Foundations of Machine Learning (ECE 5984)

- Kernel Methods-

Eunbyung Park

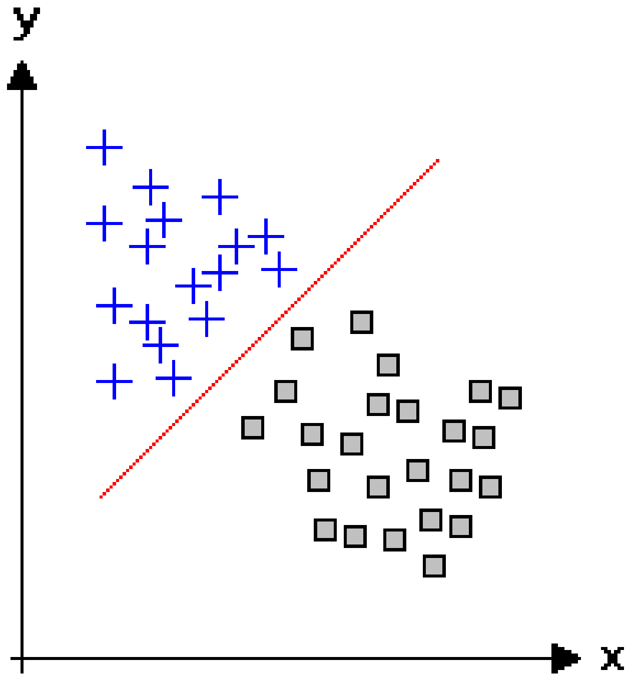
Assistant Professor

School of Electronic and Electrical Engineering

Eunbyung Park (silverbottlep.github.io)

Non-Linearity

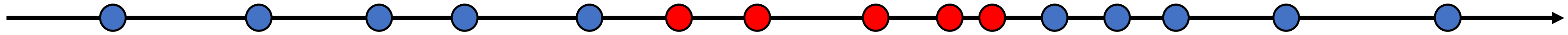
- The real world is not linear



Feature Transformation

- Can we make a non-linear decision boundary w/ linear method?

$x \in \mathbb{R}$

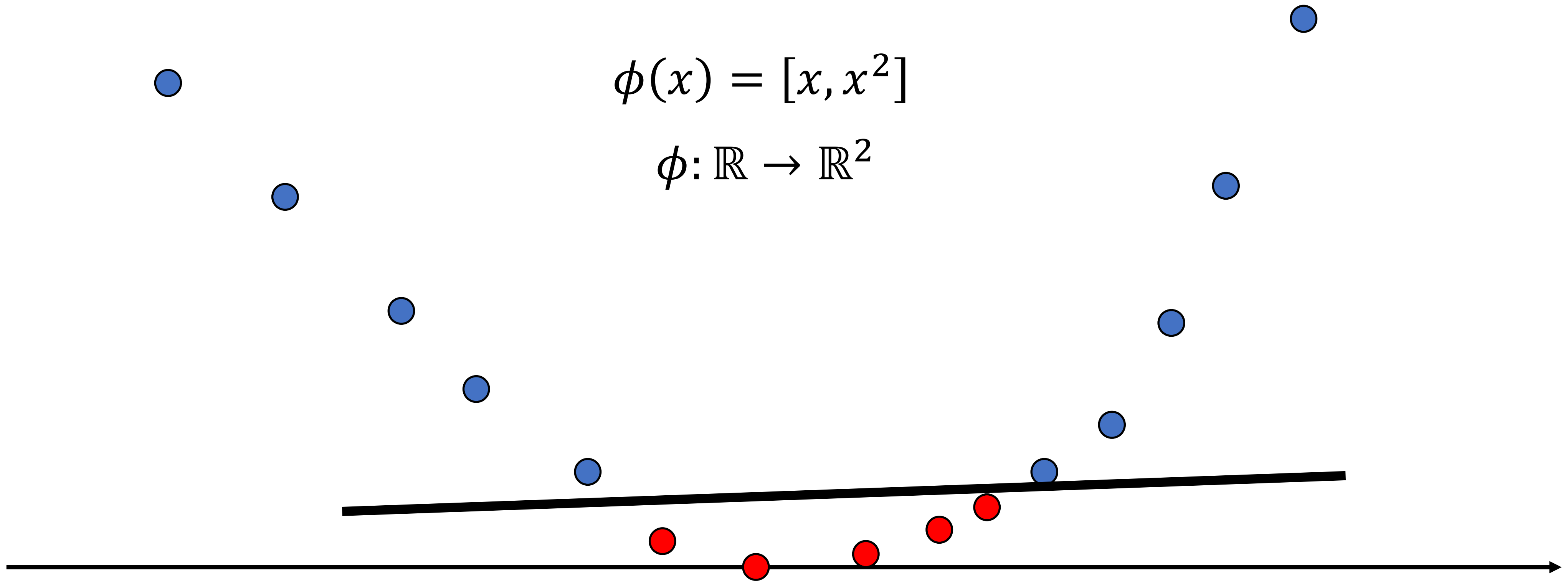


Feature Transformation

- Can we make a non-linear decision boundary w/ linear method?

