MACHINE LEARNING HOMEWORK SHEET - 07

SOFT MARGIN SVM
KERNELS

TUM ID : ge73yuw
DATE : 9/12/2018

NAME: MAHALAKSHMI SABANAYAGAM

SOFT MARGIN SVM.

Problem 1: D- Linearly separable dataset. Soft margin SVM is fitted. Is it guaranteed that all the training samples in D will be assigned the correct label by the model?

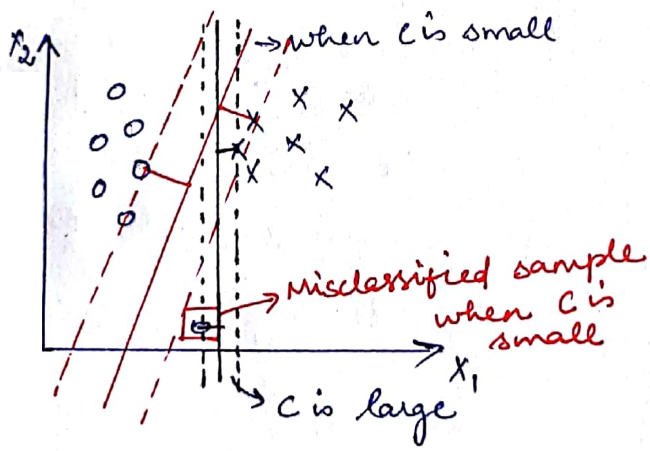
No, it is not guaranteed that all the training samples in the dataset will be assigned correct label although the dataset is linearly separable.

Soft margin relaxes the constraints but punishes the relaxation.

The new cost function for soft margin is,

$$f_0(w, b, \xi_i) = \frac{1}{2} w^T w + C \sum_{i=1}^{N} \xi_i \quad \text{where } C > 0$$

C tells us how heavy a violation is punished.
So when C is very small, the training sample could be misclassified.



The adjacent figure shows one case when the training sample is misclassified, although D is linearly separable.

Problem 2: Why $C > 0$?

The cost function of soft margin SVM is,

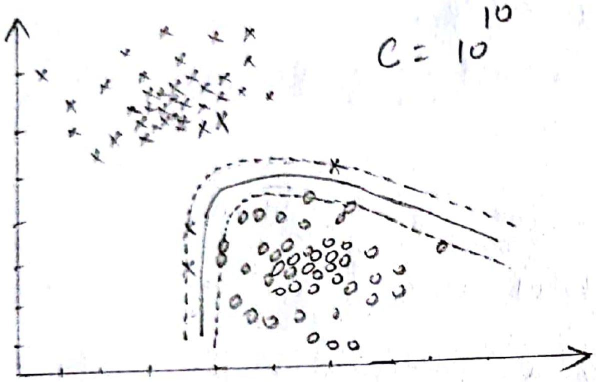
$$f_0(w, b, \xi_i) = \frac{1}{2} w^T w + C \sum_{i=1}^{N} \xi_i$$

ξ is 0 for rightly classified samples.
(0,1) for samples between the margin
=1 for the ones on the margin
>1 for misclassified samples.

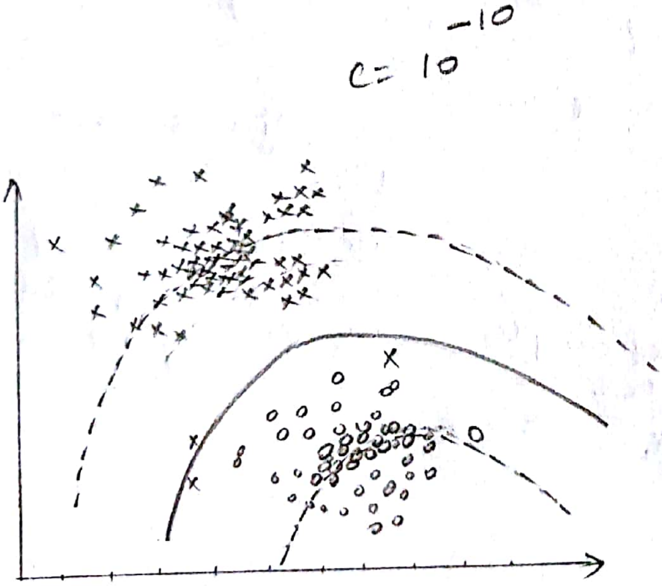
So, the parameter C is regularization term which penalises this distance measure for not rightly classified samples. Our objective is to minimize f_0 (w, b, ξ).

When $C \not> 0$ (ie) $C \leq 0$, we would be subtracting $|C|\xi$ (ie) a factor of misclassified samples' distance from ~~a~~ the original cost function which is incorrect. Thus ~~doing~~ when $C > 0$ is unsatisfied, our cost function gives undesired result.

Problem 3: Quadratic kernel.



When c is very large, the model tries to overfit and classifies all the samples correctly. So, in this case the margin will be very small.



When c is very small, it tries to underfit or over generalise the samples, leading to bigger margin.

Problem 4:

$$k(x_1, x_2) = \sum_{i=1}^{N} a_i (x_1^T x_2)^i + a_0 \quad ; \quad N \in \mathbb{N}$$

$a_i \geq 0$

$i \in [0, N]$

To prove: $k(x_1, x_2)$ is a valid kernel.

