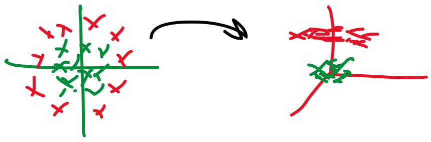
Recap

19 July 2023 17:04



$\boxed{\text{SVM}}$

L 1 — Hard margin SVM → L 2 — Soft margin SVM → L 3 — Kernels → L 4 — KKT inequ

L( )

Kernels ↓ constrained optimization prob (equality)

Lang_

KKT inequ ↓

$\boxed{\text{SVM dual formulation}}$ ↓

$\boxed{\text{Kernel trick}}$

↓ poly

↓ RBF

SVM Dual Formulation

19 July 2023 19:03

$x_i \cdot x_j$

$$\boxed{\max_{\alpha_i} \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j (x_i \cdot x_j)}$$

$\boxed{x_i \cdot x_j}$

$x_1 \quad x_1$

$y_i y_j = 1 \times 1$

$x_1 \cdot x_1 \rightarrow \boxed{x_{11}x_{11} + x_{12}x_{12}}$

$x_j \rightarrow \alpha_j$

$\boxed{\alpha_i > 0}$

$\alpha_i = 0$

25 terms

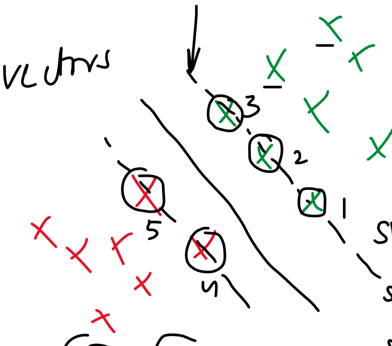
$\alpha_i \geq 0 \qquad \sum_{i=1}^{n} \alpha_i y_i = 0$

$\boxed{x_{11}x_{21} + x_{12}x_{22}}$

$\boxed{SV}$

$\alpha \begin{cases} \rightarrow \alpha_i = 0 \text{ for all non support vectors} \\ \rightarrow \alpha_i > 0 \text{ for all SV} \end{cases}$

$\underbrace{x_1 \quad x_2}$



	x_1	x_2	y
SV $\rightarrow$	x_{11}	x_{12}	1
SV $\rightarrow$	x_{21}	x_{22}	-1
SV $\rightarrow$	x_{31}	x_{32}	

5 SV

$$\max_{\alpha_i} \alpha_1 + \alpha_2 + \alpha_3 + \alpha_4 + \alpha_5 - \frac{1}{2}\left[- + - + \right]$$

j →	1	2	3	4	5
i: 1					
2					
3					
4					
5					

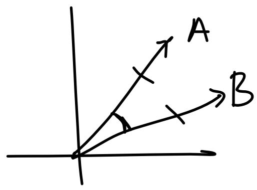
The Similarity Perspective

18 July 2023 08:52

Cosine similarity $= \dfrac{A \cdot B}{||A|| \, ||B||}$

$||A|| = 1$

$||B|| = 1$



Kernel trick

$= \boxed{A.B} \rightarrow$ dot product of A and B

x_i and x_j

$$\max_{\alpha_i} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \boxed{(x_i \cdot x_j)}$$

→ similarity

→ maximize the similarity of SV based on their sign

$$(x_i \cdot x_j) \rightarrow \boxed{sim(x_i, x_j)}$$

Kernels

Kernel SVM

19 July 2023 07:41

$$\max_{\alpha_i} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum \sum \alpha_i \alpha_j y_i y_j \boxed{K(x_i, x_j)}$$

$\hookrightarrow$ Kernel SVM

$\boxed{x_{11} x_{21} + x_{21} x_{22}}$

$x_i \cdot x_j$

$K(x_i, x_j) \rightarrow$ Kernel $\rightarrow$ similarity between x_i and x_j

$\hookrightarrow \boxed{x_i \cdot x_j} \rightarrow \boxed{x_i \cdot x_j}$ $\leftrightarrow$ creative version

