# Support vector Machines

## Pattern Recognition And Machine Learning

Indian Institute of Technology Madras

November 2, 2019

A hyper plane is defined by the equation

$$g(\bar{x}) = \bar{w}^t \bar{x} + w_0$$

let $\bar{x}$ be any point in the space, which can be written in terms of a point $\bar{x_p}$ on the hyperplane

$$\begin{aligned}
\bar{x} &= \bar{x}_p + \frac{\bar{w}}{||\bar{w}||} \cdot r \\
\implies g(\bar{x}) &= \bar{w}^t \bar{x}_p + \frac{\bar{w}^t \bar{w}}{||\bar{w}||} \cdot r + w_0 \\
&= \frac{||\bar{w}||^2}{||\bar{w}||} \cdot r \\
&= ||\bar{w}|| \cdot r \\
\implies r &= \frac{g(\bar{x})}{||\bar{w}||}
\end{aligned}$$

$r$ is the distance (margin) of the point $\bar{x}$ to the hyperplane.

Now, we define a b-margin hyper plane

$$y_n(\bar{w}^t \cdot \bar{x}_n + w_0) \geq b \tag{1}$$

Here, $y_n$ is the class label

- The objective is to find the maximum margin hyper plane
- Margin is the distance of the nearest training example
- The maximum margin hyper plane is given by

$$\begin{aligned} H_k^* &= \arg\max_k \cdot margin_k \\ &= \arg\max_k \frac{\bar{w}_k^t \bar{x} + w_0 k}{||\bar{w}_k||} \end{aligned}$$

The objective is to find w such that

$$\frac{y_n(\bar{w}^t x + w_0)}{||\bar{w}||} \geq ||margin||$$

$$\text{If} \quad y_n(\bar{w}^t x + w_0) \geq b \quad \text{Margin is} \quad \frac{b}{||w||}$$

The margin becomes $\frac{1}{||w||}$, then the separating hyper place is termed as **Canonical separating hyper plane**

## How to train SVM?

- Let $\mathcal{D}_1, \mathcal{D}_2 = \mathcal{D}$ be the data each class respectively.
- Let $\bar{w}, w_o$ the parameters that define the hyper plane $\bar{w}^t.\bar{x} + w_0 = 0$ of SVM.
- $\therefore$ We need to optimize the following cost function

$$J(\bar{w}, w_o) = \frac{1}{2}\bar{w}^t\bar{w}$$

Subject to the condition

$$y_n(\bar{w}^t\bar{x}_n + w_0) \geq 1, \; n = 1, 2, ...N$$

## How to train SVM?

- Let $\mathcal{D}_1, \mathcal{D}_2 = \mathcal{D}$ be the data each class respectively.
- Let $\bar{w}, w_o$ the parameters that define the hyper plane $\bar{w}^t.\bar{x} + w_0 = 0$ of SVM.
- $\therefore$ We need to optimize the following cost function

$$J(\bar{w}, w_o) = \frac{1}{2}\bar{w}^t\bar{w} \tag{2}$$

Subject to the condition

$$y_n(\bar{w}^t\bar{x}_n + w_0) \geq 1,\ n = 1, 2, ...N \tag{3}$$

# How to train SVM (Contd..)?

- KarushKuhnTucker conditions for optimization mentioned in previous slide can be written as

$$L_D(\bar{w}, w_o, \bar{\alpha}) = \frac{1}{2}\bar{w}^t\bar{w} - \sum_{n=1}^{N} \alpha_n \left[ y_n(\bar{w}^t\bar{x}_n + w_0) - 1 \right] \quad (4)$$

  This also called as the dual form of the optimization problem in Equation 2

- Setting the derivative $L_p$ w.r.t. $\bar{w}$ and $w_o$ to zero, we get:

$$\frac{\partial L_D}{\partial \bar{w}} = 0 \implies \qquad \bar{w} = \sum_{n=1}^{N} \alpha_n\, y_n\, \bar{x}_n \quad (5)$$

$$\frac{\partial L_D}{\partial \bar{w}_o} = 0 \implies \qquad \sum_{n=1}^{N} \alpha_n\, y_n = 0 \quad (6)$$

