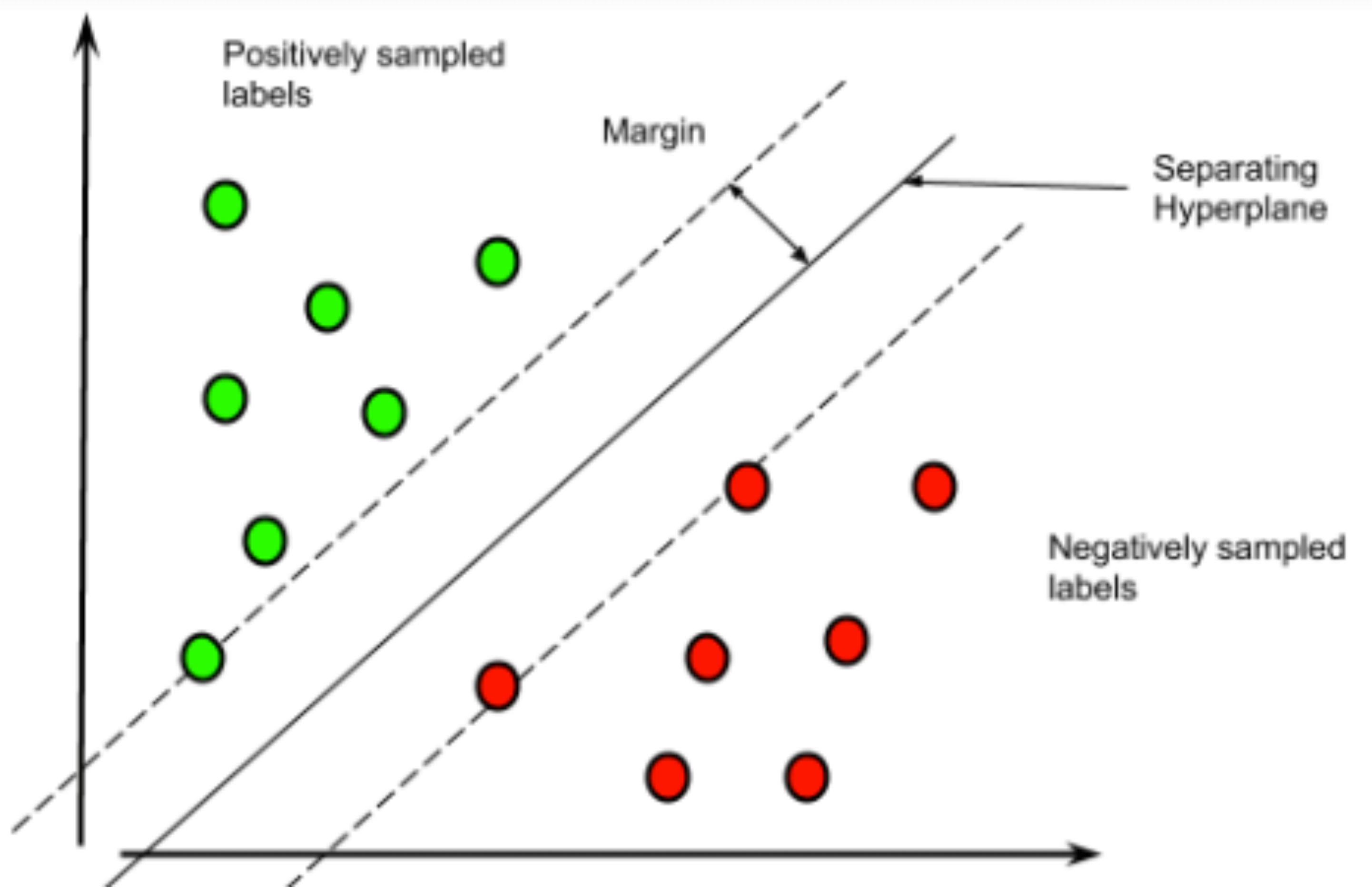


Support Vector Machines (SVM): The Dual Representation

Leandro L. Minku

Overview

- Dual representation of SVM
- Kernel trick
- Making predictions based on the dual

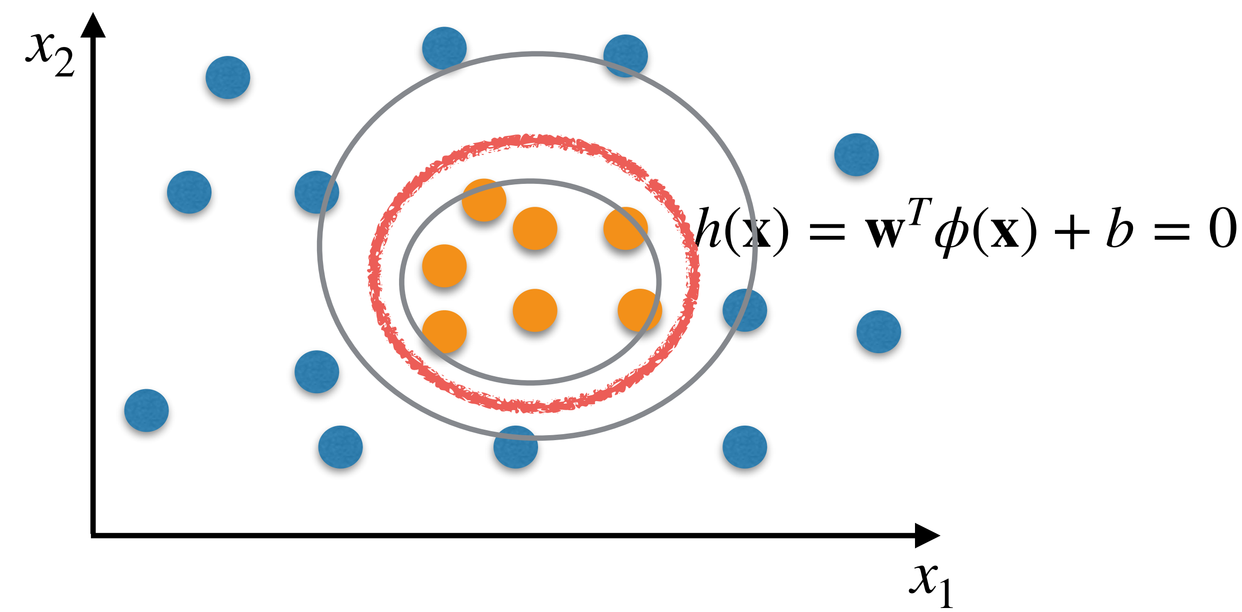
Maximum Margin Classifiers With Basis Expansion

$$\underset{\mathbf{w},b}{\operatorname{argmin}} \left\{ \frac{1}{2} \|\mathbf{w}\|^2 \right\}$$

