

Lecture 10. Support Vector Machines (cont.)

COMP90051 Statistical Machine Learning

Semester 2, 2017
Lecturer: Andrey Kan

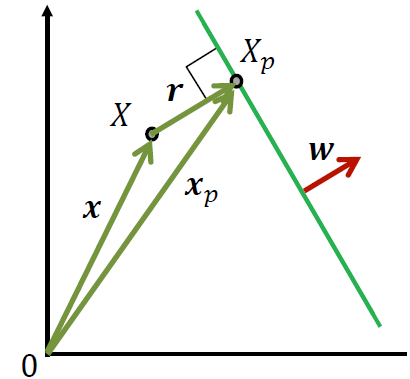
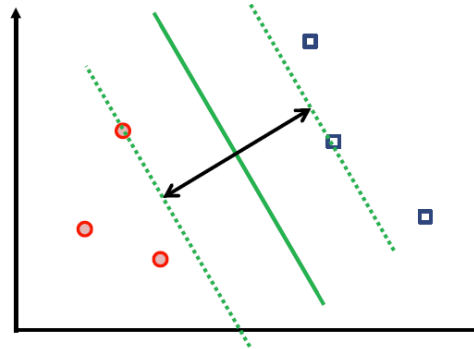
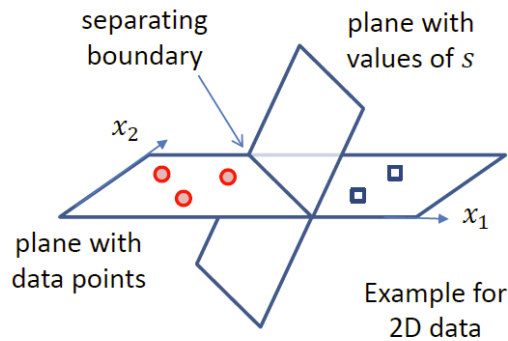


THE UNIVERSITY OF
MELBOURNE

This lecture

- Soft margin SVM
 - * Intuition and problem formulation
- Solving the optimisation
 - * Transforming the original objective
 - * Re-parameterisation
- Finishing touches
 - * Complementary slackness
 - * Solving the dual problem

Brief recap: hard margin SVM



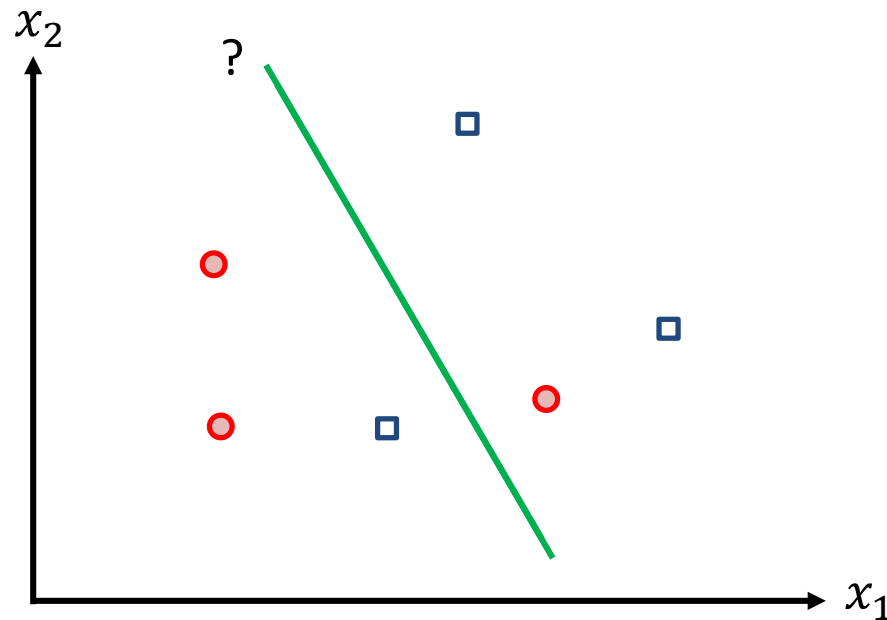
- SVM is a linear binary classifier
- Informally: aims for the safest boundary
- Formally: derive distance from point to boundary
- Trick to resolve ambiguity
- Hard margin SVM objective: $\operatorname{argmin}_{\mathbf{w}} \|\mathbf{w}\|$