# How to train SVM (Contd..)?

- Substituting Equation 5 and 6 in Equation 4, we get the primal form as

$$L_p(\bar{\alpha}) = \sum_{n=1}^{N} \alpha_n - \frac{1}{2} \sum_{n=1}^{N} \sum_{m=1}^{N} \alpha_n \, \alpha_m \, y_n \, y_m \, \bar{x}_n^t \, \bar{x}_m \tag{7}$$

Subject to the following condition

$$\alpha_n \geq 0 \tag{8}$$

$$\sum_{n=1}^{N} \alpha_n \, y_n = 0 \tag{9}$$

## How to train SVM (Contd..)?

- Solving for $\bar{\alpha}^* = argmax(L_p(\bar{\alpha}))$, we get the solution for $\bar{w}^*$ as

$$\bar{w}* = \sum_{n=1}^{N} \alpha_n^* \, y_n \, \bar{x}_n \tag{10}$$

  Where the examples with $\alpha_n^* > 0$ corresponds to the support vectors.

- $w_o*$ can be obtained by substituting $\bar{w}*$ in $y_n(\bar{w}^t \bar{x}_n + w_0) - 1 = 0$ for any support vector with $\alpha_n^* > 0$

$$w_o^* = 1 - \bar{w}^t \bar{x}_n \tag{11}$$

- The final decision boundary is given by

$$g(\bar{x}) = \bar{w}^{*t} \bar{x} + w_o^* = \sum_{n=1}^{N_s} \alpha_n^* \, y_n \, \bar{x}_n{}^t \, \bar{x} + w_o^* \tag{12}$$

# C-SVM

## Disadvantage of SVM

- SVM assumes the data to be linearly separable. But in real world the data can be overlapping.
- To overcome this, C-SVM with a slack variable was introduced.

- We introduce a slack variable to the cost function as follows

$$J(\bar{w}, w_o) = \frac{1}{2}\bar{w}^t\bar{w} + C\sum_{n=1}^{N}\xi_n$$

Subject to following follows:

$$y_n(\bar{w}^t\bar{x}_n + w_0) \geq 1 - \xi_n,\ \ n = 1, 2, ...N$$
$$\xi_n \geq 0,\ \ n = 1, 2, ...N$$

# How to train C-SVM ?

- The dual form of the optimization problem for C-SVM is given as

$$
\begin{aligned}
L_D(\bar{w}, w_o, \bar{\xi}, \bar{\alpha}, \bar{\beta}) = \frac{1}{2}\bar{w}^t\bar{w} + C\sum_{n=1}^{N}\xi_n \quad (13) \\
-\sum_{n=1}^{N}\alpha_n\left[y_n(\bar{w}^t\bar{x}_n + w_0) - 1 + \xi_n\right] \quad (14) \\
+\sum_{n=1}^{N}\beta_n\,\xi_n \quad (15)
\end{aligned}
$$

# How to train C-SVM (Contd..) ?

- Setting the derivative $L_D$ in Equation 13 w.r.t. $\bar{w}, w_o$ and $\xi_n$ to zero, we get:

$$\frac{\partial L_D}{\partial \bar{w}} = 0 \implies \qquad \bar{w} = \sum_{n=1}^{N} \alpha_n \, y_n \, \bar{x}_n \tag{16}$$

$$\frac{\partial L_D}{\partial \bar{w_o}} = 0 \implies \qquad \sum_{n=1}^{N} \alpha_n \, y_n = 0 \tag{17}$$

$$\frac{\partial L_D}{\partial \xi_n} = 0 \implies \qquad C - \alpha_n - \beta_n = 0 \tag{18}$$

$$\implies \qquad \alpha_n + \beta_n = C \tag{19}$$

# How to train C-SVM (Contd..)?

- Substituting Equation 16,17 and 19 in Equation 13, we get the primal form as

$$L_p(\bar{\alpha}) = \sum_{n=1}^{N} \alpha_n - \frac{1}{2} \sum_{n=1}^{N} \sum_{m=1}^{N} \alpha_n \, \alpha_m \, y_n \, y_m \, \bar{x}_n^t \, \bar{x}_m \tag{20}$$

Subject to the following condition

$$0 \leq \alpha_n \leq C \tag{21}$$

$$\sum_{n=1}^{N} \alpha_n \, y_n = 0 \tag{22}$$

# How to train C-SVM (Contd..)?

- Solving for $\bar{\alpha}^* = argmax(L_p(\bar{\alpha}))$, we get decision boundary as

$$g(\bar{x}) = \bar{w}^{*t}\bar{x} + w_o^* = \sum_{n=1}^{N_s} \alpha_n^* \, y_n \, \bar{x}_n^{\,t} \, \bar{x} + w_o^* \qquad (23)$$

Where $N_s$ is the total number of support vectors identified.

