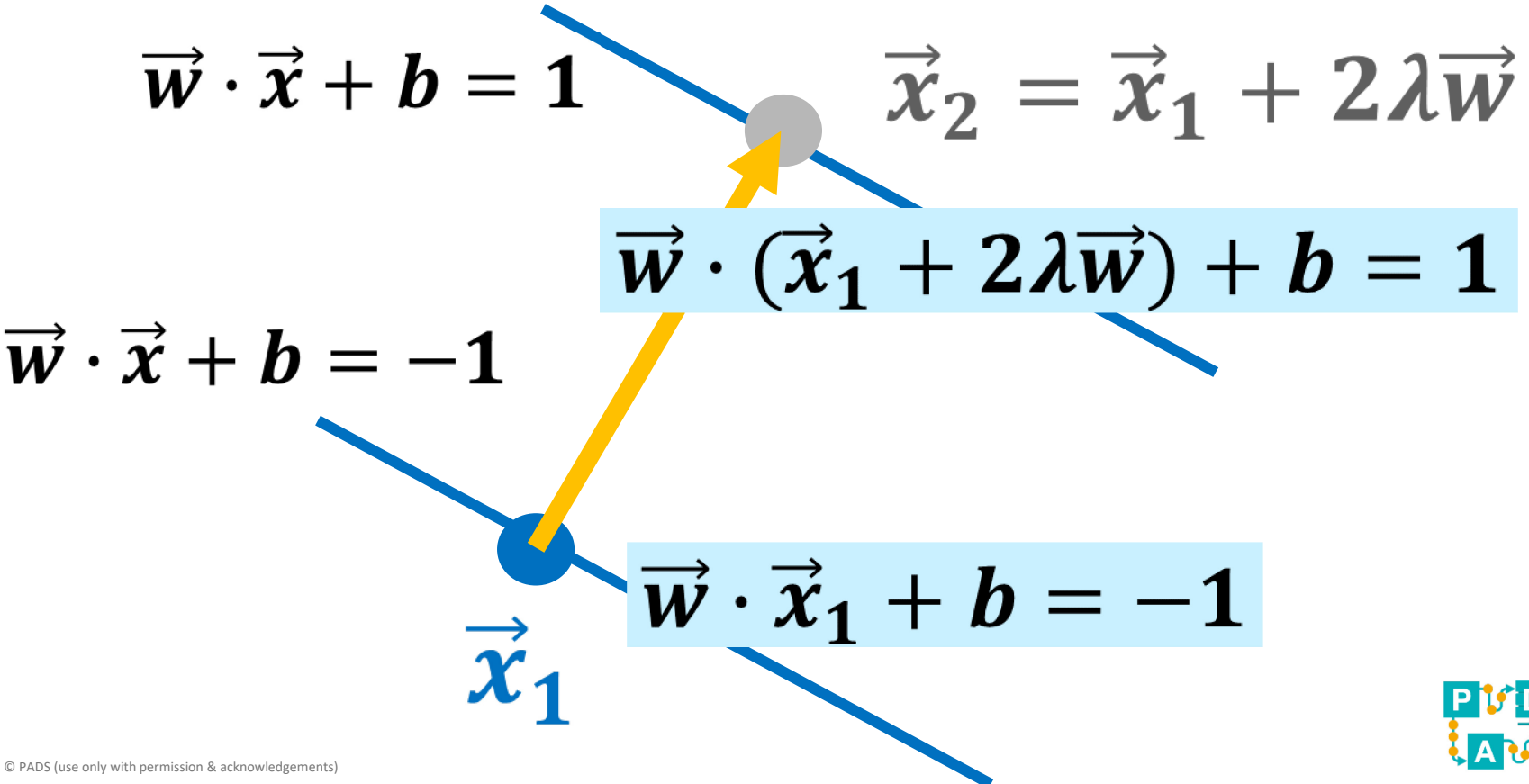
Chair of Process and Data Science

Take an arbitrary support vector



Chair of Process and Data Science

Take an arbitrary support vector



Some reasoning

$$\vec{w} \cdot (\vec{x}_1 + 2\lambda\vec{w}) + b = 1$$

$$\vec{w} \cdot \vec{x}_1 + b = -1$$

$$\vec{w} \cdot \vec{x}_1 + b + \vec{w} \cdot 2\lambda\vec{w} = 1$$

$$-1 + \vec{w} \cdot 2\lambda\vec{w} = 1$$

$$\vec{w} \cdot 2\lambda\vec{w} = 2$$

Recall the goal is to maximize $d = \|\lambda\vec{w}\|$

$$\lambda = {}^{1}/_{\vec{w} \cdot \vec{w}} = {}^{1}/_{\|\vec{w}\|^2}$$

Maximize $d = \|\lambda\vec{w}\|$

$$\lambda = {}^{1}/_{\|\vec{w}\|^2} \qquad d = \|\lambda\vec{w}\| = {}^{1}/_{\|\vec{w}\|}$$

Hence, maximizing $d = \|\lambda\vec{w}\|$ corresponds to minimizing $\|\vec{w}\|$

Reformulated problem

Given a set of m instances

$$\{(\vec{x}_i, y_i) \in \mathbb{R}^n \times \{-1, 1\} | 1 \leq i \leq m\}$$

$$\min_{\vec{w},b} \|\vec{w}\| \text{ such that}$$

$$y_i(\vec{w} \cdot \vec{x}_i + b) \geq 1 \text{ for any } i$$

Common variant (for technical reasons)

Given a set of m instances

$$\{(\vec{x}_i, y_i) \in \mathbb{R}^n \times \{-1, 1\} | 1 \leq i \leq m\}$$

$$\min_{\vec{w},b} \frac{1}{2} \|\vec{w}\|^2 \text{ such that}$$

$$y_i(\vec{w} \cdot \vec{x}_i + b) \geq 1 \text{ for any } i$$

Convex quadratic optimization problem

$$\min_{\vec{w},b} \frac{1}{2} \|\vec{w}\|^2 \text{ such that}$$

$$y_i(\vec{w} \cdot \vec{x}_i + b) \geq 1 \text{ for any } i$$

This convex quadratic optimization problem is easier to solve than the original formulation (maximize $\min_{1 \leq i \leq m} d_i$).

Solving this is outside scope, just assume there are techniques to do so.

Another reformulation of the problem

Wolfe dual Lagrangian problem:

$$\mathbf{w} = \sum_{i=1}^{m} \alpha_i y_i \mathbf{x}_i$$

$$\alpha_i \geq 0, \quad i = 1, \ldots, m$$

$$\underset{\alpha}{\text{maximize}} \quad \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j$$

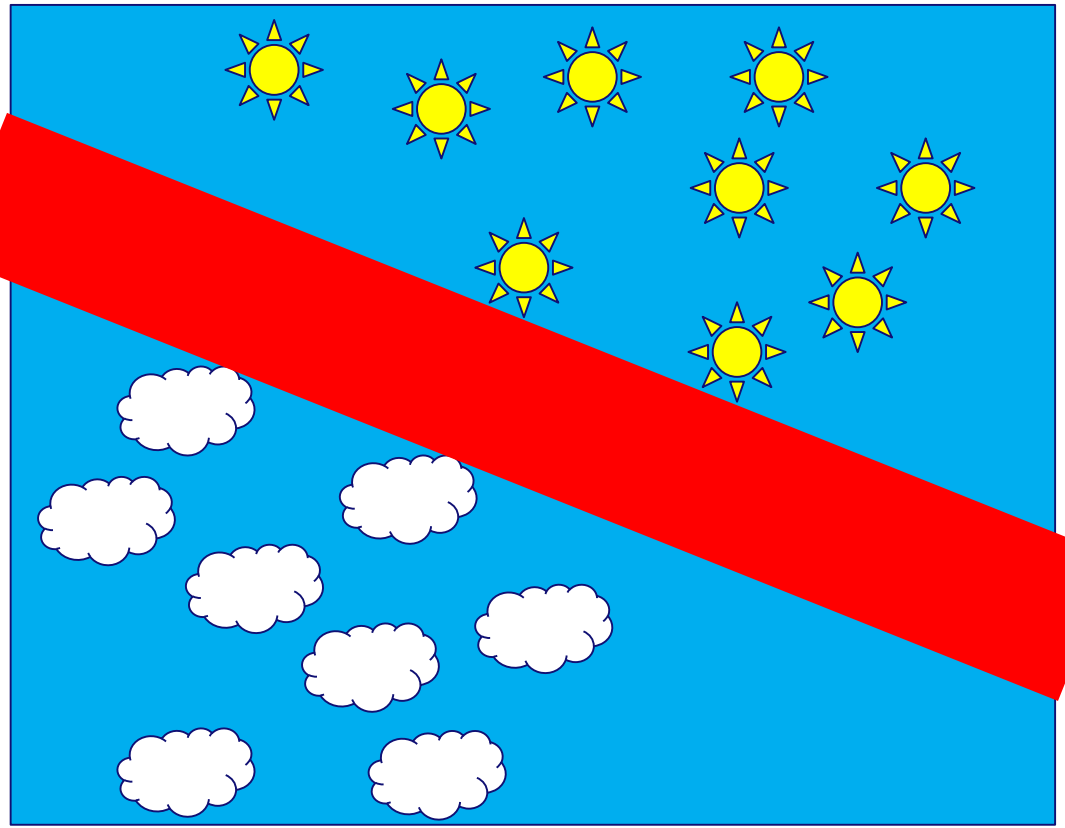
$$\text{subject to} \quad \alpha_i \geq 0, \text{ for any } i = 1, \ldots, m$$

$$\sum_{i=1}^{m} \alpha_i y_i = 0$$

No need to understand derivation, see "Support Vector Machines Succinctly" by Alexandre Kowalczyk, Syncfusion, 2017 for details.

