Kernels

Supervised learning framework:

- Given training data $D = \{x^{(i)}, y^{(i)}\}_{i=1}^{n}$, we want to find $f(x)$ s.t. $y^{(i)} = f(x^{(i)})$.
- Use Empirical Risk Minimization:
  - Decide a function class F.
  - Define an empirical loss function $L(f, D)$ for $f \in F$.
  - Solve the optimization problem: $\hat{f} = \underset{f \in F}{\operatorname{argmin}}\ L(f, D)$.

For example, for linear regression:

The function class is:

$$F \triangleq \left\{ f_\theta(x) = \sum_{l=1}^{d} \theta_l x_l + \theta_0 \;\middle|\; \forall\, \theta_l \in \mathbb{R} \right\}, \quad x = \underbrace{\begin{bmatrix} x_1 \\ \vdots \\ x_d \end{bmatrix}}_{\text{each } x^{(i)} \text{ has } d \text{ features.}}$$

The objective function is:

$$\text{Min } L(f_\theta, D) = \frac{1}{n} \sum_{i=1}^{n} \left(y^{(i)} - f_\theta(x^{(i)}) \right)^2$$

Intro Kernels

A way to incorporate non-linearities into most linear classifiers, by applying basis function (feature transformations) on the input feature vectors.

$$x \longrightarrow \underbrace{\phi(x)}_{\in \mathbb{R}^D}$$
$$\uparrow$$
$$\in \mathbb{R}^d$$

, $D >> d$, bc we add dimensions that capture non-linear interactions among original features.

Advantage: the problem stays simple, convex, well-behaved.

Disadvantage: $\phi(x)$ might be very high dimensional.

consider the following example: $x = \begin{pmatrix} x^{(1)} \\ x^{(2)} \\ \vdots \\ x^{(d)} \end{pmatrix} \longrightarrow \phi(x) = \begin{pmatrix} 1 \\ x^{(1)} \\ \vdots \\ x^{(d)} \\ \vdots \\ x^{(d-1)}\, x^{(d)} \\ \vdots \\ x^{(1)}\, x^{(2)} \ldots x^{(d)} \end{pmatrix}$

Linear Regression using Kernels:

Linear Regression: $f(x, \theta) = \sum_{l=1}^{d} \theta_l x_l = \theta^T X$, $X = \begin{bmatrix} x^{(1)} \\ \vdots \\ x^{(d)} \end{bmatrix}$ ← # of data points.

Objective: $\min_{\theta} L(f_\theta, D) = \underbrace{\frac{1}{d} \sum_{i=1}^{d} \left(y^{(i)} - f_\theta(x^{(i)})\right)^2}_{\text{mean square loss}}$

However, this set-up cannot capture non-linear relations.

Solution:

Transform linear function into non-linear function:

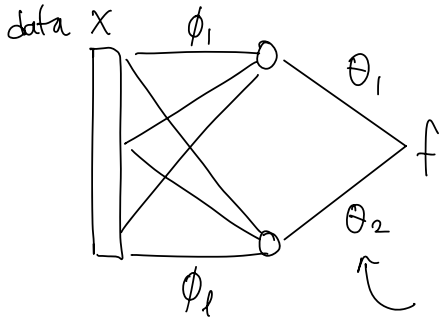
Linear: $f(x, \theta) = \sum_{l=1}^{d} \theta_l x_l = \theta^T X$

Nonlinear: $f(x, \theta) = \sum_{l=1}^{M} \theta_l \phi_l(x) = \theta^T \phi(x)$

$x_l \rightarrow \phi_l$

coefficient vector $\theta = [\theta_1, \cdots, \theta_M]^T$

set of basis functions $[\phi_1(x), \cdots, \phi_M(x)]^T$



perform linear regression on the transformed spaces.

here, # of ϕ >> # of data points.

∴ You need to manually decide what basis function to use, which is not ideal. You want this to be automatic and adaptive.

⇒ use adaptive basis function, of which kernel method is one.

How to construct the basis functions?

Adaptive basis functions:

- Kernels
- Neural networks

Kernel Method:

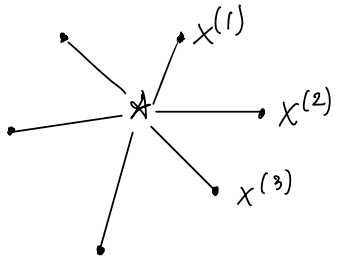
Let $k(x, x')$ be the similarity measure btwn x and x'.

