SVM Kernel trick

JrPhy

Introduction

- So far SVM can just solve linear separable data, then how can we deal with the nonlinear separable data?
- One is accept there are some mistakes, its so called soft-margin SVM.
- The other is use nonlinear function. its so called kernel SVM.
- Here we introduce the polynomial kernel and Gaussian kernel.

Polynomial kernel

- In the original SVM, we calculate $\min_{\alpha \geq 0} \left(\frac{1}{2} \left\| \sum_j \sum_i \alpha_i \alpha_j y_i y_j \vec{x}^T \vec{x}' \right\|^2 - \sum_i \alpha_i \right)$

- The $\boldsymbol{x}^T\boldsymbol{x}$ is inner product, it can be seen as linear, here we take 2nd polynomial as example. Suppose f is a function such that $f : R^2 \rightarrow R^3$ and dim($\boldsymbol{x}$) = 2, so there are some term: $(1, x_1, x_1, x_1^2, x_1x_2, x_2^2)$, so the inner product is

$$\vec{x}^T \vec{x}' \rightarrow f(x)^T f(x) = 1 + \sum_{i=1}^{2} x_i x_i' + \sum_{i=1}^{2} \sum_{j=1}^{2} x_i x_i' x_j x_j'$$

$$= 1 + \sum_{i=1}^{2} x_i x_i' + \sum_{i=1}^{2} x_i x_i' \sum_{j=1}^{2} x_j x_j' = 1 + \vec{x}^T \vec{x}' + (\vec{x}^T \vec{x}')(\vec{x}^T \vec{x}')$$

Polynomial kernel

- After transforming to 2nd order polynomial, it still calculate the inner product $\boldsymbol{x}^{\boldsymbol{T}}\boldsymbol{x}$.
- So the kernel is **transform + inner product**, here we use the symbol Φ to represent the kernel.
- So all $\boldsymbol{x} \rightarrow \boldsymbol{\Phi}(\boldsymbol{x})$ are what we want to calculate.
- $\min_{\alpha \geq 0} \left(\frac{1}{2} \left\| \sum_j \sum_i \alpha_i \alpha_j y_i y_j \Phi(\vec{x})^T \Phi(\vec{x}') \right\|^2 - \sum_i \alpha_i \right)$, $g_{\mathrm{SVM}}(\boldsymbol{x}) = \mathrm{sign}(\boldsymbol{w}^{\boldsymbol{T}}\, \boldsymbol{\Phi}(\boldsymbol{x}) + \boldsymbol{b})$
- In general the polynomial kernel has the form:
    - $K_Q(\mathbf{x}, \mathbf{x}') = (\zeta + \gamma \mathbf{x}^{\boldsymbol{T}}\mathbf{x})^Q,\ \gamma > 0,\ \zeta \geq 0$
- If you set $\zeta = \gamma = Q = 1$, it's called linear kernel.

Gaussian kernel

- The other kernel is Gaussian kernel, it uses Gaussian distribution as the basis.
- $\boldsymbol{x^T x'} =$ $\exp(-(x-x')^2) = \exp(-x^2)\exp(-x'^2)\exp(-2xx')$

$$= \exp(-x^2)\exp(-x'^2)\sum_{i=0}^{\infty}\frac{(2xx')^i}{i!}$$

$$= \sum_{i=0}^{\infty}\exp(-x^2)\sqrt{\frac{2^i}{i!}}x^i \exp(-x'^2)\sqrt{\frac{2^i}{i!}}x'^i$$

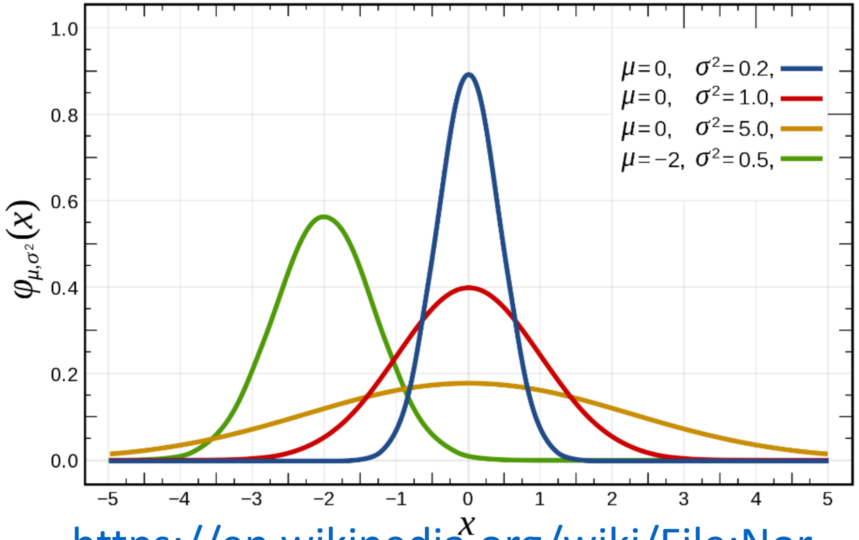
- So Gaussian kernel is $\exp(-x^2)\sqrt{\frac{2^i}{i!}}x^i$

Gaussian kernel

- In general, the Gaussian kernel is written as
    - $K(\boldsymbol{x}, \boldsymbol{x'}) = \exp(-\gamma\|\boldsymbol{x} - \boldsymbol{x'}\|^2), \gamma > 0$
- Gaussian distribution:

