Begonnen am	Donnerstag, 7. Dezember 2023, 12:45
Status	Beendet
Beendet am	Freitag, 8. Dezember 2023, 14:57
Verbrauchte Zeit	1 Tag 2 Stunden
Bewertung	**28,50** von 31,00 (**91,94**%)

Frage 1

Teilweise richtig

Erreichte Punkte 0,50 von 1,00

The dimensions of the kernel feature space can be...

Wählen Sie eine oder mehrere Antworten:

- a. ... of infinite dimensions ✓
- b. ... of lower dimension than the original data space
- c. ... of higher dimension than the original data space ✓
- d. ... of the same dimensions as the original data space

Your answer is partially correct.

Sie haben 2 richtig ausgewählt.
Die richtigen Antworten sind: ... of higher dimension than the original data space, ... of lower dimension than the original data space, ... of infinite dimensions, ... of the same dimensions as the original data space

Frage 2

Richtig

Erreichte Punkte 2,00 von 2,00

K is symmetrix，正定

↑ det >0

Assume that you have a dataset consisting of 2 points $\mathbf{x}_i \in \mathbb{R}^3$. You apply a **linear kernel** to your dataset. Which of the following matrices may be the resulting kernel matrix?

K中的 k_{ij} 是两点的点积 $x_i \cdot x_j$

K.size() → 2x2

- a. $\mathbf{K} = \begin{bmatrix} 3 & -4 \\ -4 & 5 \end{bmatrix}$ det K = 15 - 16 = -1 < 0
- b. $\mathbf{K} = \begin{bmatrix} 4 & 1 & 3 \\ 1 & 8 & 8 \\ 3 & 8 & 11 \end{bmatrix}$
- c. $\mathbf{K} = \begin{bmatrix} 6 & 2 & 1 \\ 2 & 8 & 2 \\ 1 & 2 & 7 \end{bmatrix}$
- d. $\mathbf{K} = \begin{bmatrix} 2 & 3 \\ 3 & 1 \end{bmatrix}$ det K = 2 - 9 < 0
- e. $\mathbf{K} = \begin{bmatrix} 12 & -3 \\ -3 & 1 \end{bmatrix}$ ✓ det A = 12 - 9 = 3
- f. $\mathbf{K} = \begin{bmatrix} 8 & 1 & 0 \\ 4 & 3 & 2 \\ 5 & 2 & 9 \end{bmatrix}$ ← nicht Symmetrix

Die Antwort ist richtig.

Die richtige Antwort ist:

$$\mathbf{K} = \begin{bmatrix} 12 & -3 \\ -3 & 1 \end{bmatrix}$$

Frage 3

Richtig

Erreichte Punkte 3,00 von 3,00

Your work colleague wants to train a kernelized classifier at work and computes the kernel matrix by hand. However, some entries went missing, and you want to help him reconstruct the original kernel matrix. He was using a **Gaussian kernel**. Drag the correct values inside the matrix.

e^0

$$\mathbf{K} = \begin{bmatrix} \boxed{1.0} & \boxed{0.6} & 0.5 \\ 0.6 & \boxed{1.0} & \boxed{0.3} \\ \boxed{0.5} & 0.3 & \boxed{1.0} \end{bmatrix}$$

$\boxed{0.0}$ $\boxed{0.1}$ $\boxed{0.2}$ $\boxed{0.3}$ $\boxed{0.4}$ $\boxed{0.5}$ $\boxed{0.6}$ $\boxed{0.7}$ $\boxed{0.8}$ $\boxed{0.9}$ $\boxed{1.0}$

Die Antwort ist richtig.

