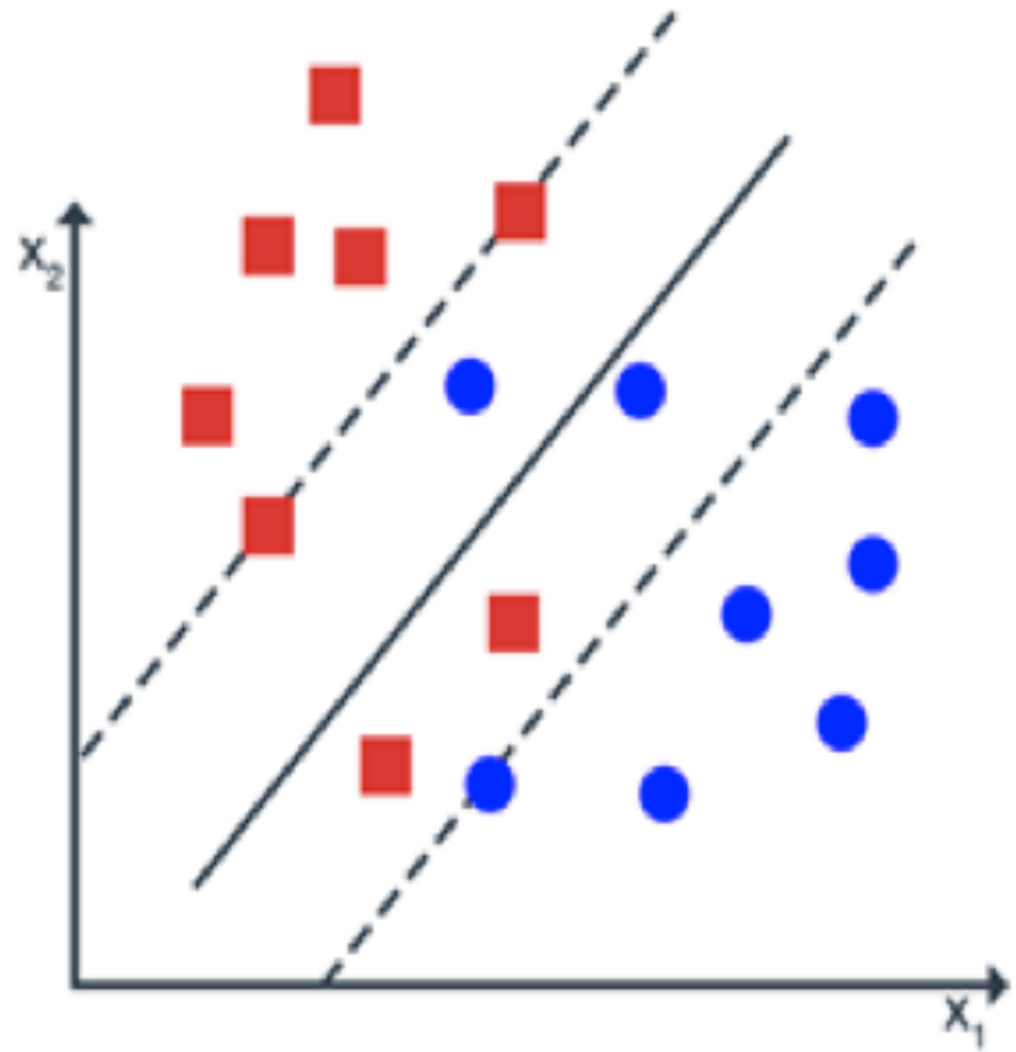


Support Vector Machines: Soft Margin

Leandro L. Minku

Overview

- Making predictions based on the dual representation
- Soft margin SVM
  - Primal
  - Dual
  - Making predictions

SVM's Primal and Dual Representations

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

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

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

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

Making Predictions

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

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

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

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

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

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

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

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

- For every training example,
  - It will be correctly classified (if the SVM problem is feasible).

When constraints are violated, this is +

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

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

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

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

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

- For every training example,
  - It will be correctly classified (if the SVM problem is feasible).
  - Either: $a^{(n)} = 0$, so the value of $y^{(n)} k(\mathbf{x}, \mathbf{x}^{(n)})$ won't matter.

When constraints are not violated, this may be -

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

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

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

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

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

- For every training example,
  - It will be correctly classified (if the SVM problem is feasible).
  - Either: $a^{(n)} = 0$, so the value of $y^{(n)} k(\mathbf{x}, \mathbf{x}^{(n)})$ won't matter.
  - Or: $a^{(n)} > 0$ and $1 - y^{(n)}(\mathbf{w}^T \phi(\mathbf{x}^{(n)}) + b) = 0$, i.e., this is a support vector.

For examples on the margin, this will be 0

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

Function h Using Only Support Vectors

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

$$\downarrow$$

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

where S is the set of indexes of the support vectors

We only need to store the support vectors for making predictions.

Calculating b

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

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

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

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

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

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

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

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 to get a numerically more stable solution:

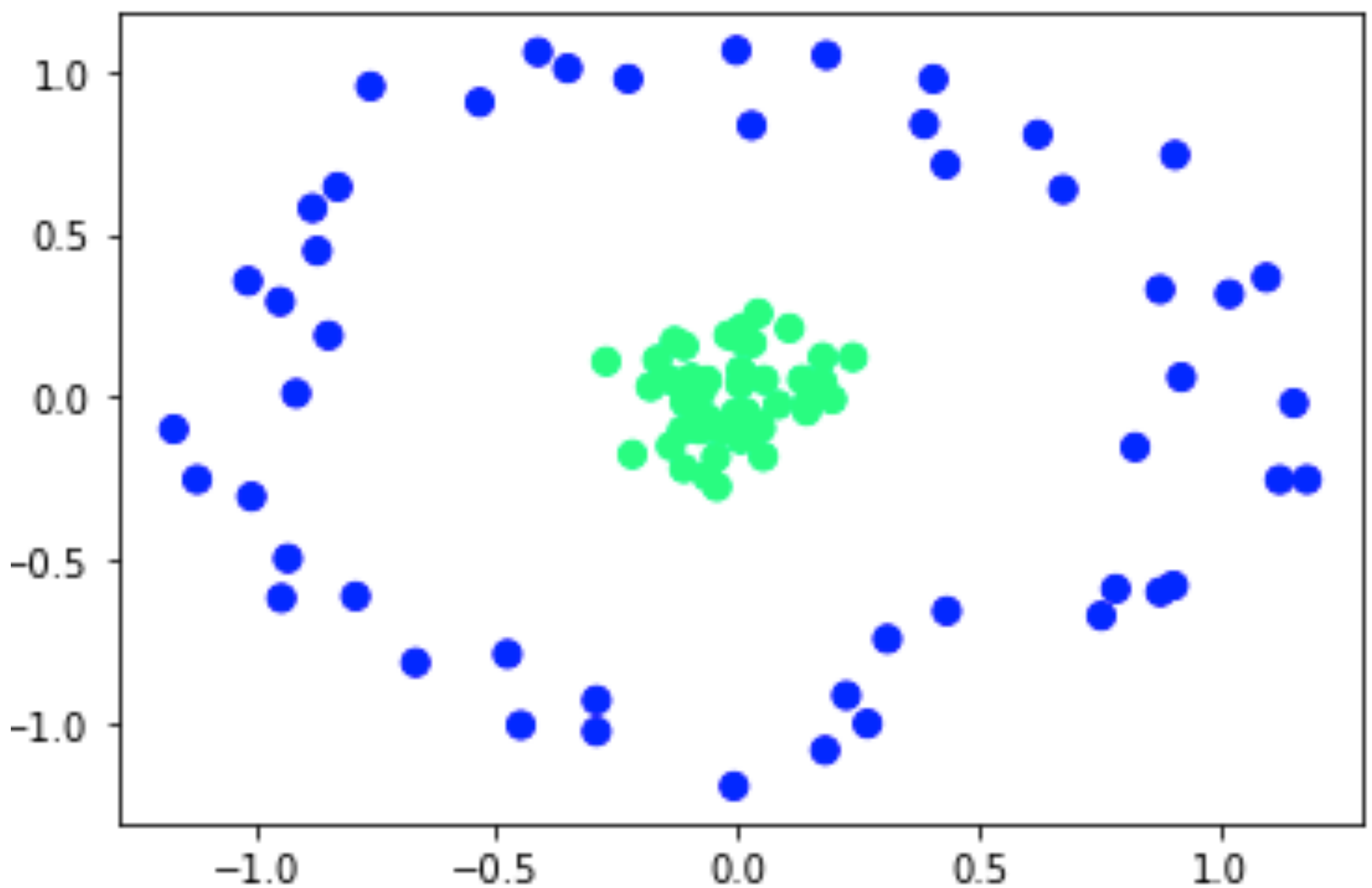
$$b = \frac{1}{N_S} \sum_{n \in S} \left(y^{(n)} - \sum_{m \in S} a^{(m)} y^{(m)} k(\mathbf{x}^{(n)}, \mathbf{x}^{(m)}) \right)$$

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

Adopting the dual representation enables us to adopt powerful kernels, e.g., the Gaussian kernel, which takes us to an infinite dimensional embedding.

