Machine Learning Methods
P160B124

Support Vector methods

assoc. prof. dr. Tomas Iešmantas
tomas.iesmantas@ktu.lt
Room 319

Hyperplane as a decision boundary

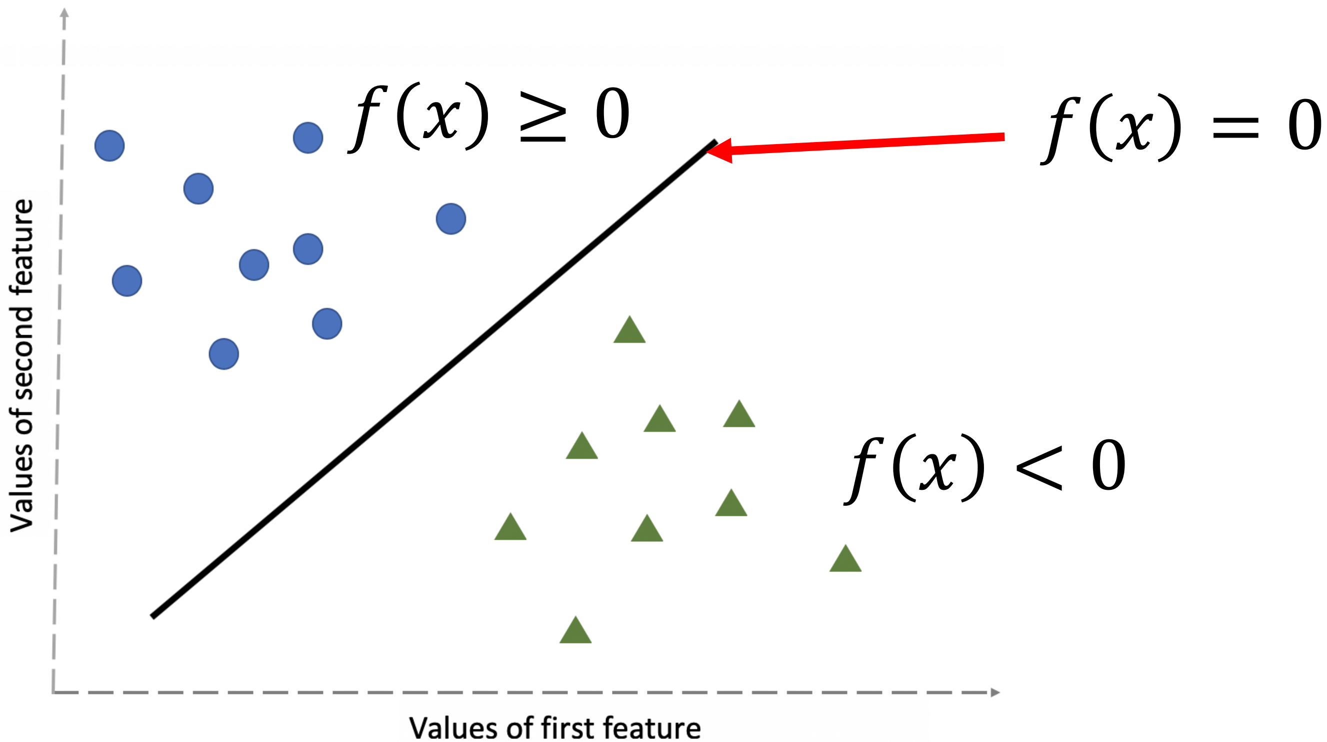
- A decision boundaries for logistic and LDA classifiers are linear, i.e. hyperplanes.
- ***Support vector classifier*** seeks such a decision boundary as well, but by a different route.
- ***Support vector machine*** also constructs a linear decision boundary but in some other high dimensional space (possibly even in infinite dimensional space).
- When talking about SVC or SVM, classes are assumed to be positive and negative, i.e. $y_i \in \{-1, +1\}$ (this is only a notational convenience).

Hyperplane as a decision boundary

- In general, a *linear classifier* f can be summarized by the following rule:

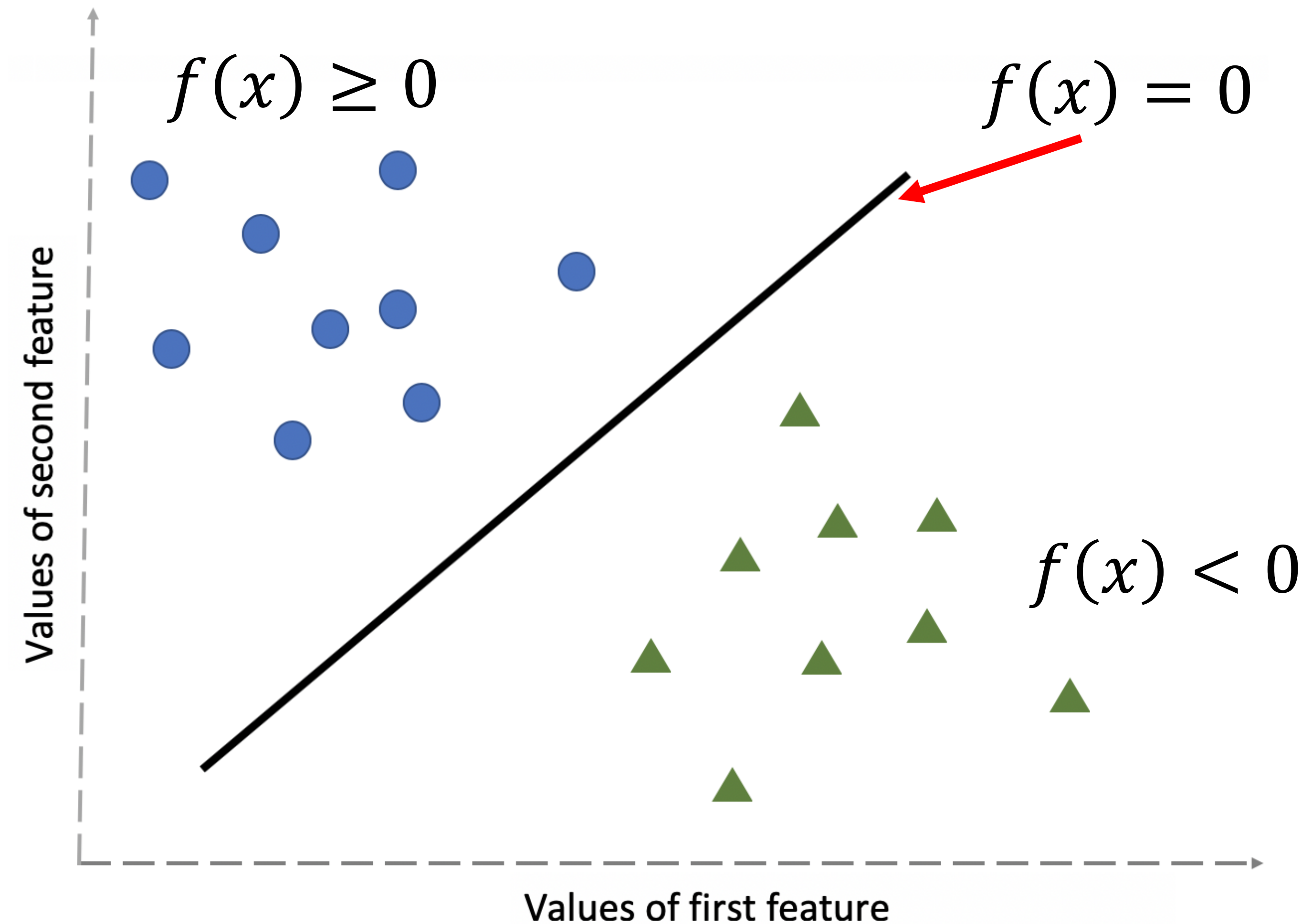
$$If\ f(x) \geq 0 \ \Rightarrow y = +1, \text{otherwise}\ y = -1$$

If $f(x)$ is very large, then $\boldsymbol{x}$ **is very far away from the boundary** – and **we are confident in the prediction**.

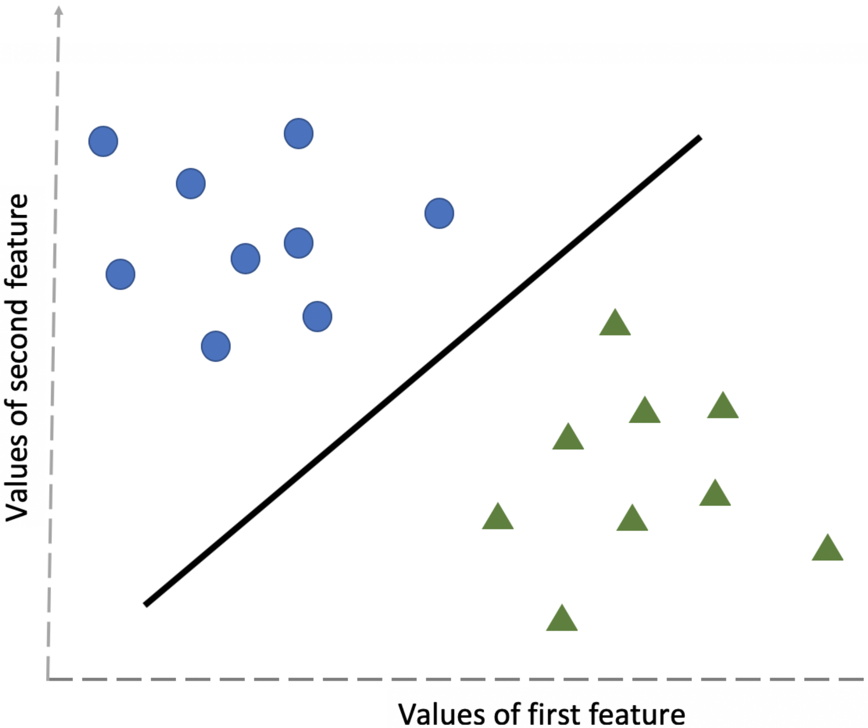


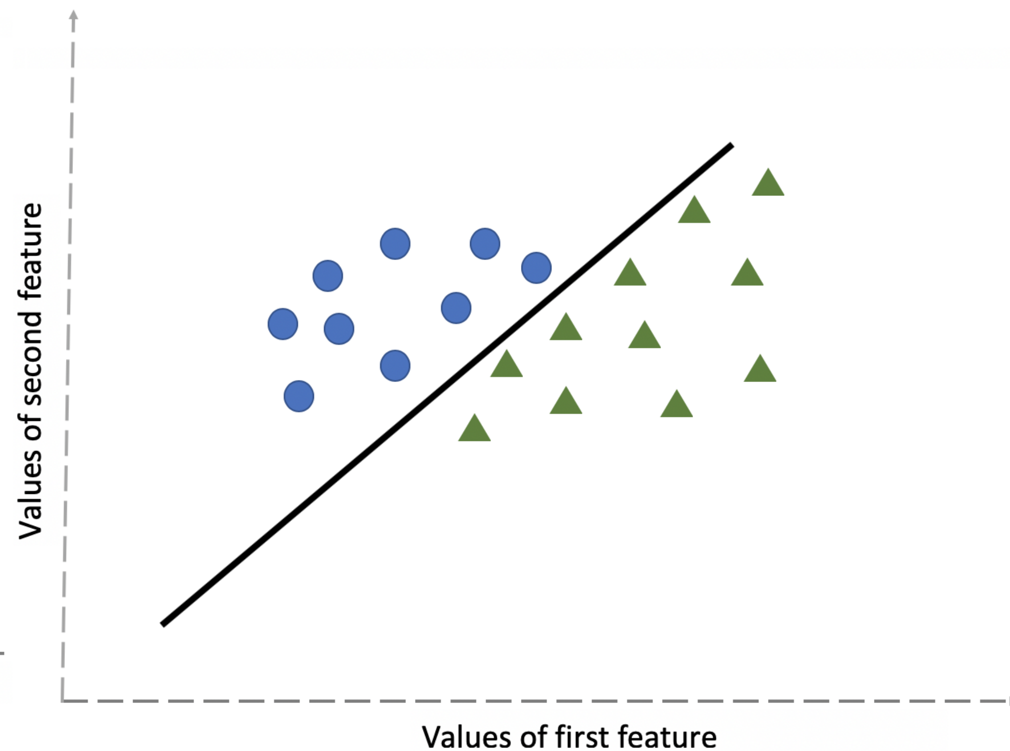
SVM and decision boundary

- SV method seeks the "best" linear decision boundary out of all. What is "best"?

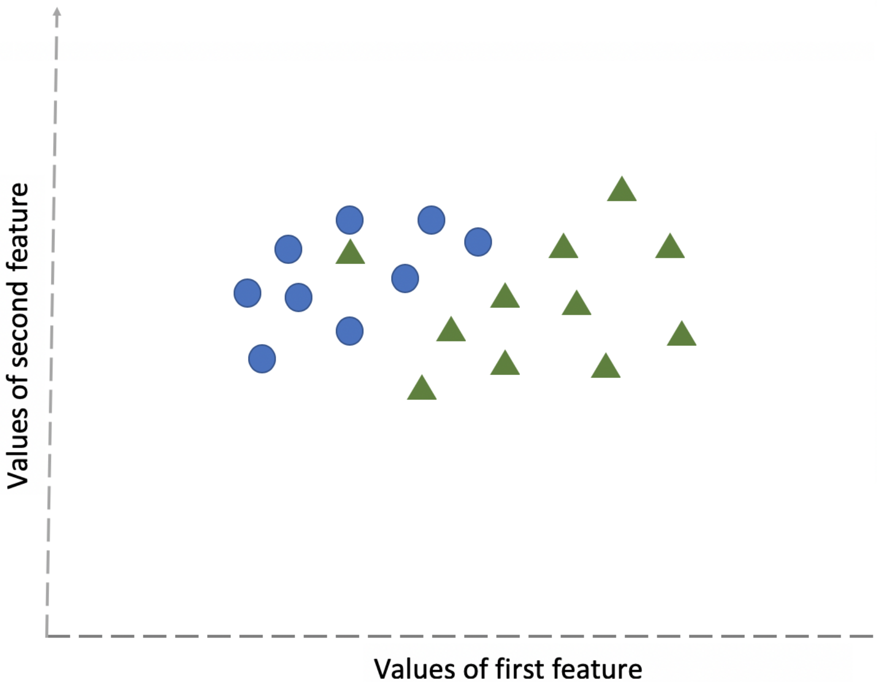


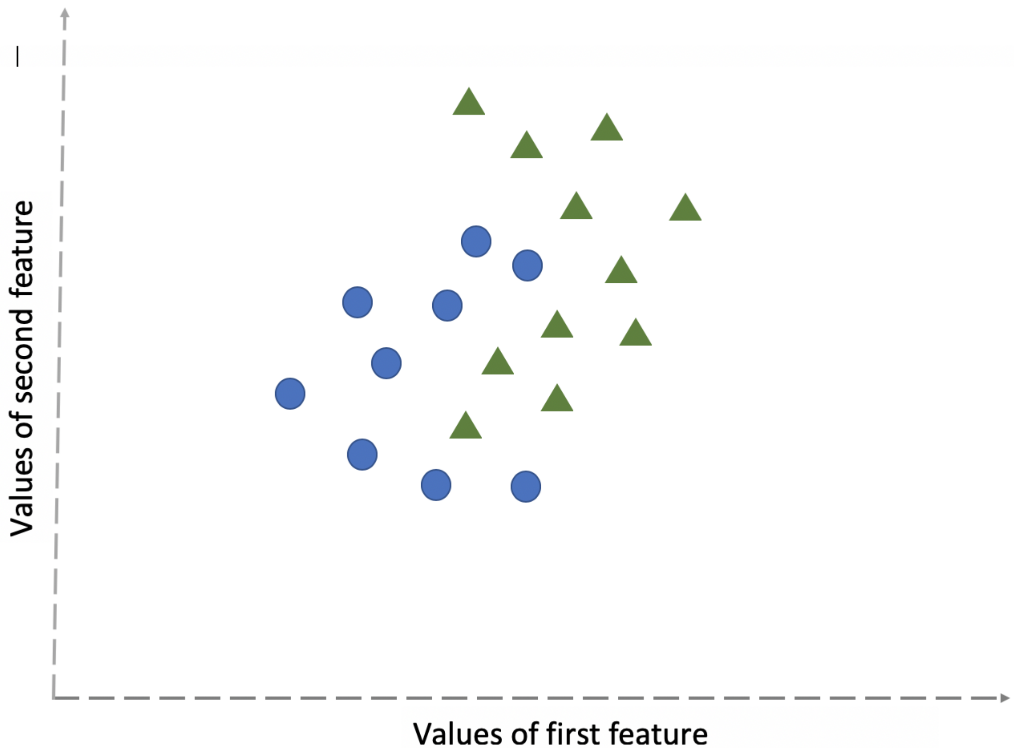
Linear separability





Linear non-separability





Linear classifier

- It's general form is this:

$$f(x) = w_0 + x^T w = 0$$

- For example: $1 - 0.4x_1 + 3x_2 = 0, w_0 = 1, w = \begin{pmatrix} -0.4 \\ 3 \end{pmatrix}$