$x_1^T x_2$ - dot pdt and it is a kernel. $(k_1(x_1, x_2))$

All polynomial kernels are valid $\Rightarrow (k_1(x_1, x_2))^n$ is valid

$\therefore (x_1^T x_2)^1, (x_1^T x_2)^2, \ldots (x_1^T x_2)^N$ are valid

$c \cdot k_1(x_1, x_2)$ is valid for $c > 0 \Rightarrow \sum_{i=1}^{N} \underbrace{a_i (x_1^T x_2)^i}_{k_{2i}(x_1, x_2)}$ is valid

$$\therefore \sum_{i=1}^{N} k_{2i}(x_1, x_2) + a_0$$

$$= \sum_{i=1}^{N} k_{2i}(x_1, x_2) + \sum_{i=1}^{N} a_0$$

$$= \sum_{i=1}^{N} k_{2i}(x_1, x_2) + N a_0$$

$\Downarrow$

$=$ sum of valid kernels $+ \; N k_3(\phi(x_1), \phi(x_2))$ where $k_3(\phi(x_1), \phi(x_2)) = a_0$ as $a_0 \geq 0$.

& $N \in \mathbb{N}$. (>0)

$\Rightarrow N a_0$ is a kernel.

$=$ sum of valid kernels $+$ valid kernel

$=$ Valid kernel.

$\therefore k(x_1, x_2)$ is a valid kernel.

Problem 5: $k(x_1, x_2) = \dfrac{1}{1 - x_1 x_2}$ $\quad x_1, x_2 \in (0, 1)$

Find feature transformation $\phi(x)$.

$x_1, x_2 \in (0, 1)$

$\Rightarrow 0 < x_1 x_2 < 1$.

$\therefore$ We can apply the expansion,

$$\frac{1}{1-x} = 1 + x + x^2 + x^3 + \ldots \quad \text{when } -1 < x < 1.$$

$$k(x_1, x_2) = \frac{1}{1 - x_1 x_2} \quad ; \; 0 < x_1 x_2 < 1$$

$$= 1 + (x_1 x_2) + (x_1 x_2)^2 + (x_1 x_2)^3 + (x_1 x_2)^4 + \cdots$$

$$\therefore \phi(x) = \{1, \sqrt{x_1 x_2}, x_1 x_2, (x_1 x_2)^{3/2}, (x_1 x_2)^2, \ldots\}$$

When $\phi(x)$ transformation is applied we get the kernel function by approximation of the sum series.

Problem 6:

a. Algorithm:
The algorithm calculates no. of occurances of each character in one string with the other.
Each character is considered unique (ie) repeated characters are counted as many times as it repeats.

b. $k: S \times S \rightarrow R$

$k(x, y)$ x, y are strings.

Let $\phi_1(x)$ be a transformation function

$$\phi_1(x) = \begin{cases} 1 & \text{if the pattern in } x_{ij} = aa \\ 0 & \text{if the pattern in } x_{ij} = ab \end{cases}$$

$$\therefore k(x, y) = sum(\phi_1(x^T y))$$

Let $\phi_2(x)$ = sum of all the elements in the matrix.

$$\therefore k(x, y) = \phi_2(\phi_1(x^T y))$$

S is set of strings over finite alphabet of size v.

$$|S| = s.$$

$$k: S \times S \rightarrow R$$

Lets look at each element in the kernel Matrix / Gram Matrix;

the diagonal entries : $k(x_1, x_1), k(x_2, x_2) \ldots k(x_s, x_s)$

$k(x_1, x_1) = \phi_2(\phi_1(x_1^T x_1)) \geq$ no. of characters in $x_1 \in N$.

~~$K(x_1, x_1)$~~ So, the diagonal entries will be $> 0 \in N$.

Other elements in the matrix,

$k(x_1, x_2)$ & $k(x_2, x_1)$

occurances of each character in x_1 with x_2 =

occurances of each character in x_2 with x_1.

$\phi_1(X)$ = matrix of 1s and 0s.

$\phi_2(\phi_1(X))$ = # 1s. & $\phi_2(X) = \phi_2(X^T)$ as ϕ_2 is scalar. (just the sum)

$$\phi_2(\phi_1(x_1^T x_2)) = \phi_2((\phi_1(x_1^T x_2))^T)$$

$\Downarrow$

$k(x_1, x_2)$

$$= \phi_2(\phi_1((x_1^T x_2)^T))$$

$$= \phi_2(\phi_1(x_2^T x_1))$$

$$= k(x_2, x_1)$$

$$\therefore k(x_1, x_2) = k(x_2, x_1)$$

$\Rightarrow K$ (Gram matrix)

$$= \begin{bmatrix} k(x_1,x_1) & k(x_1, x_2) & \cdots & k(x_1, x_s) \\ k(x_2, x_1) & k(x_2, x_2) & \cdots & k(x_2, x_s) \\ \vdots & & & \\ k(x_s, x_1) & k(x_s, x_2) & \cdots & k(x_s, x_s) \end{bmatrix}$$

$$= \begin{bmatrix} n_1 & s_{12} & \cdots & s_{1s} \\ s_{12} & n_2 & \cdots & s_{2s} \\ \vdots & \vdots & \ddots & \vdots \\ s_{1s} & s_{2s} & & n_s \end{bmatrix}$$

= Symmetric matrix.

$n_1, n_2 \ldots n_s > 0$ & $s_{12}, s_{1s} \geq 0$

$\Rightarrow$ Positive semi definite

As k is symmetric, PSD the kernel is Valid.

Problem 7:

$$k_G(x_1, x_2) = \exp\left(-\frac{|x_1 - x_2|}{2\sigma^2}\right)$$

Yes, any finite set of points can be linearly separated in the feature space of the Gaussian kernel provided σ can be chosen freely.

By choosing very small σ, we can overfit the data as each sample is σ times apart from each other and each sample determines the class in its neighbourhood.

The Gaussian kernel takes the data to infinite dimensional feature space. In the feature space, depending on the chosen σ (smaller the σ, well separated the data in the feature space), there exists a hyperplane in the infinite dimensional feature space.