Predictions when using the dual representation are based on the support vectors.

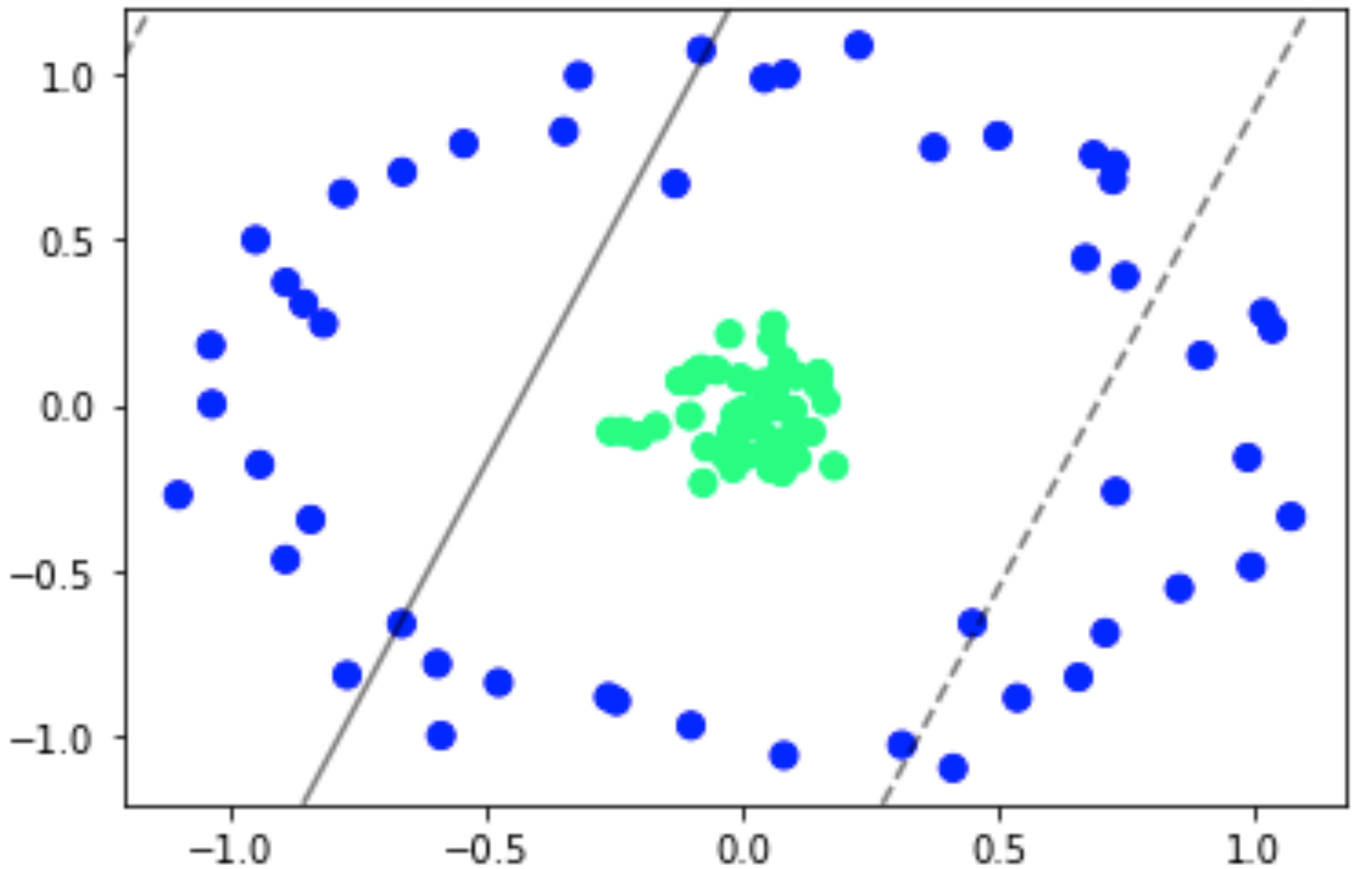
Example



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

Example

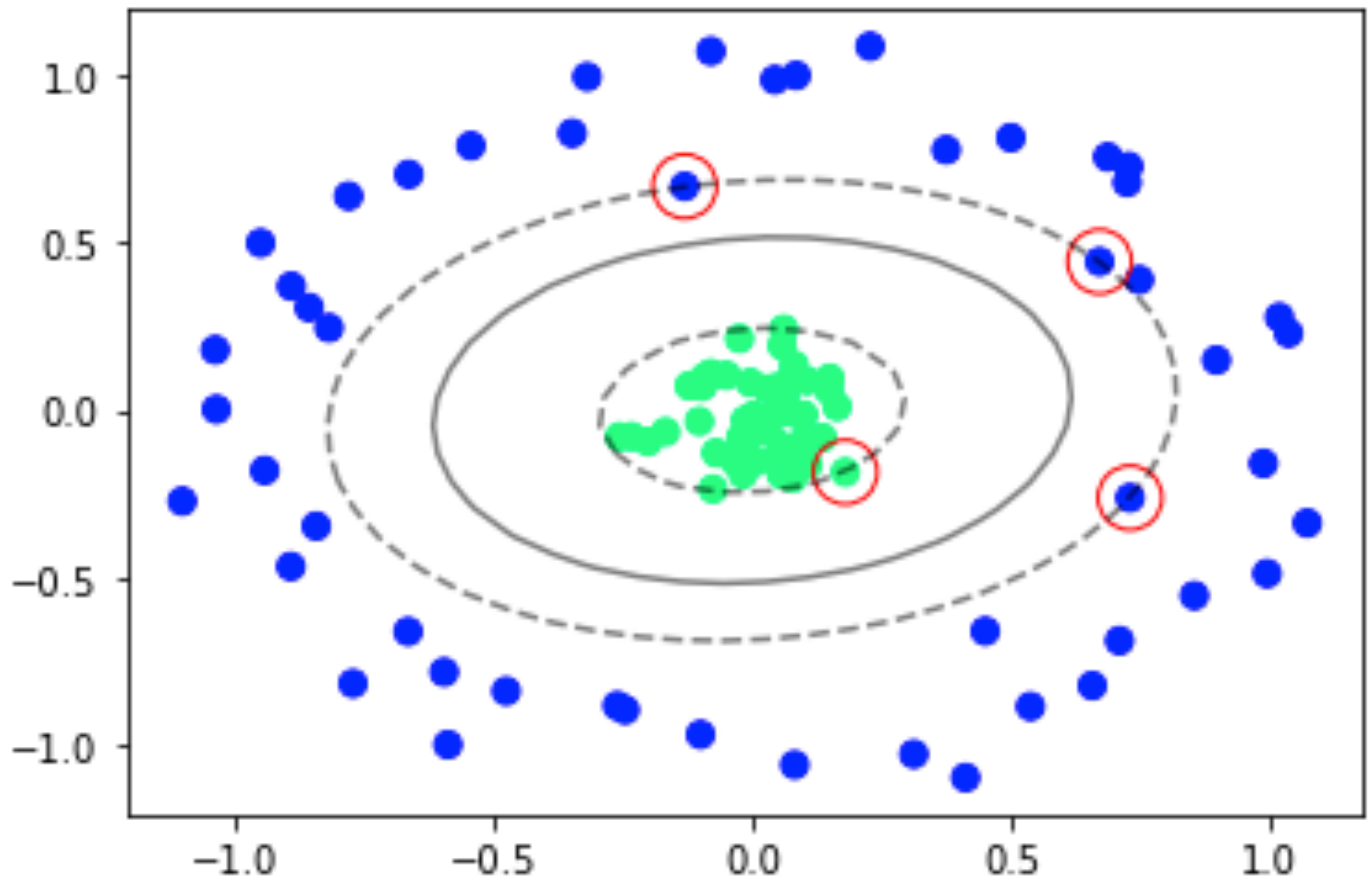
Using $\phi(\mathbf{x}) = \mathbf{x}$, linear kernel: $k(\mathbf{x}^{(n)}, \mathbf{x}^{(m)}) = \mathbf{x}^{(n)^T}\mathbf{x}^{(m)}$