$$\phi(x) = [x, x^2]$$

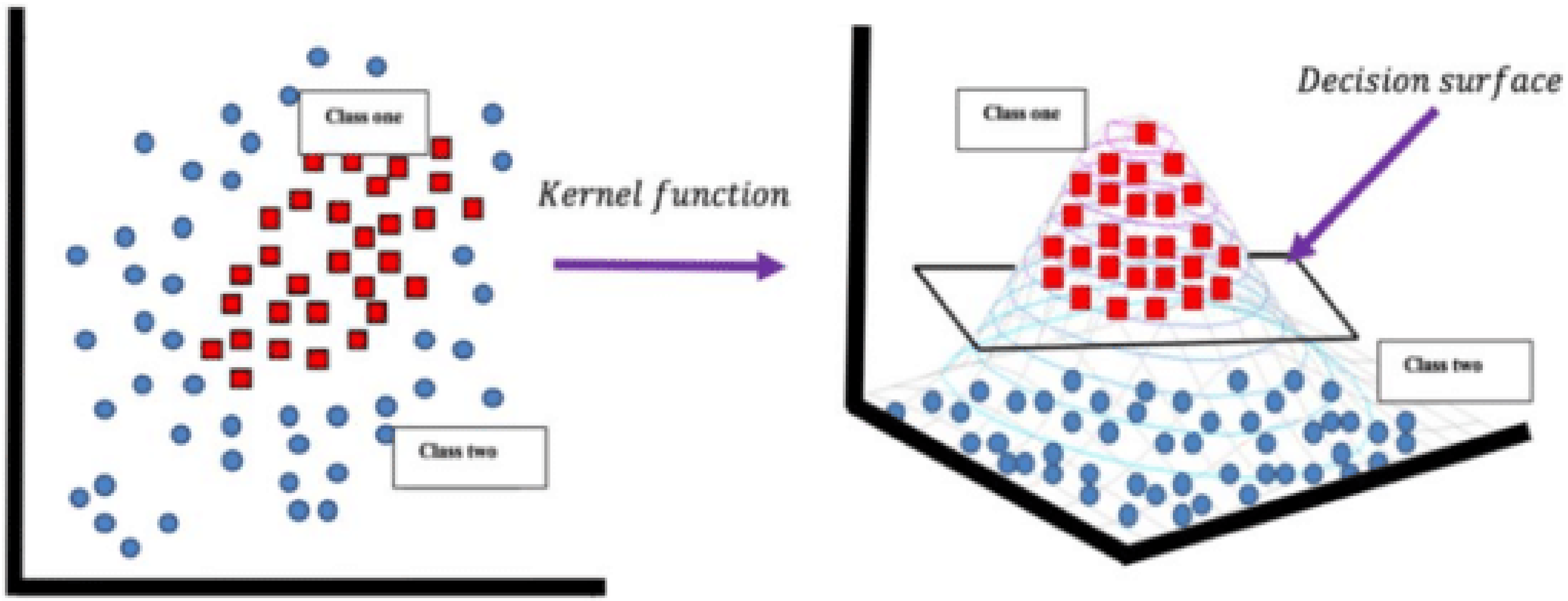
$$\phi: \mathbb{R} \to \mathbb{R}^2$$



Feature Transformation

- Can we make a non-linear decision boundary w/ linear method?

$$\phi(x) = [x_1, x_2, x_1^2 + x_2^2]$$



Feature Transformation

SVM with polynomial kernel visualization (HD) (youtube.com)

Feature Transformation

- Feature transformation
- Still linear in θ!

$$h_\theta(x) = \theta^\top \phi(x)$$

- Feature explosion (-)
  - more computationally expensive to train
  - more training examples needed to avoid overfitting

Kernels

Kernel Methods

- Kernel methods are based on pairwise comparisons
- When the feature is high-dimensional, and we only want to compute the inner product between feature vectors

