SUPPORT VECTOR MACHINE KERNEL METHODS

Pei-Yuan Wu
Electrical Engineering Department
National Taiwan University

OUT-LINE

- Representer Theorem
- Primal and Dual Formulations
- Kernel trick for nonlinear separable cases

SVM – REVIEW

- We have seen that in SVM we learn a linear classifier

$$f(\boldsymbol{x}) = \boldsymbol{w}^T\boldsymbol{x} + \boldsymbol{b}$$

by solving an optimization problem over $(\boldsymbol{w}, b)$:

$$\min_{\boldsymbol{w}\in\mathbb{R}^{\boldsymbol{d}}, b\in\mathbb{R}} \frac{1}{2}\|\boldsymbol{w}\|^{\boldsymbol{2}} + C\sum_{i=1}^{N}\max(0, 1 - y_i f(\boldsymbol{x}_i))$$

  - This quadratic optimization problem is known as the primal problem.

- By introducing the representation theorem, we can reformulate SVM as learning a linear classifier

$$f(\boldsymbol{x}) = \sum_{i=1}^{N} a_i(\boldsymbol{x}_i{}^T\boldsymbol{x}) + b$$

by solving an optimization problem (to be introduced later) over a_i.

  - This is know as the dual problem, and we will look at the advantages of this formulation.

REPRESENTER THEOREM

- Recall SVM Primal problem:

$$\underbrace{\frac{1}{2}\|\boldsymbol{w}\|^2}_{\text{regularization}} + \underbrace{C\sum_{i=1}^{N}\max(0, 1 - y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b))}_{\text{hinge loss}}$$

Representer Theorem on SVM: The global optimal solution of SVM takes the form $\boldsymbol{w} = \sum_{i=1}^{N} \alpha_i y_i \boldsymbol{x}_i$.

Proof: Express $\boldsymbol{w} = \boldsymbol{w}_\parallel + \boldsymbol{w}_\perp$, where $\boldsymbol{w}_\parallel \in Span(\boldsymbol{x}_1, \dots, \boldsymbol{x}_N)$, $\boldsymbol{w}_\perp$ is in the subspace orthogonal to $Span(\boldsymbol{x}_1, \dots, \boldsymbol{x}_N)$. Note that

$$\forall i \because \boldsymbol{w}_\perp{}^T \boldsymbol{x}_i = 0 \therefore \boldsymbol{w}^T\boldsymbol{x}_i = \boldsymbol{w}_\parallel{}^T\boldsymbol{x}_i$$

$$\because \boldsymbol{w}_\perp{}^T \boldsymbol{w}_\parallel = 0 \therefore \|\boldsymbol{w}\|^2 = \left\|\boldsymbol{w}_\parallel\right\|^2 + \|\boldsymbol{w}_\perp\|^2$$

In other words, $\boldsymbol{w}_\perp$ does not influence hinge loss, but may increase regularization loss. So if $(\boldsymbol{w}_\parallel + \boldsymbol{w}_\perp, b)$ is optimal, then $(\boldsymbol{w}_\parallel, b)$ must be optimal.

In SVM, it suffices to assume $w = \sum_{i=1}^{N} \alpha_i y_i x_i$

REPRESENTER THEOREM

- Substitute $\boldsymbol{w} = \sum_{i=1}^{N} \alpha_i y_i \boldsymbol{x}_i$ into $f(x) = \boldsymbol{w}^T \boldsymbol{x} + b$ and $\|\boldsymbol{w}\|^2$, we get

$$f(x) = \left(\sum_{i=1}^{N} \alpha_i y_i \boldsymbol{x}_i\right)^T \boldsymbol{x} + b = \sum_{i=1}^{N} \alpha_i y_i (\boldsymbol{x}_i{}^T \boldsymbol{x}) + b$$

$$\|\boldsymbol{w}\|^2 = \left(\sum_{i=1}^{N} \alpha_i y_i \boldsymbol{x}_i\right)^T \left(\sum_{j=1}^{N} \alpha_j y_j \boldsymbol{x}_j\right) = \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j (\boldsymbol{x}_i{}^T \boldsymbol{x}_j)$$

Hence, an equivalent optimization problem is over α_i

Primal problem:

$$\min_{\boldsymbol{w} \in \mathbb{R}^d, b \in \mathbb{R}, \xi_1, \ldots, \xi_N \geq 0} \frac{1}{2} \|\boldsymbol{w}\|^2 + C \sum_{i=1}^{N} \xi_i$$

subject to $y_i(\boldsymbol{w}^T \boldsymbol{x}_i + b) \geq 1 - \xi_i, \forall i$

Optimization problem over α_i

$$\min_{\boldsymbol{\alpha} \in \mathbb{R}^N, b \in \mathbb{R}, \xi_1, \ldots, \xi_N \geq 0} \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j (\boldsymbol{x}_i{}^T \boldsymbol{x}_j) + C \sum_{i=1}^{N} \xi_i$$

Subject to $y_i\left(\sum_{j=1}^{N} \alpha_j y_j (\boldsymbol{x}_j{}^T \boldsymbol{x}_i) + b\right) \geq 1 - \xi_i, \forall i$

2N variables

(3~6 hour lectures)

and ***A FEW*** more steps are required to complete the derivation (with N variables)…

SVM PRIMAL AND DUAL PROBLEMS

N is number of training points, and d is dimension of feature vector x.

Primal problem: for $\boldsymbol{w} \in \mathbb{R}^{\boldsymbol{d}}, b \in \mathbb{R}$

$$\min_{\boldsymbol{w}\in\mathbb{R}^{\boldsymbol{d}}, b\in\mathbb{R}} \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{N} \max(0, 1 - y_i f(\boldsymbol{x}_i))$$

Dual problem: for $\boldsymbol{\alpha} \in \mathbb{R}^{\boldsymbol{N}}$ (Formal proof granted after introduction of duality theorem)

$$\max_{\alpha_1,\ldots,\alpha_N\geq 0} \sum_{i=1}^{N} \alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i\alpha_j y_i y_j (\boldsymbol{x}_i{}^T\boldsymbol{x}_j)$$

Subject to $0 \leq \alpha_i \leq C, \forall i$, and $\sum_{i=1}^{N} \alpha_i y_i = 0$

KKT Condition:

$y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) > 1 \Rightarrow \xi_i = 0, \alpha_i = 0$

$y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) < 1 \Rightarrow \xi_i > 0, \alpha_i = C$

- Need to learn d parameters for primal, and N parameters for dual
- If $N \ll d$ then more efficient to solve for α than w. (d can even be infinite! See Gaussian-RBF SVM to be introduced later)
- Dual form only involves $\boldsymbol{x}_i{}^T\boldsymbol{x}_j$. We will return to why this is an advantage when we look at kernels.

