Machine Learning
CS342

Lecture 16: Support Vector Machines

Dr. Theo Damoulas
T.Damoulas@warwick.ac.uk

Office hours: Mon & Fri 10-11am @ CS 307

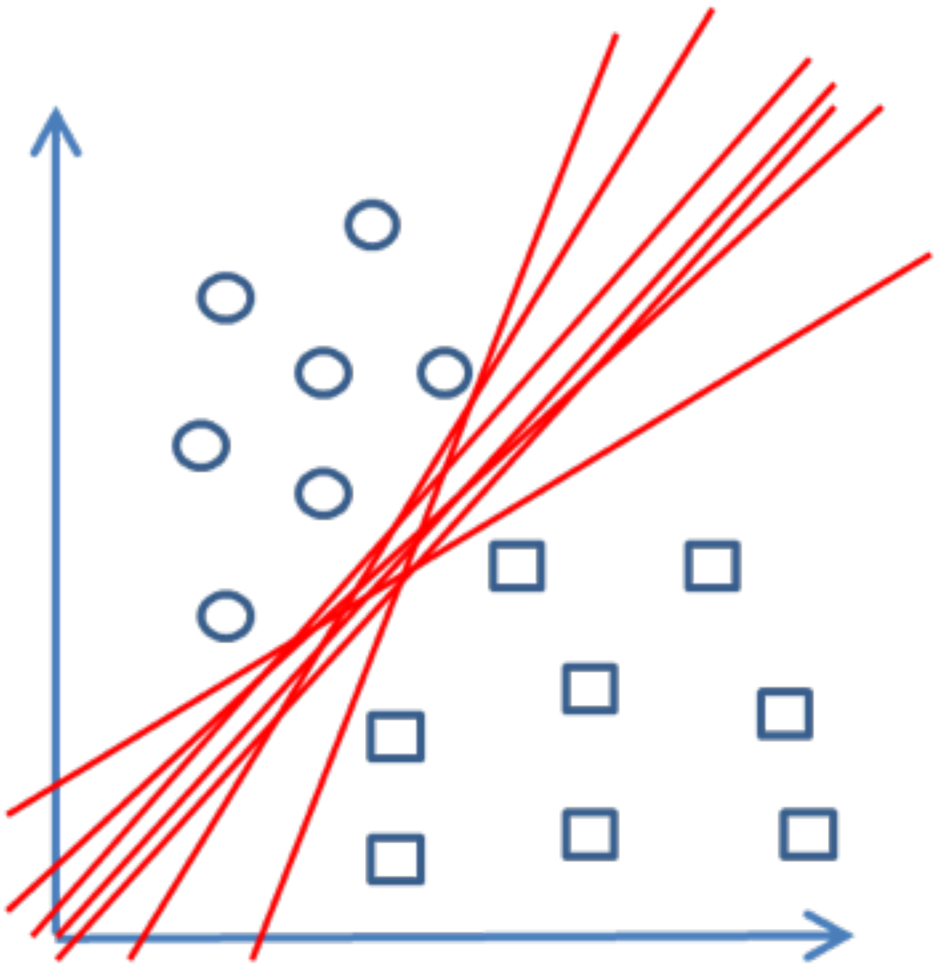
Support Vector Machines (SVMs)

Very popular framework
State of the art performance
Statistical Learning Theory
Kernel Machines

Binary Classification with a Linear Model (Hyper-plane)

Non-probabilistic models for **classification** (SVM) and regression (SVR)

Which discriminative line / decision boundary looks better?

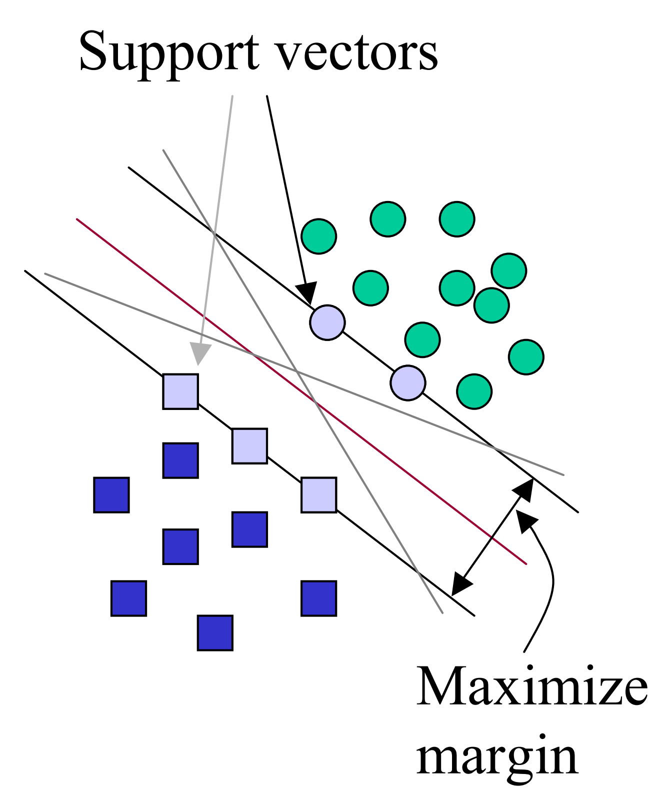


Support Vector Machine - High level view

So far we have seen optimisation-based models (point-estimators) that:

- Minimise a Loss function
- Minimise a Loss + Regulariser
- Maximise a Likelihood function (ML)
- Maximise a posterior density (MAP)

SVM: Find decision boundary that *maximises the margin*

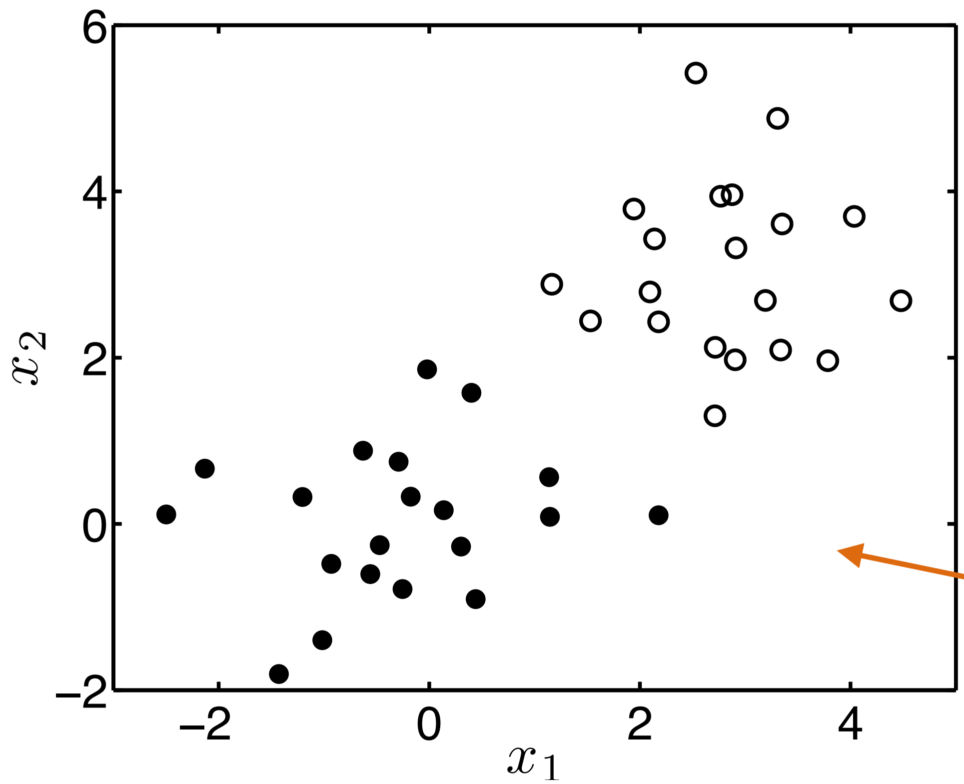


Maximum Margin Methods

SVM: Decision boundary

Rogers & Girolami, Ch. 5, section 5.3.2

Lets focus on a 2-D binary classification problem (2 attributes)



N training examples $\{\mathbf{x}_n, t_n\}_{n=1}^{N}$

2 Attributes $\mathbf{x}_n = [x_{n1} \quad x_{n2}]$

positive/negative class $t_n = \pm 1$

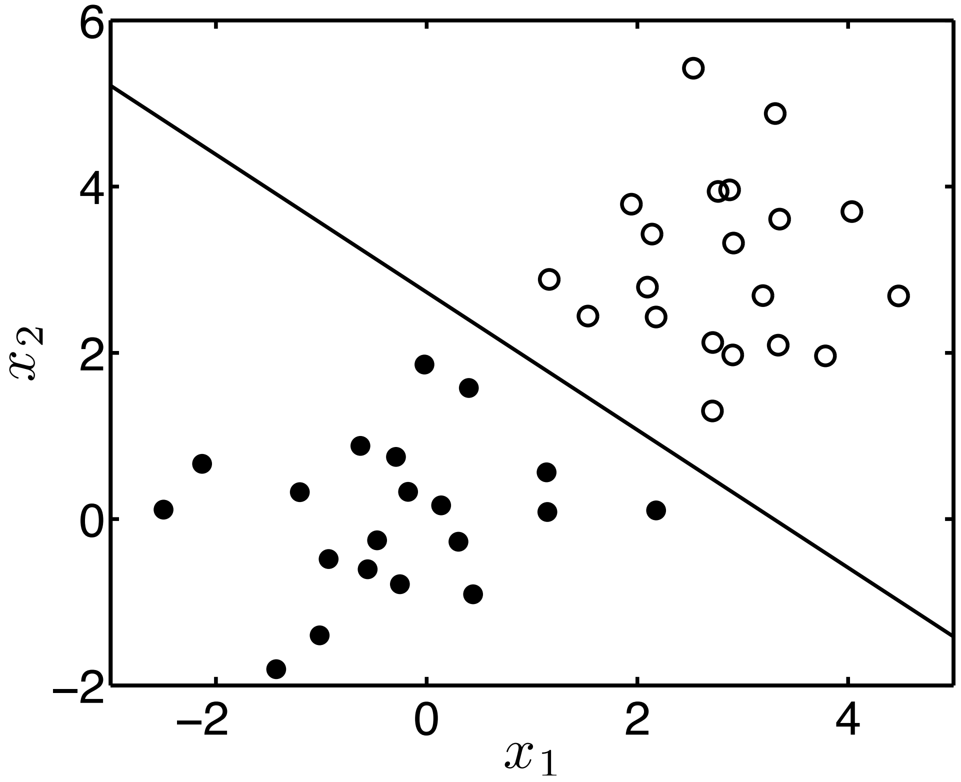
Linear decision boundary

$\mathbf{x}\mathbf{w} = 0$ remember depending on convention $\mathbf{w}^{\mathrm{T}}\mathbf{x} = 0$