Subject to

$$y^{(n)}(\mathbf{w}^T \phi(\mathbf{x}) + b) \geq 1,$$

$$\forall (\mathbf{x}^{(n)}, y^{(n)}) \in \mathcal{T}.$$



It is possible to use $\phi(\mathbf{x}) = \mathbf{x}$ if we wish

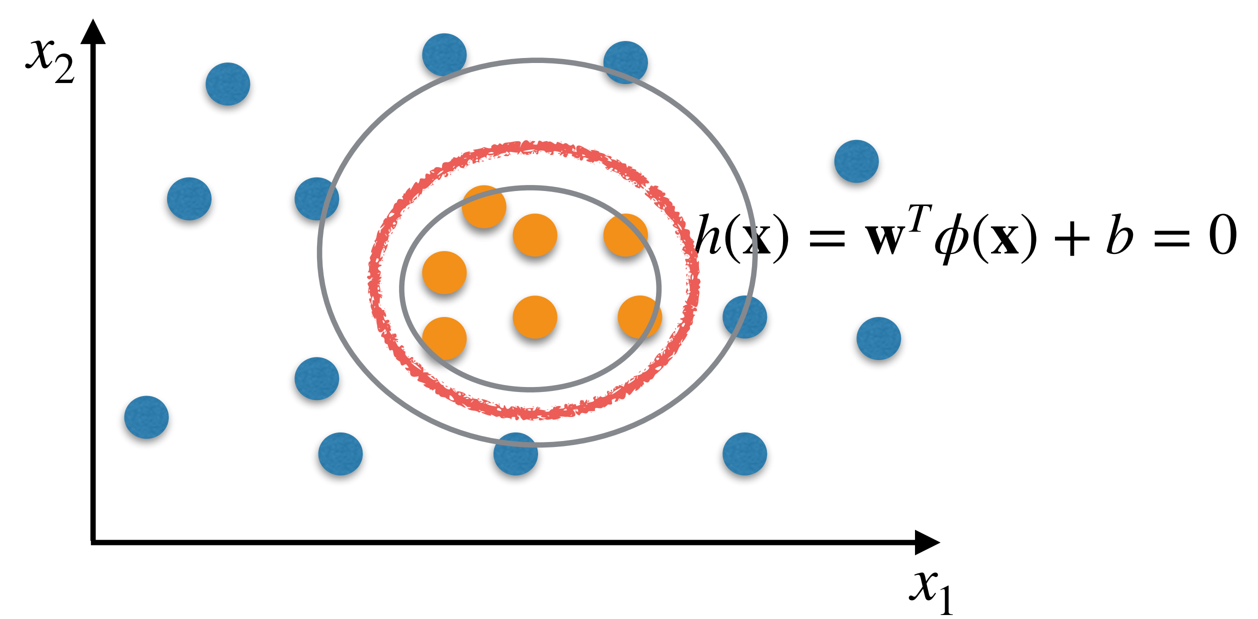
Maximum Margin Classifiers With Basis Expansion

$$\underset{\mathbf{w},b}{\operatorname{argmin}} \left\{ \frac{1}{2} \|\mathbf{w}\|^2 \right\}$$

Subject to

$$y^{(n)}(\mathbf{w}^T \phi(\mathbf{x}) + b) \geq 1,$$

$$\forall (\mathbf{x}^{(n)}, y^{(n)}) \in \mathcal{T}.$$



Depending on $\phi(\mathbf{x})$, its computation can be very expensive, as it may be taking us to a very large dimensional problem.

Rewriting Our Optimisation Problem

Primal Representation

$$\underset{\mathbf{w},b}{\operatorname{argmin}} \left\{ \frac{1}{2} \|\mathbf{w}\|^2 \right\}$$

Subject to: $y^{(n)}(\mathbf{w}^T \phi(\mathbf{x}^{(n)}) + b) \geq 1$

$\forall (\mathbf{x}^{(n)}, y^{(n)}) \in \mathcal{T}$

Dual Representation

We will get rid of $\mathbf{w}$ and b (!!!!!)

We will get rid of $\mathbf{w}^T \phi(\mathbf{x}^{(n)})$ and $\|\mathbf{w}\|^2$

Creating a "Penalty" For The Constraints

Primal Representation

$$\underset{\mathbf{w},b}{\operatorname{argmin}} \left\{ \frac{1}{2}\|\mathbf{w}\|^2 \right\}$$

Subject to: $y^{(n)}(\mathbf{w}^T\phi(\mathbf{x}^{(n)}) + b) \geq 1$

$\forall(\mathbf{x}^{(n)}, y^{(n)}) \in \mathcal{T}$

Subject to: $y^{(n)}(\mathbf{w}^T\phi(\mathbf{x}^{(n)}) + b) - 1 \geq 0$

Subject to: $1 - y^{(n)}(\mathbf{w}^T\phi(\mathbf{x}^{(n)}) + b) \leq 0$

"Penalty" for violated constraints

$$\underset{\mathbf{w},b}{\operatorname{argmin}} \left\{ \frac{1}{2}\|\mathbf{w}\|^2 + g(\mathbf{w}, b) \right\}$$

"Penalty" as a Maximisation Problem

Primal Representation

$$\underset{\mathbf{w},b}{\operatorname{argmin}} \left\{ \frac{1}{2} \|\mathbf{w}\|^2 \right\}$$

Subject to: $1 - y^{(n)}(\mathbf{w}^T \phi(\mathbf{x}^{(n)}) + b) \leq 0$
$\forall (\mathbf{x}^{(n)}, y^{(n)}) \in \mathcal{T}$

$$\underset{\mathbf{w},b}{\operatorname{argmin}} \left\{ \frac{1}{2} \|\mathbf{w}\|^2 + g(\mathbf{w}, b) \right\}$$

The penalty needs to be high when the constraints are violated, leading to the following maximisation problem:

Lagrange multipliers $a^{(n)} \geq 0$

When constraints are violated, this is +

$$g(\mathbf{w}, b) = \max_{\mathbf{a}} \sum_{n=1}^{N} a^{(n)} (1 - y^{(n)}(\mathbf{w}^T \phi(\mathbf{x}^{(n)}) + b))$$

“Penalty” as a Maximisation Problem

Primal Representation

$$\underset{\mathbf{w},b}{\operatorname{argmin}} \left\{ \frac{1}{2}\|\mathbf{w}\|^2 \right\}$$

Subject to: $1 - y^{(n)}(\mathbf{w}^T\phi(\mathbf{x}^{(n)}) + b) \leq 0$
$\forall(\mathbf{x}^{(n)}, y^{(n)}) \in \mathcal{T}$

$$\underset{\mathbf{w},b}{\operatorname{argmin}} \left\{ \frac{1}{2}\|\mathbf{w}\|^2 + g(\mathbf{w}, b) \right\}$$

The penalty needs to be high when the constraints are violated, leading to the following maximisation problem:

Lagrange multipliers $a^{(n)} \geq 0$

When constraints are not violated, this is -

$$g(\mathbf{w}, b) = \max_{\mathbf{a}} \sum_{n=1}^{N} a^{(n)} \underbrace{(1 - y^{(n)}(\mathbf{w}^T\phi(\mathbf{x}^{(n)}) + b))}_{=0}$$

“Penalty” as a Maximisation Problem

Primal Representation

$$\underset{\mathbf{w},b}{\operatorname{argmin}} \left\{ \frac{1}{2} \|\mathbf{w}\|^2 \right\}$$

Subject to: $1 - y^{(n)}(\mathbf{w}^T \phi(\mathbf{x}^{(n)}) + b) \leq 0$
$\forall (\mathbf{x}^{(n)}, y^{(n)}) \in \mathcal{T}$

$$\underset{\mathbf{w},b}{\operatorname{argmin}} \left\{ \frac{1}{2} \|\mathbf{w}\|^2 + g(\mathbf{w}, b) \right\}$$

The penalty needs to be high when the constraints are violated, leading to the following maximisation problem:

Lagrange multipliers $a^{(n)} \geq 0$

or zero

$$g(\mathbf{w}, b) = \max_{\mathbf{a}} \sum_{n=1}^{N} \overset{\geq 0}{a^{(n)}} \left(1 - y^{(n)}(\mathbf{w}^T \phi(\mathbf{x}^{(n)}) + b)\right)$$

A Minmax Problem

Primal Representation

$$\underset{\mathbf{w},b}{\text{argmin}} \left\{ \frac{1}{2}\|\mathbf{w}\|^2 \right\}$$

Subject to: $1 - y^{(n)}(\mathbf{w}^T\phi(\mathbf{x}^{(n)}) + b) \leq 0$
$\forall(\mathbf{x}^{(n)}, y^{(n)}) \in \mathcal{T}$

$$\underset{\mathbf{w},b}{\text{argmin}} \left\{ \frac{1}{2}\|\mathbf{w}\|^2 + g(\mathbf{w}, b) \right\}$$

$$\underset{\mathbf{w},b}{\text{argmin}} \left\{ \frac{1}{2}\|\mathbf{w}\|^2 + \max_{\mathbf{a}} \sum_{n=1}^{N} a^{(n)}(1 - y^{(n)}(\mathbf{w}^T\phi(\mathbf{x}^{(n)}) + b)) \right\}$$