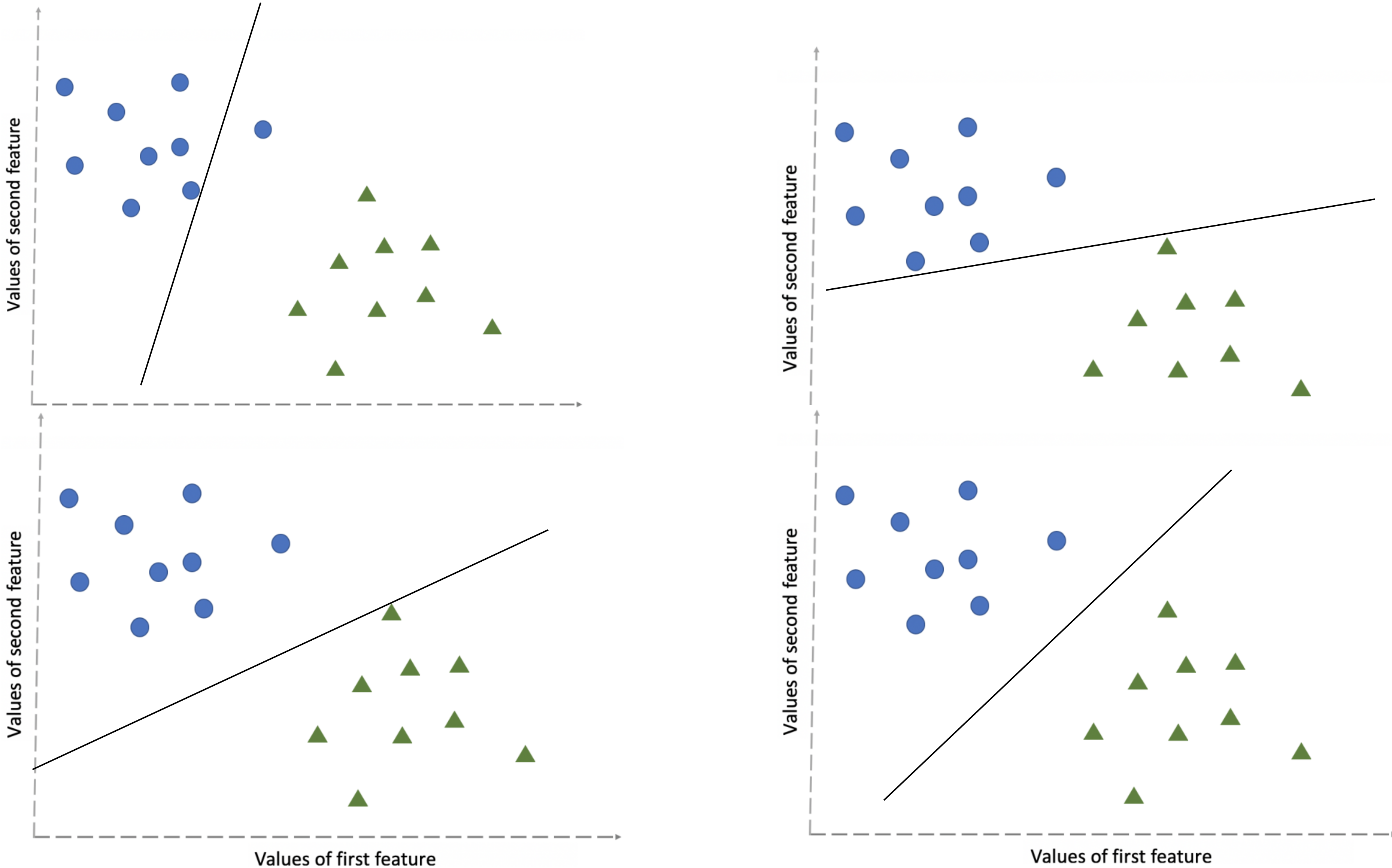
$y = +1$ if

$$1 - 0.4x_1 + 3x_2 \geq 0$$

- Training data is used to find unknown parameters w_0 and w.

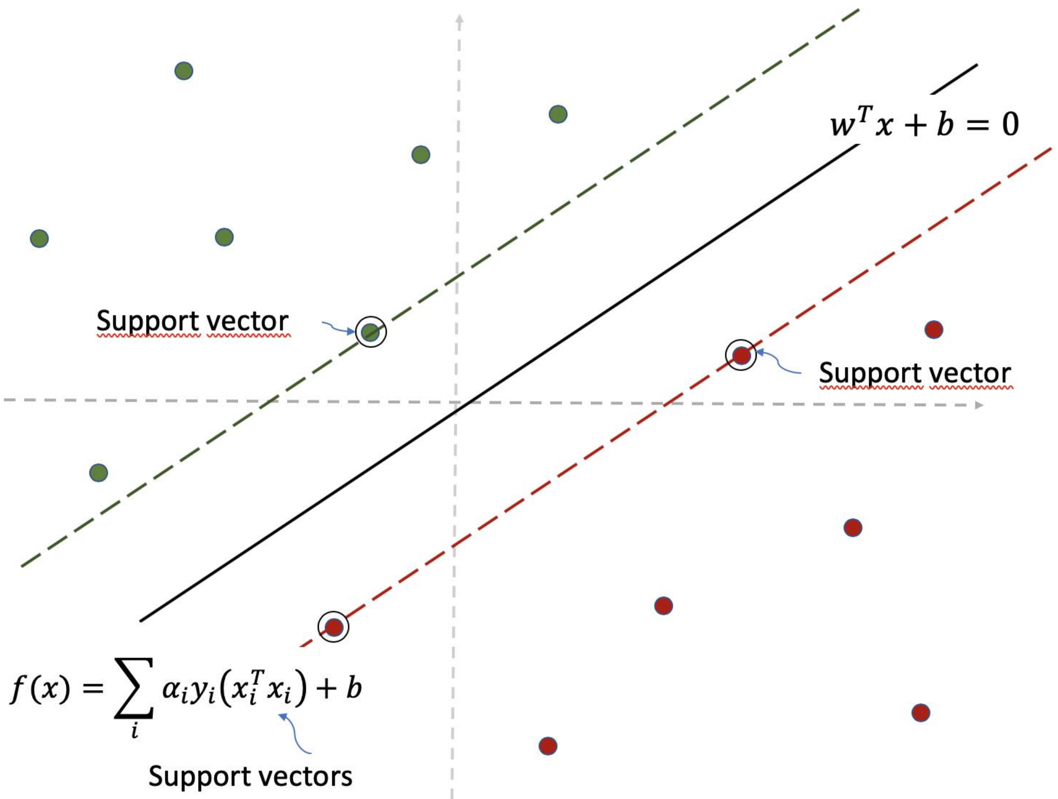
Linear classifier

- There is an infinite number of possibilities.



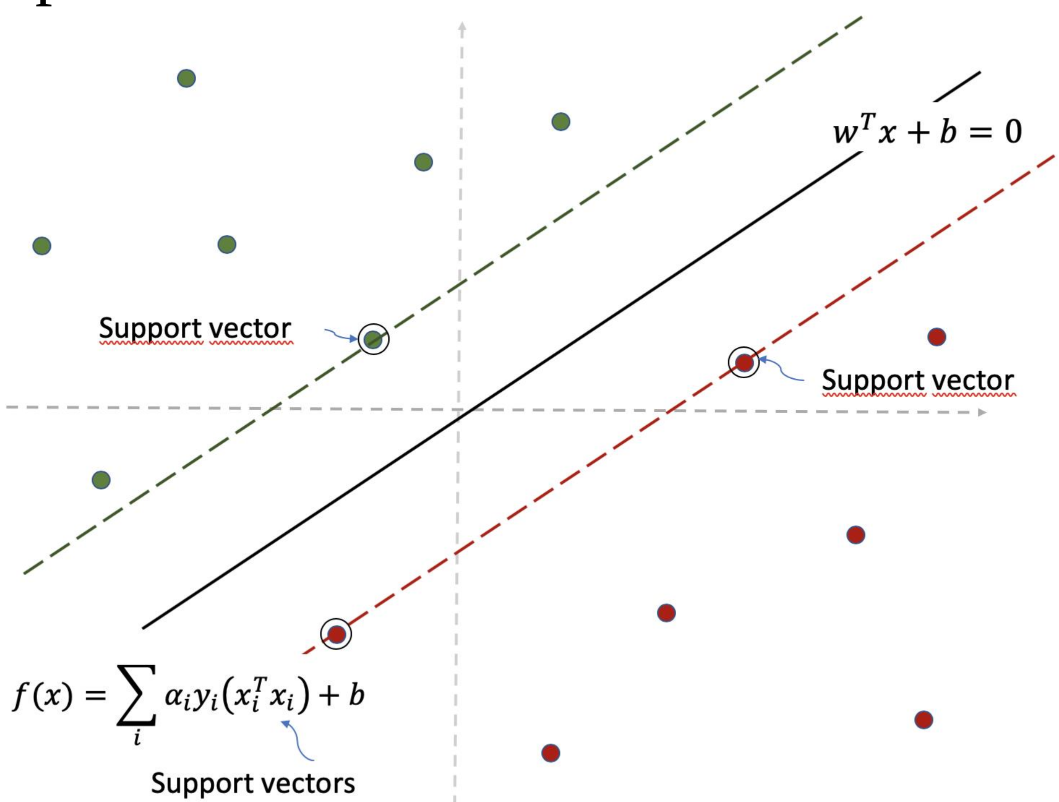
Margin of a hyperplane

- **Margin** is a distance from the hyperplane to the closest data point.
- The **maximal margin hyperplane** is the separating hyperplane for which the margin is largest. Distance to the closest training points of both classes is maximal.



Support Vectors

- **Support vectors** are training data points which are on boundary of the margins.
- They "support" the maximal margin hyperplane in the sense that if these points were moved slightly then the maximal margin hyperplane would move as well.

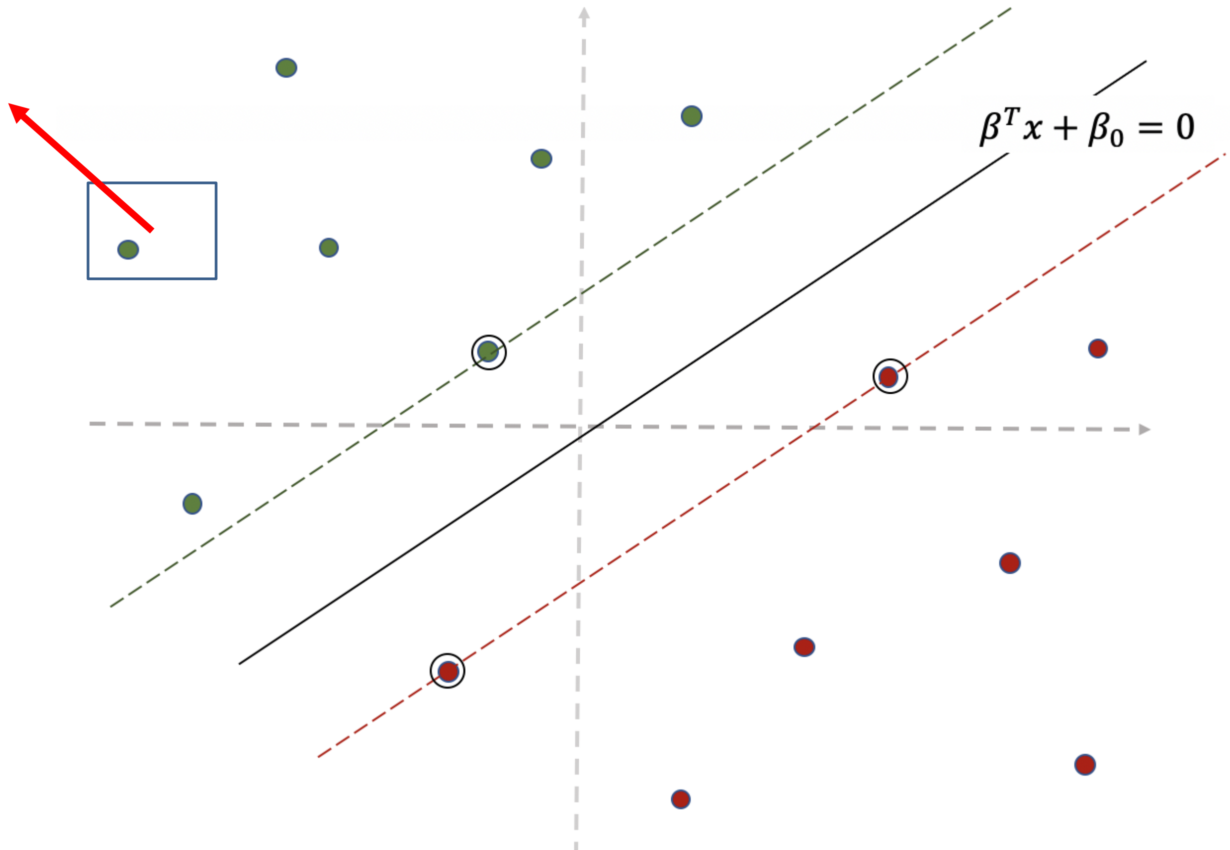


Question 1

- If we move the denoted vector to some direction, will the maximum margin hyperplane, once it is found, move as well?

  a) Yes;