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

Let $k(x, x') = k(x', x)$. // two variable symmetric function.

This similarity measure is called a kernel function.

Geometric interpretation: Given $D = \{x^{(i)}, y^{(i)}\}_{i=1}^{n}$



$k(x, x')$

Example Kernel Functions

A typical kernel function is the Gaussian radial basis function (RBF) kernel:

$$k(x, x') = \exp\left[\frac{-1}{2h^2}\|x - x'\|_2^2\right],$$

where

h = bandwidth $\in \mathbb{R}^+$. It is a scaling parameter, that should be in the range of the data points.

Ex: variance.

Another Example: $k(x, x') = \exp\left[\frac{-1}{h}\|x - x'\|\right)$ Laplace kernel.

Given a dataset $D = \{x^{(i)}\}_{i=1}^{n}$, we may construct a kernel representation of a point x by comparing it with each observed data point in the dataset:

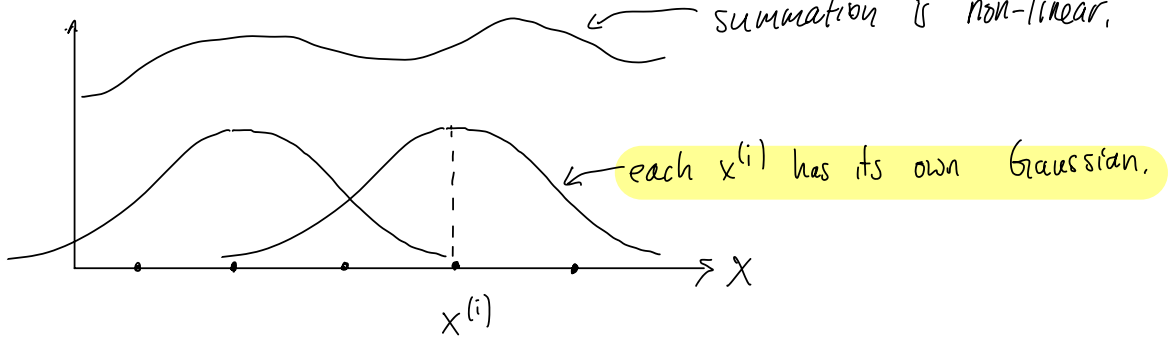
$$\theta = \begin{bmatrix} \theta_1 \\ \vdots \\ \theta_n \end{bmatrix}, \quad \phi(x) = \begin{bmatrix} k(x, x^{(1)}) \\ \vdots \\ k(x, x^{(n)}) \end{bmatrix}$$

where $\{x^{(i)}\}$s are the fixed observed data points, x is the variable of the function: $\phi_i = k(x, x^{(i)})$.

$$f_\theta(x) = \theta^T \phi(x) = \sum_{i=1}^{n} \theta_i \phi_i(x) = \sum_{i=1}^{n} \theta_i k(x, x_i)$$

Example using RBF:

suppose you have a set of 1D data:



we want:

F to be adaptive with data D

$\dim(F) = n \longrightarrow +\infty$.

The class of function that has dimension that grows with the # of data points is called non-parametric methods. Kernel method is a special case of non-parametric method.

Loss function: (n ← # data points)

$$L(\theta) = \sum_{j=1}^{n} \left[y^{(j)} - \sum_{i=1}^{n} \theta_i \underbrace{k(x^{(j)}, x^{(i)})}_{\phi_i(x)} \right]^2$$

$$= \| Y - K\theta \|_2^2 \quad , \text{ where}$$

$$Y = \begin{bmatrix} y^{(1)} \\ \vdots \\ y^{(n)} \end{bmatrix}_{n \times 1} \qquad K = \begin{bmatrix} k(x_1, x_1) & \cdots & k(x_1, x_n) \\ \vdots & & \vdots \\ k(x_n, x_1) & \cdots & k(x_n, x_n) \end{bmatrix}$$ ← gram matrix, $n \times n$.

or $K = [k(x^{(i)}, x^{(j)})]_{i,j=1}^{n}$

Objective $= \min L(\theta)$

$$= \min_{\theta} \| Y - K\theta \|_2^2$$

$\nabla L(\theta) = 2K^T(K\theta - Y)$ // shown without derivation

$\Rightarrow K^T K\theta = K^T Y.$

$\Rightarrow \boxed{\theta = K^{-1} Y}$ Need to assume K is invertible.

This regression can perfectly fit all the data points, However, it would be at risk for overfitting.

