



UNIVERSIDADE
CATÓLICA
PORTUGUESA

BRAGA

Machine Learning

Session 17 - T

Support Vector Machines – Part 2

Ciência de Dados Aplicada

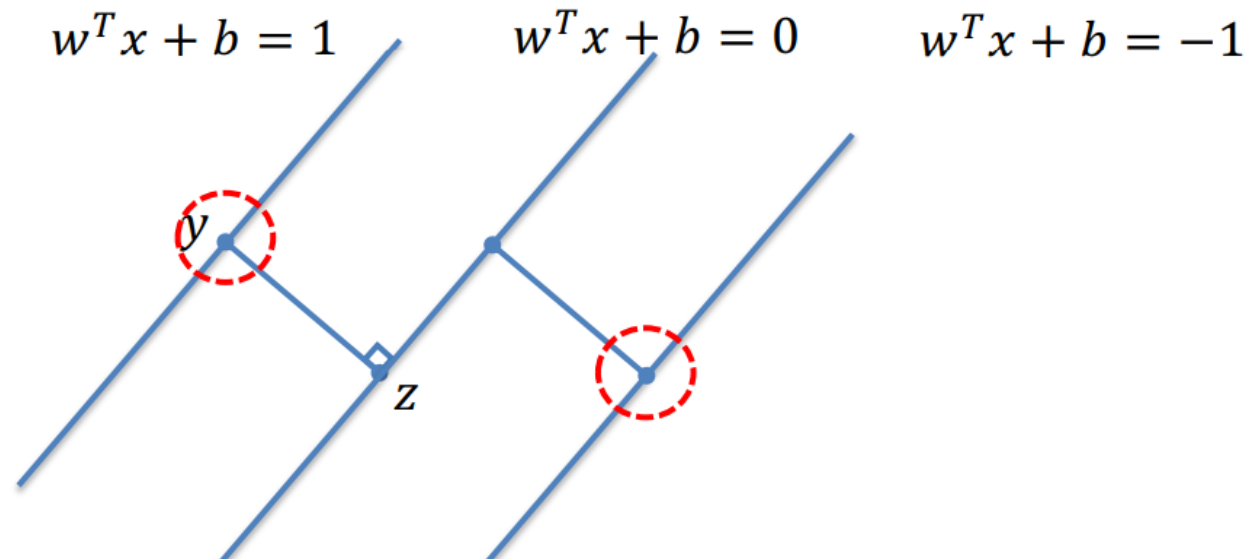
2023/2024

SVMs - Recap

$$\min_{w,b} \|w\|^2$$

such that

$$y^{(i)}(w^T x^{(i)} + b) \geq 1, \text{ for all } i$$

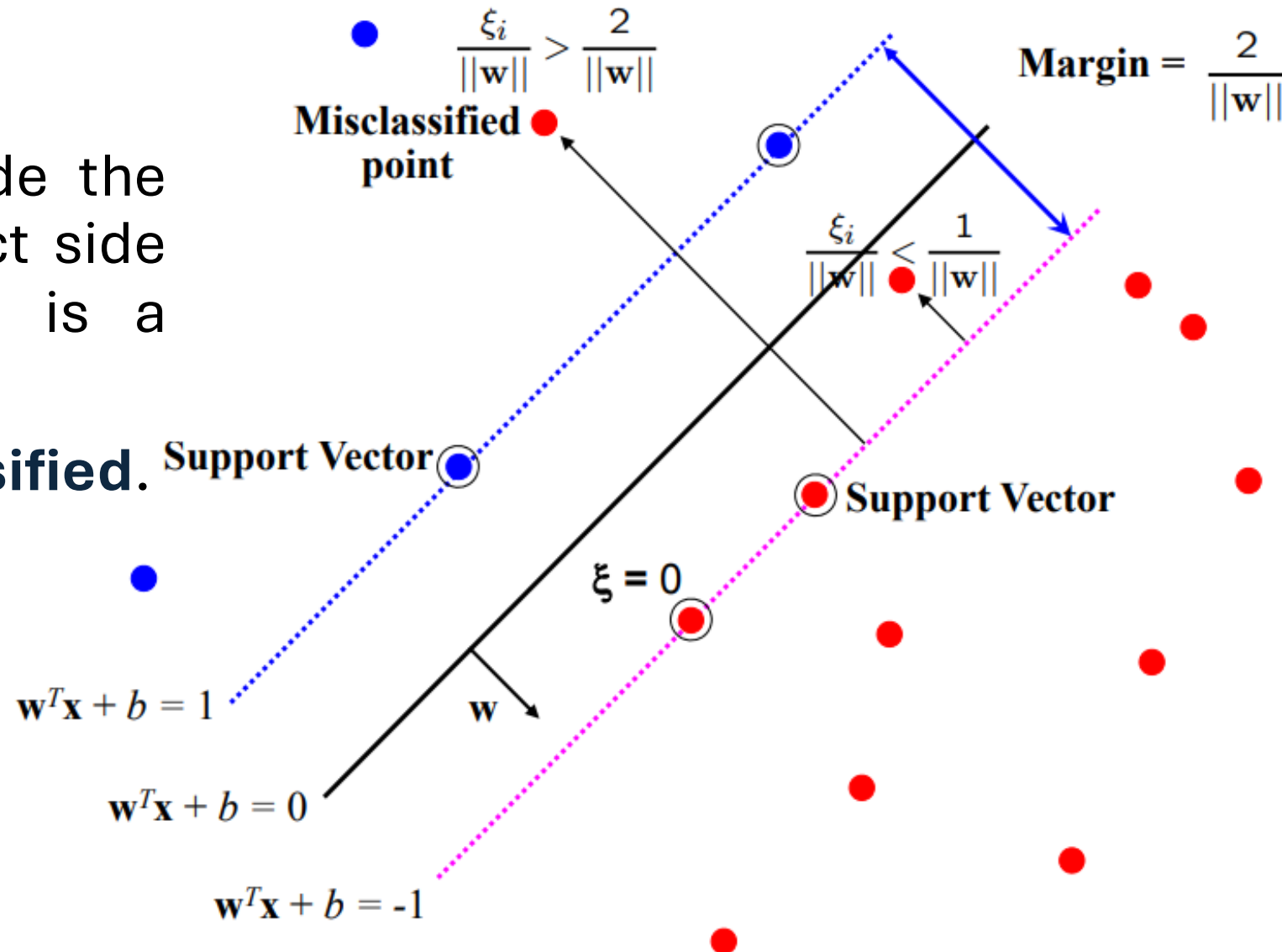


SVMs – Slack Variables

$$\xi_i \geq 0$$

- For $0 < \xi \leq 1$ point is inside the margin but on the correct side of the hyperplate. This is a **margin violation**;
- For $\xi > 1$ point is **misclassified**.

ξ allows margin violations or misclassified points, but with a **penalty**!



