The Wolfe dual problem

The lagrangian problem has m inequality constraints (where m is the number of training examples) and typically sloved using its dual form.

Duality principle: An optimization problem ~~b~~ can be viewed from two perspectives.

* The first one is the primal problem, a minimization problem in our case, and the other is dual problem, which will be a maximization problem.
* What is interesting is that the maximization of dual problem will always be less than or equal to the minimum of primal problem.

$\Rightarrow$ max(dual) $\leq$ Min(primal)

Recall: Lagrangian problem

$$\min_{w} f(w) = \frac{\|w\|^2}{2}$$

$$\text{STC} \quad -y_i(w \cdot x_i + d) + 1 \leq 0$$

Let $g_i(w) = -y_i(w \cdot x_i + d) + 1$

$$L(w, \alpha, d) = f(w) \oplus \sum_{i=1}^{m} \alpha_i g_i(w)$$

We can use + or - the output will not effect!

Define:

$$\theta_p(w) = \max_{\substack{\alpha_i \\ \boxed{\alpha_i \geq 0}}} L(w, \alpha, d)$$

$$\theta_p(w) = \max_{\substack{\alpha_i \\ \alpha_i \geq 0}} \left(f(w) + \sum_{i=1}^{m} \alpha_i g_i(w) \right)$$

If $g_i(w) > 0$ [violates condition]

$$\theta_p(w) = \infty$$

If $g_i(w) \leq 0$ Satisfying the condition then to make $\theta_p(w)$ maximum we have to take $\alpha_i = 0$ ($\because \alpha_i \geq 0$) since $g_i(w)$ is negative.

$$\Rightarrow \sum_{i=1}^{m} \alpha_i \underbrace{g_i(w)}_{=0} = 0 \quad (\because \alpha_i = 0)$$

$$\therefore \boxed{\theta_p(w) = f(w)}$$

So

$$\theta_p(w) = \begin{cases} f(w) & \rightarrow \text{Satisfies cond}^n \\ \infty & \rightarrow \text{violates} \end{cases}$$

Now, the primal problem is defined as

$$p^* = \min_{w,d} \theta_p(w)$$

$\hookrightarrow$ primal function

$$p^* = \min_{w,d}\left(\max_{\substack{\alpha_i \\ \alpha_i \geq 0}} L(w,\alpha,d)\right)$$

Dual problem :-

$$d^* = \max_{\substack{\alpha_i \\ \alpha_i \geq 0}} \min_{w,d} L(w,\alpha,d)$$

$$= \max_{\substack{\alpha_i \\ \alpha_i \geq 0}} \theta_d(w,d)$$

$\hookrightarrow$ dual function

Fact: $d^* \leq p^*$ (we discussed before)

Eg: $\max_x \min_y \cos(x+y) \leq \min_y \max_x \cos(x+y)$

$$\Rightarrow \max_x(-1) \leq \min_y(+1)$$

$$\Rightarrow -1 \leq 1 \qquad \because \cos x \in [-1,1]$$

* In our case, we are trying to solve convex optimization problem, and slater's condition holds for affine (Hyperplane) constraints.

* This means solving dual is the same thing as solving the primal, except it is easier.

The Lagrangian dual problem:

$$d^* = \max_{\substack{\alpha_i \\ \alpha_i \geq 0}} \underbrace{\min_{w, b} L(w, \alpha, b)}$$

(dual)

we already solved this! got (A)

Recall

$$W(\alpha) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j (x_i \cdot x_j) \quad \text{—(A)}$$

↑

This is the Wolfe dual Lagrangian function

The optimization problem now is called the Wolfe dual problem

$$\underset{\alpha}{\text{maximize}} \; W(\alpha)$$

Subjected to $\alpha_i \geq 0$, for any $i = 1, \ldots, m$

$$\sum_{i=1}^{m} \alpha_i y_i = 0$$

(Recall)

* The main advantage of the Wolfe dual problem over the lagrangian problem is that the objective function W now only depends on Lagrange multipliers.