Find $\mathbf{w}, b$ and $\mathbf{a}$ such that: Subject to: $a^{(n)} \geq 0, \ \forall(\mathbf{x}^{(n)}, y^{(n)}) \in \mathcal{T}$

$$\min_{\mathbf{w},b} \max_{\mathbf{a}} \left\{ \frac{1}{2}\|\mathbf{w}\|^2 + \sum_{n=1}^{N} a^{(n)}(1 - y^{(n)}(\mathbf{w}^T\phi(\mathbf{x}^{(n)}) + b)) \right\}$$

Minmax = Maxmin

$$\min_{\mathbf{w},b} \max_{\mathbf{a}} \left\{ \frac{1}{2}\|\mathbf{w}\|^2 + \sum_{n=1}^{N} a^{(n)}(1 - y^{(n)}(\mathbf{w}^T\phi(\mathbf{x}^{(n)}) + b)) \right\}$$

onvex in $\mathbf{w}, b$

If there is at least one $\mathbf{w}$ and b for each n such that $1 - y^{(n)}(\mathbf{w}^T\phi(\mathbf{x}^{(n)}) + b) < 0$

Concave in $\mathbf{a}$

Equivalent to:

$$\max_{\mathbf{a}} \min_{\mathbf{w},b} \left\{ \frac{1}{2}\|\mathbf{w}\|^2 + \sum_{n=1}^{N} a^{(n)}(1 - y^{(n)}(\mathbf{w}^T\phi(\mathbf{x}^{(n)}) + b)) \right\}$$

Further Simplifying Equations

$$\max_{\mathbf{a}} \min_{\mathbf{w},b} L(\mathbf{a}, \mathbf{w}, b) = \left\{ \frac{1}{2}\|\mathbf{w}\|^2 + \sum_{n=1}^{N} a^{(n)}(1 - y^{(n)}(\mathbf{w}^T\phi(\mathbf{x}^{(n)}) + b)) \right\}$$

Once $\mathbf{a}$ is fixed, at the optimum, the gradient $\nabla L(\mathbf{w})$ equals to zero:

$$\mathbf{w} - \sum_{n=1}^{N} a^{(n)}y^{(n)}\phi(\mathbf{x}^{(n)}) = 0 \longrightarrow \mathbf{w} = \sum_{n=1}^{N} a^{(n)}y^{(n)}\phi(\mathbf{x}^{(n)})$$

And so does $\frac{\partial L}{\partial b}$: $\quad \sum_{n=1}^{N} a^{(n)}y^{(n)} = 0$

Further Simplifying Equations

$$\max_{\mathbf{a}} \min_{\mathbf{w},b} L(\mathbf{a}, \mathbf{w}, b) = \left\{ \frac{1}{2}\|\mathbf{w}\|^2 + \sum_{n=1}^{N} a^{(n)}(1 - y^{(n)}(\mathbf{w}^T\phi(\mathbf{x}^{(n)}) + b)) \right\}$$

Considering the following as a constraint and using it to eliminate b:

$$\sum_{n=1}^{N} a^{(n)}y^{(n)} = 0$$

Substituting

$$\mathbf{w} = \sum_{n=1}^{N} a^{(n)}y^{(n)}\phi(\mathbf{x}^{(n)})$$

See the book or lecture notes for a step-by-step on how to use the information above to eliminate $\mathbf{w}$ and b.

The Dual Representation

$$\max_{\mathbf{a}} \min_{\mathbf{w},b} L(\mathbf{a}, \mathbf{w}, b) = \left\{ \frac{1}{2}\|\mathbf{w}\|^2 + \sum_{n=1}^{N} a^{(n)}(1 - y^{(n)}(\mathbf{w}^T \phi(\mathbf{x}^{(n)}) + b)) \right\}$$

$$\Downarrow$$

Lagrangian function

Kernel function

$$\operatorname*{argmax}_{\mathbf{a}} \tilde{L}(\mathbf{a}) = \sum_{n=1}^{N} a^{(n)} - \frac{1}{2} \sum_{n=1}^{N} \sum_{m=1}^{N} a^{(n)} a^{(m)} y^{(n)} y^{(m)} k(\mathbf{x}^{(n)}, \mathbf{x}^{(m)})$$

Dual Representation

Where: $k(\mathbf{x}^{(n)}, \mathbf{x}^{(m)}) = \phi(\mathbf{x}^{(n)})^T \phi(\mathbf{x}^{(m)})$

Subject to: $a^{(n)} \geq 0, \forall n \in \{1, \cdots, N\}$ $\quad \sum_{n=1}^{N} a^{(n)} y^{(n)} = 0$

Why Is This Useful?

$$\max_{\mathbf{a}} \min_{\mathbf{w},b} L(\mathbf{a},\mathbf{w},b) = \left\{ \frac{1}{2}\|\mathbf{w}\|^2 + \sum_{n=1}^{N} a^{(n)}(1 - y^{(n)}(\mathbf{w}^T\phi(\mathbf{x}^{(n)}) + b)) \right\}$$

$$\downarrow$$

Lagrangian function

Kernel function

$$\operatorname*{argmax}_{\mathbf{a}} \tilde{L}(\mathbf{a}) = \sum_{n=1}^{N} a^{(n)} - \frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N} a^{(n)}a^{(m)}y^{(n)}y^{(m)}k(\mathbf{x}^{(n)},\mathbf{x}^{(m)})$$

Dual Representation

Where: $k(\mathbf{x}^{(n)},\mathbf{x}^{(m)}) = \phi(\mathbf{x}^{(n)})^T\phi(\mathbf{x}^{(m)})$

Subject to: $a^{(n)} \geq 0, \forall n \in \{1,\cdots,N\}$ $\quad \sum_{n=1}^{N} a^{(n)}y^{(n)} = 0$

Why Is This Useful?

There is a way to compute $k(\mathbf{x}^{(n)}, \mathbf{x}^{(m)}) = \phi(\mathbf{x}^{(n)})^T \phi(\mathbf{x}^{(m)})$
without having to ever compute $\phi(\mathbf{x})$. This is called the Kernel Trick.

Calculating $k(\mathbf{x}^{(n)}, \mathbf{x}^{(m)})$: The Kernel Trick

$$\mathbf{x} = (x_1, x_2)^T \rightarrow \phi(\mathbf{x}) = (1, x_1, x_2, x_1^2, x_2^2, x_1 x_2)^T$$

$$\mathbf{x} = (x_1, x_2)^T \rightarrow \phi(\mathbf{x}) = (1, \sqrt{2}x_1, \sqrt{2}x_2, x_1^2, x_2^2, \sqrt{2}x_1 x_2)^T$$

$$k(\mathbf{x}^{(n)}, \mathbf{x}^{(m)}) = \phi(\mathbf{x}^{(n)})^T \phi(\mathbf{x}^{(m)})$$

$k(\mathbf{x}, \mathbf{z}) = \phi(\mathbf{x})^T \phi(\mathbf{z})$ Where $\mathbf{x} = \mathbf{x}^{(n)}$ and $\mathbf{z} = \mathbf{x}^{(m)}$.

$$= (1, \sqrt{2}x_1, \sqrt{2}x_2, x_1^2, x_2^2, \sqrt{2}x_1 x_2)\ (1, \sqrt{2}z_1, \sqrt{2}z_2, z_1^2, z_2^2, \sqrt{2}z_1 z_2)^T$$

$$= 1 + 2x_1 z_1 + 2x_2 z_2 + x_1^2 z_1^2 + x_2^2 z_2^2 + 2x_1 x_2 z_1 z_2$$

$$= (1 + x_1 z_1 + x_2 z_2)^2$$

$$= (1 + \mathbf{x}^T \mathbf{z})^2$$

—> these are the original input variables!

