5.2 SVM: Linear & Seperable

given D, n points $x_i \in \mathbb{R}^d$ w/ labels $y_i \in \{+1, -1\}$

Assume that a hyperplane exists that can perfectly seperate into $\{-1, +1\}$
$h(x)<0$ $h(x)>0$

Should this be the case then there is an infinite number of HPs in the space

Max Margin Hyperplane

fundemental to SVM is to choose the canonical hyperplane (w, b) that yeilds the Max margin among all possible seperating hyperplanes

If δ_h^* represents the margin for the HP $h(x)=0$ the the goal is to find the optimal h^*

$$h^* = \underset{h}{\operatorname{argmax}} \{\delta_h^*\} = \underset{w,b}{\operatorname{argmax}} \left\{ \frac{1}{\|w\|} \right\}$$

↑ Altered using scaling factor

SVM task is to find hyperplane that maximizes the margin $\frac{1}{\|w\|}$

Subject to constraints:

all $y_i(w^T x_i + b) \geq 1$ for all points x_i

Notice instead of maxing margin $\frac{1}{\|w\|}$

we can minimize just $\|w\|$

new object function: $\underset{w,b}{\operatorname{Min}} \left\{ \frac{\|w\|^2}{2} \right\}$

w/ same constraint

We can directly solve the minimization problem using standard optim algos

But it is more common to solve the Dual problem - obtained via the use of lagrange multipiles

introduce a lagrange multipiler a_i for each constraint (data point) which satisfies the Karush-Kuhn-tucker (KKT) conditions

KKT = Framework for solving constraint opt Probs

- Transform constrained probs into set of equations & inequalities that can be solved by opt techniques
- for SVMs KKT conditions are essential for deriving the dual formulation of the opt prob

$$a_i(y_i(w^T x_i + b) - 1) = 0$$

and $a_i \geq 0$

Incorp all n constraints into obj func

$$\min L = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{n} a_i \left(y_i \left(w^T x_i + b\right) - 1\right)$$

L should be min w/ respect to w & b

& Maximized with respect to a_i

Take the derivative w/ Res to w & b and set to 0

$$\frac{\partial}{\partial w} L = w - \sum_{i=1}^{n} a_i y_i x_i = 0$$

$$w = \sum_{i=1}^{n} a_i y_i x_i$$

$$\frac{\partial}{\partial b} L = \sum_{i=1}^{n} a_i y_i = 0$$

How does the lang change and we change b

This shows the optimal weight vector can be shown as a linear combination of the x_i data points w/ Lagrange multipliers as ~~co~~ coefficients, a_i, y_i

2nd eq implies the sum of lang Multi must be zero

Plug these constraints into the L function to obtain the dual Lagrangian Function

~~Primal~~

$$L_{Min} = \frac{1}{2} w^T w - \sum_{i=1}^{n} a_i (y_i (w^T x_i + b) - 1)$$

$\frac{1}{2} w^T w$: margin ($||w||^2$)

$\sum_{i=1}^{n} a_i (y_i (w^T x_i + b) - 1)$: constraints

Alot of under the hood Expands simplis takes Place

$$L_{dual} = \frac{1}{2} w^T w - w^T \underbrace{(\sum a_i y_i x_i)}_{w} - b \underbrace{\sum a_i y_i}_{0} + \sum a_i$$

$\frac{1}{2} w^T w - w^T w$

$$= -\frac{1}{2} w^T w + \sum a_i$$

- Expand, Transpose
- Distribute terms

$$= \sum a_i - \frac{1}{2} \sum \sum a_i a_j y_i y_j x_i^T x_j$$

This gives right to a new set of obj funcs

Started w/ Primal: $\min_{w,b} \left\{ \frac{\|w\|^2}{2} \right\}$

The new Dual objective:

Func: $\max_{a} L_{dual} = \sum a_i - \frac{1}{2} \sum_i \sum_j a_i a_j y_i y_j x_i^T x_j$

Constraints: $a_i \geq 0, \forall_i \in D, \sum a_i y_i = 0$

$a_n = (a_1, a_2, \ldots, a_n)^T$ vector of lag Multis

L_{dual} is convex quadratic programming prob (or a_n)
which can be solved using Standard opt techs, i.e
gradient based method for solve dual formulation

Dual = Min → Max

Cheaper computationally esp ~~dat~~ large data + Standard Techs

uses kernal tricks (???) good for non-linear data

Dual = Expressed in terms of lagrange

Dual = Same solution to Dual as primal

we do not solve w or b

only the new a_i Multipliers

Weight Vectors & bias

Once we have obtained the a_i values for $i = 1, \ldots, n$ we can solve for the weight and bias

According to KKT we have:

$$a_i (y_i (w^T x_i + b)) = 0$$

this means two things must occur:

(1) $a_i = 0$

(2) $\underbrace{y(w^T x_i + b)}_{\text{must be 1}} - 1 = 0$

important because:

- if $a_i > 0$ then $y_i (w^T x_i + b) = 1$
  - if $= 1$ then x_i point is a support vector
- otherwise > 1 & a_i has to be 0
- if not support then $a_i = 0$

Once we have solved for all a_i points we can compute the w vector

$$\text{weight} = w = \sum_{a_i > 0} a_i \, y_i \, w_i$$

note $a_i = 0$ works as a switch to turn off the non-support vectors

In other words w is obtained as a linear combination of its support vectors

The other points do not determ w

Computing b

recall, $a_i (y_i (w^T x_i + b_i) - 1) = 0$

- b is computed per support vectors, so $a_i \neq 0$
- Meaning inside has to $= 0$ for $= 0$

So, $y_i (w^T x_i + b_i) = 1$

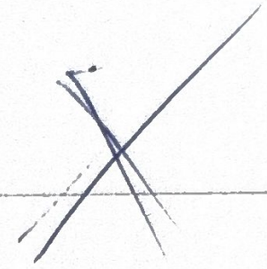
rearrange:

$$b_i = \frac{1}{y_i} - w^T x_i = y_i - w^T x_i$$

Then compute b as the av over all b's

$$b = \underset{a_i > 0}{\text{avg}} \{ b_i \}$$

SVM Classifier



Once you have found the optimal hyperplane function $h(x) = w^T x + b$

for any new data point its class can be predicted as

$$\hat{y} = \text{Sign}(h(z)) = \text{Sign}(w^T z + b)$$

Sign(·) function returns +1 if its argument is positive, -1 if neg

i.e. Depending on the side of the HP it falls on

With the same as Seperable but a new Slack & error to compute Those with δ over the margin

Hinge vs Quad loss to store Change obj function