This alternative formulation is important for "kernel trick".

Summary



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

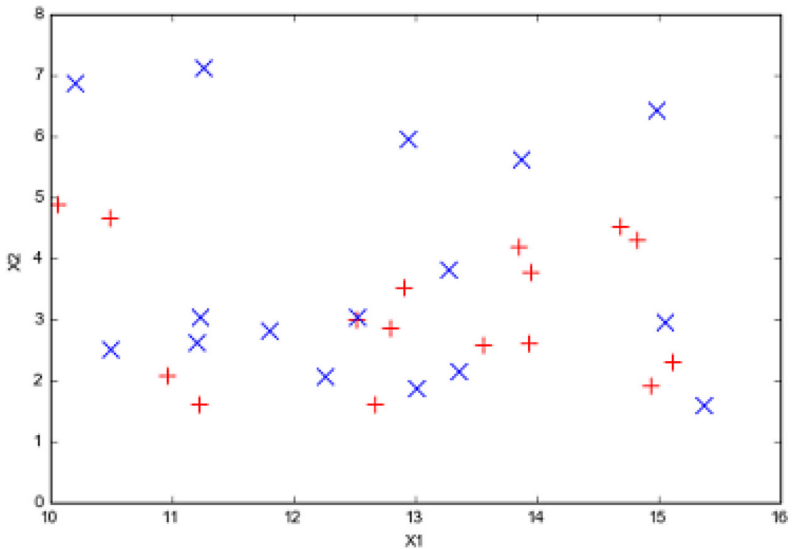
such that

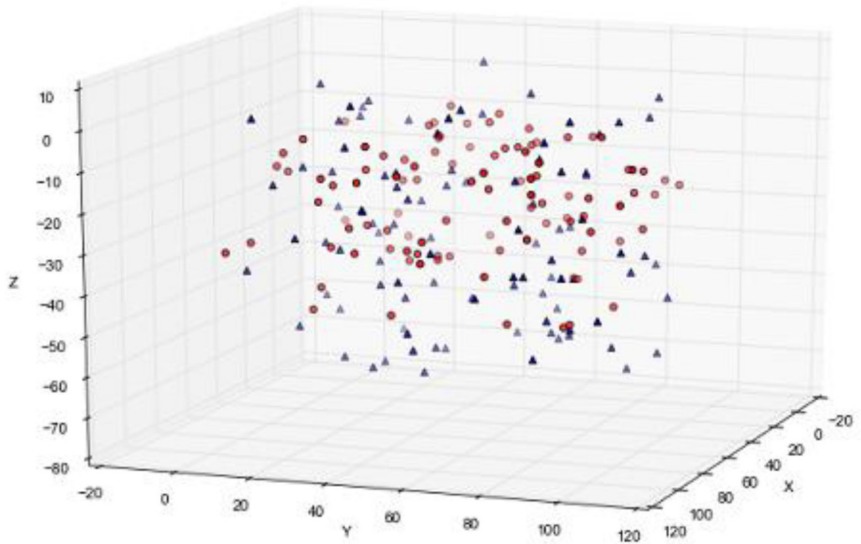
$$y_i(\vec{w} \cdot \vec{x}_i + b) \geq 1$$

for any i

Soft-margin SVMs

What if the instances are not separable?

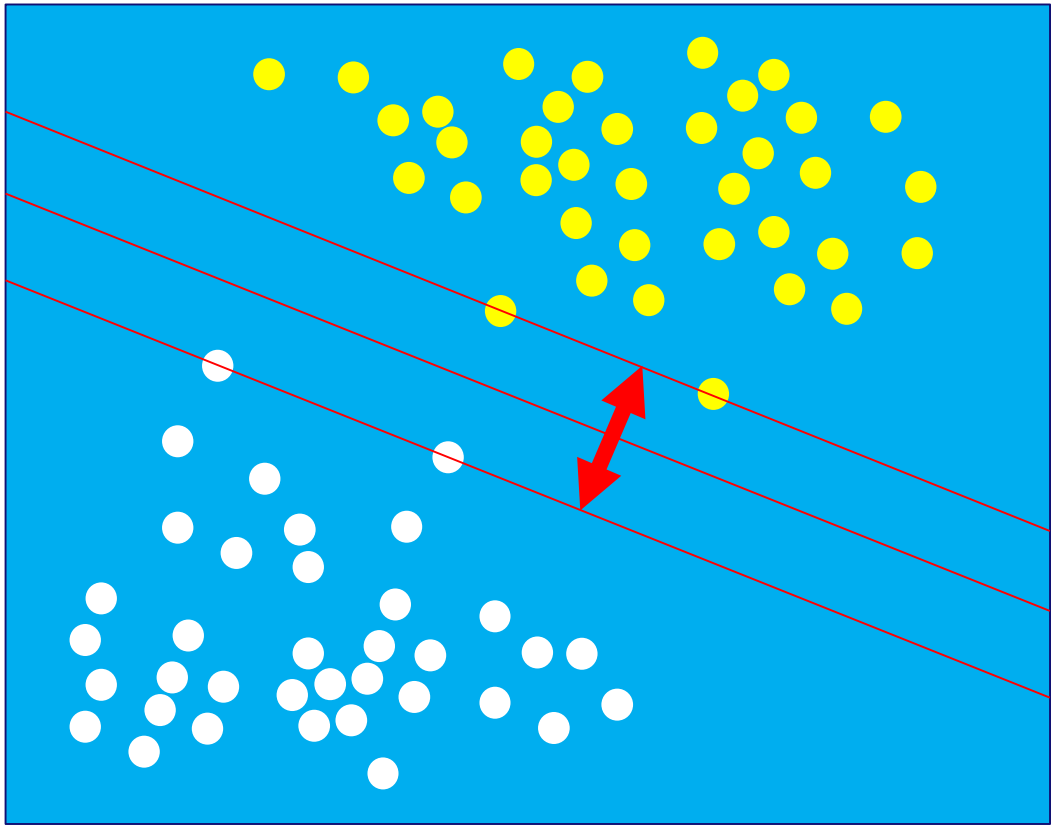




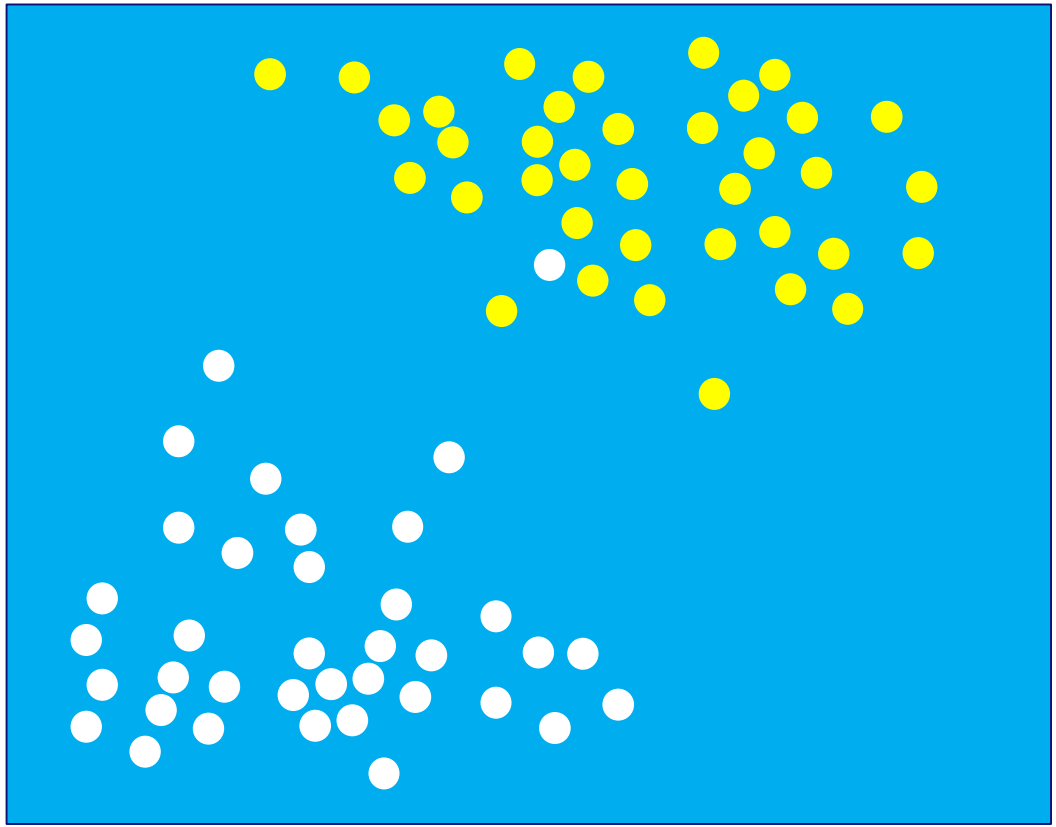
The above examples are not separable (even after removing outliers). There is no reasonable hyperplane.