Frage 4

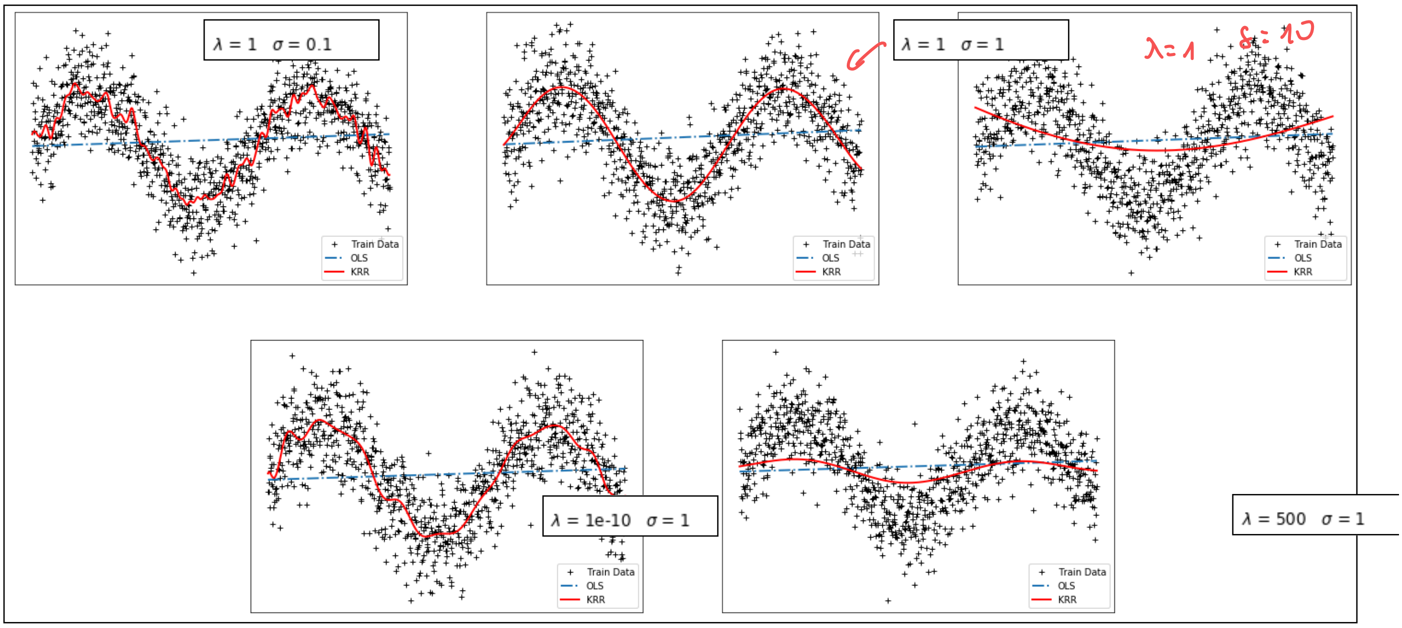
Richtig

Erreichte Punkte 5,00 von 5,00

We want to study the effect of the two hyperparameters for Kernel Ridge Regression:

- The regularization parameter λ
- The kernel width σ of the Gaussian kernel

Assign the proper values of kernel width and regularization parameter to each figure. (*Hint: look closely at the figure in the bottom left)*



Your answer is correct.

Frage 5

Richtig

Erreichte Punkte 4,00 von 4,00

Insert your implementation of the function train_krr() from the notebook. Make sure to use inputs and returns of the function just as described in the notebook.

We use the notation from assignment 3, $X_{\text{train}} \in \mathbb{R}^{D_X \times N_{tr}}, \; Y_{\text{train}} \in \mathbb{R}^{D_Y \times N_{tr}}, \; X_{\text{test}} \in \mathbb{R}^{D_X \times N_{te}}$

The function train_krr() should estimate a linear combination of the input vectors α, $\alpha = (K + \lambda I)^{-1} Y_{\text{train}}^{\mathsf{T}}$ where λ is the regularization parameter and K is the $N_{tr} \times N_{tr}$ Gaussian Kernel matrix with Kernel width σ, $K_{ij} = \exp\left(-\frac{\|X^i_{\text{train}} - X^j_{\text{train}}\|^2}{\sigma^2}\right)$. You can compute K with the provided function GaussianKernel() from the notebook.

Antwort: (Abzugssystem: 0, 0 %)

Antwort zurücksetzen

Ace-Editor nicht bereit. Vielleicht Seite neu laden?
Rückgriff auf ein Rohtext-Feld.

```
import scipy as sp
from numpy.linalg import inv
from numpy.linalg import solve
from scipy.io import loadmat
import numpy as np
from scipy.spatial.distance import cdist

def GaussianKernel(X1, X2, kwidth):
    ''' Compute Gaussian Kernel
    Input: X1     - DxN1 array of N1 data points with D features
           X2     - DxN2 array of N2 data points with D features
           kwidth - Kernel width
    Output K      - N1 x N2 Kernel matrix
    '''
    assert(X1.shape[0] == X2.shape[0])
    K = cdist(X1.T, X2.T, 'sqeuclidean')
    K = np.exp(-K / (2. * kwidth ** 2))
    return K
```