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

Example

Using polynomial embedding of degree 2

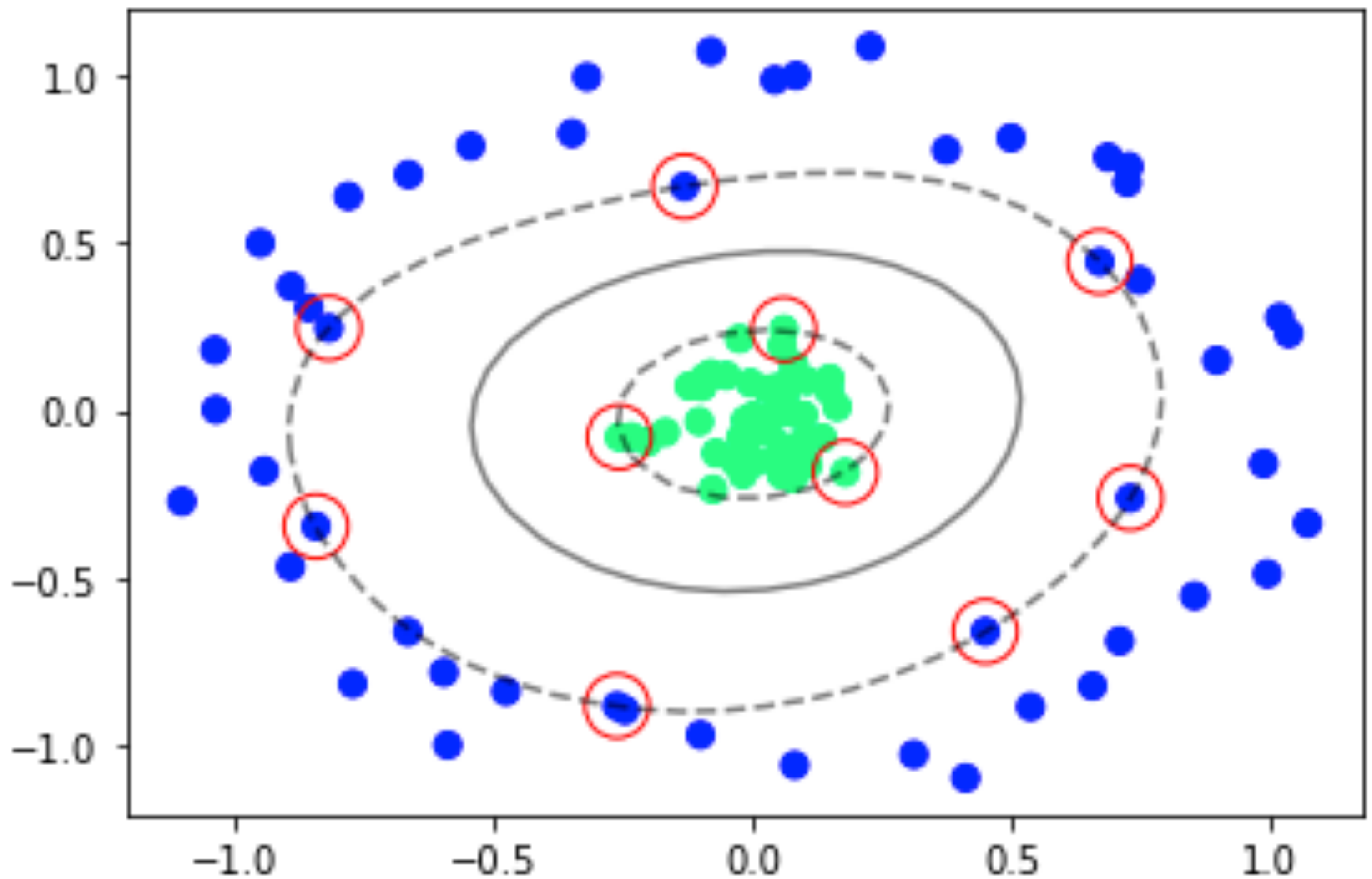


Code adapted from: https://jakevdp.github.io/PythonDataScienceHandbook/05.07-support-vector-machines.html

* Red circles represent the support vectors

Example

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

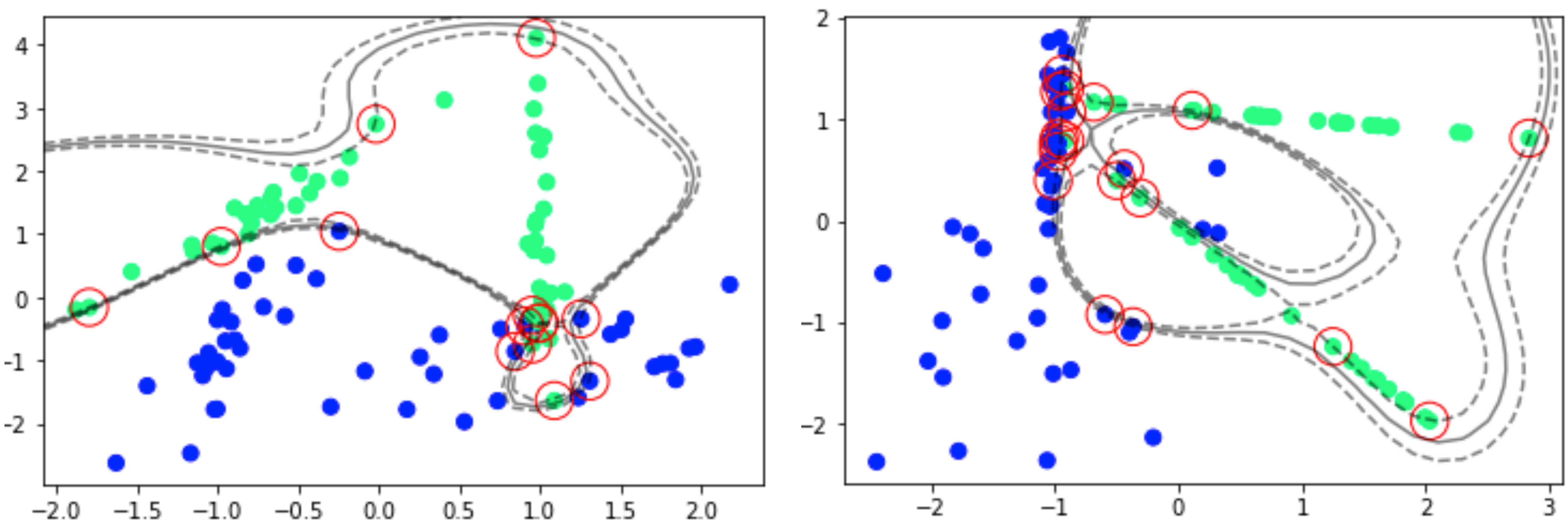


Code adapted from: https://jakevdp.github.io/PythonDataScienceHandbook/05.07-support-vector-machines.html

* 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?

- Overfitting happens when we fit noise in our training data and as a result worsen the generalisation capability.

- One may be concerned with overfitting if we are using such high dimensional embedding as the one underlying the Gaussian kernel.