Kernel Example (1)

$$\phi(x) = \begin{bmatrix} x_1^2 \\ \sqrt{2}x_1 x_2 \\ x_2^2 \end{bmatrix}$$

$$\phi(x)^\top \phi(z) = x_1^2 z_2^2 + 2x_1 z_1 x_2 z_2 + x_2^2 z_2^2$$

$$= (x_1 z_1 + x_2 z_2)^2 = (x^\top z)^2$$

$$= k(x, z)$$

Kernel Example (2)

$$\phi(x) = \begin{bmatrix} 1 \\ x_1 \\ x_2 \\ \dots \\ x_1^2 \\ x_2^2 \\ x_1 x_2 \\ x_1 x_3 \\ \dots \\ x_1^3 \\ \dots \end{bmatrix}$$

$$\phi(x)^\top \phi(z) = 1 + \sum_i x_i z_i + \sum_{i,j} x_i x_j z_i z_j + \sum_{i,j,k} x_i x_j x_k z_i z_j z_k$$

$$= 1 + x^\top z + (x^\top z)^2 + (x^\top z)^3$$

$$= k(x, z)$$

Kernel Example (3)

Hint: $\exp(x) = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \ \ldots$

$$\phi(x) = e^{-\frac{x^2}{2\sigma^2}} \begin{bmatrix} 1 \\ \sqrt{\frac{1}{1!\,\sigma^2}}\,x \\ \sqrt{\frac{1}{2!\,\sigma^4}}\,x^2 \\ \sqrt{\frac{1}{3!\,\sigma^6}}\,x^3 \\ \sqrt{\frac{1}{4!\,\sigma^8}}\,x^4 \\ \ldots \end{bmatrix}$$

$$\phi(x)^\top \phi(z)$$

Kernel Example (3)

Hint: $\exp(x) = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \ \dots$

$$\phi(x) = e^{-\frac{x^2}{2\sigma^2}} \begin{bmatrix} 1 \\ \sqrt{\frac{1}{1!\,\sigma^2}}x \\ \sqrt{\frac{1}{2!\,\sigma^4}}x^2 \\ \sqrt{\frac{1}{3!\,\sigma^6}}x^3 \\ \sqrt{\frac{1}{4!\,\sigma^8}}x^4 \\ \dots \end{bmatrix}$$

$$\phi(x)^\top \phi(z)$$

$$= \exp\left(-\frac{x^2+z^2}{2\sigma^2}\right)\left(1 + \frac{1}{1!\,\sigma^2}xz + \frac{1}{2!\,\sigma^4}x^2z^2 + \frac{1}{3!\,\sigma^6}x^3z^3 + \cdots\right)$$

$$= \exp\left(-\frac{x^2+z^2}{2\sigma^2}\right)\exp\left(\frac{xz}{\sigma^2}\right) = \exp\left(-\frac{(x+z)^2}{2\sigma^2} + \frac{xz}{\sigma^2}\right)$$

$$= \exp\left(-\frac{x^2+z^2}{2\sigma^2} + \frac{2xz}{2\sigma^2}\right) = \exp\left(-\frac{(x-z)^2}{2\sigma^2}\right)$$

Kernel Linear Regression

Gradient Descent in Linear Regression

$$L(\theta) = \frac{1}{2}\sum_{i=1}^{N}\left(y^{(i)} - \theta^\top x^{(i)}\right)^2$$

$$L(\theta) = \frac{1}{2}\sum_{i=1}^{N}\left(y^{(i)} - \theta^\top \phi\left(x^{(i)}\right)\right)^2$$

$$\nabla_\theta L = -\sum_{i=1}^{N}\left(y^{(i)} - \theta^\top x^{(i)}\right)x^{(i)}$$

$$\nabla_\theta L = -\sum_{i=1}^{N}\left(y^{(i)} - \theta^\top \phi\left(x^{(i)}\right)\right)\phi\left(x^{(i)}\right)$$

$$\theta \coloneqq \theta + \alpha\sum_{i=1}^{N}\left(y^{(i)} - \theta^\top x^{(i)}\right)x^{(i)}$$

$$\theta \coloneqq \theta + \alpha\sum_{i=1}^{N}\left(y^{(i)} - \theta^\top \phi\left(x^{(i)}\right)\right)\phi\left(x^{(i)}\right)$$

Gradient Descent in Linear Regression

- At any time t, θ can be represented as a linear combination of input features
  - The gradient is a linear combination of input features

$$\theta = \sum_{i=1}^{N} \beta_i \phi\left(x^{(i)}\right)$$

N: The number of data

$\beta \in \mathbb{R}^N$

Gradient Descent in Linear Regression

- Proof by induction
  1. Base case
  2. Assume it is true at **t**
  3. Show that it is true at **t+1**

$$\theta = \sum_{i=1}^{N} \beta_i \phi\left(x^{(i)}\right)$$

1. Base case: "*prove that the statement holds for the first natural number*"
   - It's convex, we can start from anywhere, so, we can set $\theta := 0$ at time 0, and all $\beta_i = 0$.