SVMs - Soft Margin Solution

- The optimization problem becomes

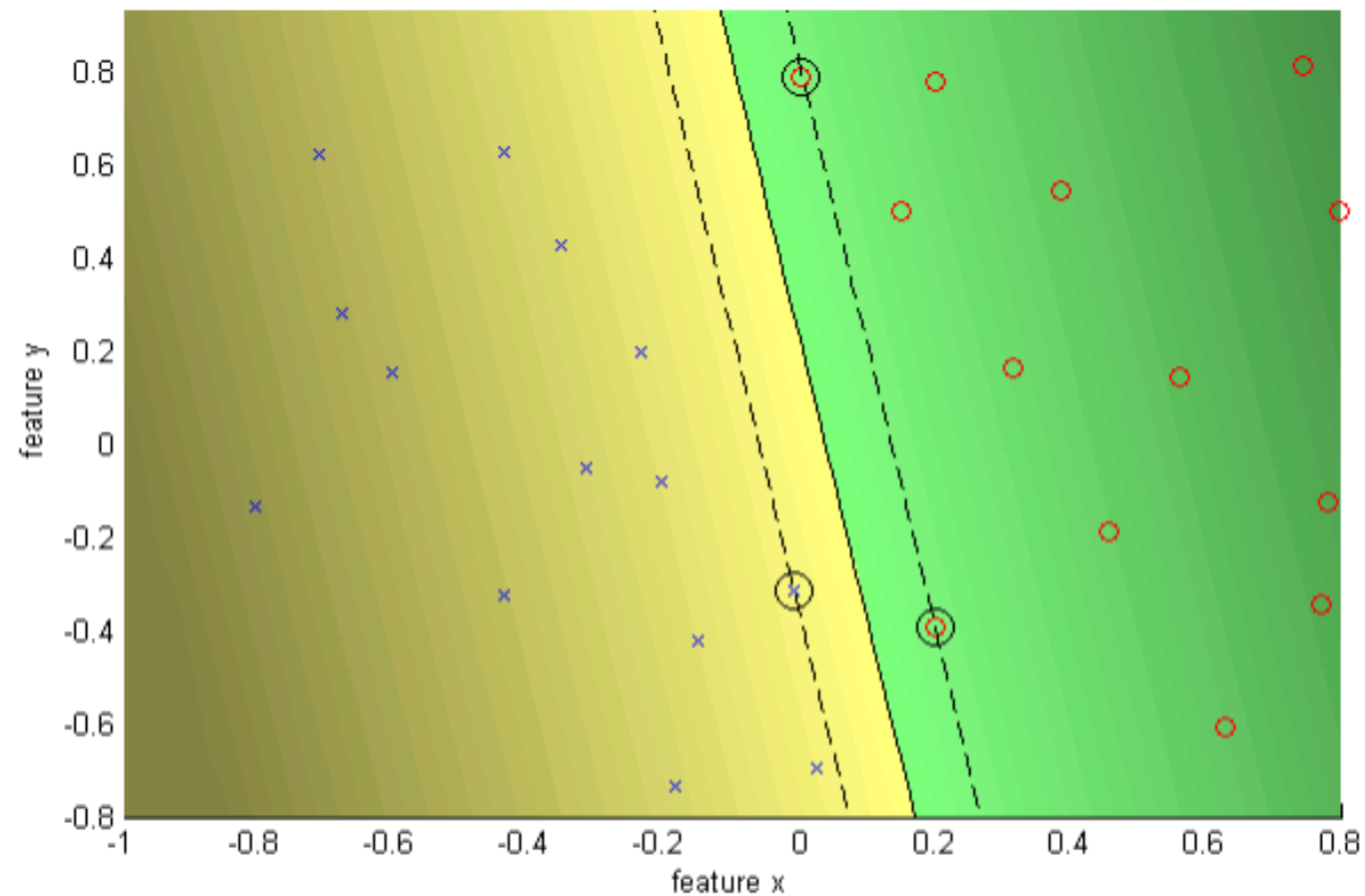
$$\min_{\mathbf{w} \in \mathbb{R}^d, \xi_i \in \mathbb{R}^+} ||\mathbf{w}'||^2 + C \sum_i^N \xi_i$$

such that $y_i (\mathbf{w}' \mathbf{x}_i + b) \geq 1 - \xi_i$ for $i = 1 \dots N$

- Every constraint can be satisfied if ξ_i is sufficiently large.
- C is a regularization parameter:
 - **Small C** allows constraints to be easily ignored \rightarrow **large margin**
 - **Large C** makes constraints hard to ignore \rightarrow **narrow margin**
 - **C = ∞** enforces all constraints \rightarrow **hard margin**

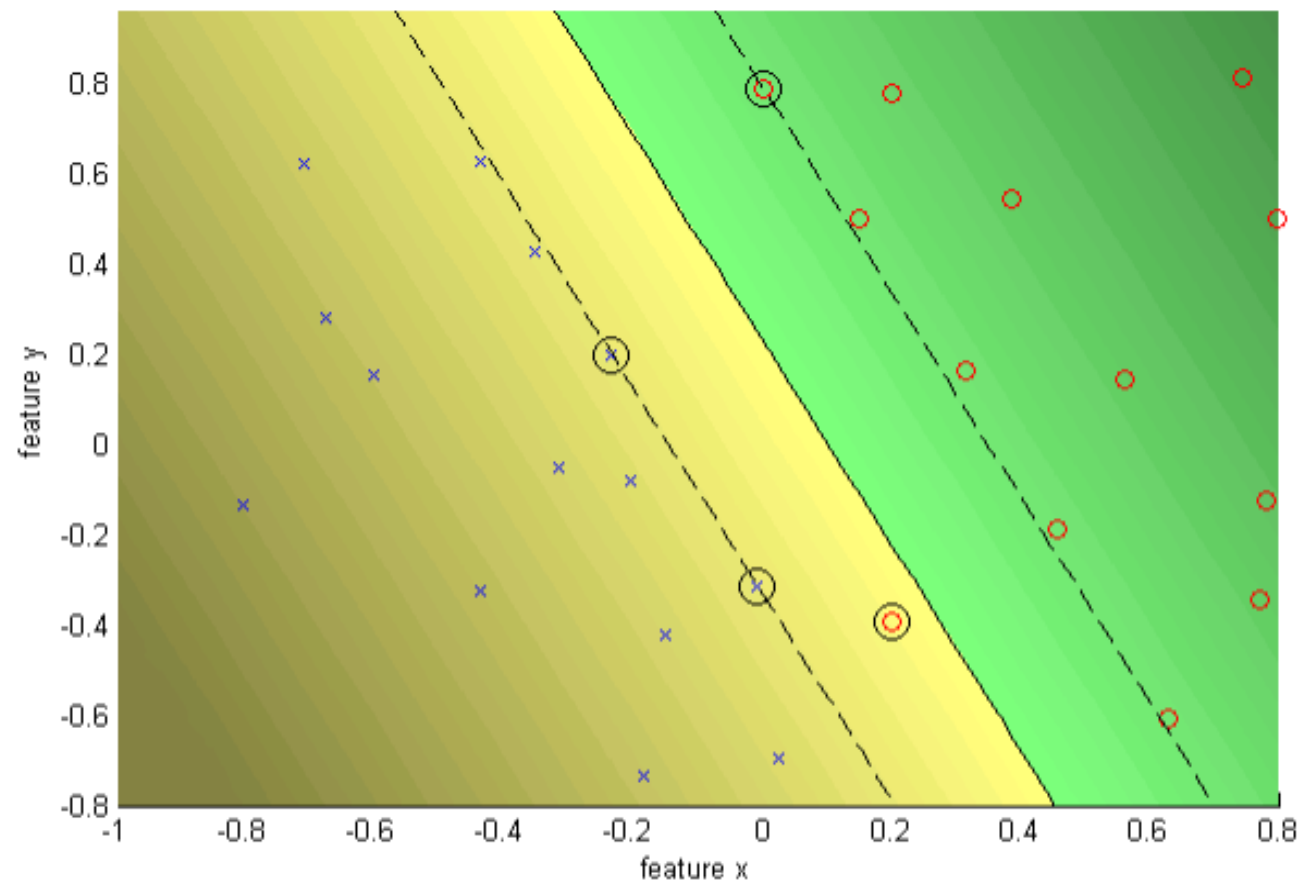
SVMs - C

$C = \text{Infinity}$ hard margin



SVMs - C

$C = 10$ soft margin



SVMs - Optimization

- Learning an SVM has been formulated as a **constrained** optimization problem over \mathbf{w} and ξ

$$\min_{\mathbf{w} \in \mathbb{R}^d, \xi_i \in \mathbb{R}^+} \|\mathbf{w}\|^2 + C \sum_i^N \xi_i \text{ subject to } y_i (\mathbf{w}^\top \mathbf{x}_i + b) \geq 1 - \xi_i \text{ for } i = 1 \dots N$$

- The constraint $y_i (\mathbf{w}^\top \mathbf{x}_i + b) \geq 1 - \xi_i$, can be written more concisely as:

$$y_i f(\mathbf{x}_i) \geq 1 - \xi_i$$

which, together with $\xi_i \geq 0$, is equivalent to:

$$\xi_i = \max(0, 1 - y_i f(\mathbf{x}_i))$$

SVMs - Optimization

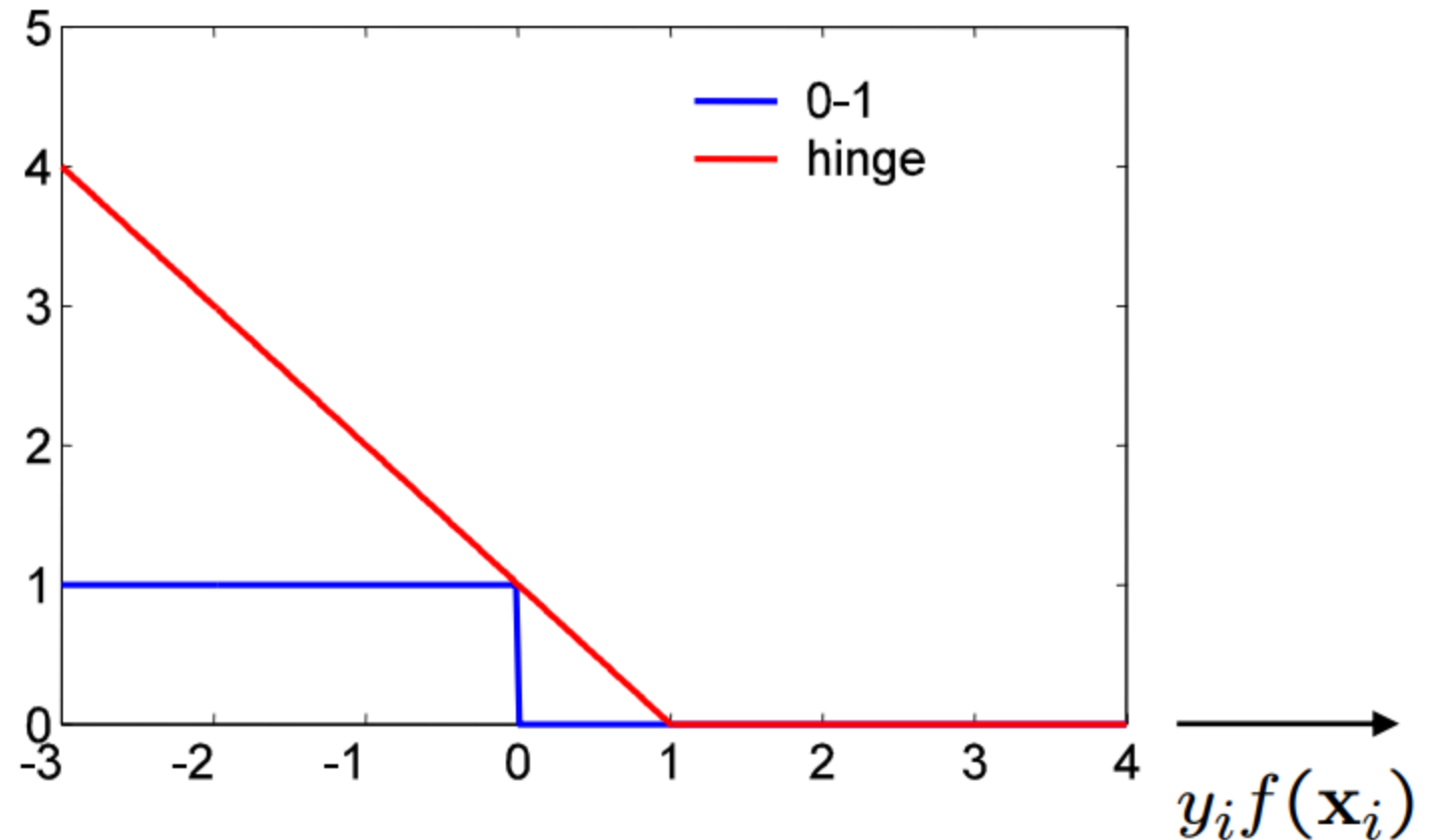
- Hence, the learning problem is equivalent to the **unconstrained** optimization problem over w :

$$\min_{w \in \mathbb{R}^d} \underbrace{\|w\|^2}_{\text{regularization}} + C \sum_i^N \underbrace{\max(0, 1 - y_i f(x_i))}_{\text{loss function}}$$

- If $y_i f(x_i) > 1$:**
 - Point is **outside the margin**. **No contribution to the loss.**
- If $y_i f(x_i) = 1$:**
 - Point is **on the margin**. **No contribution to the loss.**
- If $y_i f(x_i) < 1$:**
 - Point **violates the margin** constraint. **Contributes to the loss.**

SVMs – Hinge Loss

- SVMs uses the **Hinge Loss** $\rightarrow \max(0, 1 - y_i f(x_i))$
- Variation of the 0-1 loss.



SVMs – Dual Form

- The previous quadratic optimization problem is known as the **primal** problem.
- Instead, the SVM can be formulated to learn a linear classifier:

$$f(\mathbf{x}) = \sum_i^N \alpha_i y_i (\mathbf{x}_i^\top \mathbf{x}) + b$$

by solving a n optimization problem over α_i .

- This is known as the **dual** problem!

SVMs – Dual Form

- The [Representer Theorem](#) states that the solution w can always be written as a linear combination of the training data:

$$\mathbf{w} = \sum_{j=1}^N \alpha_j y_j \mathbf{x}_j$$

- If we substitute w in $f(x) = w^T x + b$

$$f(x) = \left(\sum_{j=1}^N \alpha_j y_j \mathbf{x}_j \right)^T \mathbf{x} + b = \sum_{j=1}^N \alpha_j y_j (\mathbf{x}_j^T \mathbf{x}) + b$$

- And for w in the cost function $\min_w ||w||^2$ subject to $y_i(w^T x_i + b) \geq 1$

$$||\mathbf{w}||^2 = \left\{ \sum_j \alpha_j y_j \mathbf{x}_j \right\}^T \left\{ \sum_k \alpha_k y_k \mathbf{x}_k \right\} = \sum_{jk} \alpha_j \alpha_k y_j y_k (\mathbf{x}_j^T \mathbf{x}_k)$$

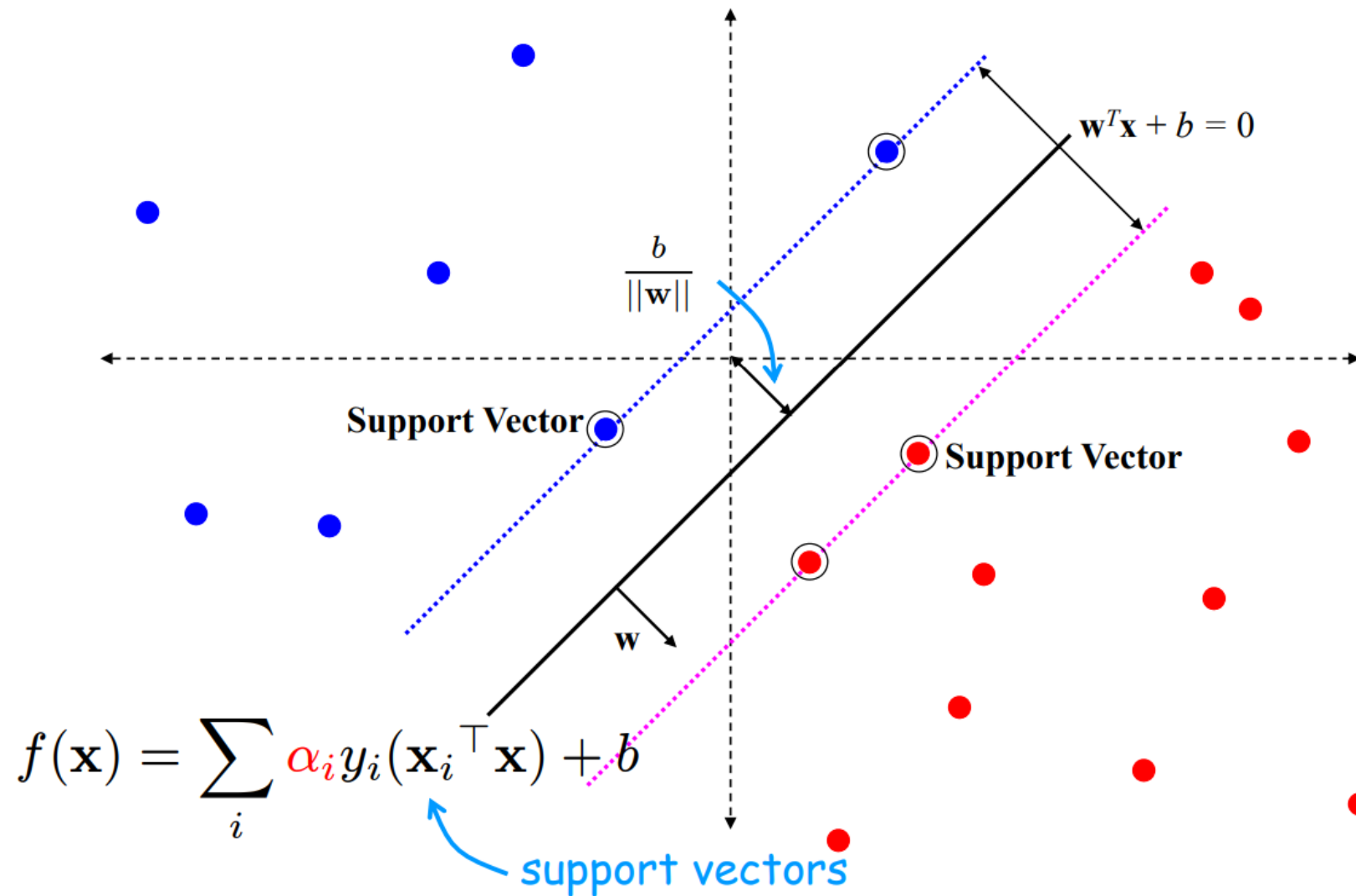
SVMs – Dual Form

- Hence, a equivalent optimization problem over α_j

$$\min_{\alpha_j} \sum_{j,k} \alpha_j \alpha_k y_j y_k (\mathbf{x}_j^\top \mathbf{x}_k) \quad \text{subject to} \quad y_i \left(\sum_{j=1}^N \alpha_j y_j (\mathbf{x}_j^\top \mathbf{x}_i) + b \right) \geq 1, \forall i$$