V.Imp This formulation will help us slove the problem in python.

What to do once we have the multipliers

* When we solve wolfe dual problem, we get a vector 'α' containing all lagrangian multipliers.

Compute W

$$\boxed{W = \sum_{i=1}^{m} \alpha_i y_i x_i} \text{ (recall)}$$

Compute d :-

once we have W, we know the closest points to the hyperplane will satisfy π^+

recall, $\pi^+ : W \cdot x_i + d = 1$

also we know constraint $y_i(w \cdot x_i + d) \geq 1$

The points that lie on π^+ will satisfy :

$$y_i(w \cdot x_i + d) = 1 \quad \text{where } y_i \in \{-1, 1\}$$

└①

we multiply both sides of the equation ① by y_i, and because $y_i^2 = 1$, it gives us:

$$w \cdot x_i + d = y_i$$

$$d = y_i - w \cdot x_i$$

x_i's ~~are~~ is the support vector.

* Instead of taking random support vector x_i, taking the average provides us with a numerical → more stable solution.

$$\therefore \boxed{d = \frac{1}{S} \sum_{i=1}^{S} (y_i - w \cdot x_i)}$$

** Using these support vectors we are able to find the equation of Hyperplane. This is reason why we call support machines as support vector machine.

Finally :

Solving SVMs with a QP Solver:

* A QP solver is a program used to solve quadratic programming problems.
* We use python package called CVXOPT (convex optimization)
* This package provides a method that is able to solve quadratic problems of the form:

$$\underset{x}{\text{minimize}} \quad \frac{1}{2}x^T P x + q^T x$$

$$\text{subjected to} \quad Gx \le h$$
$$Ax = b$$

Rewriting it with α instead of x,,

$$\underset{\alpha}{\text{minimize}} \quad \frac{1}{2}\alpha^T P \alpha + q^T \alpha$$

$$\text{subjected to} \quad G\alpha \le h$$
$$A\alpha = b$$

We will change the wolfe dual problem, from maximization problem to minimization problem by multiplying by -1

$$\underset{\alpha}{\text{minimize}} \quad \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}\alpha_i\alpha_j y_i y_j (x_i \cdot x_j) - \sum_{i=1}^{m}\alpha_i$$

subject to $-\alpha_i \leq 0$ for any $i = 1, \ldots m$

$$\sum_{i=1}^{m}\alpha_i y_i = 0$$

* Then we introduce $\alpha = (\alpha_1, \cdots \alpha_m)^T$ and

$y = (y_1, \cdots y_m)^T$ and the Gram matrix

K for all possible dot products of vectors x_i

$$K(x_1, \cdots x_m) = \begin{bmatrix} x_1 \cdot x_1 & x_1 \cdot x_2 & \cdots & x_1 \cdot x_m \\ x_2 \cdot x_1 & x_2 \cdot x_2 & \cdots & x_2 \cdot x_m \\ \vdots & & & \\ x_m \cdot x_1 & x_m \cdot x_2 & \cdots & x_m \cdot x_m \end{bmatrix}$$

$\Rightarrow \alpha_j = \alpha^T, \; \alpha_i = \alpha, \; y_i y_j = yy^T, \; x_i \cdot x_j = K$

$$\sum_{i=1}^{m}\alpha_i y_i = y \cdot \alpha \quad \text{where } y = yy^T$$

$$\underset{\alpha}{\text{minimize}} \quad \frac{1}{2}\alpha^T(yy^T K)\alpha - \alpha$$

S.T.C $\quad -\alpha \leq 0$

$$y \cdot \alpha = 0$$

* we know able to find out the value for each of the parameters

$P = yy^T K$

$q^T = -I$

$G = -I$

h = Null matrix or zero

$A = yy^T = Y$

b = zero matrix,,

Now lets get into Code Example !!

Congrats