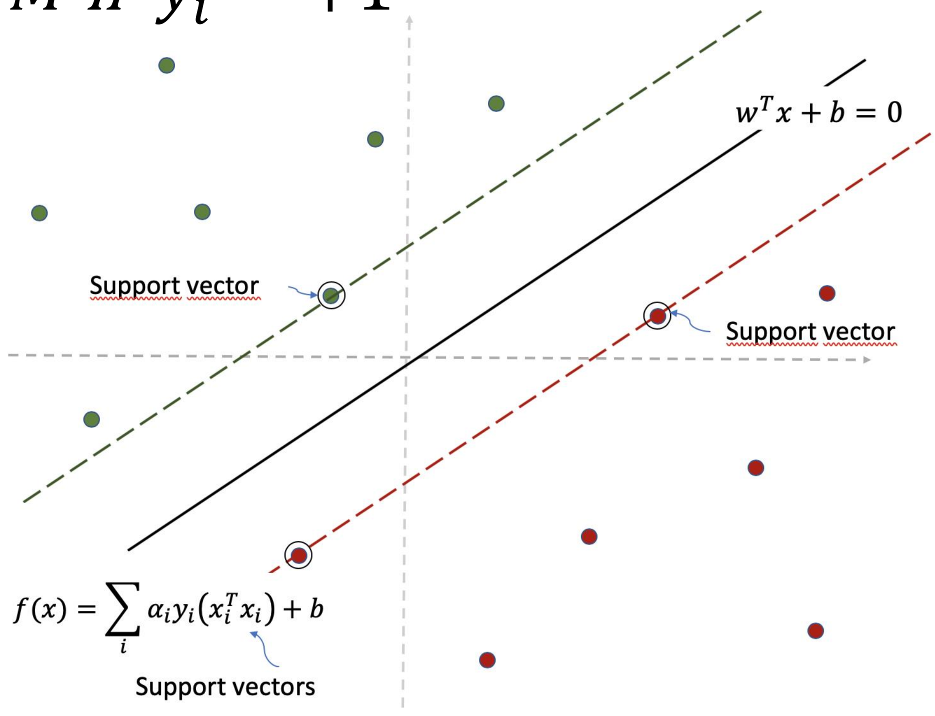
  b) No



Maximum margin classifier

- A maximum margin classifier is the classifier for which maximum margin hyperplane is the decision boundary. **The task is simple: find maximum margin!**
- If the classes are separable, then

$$w_0 + w^T x_i \geq M \ if \ y_i = +1$$
$$w_0 + w^T x_i \leq -M \ if \ y_i = +1$$



Maximum margin classifier

- If the classes are separable, then

$$w_0 + w^T x_i \geq M \;\; if \; y_i = +1$$
$$w_0 + w^T x_i \leq -M \;\; if \; y_i = -1$$

Or equivalently (with a slight abuse of notation)

$$w_0 + w^T x_i \geq +1 \; if \; y_i = +1$$
$$w_0 + w^T x_i \leq -1 \; if \; y_i = -1$$

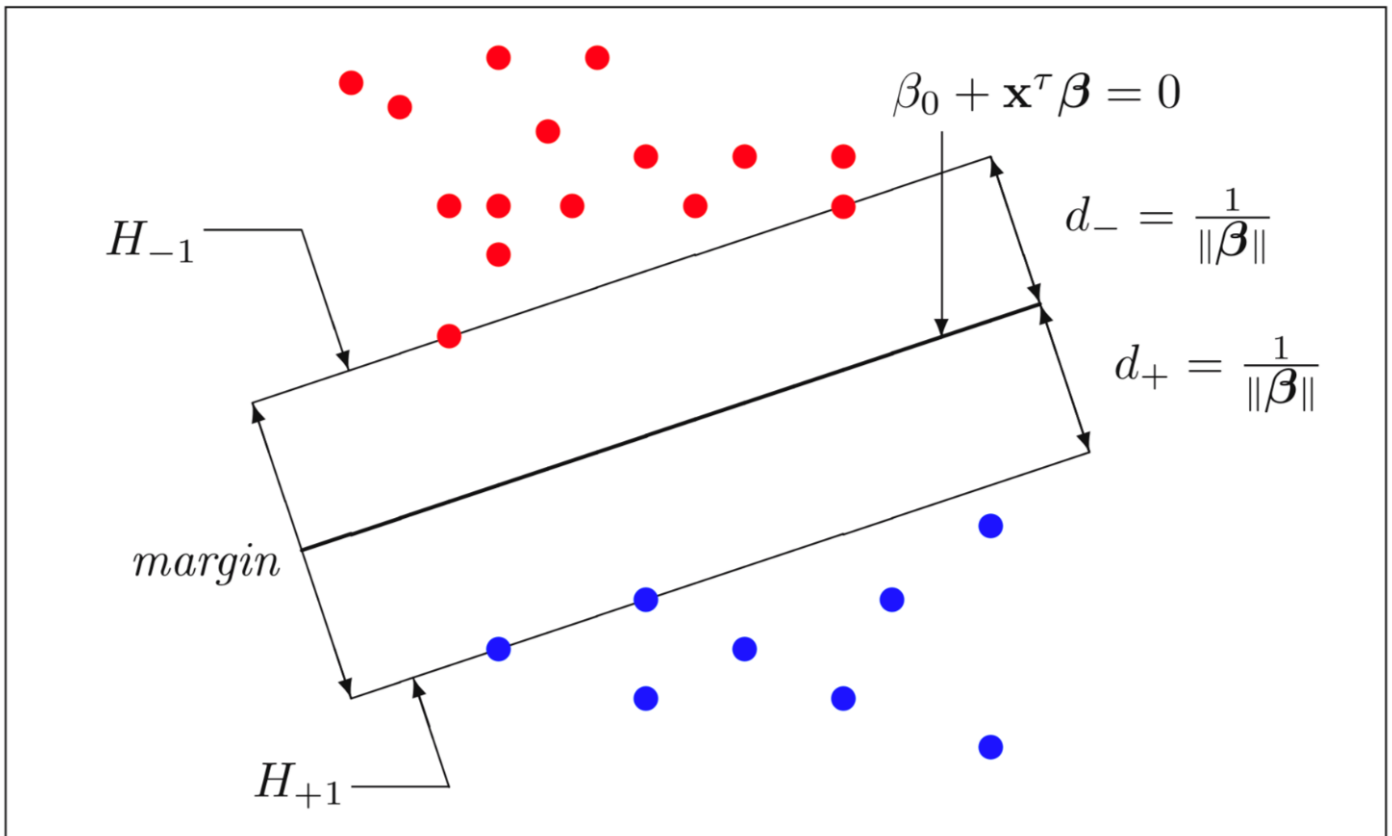
Data points x_{+1} and x_{-1} satisfying equality conditions lie on the hyperplanes

$$H_{+1}: w_0 + w^T x_{+1} + 1 = 0$$
$$H_{-1}: w_0 + w^T x_{-1} - 1 = 0$$

These points are support vectors.

Maximum margin classifier

- In general, distance between two hyperplanes $w^T x + b_1 = 0$ and $w^T x + b_2 = 0$ is $\frac{|b_1 - b_2|}{||w||}$
- Thus, in our case margin size is $d = \frac{2}{||w||}$.



Question 2

- What is the margin for a maximum margin hyperplane

$$1 - 0.4x_1 + 3x_2 = 0$$

a) 0.66

b) 0.62

c) 0.60

$$d = \frac{2}{||w||}$$

Maximum margin classifier

- Conditions

$$w_0 + w^T x_i \geq +1 \; if \;\; y_i = +1$$
$$w_0 + w^T x_i \leq -1 \; if \;\; y_i = -1$$

can be written as

$$y_i(w_0 + w^T x_i) \geq +1.$$

Thus, the task is

$$\text{to minimize } \frac{2}{||w||}$$
$$\text{subject } y_i(w_0 + w^T x_i) \geq +1.$$

Maximum margin classifier

- Or, equivalently

$$minimize\ \frac{1}{2}||w||^2,$$
$$subject\ to\ y_i(w_0 + w^T x_i) \geq 1, \forall i.$$

- This is a convex optimization problem: minimize a quadratic function subject to linear inequality constraints.
- Convexity ensures that we have a global minimum without local minima.
- The resulting optimal separating hyperplane is called the **maximal (or hard) margin classifier**.

Nonlinear optimization with constraints

- Suppose a problem

$$optimize\ f(w),$$
$$subject\ to\ g_i(w) \geq 0, \forall i.$$

- Form a Lagrangian:

$$L(w,\lambda) = f(w) - \sum_{i=1}^{N} \lambda_i \cdot g_i(w).$$

Here $\lambda_i \geq 0$ – Lagrange multipliers.

- (KKT – Karush-Kuhn-Tucker theorem) If (w^*, λ^*) is a saddle point of $L(w,\lambda)$, then it is a solution the above optimization problem.
- About saddle points: https://en.wikipedia.org/wiki/Saddle_point

Maximum margin classifier

- The solution of this problem involves only a handful of support vectors:

$$w = \sum_{i \in sv} \alpha_i y_i x_i$$

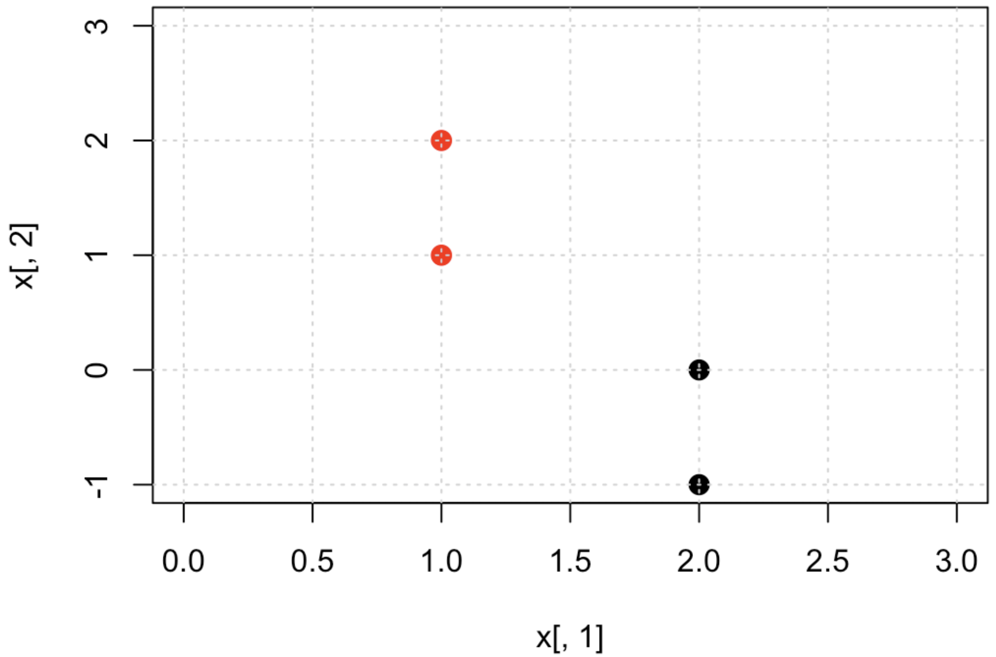
and

$$w_0 = \frac{1}{|sv|} \sum_{k \in sv} \left(\frac{1 - w^T x_k y_k}{y_k} \right)$$

Question 4

- Consider the dataset:

X1	X2	α
1	1	1
1	2	0
2	0	1
2	-1	0



- Find value of

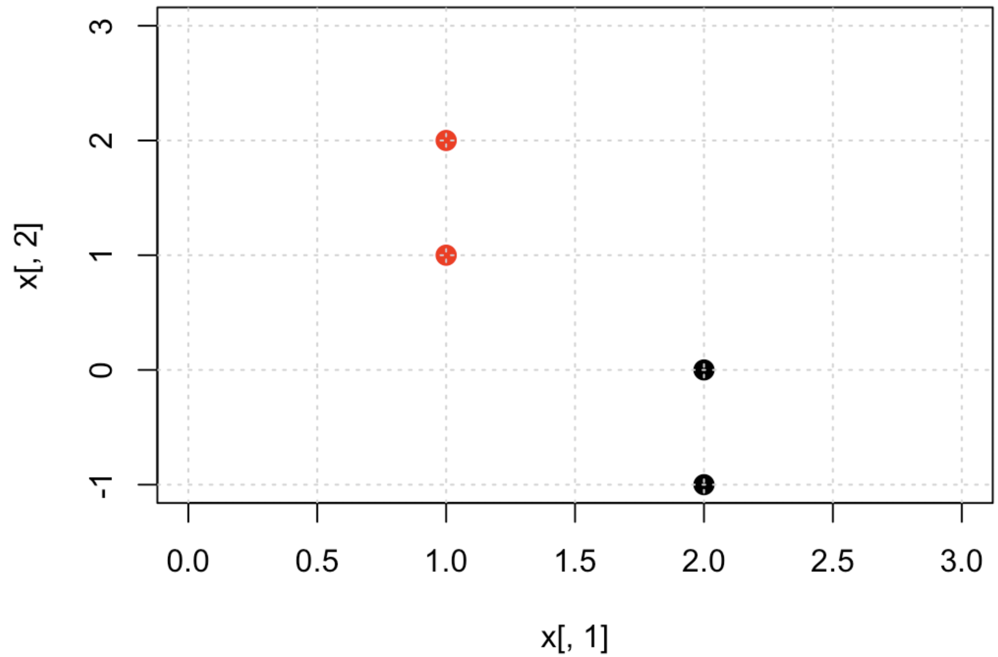
$$w = \sum_{i \in sv} \alpha_i y_i x_i$$

a) (-1,1);
b) (1,1);
c) (1,-1);
d) (0,1)

Question 5

- Consider the dataset:

X1	X2	α
1	1	1
1	2	0
2	0	1
2	-1	0



- Find value of

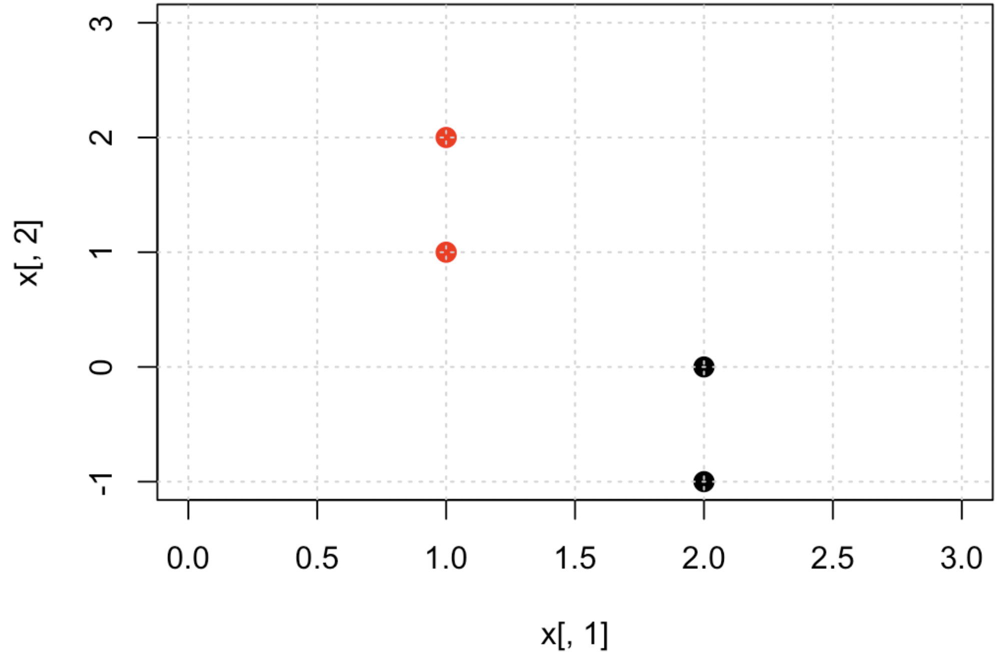
$$w_0 = \frac{1}{|sv|} \sum_{k \in sv} \left(\frac{1 - w^T x_k y_k}{y_k} \right)$$

a) -1;
b) 0;
c) 1;
d) 2

Question 6

- Consider the dataset:

X1	X2	α
1	1	1
1	2	0
2	0	1
2	-1	0



- Find equation for a separating hyperplane

$$f(x) = w_0 + w^T x = 0$$

a) x2=x1+1;

b) x2=x1-1;

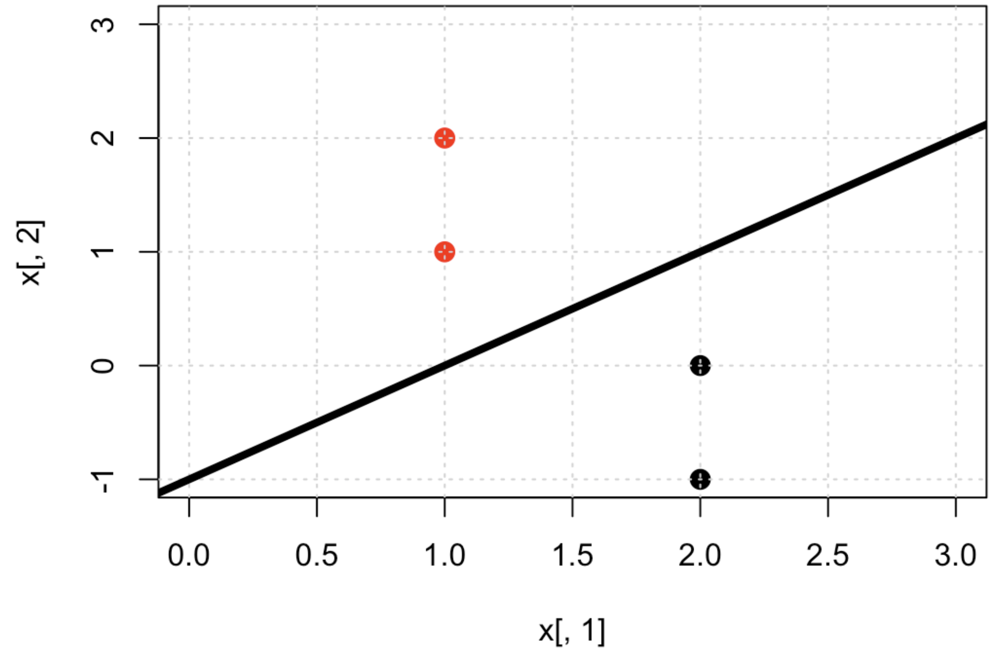
c) x2=x1;

d) x2=x1+2

Question 6

- Consider the dataset:

X1	X2	α
1	1	1
1	2	0
2	0	1
2	-1	0



- Find equation for a separating hyperplane

$$f(x) = w_0 + w^T x = 0$$

a) x2=x1+1;

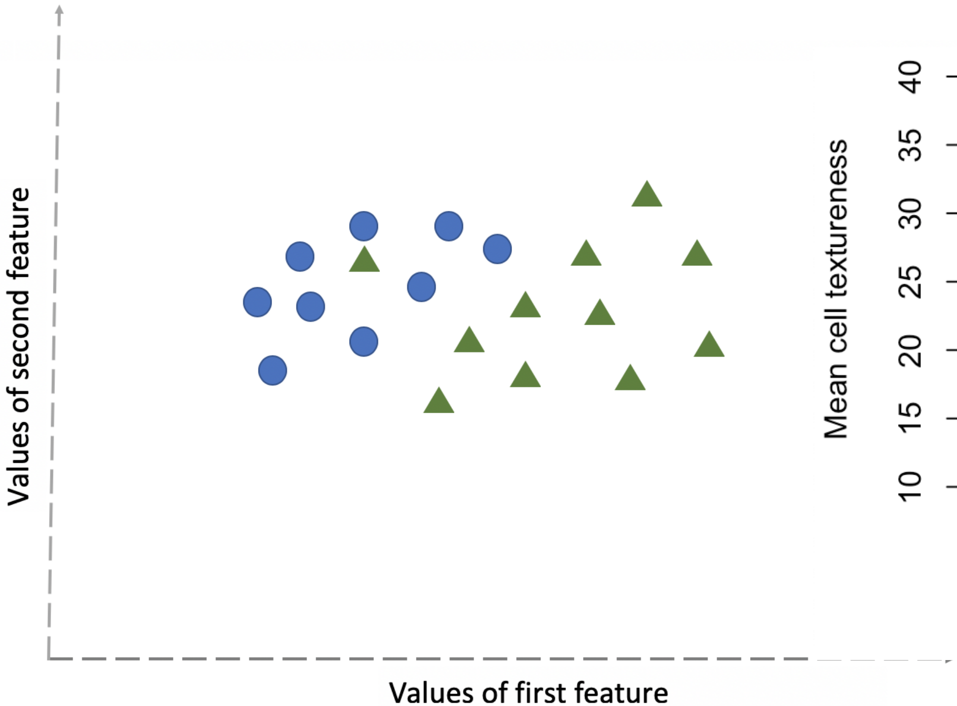
b) x2=x1-1;

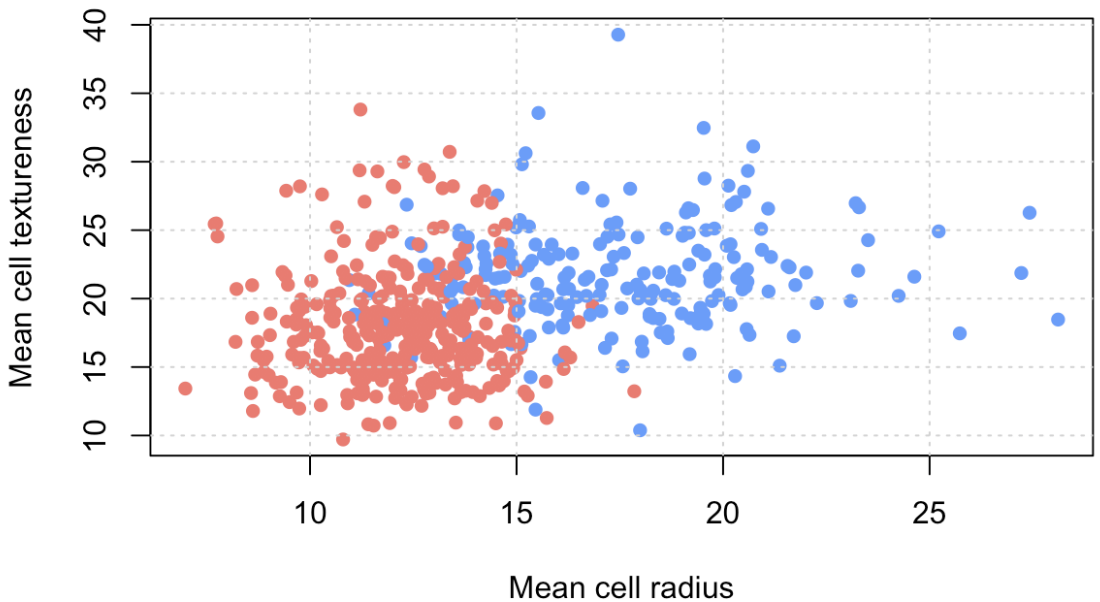
c) x2=x1;

d) x2=x1+2

Soft margin classifier/SVC

- But this requires that classes would be separable. This is never true in the real life ☹
- What if we have this data?

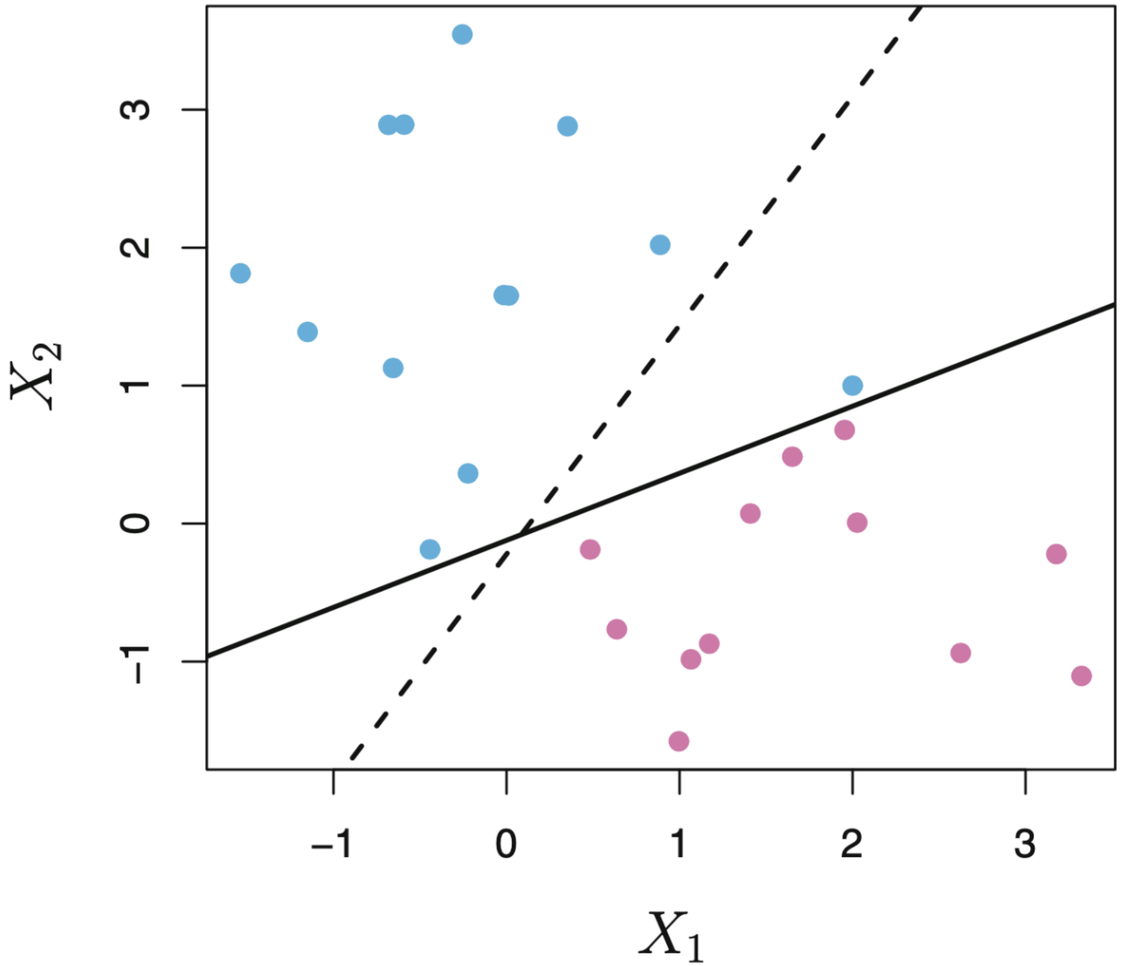




- Sacrifices must be made.

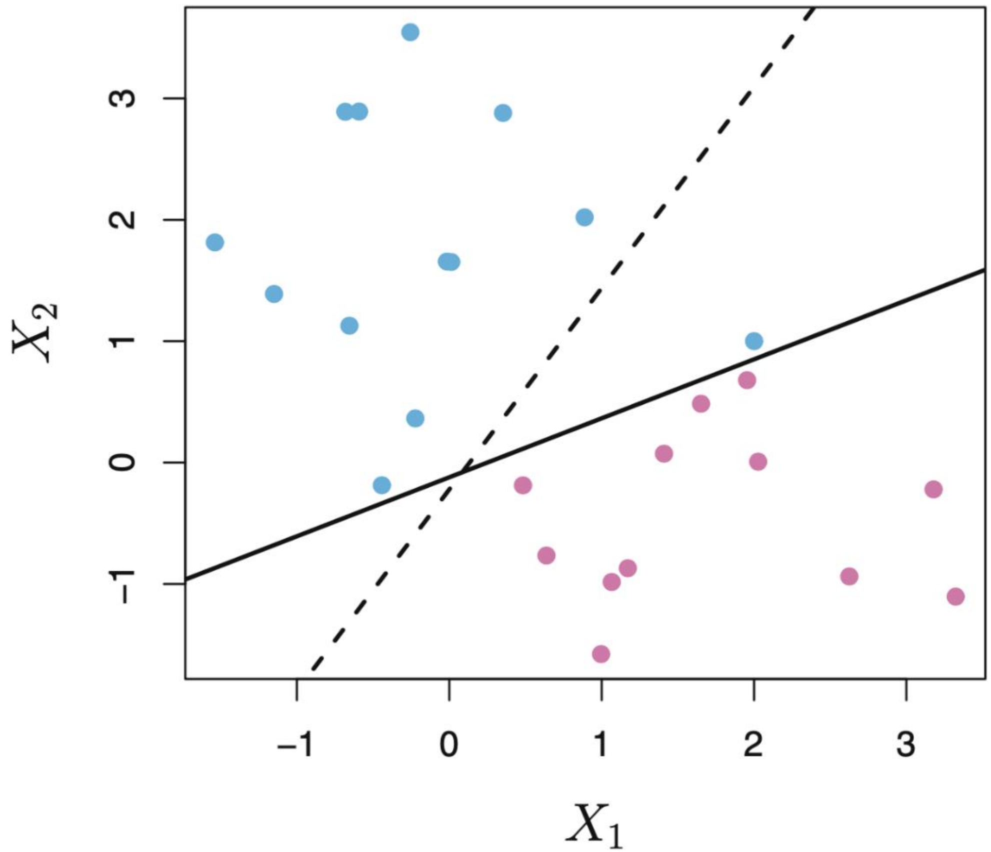
Soft margin classifier/SVC

- But this requires that classes would be separable. This is never true in the real life ☹
- Hard margin SVC may be possible but is it always desirable?



Soft margin classifier/SVC

- Allow few misclassifications in order to do a better job in classifying the remaining observations.
- The **support vector classifier**, sometimes called a **soft margin classifier**, does exactly this.