Gradient Descent in Linear Regression

- Proof by induction
  1. Base case
  2. Assume it is true at **t**
  3. Show that it is true at **t+1**

$$\theta = \sum_{i=1}^{N} \beta_i \phi\left(x^{(i)}\right)$$

$$\theta^{(t+1)} = \theta^{(t)} + \alpha \sum_{i=1}^{N} \left(y^{(i)} - {\theta^{(t)}}^{\top} \phi\left(x^{(i)}\right)\right) \phi\left(x^{(i)}\right)$$

($\theta^{(t)}$: θ at t)

Gradient Descent in Linear Regression

- Proof by induction
  1. Base case
  2. Assume it is true at **t**
  3. Show that it is true at **t+1**

$$\theta = \sum_{i=1}^{N} \beta_i \phi\left(x^{(i)}\right)$$

$$\theta^{(t+1)} = \theta^{(t)} + \alpha \sum_{i=1}^{N} \left(y^{(i)} - {\theta^{(t)}}^{\top} \phi\left(x^{(i)}\right)\right) \phi\left(x^{(i)}\right)$$

($\theta^{(t)}$: θ at t)

$$= \sum_{j=1}^{N} \beta_j^{(t)} \phi\left(x^{(j)}\right) + \alpha \sum_{i=1}^{N} \left(y^{(i)} - {\theta^{(t)}}^{\top} \phi\left(x^{(i)}\right)\right) \phi\left(x^{(i)}\right)$$

$$= \sum_{i=1}^{N} \beta_i^{(t)} \phi\left(x^{(i)}\right) + \alpha \left(y^{(i)} - {\theta^{(t)}}^{\top} \phi\left(x^{(i)}\right)\right) \phi\left(x^{(i)}\right)$$

$$= \sum_{i=1}^{N} \left(\beta_i^{(t)} + \alpha \left(y^{(i)} - {\theta^{(t)}}^{\top} \phi\left(x^{(i)}\right)\right)\right) \phi\left(x^{(i)}\right)$$

$\beta_i^{(t+1)}$

Gradient Descent in Linear Regression

$$\theta^{(t)} = \sum_{i=1}^{N} \underbrace{\left(\beta_i^{(t)} + \alpha\left(y^{(i)} - \theta^{(t)\top}\phi\left(x^{(i)}\right)\right)\right)}_{\beta_i^{(t+1)}} \phi\left(x^{(i)}\right)$$

$$\beta_i^{(t+1)} = \beta_i^{(t)} + \alpha\left(y^{(i)} - \left(\sum_{j=1}^{N} \beta_j^{(t)}\phi\left(x^{(j)}\right)\right)^{\top} \phi\left(x^{(i)}\right)\right)$$

$$= \beta_i^{(t)} + \alpha\left(y^{(i)} - \sum_{j=1}^{N} \beta_j^{(t)}\phi\left(x^{(j)}\right)^{\top}\phi\left(x^{(i)}\right)\right)$$

Gradient Descent in Linear Regression

- We can precompute all inner products!
- Inner products can be very efficient

$$\beta_i \leftarrow \beta_i + \alpha \left(y^{(i)} - \sum_{j=1}^{N} \beta_j \phi\left(x^{(j)}\right)^\top \phi\left(x^{(i)}\right) \right)$$

Gradient Descent in Linear Regression

- Vector notation

$$\beta_i = \beta_i + \alpha\left(y^{(i)} - \sum_{j=1}^{N}\beta_j\phi\left(x^{(j)}\right)^\top\phi\left(x^{(i)}\right)\right)$$

$$K\left(x^{(i)}, x^{(j)}\right) = \phi(x)^\top\phi(z)$$

$$K_{ij} = K\left(x^{(i)}, x^{(j)}\right)$$

$$\beta_i = \beta_i + \alpha\left(y^{(i)} - \sum_{j=1}^{N}\beta_j K\left(x^{(j)}, x^{(i)}\right)\right)$$

$$\beta = \beta + \alpha(Y - K\beta)$$

Gradient Descent in Linear Regression

- Testing with a new data

$$\theta^\top \phi(x^{new}) = \sum_{j=1}^{N} \beta_j \phi\left(x^{(j)}\right)^\top \phi(x^{new})$$