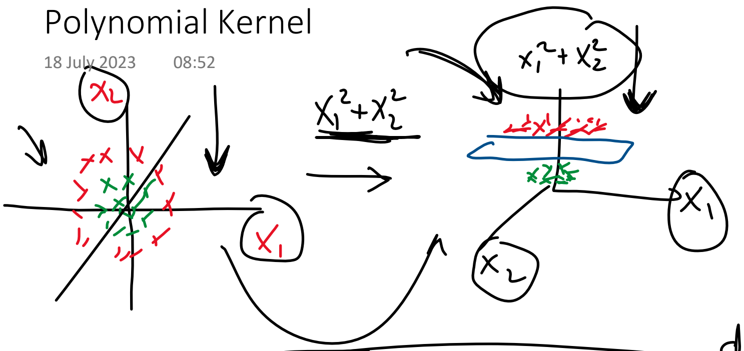
$\hookrightarrow$ linear SVM

Polynomial

RBF

Polynomial Kernel

18 July 2023 08:52



	x_1	x_2
$x_i \rightarrow$	x_{11}	x_{12}
$x_j \rightarrow$	x_{21}	x_{22}

$$\boxed{x_{11}x_{21} + x_{12}x_{22}}$$

$$\boxed{K(x_i, x_j)} = (\gamma + x_i \cdot x_j)^d \qquad \boxed{\gamma = 1} \quad d = 2, 3$$

$$(1 + x_i^T x_j)^k \qquad \gamma = 1, d = 2$$

$$= (1 + \underbrace{x_i \cdot x_j}_{})^2 = (1 + x_{11}x_{21} + x_{12}x_{22})^2$$

$$= 1 + x_{11}^2 x_{21}^2 + \underline{x_{12}^2 x_{22}^2} + \underline{2x_{11}x_{21}} + 2\underline{x_{12}x_{22}} + 2x_{11}\underline{x_{21}x_{12}x_{22}}$$

Kernel trick

↳ polynomial term

The Trick

19 July 2023 07:30

$$\left[1 + x_{11}^2 x_{21}^2 + x_{12}^2 x_{22}^2 + 2x_{11}x_{21} + 2x_{12}x_{22} + 2x_{11}x_{21}x_{12}x_{22}\right]$$

dot product of 2 vector

$$[1 \quad x_{11}^2 \quad x_{12}^2 \quad \sqrt{2}x_{11} \quad \sqrt{2}x_{12} \quad \sqrt{2}x_{11}x_{12}] \rightarrow x_1' \text{ 6d vec}$$

$$[1 \quad x_{21}^2 \quad x_{22}^2 \quad \sqrt{2}x_{21} \quad \sqrt{2}x_{22} \quad \sqrt{2}x_{12}x_{22}] \rightarrow \text{6 d vector } x_j'$$

(2) $x_1, x_2 \dots x_{10}$

2 dim → 10 dim

1 lakh (6d)

(1) $x_i(x_{11}\ x_{12}) \xrightarrow{ft} x_i'(6d)$, $x_j(x_{21}\ x_{22}) \rightarrow x_j'(6d)$ → store → $x_i' \cdot x_j' \rightarrow$ expression

(2) $x_i, x_j \rightarrow k(x_i, x_j) \rightarrow$ expression

Trick → logic/linear Polynomial deg 2

$x_1 \quad x_2$

$x_1 \mid x_1^2 \mid x_2 \mid x_2^2 \mid x_1 x_2 \mid$

What about the other Polynomial terms

19 July 2023 10:15

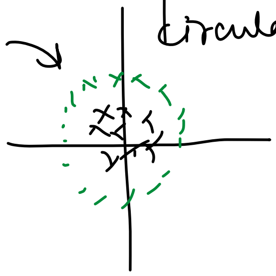
other shapes

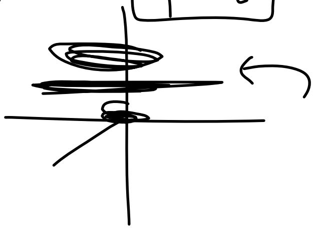
conic section

$$1 + x_{11}^2 x_{21}^2 + x_{12}^2 x_{22}^2 + 2x_{11}x_{21} + 2x_{12}x_{22} + 2x_{11}x_{21}x_{12}x_{22}$$

circular

$\boxed{x_1^2 + x_2^2}$





③

$x_1 \quad x_2$

$x_1^2 \quad x_2^2 \quad 2x_1x_2$

$x_1^3 \quad x_2^3 \quad x_1^2 x_3 \quad x_3^2 x_1$

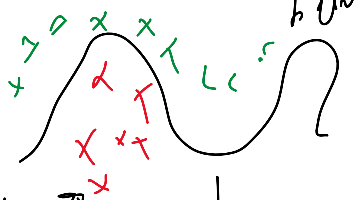
x_1

RBF Kernel

18 July 2023 08:52

Radial Basis function { Normal dist }

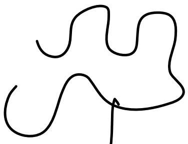
$\|x_i - x_j\| \rightarrow$ euclidean dist b/w x_i and x_j



