Lecture 17

Name: Harsh Sanjay Roniyar

Roll Number: 22B3942

Kernels

$K(x_i, x_j) \rightarrow$ Similarity between x_i and x_j

$k(x_i, x_j) = \phi(x_i)^T\phi(x_j) \leftrightarrow \text{ Ensures that Gram Matrix is P.S.D. (positive semi-definite)}$

k is a simple computable expression, e.g. $e^{-k\|x_i - x_j\|_2^2}$ and the output is scalar,
$k(x_i, x_j) = k(x_j, x_i)$ and
$\phi(x_i)$ is the theoretical vector feature computed using x_i, and can have infinite dimension.

This above kernel example is known as `Mercer Kernel`.
i.e.

1. Implements an inner product in Hilbert space
2. Has a positive semi-definite Gram matrix.

These kernels are generally simple to compute although the vector features might not be easily computable

Kernel Trick

1. Take an objective function (such as a loss function) of x and try to manipulate it, so that it becomes a function of $x_i^T x_j$
2. Replace $x_i^T x_j$ with $\phi(x_i)^T\phi(x_j)$, which is basically $k(x_i, x_j)$

`Primal Form` - Objective Function of x
`Dual Form` - Function of $x_i^T x_j$

This trick is used to make linear problems non-linear in x-space, but will be easier to solve in the kernel space.

Primal Form of SVM

$$L(w, b) = \frac{1}{2}\|w\|_2^2 - \sum_i a_i[t_i(w^T x_i + b) - 1]$$

s.t. $\forall i \; a_i \geq 0$

Optimization Background:

$$L(x, \lambda) = f(x) + \lambda g(x), \lambda \geq 0$$

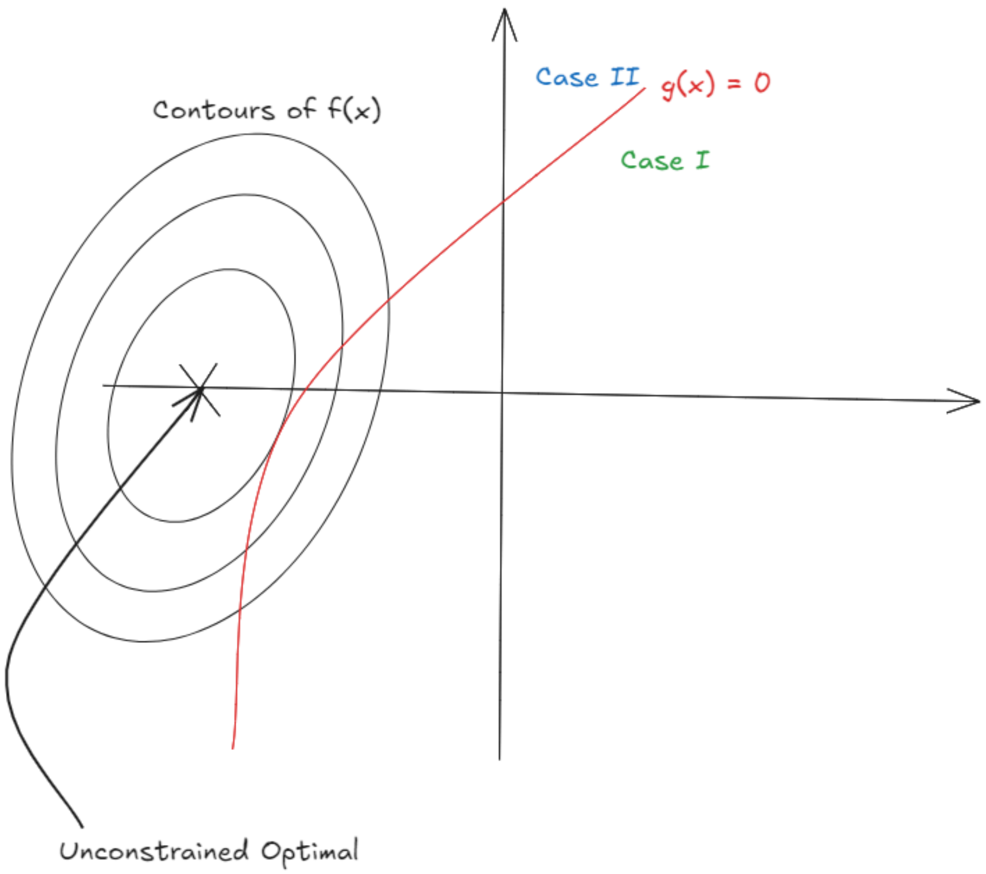
which is equivalent to

$$\max f(x), \text{s.t. } g(x) \geq 0$$

This is a single-constraint case.