	Test	Erwartet	Erhalten	

✓	`np.random.seed(5)`
	`X = np.random.rand(6,30)`
	`Y = np.random.rand(1,30)`
	`w = train_krr(X, Y,1,0)`
	`print(w[:2])`
	`print(w[-2:])`
		`[[-8.29690881]`
		` [-6.10288201]]`
		`[[  4.1884504 ]`
		` [-16.14957306]]`
			`[[-8.29690881]`
			` [-6.10288201]]`
			`[[  4.1884504 ]`
			` [-16.14957306]]`
				✓

✓	`np.random.seed(5)`
	`X = np.random.rand(60,10)`
	`Y = np.random.rand(1,10)`
	`w = train_krr(X, Y,10, 0.02)`
	`print(w[:2])`
	`print(w[-2:])`
		`[[-4.60243442]`
		` [ 2.86107076]]`
		`[[-4.5341411 ]`
		` [ 3.32084586]]`
			`[[-4.60243442]`
			` [ 2.86107076]]`
			`[[-4.5341411 ]`
			` [ 3.32084586]]`
				✓

Alle Tests bestanden! ✓

Richtig

Bewertung für diese Einreichung: 4,00/4,00.

Frage **6**

Richtig

Erreichte Punkte 2,00 von 2,00

Insert your implementation of the function apply_krr() from the notebook. Make sure to use inputs and returns of the function just as described in the notebook.

We use the notation from assignment 3, $X_{\text{train}} \in \mathbb{R}^{D_X \times N_{tr}},\ Y_{\text{train}} \in \mathbb{R}^{D_Y \times N_{tr}},\ X_{\text{test}} \in \mathbb{R}^{D_X \times N_{te}}$

The function apply_krr() uses the weights α to predict the (unknown) hand positions of new test data X_{test}:
$Y_{\text{test}} = (\mathbf{k}\alpha)^{\top}$.

where $\mathbf{k}$ is the $N_{\text{test}} \times N_{\text{train}}$ matrix $\mathbf{k}_{ij} = \exp\left(-\frac{\|X^i_{\text{test}} - X^j_{\text{train}}\|^2}{\sigma^2}\right)$.

Antwort: (Abzugssystem: 0, 0 %)

Antwort zurücksetzen

Ace-Editor nicht bereit. Vielleicht Seite neu laden?
Rückgriff auf ein Rohtext-Feld.

```
import scipy as sp
from numpy.linalg import inv
from numpy.linalg import solve
from scipy.io import loadmat
import numpy as np
import numpy as sp

from scipy.spatial.distance import cdist

def GaussianKernel(X1, X2, kwidth):
    ''' Compute Gaussian Kernel
    Input: X1    - DxN1 array of N1 data points with D features
           X2    - DxN2 array of N2 data points with D features
           kwidth - Kernel width
    Output K     - N1 x N2 Kernel matrix
    '''
    assert(X1.shape[0] == X2.shape[0])
    K = cdist(X1.T, X2.T, 'sqeuclidean')
```

	Test	Erwartet	Erhalten	

✓	`sp.random.seed(5)`
	`X_train = np.random.rand(6,8)`
	`X_test = np.random.rand(6,4)`
	`Y = np.random.rand(1,8)`
	`w = np.array([[-10.88806291],[-14.04762665],[ 9.88494076],[ -6.68802798],[  3.54537817],[ -3.43169789],[  2.86056669],[ 19.34212098]])`
	`Y_test = apply_krr(w, X_train, X_test, 10)`
	`print(Y_test)`
		`[[0.52063939 0.54835119 0.54108331 0.56645971]]`	`[[0.52063939 0.54835119 0.54108331 0.56645971]]`	✓

✓	`sp.random.seed(42)`
	`X_train = np.random.rand(6,8)`
	`X_test = np.random.rand(6,4)`
	`Y = np.random.rand(1,8)`
	`w = np.array([[-10.88806291],[-14.04762665],[ 9.88494076],[ -6.68802798],[  3.54537817],[ -3.43169789],[  2.86056669],[ 19.34212098]])`
	`Y_test = apply_krr(w, X_train, X_test, 10)`
	`print(Y_test)`
		`[[0.5310481 0.50319424 0.51638826 0.54503093]]`	`[[0.5310481 0.50319424 0.51638826 0.54503093]]`	✓