Want to find **w** (I have included the bias/intercept b in the **w** notation)

Same thing as: $\mathbf{w}^{\mathrm{T}}\mathbf{x} + b = 0$

SVM: Decision boundary



$$\mathbf{w}^{\mathrm{T}}\mathbf{x} + b = 0$$

And given a new observation how do I classify?

$$\mathbf{w}^{\mathrm{T}}\mathbf{x}^* + b > 0 \quad \text{then} \quad t^* = +1$$

$$\mathbf{w}^{\mathrm{T}}\mathbf{x}^* + b < 0 \quad \text{then} \quad t^* = -1$$

This might remind you a bit the perceptron step activation function?

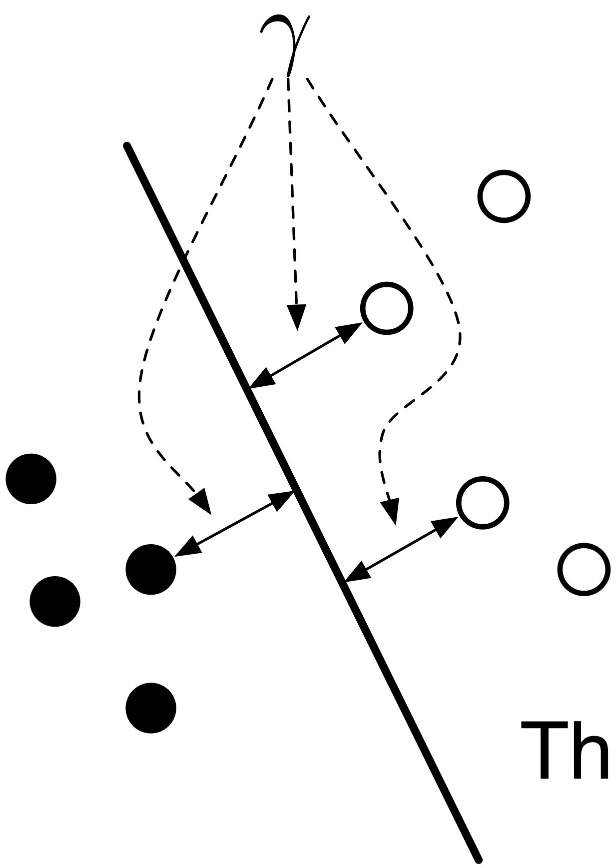
$$t^* = \mathrm{sign}(\mathbf{w}^{\mathrm{T}}\mathbf{x}^* + b)$$

Instead of minimising some loss function explicitly, we will maximise some other quantity: the margin!

SVM: The Margin

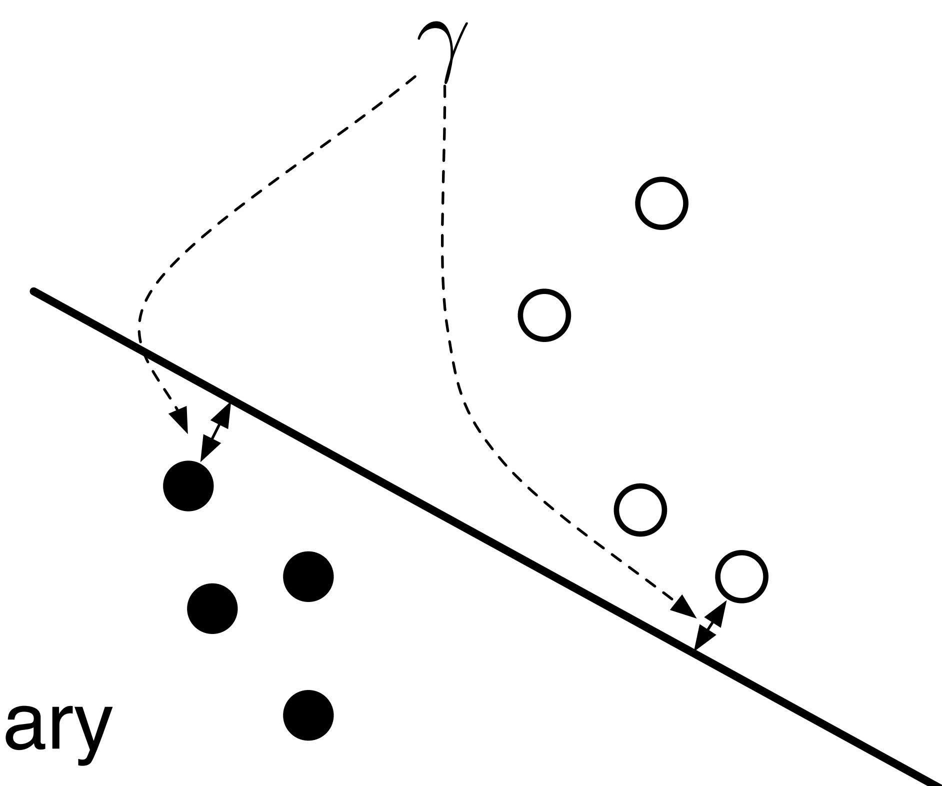
Margin: ***The perpendicular distance between the decision boundary and the closest points on each side***