Creating Polynomial Kernels

- This calculation can be generalised to basis expansions composed of all terms of order up to p.

$$k(\mathbf{x}, \mathbf{z}) = (1 + \mathbf{x}^T \mathbf{z})^p$$

Can We Generalise This Notion Further?

- It is possible to construct kernel functions directly, rather than designing and computing their basis expansions ϕ explicitly.
- Even though we don't need to compute ϕ, the kernel function must correspond to **some** embedding.
- In particular, it needs to correspond to the inner product in some embedding.

$$k(\mathbf{x}^{(n)}, \mathbf{x}^{(m)}) = \phi(\mathbf{x}^{(n)})^T \phi(\mathbf{x}^{(m)})$$

- We can check if it does based on the Mercer's condition.

Mercer's Condition

- Consider any finite set of points $\mathbf{x}^{(1)}, \cdots, \mathbf{x}^{(M)}$ (not necessarily the training set).

- Gram matrix: An $M \times M$ similarity matrix $\mathbf{K}$, whose elements are given by $K_{i,j} = k(\mathbf{x}^{(i)}, \mathbf{x}^{(j)})$.

- Mercer's condition states that $\mathbf{K}$ must be symmetric and positive semidefinite.
  - Symmetric: $k(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}) = k(\mathbf{x}^{(j)}, \mathbf{x}^{(i)})$.
  - Positive semidefinite: $\mathbf{z}^T \mathbf{K} \mathbf{z} \geq 0, \forall \mathbf{z} \in \mathbb{R}^M$.

If these conditions are satisfied, the inner product defined by the kernel in the feature space respects the properties of inner products.

Kernel Composition Rules

Given valid kernels $k_1(\mathbf{x}, \mathbf{z})$ and $k_2(\mathbf{x}, \mathbf{z})$, the following will also be valid kernels:

$$k(\mathbf{x}, \mathbf{z}) = ck_1(\mathbf{x}, \mathbf{z})$$

where $c \geq 0$ is a constant.

$$k(\mathbf{x}, \mathbf{z}) = f(\mathbf{x})k_1(\mathbf{x}, \mathbf{z})f(\mathbf{z})$$

where $f(\cdot)$ is any function.

$$k(\mathbf{x}, \mathbf{z}) = q(k_1(\mathbf{x}, \mathbf{z}))$$

where $q(\cdot)$ is a polynomial with non-negative coefficients.

$$k(\mathbf{x}, \mathbf{z}) = e^{k_1(\mathbf{x}, \mathbf{z})}$$

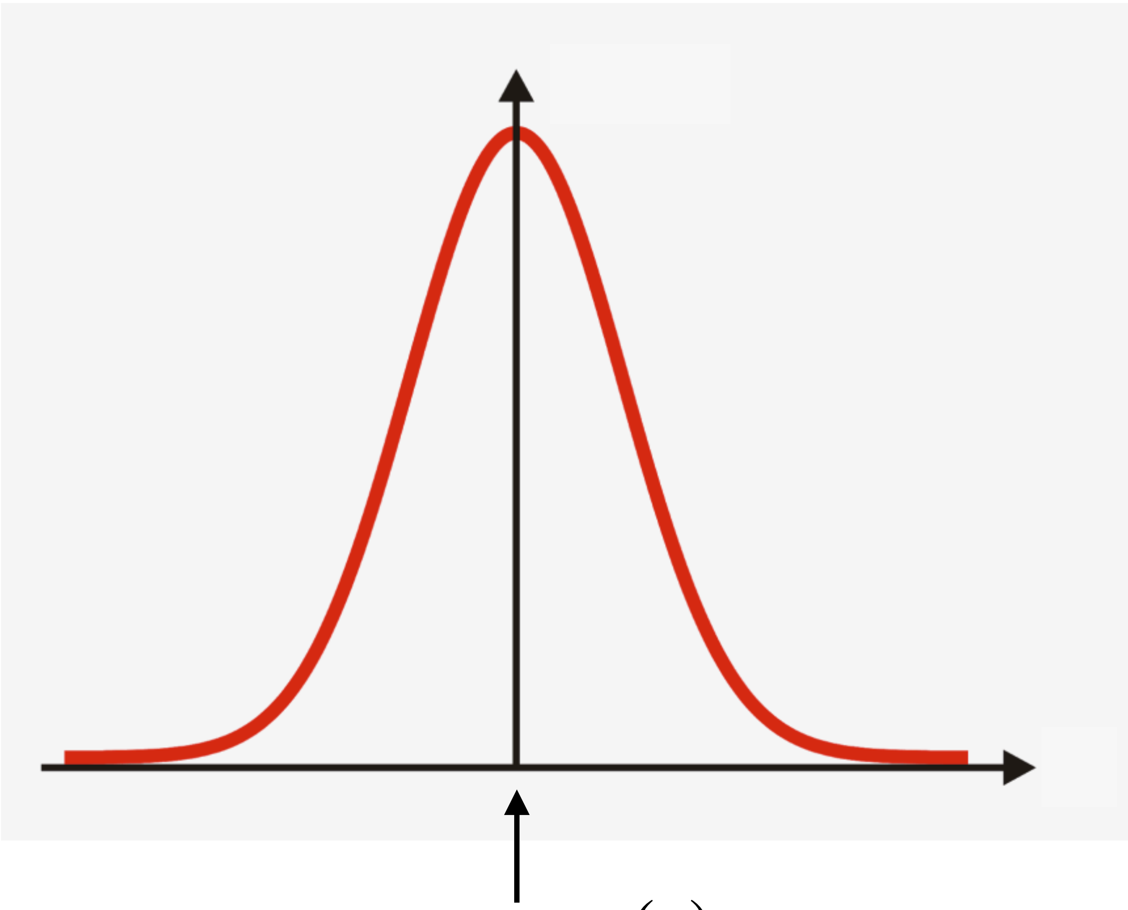
$$k(\mathbf{x}, \mathbf{z}) = k_1(\mathbf{x}, \mathbf{z}) + k_2(\mathbf{x}, \mathbf{z})$$

$$k(\mathbf{x}, \mathbf{z}) = k_1(\mathbf{x}, \mathbf{z})k_2(\mathbf{x}, \mathbf{z})$$

Gaussian Kernel

- Gaussian kernel, a.k.a. Radial Basis Function (RBF) kernel.

$$k(\mathbf{x}, \mathbf{x}^{(n)}) = e^{-\frac{\|\mathbf{x} - \mathbf{x}^{(n)}\|^2}{2\sigma^2}}$$



The embedding ϕ is infinite dimensional!

Taylor series with infinite terms gives a representation of the true Gaussian itself.

