Support Vector Machines

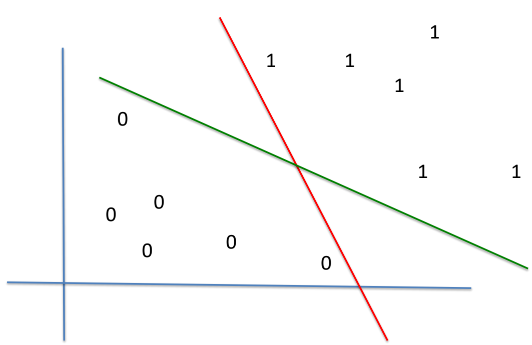
1

SVM

- Another linear classifier
- Maximum Margin classifier
- Approximation of non-linear functions using kernel methods

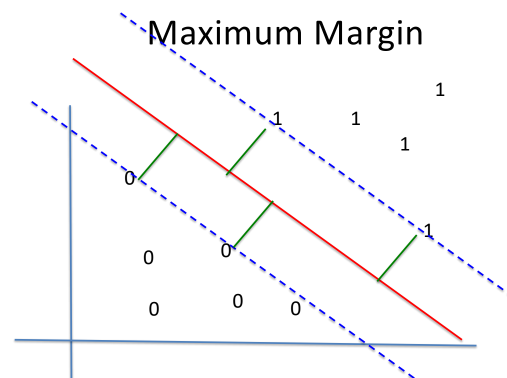
2

Maximum Margin



3

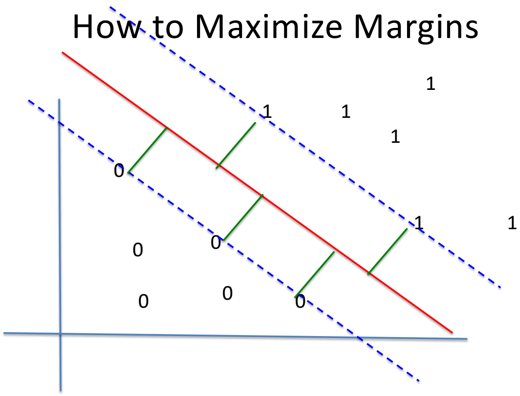
Maximum Margin



Margin, separating boundary halfway

4

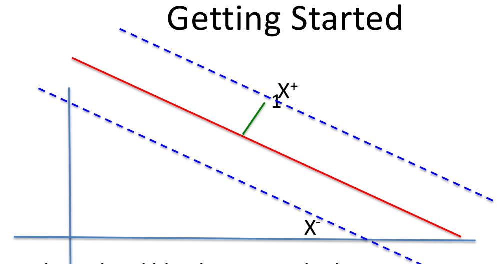
How to Maximize Margins



$w.x + b = 0$ $w.x + b = k$ $w.x + b = -k$

5

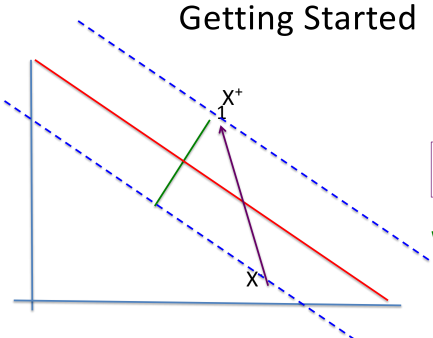
Getting Started



- The red and blue lines provide the margins.
- But we will choose them based on maximizing the margins.
- To get started we assume that the two dashed blue lines are **$w.x + b = 1$** and **$w.x + b = -1$** and the separating boundary is **$w.x + b = 0$.**

6

Getting Started



$w.x^+ + b = 1$
$w.x^- + b = -1$

$w.(x^+ - x^-) = 2$

What is $x^+ - x^-$?

What is $w.(x^+ - x^-)$?