# Non-linear Support vector Machines

- For non-linearly separable classes
- Based on Cover's theorem

### Covers theorem

A complex pattern classification problem cast in a higher dimensional space non linearly is moe likely to be linearly separable in that space than the lower dimensional space.

Let $D = \bar{x}_1, \bar{x}_2, ..., \bar{x}_N$. Each $\bar{x}_i$, which belongs to either $c_1$ or $c_2$, is of dimension $d$.

Objective: Find a surface that separates $c_1$ and $c_2$
Define $\Phi(x) = [\phi_1(x), \phi_2 x), ..., \phi_D(x)]^t$, where $D$ is the dimension of new space.
$D >> d$

$\bar{w}^t \Phi(\bar{x}) + w_0 > 0 \implies \bar{x}$ belongs to $c_1$
$\bar{w}^t \Phi(\bar{x}) + w_0 < 0 \implies \bar{x}$ belongs to $c_2$

Let $\bar{z}_n = \Phi(\bar{x}_n)$ ; $\bar{z}_n$ is of dimension $D$ and $\bar{x}_n$ is of dimension $d$.

$$\sum_{n-1}^{N} \alpha_n^* y_n \bar{z}_n^t \bar{z} + w_0^* = 0$$

$$\bar{w}^{*t} \bar{z} + w_0^* = 0$$

We define Inner product kernel as $k(\bar{x}_m, \bar{x}_n) = \Phi(\bar{x}_m)^t \Phi(\bar{x}_n)$
$K = [k(\bar{x}_m, \bar{x}_n)]_{m,n=1}^{N}$ is an $N * N$ matrix. It is semi-positive matrix.

- Linear kernel: $k(\bar{x}_m, \bar{x}_n) = \bar{x}_m^t \bar{x}_n|_{m,n=1}^{N}$
- Non-linear kernel: $k(\bar{x}_m, \bar{x}_n) = (a\bar{x}_m^t \bar{x}_n = +b)^p$ ( a polynomial kernel)
  $k(\bar{x}_m, \bar{x}_n) = (\bar{x}_m^t \bar{x}_n + b)^2 = (\bar{x}_m \bar{x}_n)^t + 2\bar{x}_m^t \bar{x}_n + 1$

Example of non-linear kernel:

Let $\bar{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$, $\bar{x}_m = \begin{bmatrix} x_{m1} \\ x_{m2} \end{bmatrix}$, and $\bar{x}_n = \begin{bmatrix} x_{n1} \\ x_{n2} \end{bmatrix}$

$(\bar{x}_m^t \bar{x}_n + 1)^2 = (x_{m1}x_{n1} + x_{m2}x_{n2} + 1)^2$

$= 1 + 2x_{m1}x_{n1} + 2x_{m2}x_{n2} + x_{m1}^2 x_{n1}^2 + x_{m2}^2 x_{n2}^2 + 2x_{m1}x_{n1}x_{m2}x_{n2}$

$\Phi(\bar{x}) = [1 \sqrt{(}2)x_1 \sqrt{(}2)x_2 x_1^2 x_2^2 \sqrt{(}2)x_1 x_2]^t$

$d = 2 \implies D = 6$
$d = 3 \implies D = 10$
$D = \frac{(p+d)!}{p!d!}$

## Mercers' theorem

$$k(\bar{x}, \bar{x}') = \sum_{i=1}^{\infty} \lambda_i \phi_i(\bar{x}) \phi_i(\bar{x}')$$
$$\phi_i - eigenfunction$$
$$\lambda_i - eigenvalue > 0$$

$$w^* = \sum_{n=1}^{N} \alpha_n^* y_n \phi(\bar{x}_n)$$

$$g(\bar{z}) = \bar{w}^{*t} z + w_0^*$$

$$g(x) = \sum_{n=1}^{N_s} \alpha_n^* y_n (\Phi(\bar{x}_n)^t \Phi(\bar{x})) + w_0^*$$