Support Vector Machines

- Used for classification problems
- Model represents a hyperplane / line that seperates two classes into +1 or -1
- Maximizes its (line) distances to them, known as the margin
- Measures points to line

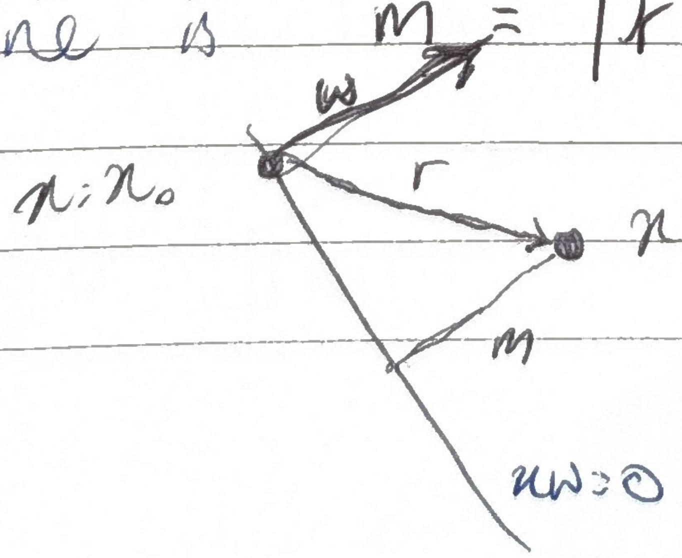
Optimal hyperplane:

- Max margin for both classes
- Correctly classify
- Means hyperplane finds midpoint

How to measure?

- length of projection of vector r on to the normal vector w of the line is $m = |r \cdot \frac{w}{||w||}$

normalise



Sub $r = x_1 - x_0$

$$m = \left| (x_1 - x_0) \cdot \frac{w}{||w||} \right|$$

remember:

(1) $y = x_0 w = 0$ on line

(2) $y = x_1 w = +1$ off line

Sub (1) from (2):

$x_1 w - x_0 w = +1 - 0$

$\Rightarrow (x_1 - x_0) w = +1$

Sub into original distance measure:

$m = \left| (x_1 - x_0) \cdot \frac{w}{\|w\|} \right|$

Sub = $(x_1 - x_0) w = 1$

$\Rightarrow m = 1 / \|w\|$

This is the same for both the -1 & +1 side

Adding the two margins (two points)

$\text{Margin} = \frac{2}{\|w\|}$

finding the optimal hyperplane:

1. Max the margin from both classes
   $\max \frac{2}{||w||}$
   or min the inverse $\frac{||w||^2}{2}$
   - this is squared to allow easy derivative

2. while still correctly classifying:
   such that $y_n(x_n w) \geq 1 \; \forall n$

SVM loss function (Primal)

$$L_{SVM} = (-)\sum_{n=1}^{N} \beta_n \underbrace{\left(y_n(x_n w) - 1\right)}_{\text{Classific Error}} + \underbrace{\frac{1}{2}||w||^2}_{\text{Marin to maximise}}$$

where β_n = lagrange muliplurs
aka coeff of addchonal constraints in a funchon to be optimized

recall from week 1

- if you want to find the optimum of a func
- take deriv of what you are trying to optim
- equate to 0
- this is the point of optimal value for that parameter in the function

Min of a loss function is a $\frac{dL(\omega)}{d\omega} = 0$
Find the ω that minimize func

Sub $L(\omega)$ for the SVM loss function

(1) $$\frac{d(-\sum \beta_n(y_n(x_n m) - 1) + \frac{1}{2}\|\omega\|^2)}{d\omega} = 0$$

(2) Apply rules of diff

$$-\sum \beta_n(y_n(x_n)) + \omega = 0$$

(3) make ω the subject

$$\omega = \sum \beta_n(y_n(x_n))$$

- now plug opt ω into loss function
- Expand & collect the terms

this gives another formulation of the loss function of the SVM

$$g(\beta) = \sum_{n=1}^{N} \beta_n - \frac{1}{2} \sum_{m=1}^{N} \beta_n \beta_m y_n y_m x_n x_m$$

where $\beta_n \geq 0$

known a Dual formulation of objective func

now instead of w to optimize we have $\beta_n \beta_m$

this makes it a constrained Quadratic Programming task

Solved using Quad Prog Solvers

not covered in mi module

MIXED DATA POINTS

with classification, data is rarely perfectly segmented

Hard Margin = all points in class lie on one side of Boundary

All correct classify

Soft Margin = Allow for some error

Optimize with a soft margin

$$\frac{\|w\|^2}{2} + C \sum \varepsilon_n$$

Such that $y_n(x_n w) \geq 1 - \varepsilon_n \quad \forall_n$

where $\varepsilon_n \geq 0$, C = Box constant

Hyperparameter

Non-linear Seperable Data

Kernal trick Solution
Pull apart data instance pairs (x_n, x_m) of different classes into a new dimensional space $K(x_n, x_m)$

Such that the data becomes sep in a new space

Dual formulation w/ kernal function

$$g(\beta) = \sum \beta_n - \frac{1}{2} \sum \beta_n \beta_m y_n y_m \underbrace{K(x_n, x_m)}_{\text{Kernal Func}}$$

We are not transforming individual pairs but instead data pairs

Default kernal is the linear kernal
$K(x_n, x_m) = x_n x_n$ i.e. no kernal

Types of non-linear kernels:
gaussian, Polynomial, Sigmoid / hyperbolic

Binary vs MultiClass

SVM is specified for binary classification

trick to use it for multiclass prob is to reformulate the problem in a series of 1 vs all or one-to-one problems

Summary

1. SVM = Maximise the hyperplane between two classes

2. Soft Margin is ~~not~~ needed for data that is not perfect linearly seperable

3. Use of non-linear kernel function for data that is not att all linearly seperable

4. Reformulate problem to create multiclass problem