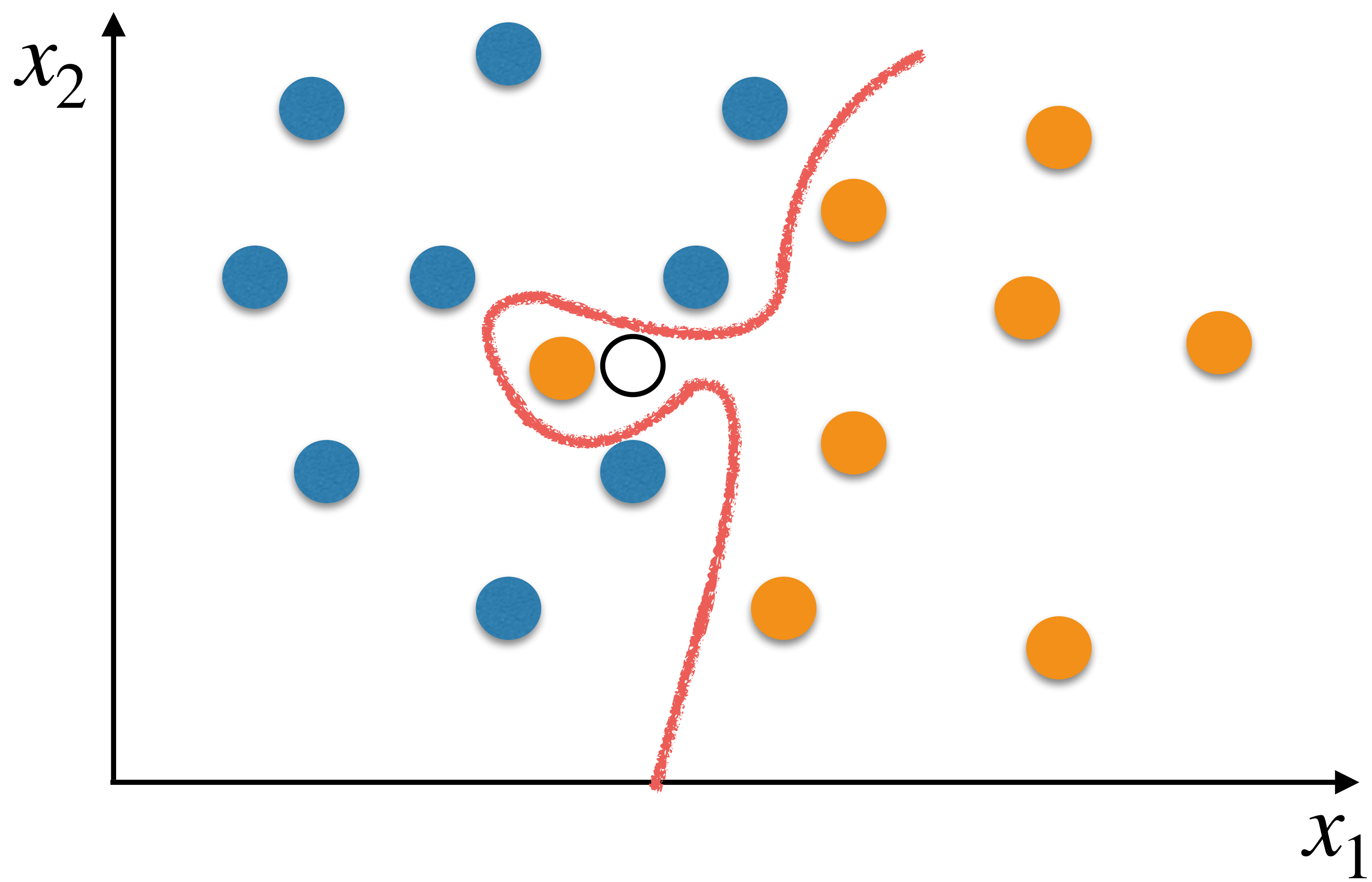
- Maximising the margin can help coping with overfitting.

- Still, some overfitting may occur. For that, we will learn about the soft margin SVM next.

Support Vector Machines: Soft Margin

General Idea

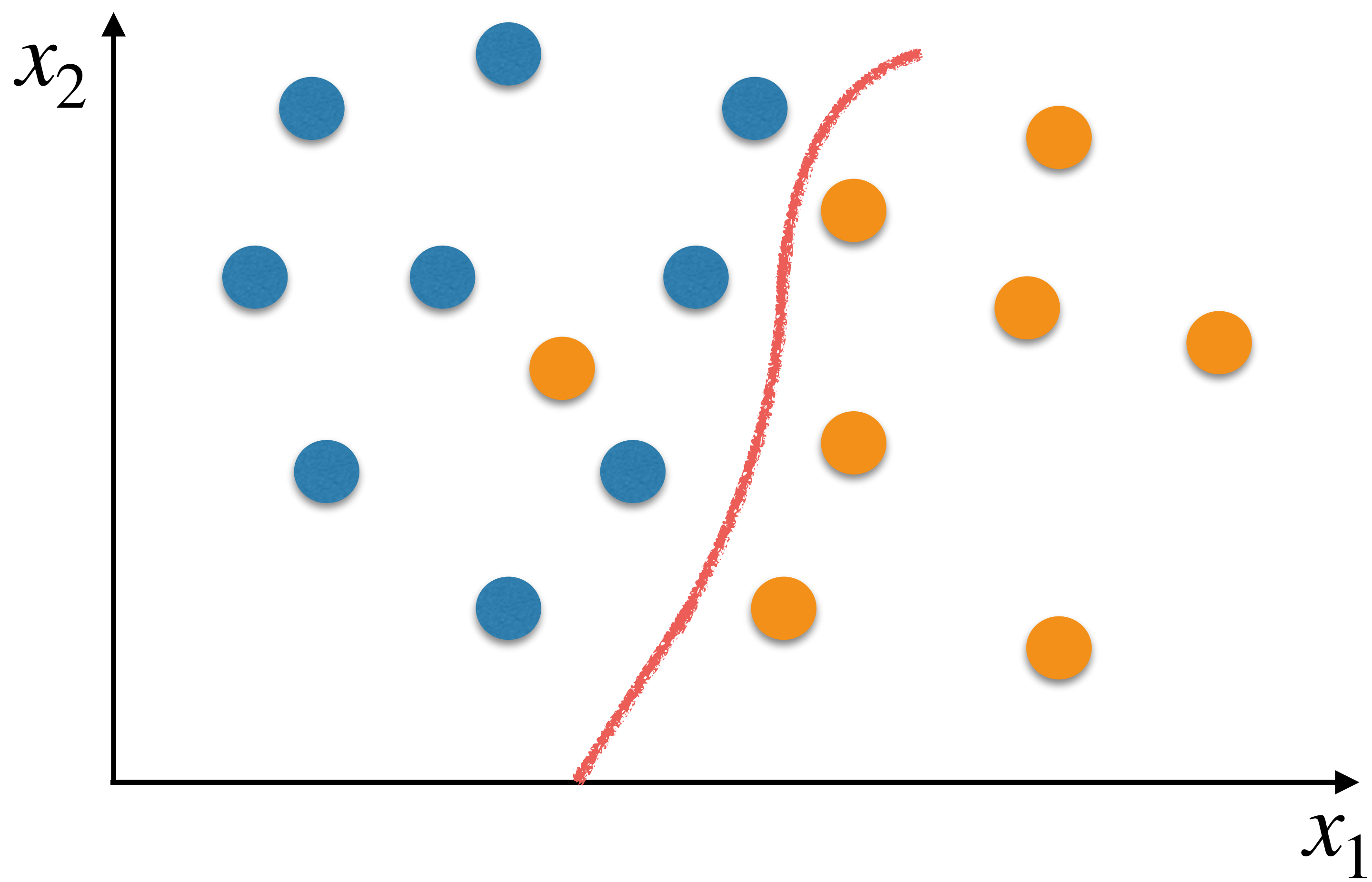
Our (kernelised) maximum margin classifiers assume that the training data are linearly separable (in the higher dimensional embedding).



They will try to perfectly separate the training examples, which may lead to poor generalisation.