Chair of Process and Data Science

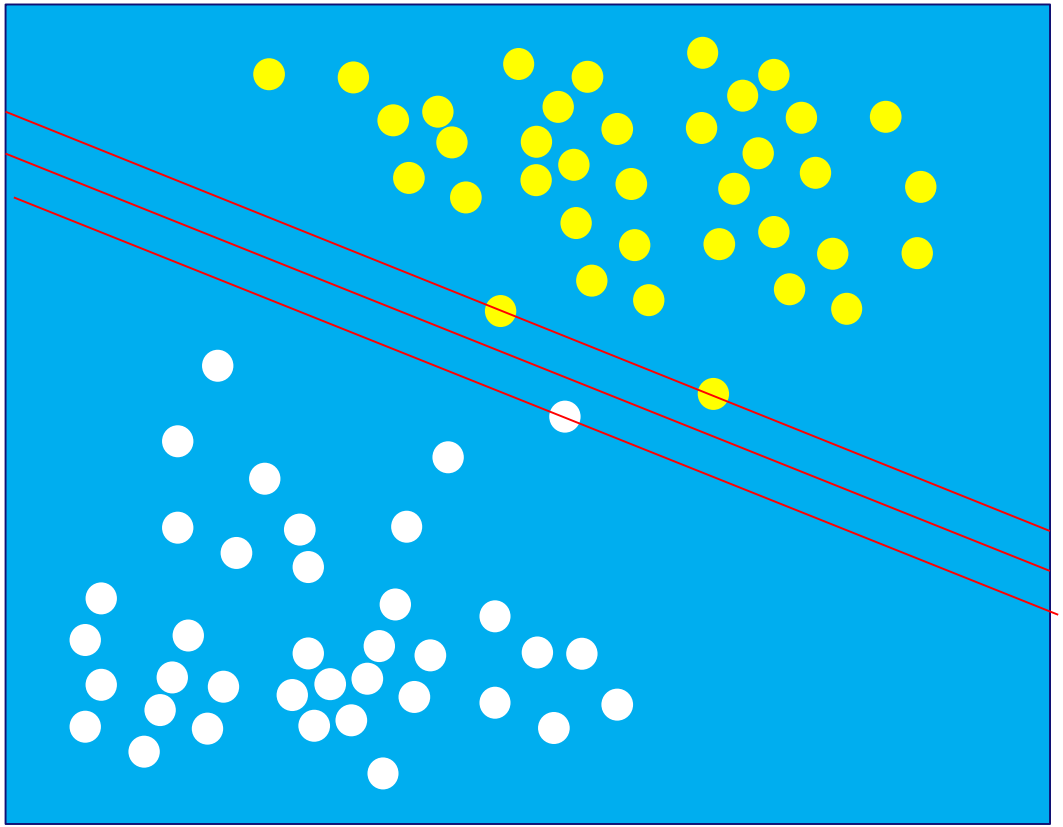
Data that are nicely separable



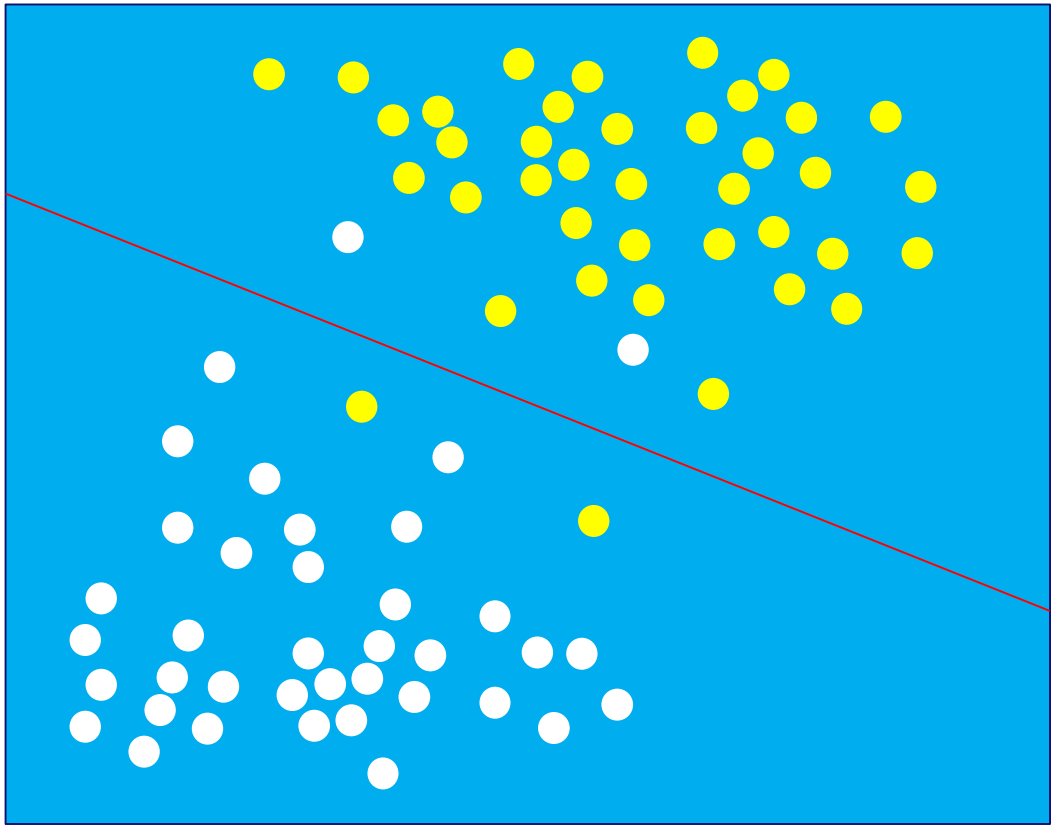
One outlier can destroy it all!