$$\frac{y_{i^*}(\mathbf{w}'x_{i^*} + b)}{\|\mathbf{w}\|} = \frac{1}{\|\mathbf{w}\|}$$

$$\text{s.t. } y_i(\mathbf{w}'x_i + b) \geq 1 \text{ for } i = 1, \dots, n$$

When data is not linearly separable

- Hard margin loss is too stringent
- Real data is unlikely to be linearly separable
- If the data is not separable, hard margin SVMs are in trouble



SVMs offer 3 approaches to address this problem:

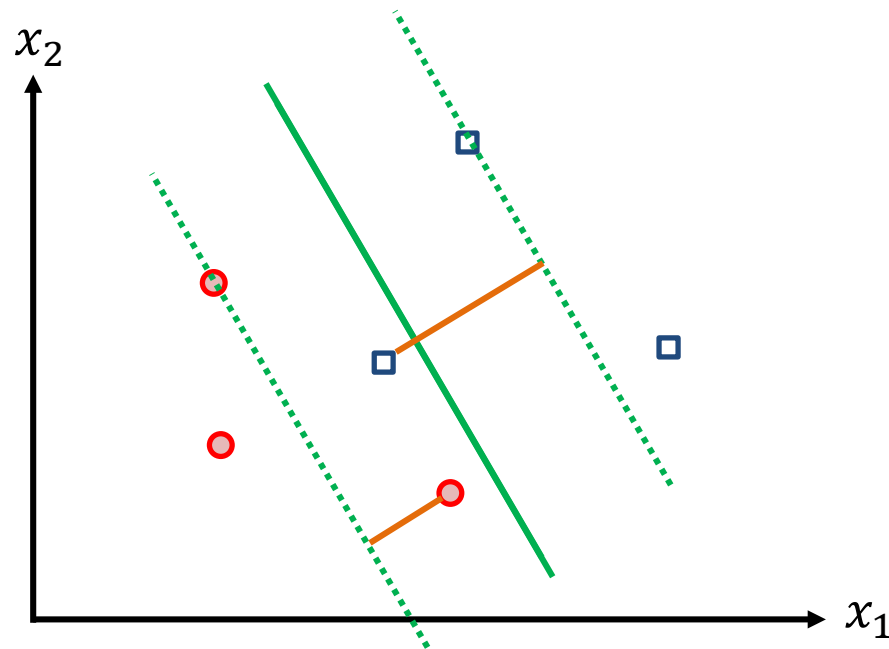
1. *Still use hard margin SVM, but transform the data (next lecture)*
2. *Relax the constraints (next slide)*
3. *The combination of 1 and 2 😊*

Addressing Non-Linearity using Soft Margin SVMs

We now do not assume that the
data is linearly separable

Soft margin SVM

- In the soft margin SVM formulation we relax the constraints to allow points to be inside the margin or even on the wrong side of the boundary



However, we penalise boundaries by the amount that reflects the extend of “violation”

In the figure, the objective penalty will take into account the orange distances

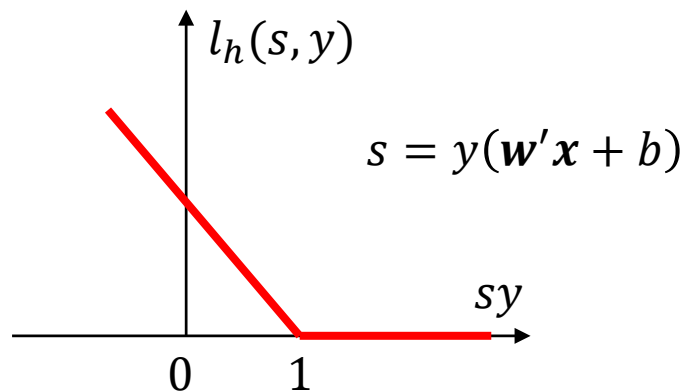
Hinge loss: soft margin SVM loss

- Hard margin SVM loss

$$l_{\infty} = \begin{cases} 0 & 1 - y(\mathbf{w}'\mathbf{x} + b) \leq 0 \\ \infty & \textit{otherwise} \end{cases}$$