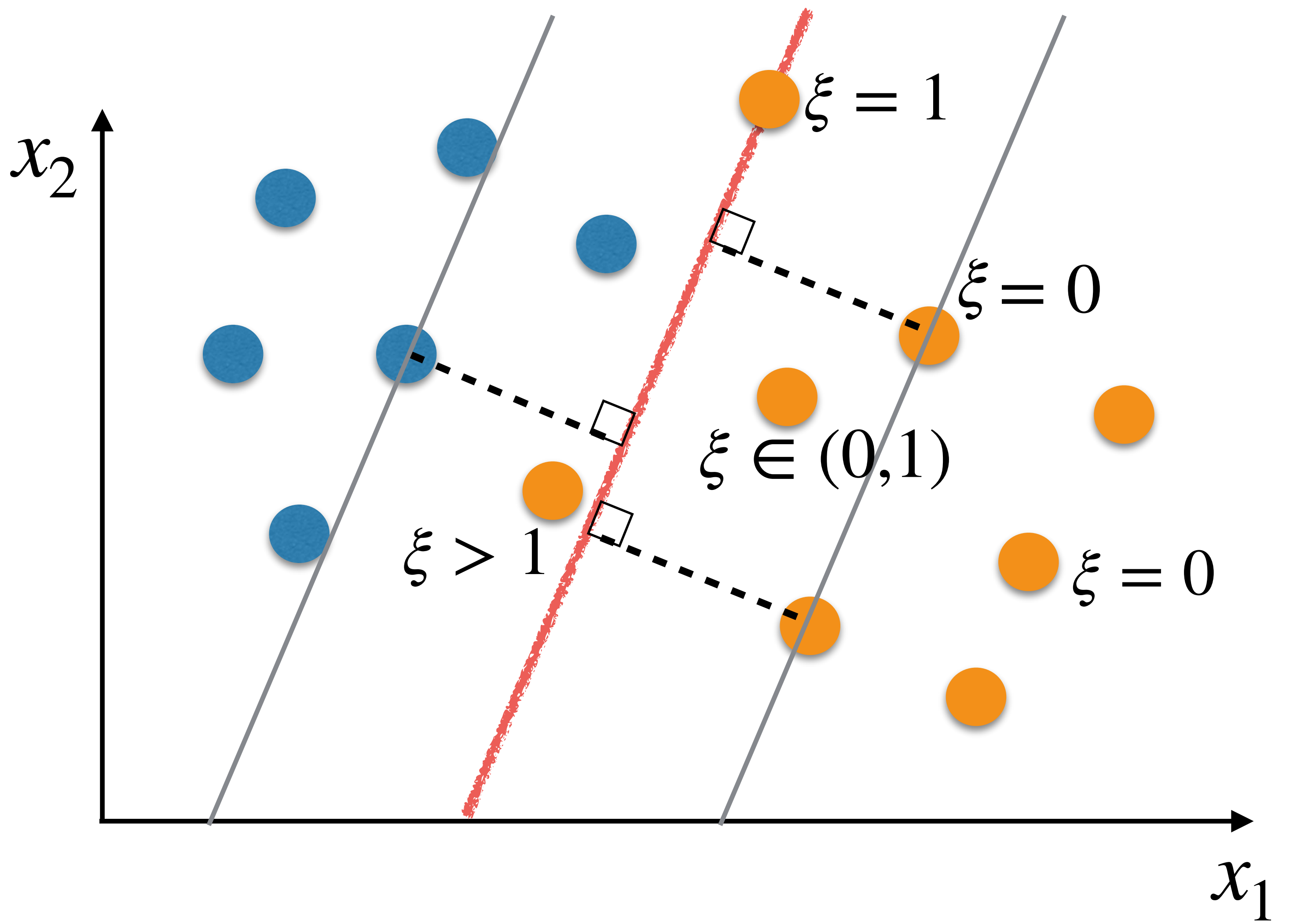
General Idea

It may be better to misclassify some training examples!



Slack Variables ξ

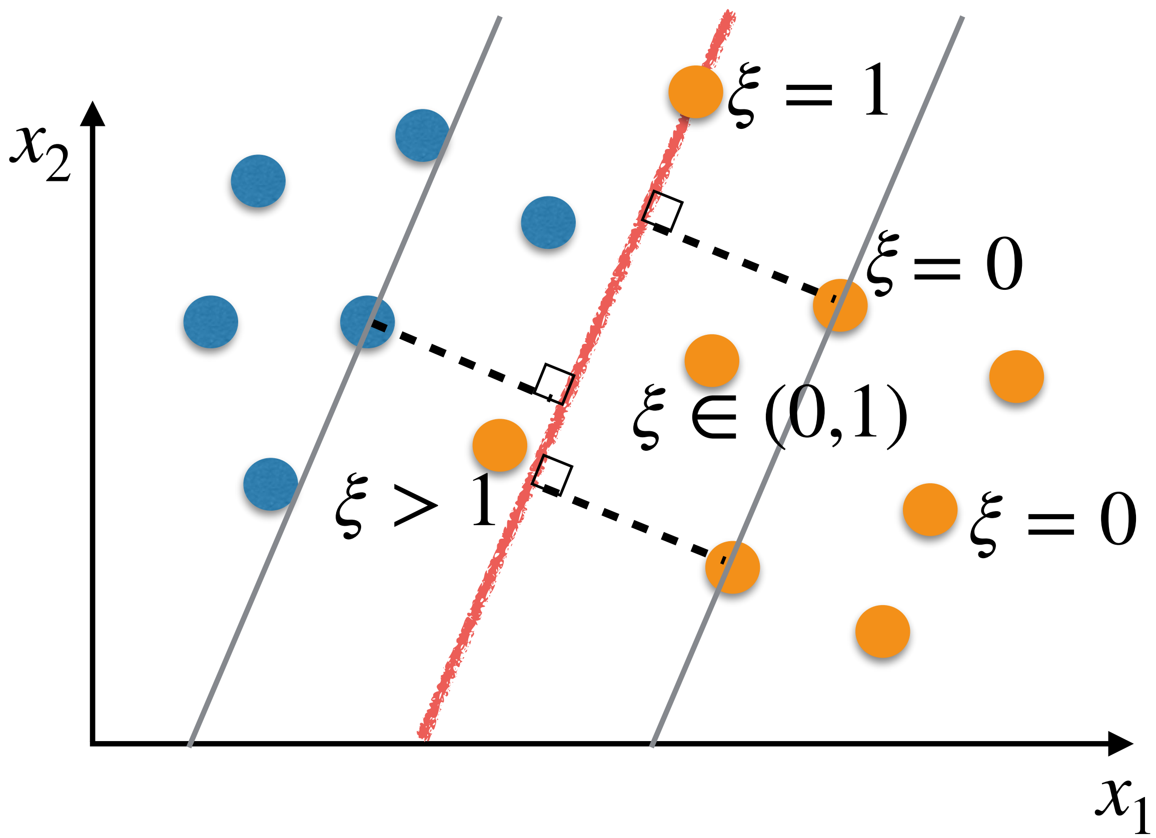
One slack variable $\xi^{(n)} \geq 0$ is associated to each training example $(\mathbf{x}^{(n)}, y^{(n)})$.



These variables tell us by how much an example can be within the margin or on the wrong side of the decision boundary.

$$y^{(n)} h(\mathbf{x}^{(n)}) \geq 1 - \xi^{(n)}$$

The Effect of ξ



$$y^{(n)}h(\mathbf{x}^{(n)}) \geq 1 - \xi^{(n)}$$

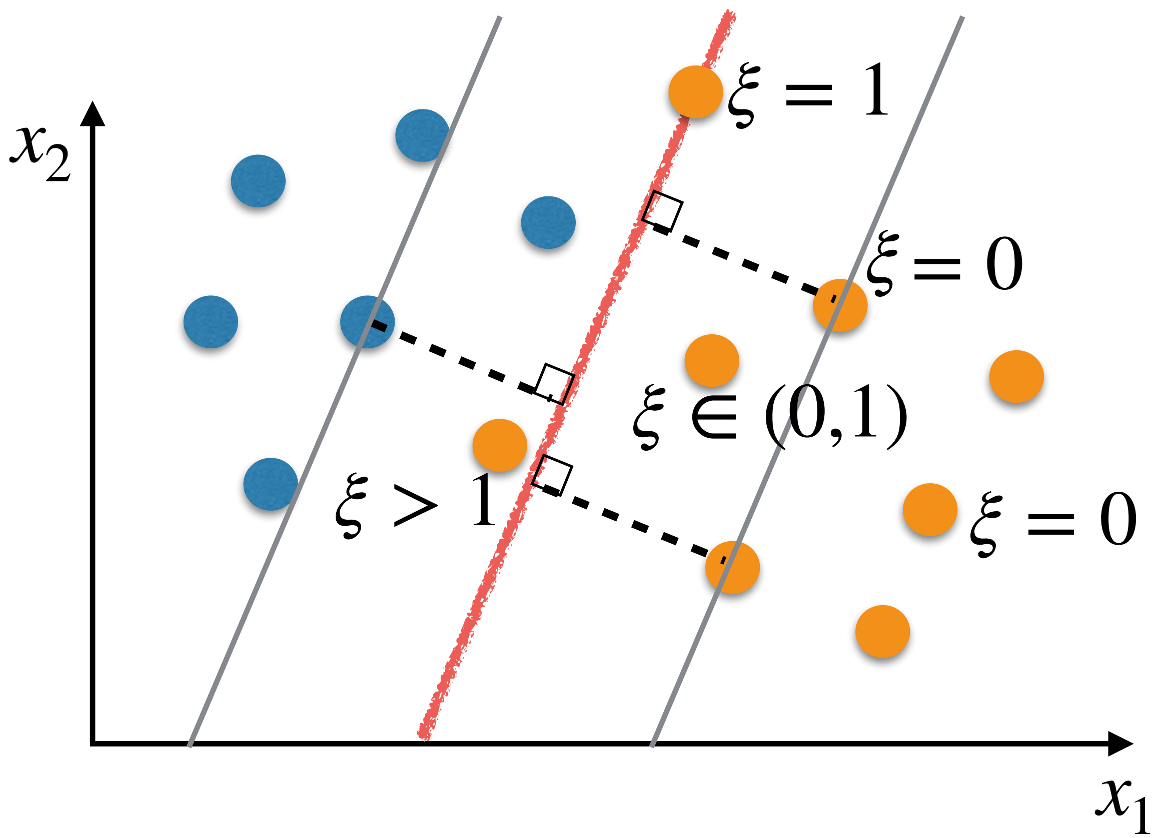
$$\xi^{(n)} = 0 \longrightarrow y^{(n)}h(\mathbf{x}^{(n)}) \geq 1$$