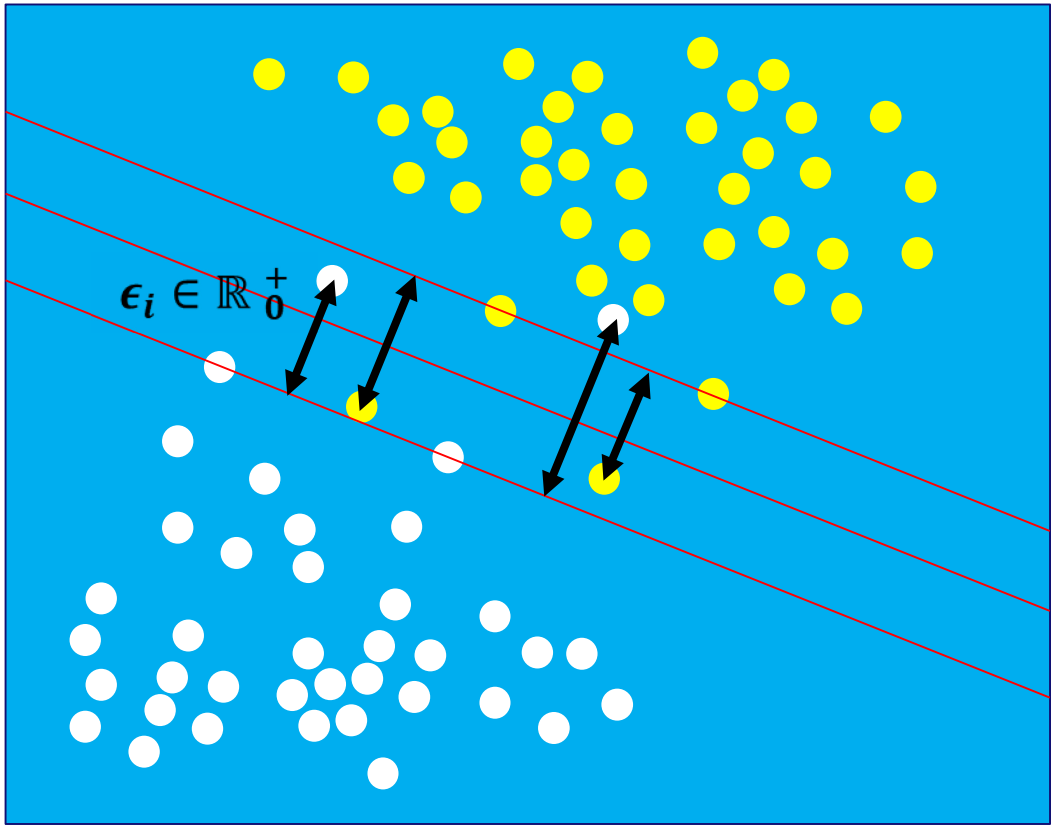
An outlier may also reduce the margin



Hyperplane seems reasonable, but …



Solution: Minimize slack variables



Soft-margin SVM

Given a set of m instances

$$\{(\vec{x}_i, y_i) \in \mathbb{R}^n \times \{-1, 1\} | 1 \leq i \leq m\}$$

$$\min_{\vec{w},b,\epsilon} \frac{1}{2}\|\vec{w}\|^2 + C\sum_{i=1}^{m} \epsilon_i$$

such that

$$y_i(\vec{w} \cdot \vec{x}_i + b) \geq 1 - \epsilon_i \text{ for any } i$$

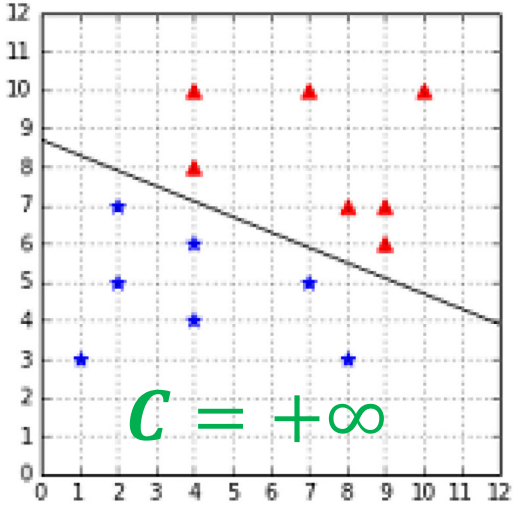
$$\epsilon_i \geq 0 \text{ for any } i$$

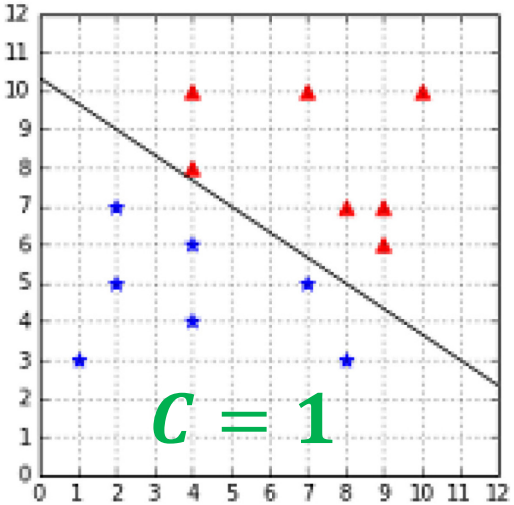
Role of C: high is strict, low is flexible

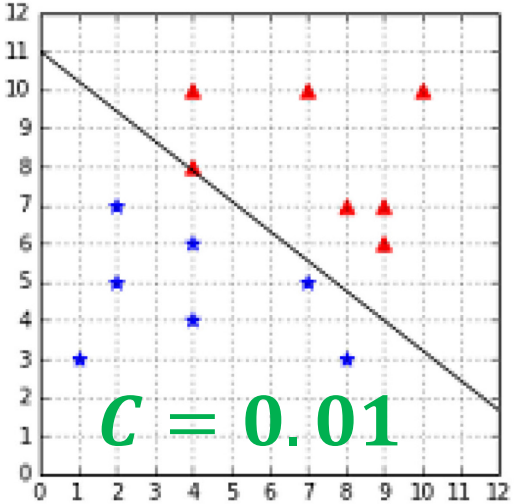
$$\min_{\vec{w},b,\epsilon} \frac{1}{2}\|\vec{w}\|^2 + C\sum_{i=1}^{m}\epsilon_i$$