Soft margin classifier/SVC

- Hard margin optimization problem can be formally stated in terms of the loss function:

$$\sum_{i=1}^{N} E_{\infty}(y_i f(x_i) - 1) + \frac{1}{2}\|w\|^2$$

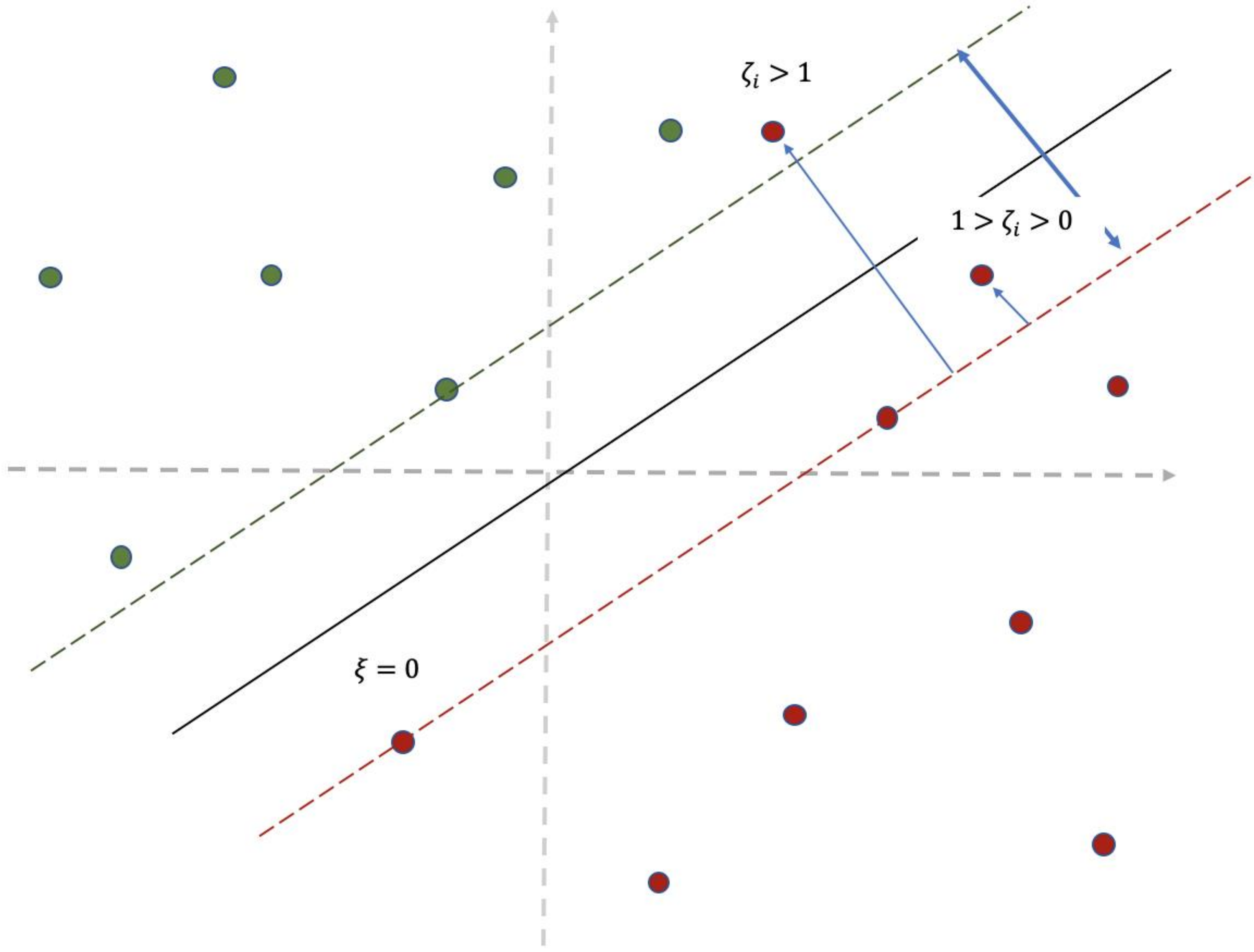
where $E_{\infty}(z)$ is 0 if $z \geq 0$ and ∞ otherwise.

- In case of overlapping classes, we might instead use a less strict error function to allow crossing of the margins.
- For this, we introduce a **slack variables** $\zeta_i \geq 0$ for each training data point.

Soft margin classifier/SVC

- Slack variables are 0 for datapoints which are on the correct side of the margin boundary and
$$\zeta_i = |y_i - f(x_i)|$$
- The slack variable ζ_i tells us where x_i is located relative to the hyperplane and the margin:
  a) If $\zeta_i = 0$, then the observation is on the correct side of the margin;
  b) If $\zeta_i > 0$, then the observation is on the wrong side of the margin;
  c) If $\zeta_i > 1$, then the observation is on the wrong side of the hyperplane

Slack variables



Soft margin classifier/SVC

- The constraints now become

$$y_i(w_0+x_i^T w) \geq 1-\zeta_i, \zeta_i \geq 0$$

- The goal is thus to maximize the margin while softly penalizing points that lie on the wrong side of the margin boundary:

$$C\sum_{i=1}^{N}\zeta_i + \frac{1}{2}\left||w|\right|^2$$

where $C > 0$ controls the tradeoff between the slack variable penalty and the margin. (In R or in Python it is called *cost*)

- Increasing C, more and more importance is given to the penalty term, thus one can expect narrowing margins.

Soft margin classifier/SVC

- Interestingly - the solution has the same form as before:

$$w = \sum_{i \in sv} \alpha_i y_i x_i$$

and

$$w_0 = \frac{1}{|sv|} \sum_{k \in sv} \left(\frac{1 - w^T x_k y_k}{y_k} \right).$$

Equation for the optimal hyperplane:

$$f(x) = w_0 + w^T x = w_0 + \sum_{i \in sv} \alpha_i y_i (x^T x_i)$$

Question 7

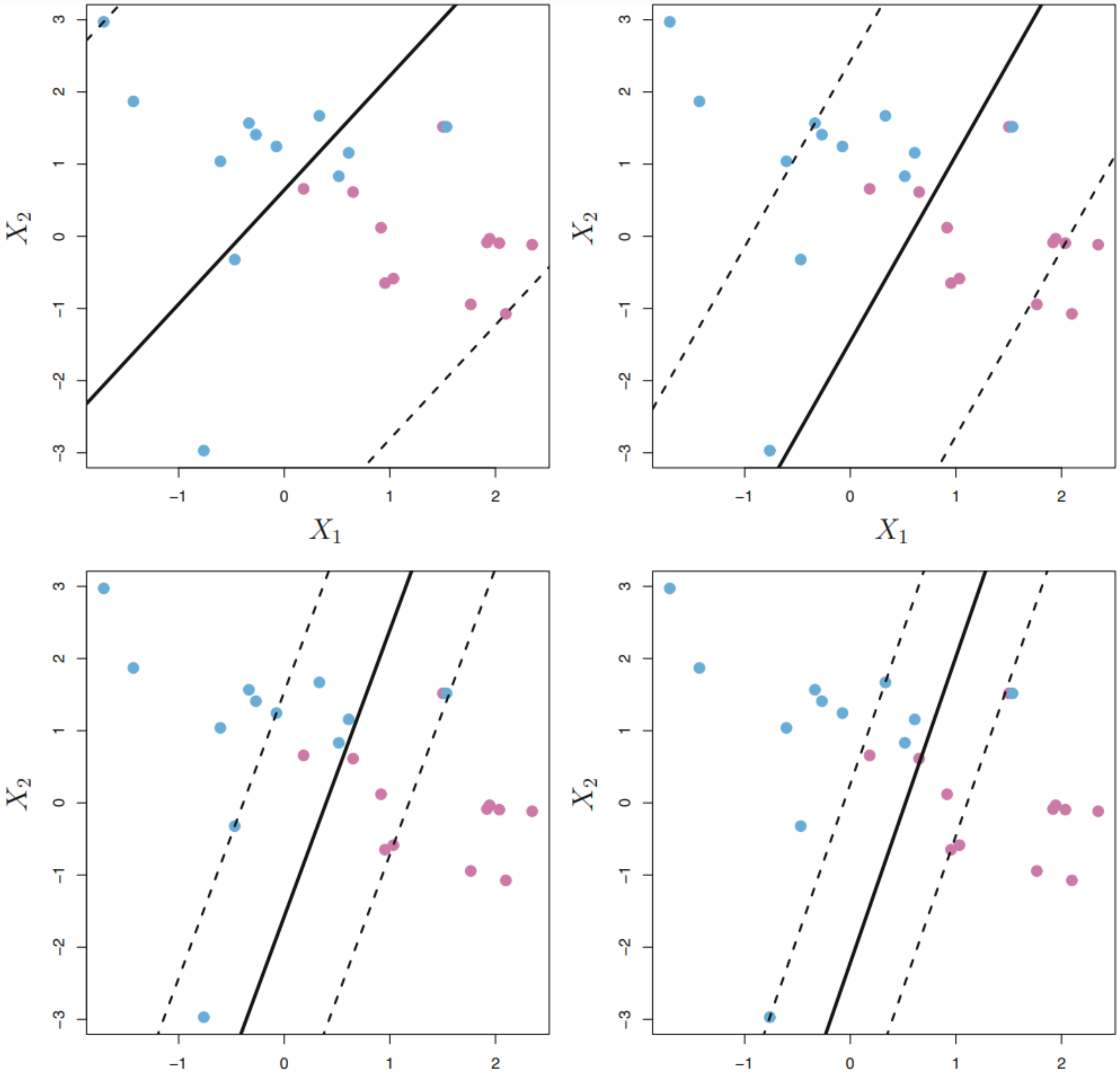
- Given the dataset below and slack variables, how many points **violated the margin but not the hyperplane** and **how many points were misclassified** by the hyperplane?

a) 9 and 5;
b) 4 and 5;
c) 3 and 2;
d) 9 and 2.

x_1	ζ
-1	0
2	0
-5	0.5
4	0.9
0	1.1
7	1.9
8	0
1.5	0.88
-2	0

- If $\zeta_i = 0$, then the observation is on the correct side of the margin;
- If $\zeta_i > 0$, then the observation is on the wrong side of the margin;
- If $\zeta_i > 1$, then the observation is on the wrong side of the hyperplane

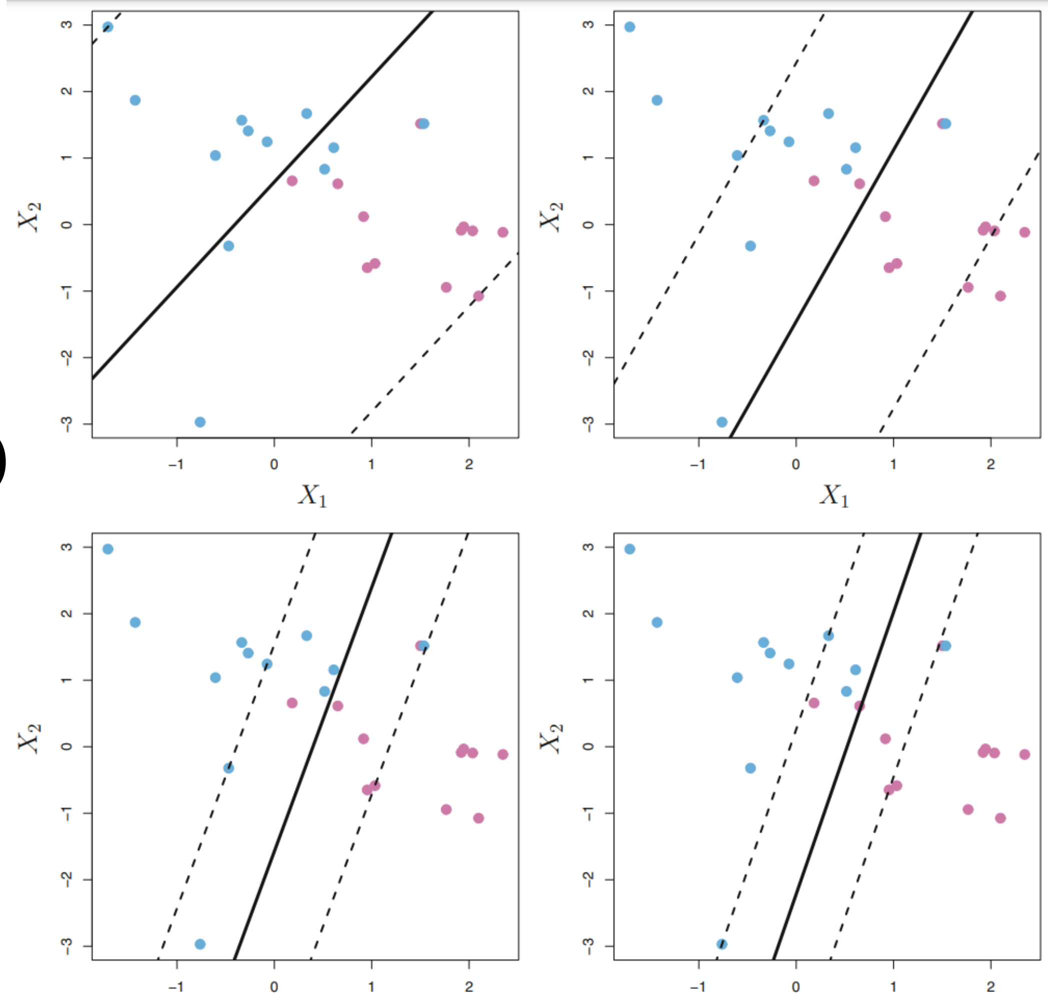
The C parameter



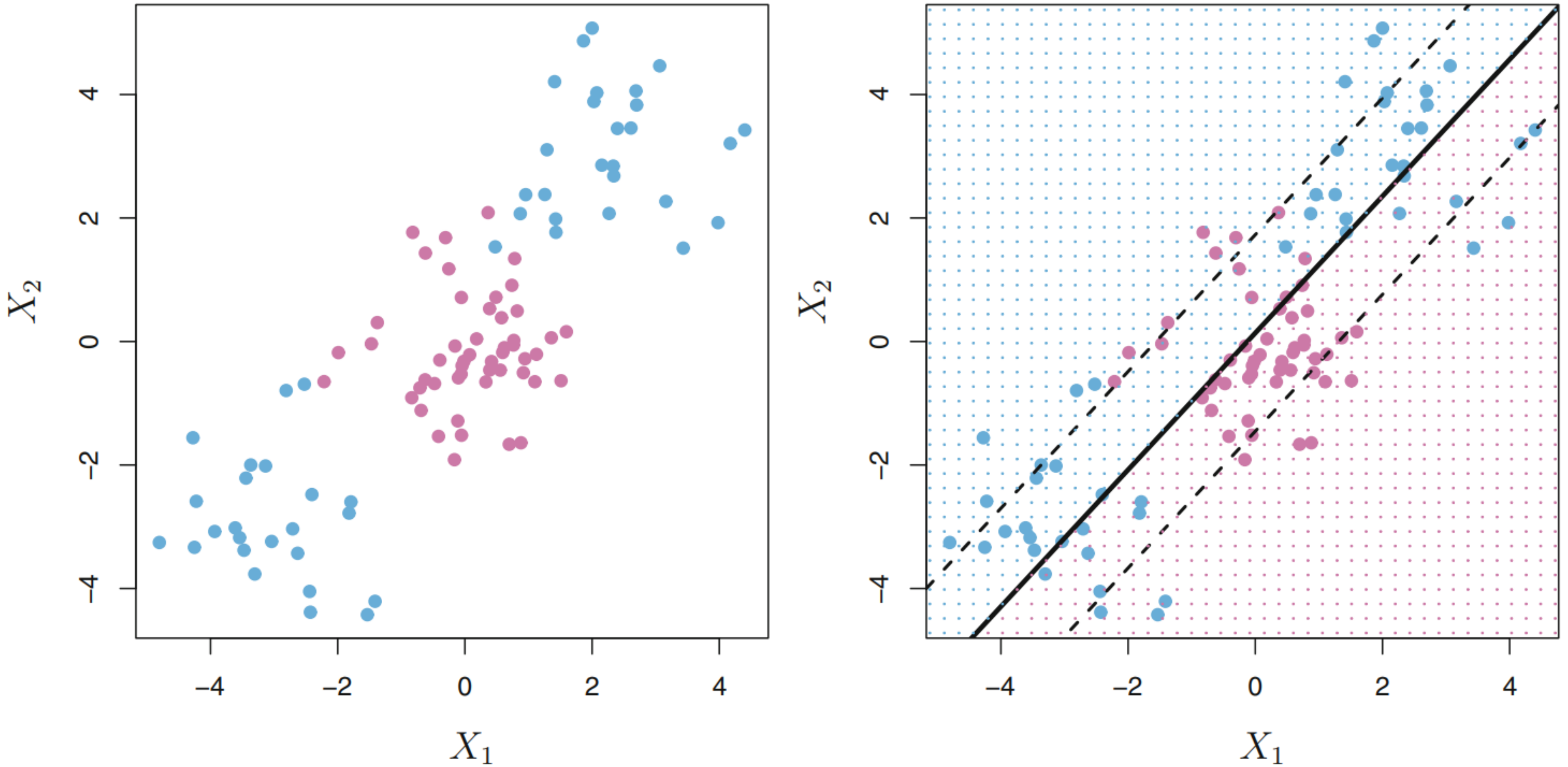
Question 8

- Which is true?
  a) Increasing C, decreases model bias and increases variance;
  b) Increasing C, increases model bias and decreases variance;