→ popular

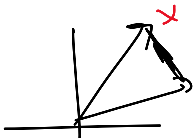
→ Best out of the box kernel

→ powerful



polynomial

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



neural networks

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

$$e^{-\gamma \|x_i - x_j\|^2}$$

γ → hyperparameter

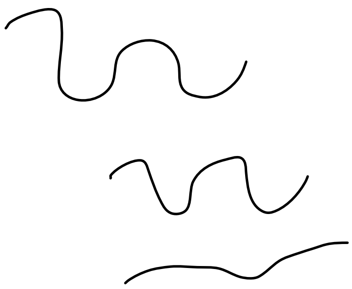
$$\boxed{\gamma = 1/2\sigma^2}$$

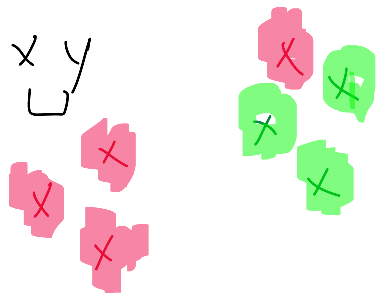
$K \propto \frac{1}{dist}$

↑ similar

- Non-linear Transformations: The RBF kernel enables the use of non-linear transformations, which can map the original feature space to a higher-dimensional space where the data becomes linearly separable. This is particularly useful for problems where the decision boundary is not linear.
- Local Decisions: Unlike some other kernels, the RBF kernel makes "local" decisions. That is, the effect of each data point is limited to a certain region around that point. This can make the model more robust to outliers and create complex decision boundaries.
- Flexibility: The RBF kernel has a parameter γ(related to the standard deviation of the Gaussian distribution) that determines the complexity of the decision boundary. By tuning this parameter, we can adjust the trade-off between bias and variance, allowing for a flexible range of decision boundaries.
- Universal Approximation Property: The RBF kernel has a property known as the "universal approximation" property, meaning it can approximate any continuous function to a certain degree of accuracy given enough data points. This makes it highly versatile and capable of modelling a wide variety of relationships in data.
- General-Purpose: The RBF kernel does not make any strong assumptions about the data and can therefore be a good choice in many different situations, making it a versatile, general-purpose kernel.