E.g., for $\sigma = 1$

$$k(\mathbf{x}, \mathbf{x}^{(n)}) = \sum_{j=0}^{\infty} \frac{(\mathbf{x}^T \mathbf{x}^{(n)})^j}{j!} e^{-\frac{1}{2}\|\mathbf{x}\|^2} e^{-\frac{1}{2}\|\mathbf{x}^{(n)}\|^2}$$

Proving That the Gaussian Kernel is a Valid Kernel, Assuming $\mathbf{x}^T\mathbf{z}$ Is A Valid Kernel

$$k(\mathbf{x}, \mathbf{z}) = e^{-\frac{\|\mathbf{x}-\mathbf{z}\|^2}{2\sigma^2}} \qquad = \sum_{j=0}^{\infty} \frac{(\mathbf{x}^T\mathbf{z})^j}{j!} e^{-\frac{1}{2}\|\mathbf{x}\|^2} e^{-\frac{1}{2}\|\mathbf{z}\|^2}$$

$$= \sum_{j=0}^{\infty} e^{-\frac{1}{2}\|\mathbf{x}\|^2} \frac{(\mathbf{x}^T\mathbf{z})^j}{j!} e^{-\frac{1}{2}\|\mathbf{z}\|^2} \qquad k(\mathbf{x}, \mathbf{z}) = k_1(\mathbf{x}, \mathbf{z}) + k_2(\mathbf{x}, \mathbf{z})$$

$$e^{-\frac{1}{2}\|\mathbf{x}\|^2} \frac{(\mathbf{x}^T\mathbf{z})^j}{j!} e^{-\frac{1}{2}\|\mathbf{z}\|^2} \qquad k(\mathbf{x}, \mathbf{z}) = f(\mathbf{x}) k_1(\mathbf{x}, \mathbf{z}) f(\mathbf{z})$$

Proving That the Gaussian Kernel is a Valid Kernel

$$e^{-\frac{1}{2}\|\mathbf{x}\|^2} \frac{(\mathbf{x}^T\mathbf{z})^j}{j!} e^{-\frac{1}{2}\|\mathbf{z}\|^2}$$

$$k(\mathbf{x}, \mathbf{z}) = f(\mathbf{x}) k_1(\mathbf{x}, \mathbf{z}) f(\mathbf{z})$$

$$\frac{(\mathbf{x}^T\mathbf{z})^j}{j!}$$

$$k(\mathbf{x}, \mathbf{z}) = c k_1(\mathbf{x}, \mathbf{z})$$

$$(\mathbf{x}^T\mathbf{z})^j$$

$$k(\mathbf{x}, \mathbf{z}) = k_1(\mathbf{x}, \mathbf{z}) k_2(\mathbf{x}, \mathbf{z})$$

Summary

- The dual representation can avoid having to compute the basis expansions (feature transformations).
- This allows us to use very high dimensional embeddings, even infinite dimensional ones such as when the kernel is Gaussian.
- It is possible to design a kernel without having to design the nonlinear transformation.
- The kernel will be a valid kernel (i.e., correspond to an inner product in some embedding) if it satisfies the Mercer's condition.
- To avoid having to prove the Mercer's condition, it is possible to create kernels based on kernel composition rules.

Making Predictions

Primal: $h(\mathbf{x}) = \mathbf{w}^T\phi(\mathbf{x}) + b$

$h(\mathbf{x}) > 0 \rightarrow$ class +1

$h(\mathbf{x}) < 0 \rightarrow$ class -1

Substituting $\mathbf{w} = \sum_{n=1}^{N} a^{(n)} y^{(n)} \phi(\mathbf{x}^{(n)})$

Dual: $h(\mathbf{x}) = \sum_{n=1}^{N} a^{(n)} y^{(n)} k(\mathbf{x}, \mathbf{x}^{(n)}) \; + b$

$h(\mathbf{x}) > 0 \rightarrow$ class +1

$h(\mathbf{x}) < 0 \rightarrow$ class -1

We are now making predictions on new examples based on the training examples.

Do we need to store and go through all training examples for making predictions?

Dual: $f(\mathbf{x}) = \sum_{n=1}^{N} a^{(n)} y^{(n)} k(\mathbf{x}, \mathbf{x}^{(n)}) \;\; + b$

$f(\mathbf{x}) > 0 \rightarrow$ class +1

$f(\mathbf{x}) < 0 \rightarrow$ class -1

- For every training example,
  - Either: $a^{(n)} = 0$, so the value of $y^{(n)} k(\mathbf{x}, \mathbf{x}^{(n)})$ won't matter.
  - Or: $1 - y^{(n)}(\mathbf{w}^T \phi(\mathbf{x}^{(n)}) + b) = 0$, we have $y^{(n)} h(\mathbf{x}^{(n)}) = 1$, i.e., this is a support vector.
  - PS: the case where $1 - y^{(n)}(\mathbf{w}^T \phi(\mathbf{x}^{(n)}) + b) > 0$ won't happen if the optimisation problem is successfully solved.

  - So, if the optimisation problem is successfully solved, we only need to store support vectors.

$$g(\mathbf{w}, b) = \max_{\mathbf{a}} \sum_{n=1}^{N} a^{(n)} (1 - y^{(n)}(\mathbf{w}^T \phi(\mathbf{x}^{(n)}) + b))$$

Function f Using Only Support Vectors

$$f(\mathbf{x}) = \sum_{n=1}^{N} a^{(n)} y^{(n)} k(\mathbf{x}, \mathbf{x}^{(n)}) \ + b$$

$$\downarrow$$

$$f(\mathbf{x}) = \sum_{n \in S} a^{(n)} y^{(n)} k(\mathbf{x}, \mathbf{x}^{(n)}) \ + b$$

where S is the set of indexes of the support vectors

Calculating b

Note that $y^{(n)} f(\mathbf{x}^{(n)}) = 1$ for all support vectors.

So, for a given support vector $(\mathbf{x}^{(n)}, y^{(n)})$, we have that:

$y^{(n)} f(\mathbf{x}^{(n)}) = 1$ Multiply by $y^{(n)}$

${y^{(n)}}^2 f(\mathbf{x}^{(n)}) = y^{(n)}$ Note that ${y^{(n)}}^2 = 1$

$f(\mathbf{x}^{(n)}) = y^{(n)}$ Substituting $f(\mathbf{x}^{(n)}) = \sum_{m \in S} a^{(m)} y^{(m)} k(\mathbf{x}^{(n)}, \mathbf{x}^{(m)}) + b$

$$\sum_{m \in S} a^{(m)} y^{(m)} k(\mathbf{x}^{(n)}, \mathbf{x}^{(m)}) + b = y^{(n)}$$

$$b = y^{(n)} - \sum_{m \in S} a^{(m)} y^{(m)} k(\mathbf{x}^{(n)}, \mathbf{x}^{(m)})$$

Averaging for All Support Vectors

$$b = y^{(n)} - \sum_{m \in S} a^{(m)} y^{(m)} k(\mathbf{x}^{(n)}, \mathbf{x}^{(m)})$$

- We have N_S support vectors.
- We can compute b for each of them and average the results:

$$b = \frac{1}{N_S} \sum_{n \in S} \left(y^{(n)} - \sum_{m \in S} a^{(m)} y^{(m)} k(\mathbf{x}^{(n)}, \mathbf{x}^{(m)}) \right)$$

where S is the set of indexes of the support vectors and N_S is the number of support vectors.