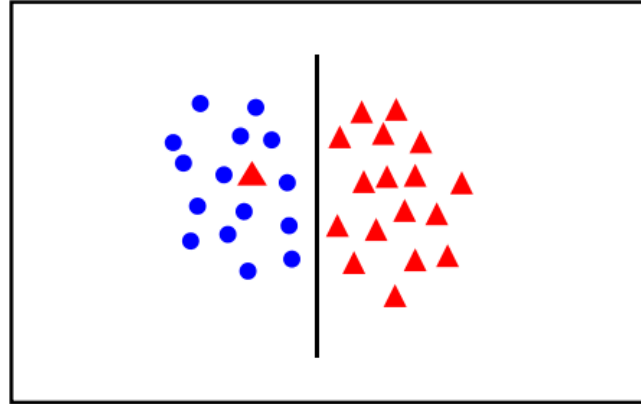
- Advantage of dual over primal form:
 - Dual form only involves $(\mathbf{x}_j^\top \mathbf{x}_k)$ - which requires the training data points!
However, many of α_i are 0 (the non support vectors).

SVMs – Dual Form

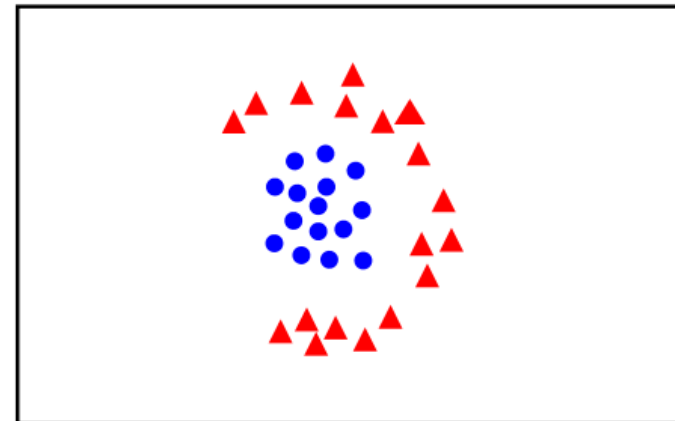


SVMs – Handling non linear data

- Introduce slack variables

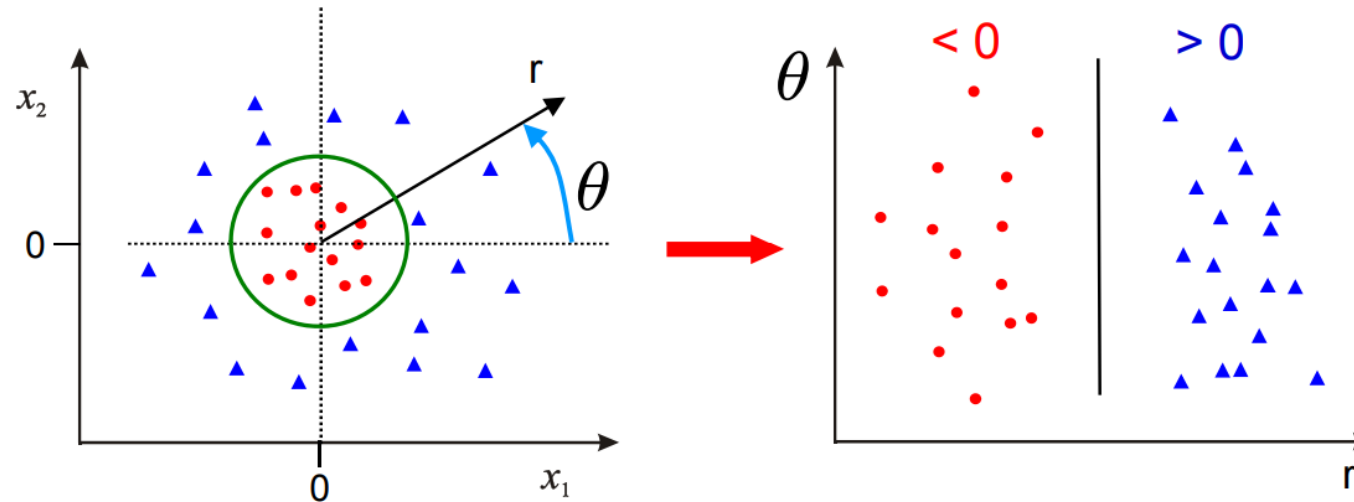


- Linear classifier not appropriate



SVMs – Solution 1

- Using polar coordinates



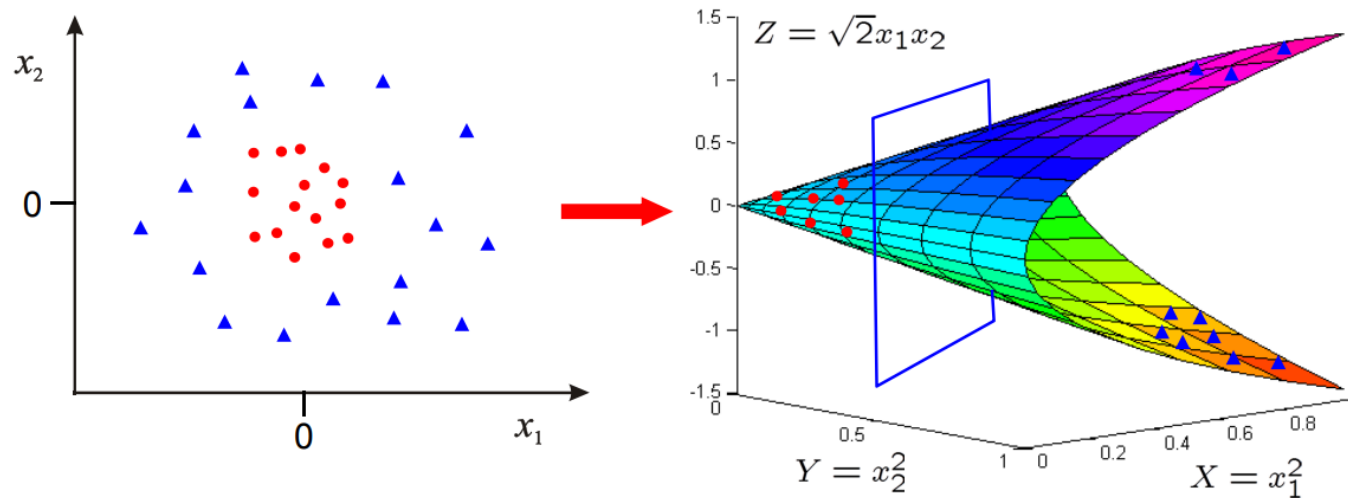
- Data is **linearly separable** in polar coordinates
- Acts non linearly in original space

$$\Phi : \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \rightarrow \begin{pmatrix} r \\ \theta \end{pmatrix} \quad \mathbb{R}^2 \rightarrow \mathbb{R}^2$$