$$\xi^{(n)} \in (0,1) \longrightarrow y^{(n)}h(\mathbf{x}^{(n)}) \in (0,1)$$

$$\xi^{(n)} = 1 \longrightarrow y^{(n)}h(\mathbf{x}^{(n)}) \geq 0$$

$$\xi^{(n)} > 1 \longrightarrow y^{(n)}h(\mathbf{x}^{(n)}) < 0$$

Margin



Our margin was previously defined by

$$\text{dist}(h, \mathbf{x}^{(k)}) = \frac{y^{(k)} h(\mathbf{x}^{(k)})}{||\mathbf{w}||} = \frac{1}{||\mathbf{w}||}$$

where $(\mathbf{x}^{(k)}, y^{(k)})$ was the closest example to the decision boundary.

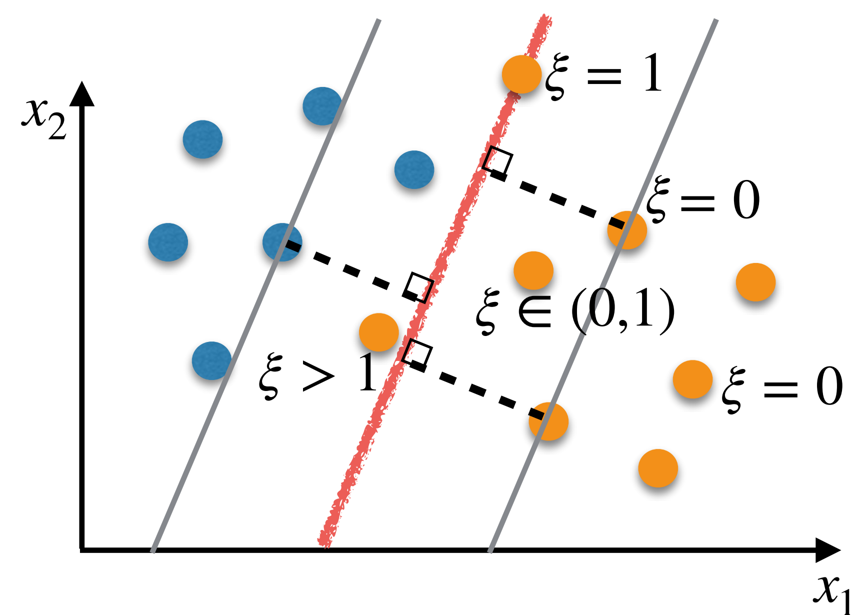
Now, our margin is simply defined as $1/||\mathbf{w}||$.

Our New Optimisation Problem

- Recap of our optimisation problem (primal representation):

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

Subject to: $y^{(n)} h(\mathbf{x}^{(n)}) \geq 1, \, \forall n \in \{1,2,\cdots,N\}$



- Using Slack

$$\underset{\mathbf{w},b,\xi}{\operatorname{argmin}} \left\{ \frac{1}{2} ||\mathbf{w}||^2 + C \sum_{n=1}^{N} \xi^{(n)} \right\}$$

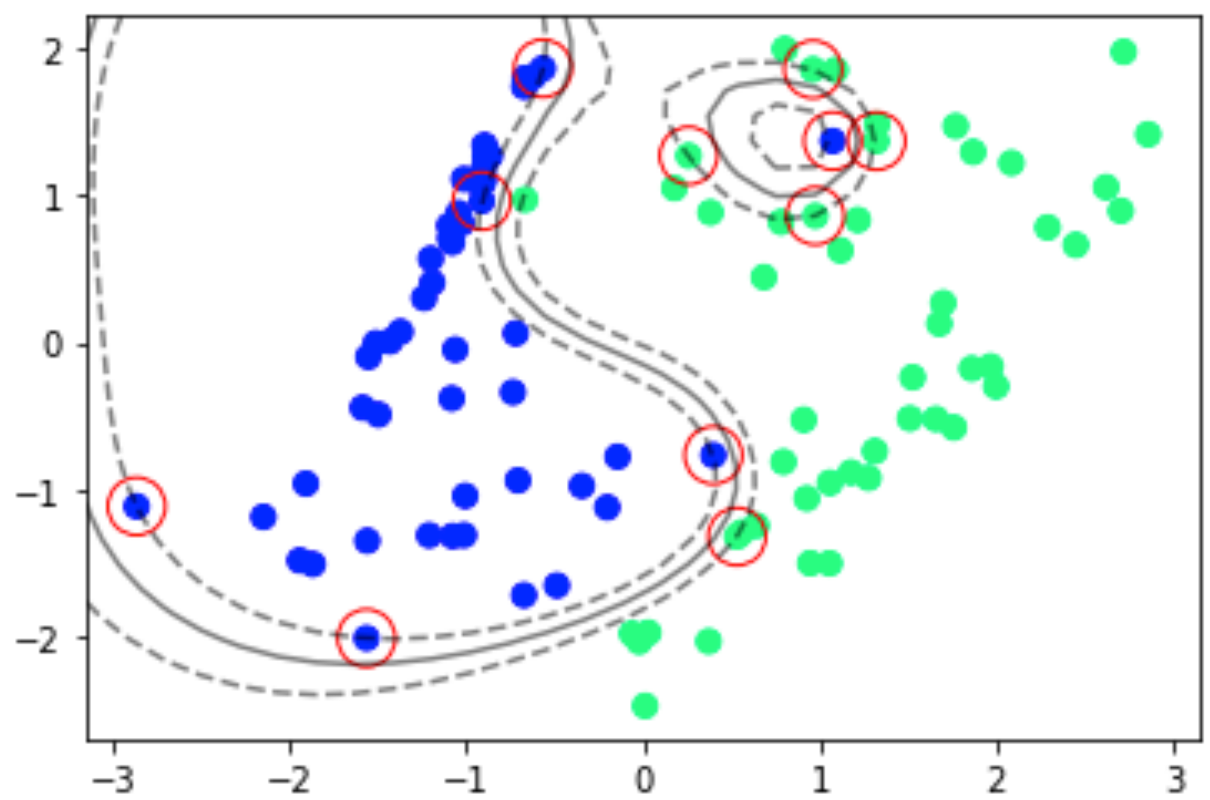
Subject to: $y^{(n)} h(\mathbf{x}^{(n)}) \geq 1 - \xi^{(n)}, \, \forall n \in \{1,2,\cdots,N\}$