- Soft margin SVM loss (hinge loss)

$$l_h = \begin{cases} 0 & 1 - y(\mathbf{w}'\mathbf{x} + b) \leq 0 \\ 1 - y(\mathbf{w}'\mathbf{x} + b) & \textit{otherwise} \end{cases}$$



compare this with
perceptron loss

Soft margin SVM objective

- Soft margin SVM loss (hinge loss)

$$l_h = \begin{cases} 0 & 1 - y(\mathbf{w}'\mathbf{x} + b) \leq 0 \\ 1 - y(\mathbf{w}'\mathbf{x} + b) & \textit{otherwise} \end{cases}$$

- By analogy with ridge regression, soft margin SVM objective can be defined as

$$\operatorname{argmin}_{\mathbf{w}} \left(\sum_{i=1}^n l_h(\mathbf{x}_i, y_i, \mathbf{w}) + \lambda \|\mathbf{w}\|^2 \right)$$

- We are going to re-formulate this objective to make it more amenable to analysis

Re-formulating soft margin objective

- Soft margin SVM loss (hinge loss)

$$l_h = \max(0, 1 - y_i(\mathbf{w}'\mathbf{x}_i + b))$$

- Define slack variables as an upper bound on loss

$$\xi_i \geq l_h = \max(0, 1 - y_i(\mathbf{w}'\mathbf{x}_i + b))$$

or equivalently $\xi_i \geq 1 - y_i(\mathbf{w}'\mathbf{x}_i + b)$ and $\xi_i \geq 0$

- Re-write the soft margin SVM objective as:

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

s.t. $\xi_i \geq 1 - y_i(\mathbf{w}'\mathbf{x}_i + b)$ for $i = 1, \dots, n$

$$\xi_i \geq 0 \text{ for } i = 1, \dots, n$$

Two variations of SVM

- Hard margin SVM objective*:

$$\operatorname{argmin}_{\mathbf{w}} \frac{1}{2} \|\mathbf{w}\|^2$$

$$\text{s.t. } y_i(\mathbf{w}'\mathbf{x}_i + b) \geq 1 \text{ for } i = 1, \dots, n$$

- Soft margin SVM objective:

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

$$\text{s.t. } y_i(\mathbf{w}'\mathbf{x}_i + b) \geq 1 - \xi_i \text{ for } i = 1, \dots, n$$

$$\xi_i \geq 0 \text{ for } i = 1, \dots, n$$

- In the second case, the constraints are relaxed (“softened”) by allowing violations by ξ_i . Hence the name “soft margin”

*Changed $\|\mathbf{w}\|$ to $0.5\|\mathbf{w}\|^2$. The modified objective leads to the same solution

The Mathy Part 2

Optimisation strikes back

SVM training preliminaries

- Training an SVM means solving the corresponding optimisation problem, either hard margin or soft margin
- We will focus on solving the hard margin SVM (simpler)
 - * Soft margin SVM training results in a similar solution
- Hard margin SVM objective is a constrained optimisation problem. This is called the *primal problem*.

$$\operatorname{argmin}_{\mathbf{w}} \frac{1}{2} \|\mathbf{w}\|^2$$

$$\text{s.t. } y_i(\mathbf{w}'\mathbf{x}_i + b) - 1 \geq 0 \text{ for } i = 1, \dots, n$$