γ



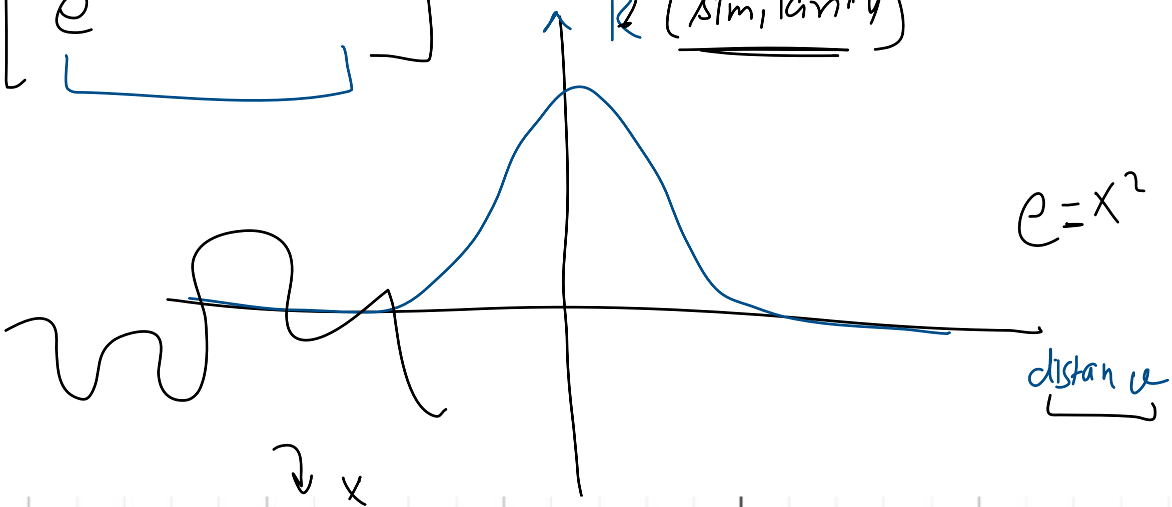


distance

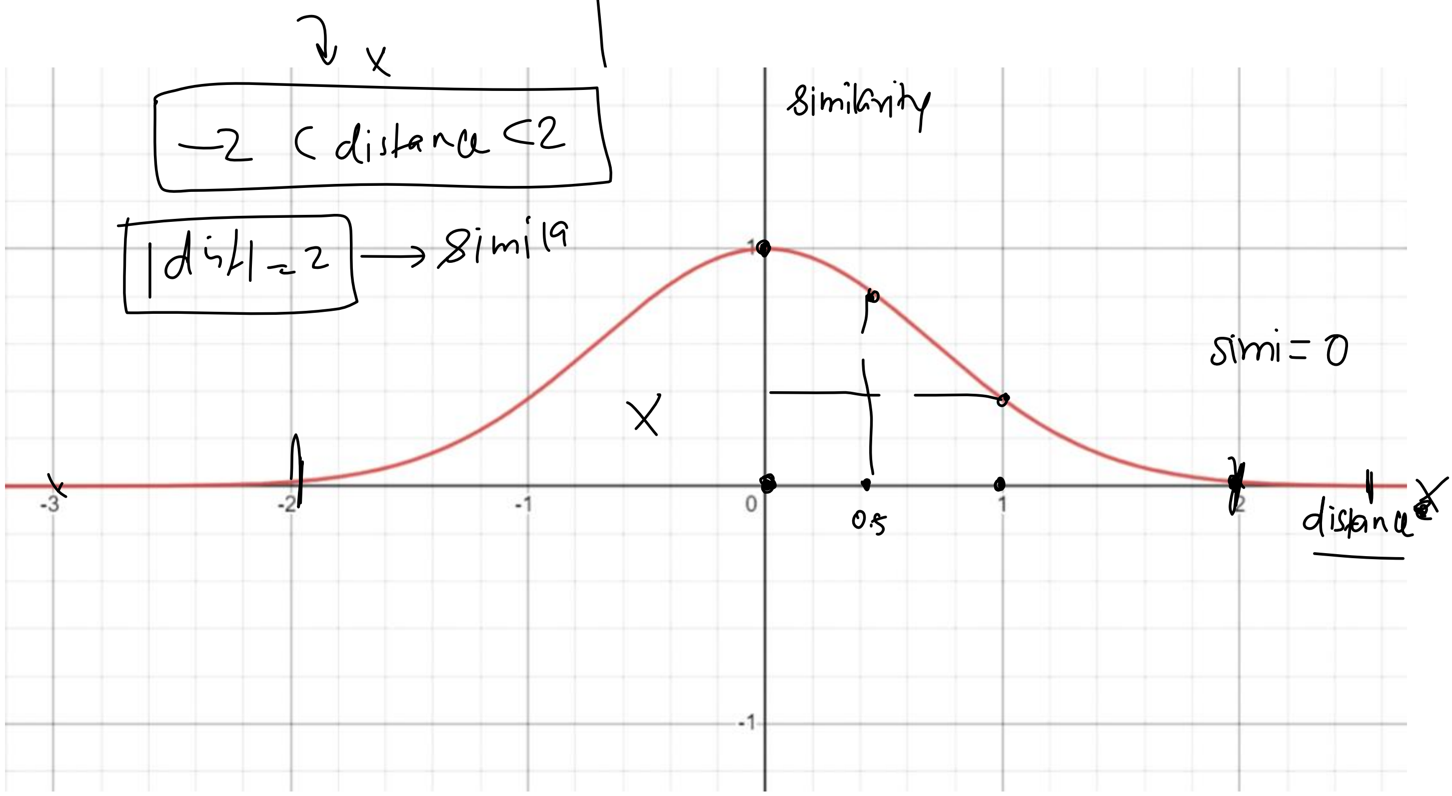
Local Decision Boundary

19 July 2023 18:28

$\left[\underbrace{e^{-distance^2}} \right] \to x$

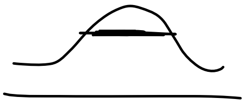


$e = x^2$



Effect of Gamma

19 July 2023 11:11



σ

$$e^{-\frac{\|x_i - x_j\|^2}{2\sigma^2}}$$

$$\boxed{\frac{1}{2\sigma^2} = \boxed{\gamma}}$$

The parameter γ in the Radial Basis Function (RBF) kernel of a Support Vector Machine (SVM) is a hyperparameter that determines the spread of the kernel and therefore the decision region.