Regularization: use regularization to avoid overfitting.

Solve a regularized version:

$$\min_{\theta} \|y - K\theta\|_2^2 + \alpha \Phi(\theta)$$

α: hyperparameter that we choose. $\Phi(\theta)$: regularization. Often L2 Norm.

Instead L2 regularization $\left(\Phi(\theta) = \|\theta\|_2^2\right)$, use k-norm:

$$\Phi(\theta) = \|\theta\|_K^2 \equiv \theta^T K \theta$$

K: Gram/kernel Matrix, size $n \times n$.

Why $\|\theta\|_K^2$ vs $\|\theta\|_2^2$? Consider an Example:

$D = \{x^{(1)}, x^{(2)}, x^{(3)}\}$ $\quad x^{(1)} = x^{(2)} \neq x^{(3)}$.

assume equal: $x^{(1)} = x^{(2)}$. • $x^{(1)} = x^{(2)}$

• $x^{(3)}$

The kernel representation is:

$$f(x) = \theta_1 k(x, x^{(1)}) + \theta_2 k(x, x^{(2)}) + \theta_3 k(x, x^{(3)})$$
$$= (\theta_1 + \theta_2) k(x, x^{(1)}) + \theta_3 (x, x^{(3)})$$

$\Rightarrow \|\theta\|_2^2 = \theta_1^2 + \theta_2^2 + \theta_3^2$, which regularizes each term, does not make sense. More reasonably to use:

$(\underbrace{\theta_1 + \theta_2}_{\text{combined}})^2 + \theta_3^2 = \|\theta\|_K^2$ as the penalty.

Kernel norm does what we want (grouping effect):

Consider the Gaussian kernel:

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

which has $k(x, x) = 1 \quad \forall x$.

If $x^{(1)}$ and $x^{(3)}$ are very different, then $k(x^{(1)}, x^{(3)}) \approx 0$

Therefore the Gram matrix is:

$$K \approx \begin{array}{c} \\ x^{(1)} \\ x^{(2)} \\ x^{(3)} \end{array} \begin{array}{c} \begin{array}{ccc} x^{(1)} & x^{(2)} & x^{(3)} \end{array} \\ \begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \end{array},$$

which suggests $\Phi(\theta) =$

$$\theta^T K \theta = \begin{bmatrix} \theta_1 & \theta_2 & \theta_3 \end{bmatrix} \begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \end{bmatrix} = \left[\theta_1 + \theta_2\right]^2 + \theta_3^2$$

Regularized Objective Function:

$$\min_{\theta} \|Y - K\theta\|_2^2 + \alpha \theta^T K \theta.$$

$\Rightarrow$ Solve in closed form: $\hat{\theta} = (K + \lambda I)^{-1} Y$

(I: Identity)

Kernel Set up:

Classification: $\{x^{(i)}, y^{(i)}\}$, $y^{(i)} \in \{-1, +1\}$ // binary Classification

$$L(\theta) = \sum_{i=1}^{n} \sigma\left(y^{(i)} f_\theta(x^{(i)})\right)$$

σ: surrogate function, can be logistic loss, hinge loss, etc.

$$= \sum_{i=1}^{n} \sigma\left(y^{(i)} \sum_{j=1}^{n} \theta_j K(x^{(j)}, x^{(i)})\right) + \lambda \theta^T K \theta$$

λ: α?

Solve numerically.

Kernel as infinite dimensional features:

Using Kernel $\Leftrightarrow$ using infinite basis functions.
Kernel classification $\Leftrightarrow$ solving an infinite dimensional optimization problem.

Define: $\{\psi_\ell(x) : \ell = 0, 1, 2, \ldots, \infty\}$

Example function: $\psi_\ell(x) = x^\ell$

Set up: $f(x) = \sum_{\ell=0}^{\infty} w_\ell \psi_\ell(x)$

ψ_ℓ: ℓ^{th} basis function of infinite sequence of basis functions.

w_ℓ: weight of the ℓ^{th} basis function.

The idea is to have infinite number of (polynomial or other) functions, and a linear combination of them can be used to approximate any function. Sort of like Taylor series, or Fourier series.

Define Loss function:

$$L(w) = \hat{E}_D\left[\left(y - \sum_{l=0}^{\infty} w_l \psi_l(x)\right)^2\right] + \sum_{l=0}^{\infty} \frac{w_l^2}{\lambda_l}$$

- $L(w)$: infinite sequence of w_l
- $\hat{E}_D$: expectation over the dataset.
- λ_l = regularization coefficient for w_l. Controls how much to penalize w_l.