PRIMAL AND DUAL FORMULATIONS

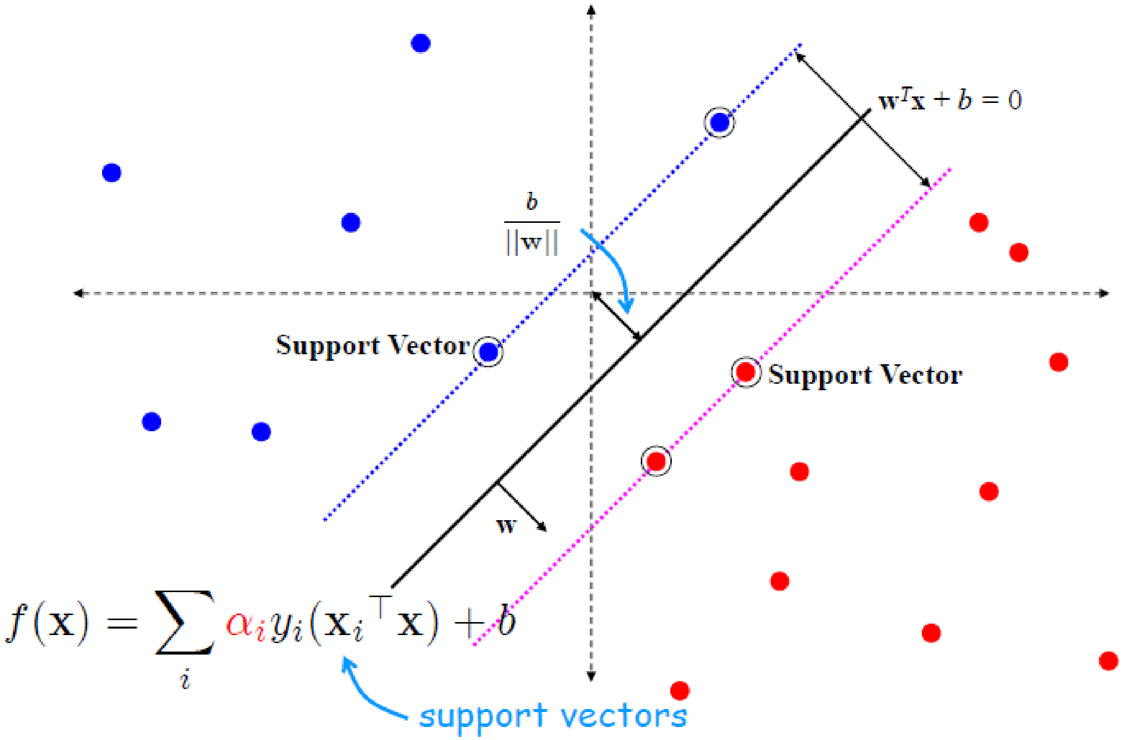
- Primal version of classifier:

$$f(x) = \boldsymbol{w}^T\boldsymbol{x} + b$$

- Dual version of classifier:

$$f(x) = \sum_{i=1}^{N} \alpha_i y_i (\boldsymbol{x}_i{}^T\boldsymbol{x}) + b$$

- At first sight the dual form appears to have the disadvantage of a K-NN classifier — it requires the training data points $\boldsymbol{x}_i$. However, many of the α_i's are zero. The ones that are non-zero define the support vectors $\boldsymbol{x}_i$.

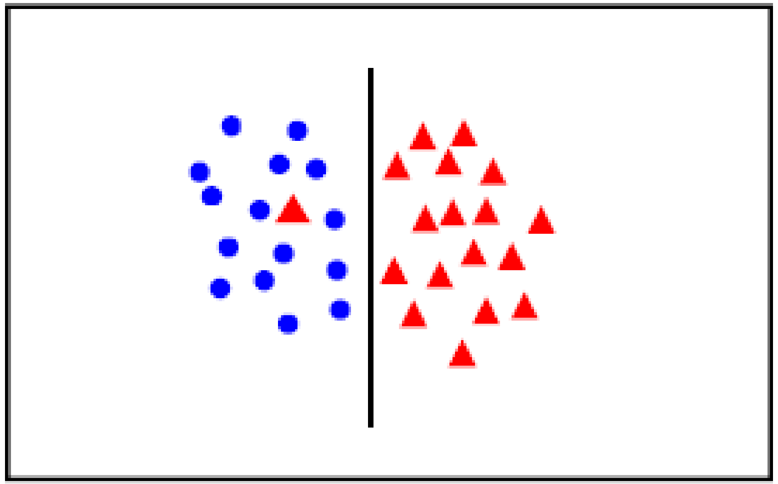


KKT Condition:

$y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) > 1 \Rightarrow \xi_i = 0, \alpha_i = 0$

$y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) < 1 \Rightarrow \xi_i > 0, \alpha_i = C$

