

# Kernels: Linear models in function space

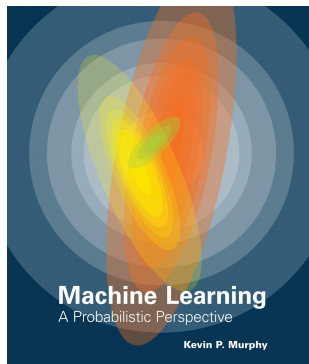
Rodrigo A. Vargas-Hernández

31/Jan/2023

# References for today's lecture

(31/Jan/2023)

## Chapter 13: Sparse linear models and Chapter 14: Kernels



- Kernel ridge regression
- extra material

# Regularization Overfitting

- One technique that is often used to control the over-fitting phenomenon in such cases is that of regularization,

$$\mathcal{L}(\boldsymbol{\theta}) = \frac{1}{2} \sum_i^N (y_i - \boldsymbol{\theta}^T \boldsymbol{\phi}(\mathbf{x}_i))^2 + \frac{\lambda}{2} \sum_j w_j^2 \quad (1)$$

- $\lambda$  is known as the **regularization** term.

## (HOMEWORK)

- What is the value of the **optimal** parameters  $\boldsymbol{\theta}^*$  (Eq. 12)?  
(tip):  $\sum_j w_j^2 = \mathbf{w}^T \mathbf{w}$  and  $\frac{\partial \mathbf{w}^T \mathbf{w}}{\partial \mathbf{w}} = 2\mathbf{w}$

# Notation

- ▶  $\mathbf{x} \rightarrow$  single data point,  $\mathbf{x} = [x_0, \dots, x_i, \dots, x_d]$
- ▶  $d \rightarrow$  total number of **features** in  $\mathbf{x}$
- ▶  $y \rightarrow$  observable of a single point <sup>1</sup>
- ▶  $\mathbf{X} \rightarrow$  all data points,  $\mathbf{X} = [\mathbf{x}_1^T, \dots, \mathbf{x}_N^T]$
- ▶  $\mathbf{y} \rightarrow$  observables for all data points,  $\mathbf{y} = [y_1, \dots, y_N]$
- ▶  $N \rightarrow$  total number of data points
- ▶  $\mathcal{D} \rightarrow$  Data set,  $\mathcal{D} = [\mathbf{X}, \mathbf{y}]$
- ▶  $\boldsymbol{\theta} \rightarrow$  all model's parameters
- ▶  $f(\cdot) \rightarrow$  model
- ▶  $\mathcal{L}(\cdot) \rightarrow$  loss, error or cost function

---

<sup>1</sup>Could be a vector, **multiple observables**

# Regularization Overfitting

- One technique that is often used to control the over-fitting phenomenon in such cases is that of regularization,

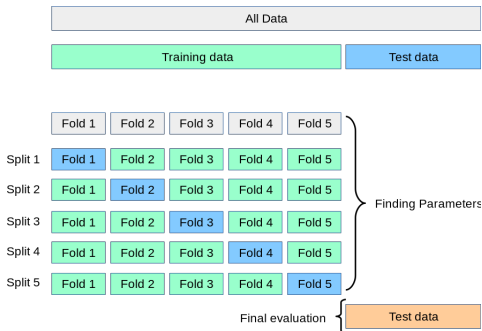
$$\mathcal{L}(\theta) = \frac{1}{2} \sum_i^N (y_i - \theta^T \phi(\mathbf{x}_i))^2 + \frac{\lambda}{2} \sum_j w_j^2 \quad (2)$$

$$\mathcal{L}(\theta) = \frac{1}{2} (\mathbf{y} - \theta^T \Phi(\mathbf{X}))^T (\mathbf{y} - \theta^T \Phi(\mathbf{X})) + \frac{\lambda}{2} \mathbf{w}^T \mathbf{w} \quad (3)$$

- ▶  $\lambda$  is known as the **regularization** term.
- ▶  $\lambda$  is optimized using [Cross-Validation](#).
- ▶  $\Phi(\mathbf{X})$  is the representation of all data points  $\mathbf{X}$  in the feature space  $\phi(\cdot)$ .