$\xi^{(n)} \geq 0$

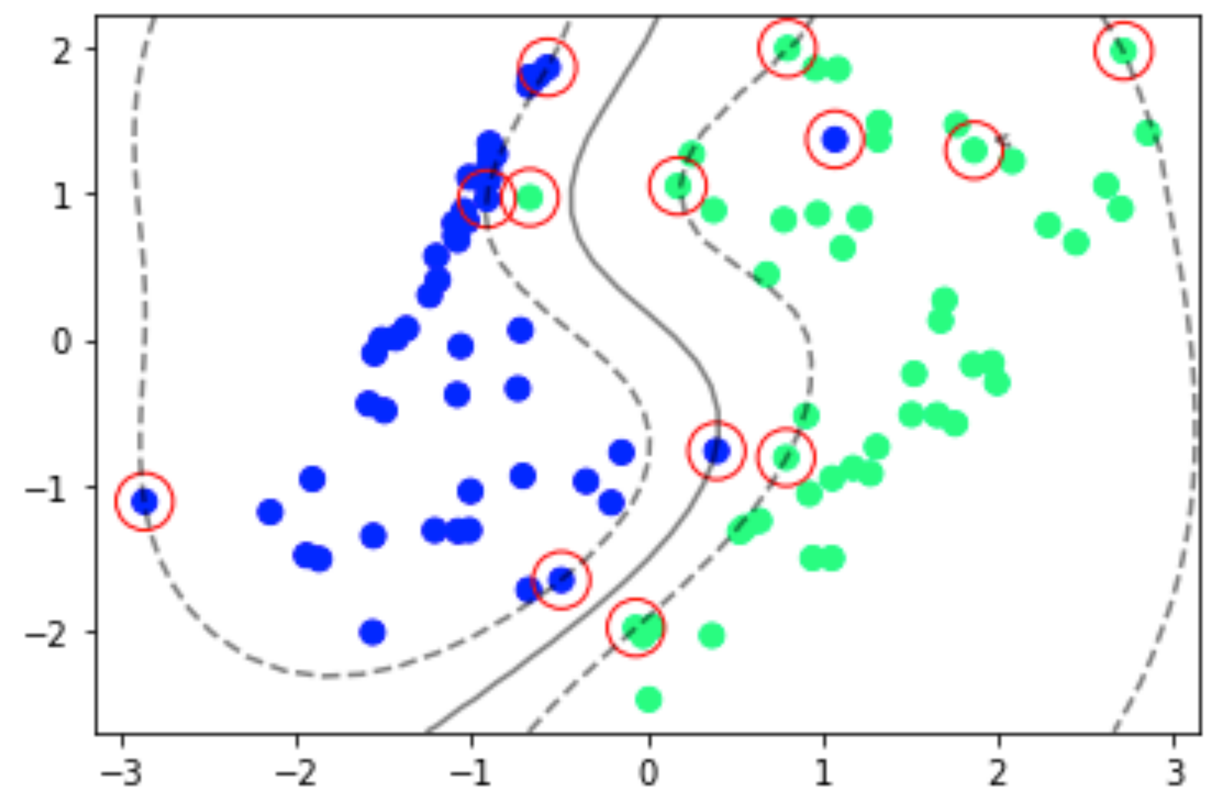
C is a hyperparameter that controls the trade-off between the slack variable penalty and the margin

Examples

Larger *C*



Smaller *C*



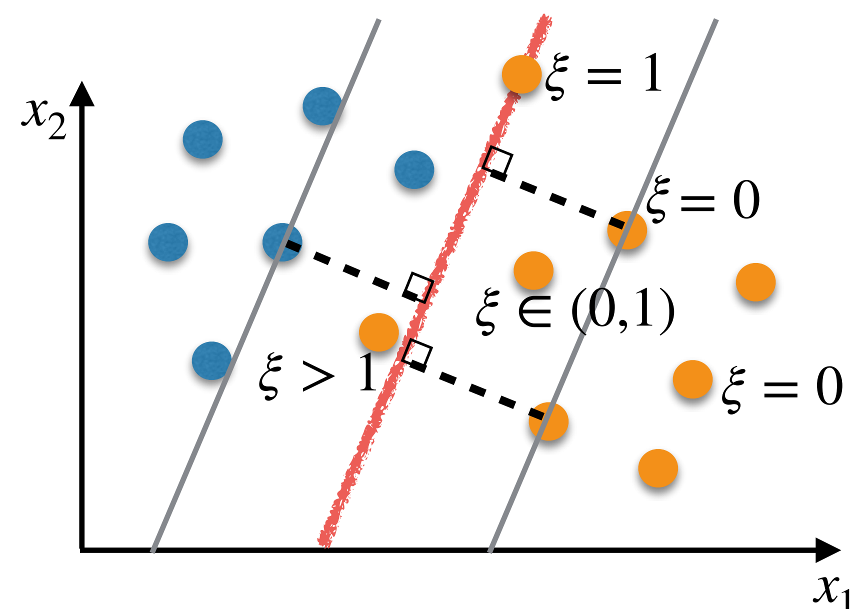
Code adapted from: https://jakevdp.github.io/PythonDataScienceHandbook/05.07-support-vector-machines.html

Soft Margin SVM

- Recap of our optimisation problem (primal representation):

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

Subject to: $y^{(n)} h(\mathbf{x}^{(n)}) \geq 1, \ \forall n \in \{1,2,\cdots,N\}$



- Using Slack

$$\underset{\mathbf{w},b,\xi}{\text{argmin}} \left\{ \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{n=1}^{N} \xi^{(n)} \right\}$$

Subject to: $y^{(n)} h(\mathbf{x}^{(n)}) \geq 1 - \xi^{(n)}, \ \forall n \in \{1,2,\cdots,N\}$

$\xi^{(n)} \geq 0$

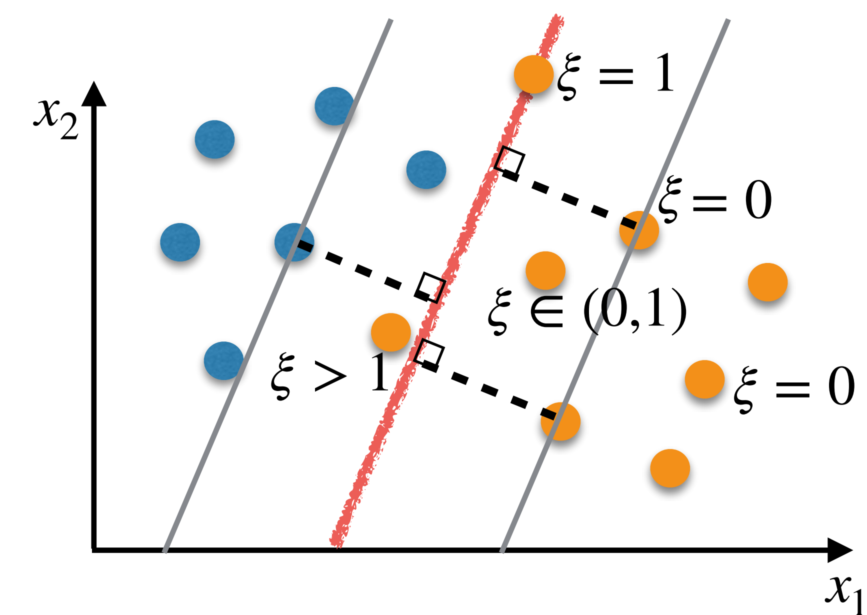
We will allow examples to be within the margin or in the wrong side of the decision boundary based on $\xi^{(n)}$

Soft Margin SVM

- Recap of our optimisation problem (primal representation):

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

Subject to: $y^{(n)} h(\mathbf{x}^{(n)}) \geq 1, \ \forall n \in \{1,2,\cdots,N\}$



- Using Slack

$$\underset{\mathbf{w},b,\xi}{\text{argmin}} \left\{ \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{n=1}^{N} \xi^{(n)} \right\}$$

Subject to: $y^{(n)} h(\mathbf{x}^{(n)}) \geq 1 - \xi^{(n)}, \ \forall n \in \{1,2,\cdots,N\}$

$$\xi^{(n)} \geq 0$$

When we allow slacks > 0, the margin is called a "**soft margin**", as opposed to a "**hard margin**".