HANDLING DATA THAT IS NOT LINEARLY SEPARABLE

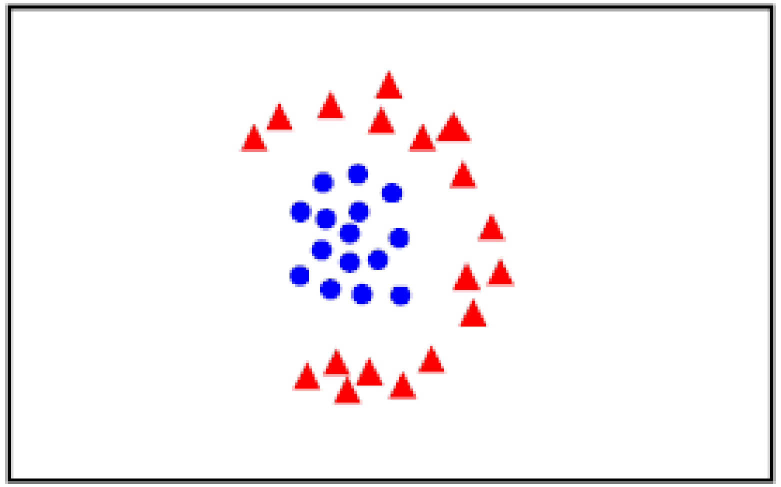


- introduce slack variables

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

subject to

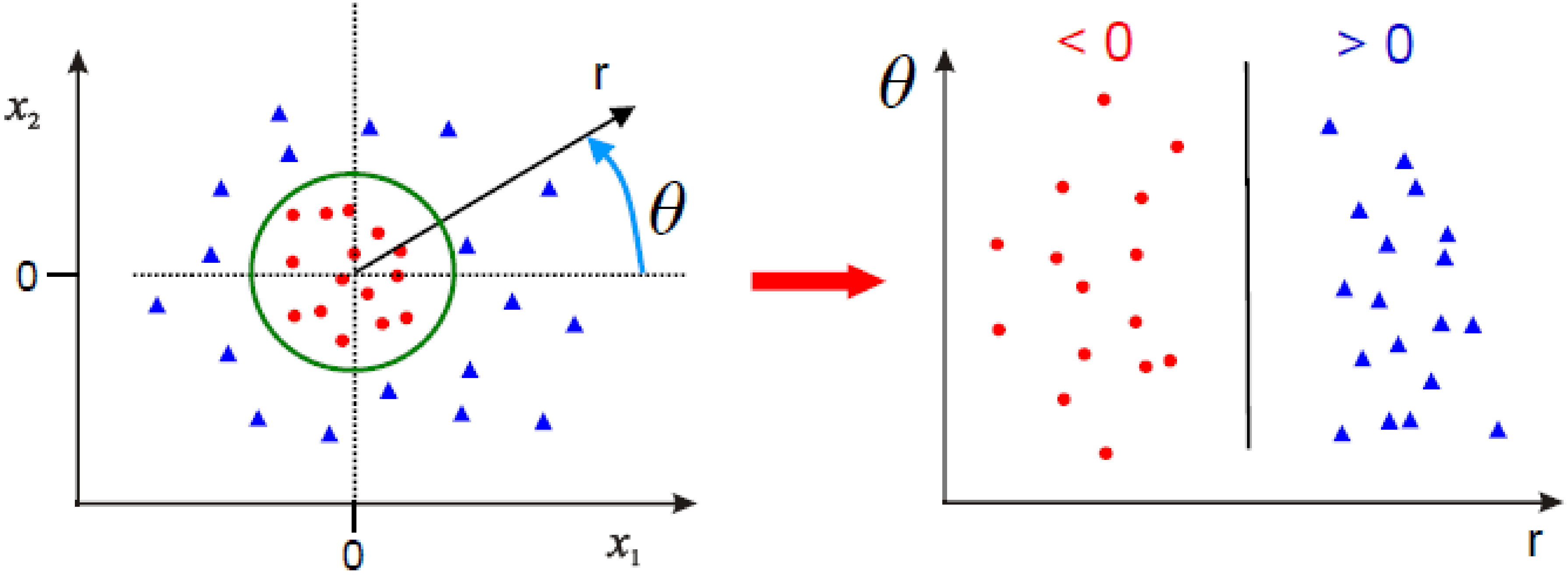
$$y_i\left(\mathbf{w}^\top \mathbf{x}_i + b\right) \geq 1 - \xi_i \text{ for } i = 1 \ldots N$$



- linear classifier not appropriate

??

SOLUTION 1: USE POLAR COORDINATES

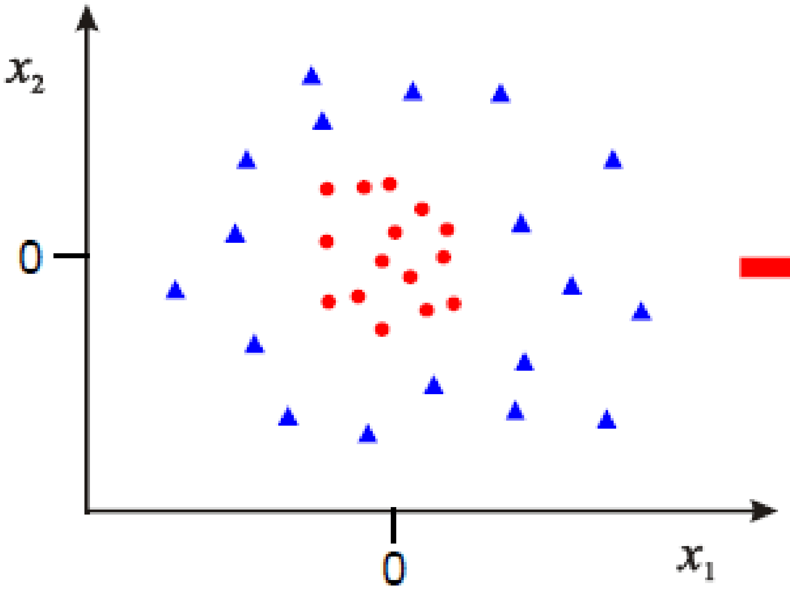


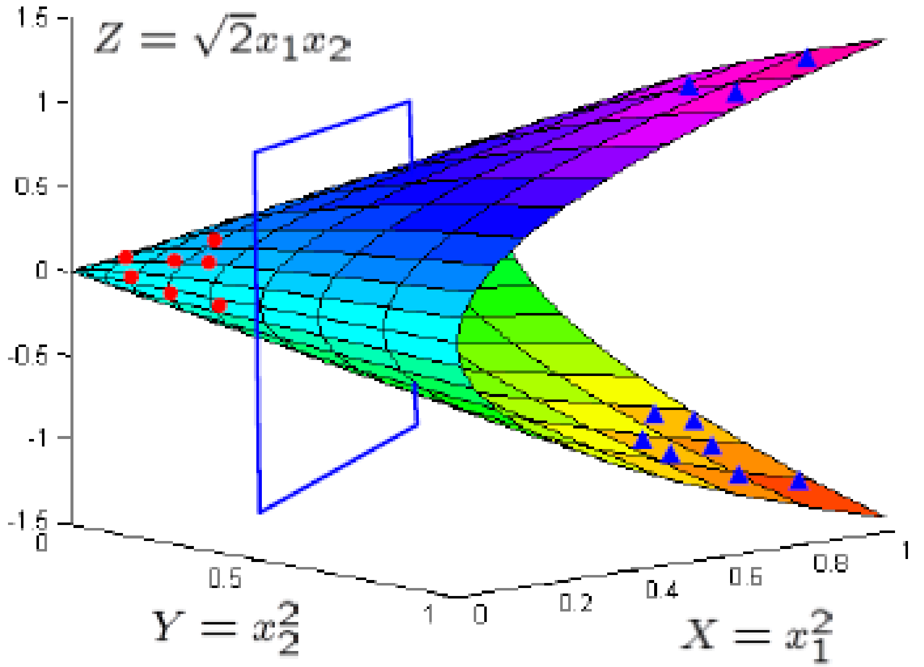
- Data is linearly separable in polar coordinates
- Acts non-linearly in original space