Summary

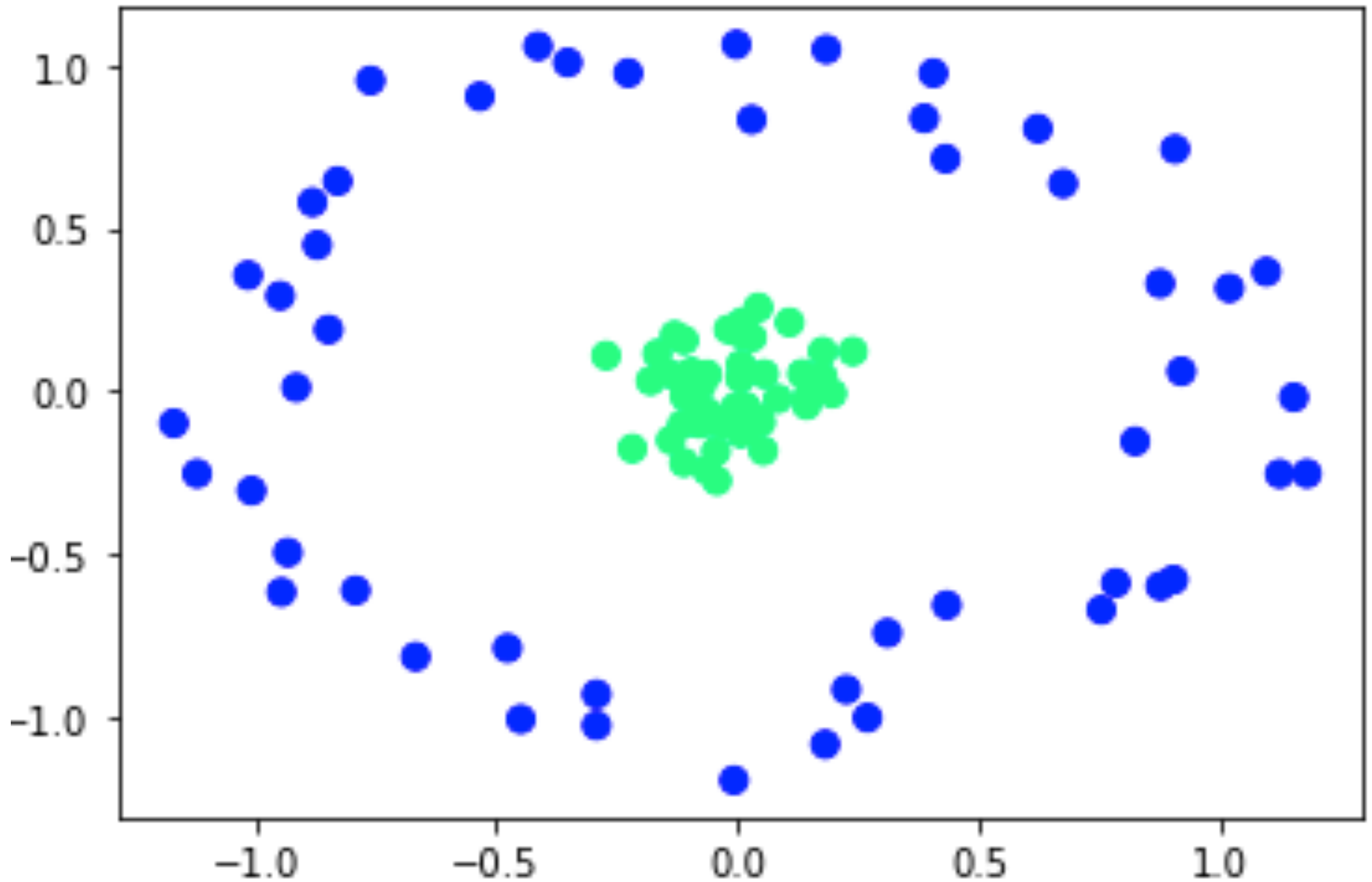
- Predictions using the dual representation are based on the support vectors!

Further Reading

- Essential:
  - Abu-Mostafa et al.'s Learning from Data: A Short Course, e-Chapter 8 (Support Vector Machines): https://amlbook.com/eChapters/8-Jan2015-readeronly.pdf. Read Section 8.2 (Dual Formulation of the SVM) and Section 8.3 (Kernel Trick).

- Recommended:
  - Bishop's "Pattern Recognition and Machine Learning", Section 7.1 (Maximum Margin Classifiers) until right before Section 7.1.1.
  - Leandro's notes on SVMs, Sections 1—5: https://canvas.bham.ac.uk/files/15659719/download?download_frd=1

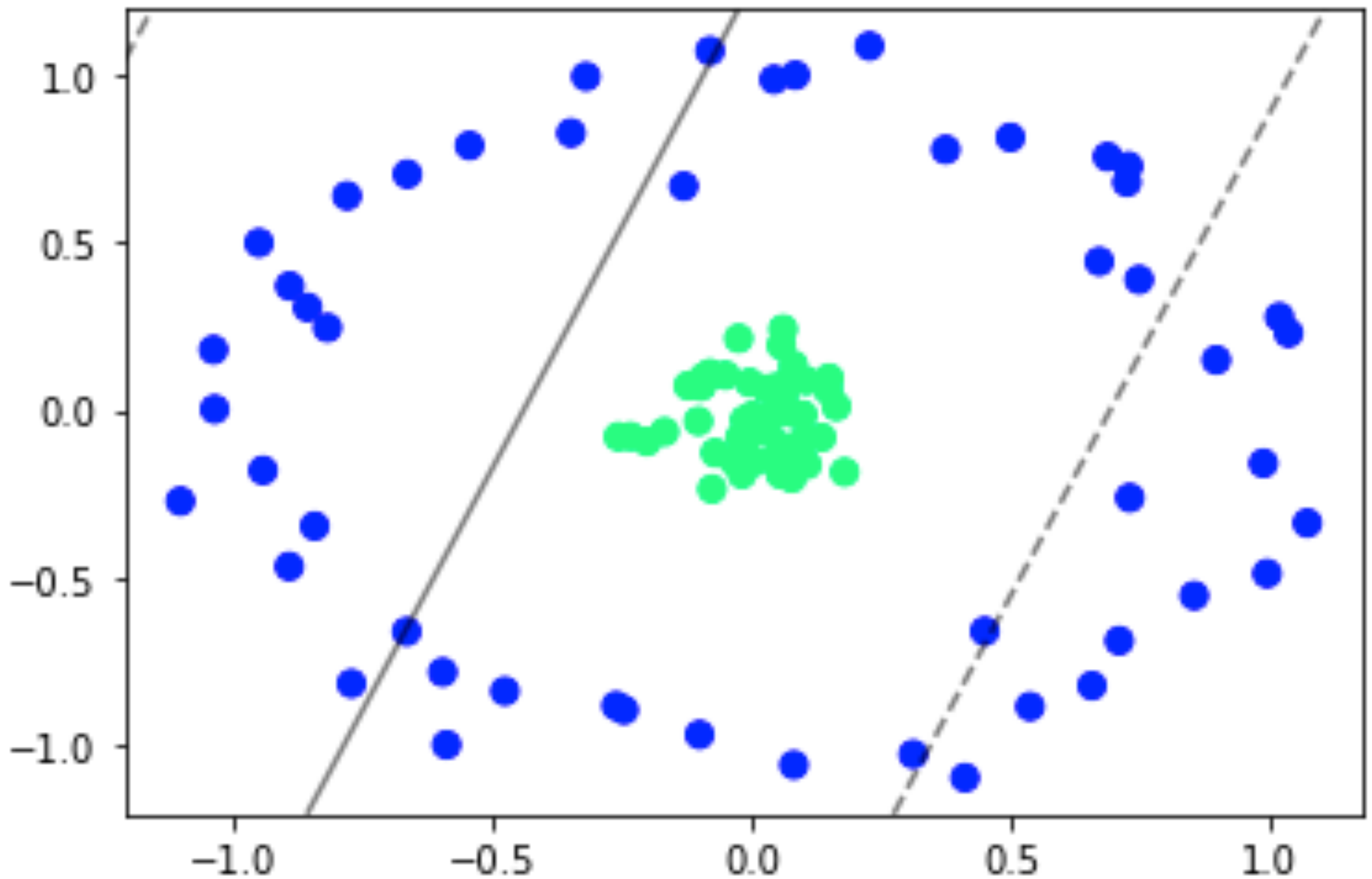
Example



Code adapted from: https://jakevdp.github.io/PythonDataScienceHandbook/05.07-support-vector-machines.html