# (Quick primer) Cross-Validation

- Search algorithm to optimize **hyper-parameters** in ML models.



- ▶ For every fold we search for the best  $\lambda$ .
- ▶ Average all the best  $\lambda$ s,  $\lambda^* = \frac{1}{K} \sum_j^K \lambda_j$ .
- ▶ **What are the cons of CV?**

## (code) Cross-Validation

```
import numpy as np
from sklearn.model_selection import KFold

data = load_data() # data = (X,y)

n_folds = 5 # number of folds
kf = KFold(n_splits=n_folds)

# grid on the possible values of lambda
lambda_grid = np.array([0.,0.001,0.01,0.1,0.5,1.])

l_ = []
# iterate over the k-folds
for train, val in kf.split(X): # index
    X_train, y_train = X[train], y[train]
    X_val, y_val = X[val], y[val]

    # search algorithm for lambda
    lambda_opt = solve_for_lambda((X_train,y_train), (X_val,y_val),lambda_grid)
    l_.append(lambda_opt)

l_ = np.array(l_)
best_lambda = np.mean(l_)
```

## solution of linear regression + regularization

$$\mathcal{L}(\theta) = \frac{1}{2} (\mathbf{y} - \theta^T \Phi(\mathbf{X}))^T (\mathbf{y} - \theta^T \Phi(\mathbf{X})) + \frac{\lambda}{2} \theta^T \theta \quad (4)$$

- Solve for  $\theta$  using  $\nabla_{\theta} \mathcal{L}(\theta) = 0$ .

**Solution:** (last week's lecture notes)

$$\theta^* = (\Phi(\mathbf{X})^T \Phi(\mathbf{X}) + \lambda \mathbb{I}_d)^{-1} \Phi(\mathbf{X})^T \mathbf{y} \quad (5)$$

- ▶ What is  $\Phi(\mathbf{X})^T \Phi(\mathbf{X})$ ?
- ▶ What are the dimensions of  $\theta^*$  and  $\Phi(\mathbf{X})^T \Phi(\mathbf{X})$ ?



# solution of linear regression + regularization

- What is  $\Phi(\mathbf{X})^T \Phi(\mathbf{X})$ ?

$$\Phi(\mathbf{X})^T \Phi(\mathbf{X}) = \begin{pmatrix} \phi_0(\mathbf{x}_1) & \phi_0(\mathbf{x}_2) & \cdots & \phi_0(\mathbf{x}_N) \\ \phi_1(\mathbf{x}_1) & \phi_1(\mathbf{x}_2) & \cdots & \phi_1(\mathbf{x}_N) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_{d-1}(\mathbf{x}_1) & \phi_{d-1}(\mathbf{x}_2) & \cdots & \phi_{d-1}(\mathbf{x}_N) \\ \phi_d(\mathbf{x}_1) & \phi_d(\mathbf{x}_2) & \cdots & \phi_d(\mathbf{x}_N) \end{pmatrix} \begin{pmatrix} \phi_0(\mathbf{x}_1), & \phi_1(\mathbf{x}_1), & \cdots, & \phi_d(\mathbf{x}_1) \\ \phi_0(\mathbf{x}_2), & \phi_1(\mathbf{x}_2), & \cdots, & \phi_d(\mathbf{x}_2) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_0(\mathbf{x}_{N-1}), & \phi_1(\mathbf{x}_{N-1}), & \cdots, & \phi_d(\mathbf{x}_{N-1}) \\ \phi_0(\mathbf{x}_N), & \phi_1(\mathbf{x}_N), & \cdots, & \phi_d(\mathbf{x}_N) \end{pmatrix}$$

- $\phi_j(\mathbf{x}_i)$ , feature  $j$  in  $\phi(\cdot)$  for point  $i$

**Homework:** proof that  $\Phi(\mathbf{X})^T \Phi(\mathbf{X}) = \sum_i^N \phi(\mathbf{x}_i) \phi(\mathbf{x}_i)^T$

remember,  $\phi(\mathbf{x}_i)^T = [\phi_0(\mathbf{x}_i), \phi_1(\mathbf{x}_i), \cdots, \phi_d(\mathbf{x}_i)]$ , (vector of  $(1, d)$  dimensions)