such that

$y_i(\vec{w}\cdot\vec{x}_i + b) \geq 1 - \epsilon_i$ **for any** i

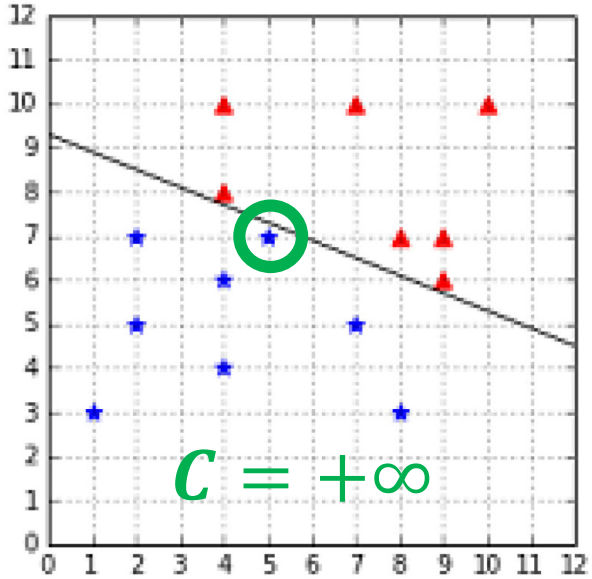


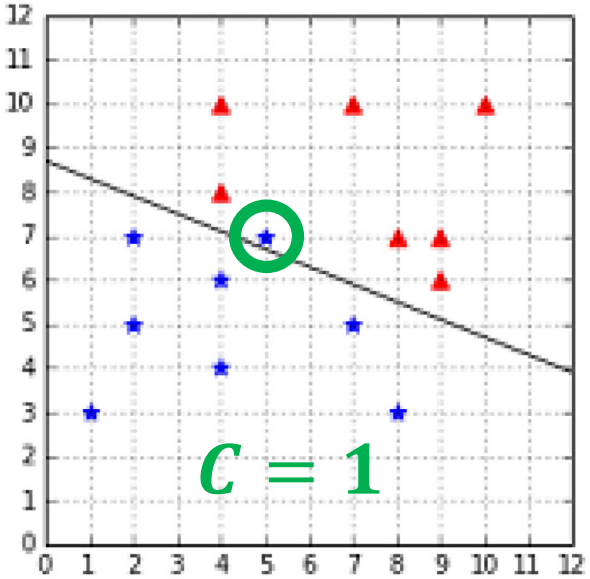


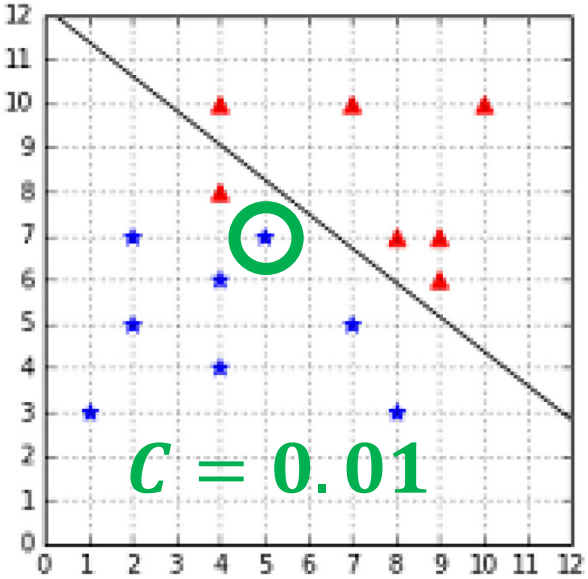


Chair of Process and Data Science

Combined effect of C and an outlier



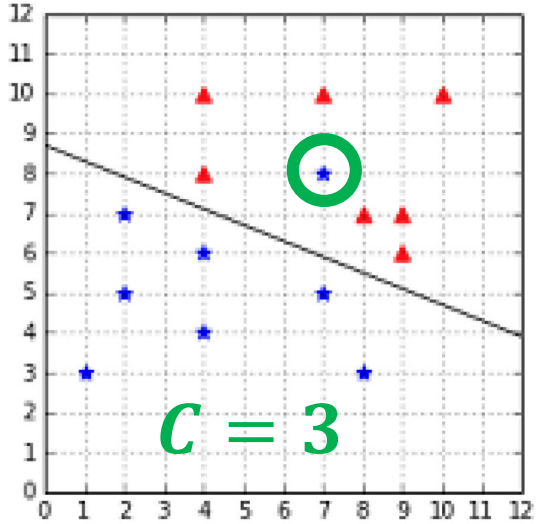


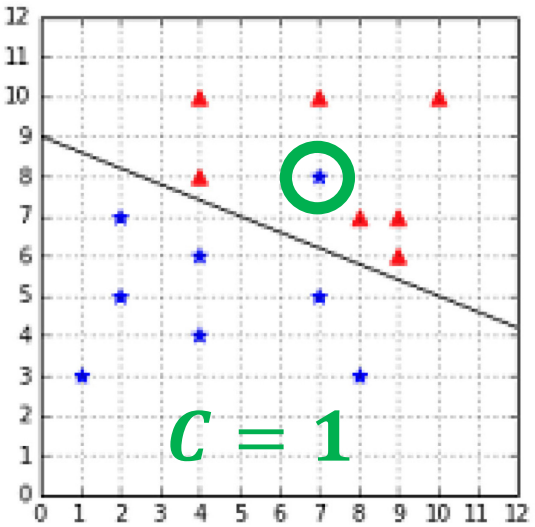


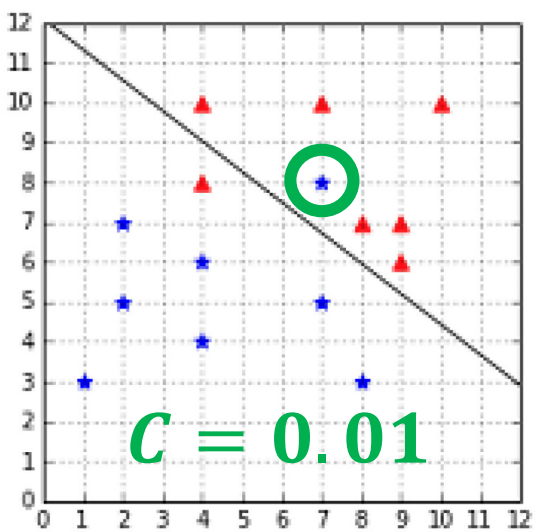
Data is still separable.

Chair of Process and Data Science

Combined effect of C and an outlier







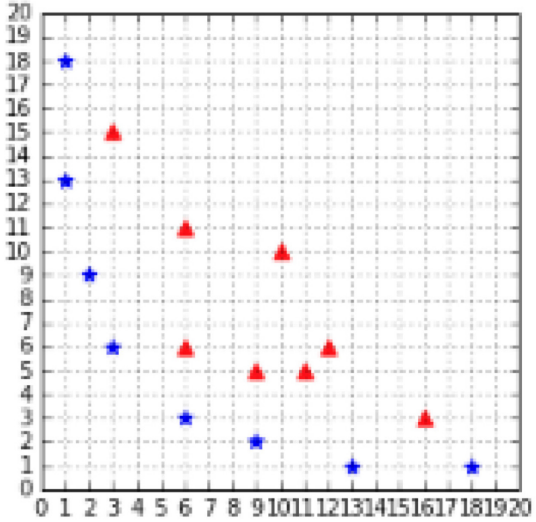
What if $C = +\infty$?

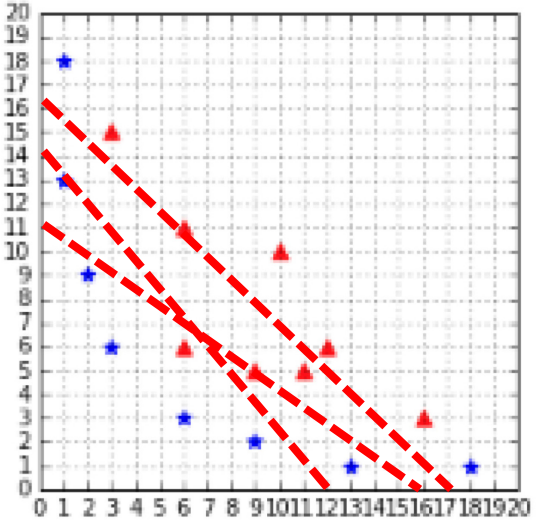
Data is no longer separable!