We want to **maximise the margin**



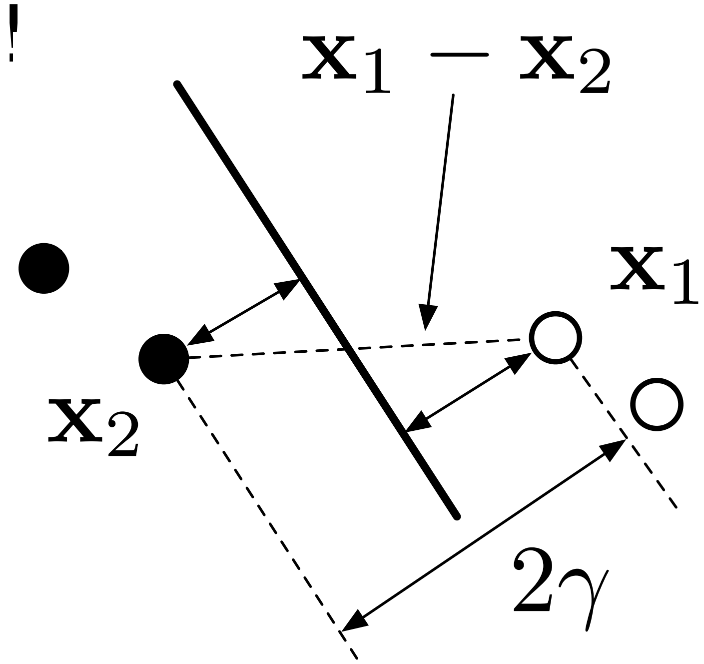
Why?
Which one is better?

There is one ‘best’ boundary



SVM: The Margin

Lets compute the margin!



(1) $$2\gamma = \frac{1}{||\mathbf{w}||}\mathbf{w}^{\mathrm{T}}(\mathbf{x}_1 - \mathbf{x}_2)$$

Fix the scale such that:

$$\mathbf{w}^{\mathrm{T}}\mathbf{x}_2 + b = -1 \qquad \mathbf{w}^{\mathrm{T}}\mathbf{x}_1 + b = +1$$

Subtracting them

$$(\mathbf{w}^{\mathrm{T}}\mathbf{x}_1 + b) - (\mathbf{w}^{\mathrm{T}}\mathbf{x}_2 + b) = 2$$

Leads to (2):

$$\mathbf{w}^{\mathrm{T}}(\mathbf{x}_1 - \mathbf{x}_2) = 2$$

Substituting (2) to (1): $\boxed{\gamma = \frac{1}{||\mathbf{w}||}}$

Another way to derive the margin

Given a line: $ax + by + c = 0$ And a point: (x_0, y_0)

The perpendicular distance of the point to the line is given by:

$$\text{distance} = \frac{|ax_0 + by_0 + c|}{\sqrt{a^2 + b^2}}$$

Our line is $\mathbf{w}^{\mathrm{T}}\mathbf{x} + b = 0$ A point $\mathbf{x}_n = [x_{n1} \quad x_{n2}]$

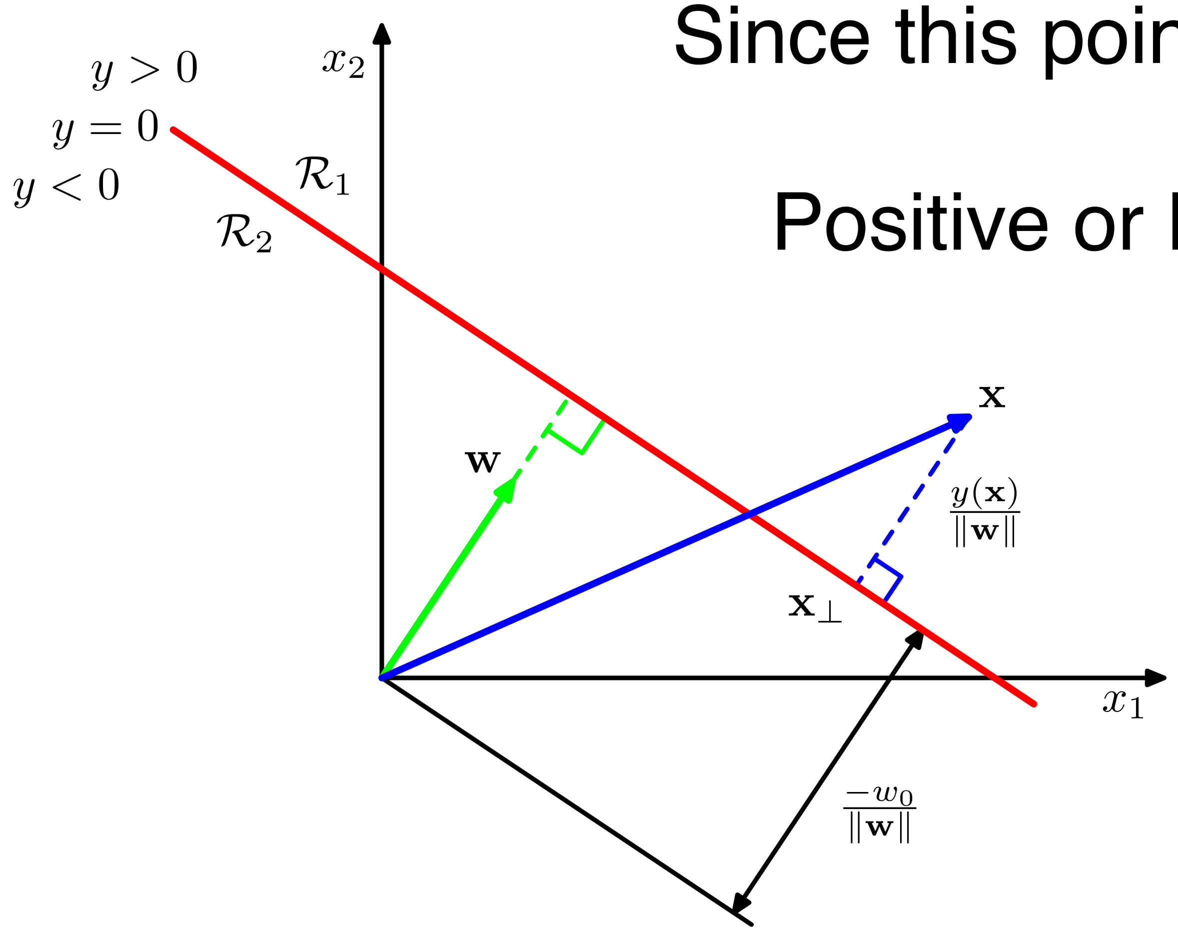
$$\text{distance} = \frac{|w_1 x_{n1} + w_2 x_{n2} + b|}{\sqrt{w_1^2 + w_2^2}} = \frac{|w_1 x_{n1} + w_2 x_{n2} + b|}{||\mathbf{w}||}$$