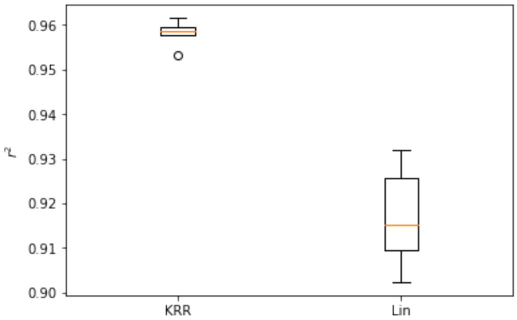
Alle Tests bestanden! ✓

Richtig

Bewertung für diese Einreichung: 2,00/2,00.

Information

The image below shows a boxplot for the linear regression and the Kernel Ridge Regression.



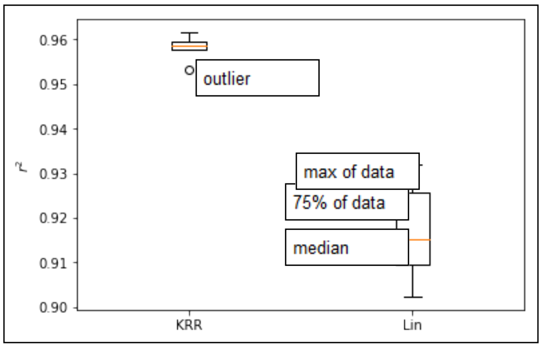
Check the help function in python or the wikipedia article to familarize yourself with boxplots to answer the following questions

Frage 7

Richtig

Erreichte Punkte 1,00 von 1,00

Drop the labels at the right position on the image.



Your answer is correct.

Frage **8**

Richtig

Erreichte Punkte 4,00 von 4,00

Based on the image shown above, complete the following sentences:

Better performance is achieved with the [kernel ridge regression] ✓ . The kernel ridge regression has [higher] ✓ accuracy than the linear regressions while the linear regression has [higher] ✓ variance than kernel ridge regression. Outliers do not occur for [linear regression] ✓ . Judging from the boxplot, the [kernel ridge regression] ✓ algorithm is preferable for this task.

When large data sets are used, [kernel ridge regression] ✓ can become significantly slower than [linear regression] ✓ because [it compares new data points to all known data points] ✓ .

Your answer is correct.

Die richtige Antwort lautet:
Based on the image shown above, complete the following sentences:

Better performance is achieved with the [kernel ridge regression]. The kernel ridge regression has [higher] accuracy than the linear regressions while the linear regression has [higher] variance than kernel ridge regression. Outliers do not occur for [linear regression]. Judging from the boxplot, the [kernel ridge regression] algorithm is preferable for this task.

When large data sets are used, [kernel ridge regression] can become significantly slower than [linear regression] because [it compares new data points to all known data points].

Frage 9

Richtig

Erreichte Punkte 4,00 von 4,00

Use the provided snippets of code to reconstruct the nested cross validation algorithm as presented in the lecture

1. Split data in F disjunct folds
2. for outer folds f_{outer} = 1, . . . , F do
3. Pick folds {1, . . . , F } \ f_{outer} for Model Selection
4. **# Model Selection**
5. for Fold f_{inner} = 1, . . . , F − 1 do
6. for Parameter s = 1, . . . , S do
7. Train model on folds {1, . . . , F } \ {f_{outer} ,f_{inner} } with parameter sigma_s
8. Compute prediction on fold f_{inner}
9. END FOR
10. END FOR
11. Pick best parameter sigma_s for all f_{inner}
12. **# Model Evaluation**
13. Train model on folds {1, . . . , F } \ f_{outer} with parameter sigma_s
14. Test model on fold f_{outer}
15. END FOR
16. RETURN Average Performance on Outer Folds

Your answer is correct.