## Kernel space

$$\theta^* = (\Phi(\mathbf{X})^T \Phi(\mathbf{X}) + \lambda \mathbb{I}_d)^{-1} \Phi(\mathbf{X})^T \mathbf{y} \quad (6)$$

Matrix identity (Eq. 167):  $(\mathbb{I} + \mathbf{AB})^{-1} \mathbf{A} = \mathbf{A} (\mathbb{I} + \mathbf{BA})^{-1}$

$$\theta^{**} = \Phi(\mathbf{X})^T (\Phi(\mathbf{X}) \Phi(\mathbf{X})^T + \lambda \mathbb{I}_N)^{-1} \mathbf{y} \quad (7)$$

- What is  $\Phi(\mathbf{X}) \Phi(\mathbf{X})^T$ ?

$$\Phi(\mathbf{X}) \Phi(\mathbf{X})^T = \begin{pmatrix} \phi_0(\mathbf{x}_1), & \phi_1(\mathbf{x}_1), & \cdots, & \phi_d(\mathbf{x}_1) \\ \phi_0(\mathbf{x}_2), & \phi_1(\mathbf{x}_2), & \cdots, & \phi_d(\mathbf{x}_2) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_0(\mathbf{x}_{N-1}), & \phi_1(\mathbf{x}_{N-1}), & \cdots, & \phi_d(\mathbf{x}_{N-1}) \\ \phi_0(\mathbf{x}_N), & \phi_1(\mathbf{x}_N), & \cdots, & \phi_d(\mathbf{x}_N) \end{pmatrix} \begin{pmatrix} \phi_0(\mathbf{x}_1) & \phi_0(\mathbf{x}_2) & \cdots & \phi_0(\mathbf{x}_N) \\ \phi_1(\mathbf{x}_1) & \phi_1(\mathbf{x}_2) & \cdots & \phi_1(\mathbf{x}_N) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_{d-1}(\mathbf{x}_1) & \phi_{d-1}(\mathbf{x}_2) & \cdots & \phi_{d-1}(\mathbf{x}_N) \\ \phi_d(\mathbf{x}_1) & \phi_d(\mathbf{x}_2) & \cdots & \phi_d(\mathbf{x}_N) \end{pmatrix}$$

- What are the matrix elements of  $\Phi(\mathbf{X}) \Phi(\mathbf{X})^T$ ,  $[\phi(\mathbf{x}_i)^T \phi(\mathbf{x}_j)]_{ij}$ ?

# Kernel space as linear model

Solution of standard linear regression,

$$f(\mathbf{x}, \boldsymbol{\theta}^*) = \sum_i^d \theta_i^* \phi(\mathbf{x})^i = \boldsymbol{\theta}^{*\text{T}} \boldsymbol{\phi}(\mathbf{x}) \quad (8)$$

$$= \left[ (\boldsymbol{\Phi}(\mathbf{X})^{\text{T}} \boldsymbol{\Phi}(\mathbf{X}) + \lambda \mathbb{I}_d)^{-1} \boldsymbol{\Phi}(\mathbf{X})^{\text{T}} \mathbf{y} \right]^{\text{T}} \boldsymbol{\phi}(\mathbf{x}) \quad (9)$$

OTHER solution,

$$f(\mathbf{x}, \boldsymbol{\theta}^{**}) = \boldsymbol{\theta}^{**\text{T}} \boldsymbol{\phi}(\mathbf{x}) = \boldsymbol{\phi}(\mathbf{x})^{\text{T}} \boldsymbol{\theta}^{**} \quad (10)$$

$$= \underbrace{\boldsymbol{\phi}(\mathbf{x})^{\text{T}} \boldsymbol{\Phi}(\mathbf{X})^{\text{T}}}_{\boldsymbol{\kappa}^{\text{T}}} \underbrace{(\boldsymbol{\Phi}(\mathbf{X}) \boldsymbol{\Phi}(\mathbf{X})^{\text{T}} + \lambda \mathbb{I}_d)^{-1} \mathbf{y}}_{\boldsymbol{\alpha}} \quad (11)$$