$$= \sum_{j=1}^{N} \beta_j K\left(x^{(j)}, x^{new}\right)$$

- Only kernel computation
- No need to compute θ and $\phi(x^{new})$ explicity

Kernel Logistic Regression

Gradient Descent in Logistic Regression

$$L(\theta) = \frac{1}{2}\sum_{i=1}^{N}\left(y^{(i)} - \sigma\left(\theta^\top x^{(i)}\right)\right)^2$$

$$L(\theta) = \frac{1}{2}\sum_{i=1}^{N}\left(y^{(i)} - \sigma\left(\theta^\top \phi\left(x^{(i)}\right)\right)\right)^2$$

$$\nabla_\theta L = -\sum_{i=1}^{N}\left(y^{(i)} - \sigma\left(\theta^\top x^{(i)}\right)\right) x^{(i)}$$

$$\nabla_\theta L = -\sum_{i=1}^{N}\left(y^{(i)} - \sigma\left(\theta^\top \phi\left(x^{(i)}\right)\right)\right) \phi\left(x^{(i)}\right)$$

$$\theta \coloneqq \theta + \alpha\sum_{i=1}^{N}\left(y^{(i)} - \sigma\left(\theta^\top x^{(i)}\right)\right) x^{(i)}$$

$$\theta \coloneqq \theta + \alpha\sum_{i=1}^{N}\left(y^{(i)} - \sigma\left(\theta^\top \phi\left(x^{(i)}\right)\right)\right) \phi\left(x^{(i)}\right)$$

Kernel Linear Regression vs. Kernel Logistic Regression

Kernel Linear Regression

$$\beta_i \leftarrow \beta_i + \alpha \left(y^{(i)} - \sum_{j=1}^{N} \beta_j \phi\left(x^{(j)}\right)^{\top} \phi\left(x^{(i)}\right) \right)$$

$$\theta^{\top} \phi(x^{new}) = \sum_{j=1}^{N} \beta_j \phi\left(x^{(j)}\right)^{\top} \phi(x^{new})$$

$$= \sum_{j=1}^{N} \beta_j K\left(x^{(j)}, x^{new}\right)$$

Kernel Logistic Regression

$$\beta_i \leftarrow \beta_i + \alpha \left(y^{(i)} - \sigma\left(\sum_{j=1}^{N} \beta_j \phi\left(x^{(j)}\right)^{\top} \phi\left(x^{(i)}\right) \right) \right)$$

$$\sigma\left(\theta^{\top} \phi(x^{new})\right) = \sigma\left(\sum_{j=1}^{N} \beta_j \phi\left(x^{(j)}\right)^{\top} \phi(x^{new}) \right)$$

$$= \sigma\left(\sum_{j=1}^{N} \beta_j K\left(x^{(j)}, x^{new}\right) \right)$$

Valid Kernels

Kernel Examples

- Linear kernel: $K(x,z) = x^{\top}z$
- Polynomial kernel: $K(x,z) = (1 + x^{\top}z)^{d}$
- RBF kernel (a.k.a Gaussian kernel): $K(x,z) = \exp\left(\frac{-||x-z||^{2}}{\sigma^{2}}\right)$
- Exponential kernel: $K(x,z) = \exp\left(\frac{-||x-z||_{2}}{\sigma^{2}}\right)$
- Laplacian kernel: $K(x,z) = \exp\left(\frac{-|x-z|}{\sigma}\right)$

RBF Kernel

- RBF kernel (a.k.a Gaussian kernel)

$$K(x,z) = \exp(-\gamma(x-z)^2) \qquad \text{(1d case)} \qquad\qquad \exp(x) = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \ \ldots$$

$$K(x,z) = \exp(-\gamma x^2 - \gamma z^2)\exp(2\gamma xz)$$

$$= \exp(-\gamma x^2 - \gamma z^2)\left(1 + \frac{2\gamma xz}{1!} + \frac{2\gamma xz^2}{2!} + \frac{2\gamma xz^3}{3!} + \ \ldots\right)$$

$$= \exp(-\gamma x^2 - \gamma z^2)\left(1 + \frac{\sqrt{2\gamma}}{1}x\frac{\sqrt{2\gamma}}{1}z + \frac{\sqrt{(2\gamma)^2}}{\sqrt{2!}}x^2\frac{\sqrt{(2\gamma)^2}}{\sqrt{2!}}z^2 + \ \ldots\right) = \phi(x)^\top\phi(z)$$

Properties of Kernels

- What kinds of functions $K(\cdot,\cdot)$ can correspond to some feature map ϕ?
- In other words, can we tell if there is some feature mapping ϕ so that $K(x,z) = \phi(x)^\top \phi(z)$?