There are two cases possible here:

1. $g(x) > 0$ is above the $g(x) = 0$ curve
2. $g(x) > 0$ is below the $g(x) = 0$ curve



For Case I, $\nabla f(x)$ is in opposite direction to $\nabla g(x) \implies \lambda > 0$
For Case II, $\nabla g(x)$ does not matter $\implies \lambda = 0$, since now we can directly find the unconstrained optimal value as we are guaranteed to lie on the safe side for $\lambda = 0$

The final optimization would look like

$$\min L(w, b, a) = \frac{1}{2}\|w\|_2^2 - \sum_i a_i[t_i(w^T x_i + b) - 1], a_i \geq 0$$

$$\frac{\partial L}{\partial w} = 0 \implies w = \sum_i a_i t_i x_i$$

$$\frac{\partial L}{\partial b} = 0 \implies 0 = \sum_i a_i t_i$$

Substituting the values, from above in the loss function $\rightarrow$
Therefore, the `Dual Formulation` in a will be -

$$L(a) = \sum_i x_i - \frac{1}{2} \sum_i \sum_j a_i a_j t_i t_j x_i^T x_j$$

second term derived from

$$\left[\sum_i a_i t_i x_i\right]^T \left[\sum_j a_j t_j x_j\right]$$

and we will replace $x_i^T x_j$ with our kernel function $k(x_i, x_j)$ which would apply $\phi(x_i)^T \phi(x_j)$.

Now, we will also need to compute the class, hence

$$y = \left(\sum_i a_i t_i x_i^T x_i\right) + b$$

Now, for eliminating b,

$$b = \frac{1}{N_S} \sum_{i \in S} \left[t_i - \sum_{j \in S} a_j t_j x_j^T x_j\right]$$

where, S is the set of support vectors ($a > 0$)

Case 1	Case 2
$g(x) = 0$ $\lambda > 0$	$g(x) > 0$ $\lambda = 0$
$\lambda g(x) = 0$	$\lambda g(x) = 0$

1. $a_i \geq 0, \forall i$
2. $t_i y(x_i) - 1 \geq 0$
3. $a_i\,[t_i y(x_i) - 1] = 0$ `where the first term` (a_i) `is zero for non support vector and` $(t_i y(x_i) - 1 = 0)$ `for support vectors`.

These three conditions are known as the `KKT Conditions` (Karush-Kuhn-Tucker)

For Soft SVM

Primal Form:

$$L(w, b, a, \mu) = \frac{1}{2}\|w\|_2^2 - \sum_i a_i \left[t_i y(x_i) - 1 + \xi_i\right] + C\sum_i \xi_i - \sum_i \mu_i \xi_i, \text{ for } a_i \geq 0, \mu_i \geq 0 \forall i$$

where the last term is the Lagrangian, and now we will have `six` KKT conditions, since now we will also have a condition from

$$\frac{\partial L}{\partial \xi_i} = 0$$

The conditions are -

1. $a_i \geq 0$
2. $t_i y(x_i) - 1 + \xi_i \geq 0$
3. $a_i \left[t_i y(x_i) - 1 + \xi_i\right] = 0$
4. $\mu_i \geq 0$
5. $\xi_i \geq 0$
6. $\mu_i \xi_i = 0$

After following a similar process as before for the Hard SVM, we get the following `Dual Formulation` -

$$\tilde{L}(a) = \sum_i a_i - \frac{1}{2}\sum_i \sum_j a_i a_j t_i t_j x_i^T x_j$$

where, $0 \leq a_i \leq c$ and $\sum_i a_i t_i = 0$

`If` $a_i \neq 0$ `then implies not a support vector` $\implies$ away from margin
`Else If` $0 < a_i < c$ $\implies$ support vector but on the margin
`Else` $a_i = c$ $\implies$ inside the margin but support vector depending on value of ξ_i.