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

SOLUTION 2: MAP DATA TO HIGHER DIMENSION

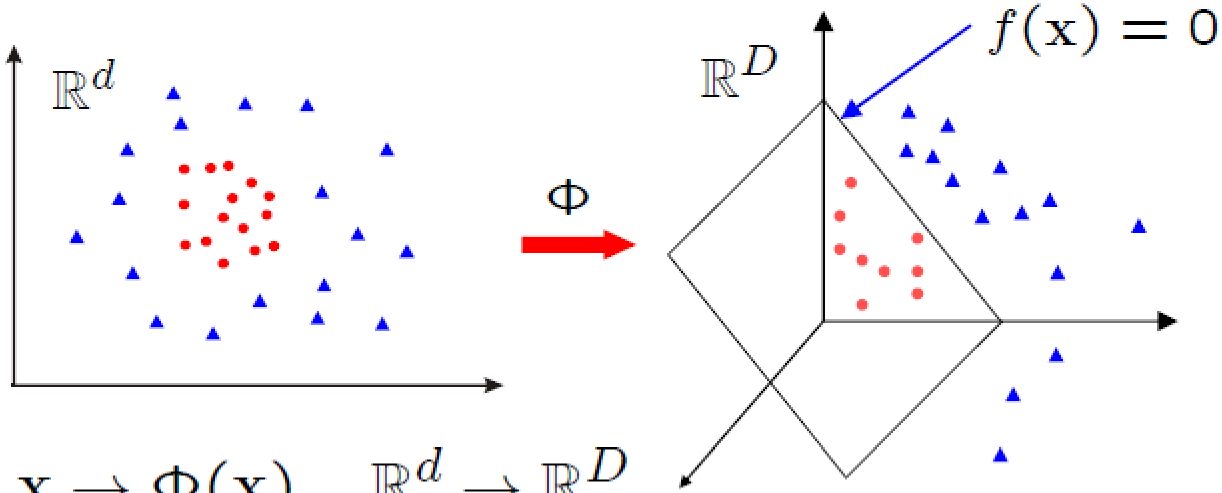
$$\Phi : \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \rightarrow \begin{pmatrix} x_1^2 \\ x_2^2 \\ \sqrt{2}x_1x_2 \end{pmatrix} \qquad \mathbb{R}^2 \rightarrow \mathbb{R}^3$$





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

SVM CLASSIFIERS IN A TRANSFORMED FEATURE SPACE



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

Learn classifier linear in $\mathbf{w}$ for $\mathbb{R}^D$:

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

$\Phi(\mathbf{x})$ is a feature map

PRIMAL CLASSIFIER IN TRANSFORMED FEATURE SPACE

Classifier, with $\boldsymbol{w} \in \mathbb{R}^D$

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

Learning, for $\boldsymbol{w} \in \mathbb{R}^D$

$$\min_{\boldsymbol{w} \in \mathbb{R}^{\boldsymbol{D}}, b \in \mathbb{R}} \frac{1}{2} \|\boldsymbol{w}\|^2 + C \sum_{i=1}^{N} \max(0, 1 - y_i f(\boldsymbol{x}_i))$$

- Simply map x to $\Phi(\mathrm{x})$ where data is separable
- Solve for w in high dimensional space $\mathbb{R}^D$
- If $D \gg d$ then there are many more parameters to learn for w. Can this be avoided?

DUAL CLASSIFIER IN TRANSFORMED FEATURE SPACE

- Classifier:

$$f(x) = \sum_{i=1}^{N} \alpha_i y_i (\boldsymbol{x}_i{}^T \boldsymbol{x}) + b$$

$$f(x) = \sum_{i=1}^{N} \alpha_i y_i \Phi(\boldsymbol{x}_i)^T \Phi(\boldsymbol{x}) + b$$

- Learning:

$$\max_{\alpha_1,\ldots,\alpha_N \geq 0} \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_i (\boldsymbol{x}_i{}^T \boldsymbol{x}_j)$$

$$\max_{\alpha_1,\ldots,\alpha_N \geq 0} \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_i \Phi(\boldsymbol{x}_i)^T \Phi(\boldsymbol{x}_j)$$

Subject to $0 \leq \alpha_i \leq C, \forall i$, and $\sum_{i=1}^{N} \alpha_i y_i = 0$

- $\mathrm{k}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \Phi(\boldsymbol{x}_i)^T \Phi(\boldsymbol{x}_j)$ is called **kernel** function.
- Note that $\Phi(\mathrm{x})$ only occurs in pairs $\Phi(\boldsymbol{x}_i)^T \Phi(\boldsymbol{x}_j)$
  - ➢ Once the scalar products are computed, only the N dimensional vector α needs to be learnt.
  - ➢ No need to learn in the D dimensional space, as it is for the primal.

KERNEL SVM

- Classifier:

$$f(x) = \sum_{i=1}^{N} \alpha_i y_i \mathrm{k}(\boldsymbol{x}_i, \boldsymbol{x}) + b$$

- Learning:

$$\max_{\alpha_1,\dots,\alpha_N \geq 0} \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j \mathrm{k}(\boldsymbol{x}_i, \boldsymbol{x}_j)$$

Subject to $0 \leq \alpha_i \leq C, \forall i$, and $\sum_{i=1}^{N} \alpha_i y_i = 0$

Kernel Trick

- Classifier can be learnt and applied without explicitly computing $\Phi(\mathrm{x})$
- All that is required is the kernel $\mathrm{k}(\boldsymbol{x}, \boldsymbol{x}')$.
- Complexity of learning depends on N (typically $O(TN^2)$) but not on D.

KERNEL AND SPECIAL TRANSFORMATIONS

Spectral Transform **Kernel Function**

$$\Phi: \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \mapsto \begin{bmatrix} \sqrt{2}x_1x_2 \\ x_1^2 \\ x_2^2 \end{bmatrix} \Rightarrow \Phi(\boldsymbol{x})^T\Phi(\boldsymbol{z}) = \begin{bmatrix} \sqrt{2}x_1x_2 \\ x_1^2 \\ x_2^2 \end{bmatrix}^T \begin{bmatrix} \sqrt{2}z_1z_2 \\ z_1^2 \\ z_2^2 \end{bmatrix}$$

$$= 2x_1z_1x_2z_2 + x_1^2z_1^2 + x_2^2z_2^2$$

$$= (x_1z_1 + x_2z_2)^2$$

$$\Phi: \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \mapsto \begin{bmatrix} 1 \\ \sqrt{2}x_1 \\ \sqrt{2}x_2 \\ \sqrt{2}x_1x_2 \\ x_1^2 \\ x_2^2 \end{bmatrix} \Rightarrow \Phi(\boldsymbol{x})^T\Phi(\boldsymbol{z}) = \begin{bmatrix} 1 \\ \sqrt{2}x_1 \\ \sqrt{2}x_2 \\ \sqrt{2}x_1x_2 \\ x_1^2 \\ x_2^2 \end{bmatrix}^T \begin{bmatrix} 1 \\ \sqrt{2}z_1 \\ \sqrt{2}z_2 \\ \sqrt{2}z_1z_2 \\ z_1^2 \\ z_2^2 \end{bmatrix}$$

$$= 1 + 2x_1z_1 + 2x_2z_2 + 2x_1z_1x_2z_2 + x_1^2z_1^2 + x_2^2z_2^2$$

$$= (1 + x_1z_1 + x_2z_2)^2$$

KERNEL EXAMPLES

Feature map: $\Phi(\boldsymbol{x}) = \begin{bmatrix} \phi_1(\boldsymbol{x}) \\ \phi_2(\boldsymbol{x}) \\ \vdots \\ \phi_D(\boldsymbol{x}) \end{bmatrix}$

Let $\boldsymbol{x}, \boldsymbol{x}' \in \mathbb{R}^d$, where we denote $\boldsymbol{x} = (x_1, \dots, x_d)$.