Mercer's Theorem

$K: \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}$*, then for K to be valid kernel, it is necessary and sufficient,*

$\{x^{(1)}, \dots, x^{(N)}\}$*, the kernel matrix is symmetric positive semi-definite*

1. Kernel matrix -> symmetric positive semi-definite (necessary condition)

2. Symmetric positive semi-definte -> kernel matrix (sufficient condition)

Well-defined Kernels

1. $K(x, z) = x^\top z$
2. $c\,K(x, z)$
3. $K_1(x, z) + K_2(x, z)$
4. $g(K(x, z))$, where g is a polynomial function w/ positive coefficient
5. $K_1(x, z)K_2(x, z)$
6. $f(x)K(x, z)f(z)$
7. $\exp(K(x, z))$
8. $\exp\left(\frac{-||x-z||^2}{\sigma^2}\right)$

Kernel SVM

Maximum Margin Classifier (Recap)

- The new objective

$$\min_{w,b} w^\top w \qquad \forall i, \in D, \qquad y^{(i)}\left(w^\top x^{(i)} + b\right) \geq 1$$

Quadratic objective

Linear constraints

1. Convex
2. We can use Quadratic Programing
   - very well-established methods and software

Maximum Margin Classifier

- Lagrangian

$$\min_{w,b} \frac{1}{2} w^\top w \qquad \forall i, \in D, \qquad y^{(i)}\left(w^\top x^{(i)} + b\right) \geq 1$$

$$L(w, b, \alpha) = \frac{1}{2} w^\top w - \sum_{i=1}^{N} \alpha_i \left(y^{(i)}\left(w^\top x^{(i)} + b\right) - 1\right)$$

Duality

- Duality

$$d^* = \max_{\alpha \geq 0} \min_{w,b} L(w, b, \alpha) \leq \min_{w,b} \max_{\alpha \geq 0} L(w, b, \alpha) = p^*$$

- In SVM

$$d^* = \max_{\alpha \geq 0} \min_{w,b} L(w, b, \alpha) = \min_{w,b} \max_{\alpha \geq 0} L(w, b, \alpha) = p^*$$

Dual Optimization

$$\max_{\alpha \geq 0} \min_{w,b} L(w, b, \alpha) = \frac{1}{2} w^\top w - \sum_{i=1}^{N} \alpha_i \left(y^{(i)} \left(w^\top x^{(i)} + b \right) - 1 \right)$$

$$\min_{w,b} L(w, b, \alpha)$$

Dual Optimization

$$\max_{\alpha \geq 0} \min_{w,b} L(w,b,\alpha) = \frac{1}{2} w^\top w - \sum_{i=1}^{N} \alpha_i \left(y^{(i)} \left(w^\top x^{(i)} + b \right) - 1 \right)$$

$$\min_{w,b} L(w,b,\alpha)$$

$$\nabla_w L(w,b,\alpha) = w - \sum_{i=1}^{N} \alpha_i y^{(i)} x^{(i)} = 0 \qquad \nabla_b L(w,b,\alpha) = -\sum_{i=1}^{N} \alpha_i y^{(i)} = 0$$

$$\min_{w,b} L(w,b,\alpha) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} y^{(i)} y^{(j)} \alpha^{(i)} \alpha^{(j)} {x^{(i)}}^\top x^{(j)}$$

Dual Optimization

$$\max_{\alpha \geq 0} \min_{w,b} L(w,b,\alpha) = \frac{1}{2} w^\top w - \sum_{i=1}^{N} \alpha_i \left(y^{(i)} \left(w^\top x^{(i)} + b \right) - 1 \right)$$

$$\max_{\alpha} \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} y^{(i)} y^{(j)} \alpha^{(i)} \alpha^{(j)} x^{(i)\top} x^{(j)} \qquad s.t \sum_{i=1}^{N} \alpha^{(i)} y^{(i)} = 0 \,, \alpha_i \geq 0$$

1. Convex
2. Also Quadratic Programing

Testing with a new example

$$w^{\top} x^{new} + b = \left(\sum_{i=1}^{N} \alpha_i y^{(i)} x^{(i)} \right)^{\top} x^{new} + b = \sum_{i=1}^{N} \alpha_i y^{(i)} {x^{(i)}}^{\top} x^{new} + b$$