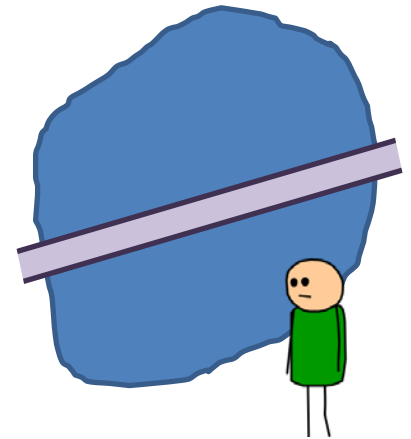
Side note: Constrained optimisation

- In general, a constraint optimisation problem is

minimise $f(\mathbf{x})$

s.t. $g_i(\mathbf{x}) \leq 0, i = 1, \dots, n$

s.t. $h_j(\mathbf{x}) = 0, j = 1, \dots, m$



- E.g., find deepest point in the lake, *south of the bridge*
- Big deal: common optimisation methods, such as gradient descent, cannot be directly applied to solve constrained optimisation
- Method of Lagrange/Karush-Kuhn-Tucker (KKT) multipliers
 - * Transform the original (primal) problem into an unconstrained optimisation problem (dual)
 - * Analyse/relate necessary and sufficient conditions for solutions for both problems

Side note: Lagrangian/KKT multipliers

- Introduce extra variables and define an auxiliary objective function

$$L(\mathbf{x}, \boldsymbol{\lambda}, \mathbf{v}) = f(\mathbf{x}) + \sum_{i=1}^n \lambda_i g_i(\mathbf{x}) + \sum_{j=1}^m v_j h_j(\mathbf{x})$$

- * Auxiliary function $L(\mathbf{x}, \boldsymbol{\lambda}, \mathbf{v})$ is called *Lagrangian*, and also sometimes it is called *KKT objective*
 - * Additional variables $\boldsymbol{\lambda}$ and \mathbf{v} are called *Lagrange multipliers* ($\boldsymbol{\lambda}$ are also called *KKT multipliers*)
- This is accompanied by the list of KKT conditions
 - * Next slides will show KKT conditions for SVM
 - Under mild assumptions on $f(\mathbf{x})$, $g_i(\mathbf{x})$ and $h_i(\mathbf{x})$, KKT conditions are both necessary and sufficient for \mathbf{x}^* , $\boldsymbol{\lambda}^*$ and \mathbf{v}^* being primal and dual optimal points
 - * The assumptions about the problem are called regularity conditions or constraint qualifications

Lagrangian for hard margin SVM

- Hard margin SVM objective is a constrained optimisation problem:

$$\operatorname{argmin}_{\mathbf{w}} \frac{1}{2} \|\mathbf{w}\|^2$$

$$\text{s.t. } y_i(\mathbf{w}'\mathbf{x}_i + b) - 1 \geq 0 \text{ for } i = 1, \dots, n$$

- We approach this problem using the method of Lagrange multipliers/KKT conditions
- To this end, we first define the Lagrangian/KKT objective

$$L_{KKT}(\mathbf{w}, b, \boldsymbol{\lambda}) = \frac{1}{2} \|\mathbf{w}\|^2 - \sum_{i=1}^n \lambda_i (y_i(\mathbf{w}'\mathbf{x}_i + b) - 1)$$

primal objective

constraints

KKT conditions for hard margin SVM

- Our Lagrangian/KKT objective is

$$L_{KKT}(\mathbf{w}, b, \boldsymbol{\lambda}) = \frac{1}{2} \|\mathbf{w}\|^2 - \sum_{i=1}^n \lambda_i (y_i (\mathbf{w}' \mathbf{x}_i + b) - 1)$$

- The corresponding KKT conditions are:

- $y_i ((\mathbf{w}^*)' \mathbf{x}_i + b^*) - 1 \geq 0$ for $i = 1, \dots, n$

this just repeats
constraints, trivial

- $\lambda_i^* \geq 0$ for $i = 1, \dots, n$

require non-negative
multipliers in order for
the theorem to work

- $\lambda_i^* (y_i ((\mathbf{w}^*)' \mathbf{x}_i + b^*) - 1) = 0$

“complementary slackness”,
we’ll come back to that

- $\nabla_{\mathbf{w}, b} L_{KKT}(\mathbf{w}^*, b^*, \boldsymbol{\lambda}^*) = 0$

zero gradient, somewhat similar
to unconstrained optimisation

Why use KKT conditions

- Proposition:
- If \mathbf{w}^* and b^* is a solution of the primal hard margin SVM problem, then there exists λ^* , such that together \mathbf{w}^* , b^* and λ^* satisfy the KKT conditions
- If some \mathbf{w}^* , b^* and λ^* satisfy KKT conditions then \mathbf{w}^* and b^* is a solution of the primal problem
- Proof is outside the scope of this subject
 - * Verify that SVM primal problem satisfies certain regularity conditions