$$f(x) = w_0 + \sum_{i \in sv} \alpha_i y_i (x^T x_i)$$

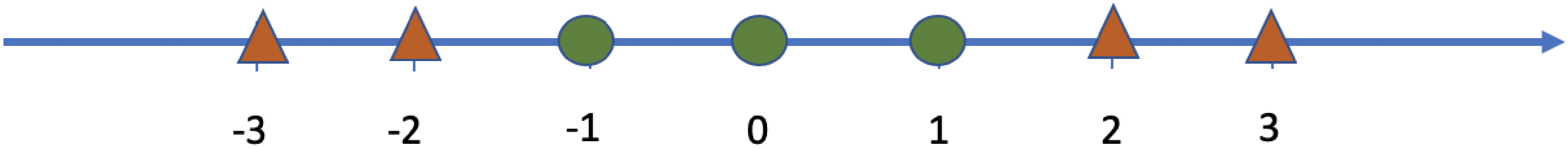


How about this data set?



SVM: non-linearly separable data

Consider this very simple dataset:

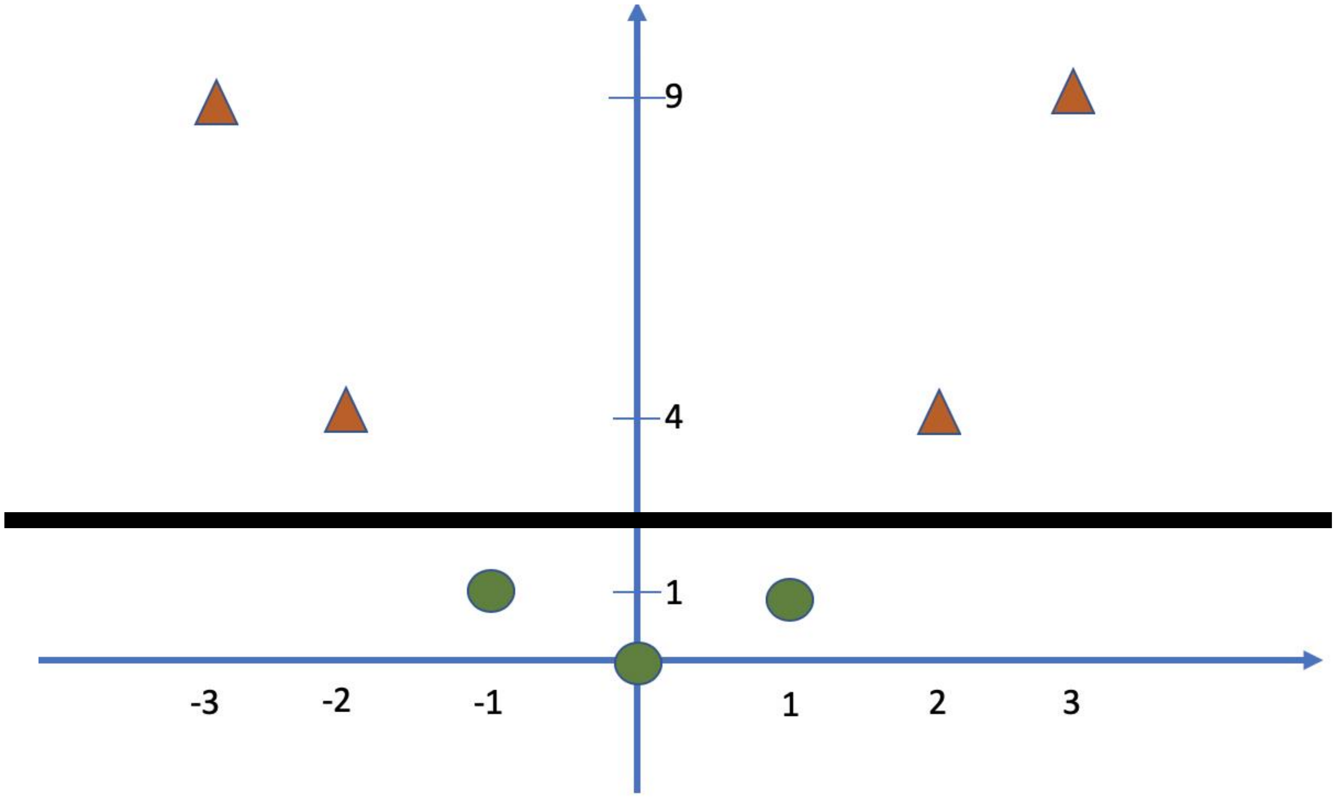


Feature value	Class
-3	▲
-2	▲
-1	●
0	●
1	●
2	▲
3	▲

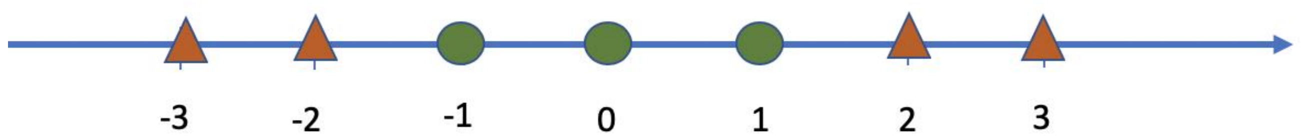
SVM: non-linearly separable data

Let's make the transformation $x \to (x, x^2)$. This makes classes separable in R^2.

2D space

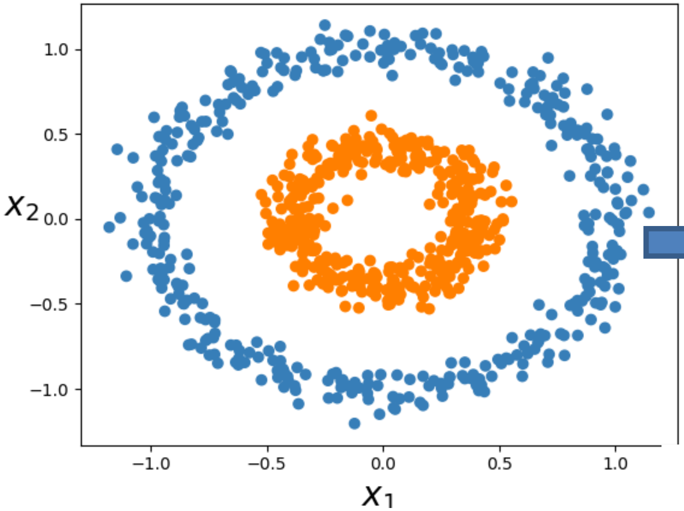


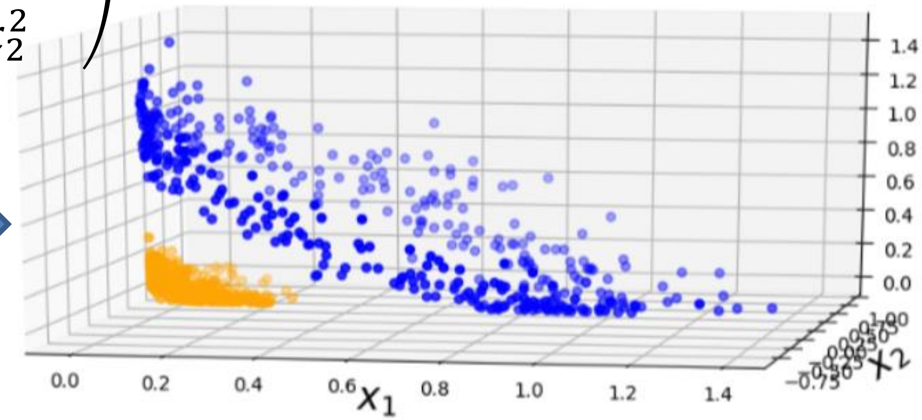
1D space

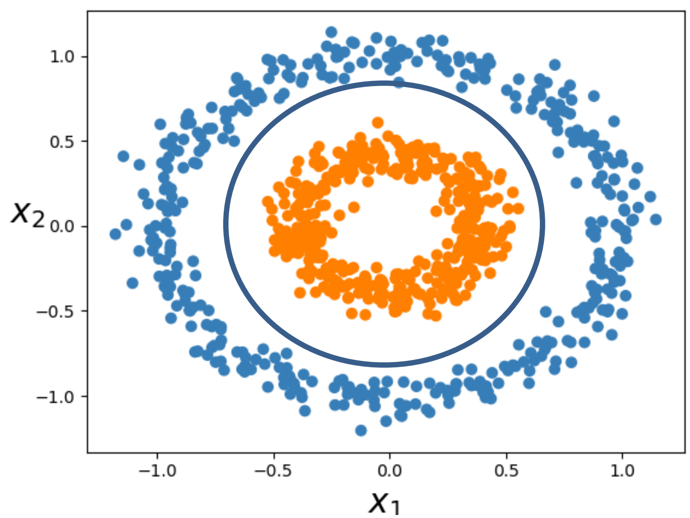


SVM: non-linearly separable data

$$\Phi\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} x_1^2 \\ \sqrt{2}x_1 x_2 \\ x_2^2 \end{pmatrix}$$