Support vectors $\alpha_i > 0$, otherwise $\alpha_i = 0$

Soft-margin SVM

- Primal

$$\min_{w,b} w^\top w + C \sum_{i=1}^{N} \xi^{(i)} \qquad \forall i, \in D, \quad y^{(i)}\left(w^\top x^{(i)} + b\right) \geq 1 - \xi^{(i)}$$

$$\forall i, \in D, \quad \xi^{(i)} \geq 0$$

- Dual

$$\max_{\alpha} \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} y^{(i)} y^{(j)} \alpha^{(i)} \alpha^{(j)} x^{(i)\top} x^{(j)} \qquad s.t. \sum_{i=1}^{N} \alpha^{(i)} y^{(i)} = 0$$

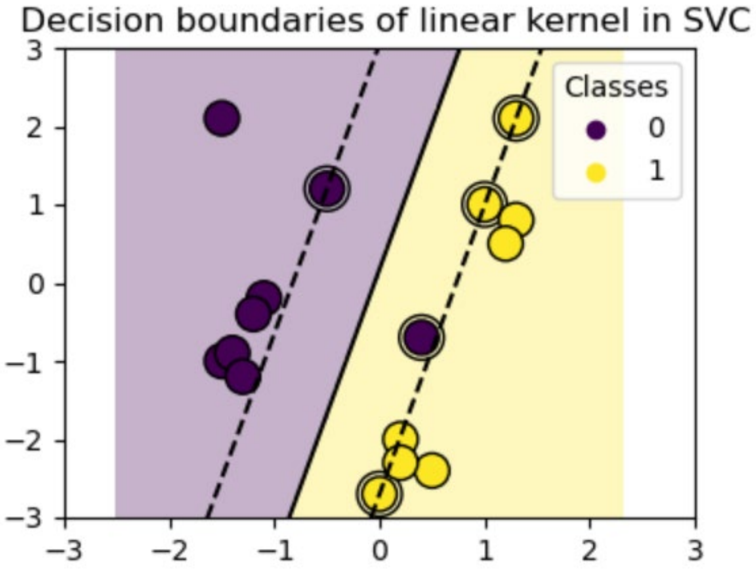
$$0 \leq \alpha_i \leq C$$

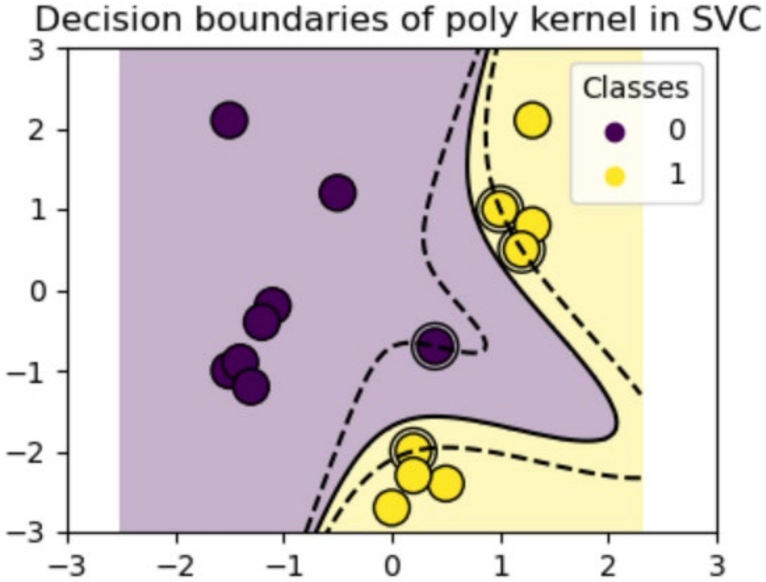
Different Kernels