Gradient of Lagrangian

- Our Lagrangian/KKT objective is

$$L_{KKT}(\mathbf{w}, b, \boldsymbol{\lambda}) = \frac{1}{2} \|\mathbf{w}\|^2 - \sum_{i=1}^n \lambda_i (y_i (\mathbf{w}' \mathbf{x}_i + b) - 1)$$

- The following conditions are necessary:

$$\frac{\partial L_{KKT}}{\partial b} = \sum_{i=1}^n \lambda_i y_i = 0$$

$$\frac{\partial L_{KKT}}{\partial w_j} = w_j - \sum_{i=1}^n \lambda_i y_i (\mathbf{x}_i)_j = 0$$

- Substitute the conditions into Lagrangian to obtain

$$L_{KKT}(\boldsymbol{\lambda}) = \sum_{i=1}^n \lambda_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \lambda_i \lambda_j y_i y_j \mathbf{x}_i' \mathbf{x}_j$$

Re-parameterisation

- Let \mathbf{w}^* and b^* be a solution of the primal problem. From the last KKT condition (zero gradient) we have

$$L_{KKT}(\mathbf{w}^*, b^*, \boldsymbol{\lambda}) = L_{KKT}(\boldsymbol{\lambda}) = \sum_{i=1}^n \lambda_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \lambda_i \lambda_j y_i y_j \mathbf{x}'_i \mathbf{x}_j$$

- Parameters $\boldsymbol{\lambda}$ are still unknown
- In order for \mathbf{w}^* , b^* and $\boldsymbol{\lambda}^*$ to satisfy all KKT conditions, $\boldsymbol{\lambda}^*$ must maximise $L_{KKT}(\boldsymbol{\lambda})$
 - * Proof is outside the scope of the subject

Lagrangian dual for hard margin SVM

- Given the above considerations, in order to solve the primal problem, we pose a new optimisation problem, called *Lagrangian dual problem*

$$\operatorname{argmax}_{\lambda} \sum_{i=1}^n \lambda_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \lambda_i \lambda_j y_i y_j \mathbf{x}'_i \mathbf{x}_j$$

$$\text{s.t. } \lambda_i \geq 0 \text{ and } \sum_{i=1}^n \lambda_i y_i = 0$$

- This is a so-called *quadratic optimisation problem*, a standard problem that can be solved using off-the-shelf software

Hard margin SVM

- Training: finding λ that solve

$$\operatorname{argmax}_{\lambda} \sum_{i=1}^n \lambda_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \lambda_i \lambda_j y_i y_j \mathbf{x}'_i \mathbf{x}_j$$

$$\text{s.t. } \lambda_i \geq 0 \text{ and } \sum_{i=1}^n \lambda_i y_i = 0$$

- Making predictions: classify new instance \mathbf{x} based on the sign of


$$s = b^* + \sum_{i=1}^n \lambda_i^* y_i \mathbf{x}'_i \mathbf{x}$$

- Here b^* can be found by noting that for arbitrary training example j we must have $y_j (b^* + \sum_{i=1}^n \lambda_i^* y_i \mathbf{x}'_i \mathbf{x}_j) = 1$

Soft margin SVM

- Training: finding λ that solve

$$\underset{\lambda}{\operatorname{argmax}} \sum_{i=1}^n \lambda_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \lambda_i \lambda_j y_i y_j \mathbf{x}'_i \mathbf{x}_j$$