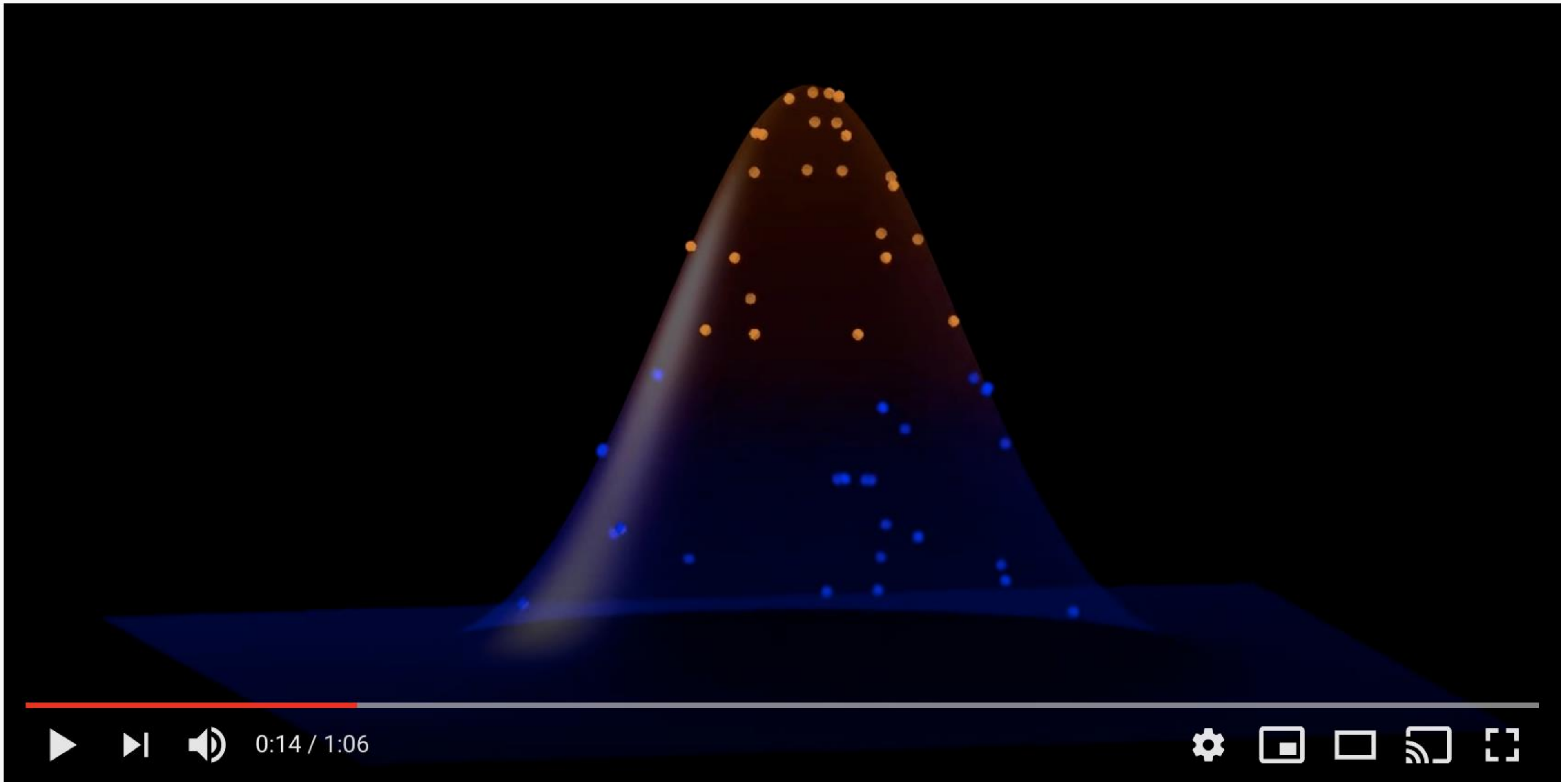




Φ^{-1}

SVM: data transformation

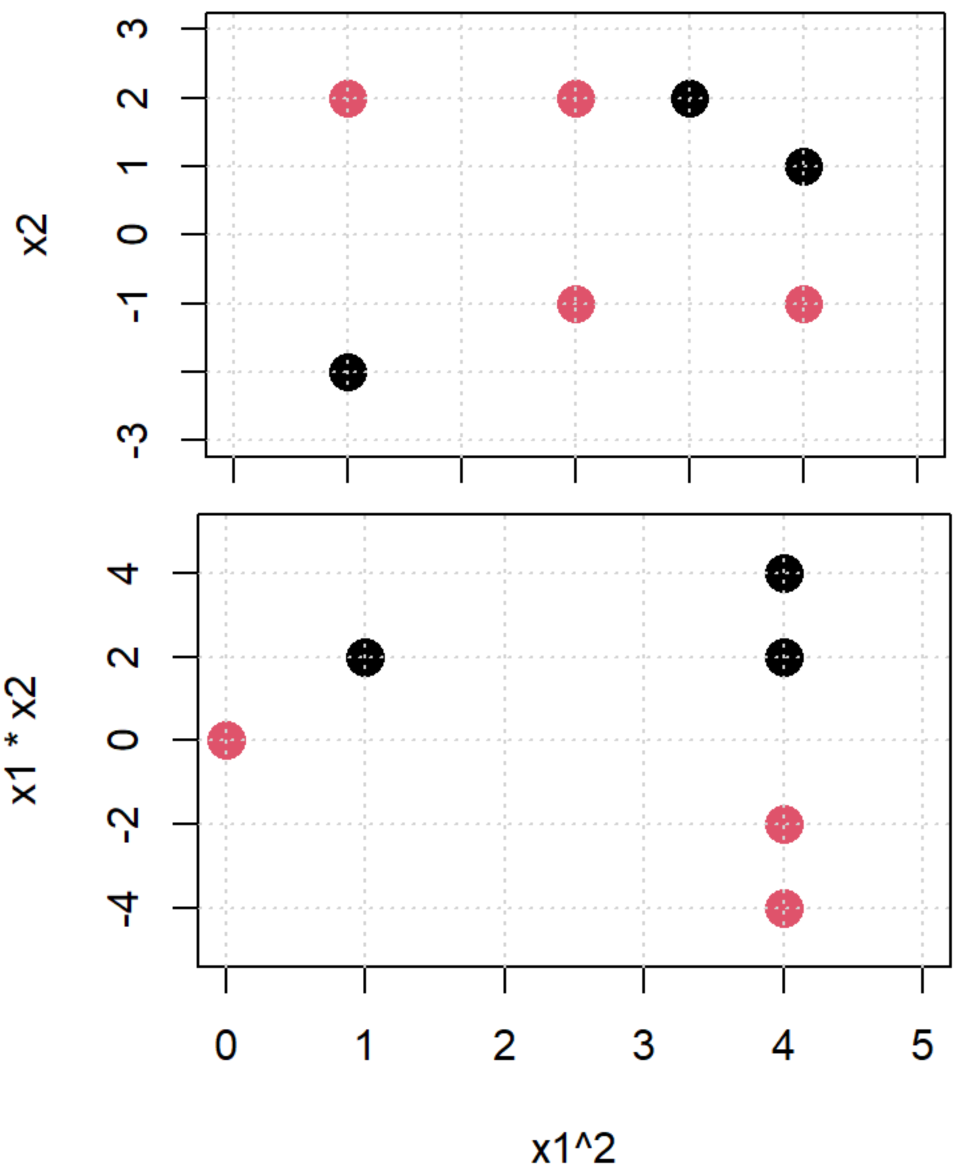
https://www.youtube.com/watch?v=9NrALgHFwTo&feature=emb_logo



SVM: data transformation

Consider a dataset and a transformation $\phi(x) = (x_1^2, x_1 x_2)^T$

x_1	x_2	Class
–2	–2	A
-2	-1	A
1	2	A
2	1	A
-2	2	B
0	2	B
0	-1	B
2	-1	B

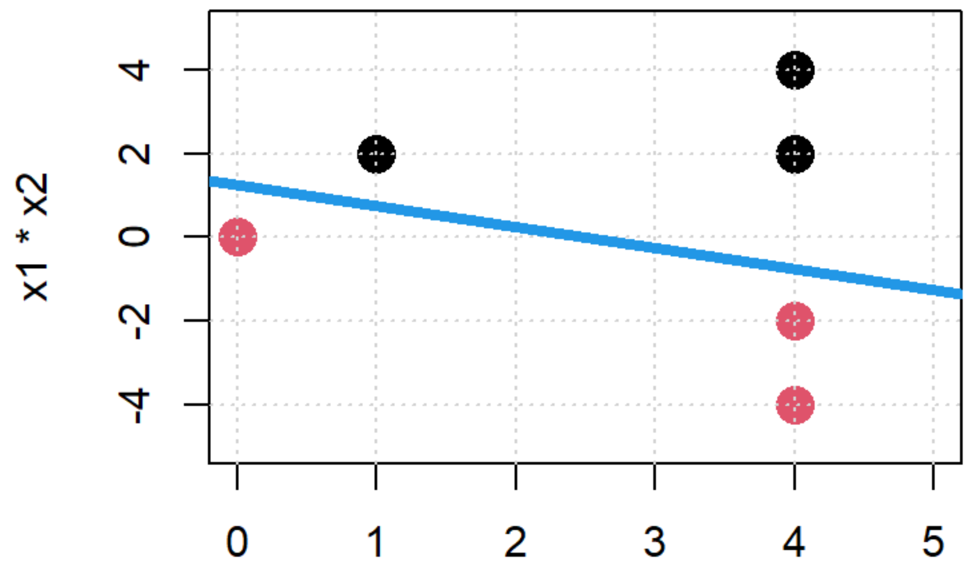


Original data

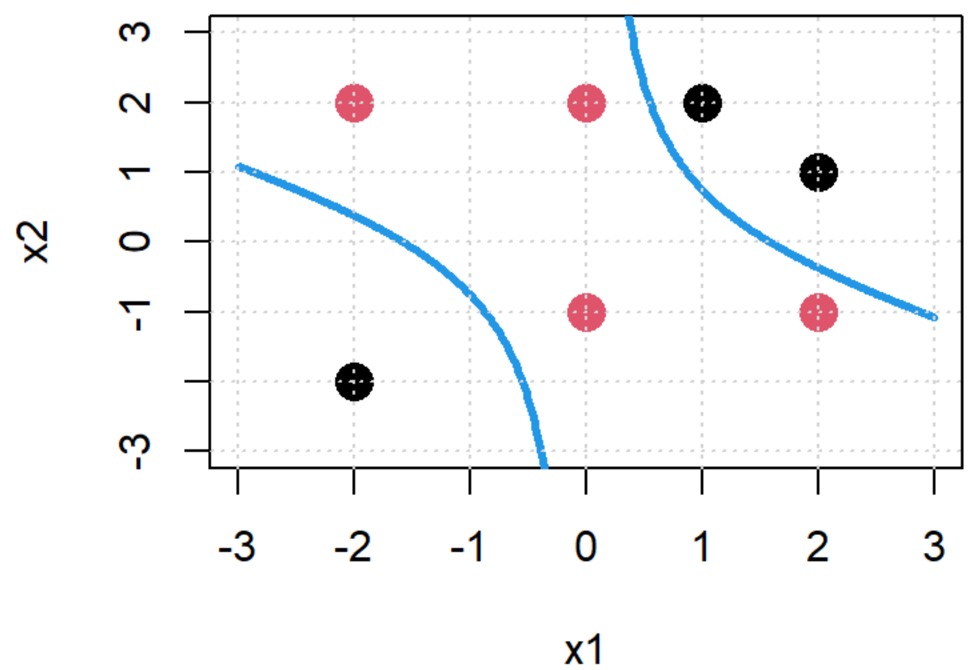
Transformed data

SVM: data transformation

Hard margin solution:
$x_2' = 1.25 - 0.5x_1'$



Solution in original space



SVM: Kernel trick

- Using transformations Φ and going back to the original space is highly inefficient, because we would have to calculate in very high dimensional spaces (possibly infinite dimensional).
- A kernel is a very special function which produces the same effect but without going back and forth between original space and those high dimensional spaces.
- A function $K(x, x')$ is called **a kernel** if there exists a transform ϕ such that $K(x, x') = \phi^T(x)\phi(x')$.

- Why kernel is important?

SVM: Kernel trick

- Lagrangian functional (in case of hard margin as well as soft margin problems) is only a function of features through the dot product:

$$x_i^T x_j$$

- If we transform features with $\phi(x)$, the Lagrangian is still only a function of a dot product:

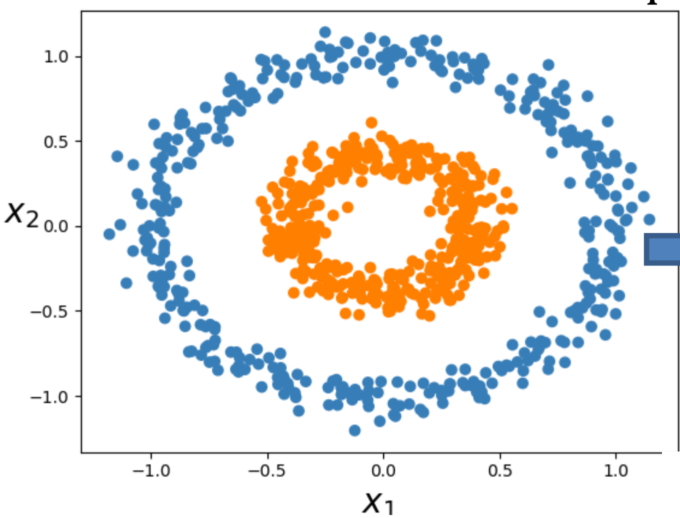
$$\phi^T(x_i)\phi(x_j)$$

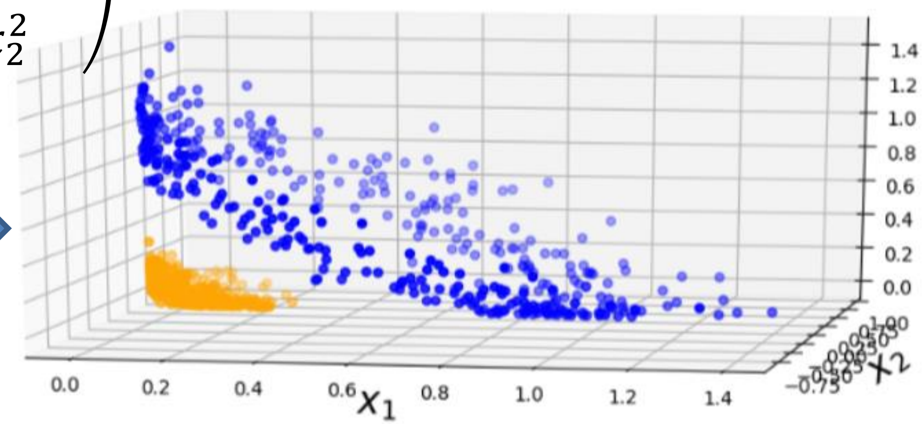
- Then let's exchange transformation process (and an inverse calculation) with the kernel

$$K\left(x_i, x_j\right) = \phi^T(x_i)\phi\left(x_j\right)$$

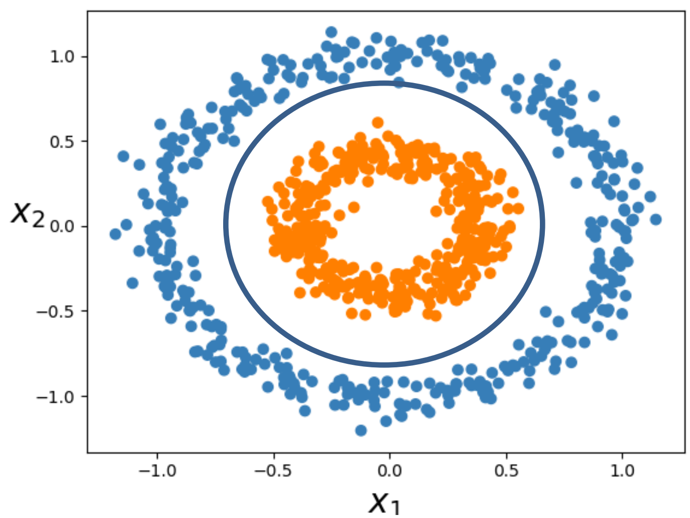
SVM: Kernel trick

$$\Phi\begin{pmatrix}x_1\\x_2\end{pmatrix} = \begin{pmatrix}x_1^2\\ \sqrt{2}x_1x_2\\ x_2^2\end{pmatrix}$$





Kernel



Φ^{-1}

SVM: Kernel trick

- Hard margin SVC solution:

$$f(x) = w_0 + \sum_{i \in sv} \alpha_i y_i (x^T x_i)$$

It depends on so called scalar product $x^T x_i$ of vectors in the original space.

- Hard margin SVC solution with transformation ϕ:

$$f(x) = w_0 + \sum_{i \in sv} \alpha_i y_i (\phi(x^T) \phi(x_i))$$

- What if we could find another function, which has the same value as scalar product, but without explicit transformation to higher dimensional space?

SVM: Kernel trick

- $\phi\binom{x_1}{x_2} = \begin{pmatrix} x_1^2 \\ \sqrt{2}x_1x_2 \\ x_2^2 \end{pmatrix}$; $\phi^T(x)\Phi(\mathrm{x}') = (x_1x_1' + x_2x_2')^2$
- So, instead of explicitly transforming the data into another space with ϕ and calculating scalar product, we can just stay in the original space and use $K(x, x') = (x_1x_1' + x_2x_2')^2$ function instead.
- We can completely abandon the notion of transformation to other spaces and simply use kernels all the time.
- The final solution of kernel SVM optimization problem is

$$f(x) = w_0 + \sum_{i \in sv} \alpha_i y_i K(x, x_i)$$

Kernels

- **Inhomogeneous polynomial kernel** of degree d
$$K(x,x') = (\langle x,x'\rangle + c)^d.$$
- Consider $m = 2$ and $d = 2$.
$$K(x,x') = (\langle x,x'\rangle + c)^2 =$$
$$(x_1 x'_1 + x_2 x'_2 + c)^2 =$$
$$\phi^T(x)\phi(x').$$

$$\phi(x) = \left(x_1^2, x_2^2, \sqrt{2}x_1x_2, \sqrt{2c}x_1, \sqrt{2c}x_2, c\right)^T.$$

Thus, the feature space, equivalent to the action of the kernel, is 6 dimensional.

- Assume that data points are images of size 16x16.
  - If $d = 2$ then we have 33670 dimensional feature space
  - $If\ d = 4$, we get 186 043 585 dimensions.

Kernels

- **Inhomogeneous polynomial kernel** of degree d
$$K(x, x') = (\langle x, x' \rangle + c)^d.$$
- A corresponding transformation
$$\phi(x) = \left(x_1^2, x_2^2, \sqrt{2}x_1 x_2, \sqrt{2c}x_1, \sqrt{2c}x_2, c\right)^T.$$
- Let $x = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, x' = \begin{pmatrix} 0 \\ -1 \end{pmatrix}, \mathrm{c} = 2, \mathrm{d} = 2.$
- Corresponding transformations are:
$$\phi(x) = (1,0,0,2,0,2\ )^T$$
$$\phi(x') = (0,1,0,0,-2,2\ )^T$$
- Scalar product: $\langle \phi(x), \phi(x') \rangle = 0 + 0 + 0 + 0 + 0 + 4 = 4$
- Kernel function value: $K(x, x') = \big((0 + 0) + 2\big)^2 = 4$

Kernels

- A very popular Gaussian radial basis function

$$K(x, x') = exp\left\{-\frac{\|x - x'\|^2}{2\sigma^2}\right\}$$

corresponds to the infinite dimensional feature space!