Example

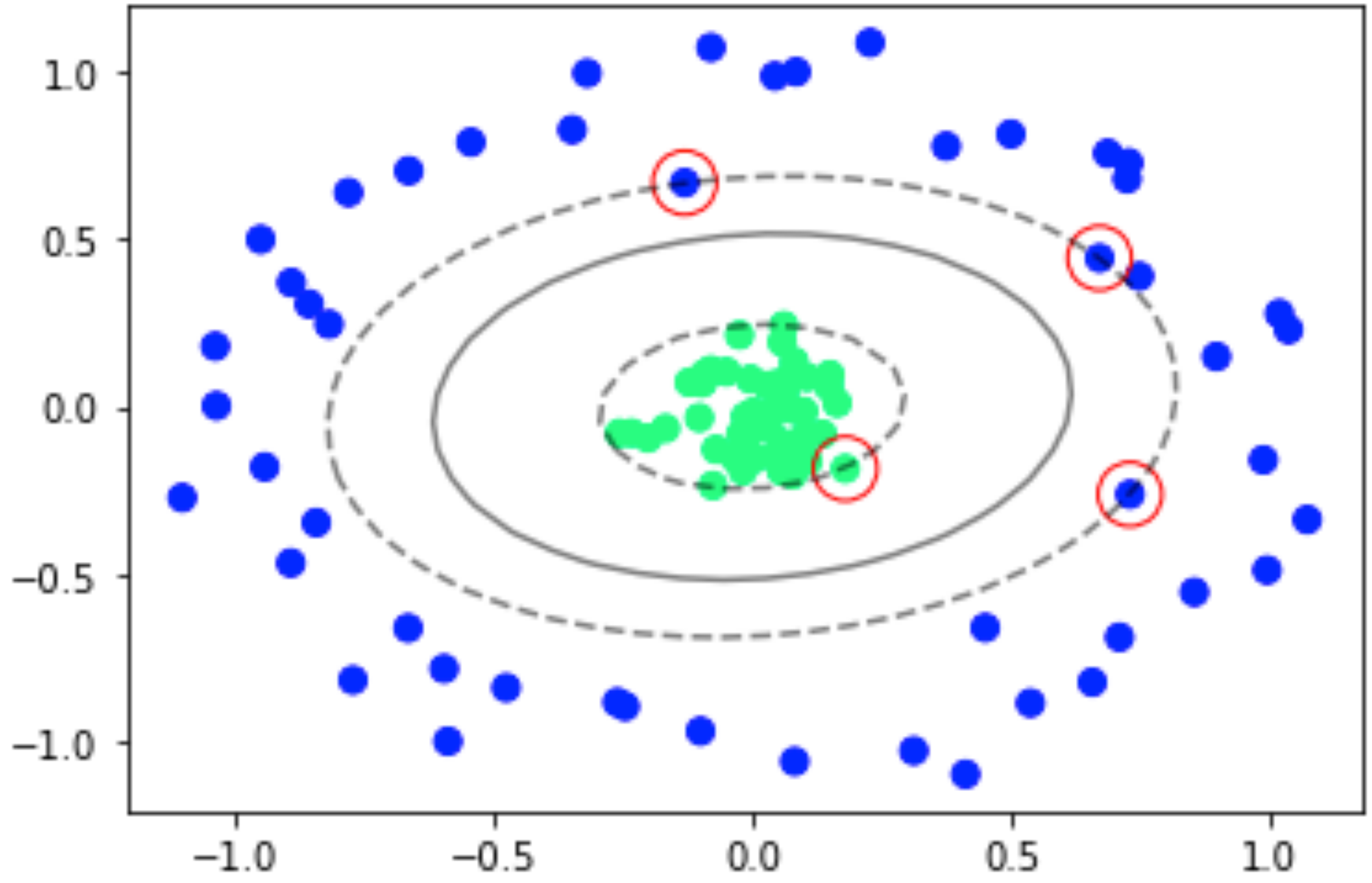
Using $\phi(\mathbf{x}) = \mathbf{x}$, linear kernel: $k(\mathbf{x}^{(n)}, \mathbf{x}^{(m)}) = \mathbf{x}^{(n)^T}\mathbf{x}^{(m)}$



Code adapted from: https://jakevdp.github.io/PythonDataScienceHandbook/05.07-support-vector-machines.html

Example

Using polynomial embedding of degree 2

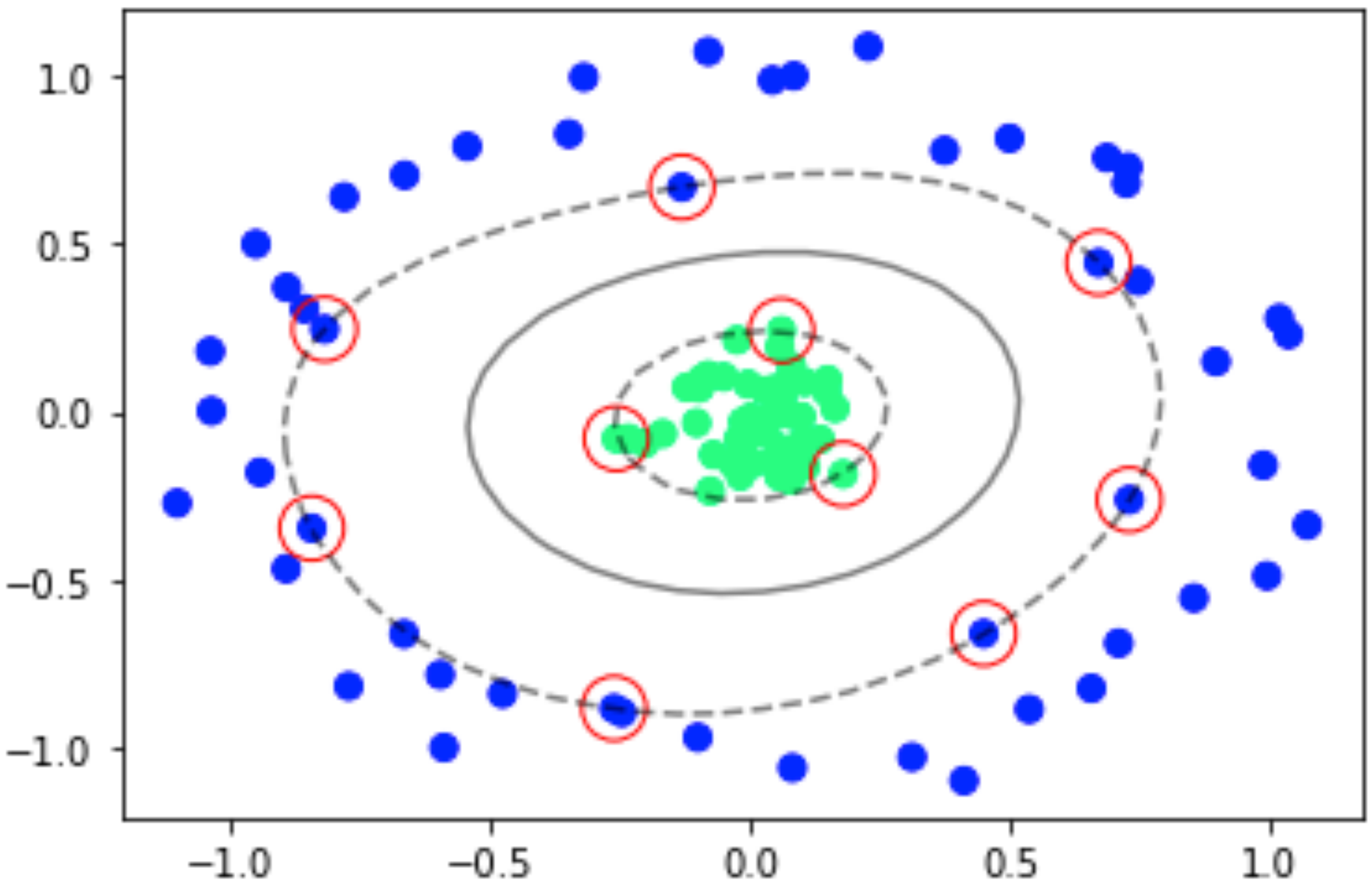


Code adapted from: https://jakevdp.github.io/PythonDataScienceHandbook/05.07-support-vector-machines.html

* Red circles represent the support vectors

Example

Using Gaussian kernel: $k(\mathbf{x}^{(n)}, \mathbf{x}^{(m)}) = e^{-\frac{\|\mathbf{x}^{(n)} - \mathbf{x}^{(m)}\|^2}{2\sigma^2}}$