Die richtige Antwort lautet:

Use the provided snippets of code to reconstruct the nested cross validation algorithm as presented in the lecture

1. [Split data in F disjunct folds]
2. [for outer folds f_{outer} = 1, . . . , F do]
3. [Pick folds {1, . . . , F } \ f_{outer} for Model Selection]
4. **# Model Selection**
5. for Fold f_{inner} = 1, . . . , F − 1 do
6. [for Parameter s = 1, . . . , S do]
7. [Train model on folds {1, . . . , F } \ {f_{outer} ,f_{inner} } with parameter sigma_s]
8. [Compute prediction on fold f_{inner}]
9. END FOR
10. END FOR

11. [Pick best parameter sigma_s for all f_{inner}]
12. **# Model Evaluation**
13. [Train model on folds {1, . . . , F } \ f_{outer} with parameter sigma_s]
14. [Test model on fold f_{outer}]
15. END FOR
16. RETURN Average Performance on Outer Folds

Information

In the following tasks, we want to compare the running time of kernelized Ridge Regression with Ordinary Least Squares Regression.

You have a dataset $X \in \mathbb{R}^{d \times n}$ with n elements of dimensionality d, and targets $Y \in \mathbb{R}^n$.

FYI: the year is 1888 and matrix inversion takes $\mathcal{O}(n^3)$, while matrix multiplication takes $\mathcal{O}(abc)$ for matrices of dimension $a \times b$ and $b \times c$

Hint: assume that computing the kernel function $k(x, x')$ takes t time

To get full points, you are required to give the smallest upper-bounding running time and simplify as much as possible!

Frage 10

Falsch

Erreichte Punkte 0,00 von 1,00

Computing the OLS weight vector takes $\mathcal{O}($ n^3 $d^2 \cdot n + d^3$ $)$ time.

OLS: $d \times 1$, $d \times n$, $n \times 1$, $d \times d$, $d \times n$, $d \times n$, $n \times d$

$w = (\underbrace{XX^T}_{d^2 \cdot n})^{-1} X y^T$

Ihre letzte Antwort wurde folgendermaßen interpretiert:

$$n^3$$

In Ihrer Antwort wurden die folgenden Variablen gefunden: $[n]$

$O(d^2 \cdot n + n^3 + d^2 n)$

$d^2 n + n^3$

Falsche Antwort.

Eine richtige Antwort ist $d^2 \cdot n + d^3$. Sie kann so eingegeben werden: `d^2*n+d^3`

Frage 11

Falsch

Erreichte Punkte 0,00 von 1,00

$n^2 t + n^3$

Computing α for KRR takes $\mathcal{O}($ n^3 $)$ time. $w = (XX^T + \lambda I)^{-1} X y^T$

$d^2 n$

Ihre letzte Antwort wurde folgendermaßen interpretiert:

$\alpha = (K + \lambda I)^{-1} y^T$

$$n^3$$

In Ihrer Antwort wurden die folgenden Variablen gefunden: $[n]$ $O(n^2 \cdot t + n^3)$

Falsche Antwort.

Eine richtige Antwort ist $n^2 \cdot t + n^3$. Sie kann so eingegeben werden: `n^2*t+n^3`

Frage 12

Richtig

Erreichte Punkte 1,00 von 1,00

Predicting the output for a new point in OLS takes $\mathcal{O}($ d $)$ time

Ihre letzte Antwort wurde folgendermaßen interpretiert:

$w \cdot x$

$$d$$

$1 \times d$ $d \times 1$

In Ihrer Antwort wurden die folgenden Variablen gefunden: $[d]$

Richtige Antwort, gut gemacht!

Eine richtige Antwort ist d. Sie kann so eingegeben werden: `d`

Frage 13

Richtig

Erreichte Punkte 1,00 von 1,00

Predicting the output for a new point using KRR takes $\mathcal{O}($ t*n $)$ time.

Ihre letzte Antwort wurde folgendermaßen interpretiert:

$$t \cdot n$$

$y = \sum_{i=1}^{n} \alpha_i k(x_i, x_{new})$

In Ihrer Antwort wurden die folgenden Variablen gefunden: $[n, t]$

Richtige Antwort, gut gemacht!

Eine richtige Antwort ist $n \cdot t$. Sie kann so eingegeben werden: `n*t`

Frage 14

Richtig

Erreichte Punkte 1,00 von 1,00

If you have a dataset with many dimensions but few data points, from a purely runtime perspective it would be ______ to use Kernel Ridge Regression instead of OLS

Wählen Sie eine Antwort:

- a. Less efficient
- b. More efficient ✓
- c. The same
- d. It depends on data set

Die Antwort ist richtig.

Die richtige Antwort ist: More efficient