- Linear kernels: $\mathrm{k}(\boldsymbol{x}, \boldsymbol{x}') = \boldsymbol{x}^T \boldsymbol{x}'$
- Polynomial kernels: $\mathrm{k}(\boldsymbol{x}, \boldsymbol{x}') = (1 + \boldsymbol{x}^T \boldsymbol{x}')^m$, for any $m \in \mathbb{N}$.
  - ➢ $\Phi(\boldsymbol{x})$ contains all polynomials up to degree m.

    $$\phi_j(\boldsymbol{x}) \propto {x_1}^{n_1} {x_2}^{n_2} \dots {x_d}^{n_d}$$

    where $n_1 + \cdots + n_d \le m, n_1, \dots, n_d \in \mathbb{N} \cup \{0\}$
  - ➢ Feature space dimension $\mathrm{D} = \binom{d+m}{d}$
- Gaussian kernels: $\mathrm{k}(\boldsymbol{x}, \boldsymbol{x}') = exp\left(-\frac{\|\boldsymbol{x}-\boldsymbol{x}\prime\|^2}{2\sigma^2}\right)$ for $\sigma > 0$
  - ➢ $\Phi(\boldsymbol{x})$ contains all functions of the form

    $$\phi_j(\boldsymbol{x}) \propto \left(\frac{x_1}{\sigma}\right)^{m_1} \left(\frac{x_2}{\sigma}\right)^{m_2} \dots \left(\frac{x_d}{\sigma}\right)^{m_d} e^{-\frac{\|\boldsymbol{x}\|^2}{2\sigma^2}}$$

    where $n_1, \dots, n_d \in \mathbb{N} \cup \{0\}$
  - ➢ Feature space dimension $\mathrm{D} = \infty$

SPECTRAL TRANSFORMATION OF GAUSSIAN KERNEL

For simplicity, consider $d = 1$. Then

$$exp\left(-\frac{(x-z)^2}{2\sigma^2}\right) = exp\left(-\frac{x^2 - 2xz + z^2}{2\sigma^2}\right)$$

$$= exp\left(-\frac{x^2}{2\sigma^2}\right) exp\left(\frac{xz}{\sigma^2}\right) exp\left(-\frac{z^2}{2\sigma^2}\right)$$

$$= exp\left(-\frac{x^2}{2\sigma^2}\right)\left(1 + \frac{xz}{\sigma^2} + \cdots + \frac{1}{n!}\left(\frac{xz}{\sigma^2}\right)^n + \cdots\right) exp\left(-\frac{z^2}{2\sigma^2}\right)$$

$$= {\color{red} exp\left(-\frac{x^2}{2\sigma^2}\right)\begin{bmatrix} 1 \\ x/\sigma \\ \vdots \\ \left(x/\sqrt{n!}\,\sigma\right)^n \\ \vdots \end{bmatrix}^T} {\color{blue} \begin{bmatrix} 1 \\ z/\sigma \\ \vdots \\ \left(z/\sqrt{n!}\,\sigma\right)^n \\ \vdots \end{bmatrix} exp\left(-\frac{z^2}{2\sigma^2}\right)}$$

$$= {\color{red}\Phi(x)}^T {\color{blue}\Phi(z)}$$

Feature map:

$$\Phi(x) = exp\left(-\frac{x^2}{2\sigma^2}\right)\begin{bmatrix} 1 \\ x/\sigma \\ \vdots \\ \left(x/\sqrt{n!}\,\sigma\right)^n \\ \vdots \end{bmatrix}$$

GAUSSIAN RADIAL BASIS FUNCTION (RBF) SVM

Classifier: weight (may be zero) support vector

$$f(x) = \sum_{i=1}^{N} \alpha_i y_i \mathrm{k}(\boldsymbol{x}_i, \boldsymbol{x}) + b$$

Learning:

$$\max_{\alpha_1,\ldots,\alpha_N \geq 0} \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j \mathrm{k}(\boldsymbol{x}_i, \boldsymbol{x}_j)$$

Subject to $0 \leq \alpha_i \leq C, \forall i$, and $\sum_{i=1}^{N} \alpha_i y_i = 0$

Gaussian kernel: $\mathrm{k}(\boldsymbol{x}, \boldsymbol{x}') = exp\left(-\frac{\|\boldsymbol{x}-\boldsymbol{x}\prime\|^2}{2\sigma^2}\right)$