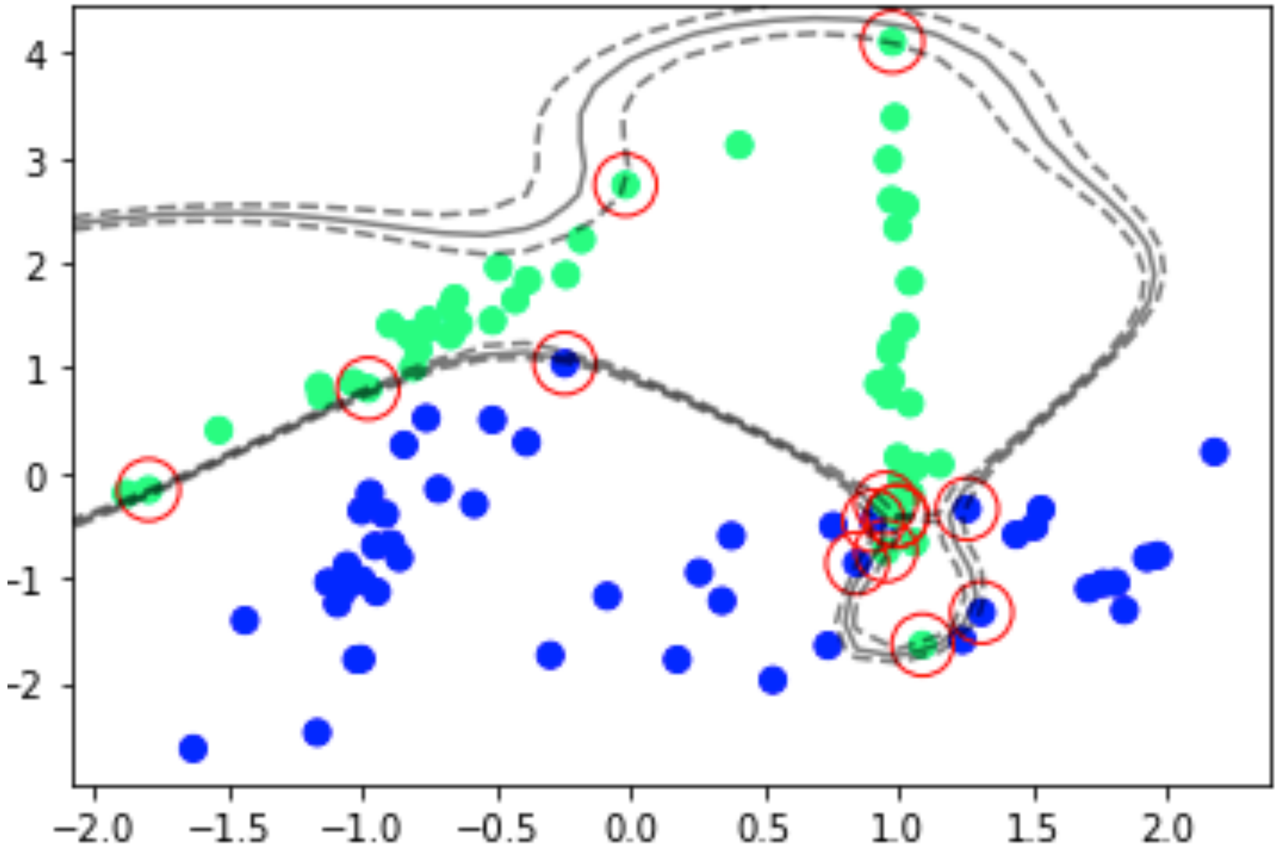


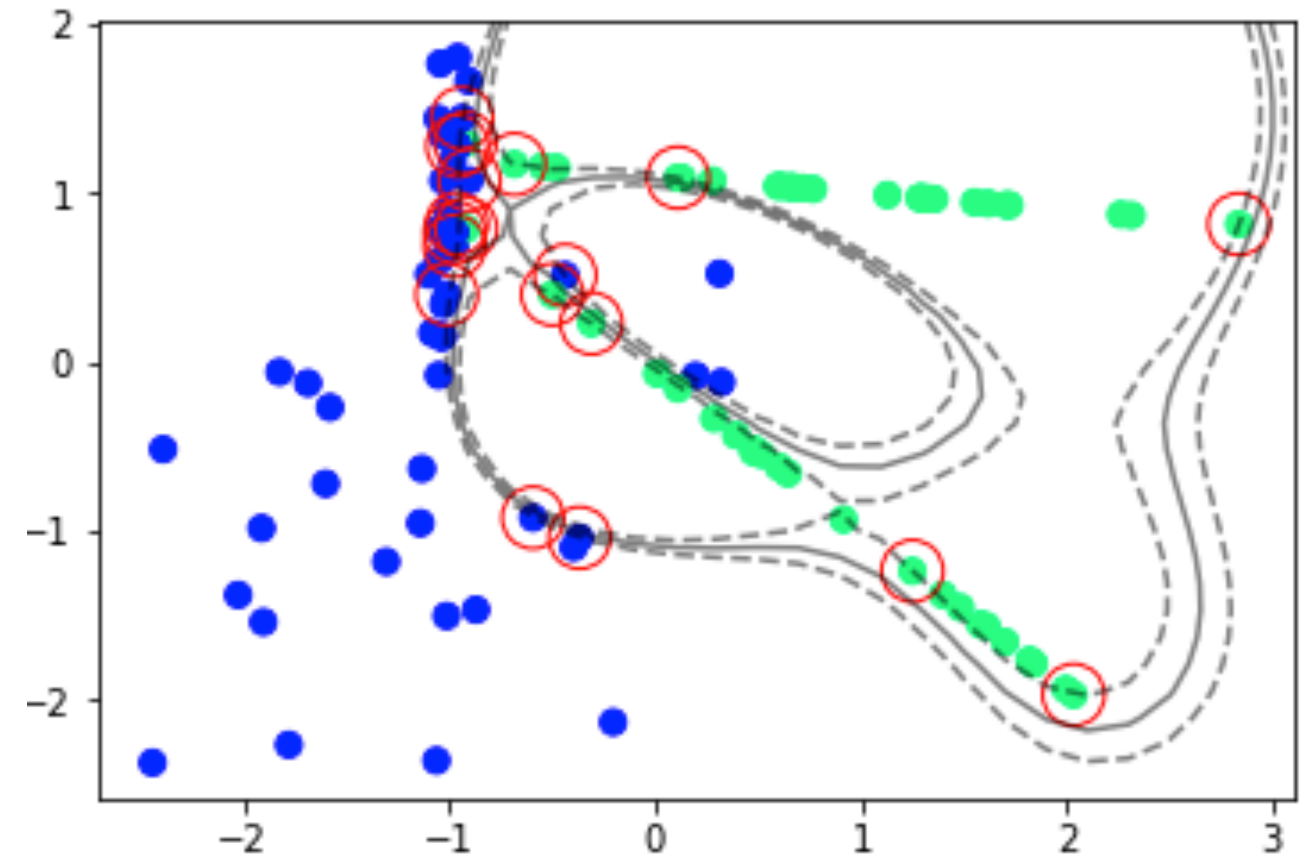
Code adapted from: https://jakevdp.github.io/PythonDataScienceHandbook/05.07-support-vector-machines.html

* Red circles represent the support vectors

Example

Using Gaussian kernel: $k(\mathbf{x}^{(n)}, \mathbf{x}^{(m)}) = e^{-\frac{\|\mathbf{x}^{(n)} - \mathbf{x}^{(m)}\|^2}{2\sigma^2}}$





Code adapted from: https://jakevdp.github.io/PythonDataScienceHandbook/05.07-support-vector-machines.html

* Red circles represent the support vectors

Overfitting?

- One may be concerned with overfitting if we are using such high dimensional embedding as the one underlying the Gaussian kernel.
- Maximising the margin can help coping with overfitting.
- Still, some overfitting may occur. For that, we will learn about the soft margin SVM next.