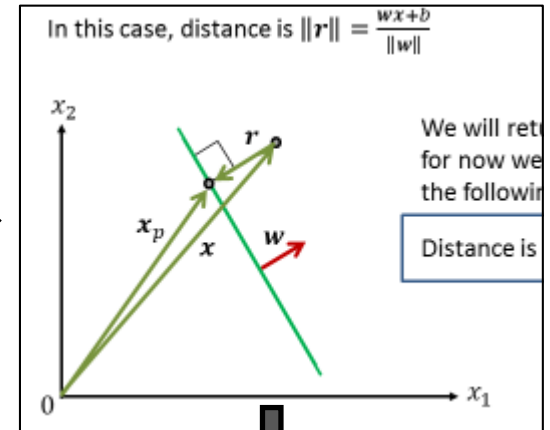
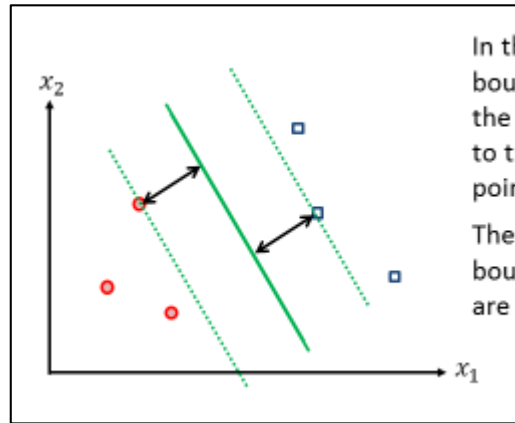
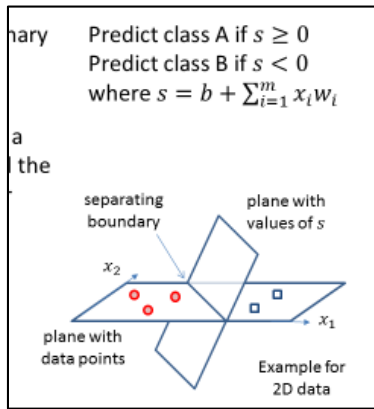
box constraints 

$$\text{s.t. } \boxed{C \geq \lambda_i \geq 0} \text{ and } \sum_{i=1}^n \lambda_i y_i = 0$$

- Making predictions: classify new instance \mathbf{x} based on the sign of

$$s = b^* + \sum_{i=1}^n \lambda_i^* y_i \mathbf{x}'_i \mathbf{x}$$

- Here b^* can be found by noting that for arbitrary training example j we must have $y_j (b^* + \sum_{i=1}^n \lambda_i^* y_i \mathbf{x}'_i \mathbf{x}_j) = 1$



- Training: finding λ that solve

$$\operatorname{argmax}_{\lambda} \sum_{i=1}^n \lambda_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \lambda_i \lambda_j y_i y_j x_i x_j$$
 s.t. $\lambda_i \geq 0$ and $\sum_{i=1}^n \lambda_i y_i = 0$
- Making predictions: classify new instance x based on sign of

$$s = b + \sum_{i=1}^n \lambda_i y_i x_i x$$

Hard margin SVM objective is a constrained optimisation problem:

$$\operatorname{argmin}_w \frac{1}{2} \|w\|^2$$

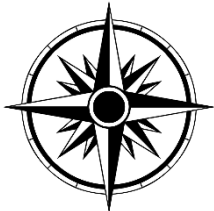
s.t. $y_i(w x_i + b) - 1 \geq 0$ for $i = 1, \dots, n$

- The corresponding KKT conditions are
- $y_i(w x_i + b) - 1 \geq 0$ for $i = 1, \dots, n$
- $\lambda_i \geq 0$ for $i = 1, \dots, n$
- $\lambda_i (y_i(w x_i + b) - 1) = 0$
- $\nabla_{w,b} L_{KKT}(w, b, \lambda) = 0$ (zero gradient to unconstrained)

