



CS-F415: Data Mining



Soft, and Kernel SVM



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

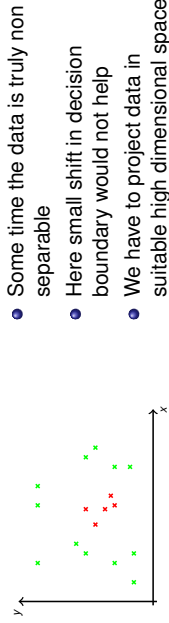
Feb 21, 2024

MW/F 4-00pm

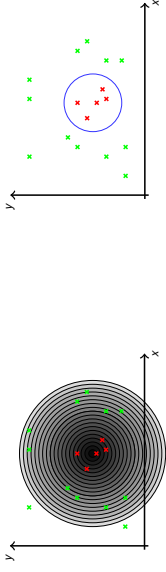
6101 @ BITS-Pilani [Jan-May 2024]

<http://katiwari.in/dm>

Truly non linearly separable data



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



Example of a kernel function

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

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

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

- 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

$$K(x_i, x_j) = \phi(x_i) \cdot \phi(x_j)$$

Soft Margin SVM

SVM: linear decision machine, using $\text{sign}(W^T x^{(i)} + b)$ for decision

- With linearly separable data
- A fat margin classifier

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

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

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

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

What is right C? Higher or lower value

Data Mining (CS-F415)

MW/F (4:00pm) 6101@BITS-Pilani

Lecture-09 (Feb 21, 2024)

2 / 8

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 $O(d)$, if d increases we have to pay more
- Kernel functions** can help in the $\phi(x)$ transformation
- They have a property that $F(x^{(i)}, x^{(j)}) = \phi(x^{(i)}) \cdot \phi(x^{(j)})$
- 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 $\phi(x^{(i)}) \cdot \phi(x^{(j)})$

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

Data Mining (CS-F415)

MW/F (4:00pm) 6101@BITS-Pilani

Lecture-09 (Feb 21, 2024)

4 / 8

Popularly used kernel functions

- Linear Kernel** $K(x_i, x_j) = x_i \cdot x_j$
- Polynomial Kernel** $K(x_i, x_j) = (1 + x_i \cdot x_j)^p$
- Gaussian Kernel** (radial-basis function)

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

Note that $e^x = 1 + \frac{1}{1!}x + \frac{1}{2!}x^2 + \frac{1}{3!}x^3 + \dots + \frac{1}{\infty!}x^\infty$

Projecting the data in an infinite dimensional space.

- Signoid Kernel** $K(x_i, x_j) = \tanh(\beta_0 x_i \cdot x_j + \beta_1)$

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

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

Data Mining (CS-F415)

MW/F (4:00pm) 6101@BITS-Pilani

Lecture-09 (Feb 21, 2024)

6 / 8

Kernel functions

Thank You!

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

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

Thank you very much for your attention!
Queries ?

¹ $\vec{a} \vec{b} = \|\vec{a}\| \cdot \|\vec{b}\| \cos \theta$