SVMs – Solution 2

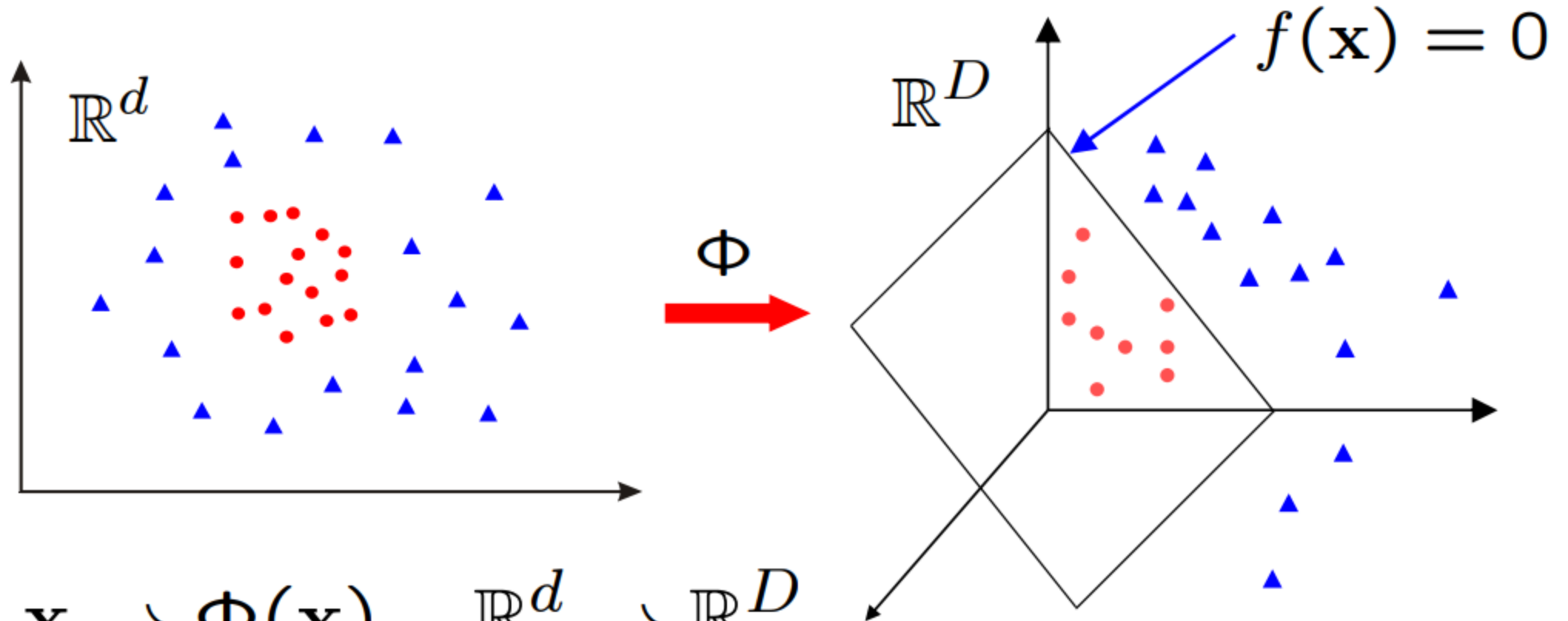
- Map data to a **higher dimension**

$$\Phi : \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \rightarrow \begin{pmatrix} x_1^2 \\ x_2^2 \\ \sqrt{2}x_1x_2 \end{pmatrix} \quad \mathbb{R}^2 \rightarrow \mathbb{R}^3$$



- Data is **linearly separable** in 3D.
- This means that the problem can still be solved by a **linear classifier**.

SVMs – Transformed Feature Space



$$\Phi : \mathbf{x} \rightarrow \Phi(\mathbf{x}) \quad \mathbb{R}^d \rightarrow \mathbb{R}^D$$

- Learn a linear classifier in w for \mathbb{R}^D
 $\Phi(\mathbf{x})$ is a **feature map**

$$f(\mathbf{x}) = \mathbf{w}^\top \Phi(\mathbf{x}) + b$$

Primal Classifier – Transformed Feature Space

- **Classifier**, with $\mathbf{w} \in \mathbb{R}^D$:

$$f(\mathbf{x}) = \mathbf{w}^\top \Phi(\mathbf{x}) + b$$

- **Learning**, for $\mathbf{w} \in \mathbb{R}^D$:

$$\min_{\mathbf{w} \in \mathbb{R}^D} \|\mathbf{w}\|^2 + C \sum_i^N \max(0, 1 - y_i f(\mathbf{x}_i))$$

- Map \mathbf{x} to $\Phi(\mathbf{x})$ where data is linearly separable
- Solve for \mathbf{w} in high dimensional space

Dual Classifier – Transformed Feature Space

• **Classifier:**

$$f(\mathbf{x}) = \sum_i^N \alpha_i y_i \mathbf{x}_i^\top \mathbf{x} + b$$

$$\rightarrow f(\mathbf{x}) = \sum_i^N \alpha_i y_i \Phi(\mathbf{x}_i)^\top \Phi(\mathbf{x}) + b$$

• **Learning:**

$$\max_{\alpha_i \geq 0} \sum_i \alpha_i - \frac{1}{2} \sum_{jk} \alpha_j \alpha_k y_j y_k \mathbf{x}_j^\top \mathbf{x}_k$$

$$\rightarrow \max_{\alpha_i \geq 0} \sum_i \alpha_i - \frac{1}{2} \sum_{jk} \alpha_j \alpha_k y_j y_k \Phi(\mathbf{x}_j)^\top \Phi(\mathbf{x}_k)$$

subject to

$$0 \leq \alpha_i \leq C \text{ for } \forall i, \text{ and } \sum_i \alpha_i y_i = 0$$

Dual Classifier – Transformed Feature Space

- Note that $\Phi(\mathbf{x})$ only occurs in pairs $\Phi(\mathbf{x}_j)^\top \Phi(\mathbf{x}_i)$
- Once the scalar products are computed, only the N dimensional vector α needs to be learnt; it is not necessary to learn in the D dimensional space, as it is for the primal
- Write $k(\mathbf{x}_j, \mathbf{x}_i) = \Phi(\mathbf{x}_j)^\top \Phi(\mathbf{x}_i)$. This is known as a **Kernel**

- **Classifier:**
$$f(\mathbf{x}) = \sum_i^N \alpha_i y_i k(\mathbf{x}_i, \mathbf{x}) + b$$

- **Learning:**
$$\max_{\alpha_i \geq 0} \sum_i \alpha_i - \frac{1}{2} \sum_{jk} \alpha_j \alpha_k y_j y_k k(\mathbf{x}_j, \mathbf{x}_k)$$

subject to
$$0 \leq \alpha_i \leq C \text{ for } \forall i, \text{ and } \sum_i \alpha_i y_i = 0$$

SVMs – Kernel Trick

$$\Phi : \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \rightarrow \begin{pmatrix} x_1^2 \\ x_2^2 \\ \sqrt{2}x_1x_2 \end{pmatrix} \quad \mathbb{R}^2 \rightarrow \mathbb{R}^3$$

$$\begin{aligned} \Phi(\mathbf{x})^\top \Phi(\mathbf{z}) &= (x_1^2, x_2^2, \sqrt{2}x_1x_2) \begin{pmatrix} z_1^2 \\ z_2^2 \\ \sqrt{2}z_1z_2 \end{pmatrix} \\ &= x_1^2z_1^2 + x_2^2z_2^2 + 2x_1x_2z_1z_2 \\ &= (x_1z_1 + x_2z_2)^2 \\ &= (\mathbf{x}^\top \mathbf{z})^2 \end{aligned}$$

- **Kernel Trick**

- Classifier can be learnt and applied without explicitly computing $\Phi(\mathbf{x})$
- All that is required is the kernel $k(\mathbf{x}, \mathbf{z}) = (\mathbf{x}^\top \mathbf{z})^2$

SVMs – Example Kernels

- **Linear** kernels $k(\mathbf{x}, \mathbf{x}') = \mathbf{x}^\top \mathbf{x}'$
- **Polynomial** kernels $k(\mathbf{x}, \mathbf{x}') = (1 + \mathbf{x}^\top \mathbf{x}')^d$ for any $d > 0$
 - Contains all polynomials terms up to degree d
- **Gaussian** kernels $k(\mathbf{x}, \mathbf{x}') = \exp(-\|\mathbf{x} - \mathbf{x}'\|^2 / 2\sigma^2)$ for $\sigma > 0$
 - Infinite dimensional feature space