Making Predictions

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

$\mathbf{w}^T\mathbf{x} + b > 0 \rightarrow$ class +1

$\mathbf{w}^T\mathbf{x} + b < 0 \rightarrow$ class -1

Dual Representation

- Recap of our optimisation problem:

$\text{argmax}_{\mathbf{a}} \tilde{L}(\mathbf{a})$

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

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

- Using Slack

Box constraints

$\text{argmax}_{\mathbf{a}} \tilde{L}(\mathbf{a})$

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

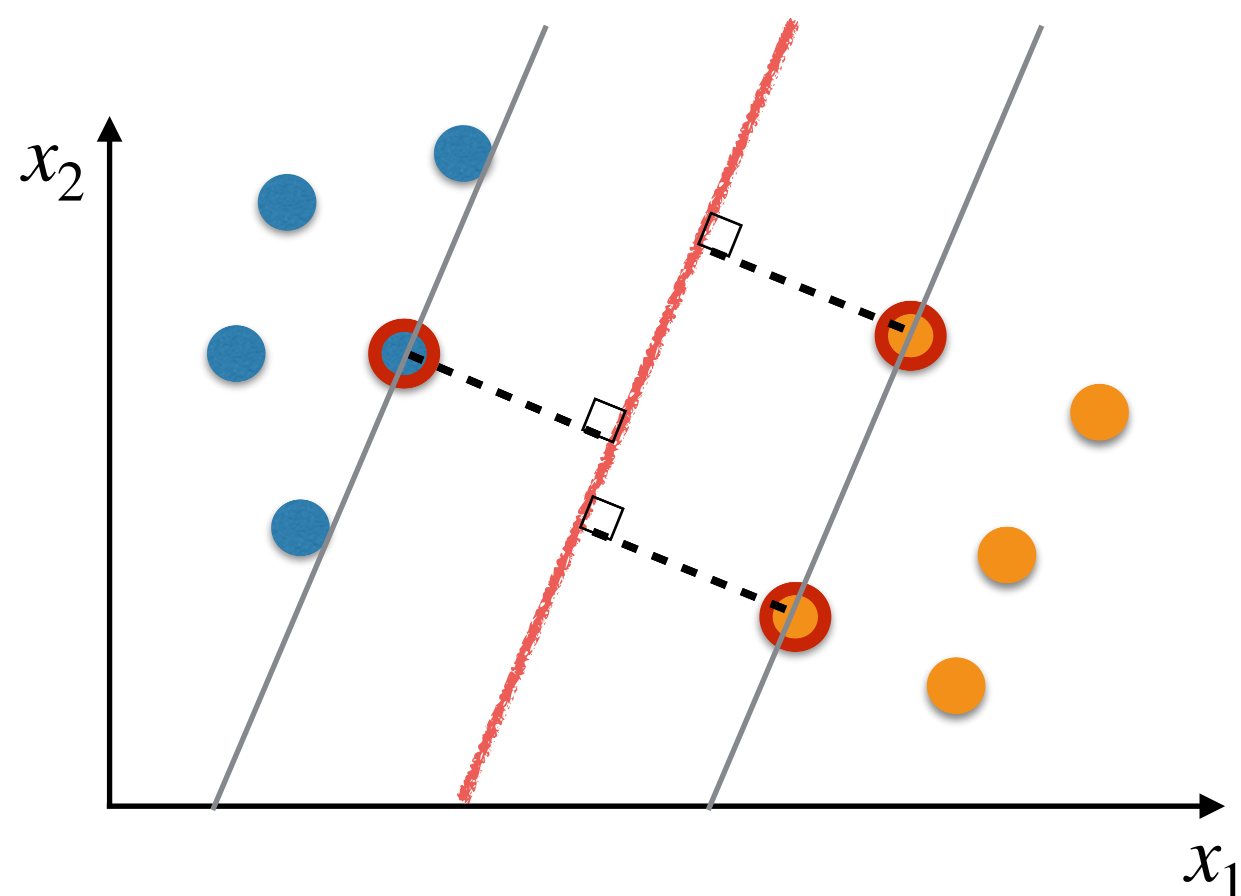
Subject to: $0 \leq a^{(n)} \leq C, \forall n \in \{1, \cdots, N\}$ $\quad \sum_{n=1}^{N} a^{(n)} y^{(n)} = 0$

Support Vectors and Making Predictions

- Before slack variables

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

All examples for which $a^{(n)} \neq 0$ (and $y^{(n)} h(\mathbf{x}^{(n)}) = 1$) are support vectors.

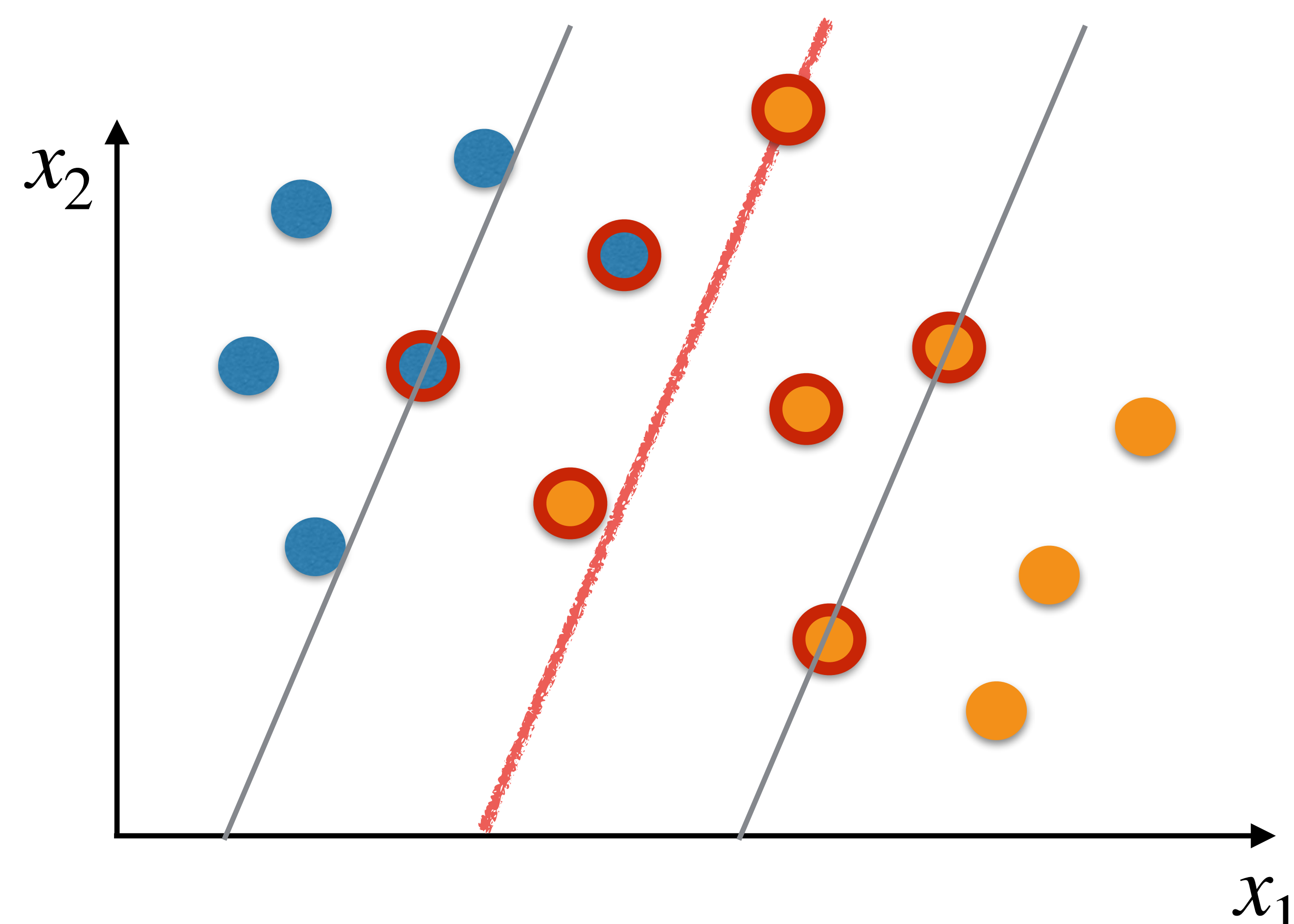


Support Vectors and Making Predictions

- With slack variables

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

All examples for which $a^{(n)} \neq 0$ (and $y^{(n)} h(\mathbf{x}^{(n)}) = 1 - \xi^{(n)}$) are support vectors.



Calculation of b

Our calculation of b was based on examples for which $y^{(n)}h(\mathbf{x}^{(n)}) = 1$.

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

where M is the set of indexes of the support vectors that are on the margin and N_M is the number of such support vectors.

Summary

- We've seen how to make predictions when using the dual representation.
- Soft margin SVM allows some training examples to be within the margin or misclassified.
- This can improve generalisation.
- We can use soft margin SVM both in the primal and dual format.