7

Minimization

- **$w.(x^+ - x^-) = ||w|| \; ||(x^+ - x^-)||$** $\cos \theta$
- But **$||(x^+ - x^-)||$** $\cos \theta = d$
- So $||$**w**$||\ d = 2$
- Or $d = 2/(||$**w**$||)$

8

Maximize Margins

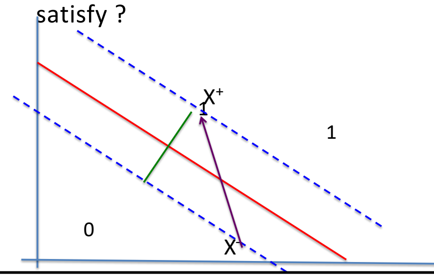
- Max d= 2/(||**w**||)
- Minimize (||w||)/2 or alternatively, minimize $(||w||)^2/2$

- But that can't work!
- Has nothing to do with the training data.
- Answer is the same where ||w||=0 regardless of the training set.

9

Accounting for training data

- If **w.x**+b=0 is indeed correct and the nearest positive training instances are on **w.x**+b=+1
- Then a positive/negative instance **x** must satisfy ?



10

Inequality Constraints

- All training instances, **x**, will lie on some parallel line.
- So **w.x** + b = k, for some k
  - If it is a positive instance k≥1 → |k| ≥ 1
  - If it is a negative instance, k≤ -1 → |k| ≥ 1

- So for all instances |**w.x** + b| ≥ 1
- Now we can minimize **w** subject to these conditions.

11

SVM training as an Optimization Problem

- Minimize (||**w**||)/2 subject to

  $|\mathbf{w.x}^i + b| \geq 1$ for $1 \leq i \leq N$

- Mathematically, hard to manipulate absolute values ($|\mathbf{w.x}^i + b|$).
  - We let the target values be 1 and -1 instead of 1 and 0.

12

-1 to the rescue

- If $\mathbf{x^i}$ is a positive instance (i.e., y^i=1) then
  - $\mathbf{w.x^i} + b \geq 1$
- So $y^i(\mathbf{w.x^i} + b) \geq 1$
- If $\mathbf{x^i}$ is a negative instance, (i.e., y^i=-1) then
  - $\mathbf{w.x^i} + b \leq -1$
- So $y^i (\mathbf{w.x^i} + b) \geq 1$

13

Optimization problem

- Min $(||\mathbf{w}||)^2/2$
  - subject to $y^i (\mathbf{w.x^i} + b) \geq 1$ for $1 \leq i \leq N$
- i.e., $y^i (\mathbf{w.x^i} + b) -1 \geq 0$ for $1 \leq i \leq N$
- i.e., , $- (y^i (\mathbf{w.x^i} + b) -1) \leq 0$ for $1 \leq i \leq N$

14

Primary Lagrangian objective

- $L_P = ½ \ (||\mathbf{w}||)^2 – \Sigma_i \lambda_i (y^i (\mathbf{w.x^i} + b) -1)$
- $dL_P/d\mathbf{w} =0 \rightarrow \mathbf{w} = \Sigma_i \lambda_i y^i \mathbf{x^i}$
- $dL_P/db =0 \rightarrow \Sigma_i \lambda_i y^i = 0$
- Additionally, we impose the KKT conditions
  - $\lambda_i \geq 0$
  - $\lambda_i(y^i (\mathbf{w.x^i} + b) -1) = 0$

15

Dual Form

- $L_D = \Sigma_i \lambda_i – ½ \Sigma_i \lambda_i \lambda_j y^i y^j \mathbf{x}_i.\mathbf{x}_j$
- Numerical techniques to find the λ's.
- Recall KKT conditions say:

  $\lambda_i \geq 0$

  $\lambda_i(y^i (\mathbf{w.x^i} + b) -1) = 0$

  $y^i (\mathbf{w.x^i} + b) -1) \geq 0$

Support Vectors are those instances for which $y^i (\mathbf{w.x^i} + b) -1) = 0$

16

Why Support Vectors

- Recall $\mathbf{w} = \Sigma_i \lambda_i y^i \mathbf{x}^i$
- So weight vector determined completely by the support vectors and their λ's. The non-support vectors have zero-valued λs.