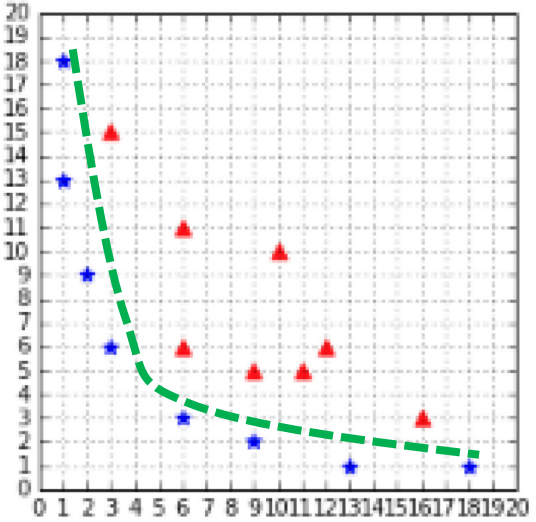
Chair of Process and Data Science

Non-linear decision boundaries

Non-linear separation

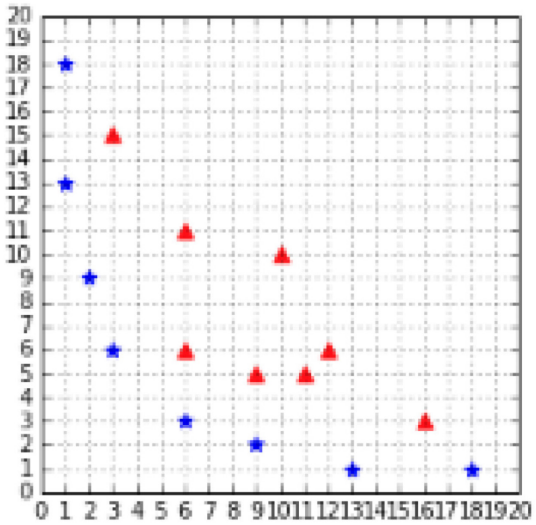


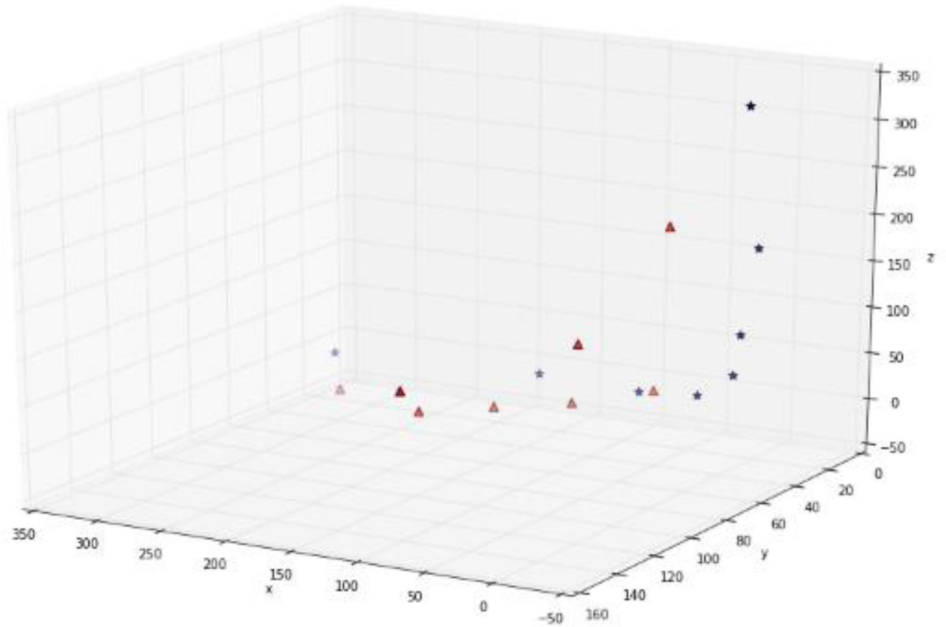




Idea: Lift the number of dimensions

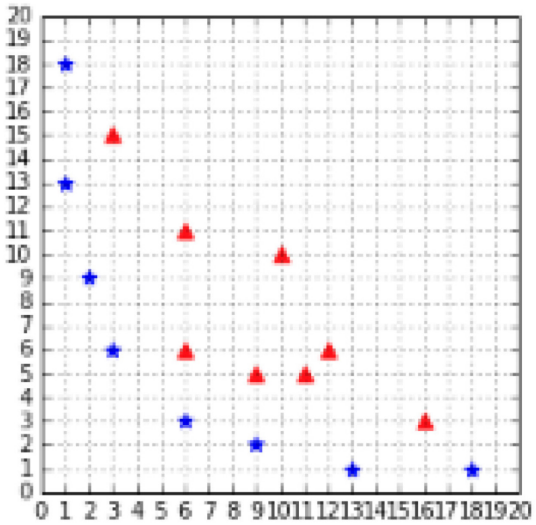
Lift from n=2 to n=3

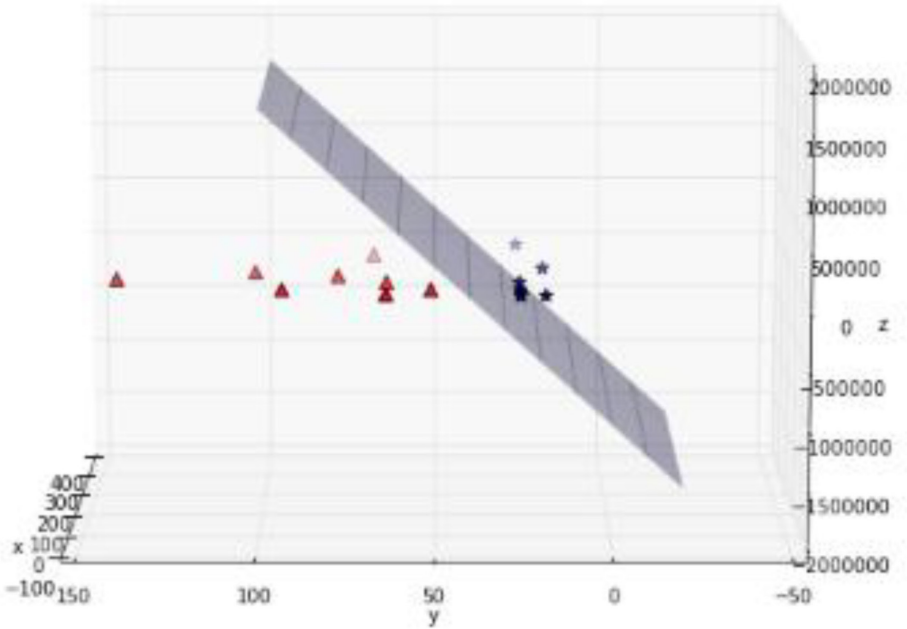




Chair of Process and Data Science

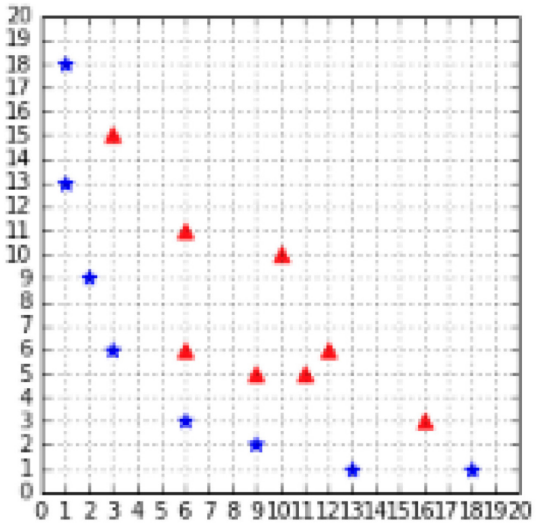
Lift from n=2 to n=3

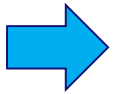


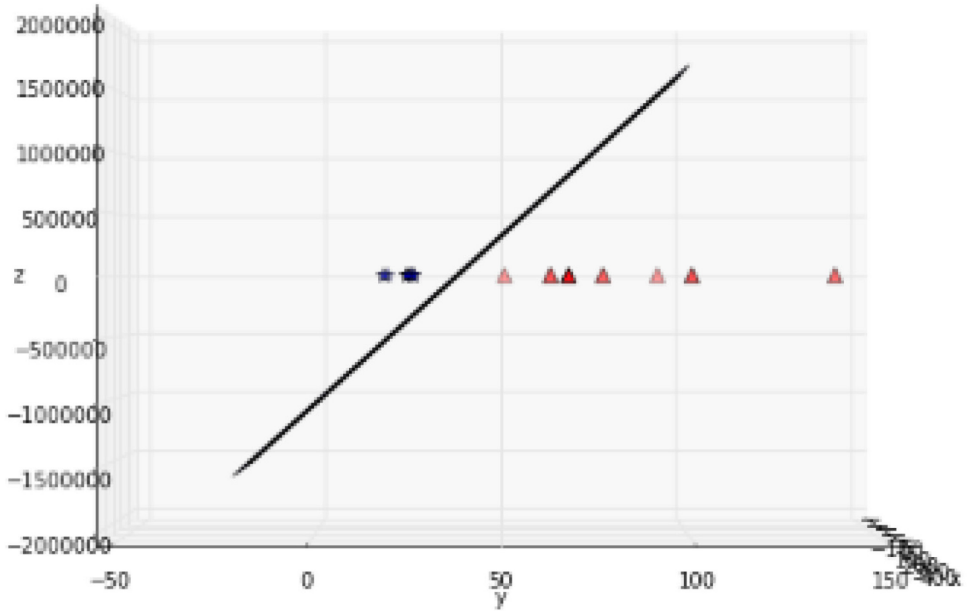


Chair of Process and Data Science

Lift from n=2 to n=3







Chair of Process and Data Science

Mapping

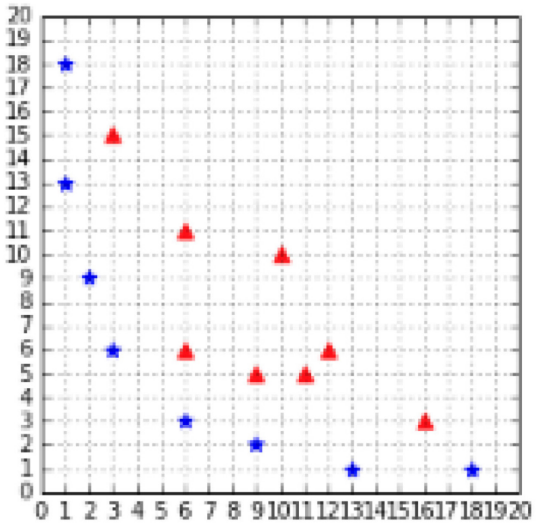
$$\boldsymbol{\phi} \in \mathbb{R}^{n} \rightarrow \mathbb{R}^{q} \textbf{ with } q > n$$

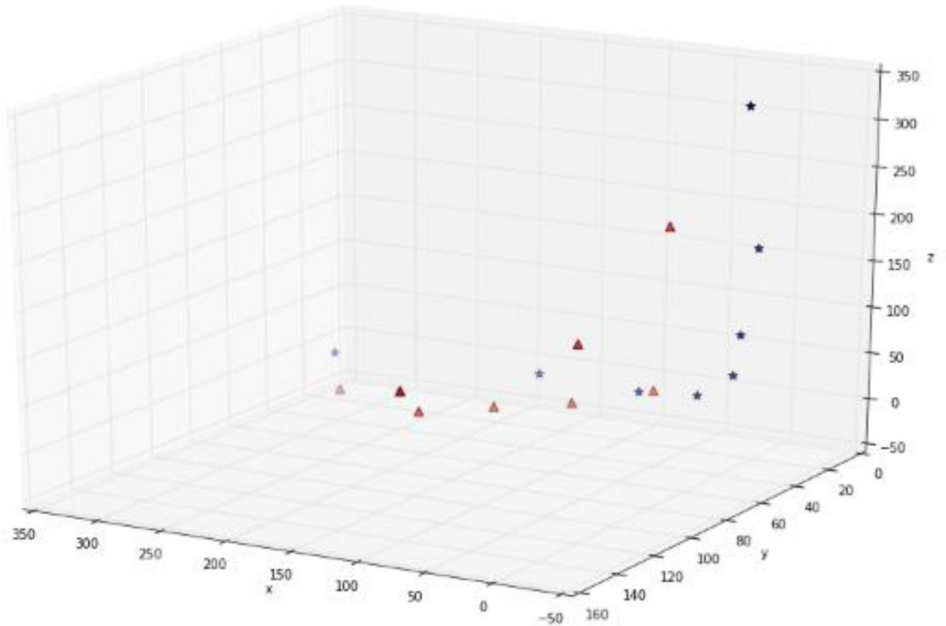
$$\min_{\vec{w},b,\epsilon} \frac{1}{2} \|\vec{w}\|^2 + C \sum_{i=1}^{m} \epsilon_i$$

such that

$$y_i(\vec{w} \cdot \boldsymbol{\phi}(\vec{x}_i) + b) \geq 1 - \epsilon_i \textbf{ for any } i$$