Define $\lambda_l := \frac{1}{\alpha_l} > 0$. ?

$\lambda_1 \geq \lambda_2 \geq \cdots \geq 0$

Note: If λ_l is small, then penalty is huge; $w_l \to 0$.
$\Rightarrow$ Higher order terms are penalized exponentially; they contribute much more to the Loss function.

Result:
Solving this infinite dimensional optimization problem is equivalent to solving a finite dimensional kernel representation. This is due to the regularization term turning off the higher order terms.
In practice, the optimization problem is not infinite, because we don't have infinite data.

How is kernel related to infinite dimensional features?

Define $\hat{w} = \underset{w}{\operatorname{argmin}}\, L(w)$

$\hat{w}$: optimal solution.

$$= \underset{w}{\operatorname{argmin}} \sum_{i=1}^{n} \left(y^{(i)} - \sum_{l=0}^{\infty} w_l \psi_l(x^{(i)})\right)^2 + \sum_{l=1}^{\infty} \frac{w_l^2}{\lambda_l}$$

vector of weights; vector of infinite basis functions

$$= \underset{w}{\operatorname{argmin}} \hat{E}_D\Big[\big(y - \underbrace{\sum_{l=0}^{\infty} w_l \psi_l(x)}_{\text{define as } f(x)}\big)^2\Big] + \underbrace{\sum_{l=0}^{\infty} \frac{w_l^2}{\lambda_l}}_{\text{regularization term; prevents norm of } \{w_l\} \text{ from being too large.}}$$

Define $\hat{f}(x) = \sum_{l=0}^{\infty} \hat{w}_l \psi_l(x)$ (infinite sequence representation of the data)

($\hat{f}$: corresponding optimal function for $\hat{w}$)

Then: $\hat{f}(x) = \sum_{i=1}^{n} \theta_i \underbrace{k(x, x^{(i)})}_{\text{replaced } \psi}$ for some θ_i (finite kernel representation of the data)

(n: replaced w)

and $\boxed{k(x, x') \equiv \sum_{l=0}^{\infty} \lambda_l \psi_l(x) \psi_l(x')}$ (definition of kernel function)

and $\boxed{\sum_{l=0}^{\infty} \frac{w_l^2}{\lambda_l} \equiv \theta^T K \theta}$

~~Note: $\theta_i = y^{(i)} - \hat{f}(x^{(i)})$?~~

The infinite representation then reduces to:

$$\Leftrightarrow \hat{\theta} = \underset{\theta}{\operatorname{argmin}} \hat{E}_D\Big[\big(y - \sum_{i=1}^{n} \theta_i k(x, x^{(i)})\big)^2\Big] + \theta^T K \theta$$

(no longer infinite; regularization term is = kernel loss.)

~~$\Rightarrow \hat{\theta} = \underset{\theta}{\operatorname{argmin}} \hat{E}_D\Big[\big(y - \sum_{i=1}^{n} \theta_i$~~

proof: Find the zero gradient condition, which implies this relation at optimal point.

Objective is to solve this optimization problem:

$$\min_{w} \hat{E}_D\left[\underbrace{\left(y - \sum_{l=0}^{\infty} w_l \psi_l(x)\right)^2}\right] + \sum_{l=0}^{\infty} \frac{w_l^2}{\lambda_l}$$

(infinite sequence representation.)

Define $\delta(x) = \underbrace{y - \sum_{l=0}^{\infty} w_l \psi_l(x)}_{f(x)}$

take gradient $\longrightarrow$ $\nabla_{w_l} L(\hat{w}) = 2\hat{E}_D\left[-\delta(x)\psi_l(x)\right] + \frac{2\hat{w}_l}{\lambda_l} \overset{\text{set} = 0, \text{ since at optimal point, gradient} = 0}{=} 0$

$$\Rightarrow \frac{2w_l}{\lambda_l} = 2\hat{E}_D\left[\delta(x)\psi_l(x)\right] \Rightarrow \hat{w}_l = \lambda_l \hat{E}_D\left[\delta(x)\psi_l(x)\right]$$

$$\Rightarrow \boxed{\hat{w}_l = \frac{\lambda_l}{n}\sum_{i=1}^{n}\left[\delta(x^{(i)})\psi(x^{(i)})\right]}$$

