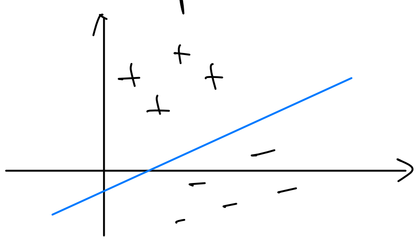
KERNEL METHOD

Main idea: swap Weighted Features for Similarity function

- So far we know to separate data points in a linear way

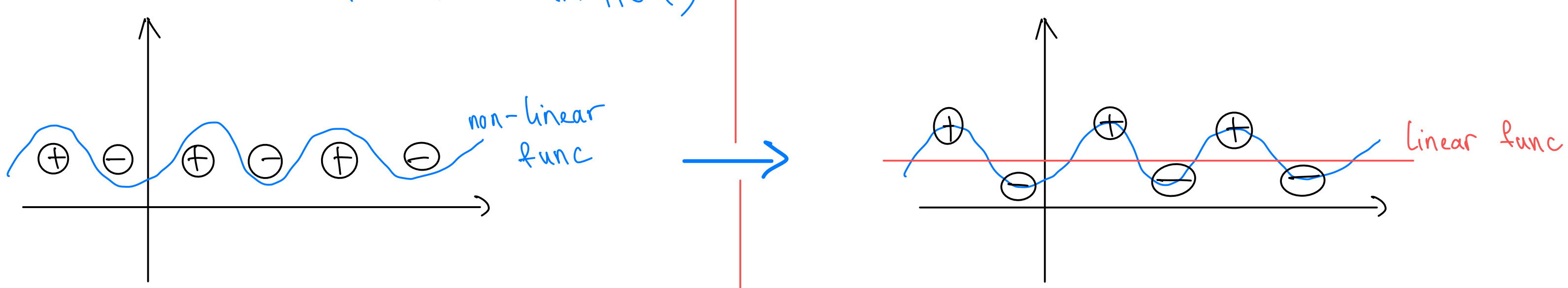


- In pratice, data is not always linearly separable, that is where kernel method comes in:
- Kernel method transforms the data points into higher dimension, where it is easier to separate than the original space. Then use linear function to separate the transformed data points.

Kernel Method

i) Find (automatically) non-linear function (basis function)

ii) Use linear function to separate the data



How Kernel method fits in the Supervised Learning Framework?

- Recall Supervised Learning Framework:

i) Given dataset $D = \{x_i, y_i\}_{i=1}^{n}$

ii) Find $f(x_i)$ such that $f(x_i) \approx y_i$

iii) To find that function $f(x_i)$, minimize empirical risk:

$$\hat{f} = \underset{f}{\operatorname{argmin}}\ \mathcal{L}(f, D)$$

where $\begin{cases} f \in F, & \text{class of functions} \\ \mathcal{L}(f, D): & \text{empirical risk (loss function)} \end{cases}$

$\hookrightarrow$ f can be

- linear: $f_\theta(x) = \sum_{l=1}^{d} \theta_l x_l$

- non-linear: $f_\theta(x) = \sum_{l=1}^{m} \theta_l \phi_l(x)$

$x = \begin{bmatrix} x_1 \\ \vdots \\ x_d \end{bmatrix}$

# of basis functions

where ϕ_l is a basis function. Since different problems require different basis function, we need method to automate the process of finding the right basis function

Kernel method : Definition

A method is kernel if it is symmetric. Given 2 input x and x':

$$K(x, x') = K(x', x)$$

Example: Gaussian Radical Basis Function (RBF)

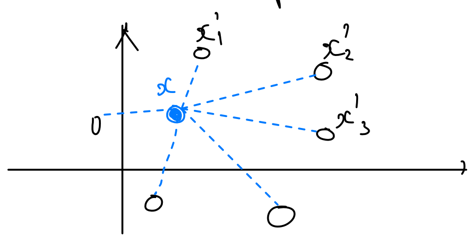
$$K(x, x') = \exp\left(-\frac{1}{2h^2}\|x - x'\|^2\right)$$

is kernel because:

$$K(x', x) = \exp\left(-\frac{1}{2h^2}\|x' - x\|^2\right)$$

Kernel method = Similarity function

- Similarity function measure how similar 2 data points are. Visually, consider data points in 2 dimensional space:



Similarity function takes 2 data points and output value $\in \mathbb{R}$:

$$K(x, x') : X \times X \mapsto \mathbb{R}$$

which is what kernel method does

Kernel Method in Linear Regression

- Recall in linear regression, we try to find the best function f that represent the relationship between data points:

i) $\hat{\theta} = \underset{\theta}{\operatorname{argmin}} \| y - X\theta \|_2^2$

best weight vector $d \times 1$

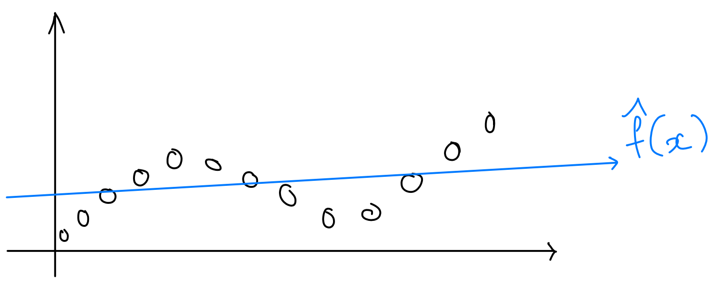
label vector

weight vector $[\ ]d$

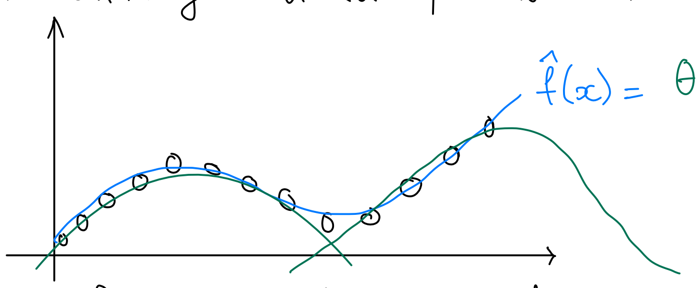
data set $\begin{array}{c} d \\ [\quad]N \end{array}$

ii) $\hat{f}(x) = x^T\hat{\theta}$

linear



- In practice, the data is not always linear, for example:



$\hat{f}(x) = \theta_1 k(x, x_1) + \theta_2 k(x, x_2) + \dots \theta_N k(x, x_N)$

This is the perfect use case for kernel method.

- **Linear regression with kernel method**: Find best non-linear function f to represent the relationship between data points

i) $$\hat{\theta} = \underset{\theta}{\operatorname{argmin}} \| y - K \theta \|_2^2$$

best dual coefficient vector $(N \times 1)$

vector of dual coefficient $(N \times 1)$

Kernel Matrix: $(N \times N)$

$$K = \begin{bmatrix} k(x_1, x_1) & \cdots & k(x_1, x_N) \\ \vdots & k(x_2, x_2) & \vdots \\ k(x_N, x_1) & \cdots & k(x_N, x_N) \end{bmatrix}$$

Explicitly: $$\hat{\theta} = \underset{\theta}{\operatorname{argmin}} \sum_{i=1}^{N} \left(y_i - \sum_{j=1}^{N} \theta_j k(x_i, x_j) \right)$$

$= K\theta$ (Similarity Function)

Closed Method Solution (only if K is invertible):

$$\hat{\theta} = K^{-1} y$$

ii) $$\hat{f}(x) = \begin{bmatrix} k(x, x_1) \\ \vdots \\ k(x, x_N) \end{bmatrix}^T \hat{\theta}$$

non-linear

$$= \begin{bmatrix} k(x, x_1) \\ \vdots \\ k(x, x_N) \end{bmatrix}^T \begin{bmatrix} \hat{\theta}_1 \\ \vdots \\ \hat{\theta}_N \end{bmatrix}$$

- Avoid overfitting, use **regularization**, so the formula becomes:

$$\hat{\theta} = \underset{\theta}{\operatorname{argmin}} \| y - K\theta \|_2^2 + \alpha \phi(\theta)$$

error parameter

Regularization (idea: pay a price for non-zero elements in θ)

$\Rightarrow$ penalize for having large coefficients

$\phi(\theta)$ can be
- L_2 norm: $\| \theta \|_2^2 = \theta^T \theta$
- K norm: $\| \theta \|_K^2 = \theta^T K \theta$

K norm is better than L_2 norm

shrinkage parameter

Explicitly: $$\boxed{\hat{\theta} = \underset{\theta}{\operatorname{argmin}} \| y - K\theta \|_2^2 + \alpha \theta^T K \theta}$$

"range" of θ can have

$\Rightarrow$ the smaller α, the larger the range $\Rightarrow$ overfit

Closed form solution (if K is invertible):

$$\boxed{\hat{\theta} = (K + \alpha I)^{-1} y}$$

Kernel Method in Classification (Logistic Regression)

- Recall in Classification problem, we try to find function f such that it minimize the number of misclassifications the loss function:

  i) Calculate predicted $\hat{y}$:

  $$\hat{y}_i = \frac{1}{1 + \exp(-\theta^T x_i)}$$

  ii) Minimize loss function $L(\theta)$:

  weight vector $(d \times 1)$

  weighted features

  $$\begin{aligned}\hat{\theta} &= \underset{\theta}{\text{argmin}}\ L(\theta) \\ &= \underset{\theta}{\text{argmin}} \sum_{i=1}^{N} \ell(y_i, \hat{y}_i) \\ &= \begin{cases} \underset{\theta}{\text{argmin}} \sum_{i=1}^{N} \log(1 + \exp(-y_i \theta^T x_i)) & \text{if } y \in \{\pm 1\} \\ \underset{\theta}{\text{argmin}} \sum_{i=1}^{N} -\left[y_i \log(\hat{y}_i) + (1-y_i)\log(1-\hat{y}_i) \right] & \text{if } y \in \{0,1\} \end{cases}\end{aligned}$$

- Logistic Regression with kernel method: consider the case $y = \{\pm 1\}$, the loss function becomes:

  dual coefficient vector $(N \times 1)$

  similarity

  $$\begin{aligned}\hat{\theta} &= \underset{\theta}{\text{argmin}} \sum_{i=1}^{N} \ell(y_i, K\theta) \\ &= \underset{\theta}{\text{argmin}} \sum_{i=1}^{N} \ell\left(y_i, \sum_{j=1}^{N} \theta_j k(x_i, x_j)\right) \\ &= \underset{\theta}{\text{argmin}} \sum_{i=1}^{N} \log\left(1 + \exp\left(-y_i \sum_{j=1}^{N} \theta_j k(x_i, x_j)\right)\right)\end{aligned}$$

  Add regularization term to avoid overfitting:

  $$\boxed{\hat{\theta} = \underset{\theta}{\text{argmin}} \sum_{i=1}^{N} \log\left(1 + \exp\left(-y_i \sum_{j=1}^{N} \theta_j k(x_i, x_j)\right)\right) + \alpha \theta^T K \theta}$$