17

The Separating Boundary

- Once we have figured out what **w** is, we can find out what b is.
- Pick any support vector, say **x'**. WLOG, we assume it is a positive instance. Since it is a support vector, we can plug it into the equation
- $\mathbf{w.x'} + b = 1$ and solve for b.
- Now, we know both **w** and b and thus we know the separator $\mathbf{w.x} + b = 0$.

18

Using SVM for Prediction

- Given an unseen data instance, say **z**, we can predict output by first computing
- $\mathbf{w.z}$+b. If it is >0 then we predict positive (1). Otherwise if the value is <0 then we predict z is a negative (-1) instance.

19

Without Computing **w**

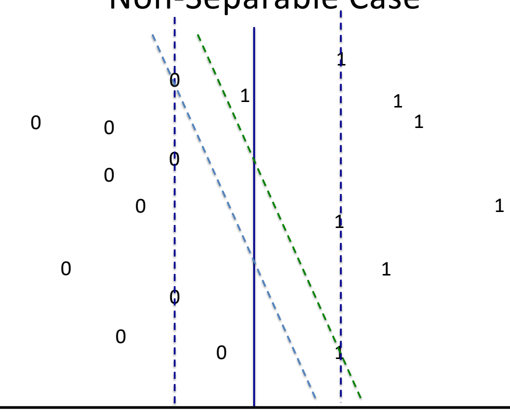
- We don't really need to compute **w.** Given an instance, **z**, and knowing $\mathbf{w} = \Sigma_i \lambda_i y^i \mathbf{x}^i$, we can compute $[(\Sigma_i \lambda_i y^i \mathbf{x}^i).\mathbf{z} + b]$ instead.
- Recall the support vectors are the only training instances that have non-zero λ's.
- We can rewrite this as $(\Sigma_i \lambda_i y^i (\mathbf{x}^i.\mathbf{z})) + b$
  - weighted sum of dot products with support vectors (SV).
- Weight for each SV is given by $\lambda_i y^i$ (positive when $y^i = 1$ and negative when $y^i = -1$).

20

Linear but Non Seperable

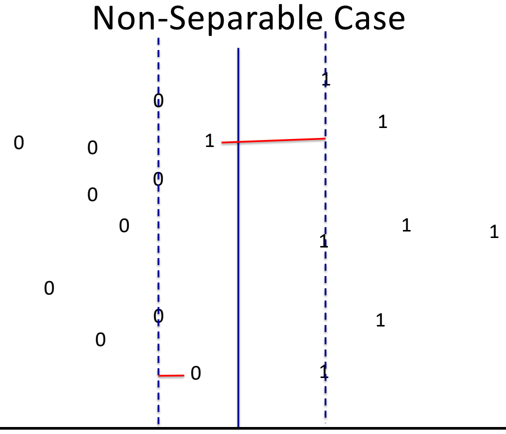
21

Non-Separable Case



22

Non-Separable Case



23

Error Term

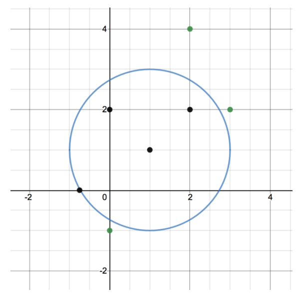
- $y^i(\mathbf{w}.\mathbf{x}^i + b) \geq (1 - e^i)$, $e^i \geq 0$
- Minimize $\frac{1}{2}(||\mathbf{w}||)^2 + C\,\Sigma_i\, e^i$
- Additional KKT $\mu_i e^i = 0$
- Same way to compute weights (using support vectors only).
- C is a parameter that allows us to trade off between generality (larger margins) and errors.

24

Kernel Trick

25

Capturing a Circle



Positive instances outside the circle

(0,-1), (2,4), (3,2)

Negative instances inside the circle

((-0.75,0), (0,2), (1,1), (2,2)

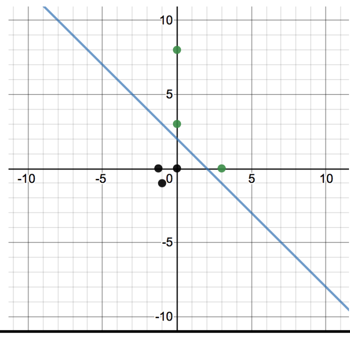
26

- $(x_1\text{-}1)^2 + (x_2\text{-}1)^2 = 4$
- $x_1^2 \text{-}2x_1\text{+}1 + x_2^2 \text{-}2x_2\text{+}1 = 4$
- $x_1^2 \text{-}2x_1 + x_2^2 \text{-}2x_2 – 2 = 0$
- Map <x_1, x_2> to <z_1, z_2> where
- $z_1 = x_1^2 \text{-}2x_1$ and $z_2 = x_2^2 \text{-}2x_2$
- $z_1 + z_2 - 2 = 0$ (a straight line!)