L2 norm

Department of COMPUTER SCIENCE

Calculating the margin again

$$\text{distance} = \frac{|w_1 x_{n1} + w_2 x_{n2} + b|}{\sqrt{w_1^2 + w_2^2}} = \frac{|w_1 x_{n1} + w_2 x_{n2} + b|}{||\mathbf{w}||}$$



Since this point is not on the line it wont evaluate to 0

Positive or Negative and I can rescale to be +1/-1

So the margin will be

$$\boxed{\gamma = \frac{1}{||\mathbf{w}||}}$$

SVM: Maximising the margin

We want to maximise the margin $\gamma = \dfrac{1}{||\mathbf{w}||}$

Equivalent to minimising the L2 norm $||\mathbf{w}||$

Which in turn is equivalent to minimising $\dfrac{1}{2}||\mathbf{w}||^2 = \dfrac{1}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w}$

Subject to some constraints:

if $t_n = 1$ then $\mathbf{w}^{\mathrm{T}}\mathbf{x}_n + b \geq 1$

if $t_n = -1$ then $\mathbf{w}^{\mathrm{T}}\mathbf{x}_n + b \leq -1$

So maximise margin s.t. correct class assignment

SVM: Optimisation problem

$$\boxed{\underset{\mathbf{w}}{\operatorname{argmin}} \ \frac{1}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w}}$$

subject to the following constraint: $\boxed{t_n(\mathbf{w}^{\mathrm{T}}\mathbf{x}_n + b) \geq 1}$

(Thats why we used +/-1) a compact way of combining these:

$t_n = 1$ then $\mathbf{w}^{\mathrm{T}}\mathbf{x}_n + b \geq 1$

$t_n = -1$ then $\mathbf{w}^{\mathrm{T}}\mathbf{x}_n + b \leq -1$

Leads to a standard Quadratic Programming optimisation problem

unique global minimum so nice guarantees!

e.g. Python: QP solvers at CVXOPT

SVM: Optimisation problem

Introduce Lagrange multipliers: Combine minimisation with constraints

$$\boxed{\underset{\mathbf{w}}{\operatorname{argmin}} \ \frac{1}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w}}$$ subject to $$\boxed{t_n(\mathbf{w}^{\mathrm{T}}\mathbf{x}_n + b) \geq 1}$$

becomes:

Primal $$\underset{\mathbf{w}}{\operatorname{argmin}} \frac{1}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w} - \sum_{n=1}^{N} \alpha_n (t_n(\mathbf{w}^{\mathrm{T}}\mathbf{x}_n + b) - 1)$$

$$\text{subject to}: \quad \alpha_n \geq 0$$

Some analogies to Ridge regression?