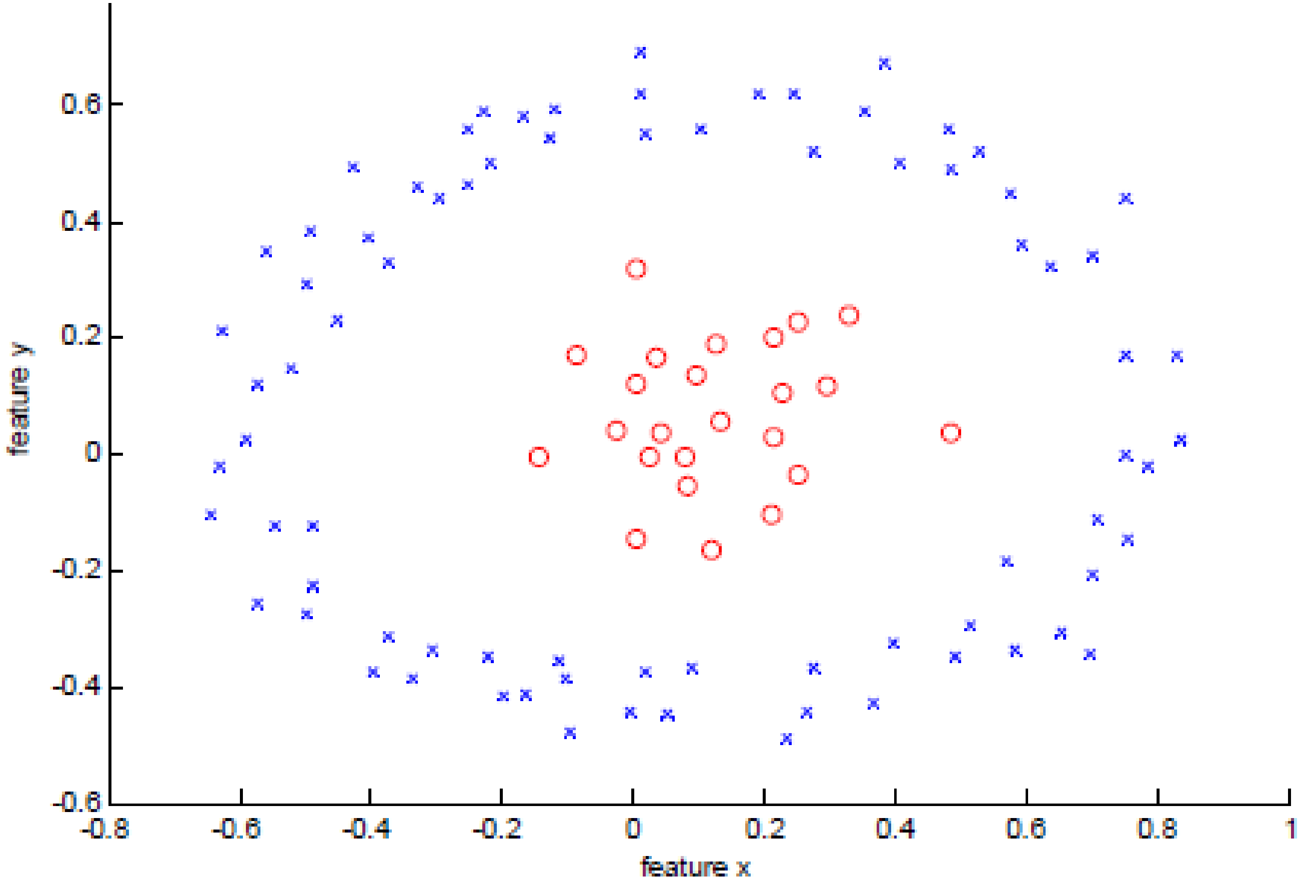
Gaussian Radial Basis Function SVM

$$f(\boldsymbol{x}) = \sum_{i=1}^{N} \alpha_i y_i exp\left(-\frac{\|\boldsymbol{x} - \boldsymbol{x}_i\|^2}{2\sigma^2}\right) + b$$

Radial basis function kernel:

$\mathrm{k}(\boldsymbol{x}, \boldsymbol{z})$ only depends on $\|\boldsymbol{x} - \boldsymbol{z}\|$

RBF KERNEL SVM EXAMPLE



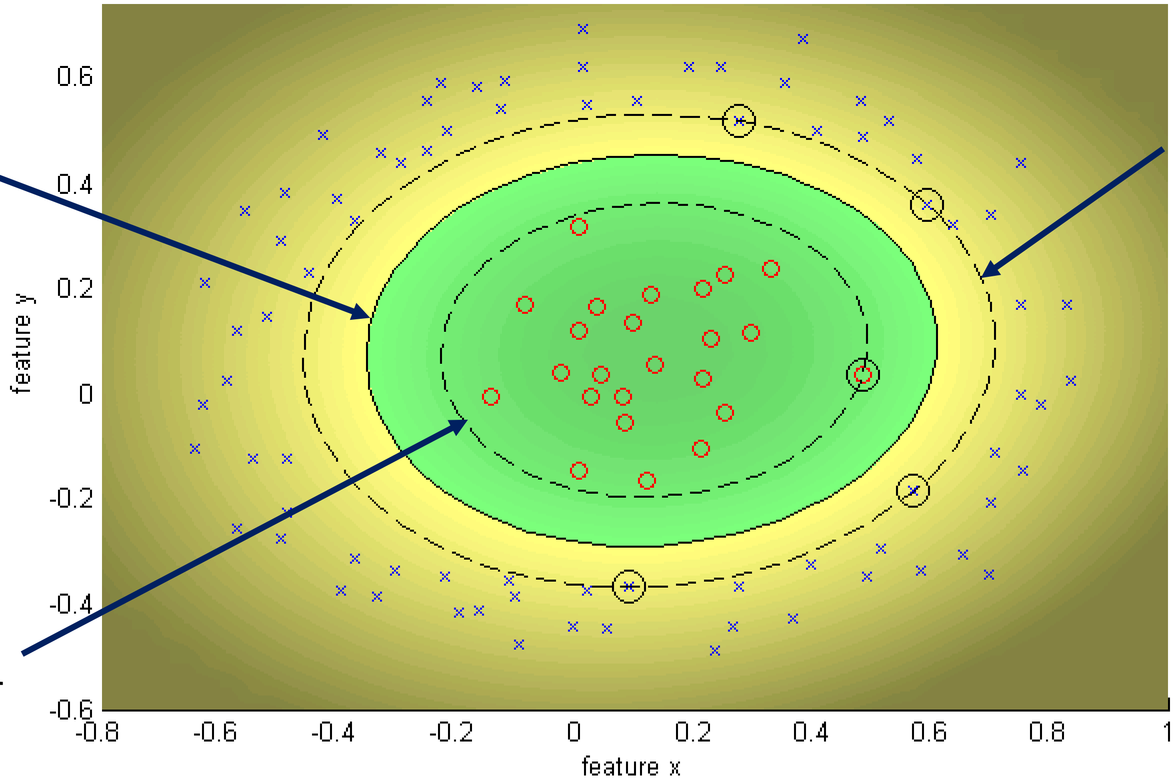
data is not linearly separable in original feature space