Then substitute $\hat{w}$ into $f(x)$:

$$\hat{f}(x) = \sum_{l=0}^{\infty} \hat{w}_l \psi_l(x) = \sum_{l=0}^{\infty}\underbrace{\left[\frac{\lambda_l}{n}\sum_{i=1}^{n}\delta(x^{(i)})\psi_l(x^{(i)})\right]}_{\hat{w}_l}\psi_l(x)$$

(summation over all data points; summation over all basis functions)

$$= \sum_{l=0}^{\infty}\sum_{i=1}^{n}\frac{\lambda_l}{n}\delta(x^{(i)})\psi_l(x^{(i)})\psi_l(x)$$

// take summation out.

$$= \sum_{i=1}^{n} \sum_{\ell=0}^{\infty} \frac{\lambda_\ell}{n} \delta(x^{(i)}) \psi_\ell(x^{(i)}) \psi_\ell(x)$$ // switched order of summation.

$$= \sum_{i=1}^{n} \frac{\delta(x^{(i)})}{n} \underbrace{\sum_{\ell=0}^{\infty} \lambda_\ell \psi_\ell(x^{(i)}) \psi_\ell(x)}_{\text{kernel function } k(x', x)}$$ // move out of summation, bc $\delta(x^{(i)})$ is independent of ℓ.

$$= \boxed{\sum_{i=1}^{n} \theta_i \, k(x^{(i)}, x) \quad \text{where} \quad \theta_i \triangleq \frac{\delta(x^{(i)})}{n}}$$

= linear combo of kernel functions.

This proves the first part of the result.

The second part is in textbook.

Two views that are equivalent:

1) $\hat{f}(x) = \sum_{i=1}^{n} \hat{\theta}_i \, k(x^{(i)}, x)$ // finite linear combo of kernel functions evaluated at data point.

2) $\hat{f}(x) = \sum_{\ell=0}^{\infty} \hat{\omega}_\ell \psi_\ell(x)$ // infinite basis functions. Theoretical purpose.

connected by: $\underbrace{k(x, x')}_{\text{any kernel function}} = \underbrace{\sum_{\ell=0}^{\infty} \lambda_\ell \psi_\ell(x) \psi_\ell(x')}_{\text{infinite basis functions}}$ $\lambda_\ell \geq 0$.

↑ inverse regularization coefficient.

∴ Any kernel function can be represented using some basis functions.

Mercer Theorem: any continous positive definite kernel can be represented by $k(x, x') = \sum_{\ell=0}^{\infty} \lambda_\ell \psi_\ell(x) \psi_\ell(x')$ with $\lambda_\ell \geq 0$.

1) If $k(x, x')$ is positive definite $\Leftrightarrow$ $K = \left[k(x^{(i)}, x^{(j)})\right]_{j=1}^{n} \geq 0$ for any n and any data point $\{x^{(i)}\}$.

2) $\Leftrightarrow \exists\ \psi_\ell,\ \lambda_\ell \geq 0$.

The typical kernels are positive definite, so don't worry!

Example: Gaussian RBF kernel:

$$k(x, x') = \exp\left[-\gamma \|x - x'\|^2\right] \quad \text{// definition.}$$

$$= \exp\left[-\gamma \|x\|^2 + 2\gamma x^T x' - \gamma \|x'\|^2\right] \quad \text{// expanded form}$$

$$= \exp\underbrace{\left(-\gamma \|x\|^2\right)}_{\text{depends on } x} \underbrace{\exp\left(-\gamma \|x'\|^2\right)}_{\text{depends on } x'} \underbrace{\exp\left(2\gamma x^T x'\right)}_{\text{depends on both.}}$$

Taylor Expansion: $\exp(t) = 1 + t + \frac{t^2}{2} + \cdots + \frac{t^n}{n!} = \sum_{\ell=0}^{\infty} \frac{t^\ell}{\ell!}$

$\exp\left(2\gamma x^T x'\right) = \sum_{\ell=0}^{\infty} \frac{(2\gamma x^T x')^\ell}{\ell!}$, etc, for the other 2 terms.

$\Rightarrow k(x, x') = \sum_{\ell=0}^{\infty} \lambda_\ell \psi_\ell(x) \psi_\ell(x')$, Assume $x \in \mathbb{R}$.

where $\psi_\ell(x) = \exp\left(-\gamma x^2\right) x^\ell$,

$$\lambda_\ell = \frac{(2\gamma)^\ell}{\ell!}.$$

$\Rightarrow \lambda$ is larger than 0, therefore RBF is Positive definite.