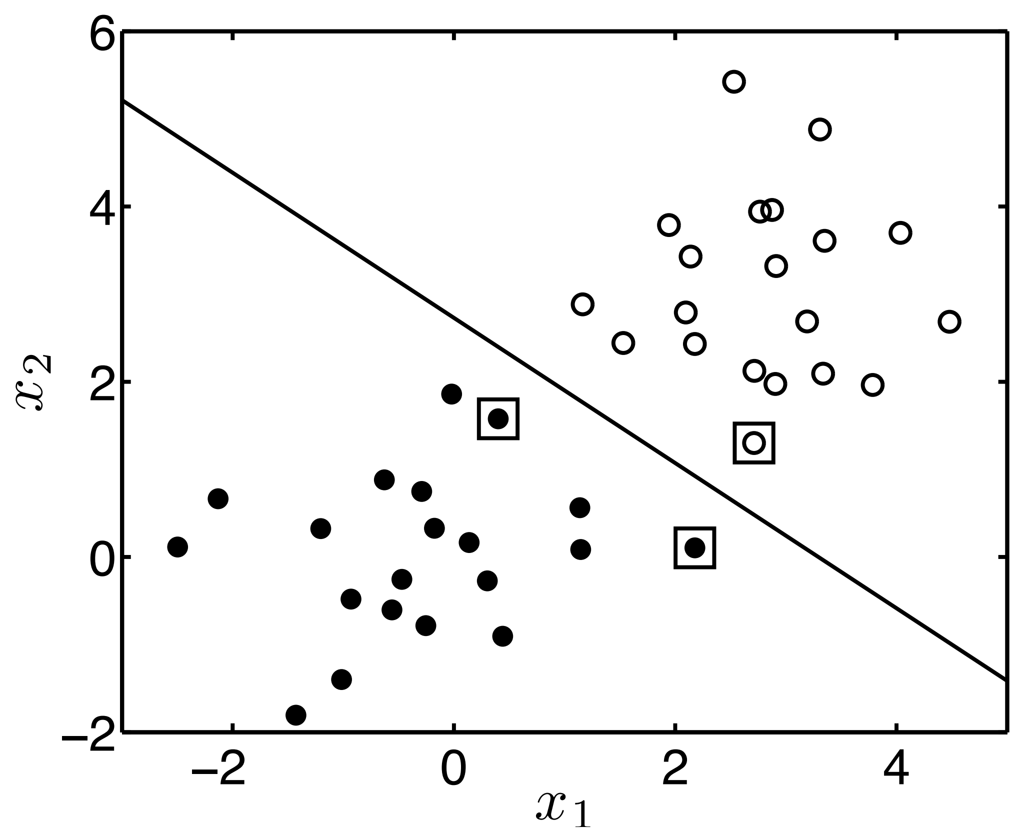
SVM: Support Vectors

We will revisit/rewrite the optimisation problem (dual) on friday

see p. 189-190 in Rogers & Girolami

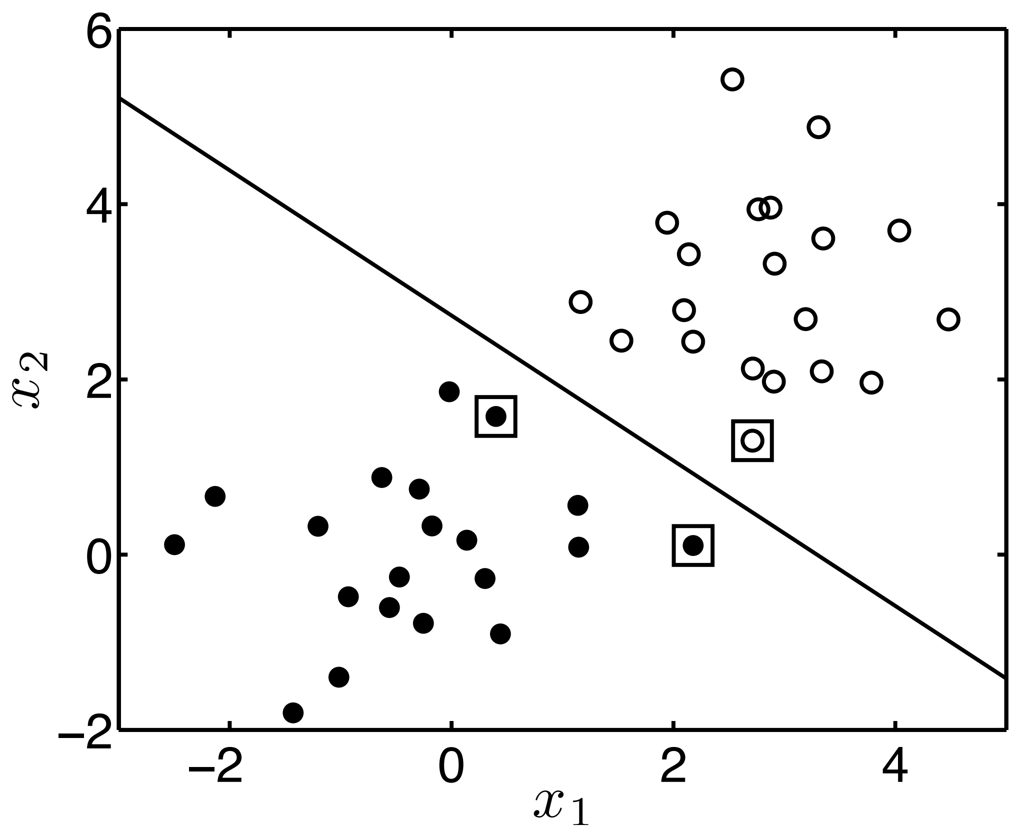
As a result of re-formulating the optimisation problem (called the dual OPT) only few observations become/remain important:

The ones that define/support the decision boundary: **Support Vectors**



Max the Margin
which in turn defined by SVs

SVM: Support Vectors



Predictions only depend on these points!