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

$f(x) = 0$

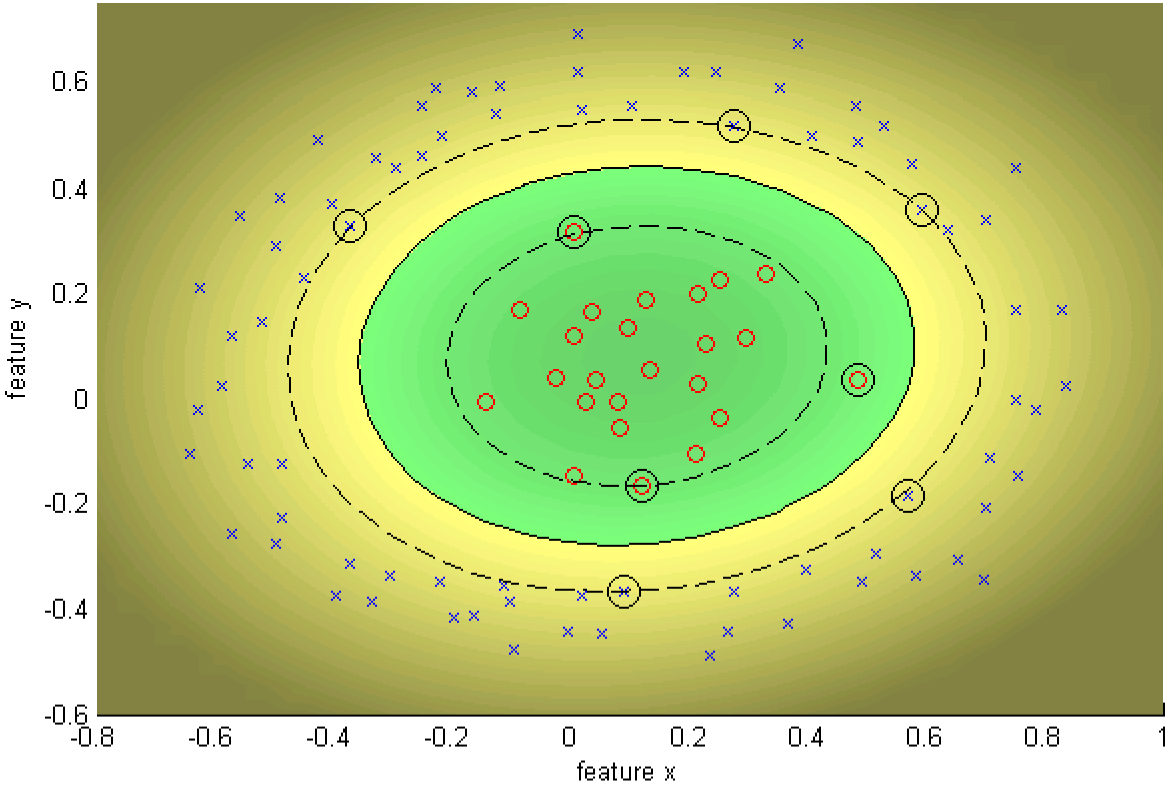
$f(x) = 1$

$f(x) = -1$



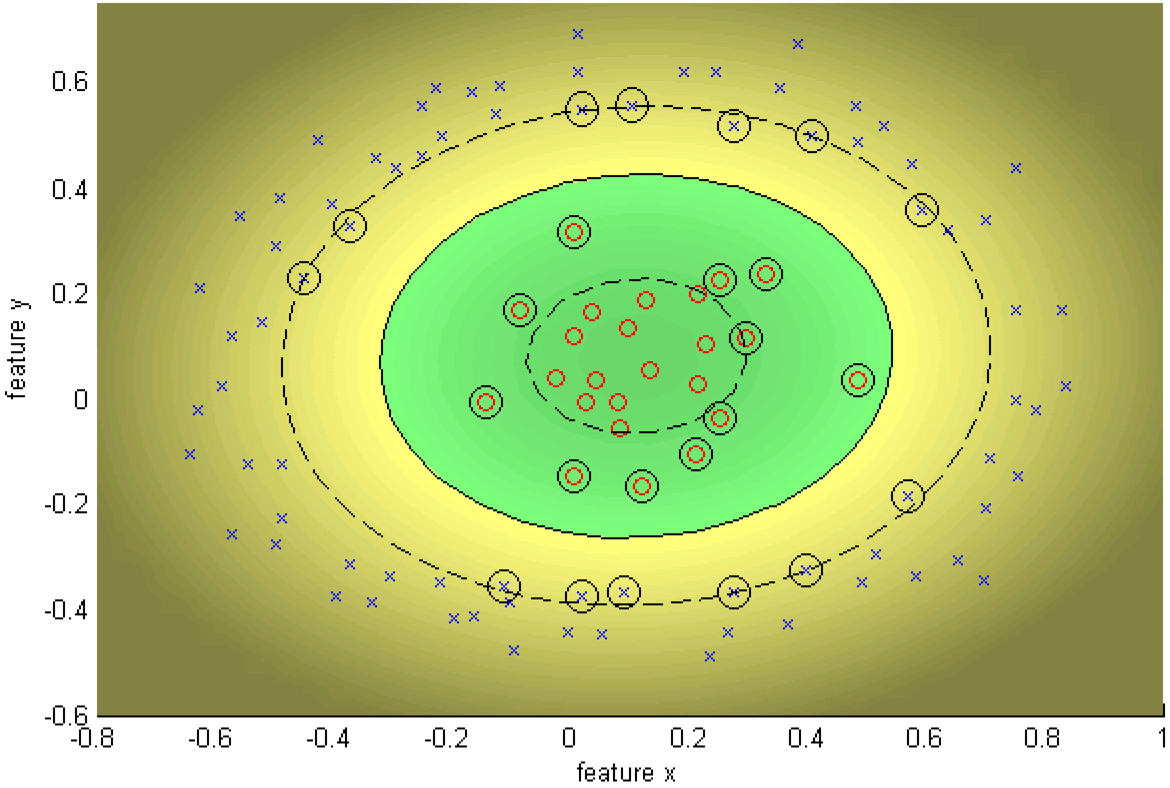
$$f(x) = \sum_{i=1}^{N} \alpha_i y_i exp\left(-\frac{\|\boldsymbol{x} - \boldsymbol{x}_i\|^2}{2\sigma^2}\right) + b$$