$\phi \in \mathbb{R}^2 \rightarrow \mathbb{R}^3$





$$\phi(x_1, x_2) = (x_1^2, \sqrt{2}x_1x_2, x_2^2)$$

Using kernels

Wolfe dual Lagrangian problem

$$\mathbf{w} = \sum_{i=1}^{m} \alpha_i y_i \mathbf{x}_i$$

$$\alpha_i \geq 0, \quad i = 1, \ldots, m$$

$$\underset{\alpha}{\text{maximize}} \quad \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j$$

$$\text{subject to} \quad \alpha_i \geq 0, \text{ for any } i = 1, \ldots, m$$

$$\sum_{i=1}^{m} \alpha_i y_i = 0$$

Wolfe dual Lagrangian problem

$$\mathbf{w} = \sum_{i=1}^{m} \alpha_i y_i \mathbf{x}_i$$

$$\alpha_i \geq 0, \quad i = 1, \ldots, m$$

$$\underset{\alpha}{\text{maximize}} \quad \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j$$

$$\text{subject to} \quad \alpha_i \geq 0, \text{ for any } i = 1, \ldots$$

$$\sum_{i=1}^{m} \alpha_i y_i = 0$$

$\vec{x}_i \cdot \vec{x}_j$

$\phi(\vec{x}_i) \cdot \phi(\vec{x}_j)$

$\phi \in \mathbb{R}^n \rightarrow \mathbb{R}^q$ **with** $q > n$

How to compute $\phi(\vec{x}_i) \cdot \phi(\vec{x}_j)$ efficiently?

Find a kernel function

$$K(\vec{x}_i, \vec{x}_j) = \phi(\vec{x}_i) \cdot \phi(\vec{x}_j)$$

without doing the actual mappings $\phi(\vec{x}_i)$ and $\phi(\vec{x}_j)$ and computing $\phi(\vec{x}_i) \cdot \phi(\vec{x}_j)$.

Example

Find a kernel function

$K(\vec{x}_i, \vec{x}_j) = \phi(\vec{x}_i) \cdot \phi(\vec{x}_j)$

without doing the actual mappings $\phi(\vec{x}_i)$ and $\phi(\vec{x}_j)$ and computing $\phi(\vec{x}_i) \cdot \phi(\vec{x}_j)$.

$$\phi(x_1, x_2) = (x_1^2, \sqrt{2}x_1x_2, x_2^2)$$

$$\phi(x_{i,1}, x_{i,2}) = (x_{i,1}^2, \sqrt{2}x_{i,1}x_{i,2}, x_{i,2}^2)$$

$$\phi(x_{j,1}, x_{j,2}) = (x_{j,1}^2, \sqrt{2}x_{j,1}x_{j,2}, x_{j.2}^2)$$

Example

Find a kernel function

$K(\vec{x}_i, \vec{x}_j) = \phi(\vec{x}_i) \cdot \phi(\vec{x}_j)$

without doing the actual mappings $\phi(\vec{x}_i)$ and $\phi(\vec{x}_j)$ and computing $\phi(\vec{x}_i) \cdot \phi(\vec{x}_j)$.

$$\phi(\vec{x}_i) \cdot \phi(\vec{x}_j)$$

$$= x_{i,1}^2 x_{j,1}^2 + 2x_{i,1}x_{i,2}x_{j,1}x_{j,2} + x_{i,2}^2 x_{j,2}^2$$

$$= (x_{i,1}x_{j,1})^2 + 2(x_{i,1}x_{j,1})(x_{i,2}x_{j,2}) + (x_{i,2}x_{j,2})^2$$

$$= (x_{i,1}x_{j,1} + x_{i,2}x_{j,2})^2$$

Different kernel functions

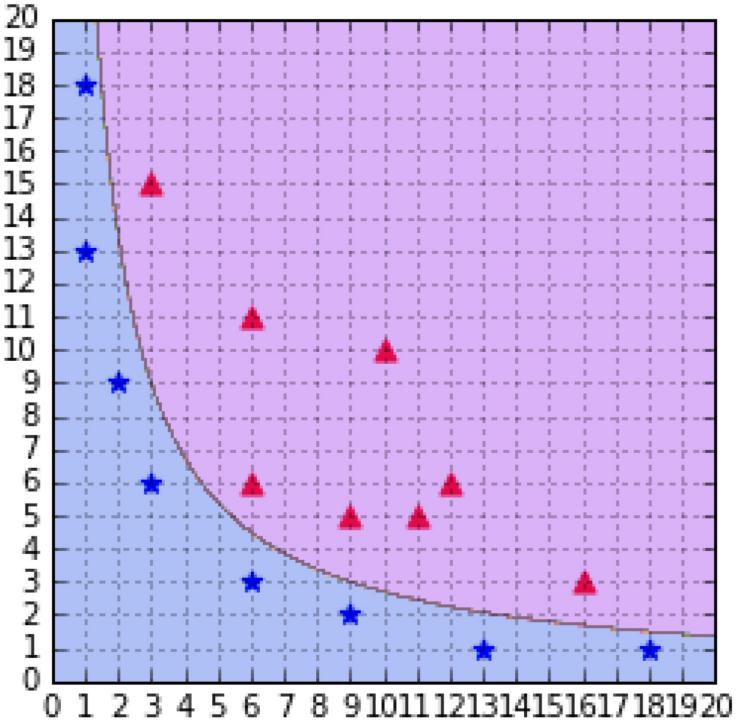
Polynomial kernel with parameters c and d:

$$K(\vec{x}_i, \vec{x}_j) = \phi(\vec{x}_i) \cdot \phi(\vec{x}_j) = (\vec{x}_i \cdot \vec{x}_j + c)^d$$

d = degree

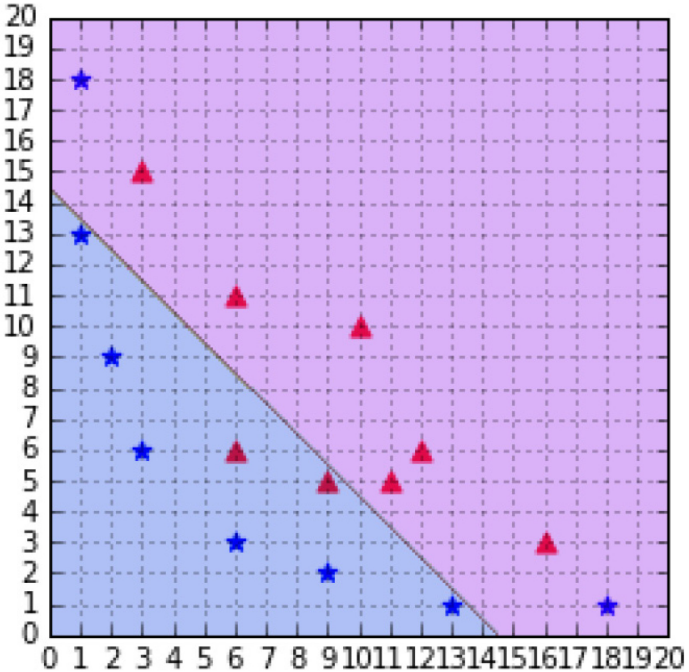
c = constant

Polynomial kernel ($d = 2$)