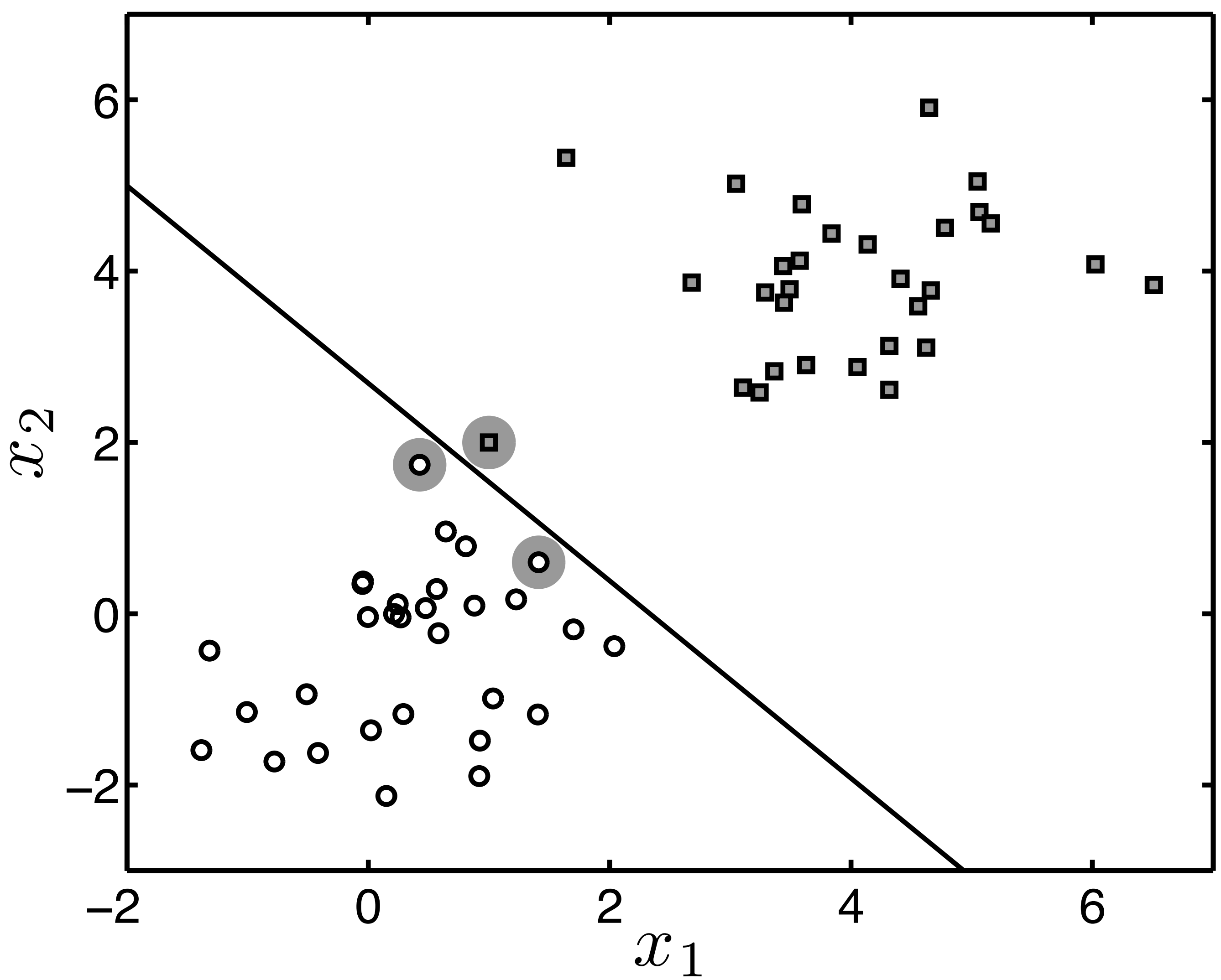
We only need these SVs to define the decision boundary

Sparse solution!
Sparsity in terms of observations

Very fast prediction times since small number of vectors/observations retained and utilised for predictions

Is sparseness always good?

SVM: Support Vectors



Not always!

This happens because of our constraints!

$$t_n(\mathbf{w}^{\mathrm{T}}\mathbf{x}_n + b) \geq 1$$

All points must be correctly classified! This is a **Hard margin**

Might remind you of the Perceptron with step Heaviside function linearly separable else not converging

SVM: Soft margins

To allow for miss-classifications (non-linearly separable problems)

we relax the constraints from:

$$t_n(\mathbf{w}^{\mathrm{T}}\mathbf{x}_n + b) \geq 1$$

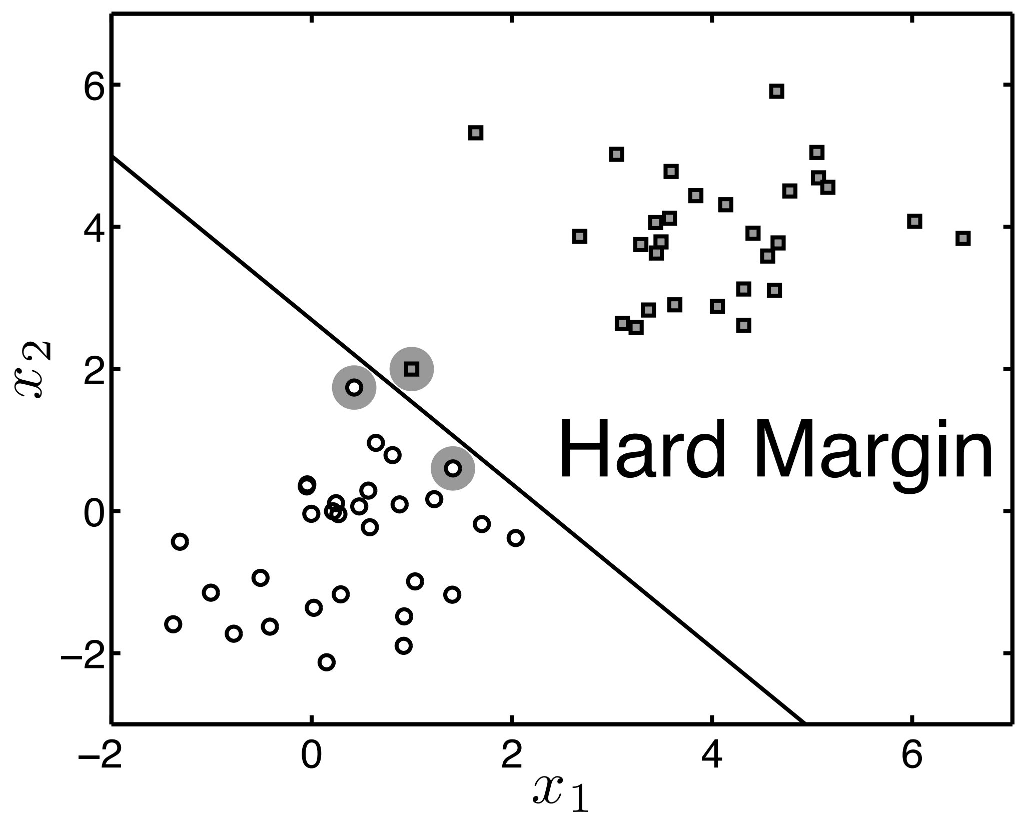
to:

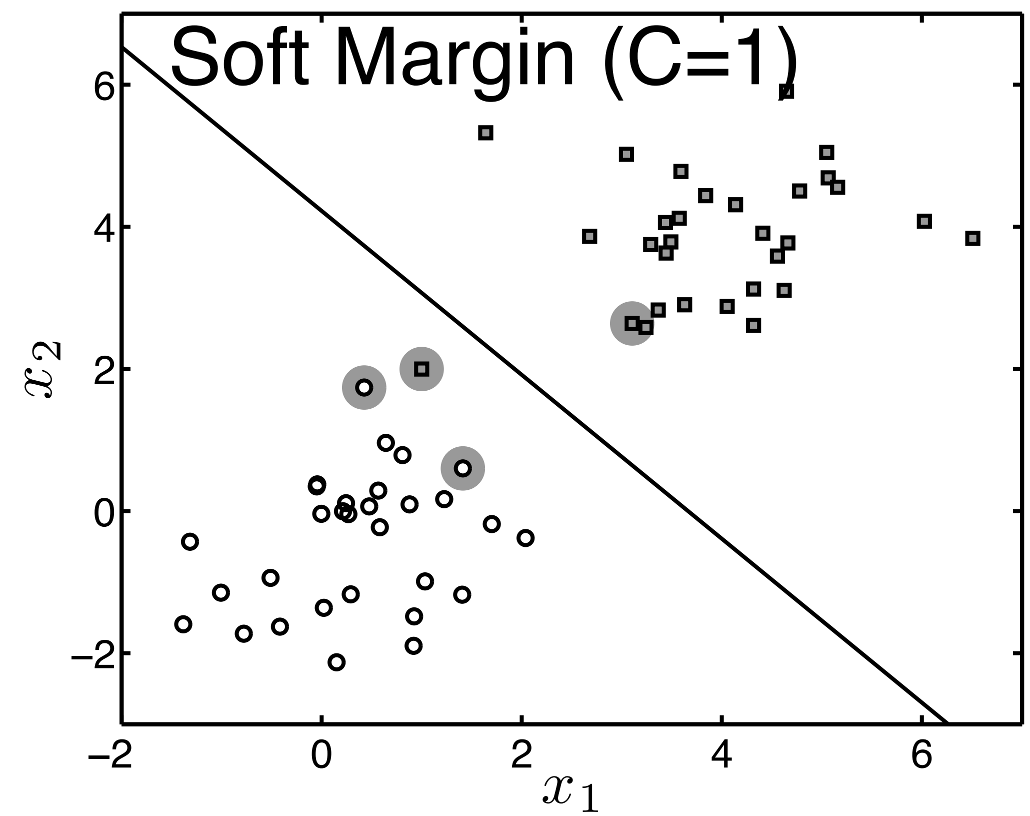
$$t_n(\mathbf{w}^{\mathrm{T}}\mathbf{x}_n + b) \geq 1 - \xi_n, \quad \xi_n \geq 0$$

These are called ***slack variables*** and we have a controlling parameter C

C is a parameter that controls to what extend
are we willing to allow points to sit within the margin or wrong side of DB

SVM: Soft margins





We now have an extra SV
And a better decision boundary!

C is one of the parameters you will need to set via Cross-Validation!

- C too high and we overfit to noise
- C too low and we underfit and loose sparsity

Support Vector Machines

- Max Margin classifiers
- We talked about the Linear SVM today and the Primal optimisation
- Margin as the objective to maximise subject to constraints
- Quadratic Optimisation problem easily solved by standard QP solvers
- Unique global solution
- Can relax the “Hard Margin” constraints by including slack variables
- “Soft Margin” can give better DBs and non-linearly separable problems
- C controls amount of slack - high C overfit, low C underfit. Use CV
- Support vectors: few observations that support/define the DB
- Sparse solutions because of few Support Vectors needed
- Very fast prediction times due to this sparsity

We will see the *dual optimisation problem* next time as we move to Kernels