$\sigma \downarrow$ $\sigma \uparrow$

The effect of γ can be summarized as follows:

- If γ is too large, the exponential will decay very quickly, which means that each data point will only have an influence in its immediate vicinity. The result is a more complex decision boundary, which might overfit the training data.
- If γ is too small, the exponential will decay slowly, which means that each data point will have a wide range of influence. The decision boundary will therefore be smoother and more simplistic, which might underfit the training data.

$\sigma^2 = 1$

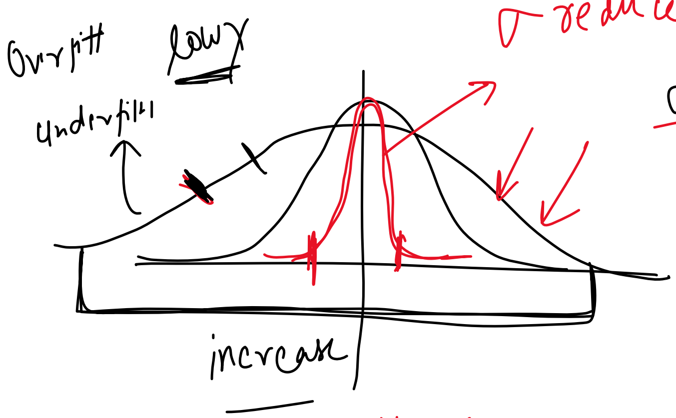
In a sense, γ in the RBF kernel plays a role similar to that of the inverse of the regularization parameter: it controls the trade-off between bias (underfitting) and variance (overfitting). High γ values can lead to high variance (overfitting) due to more flexibility in shaping the decision boundary, while low γ values can lead to high bias (underfitting) due to a more rigid, simplistic decision boundary.

Tuning the γ parameter using cross-validation or a similar technique is typically a crucial step when training SVMs with an RBF kernel.

$\sigma = 10$ $\sigma^2 = 100$

$\sigma = 0.1$ $\sigma^2 = 0.01$

σ reduce $\rightarrow 0.1$



Bias variance tradeoff

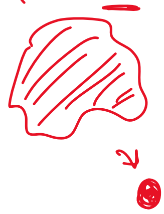
$\sigma \uparrow$ range $\uparrow$ accuracy locality $\uparrow$

$\sigma \rightarrow$ hyperparameter

$\gamma \downarrow \rightarrow$ locality $\uparrow$

$\gamma \uparrow \rightarrow$ locality $\downarrow$

$\sigma \propto \frac{1}{\gamma}$



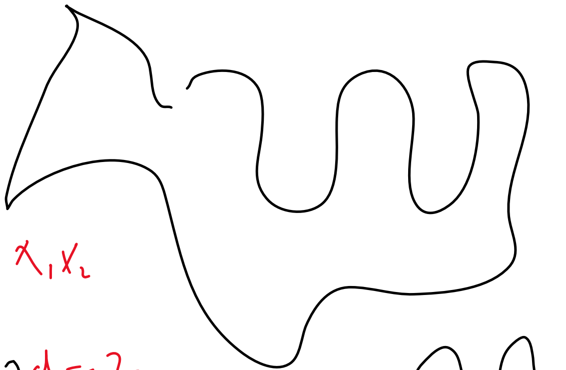
$\sigma \downarrow \gamma \uparrow \rightarrow$ overfitting

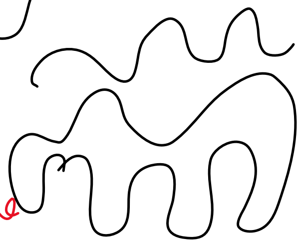
$\sigma \uparrow \gamma \downarrow \rightarrow$ underfitting

