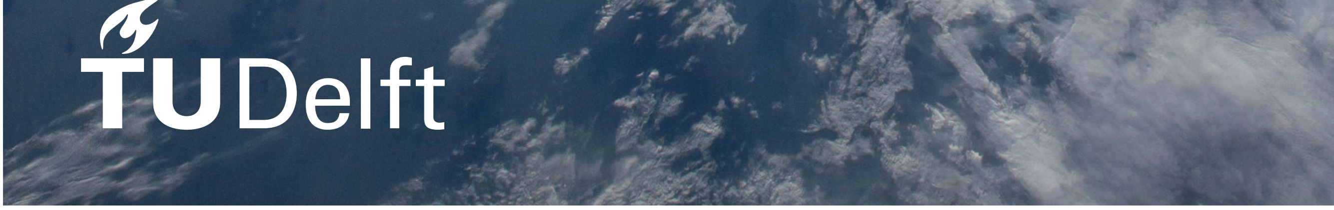
TU Delft

Foundations of Machine Learning

learning note For reading translation

by

我真的不懂忧郁

Student Name	Student Number
First Surname	1234567

Instructor:	I. Surname
Teaching Assistant:	I. Surname
Project Duration:	Month, Year - Month, Year
Faculty:	Faculty of Aerospace Engineering, Delft

TU Delft

Cover:	Canadarm 2 Robotic Arm Grapples SpaceX Dragon by NASA under CC BY-NC 2.0 (Modified)
Style:	TU Delft Report Style, with modifications by Daan Zwaneveld

Preface

A preface...

我真的不懂忧郁
Delft, September 2024

Summary

A summary...

目录

Nomenclature

If a nomenclature is required, a simple template can be found below for convenience. Feel free to use, adapt or completely remove.

Abbreviations

Abbreviation	Definition
ISA	International Standard Atmosphere
...	

Symbols

Symbol	Definition	Unit
V	Velocity	[m/s]
...		
ρ	Density	[kg/m^3]
...		

Chapter 1

Kernel Methods

1.1. Introduction

$K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ 称为 $\mathcal{X}$ 上的 **Kernels**。

theorem 1.1.1: ***(Mercer's condition)*** 令 $\mathcal{X} \subset \mathbb{R}^N$ 是一个紧集[a]，$K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ 是一个对称连续函数，则

$$K(x, x') = \sum_{n=0}^{\infty} \lambda_n \phi_n(x) \phi_n(x'),\ a_n > 0\ is\ eigenvalue \tag{1.1}$$

当且仅当 $\forall c \in L^2(\mathcal{X})$，下面的条件成立

$$\int \int_{\mathcal{X} \times \mathcal{X}} c(x) c(x') K(x, x') dx dx' \geqslant 0 \tag{1.2}$$

[a] $\mathcal{X}$ 是紧集，则存在有限个开覆盖

proof.

□

Mercer's condition 是核方法中的一个重要概念，尤其在支持向量机（SVM）和核函数的理论中起着关键作用。它为一个函数能否作为合法的核函数提供了数学判据。合法的核函数用于将数据从低维空间映射到高维空间，在高维空间中可以更加容易地进行线性分割。

1.2. Positive definite symmetric kernel

$K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ 称为**正定核** (*positive definite symmetric,PDS*)，当对于任何 $\{x_1, \cdots, x_m\} \subseteq \mathcal{X}$，矩阵

$$\mathbf{K} = [K(x_i, x_j)]_{ij} \in \mathbb{R}^{m \times m} \tag{1.3}$$

是半正定对称矩阵，即 $\forall \mathbf{c} = (c_1, \cdots, c_m)^T \in \mathbb{R}^{m\times 1}$，

$$\mathbf{c}^T\mathbf{K}\mathbf{c} = \sum_{i,j=1}^{n} c_i c_j K(x_i, x_j) \geqslant 0 \tag{1.4}$$

example 1.2.1: ***(Polynomial Kernels)***

example 1.2.2: ***(Gaussian Kernels)***

example 1.2.3: ***(Sigmoid Kernels)***

证明 S^2 是 R^3 的光滑子流形

1.3. Reproducing kernel Hilbert Space

theorem 1.3.1: 令 $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ 是一个 *PDS* 核，则存在一个 *Hilbert Space* $\mathbb{H}$ 以及 $\Phi : \mathcal{X} \to \mathbb{H}$，使得

$$\forall x, x; \in \mathcal{X},\ \ K(x, x') = \langle \Phi(x), \Phi(x') \rangle \tag{1.5}$$

$\mathbb{H}$ 有如下名为再生（*Reproducing*）的性质

$$\forall h \in \mathbb{H}, \forall x \in \mathcal{X},\ h(x) = \langle h, K(x, \cdot) \rangle \tag{1.6}$$

$\mathbb{H}$ 称为再生核希尔伯特空间 *(reproducing kernel Hilbert Space,RKHS)*。

proof.

□

Normlized PDS Kernels

lemma 1.3.2: 令 K 是一个 *PDS kernel*，则 K 的规范核 K' 也是 *PDS kernel.*

PDS Kernels Closure Properies

theorem 1.3.3: *PDS kernel* 在和、积、张量积、逐点极限下是闭集，且可以展开成幂级数

$$\sum_{n=0}^{\infty} a_n x^n,\ a_n \geqslant 0\ for\ \forall n \in \mathbb{N} \tag{1.7}$$

Chapter 2

基于流形的学习

2.1. PCA 和 LDA

2.2. 拓扑流形的概念

2.3. 多尺度变换

保持度量不变

2.4. 局部线性嵌入

保持线性结构不变

2.5. 拉普拉斯特征映射

近邻图，拉普拉斯矩阵

2.6. 核函数与度量——NDS 核

2.7. 理论成果

References

[1] I. Surname, I. Surname, and I. Surname. “The Title of the Article”. In: *The Title of the Journal* 1.2 (2000), pp. 123–456.

Chapter A

Source Code Example

Adding source code to your report/thesis is supported with the package `listings`. *An example can be found below. Files can be added using* `\lstinputlisting[language=<language>]{<filename>}`.

```python
"""
ISA Calculator: import the function, specify the height and it will return a
list in the following format: [Temperature,Density,Pressure,Speed of Sound].
Note that there is no check to see if the maximum altitude is reached.
"""

import math
g0 = 9.80665
R = 287.0
layer1 = [0, 288.15, 101325.0]
alt = [0,11000,20000,32000,47000,51000,71000,86000]
a = [-.0065,0,.0010,.0028,0,-.0028,-.0020]

def atmosphere(h):
    for i in range(0,len(alt)-1):
        if h >= alt[i]:
            layer0 = layer1[:]
            layer1[0] = min(h,alt[i+1])
            if a[i] != 0:
                layer1[1] = layer0[1] + a[i]*(layer1[0]-layer0[0])
                layer1[2] = layer0[2] * (layer1[1]/layer0[1])**(-g0/(a[i]*R))
            else:
                layer1[2] = layer0[2]*math.exp((-g0/(R*layer1[1]))*(layer1[0]-layer0[0]))
    return [layer1[1],layer1[2]/(R*layer1[1]),layer1[2],math.sqrt(1.4*R*layer1[1])]
```

Chapter B

Task Division Example

If a task division is required, a simple template can be found below for convenience. Feel free to use, adapt or completely remove.

表 **B.1:** Distribution of the workload

	Task	Student Name(s)
	Summary	
Chapter 1	Introduction	
Chapter 2		
Chapter 3		
Chapter *		
Chapter *	Conclusion	
	Editors	
	CAD and Figures	
	Document Design and Layout	