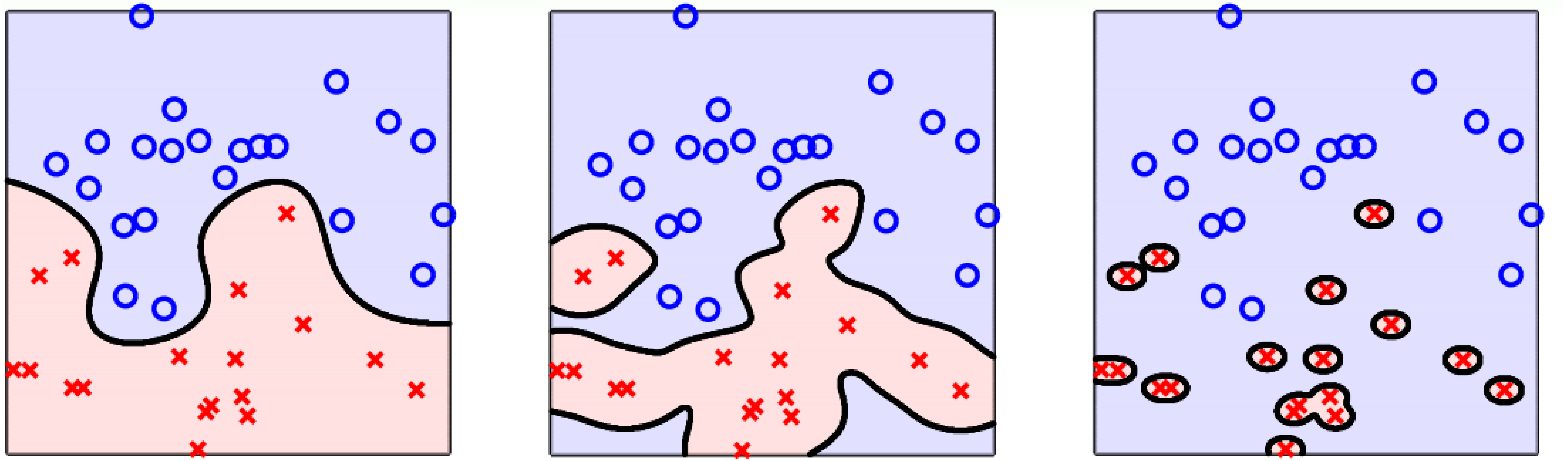
$$N(\mu, \sigma) = \frac{1}{\sqrt{2\pi\sigma}} \exp\left(-\frac{\|x - \mu\|^2}{2\sigma^2}\right)$$

- The central of the distribution is at $x = \mu$, and the width is determined by σ.
- The central is on the support vector, and the distance is determined by γ,



https://en.wikipedia.org/wiki/File:Normal_Distribution_PDF.svg

Gaussian kernel

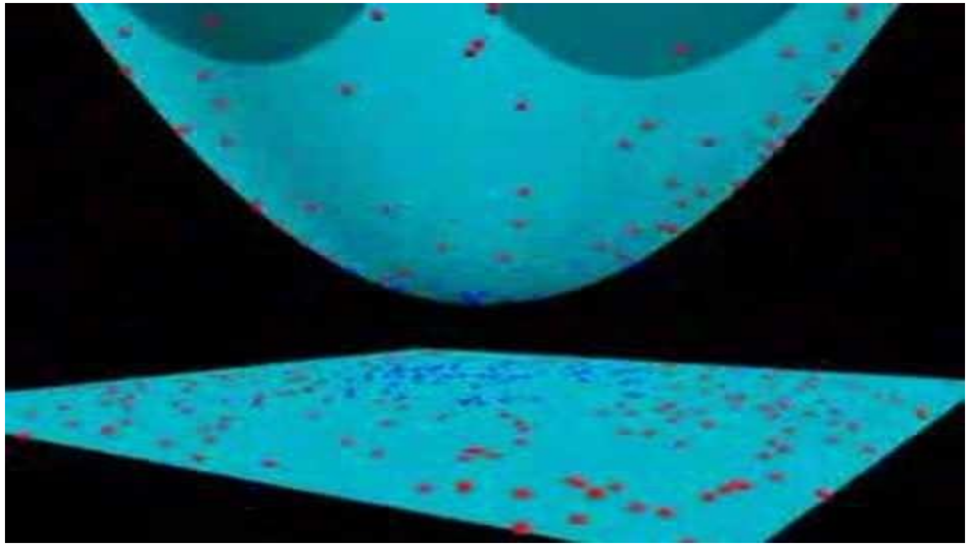


$\exp(-1\|\mathbf{x}-\mathbf{x}'\|^2)$ $\exp(-10\|\mathbf{x}-\mathbf{x}'\|^2)$ $\exp(-100\|\mathbf{x}-\mathbf{x}'\|^2)$

Hsuan Tien Lin, mltech/203_handout

Viewpoint of Gaussian and polynomial kernel

- We use kernel to transform our data to higher dimension space, then find a plane to separate the data. Just watch the video below.



Compare

	Linear $K(x, x') = x^T x'$	**Polynomial** $K_Q(x, x') = (\zeta + \gamma x^T x)^Q$, $\gamma > 0, \zeta \geq 0$	**Guass** $K(x, x') =$ $\exp(-\gamma \|x - x'\|^2)$, $\gamma > 0$
Pros	1. Easy to interprets 2. Fast: QP solver 3. Hard to over-fitted	Can use nonlinear separable data.	Most powerfu;
Cons	Can use only linear separable data	1. Slower than linear 2. Not numerical stable if $\|\zeta + \gamma \mathbf{x}^T \mathbf{x}\|$ too big or small	1. Slowest 2. Easy to over-fitted 3. Hard to interprets

Use your own kernel

- Polynomial kernel and Gaussian kernel are used widely, you can use other kernels.
- Kernel is from the inner product, so it satisfies:
    - $K(x, y) = K(y, x)$ and K is semi-definite matrix