27

- $z_1 = x_1^2 \text{-}2x_1$ and $z_2 = x_2^2 \text{-}2x_2$
- $z_1 + z_2 = 2$ (a straight line!)
- +ve: (0,-1), (2,4), (3,2) ➔ (0,3), (0,8),(3,0)
- -ve: (-0.5,0), (0,2), (1,1), (2,2) ➔ (-1.25,0), (0,0), (-1,-1), (0,0)

28



29

- $(x_1\text{-}1)^2 + (x_2\text{-}1)^2 = 4 \Rightarrow x_1^2 - 2x_1 + x_2^2 - 2x_2 = 2$
- $z_1 = x_1^2 - 2x_1$ and $z_2 = x_2^2 - 2x_2$
- $z_1 + z_2 = 2$ in the new space

- But $\phi(<x_1, x_2>)$ to map training instances
- $\phi(<x_1, x_2>) = <x_1^2, x_1, x_2^2, x_2>$ will do the same trick.
- Learning will result in **w** = <1,-2,1,-2,> & w_0=2.

30

Kernels

- Recall in SVM $\mathbf{w} = \Sigma_i \lambda_i y^i \mathbf{x}^i$
- The decision boundary is $(\Sigma_i \lambda_i y^i \mathbf{x}^i).\mathbf{x} + b = 0$ i.e., $(\Sigma_i \lambda_i y^i \mathbf{x}^i .\mathbf{x}) + b = 0$
- When we use the mapping ϕ, this becomes

  $(\Sigma_i \lambda_i y^i \phi(\mathbf{x}^i) .\phi(\mathbf{x})) + b = 0$

- But we don't know ϕ
- What if there is a function $K(\mathbf{x}.\mathbf{x'})$ such that $K(\mathbf{x}.\mathbf{x'}) = \phi(\mathbf{x}) .\phi(\mathbf{x'})$

31

Existence of Kernels

- Computation in original space
- If K() is a kernel function. Then there is ϕ with

$K(\mathbf{x},\mathbf{y}) = \phi(\mathbf{x}).\phi(\mathbf{y})$

- Examples
  - Polynomial kernels $(\mathbf{x}.\mathbf{y}+1)^p$
  - Radial basis function kernel

32

Examples of Kernels

- $(\mathbf{x.y} + 1)^2 = (\langle x_1,x_2\rangle . \langle y_1,y_2\rangle + 1)^2 =$
- $(x_1y_1 + x_2y_2 + 1)^2 =$
- $x_1^2y_1^2 + 2x_1y_1x_2y_2 + 2x_1y_1 + 2\, x_2y_2 + x_2^2y_2^2 + 1 =$
- $\phi(\langle x_1,x_2\rangle).\phi(\langle y_1,y_2\rangle) =$
- $\langle x_1^2, \sqrt{2}\, x_1x_2, \sqrt{2}\, x_1, \sqrt{2}\, x_2, 1\rangle . \langle y_1^2, \sqrt{2}\, y_1y_2, \sqrt{2}\, y_1, \sqrt{2}\, y_2, 1\rangle$

$\phi(\langle x_1,x_2\rangle) = \langle x_1^2, \sqrt{2}\, x_1x_2, \sqrt{2}\, x_1, \sqrt{2}\, x_2, 1\rangle$

33

Second Example

- $K(D_1,D_2)$ = number of words they have in common.
- What is $\phi(D)$?
- $\phi(D) = \langle 0,0,1,0,1,1,\ldots\rangle$
- A component recording presence/absence of each word in the vocabulary.
- What is dot product?

34