SVM Classifier with Gaussian Kernel

N = size of training data

$$f(\mathbf{x}) = \sum_i^N \alpha_i y_i k(\mathbf{x}_i, \mathbf{x}) + b$$

weight (may be zero)

support vector

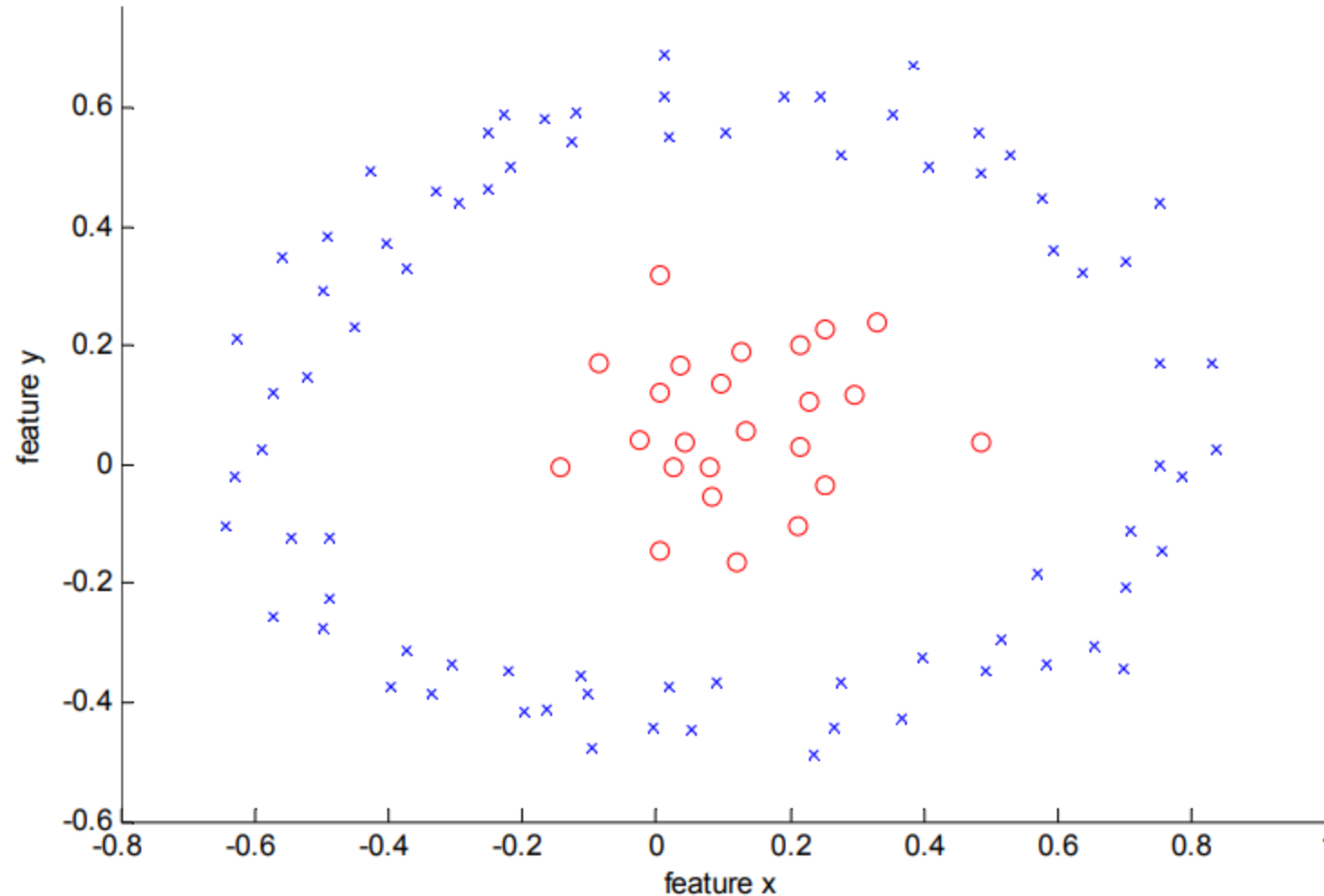
Gaussian kernel $k(\mathbf{x}, \mathbf{x}') = \exp(-\|\mathbf{x} - \mathbf{x}'\|^2 / 2\sigma^2)$

Radial Basis Function (RBF) SVM

$$f(\mathbf{x}) = \sum_i^N \alpha_i y_i \exp(-\|\mathbf{x} - \mathbf{x}_i\|^2 / 2\sigma^2) + b$$