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

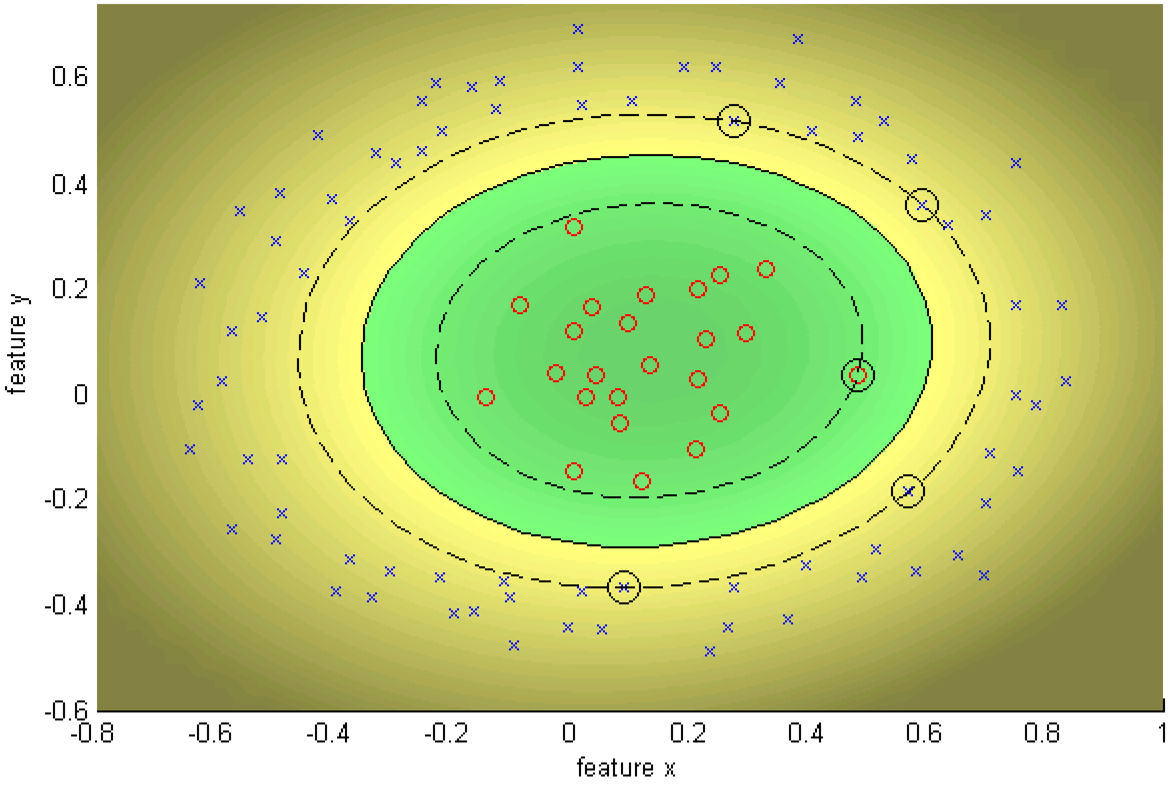


Decrease C, gives wider (soft) margin

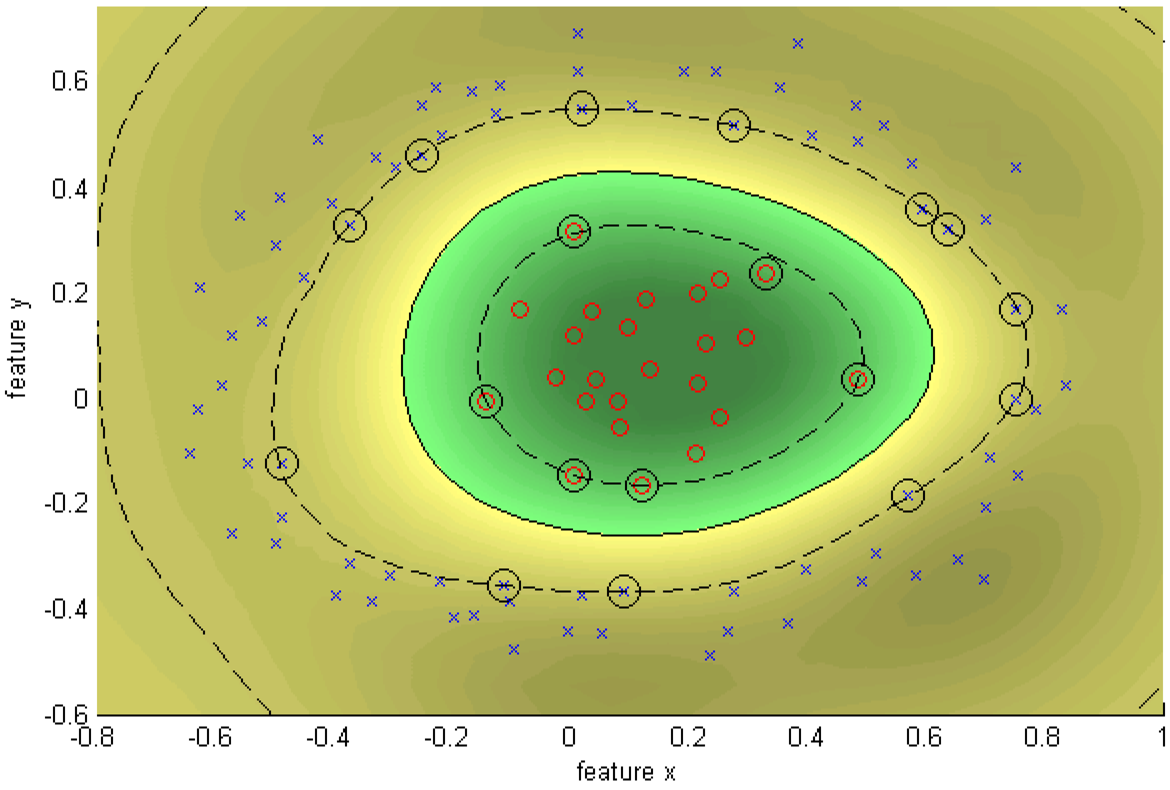
$$\sigma = 1.0 \quad C = 10$$



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

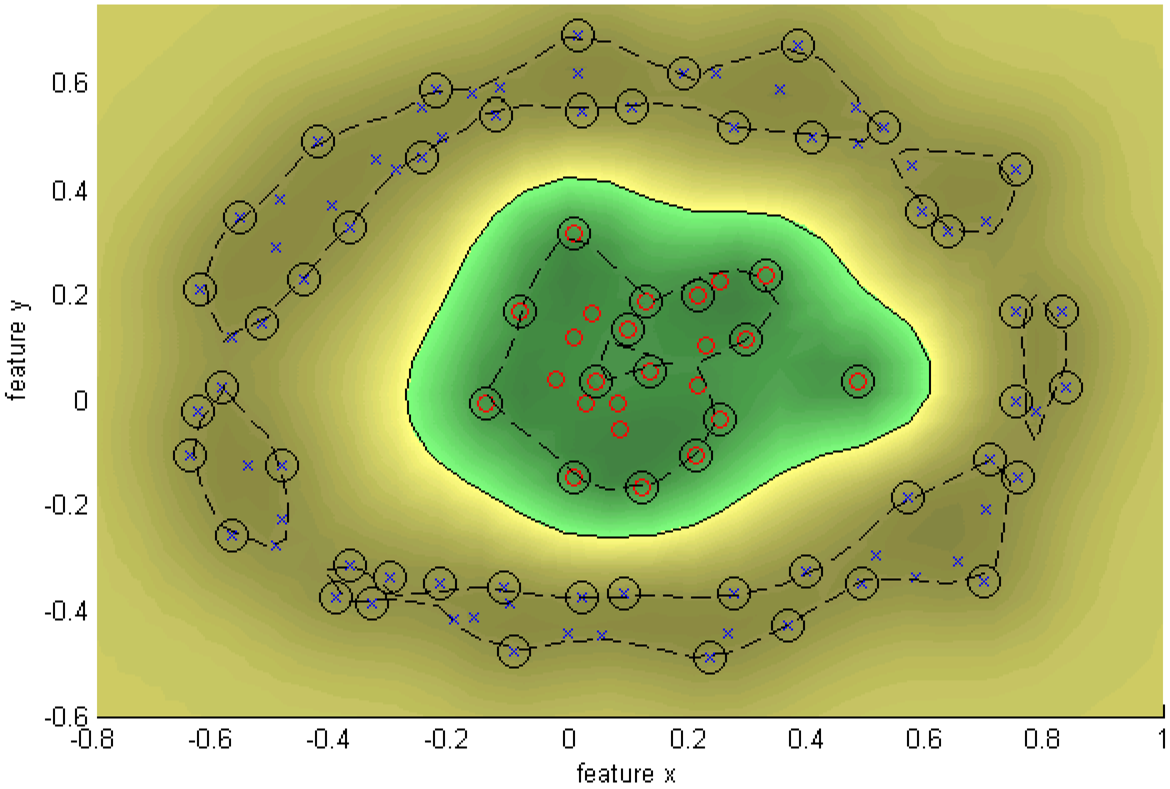


$$\sigma = 0.25 \quad C = \infty$$



Decrease sigma, moves towards nearest neighbor classifier

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



KERNEL TRICK - SUMMARY

- Data may be linearly separable in the high dimensional space, but not linearly separable in the original feature space.
- Classifiers can be learnt for high dimensional features spaces, without actually having to map the points into the high dimensional space.
- Kernels can be used for an SVM because of the scalar product in the dual form, but can also be used elsewhere – they are not tied to the SVM formalism.
- Kernels apply also to objects that are not vectors.