- Let $x \in R^1$ and $\frac{1}{2\sigma^2} = 1$, then

$$\phi(x) = e^{-x^2}\left[1, \sqrt{\frac{2}{1!}}x, \sqrt{\frac{2^2}{2!}}x^2, \sqrt{\frac{2^3}{3!}}x^3, \cdots\right]^T$$

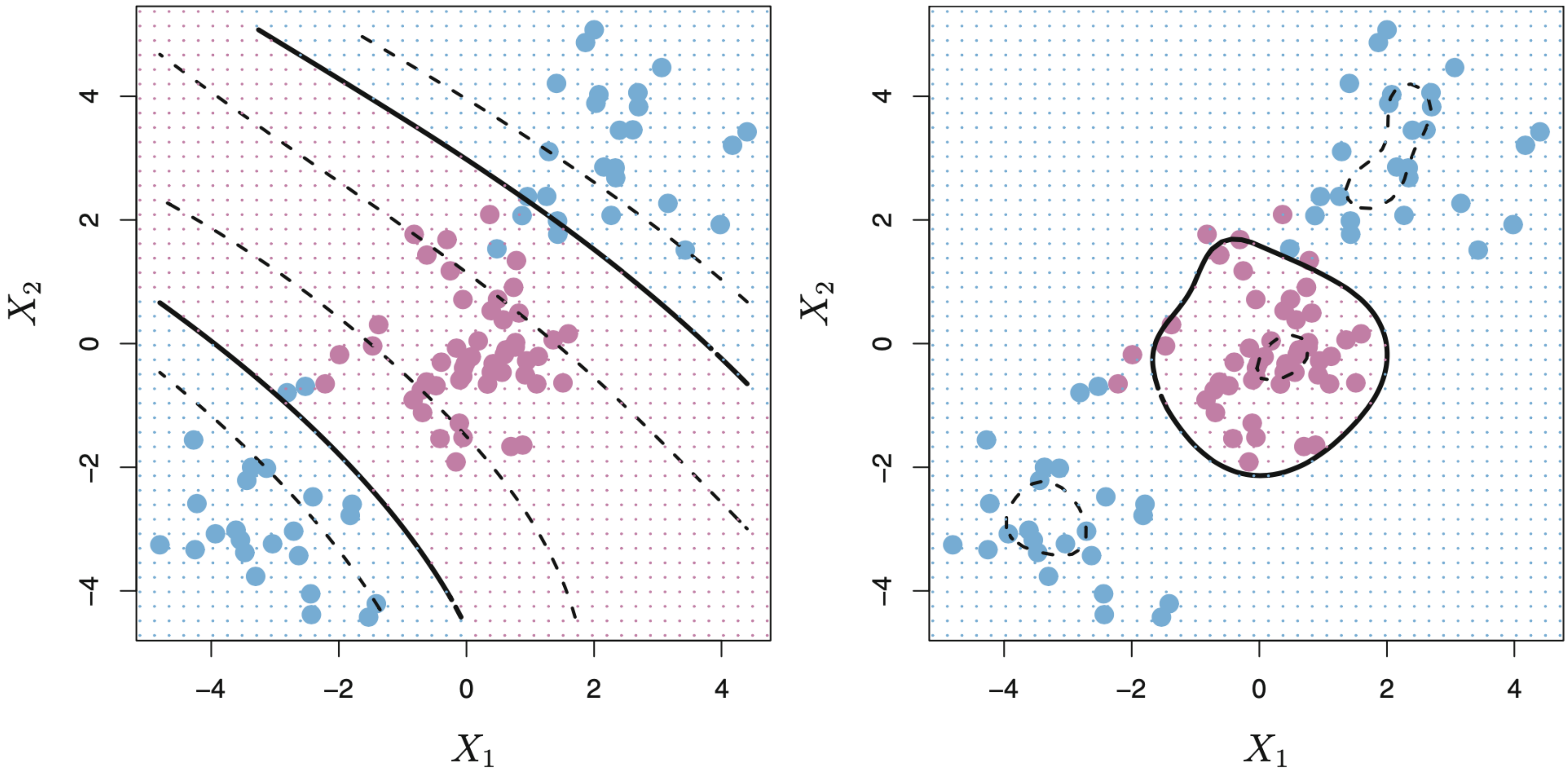
…which is an infinite-dimensional transformation!

SVM: Kernel trick

- All kernels have additional hyperparameters.
- So, controlling SVM involves adjustment of C and kernel hyperparameters.
- Often the search for best combination of these parameters involves 10-fold CV.

```
- Detailed performance results
  cost error dispersion
1 0.01  0.55  0.4377975
2 0.03  0.45  0.3689324
3 0.05  0.30  0.2581989
4 0.10  0.05  0.1581139
5 0.30  0.10  0.2108185
6 0.50  0.10  0.2108185
7 1.00  0.15  0.2415229
8 3.00  0.10  0.2108185
```

SVM: Kernel trick



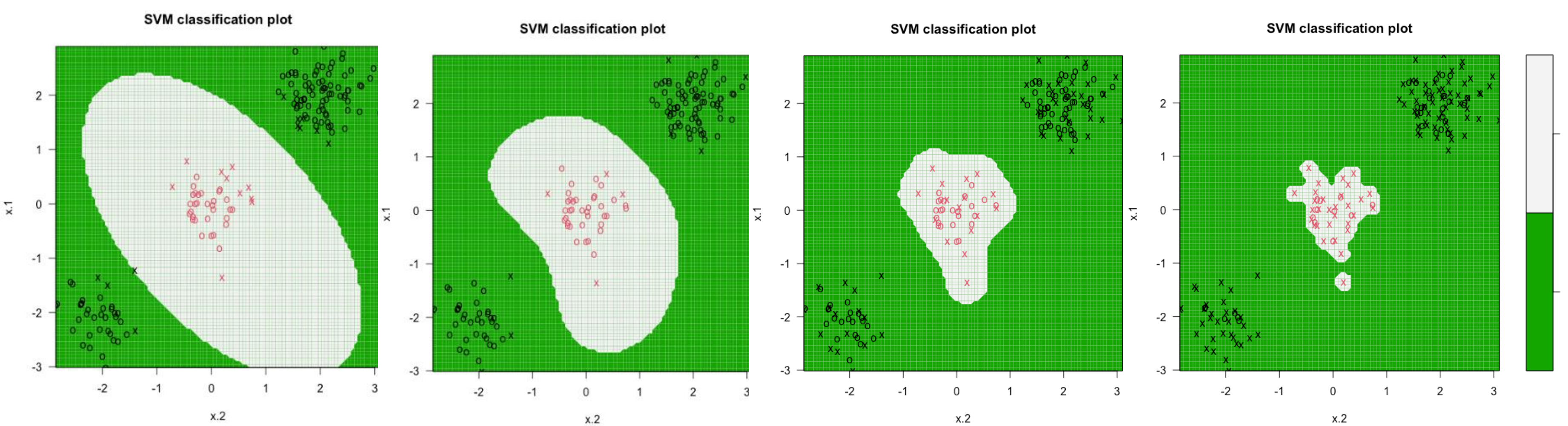
Polynomial kernel of degree 3

Radial kernel

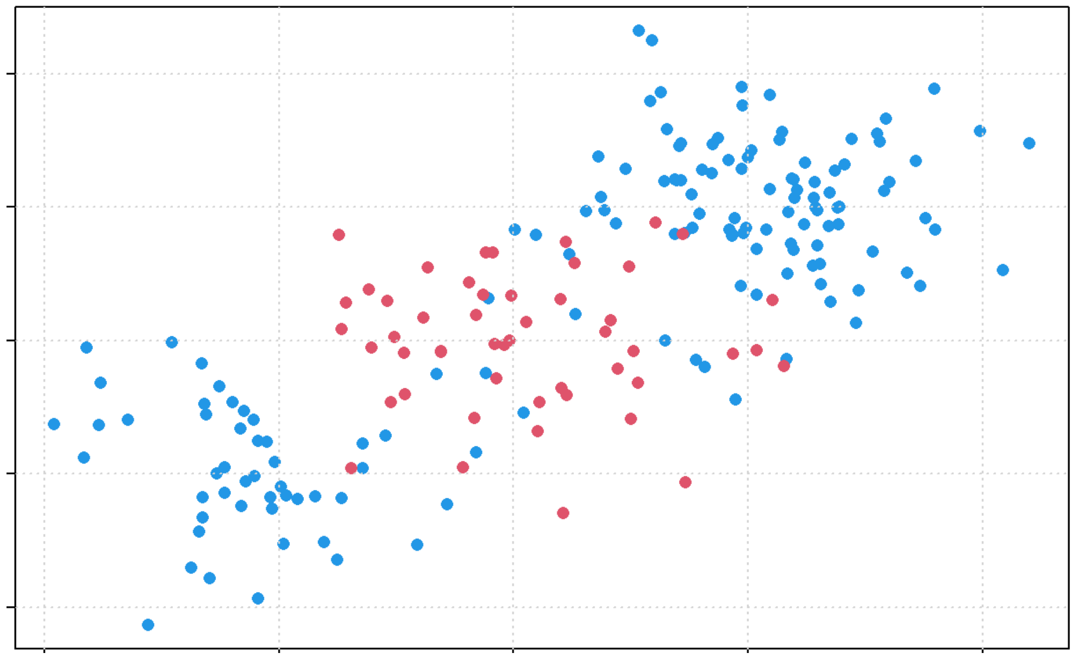
SVM: Radial kernel

Kernel: $K(x, y) = exp\left\{-\frac{\|x-y\|^2}{2\sigma^2}\right\}$

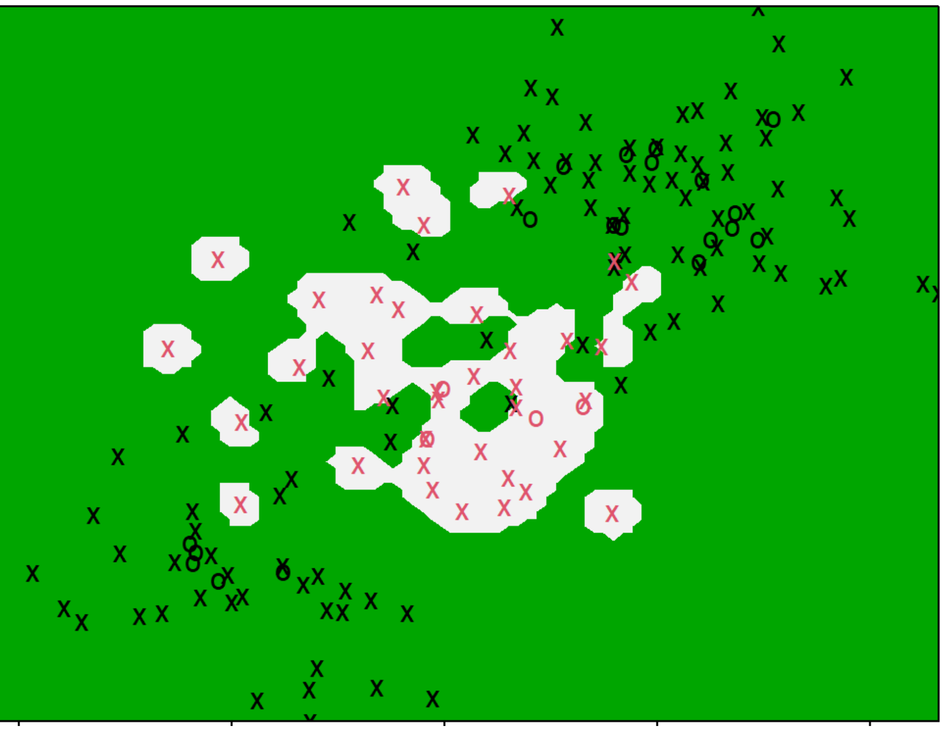
Small values of σ increases the fit to the training data

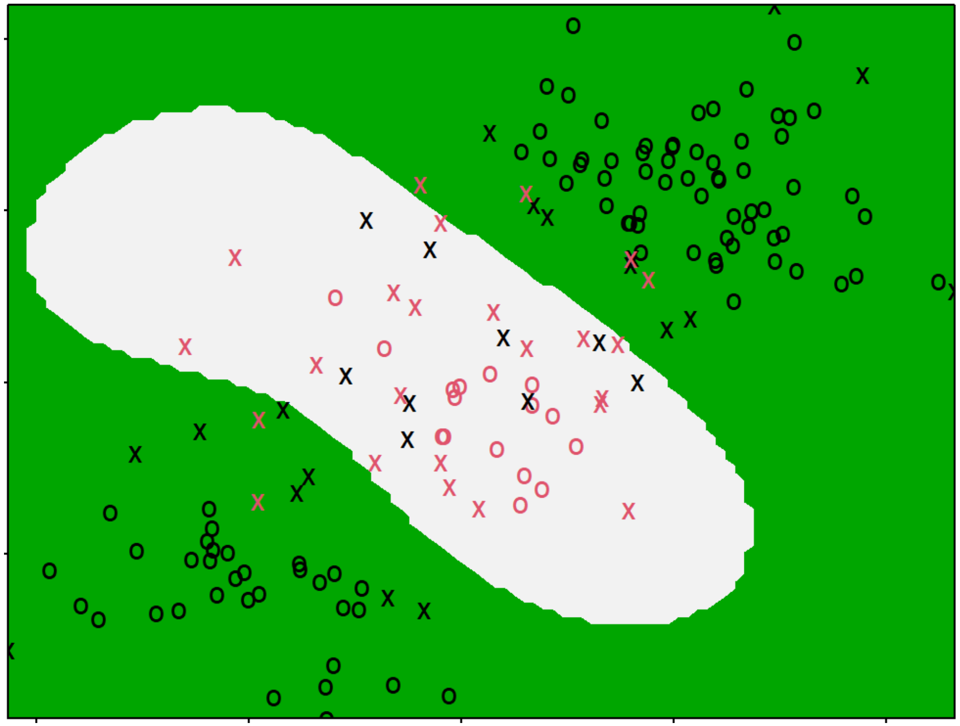


Decreasing σ $(=\frac{1}{gamma}$, as '*gamma*' is used in R)



Which case is better?





SVM: More than one class

- One vs. one. A separate SVM is fitted to each combination of class pairs. The prediction is obtained by majority vote.
- Say, we have 3 classes. Then we have to construct three different SVMs’:

$$\{1,2\}, \{1,3\}, \{2,3\}$$

- Prediction for x_0 is carried out by the majority rule.

Support vector methods: summary

- **Hard margin classifier**: when classes are linearly separable;
- **Soft margin classifier** or **Support Vector Classifier**: When classes are not linearly separable, but still a linear boundary is reasonable;
- **Support Vector Machines**: When classes are not linearly separable and a linear boundary is not a reasonable choice;

  Note: SVC is a case of SVM with linear kernel.

Vocabulary

- **Linear separability** – tiesinis atskiriamumas (separabilumas);
- **Support vectors** – atraminiai vektoriai;
- **Margin** – paraštė;
- **Hard margin** – kieta paraštė;
- **Soft margin** – minkšta paraštė.
- **Kernel** – branduolys;
- **slack variable** – laisvumo parametras.