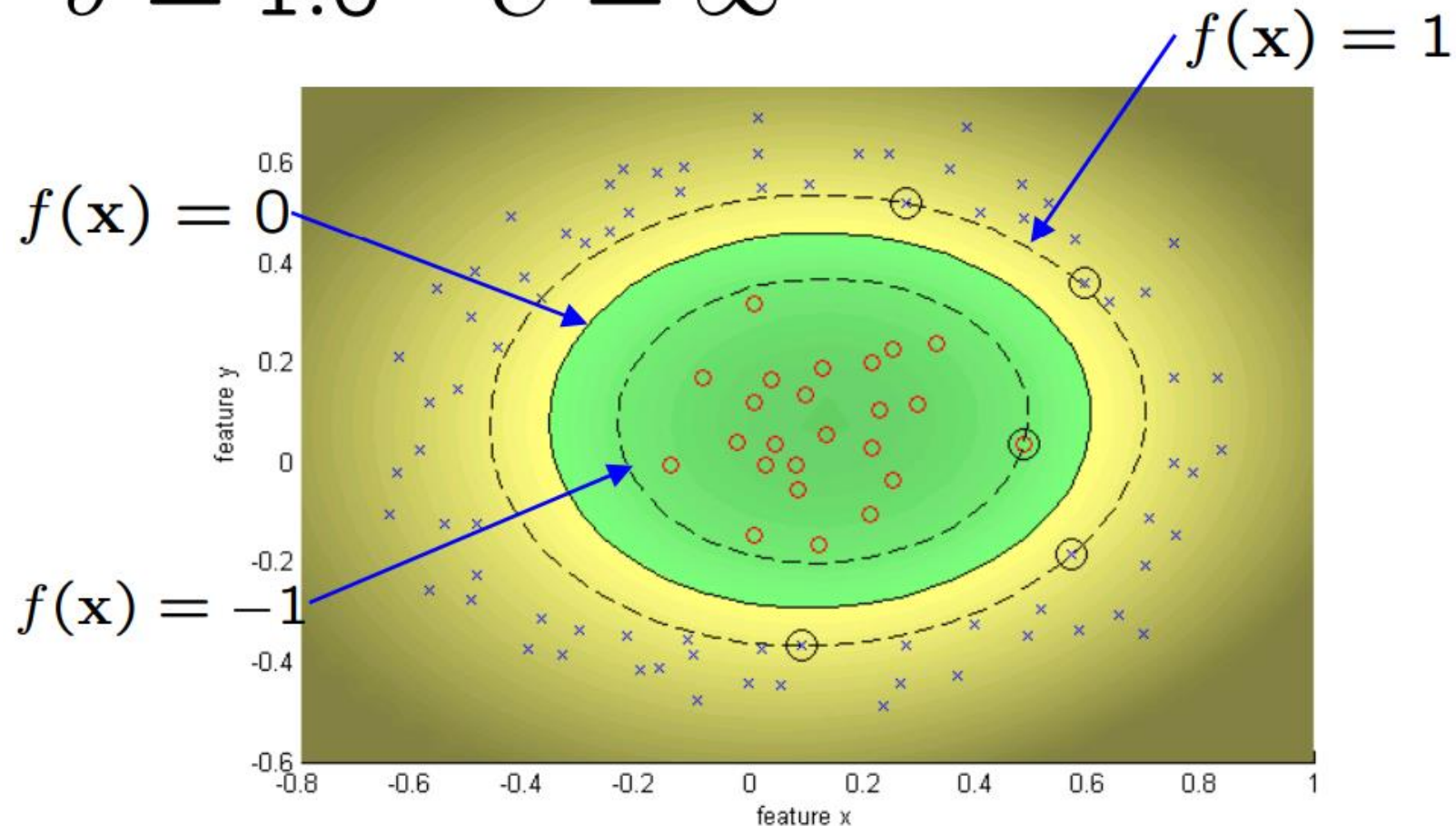
SVMs – RFB Kernel

- Data is not linearly separable in the original feature space



SVMs – RFB Kernel

$$\sigma = 1.0 \quad C = \infty$$

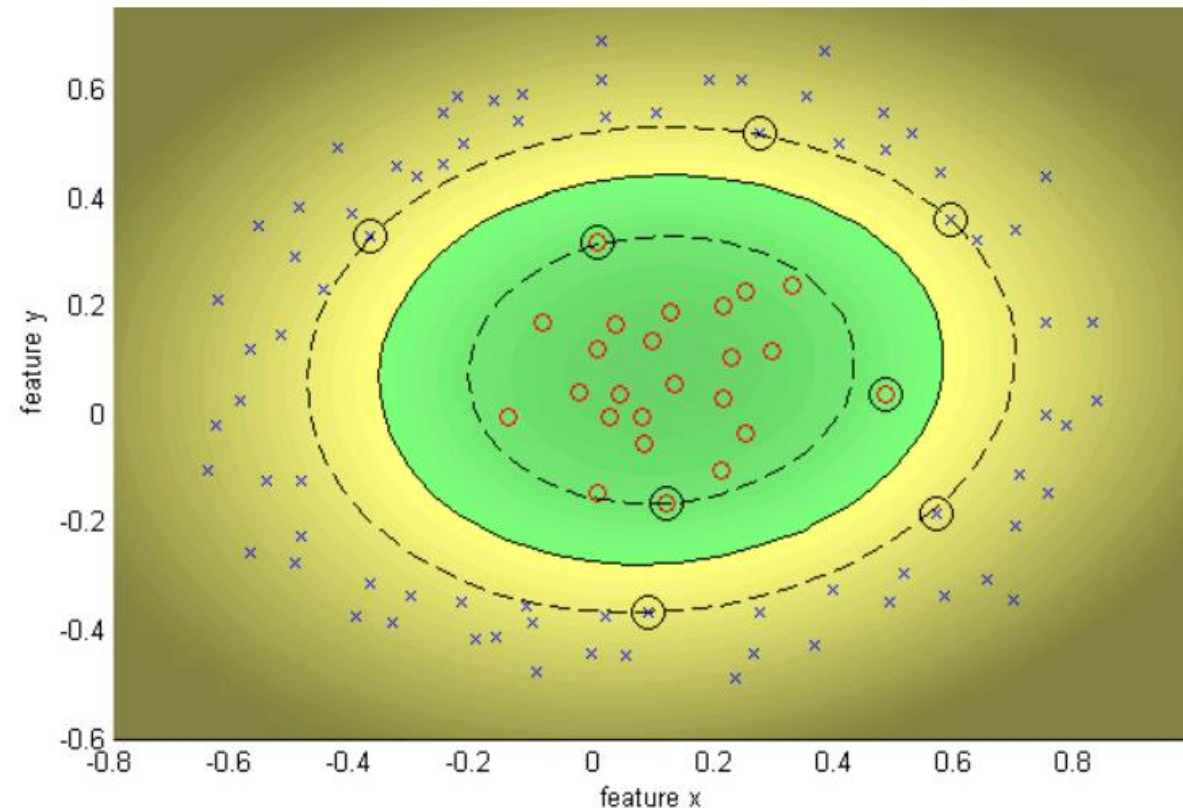


$$f(\mathbf{x}) = \sum_i^N \alpha_i y_i \exp \left(-\|\mathbf{x} - \mathbf{x}_i\|^2 / 2\sigma^2 \right) + b$$

SVMs – RFB Kernel

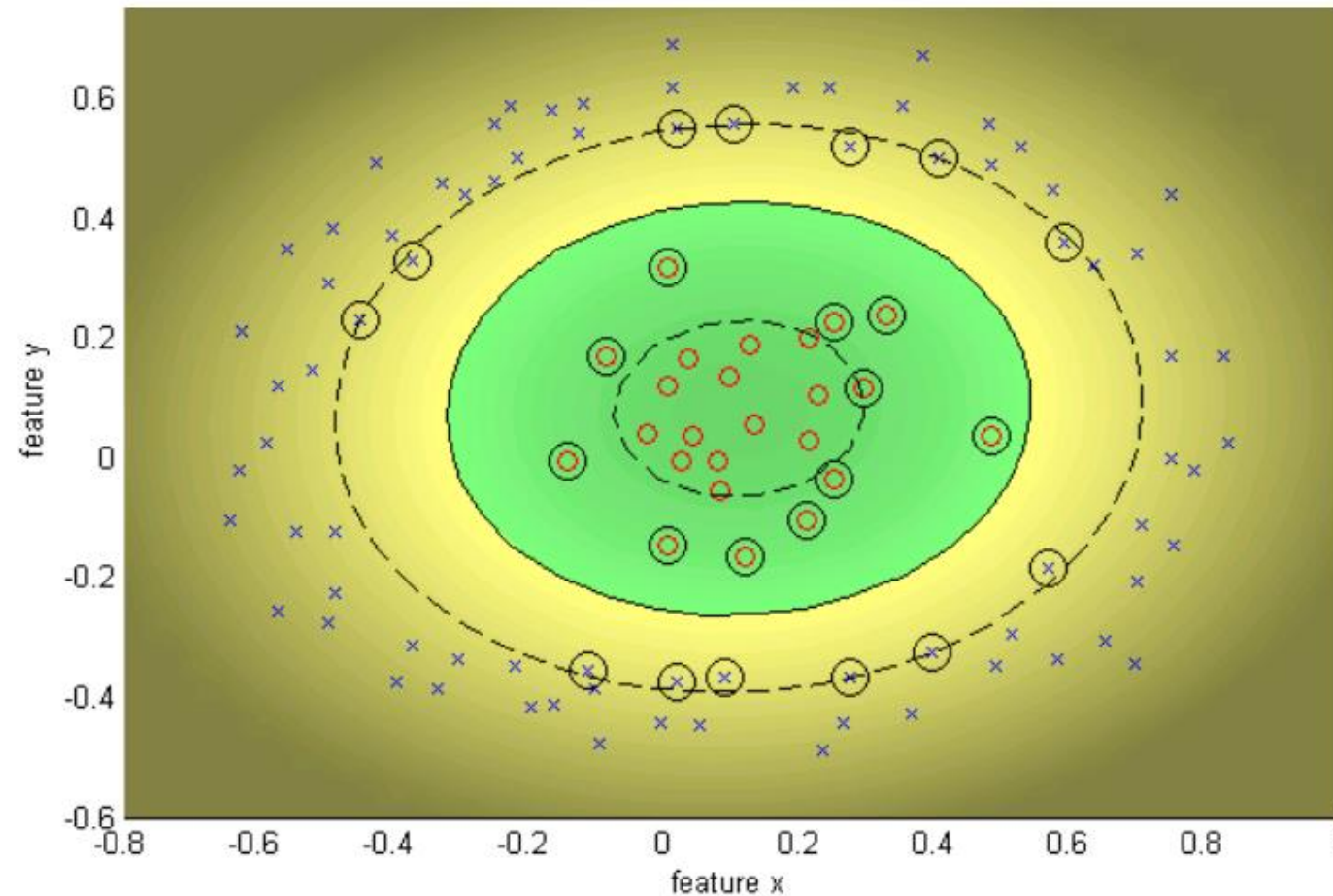
- Decrease C , gives wider (soft) margin.