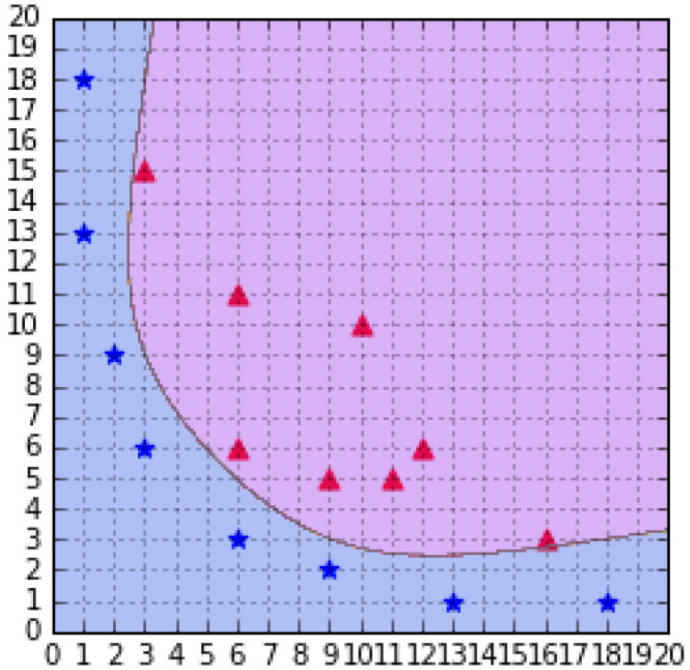
$$K(\vec{x}_i, \vec{x}_j) = \phi(\vec{x}_i) \cdot \phi(\vec{x}_j)$$
$$= (\vec{x}_i \cdot \vec{x}_j + 0)^2$$

Polynomial kernel ($d = 1$)



$$K(\vec{x}_i, \vec{x}_j) = \phi(\vec{x}_i) \cdot \phi(\vec{x}_j)$$
$$= (\vec{x}_i \cdot \vec{x}_j + 0)^1$$

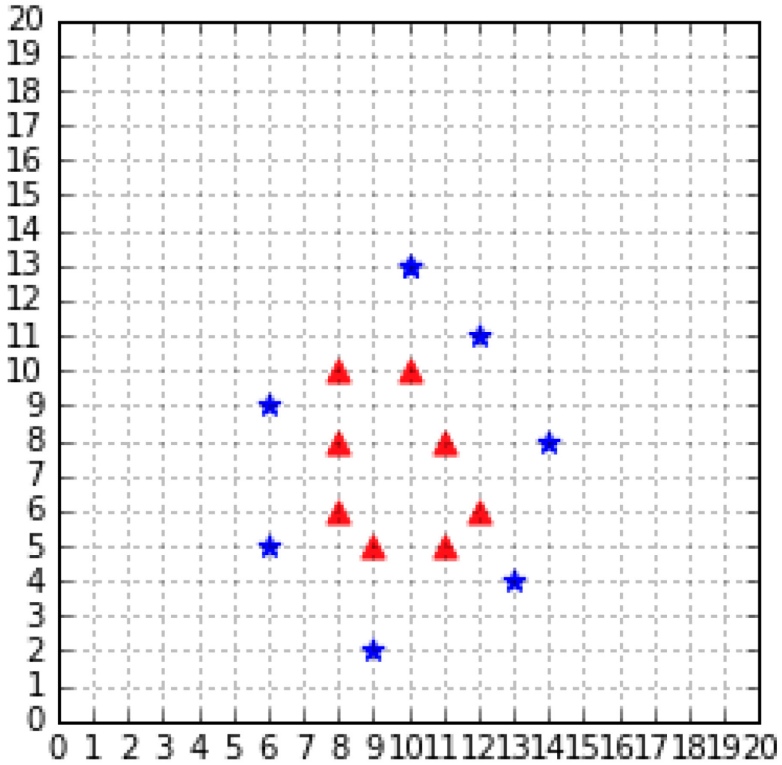
Polynomial kernel ($d = 6$)

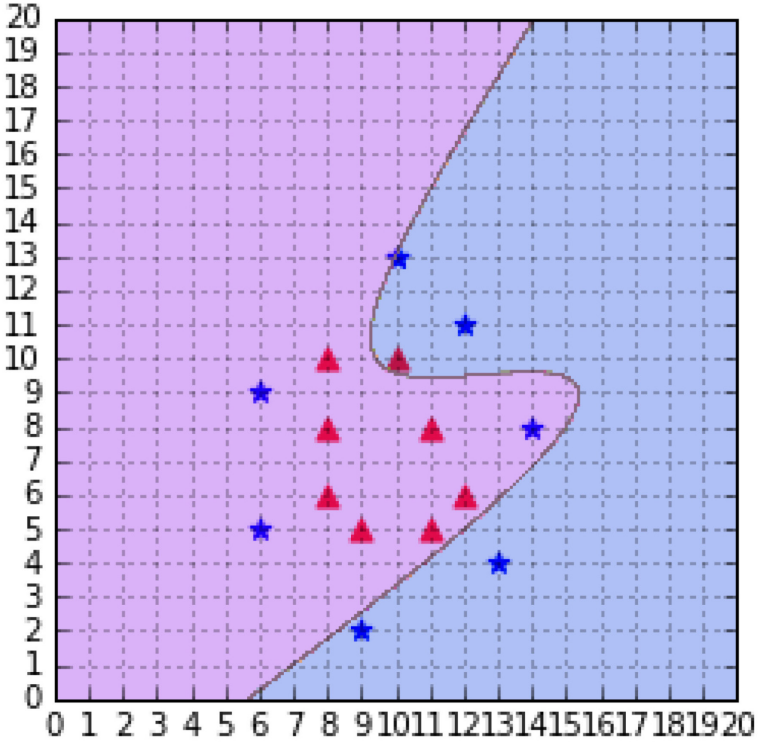


$$K(\vec{x}_i, \vec{x}_j) = \phi(\vec{x}_i) \cdot \phi(\vec{x}_j)$$
$$= (\vec{x}_i \cdot \vec{x}_j + 0)^6$$

Note the danger of overfitting the data when the degree increases!

Cannot be handled by any polynomial kernel

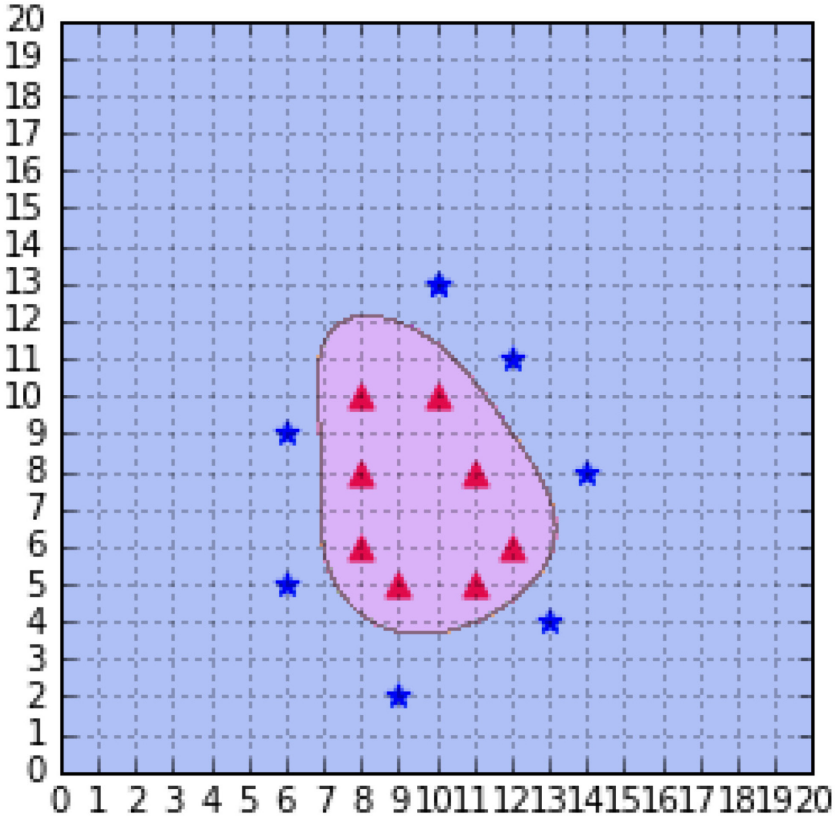




d=3, C = 100 to allow for error

Chair of Process and Data Science

Other kernels



Chair of Process and Data Science

Conclusion

Short summary of lecture

- SVMs provided a supervised learning approach able to classify unseen data.
- Able to deal with higher dimensional problems (many features and fewer instances).
- Focus on border cases: support vectors.

Short summary of lecture

- **Extensions to handle outliers and non-linear decision surfaces.**
- **Many “tricks” needed to get a good performance and to fit the data.**

Relevant Literature

- Support Vector Machines Succinctly by Alexandre Kowalczyk, Syncfusion, 2017.
- Chapter 7, Section 7.4.7, Fundamentals of Machine Learning for Predictive Data Analytics: Algorithms, Worked Examples, and Case Studies by John D. Kelleher, Brian Mac Namee and Aoife D'Arcy. MIT Press, 2015
- Chapter 9, Section 9.3, Data Mining: Concepts and Techniques (3rd edition) by Jiawei Han , Micheline Kamber , Jian Pei. The Morgan Kaufmann Series in Data Management Systems, Elsevier, 2011.