Relationship Between RBF and Polynomial Kernel

19 July 2023 14:36

Infinite Dimensional Mapping: The RBF kernel implicitly maps input data to an infinite-dimensional feature space, which allows for even greater flexibility in forming decision boundaries





degree → 2

$\boxed{x_1 \quad x_2}$

$x_1 \quad x_2 \quad x_1^2 \quad x_2^2 \quad x_1 x_2$

$\boxed{x_1 \quad x_2}$ → ∞ dim vector

$x_1 \quad x_2 \quad x_1^2 \quad x_2^2 \quad x_1 x_2$ → d = 2

$x_1^3 \quad x_2^2 \quad x_1^2 x_2 \quad x_2^2 x_1$ → d = 3

$x_1^4 \quad x_2^4 \quad x_1^3 x_2^1 \quad x_3^2 x_2^2 \quad x_2^3 x^1$ → ∞d

infite dim

d = 5

d = 6

⋮

d = ∞

$$e^{-\gamma \|x_i - x_j\|^2}$$

$$(1 + x_i \cdot x_j)^{2^4}$$

rbf → poly

$\sigma = 1 \quad \sigma^2 = 1$

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

$$= e^{-\frac{\|x_i - x_j\|^2}{2}}$$

$$= e^{-\frac{(x_i - x_j)^T (x_i - x_j)}{2}} = e^{-\frac{(x_i^T - x_j^T)(x_i - x_j)}{2}}$$

$$= e^{-\frac{[x_i^T x_i - x_i^T x_j - x_j^T x_i + x_j^T x_j]}{2}}$$

$x_{11} x_{21} + x_{12} x_{22}$

$x_i = [x_{11} \quad x_{12}]$

$x_j = [x_{21} \quad x_{22}]$

$x_i^T x_j =$

$x_j^T x_i =$

$$= e^{-\frac{[x_i^T x_i + x_j^T x_j - x_i^T x_j - x_j^T x_i]}{2}}$$

$$= e^{-\frac{[x_i^T x_i + x_j^T x_j - 2 x_i^T x_j]}{2}}$$

$\boxed{x_i \cdot x_j}$

$$= e^{-\frac{1}{2}[x_i^T x_i + x_j^T x_j]} \, e^{x_i^T x_j}$$

$$e^{1 + x_i^T x_j - 1}$$

$$= C e^{1 + x_i^T x_j - 1}$$

$$= C \, e^{1 + x_i^T x_j} \, e^{-1}$$

$$= c' \, e^{1 + x_i^T x_j} \rightarrow = C' \sum_{K=0}^{\infty} \frac{\boxed{(1 + x_i^T x_j)^K}}{K!}$$

$$C' \sum_{K=0}^{\infty} \frac{K_{poly}(x_i, x_j)^K}{K!}$$

$$e^X = \sum_{K=0}^{\infty} \frac{X^K}{K!}$$

$$C' \left[1 + \frac{(1 + x_i^T x_j)}{1!} + \frac{(1 + x_i^T x_j)^2}{2!} + \frac{(1 + x_i^T x_j)^3}{3!} + \ldots + \frac{(1 + x_i^T x_j)^\infty}{\infty!} \right]$$

$$rbf = \sum_{i=2}^{\infty} K_{poly}(x_i, x_j)$$

Custom Kernels

19 July 2023 16:27

Custom kernels

1. **String kernels**: These are used for classifying text or sequences, where the input data is not numerical. String kernels measure the similarity between two strings. For example, a simple string kernel might count the number of common substrings between two strings.
2. **Chi-square kernel**: This kernel is often used in computer vision problems, especially for histogram comparison. It's defined as $K(x, y) = \exp(-\gamma\chi^2(x, y))$, where $\chi^2(x, y)$ is the chi-square distance between the histograms x and y.
3. **Intersection kernel**: This is another kernel commonly used in computer vision, which computes the intersection between two histograms (or generally non-negative feature vectors).
4. **Hellinger's kernel**: Hellinger's kernel, or Bhattacharyya kernel, is used for comparing probability distributions and is popular in image recognition tasks.
5. **Radial basis function network (RBFN) kernels**: These are similar to the standard RBF kernel, but the centers and widths of the RBFs are learned from the data, rather than being fixed a priori.
6. **Spectral kernels**: These kernels use spectral analysis techniques to compare data points. They can be particularly useful for dealing with cyclic or periodic data.

Doubts

20 July 2023 15:42

1. What is complementary Slackness
2. Why did min change to max?
3. Observations
    a. Effect on high dimensional data
4. Prediction is done?