BITS F464: Machine Learning

26 Kernel SVM

Dr. Kamlesh Tiwari
Assistant Professor, Department of CSIS,
BITS Pilani, Pilani Campus, Rajasthan-333031 INDIA

March 24, 2021 ONLINE (Campus @ BITS-Pilani Jan-May 2021)

http://ktiwari.in/ml

Recap: Support Vector Machine (SVM)

SVM is liner decision machine; use sign($W^T x^{(i)} + b$) for decision

- With linearly separable data
- A fat margin classifier

$$\underset{W,b}{\operatorname{argmin}}\ W^T W$$

such that $y^{(i)}(W^T x^{(i)} + b) \geq 1$

When data is NOT linearly separable (try soft margin)

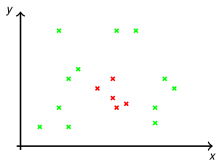
- Use slack variable $\xi \geq 0$ to optimise

$$\underset{W,b}{\operatorname{argmin}}\ \left[W^T W + C\sum_{i=1}^{m} \xi_i\right]$$

such that $y^{(i)}(W^T x^{(i)} + b) \geq 1 - \xi_i$

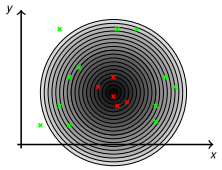
What is right C? Higher or lower value

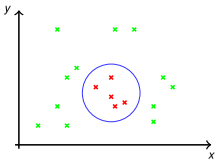
Truly non linearly separable data



- Some time the data is truly non separable
- Here small shift in decision boundary would not help
- We have to project data in suitable high dimensional space

- We have seen, transformation of 2D points $(x^{(i)}, y^{(i)})$ in 3D as $(x^{(i)}, y^{(i)}, z)$ where $z = (x^{(i)} - x_c)^2 + (y^{(i)} - y_c)^2$





Implications of feature transformation

- Recall $L(w,b) = \sum \alpha_i - \frac{1}{2}\sum \alpha_i \alpha_j y^{(i)} y^{(j)} x^{(i)} x^{(j)}$
- Computational cost is $O(d^2)$, if d increases we have to pay more
- **Kernal functions** can help in the $\phi(x)$ transformation
  They have a property that $F(x^{(i)}, x^{(j)}) = \phi(x^{(i)}).\phi(x^{(j)})$
- Function F depends on ϕ. It should be easy to compute
- So, the expansion of $\phi(x^{(i)})$ and $\phi(x^{(j)})$ is not needed for the computation of dot product $\phi(x^{(i)}).\phi(x^{(j)})$

$$L(w,b) = \sum \alpha_i - \frac{1}{2}\sum \alpha_i \alpha_j y^{(i)} y^{(j)} F(x^{(i)}, x^{(j)})$$

Example of a kernel function

$$K(x_i, x_j) = (1 + x_i.x_j)^2$$

- Consider 2-D vectors $x_i = (x_{i1}, x_{i2})$ and $x_j = (x_{j1}, x_{j2})$

$$\begin{aligned}(1 + x_i.x_j)^2 &= (1 + x_{i1}x_{j1} + x_{i2}x_{j2})^2 \\ &= (1 + x_{i1}^2 x_{j1}^2 + x_{i2}^2 x_{j2}^2 + 2x_{i1}x_{j1}x_{i2}x_{j2} + 2x_{i1}x_{j1} + 2x_{i2}x_{j2}) \\ &= (1, x_{i1}^2, x_{i2}^2, \sqrt{2}x_{i1}x_{i2}, \sqrt{2}x_{i1}, \sqrt{2}x_{i2}). \\ &\quad (1, x_{j1}^2, x_{j2}^2, \sqrt{2}x_{j1}x_{j2}, \sqrt{2}x_{j1}, \sqrt{2}x_{j2})\end{aligned}$$

- If $\phi(x) = (1, x_1^2, x_2^2, \sqrt{2}x_1x_2, \sqrt{2}x_1, \sqrt{2}x_2)$
- Vector is transformed to six dimensional space, and

$$K(x_i, x_j) = \phi(x_1).\phi(x_2)$$

Popularly used kernel functions

- Linear Kernal $K(x_i, x_j) = x_i.x_j$
- Polynomial Kernal $K(x_i, x_j) = (1 + x_i.x_j)^p$
- Gaussian Kernal (radial-basis function)

$$K(x_i, x_j) = e^{-\frac{||x_i - x_j||^2}{2\sigma^2}}$$

Note that $e^x = 1 + \frac{1}{1!}x + \frac{1}{2!}x^2 + \frac{1}{3!}x^3 + ... + \frac{1}{\infty!}x^\infty$
Projecting the data in an infinite dimensional space.

- Signoid Kernal $K(x_i, x_j) = \tanh(\beta_0 x_i.x_j + \beta_1)$

Functions satisfying Mercer's condition, can be used as kernel functions (positive semi-definite)

$$\begin{bmatrix} K(x_1,x_1) & K(x_1,x_2) & K(x_1,x_3) & \dots & K(x_1,x_n) \\ K(x_2,x_1) & K(x_2,x_2) & K(x_2,x_3) & \dots & K(x_2,x_n) \\ K(x_3,x_1) & K(x_3,x_2) & K(x_3,x_3) & \dots & K(x_3,x_n) \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ K(x_n,x_1) & K(x_n,x_2) & K(x_n,x_3) & \dots & K(x_n,x_n) \end{bmatrix}$$

Kernel functions

- Represents similarity between input vectors [1]
- Symmetric in nature
- For SVM, it brings efficiency and performance
- Multiple kernal functions could be composed to make new ones
- Mercer's condition

$$\sum_{i,j} K(x_i, x_j) c_i c_j \geq 0$$

[1] $\vec{a}.\vec{b} = ||\vec{a}||.||\vec{b}|| \cos\theta$

Thank You!

Thank you very much for your attention!

Queries ?

(Reference[2])

[2] [1] Text Book: *Machine Learning*, by *Tom M Mitchell* [2] Mod-01 Lec-29 Support Vector Machine https://www.youtube.com/watch?v=SRVswRH5Q7E