$$\sigma = 1.0 \quad C = 100$$



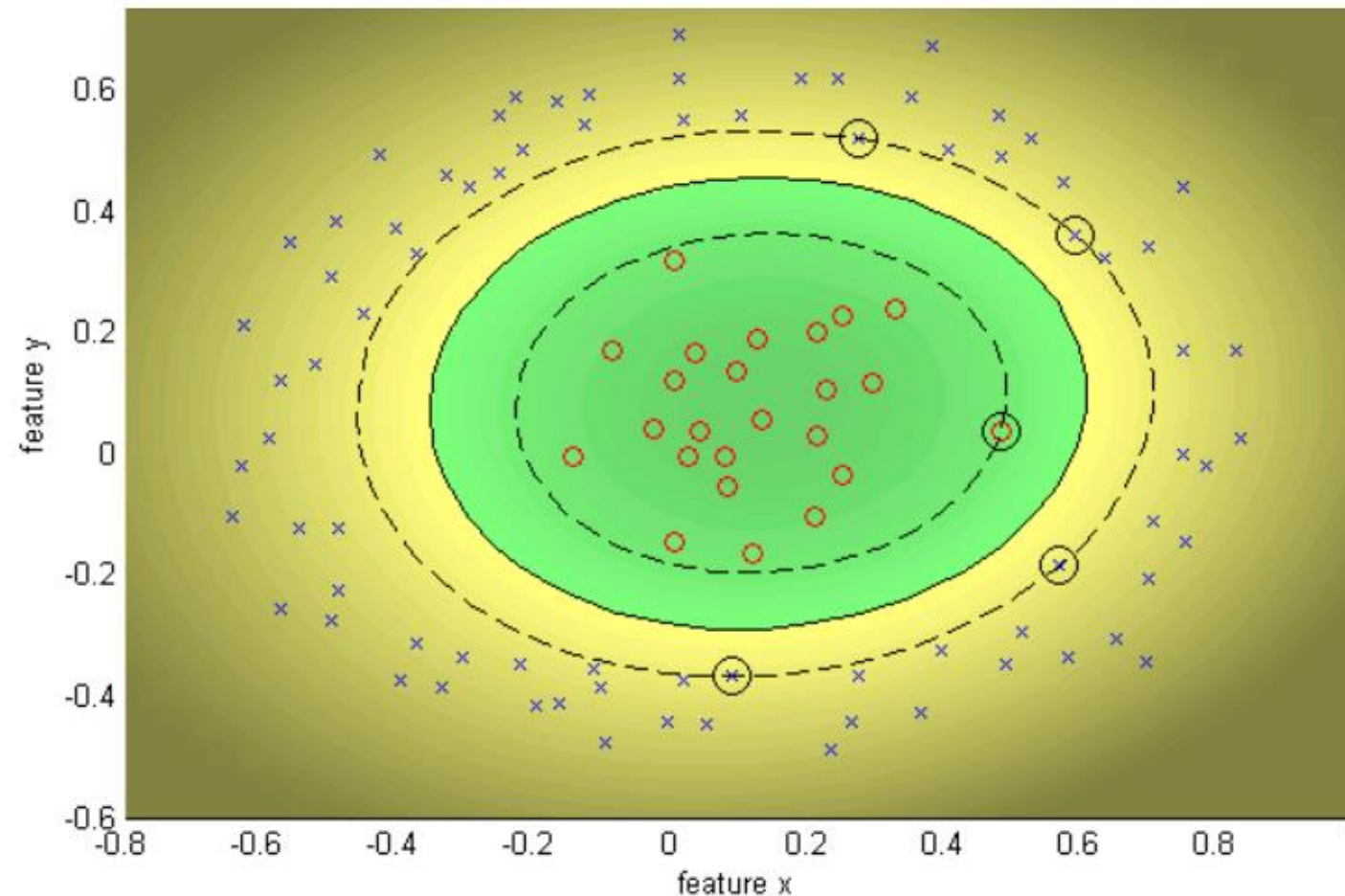
SVMs – RFB Kernel

$$\sigma = 1.0 \quad C = 10$$



SVMs – RFB Kernel

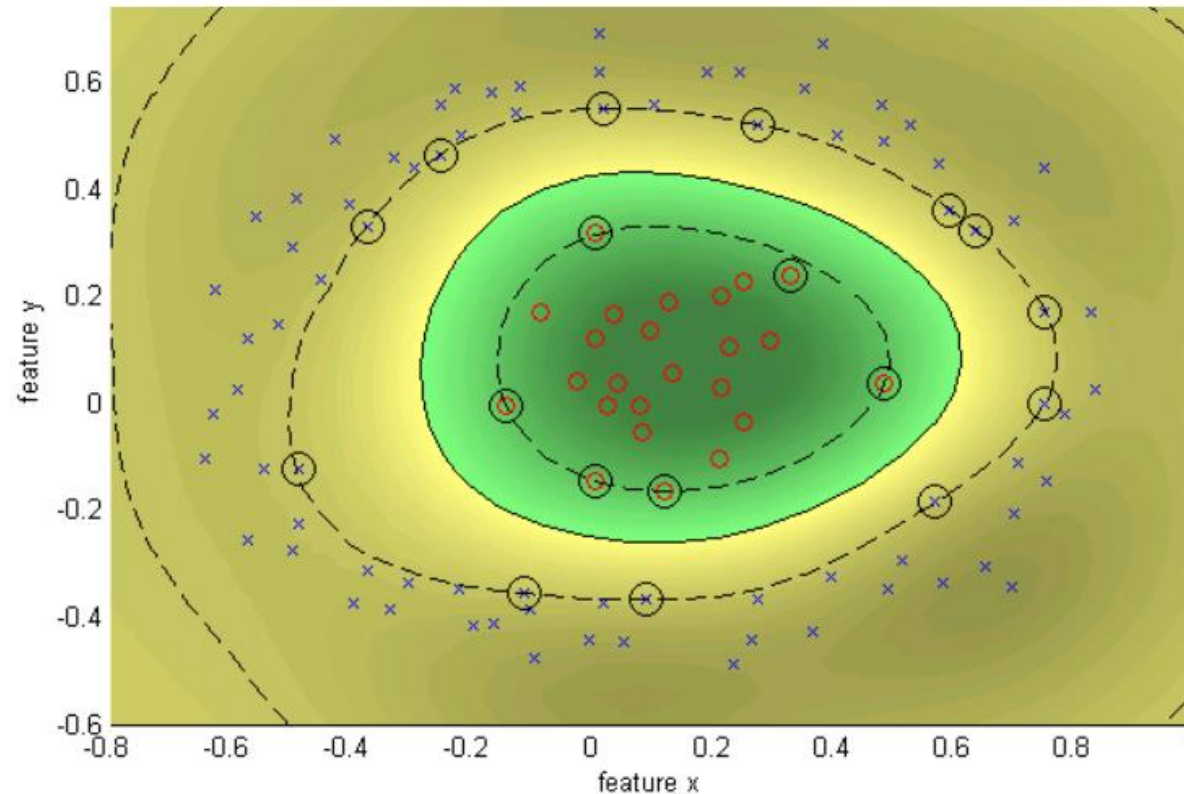
$$\sigma = 1.0 \quad C = \infty$$



SVMs – RFB Kernel

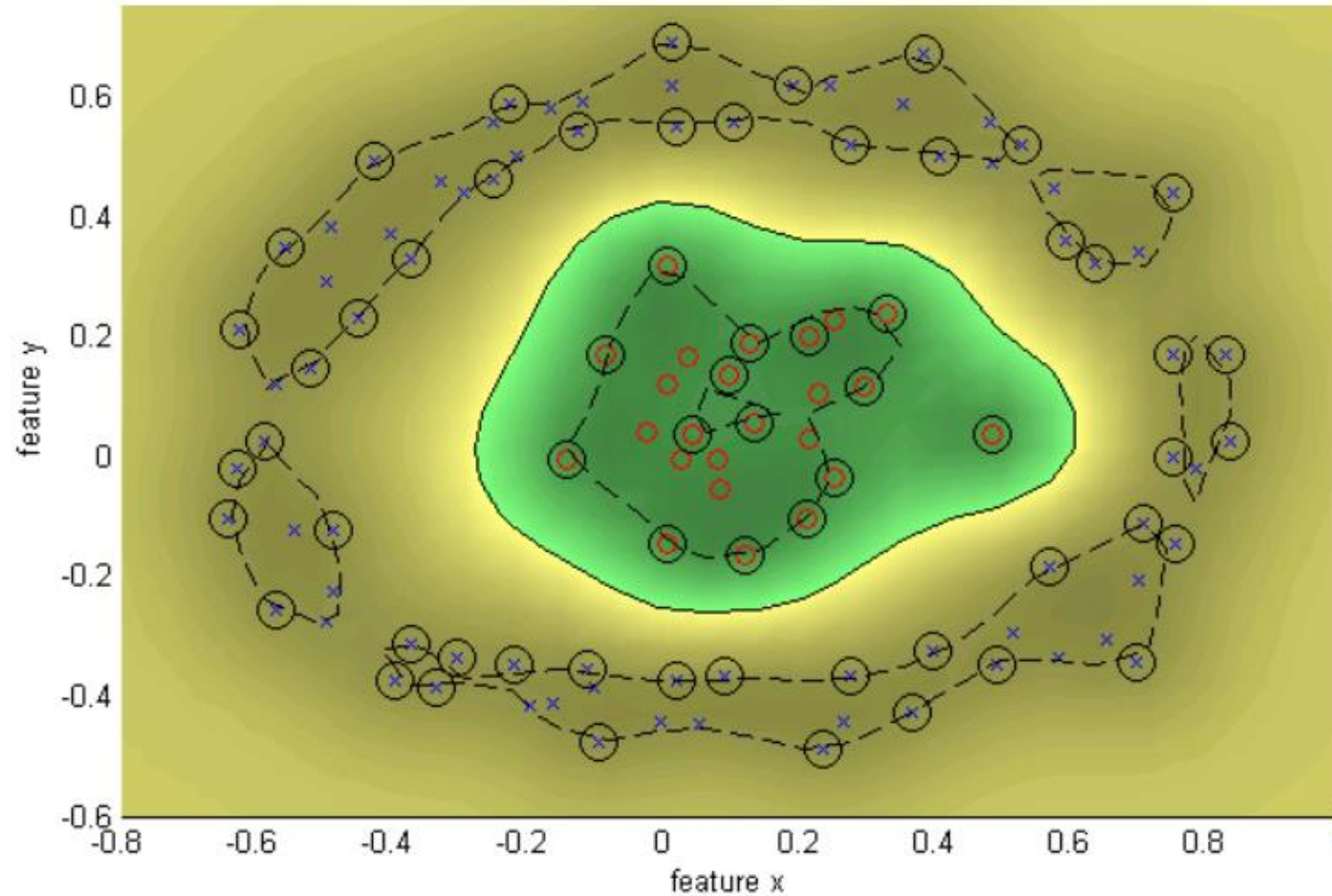
- Decrease sigma, moves towards nearest neighbour classifier

$$\sigma = 0.25 \quad C = \infty$$



SVMs – RFB Kernel

$$\sigma = 0.1 \quad C = \infty$$



Resources



- <https://www.youtube.com/watch?v=efR1C6CvhmE>
- <https://www.youtube.com/watch?v=Toet3EiSFcM>
- https://www.youtube.com/watch?v=Qc5IyLW_hns&t=1s