Hard margin SVM Lagrangian dual problem is

$$\operatorname{argmax}_{\lambda} \sum_{i=1}^n \lambda_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \lambda_i \lambda_j y_i y_j x_i x_j$$

s.t. $\lambda_i \geq 0$ and $\sum_{i=1}^n \lambda_i y_i = 0$



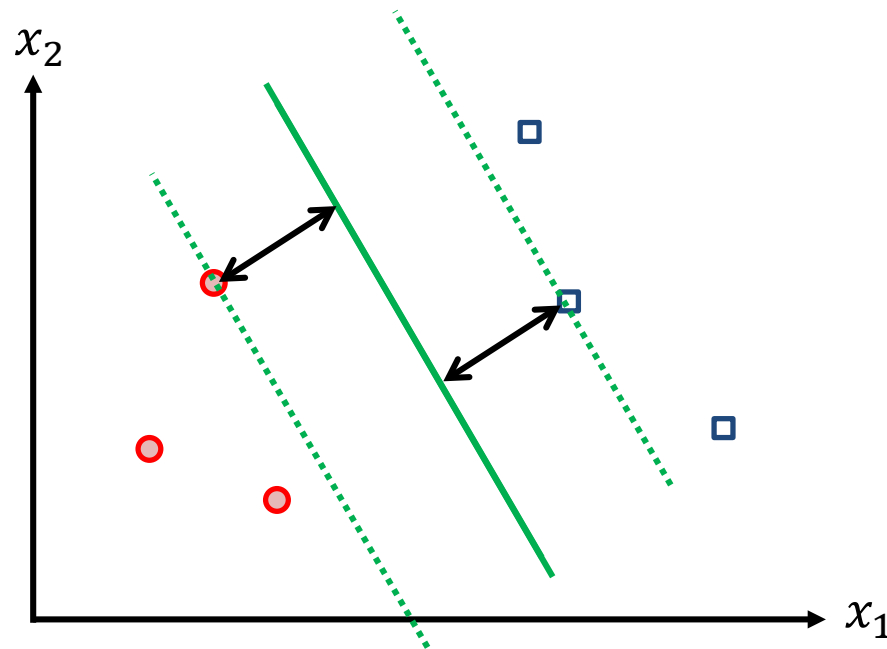
Finishing Touches

Complementary slackness

- Recall that one of the KKT conditions is *complementary slackness*

$$\lambda_i^*(y_i((\mathbf{w}^*)' \mathbf{x}_i + b^*) - 1) = 0$$

- Remember that $y_i(\mathbf{w}' \mathbf{x}_i + b) - 1 > 0$ means that \mathbf{x}_i is outside the margin



Points outside the margin must have $\lambda_i^* = 0$

Since most of the training points are outside the margin, we expect most of the λ s to be zero: the solution is *sparse*

The points with non-zero λ s are *support vectors*

Making predictions

- Predictions are based on the sign of $(b^* + \sum_{i=1}^n \lambda_i^* y_i \mathbf{x}'_i \mathbf{x})$
- Here \mathbf{x} is a new, previously unseen instance we need to classify
- The prediction depends on a weighted sum of terms
- The sum iterates over training data points, but λ_i^* are non-zero only for support vectors
- So effectively, the prediction is made by “dot-producting” (sort of “comparing”) the new point with each of the support vectors

Solving the dual problem

- The SVM Lagrangian dual problem is a *quadratic optimisation problem*. Using standard algorithms this problem can be solved in $O(n^3)$
- This is still inefficient for large data. Several specialised solutions have been proposed
- These solutions mostly involve decomposing the training data and breaking down the problem into a number of smaller optimisation problems that can be solved quickly
- The original SVM training algorithm called *chunking* exploits the fact that many of λ s will be zero
- *Sequential minimal optimisation* (SMO) is another algorithm which can be viewed as an extreme case of chunking. SMO is an iterative procedure that analytically optimises randomly chosen pairs of λ s in each iteration

This lecture

- Soft margin SVM
 - * Intuition and problem formulation
- Solving the optimisation
 - * Transforming the original objective
 - * Re-parameterisation
- Finishing touches
 - * Complementary slackness
 - * Solving the dual problem