## Kernel space as linear model

$$f(\mathbf{x}, \theta^{**}) = \underbrace{\phi(\mathbf{x})^T \Phi(\mathbf{X})^T}_{\boldsymbol{\kappa}^T} \underbrace{(\Phi(\mathbf{X})\Phi(\mathbf{X})^T + \lambda \mathbb{I}_d)^{-1} \mathbf{y}}_{\boldsymbol{\alpha}} \quad (12)$$

$$= \phi(\mathbf{x})^T \theta^{**} = \boldsymbol{\kappa}^T \boldsymbol{\alpha} = \sum_i^N \kappa(\mathbf{x}, \mathbf{x}_i) \alpha_i \quad (13)$$

$$(14)$$

- What is  $\phi(\mathbf{x})^T \Phi(\mathbf{X})^T$  and/or  $\Phi(\mathbf{X})\Phi(\mathbf{X})^T$ ?

$$\begin{aligned} \phi(\mathbf{x})^T \Phi(\mathbf{X})^T &= (\phi_0(\mathbf{x}), \quad \phi_1(\mathbf{x}), \quad \dots, \quad \phi_d(\mathbf{x})) \begin{pmatrix} \phi_0(\mathbf{x}_1) & \phi_0(\mathbf{x}_2) & \dots & \phi_0(\mathbf{x}_N) \\ \phi_1(\mathbf{x}_1) & \phi_1(\mathbf{x}_2) & \dots & \phi_1(\mathbf{x}_N) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_{d-1}(\mathbf{x}_1) & \phi_{d-1}(\mathbf{x}_2) & \dots & \phi_{d-1}(\mathbf{x}_N) \\ \phi_d(\mathbf{x}_1) & \phi_d(\mathbf{x}_2) & \dots & \phi_d(\mathbf{x}_N) \end{pmatrix} \\ &= (\phi(\mathbf{x})^T \phi(\mathbf{x}_0), \quad \phi(\mathbf{x})^T \phi(\mathbf{x}_1), \quad \dots, \quad \phi(\mathbf{x})^T \phi(\mathbf{x}_N)) \end{aligned}$$

- Do we need to compute  $\phi(\mathbf{x}_i)^T \phi(\mathbf{x}_j)$ ?

# Kernel function

- $\kappa(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)^\top \phi(\mathbf{x}_j)$ 
  - ▶  $\kappa(\mathbf{x}_i, \mathbf{x}_j)$  is the dot or inner product between functions,

$$\kappa(\mathbf{x}_i, \mathbf{x}_j) = \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle_\nu \quad (15)$$

- (From Wikipedia): On the other hand, an explicit representation for  $\varphi$  is not necessary, as long as  $\nu$  is an inner product space.
- For  $\phi(\cdot)$  as an infinite polynomial,  $\kappa(\mathbf{x}_i, \mathbf{x}_j)$  is the **Squared Exponential Kernel**,

$$\kappa(\mathbf{x}_i, \mathbf{x}_j) = \sigma^2 \exp^{-\frac{(\mathbf{x}_i - \mathbf{x}_j)^2}{2\ell^2}} \quad (16)$$

- How are we optimizing  $\sigma$  and  $\ell$  ?
- What are the **free-parameters** of this class of models?

# Kernel functions

The Kernel Cookbook by David Duvenaud

Sklearn tutorial

Faster optimization of kernel ridge regression

# Lasso regression

MUST STUDY THIS ON YOUR OWN!

$$\begin{aligned}\mathcal{L}(\theta) &= \frac{1}{2} \sum_i^N (y_i - \theta^T \phi(\mathbf{x}_i))^2 + \underbrace{\frac{\lambda}{2} \sum_j |w_j|}_{\text{absolute value}} \\ &= \frac{1}{2} (\mathbf{y} - \theta^T \Phi(\mathbf{X}))^T (\mathbf{y} - \theta^T \Phi(\mathbf{X})) + \underbrace{\frac{\lambda}{2} \|\mathbf{w}\|_1}_{\text{regularization}}\end{aligned}$$

Useful links:

- ▶ [Wikipedia](#)
- ▶ [original paper](#)
- ▶ [Sklern tutorial](#)

- What is Lasso used for?