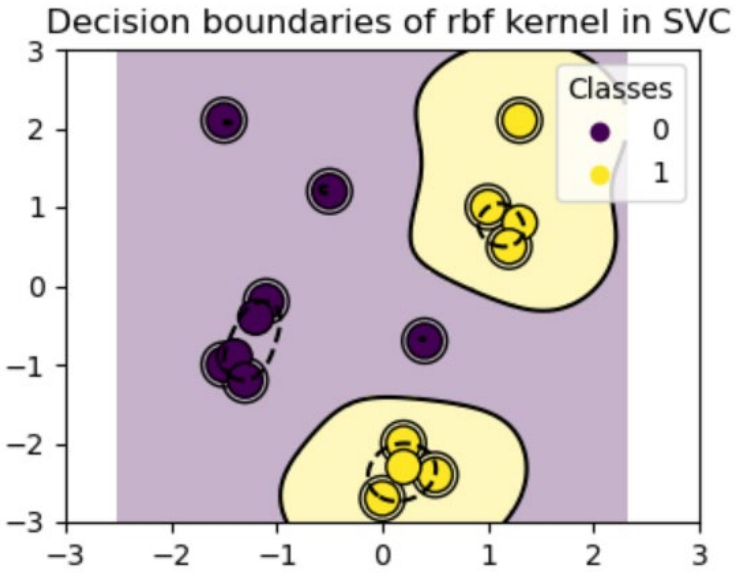
$K(x, z) = x^\top z$

$K(x, z) = (\gamma x^\top z + r)^d$

$K(x, z) = \exp(-\gamma ||x - z||^2)$







[Plot classification boundaries with different SVM Kernels — scikit–learn 1.4.2 documentation]

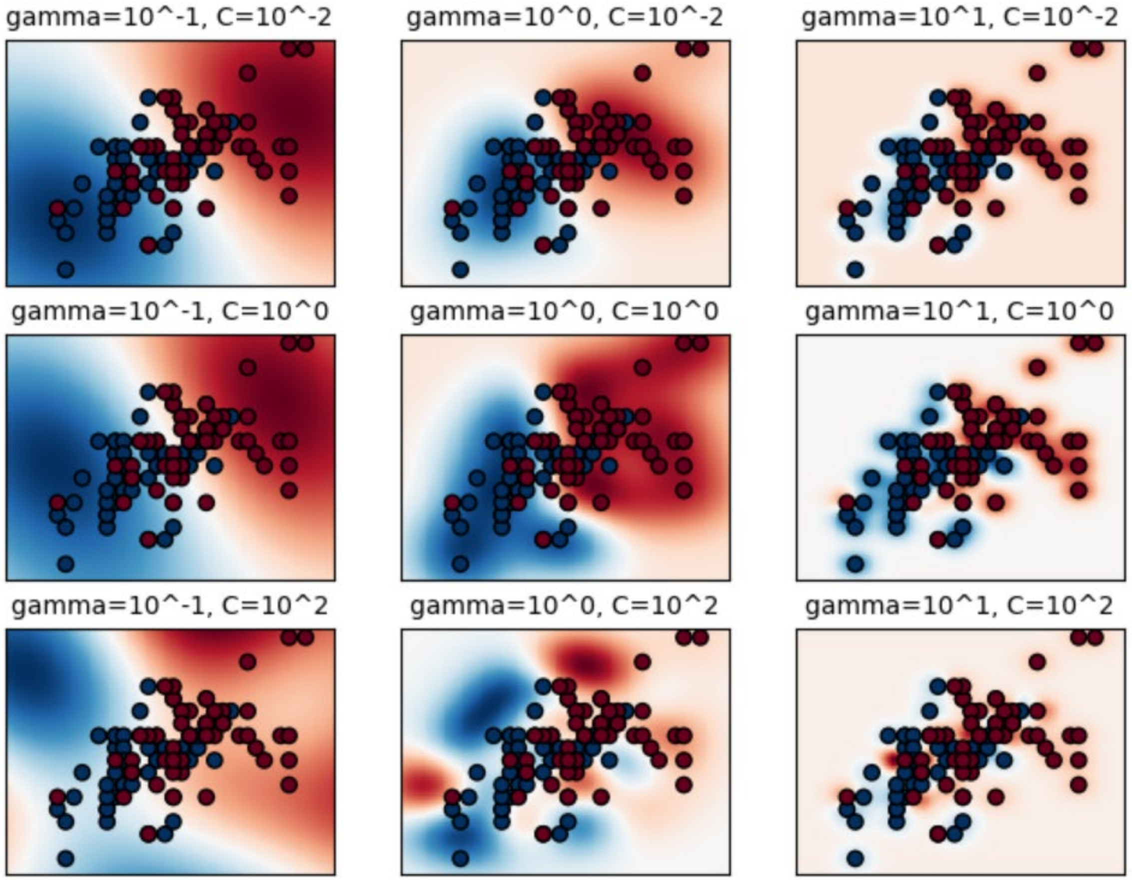
RBF Kernels

$$\min_{w,b} w^\top w + C \sum_{i=1}^{N} \xi^{(i)}$$

$$\forall i, \in D, \qquad y^{(i)}\left(w^\top x^{(i)} + b\right) \geq 1 - \xi^{(i)}$$

$$\forall i, \in D, \qquad \xi^{(i)} \geq 0$$

$$K(x, z) = \exp(-\gamma ||x - z||^2)$$



[RBF SVM parameters — scikit–learn 1.4.2 documentation]