Medical Data Science

WS 2022/23

Prof. Dr. Nico Pfeifer

1	2	3	4	$\sum$

Marina Dittschar & Clarissa Auckenthaler

Assignment 2

(Submission date 25.11.2022)

Problem 1 (T, 15 Points)

(a) A kernel matrix is a symmetric and positive (semi-)definite matrix. Explain in your own words the term positive semi definite. (1P)

Positive Semidefinite Matrix: An n × n matrix A is positive semidefinite if[2]:

$$v^T H v \geq 0, v \in \mathbb{R}^n \tag{1}$$

A matrix is positive definite if:

$$v^T H v > 0, v \in \mathbb{R}^n \tag{2}$$

This means that if oyu have any vector and you multiply its transverse with a matrix H and with the vector again the result needs to be positive for positive definite matrices and greater or equal than zero for positive semi-definite matrices.

(b) What are differences and similarities between the weighted degree kernel (WDK) and the weighted degree kernel with shifts (WDS)?Give one example where the kernel values are the same and one example where they are different.(3P)

- WDK: compares all subsequences in a certain length at the same position of two sequences.
- WDKS: can detect similarities between two sequences even if they are shifted.

The similaritie beteween both kernels is, that both compare two sequences at a certain position. For the WDK they look always at the same position and for the WDKS the possition is not equal to the position in sequence one, there is a maximal shift defined to look on both sides of the subsequence position.

If there is no shift, the kernel values are the same, because the double identity function cancels out the 0.5 factor δ_s. If there is a shift however, the values diverge.

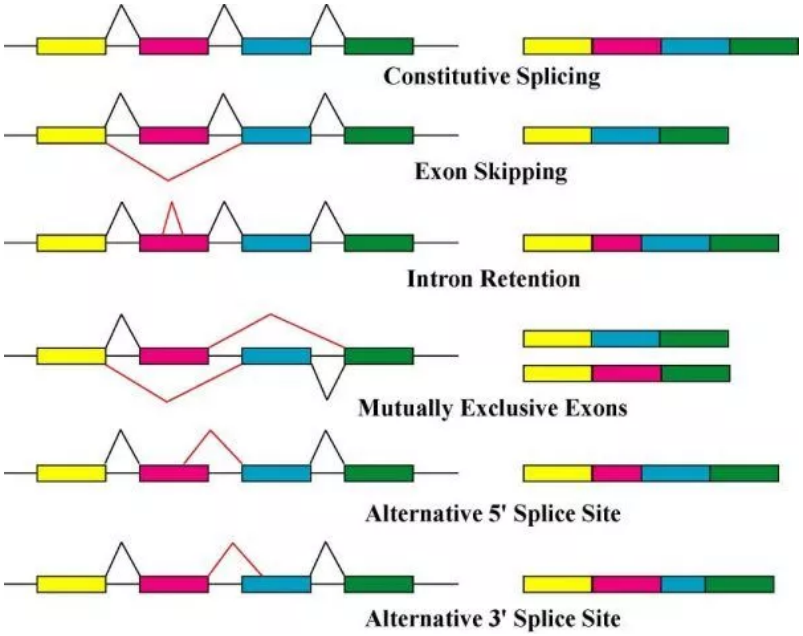

Figure 1: Examples of alternative splicing.[1]

(c) Describe shortly the concept of alternative splicing using the terms exon, intron, donor and acceptor splice site. How does splicing influence the number of possible proteins?(3P)

The mRNA has different regions called introns and exons, to translate it into a functional protein the introns need to be removed before the translation into a protein.
The method alternative splicing enables to create many proteins from the same strand of DNA by using different options to stitch exons together.
Examples: In figure1 the 'constitutive Splicing' is the simplest way in remove the introns. The yellow bar is the first exon and the line starts at the donor site of the exon and ends at the acceptor site of the second exon (pink bar) the straight line between the bars displays the intron which we want to remove. In the constitutive splicing each exon is connected. Another possible splicing is the 'exon skipping' (position 2 in figure1 here the second exon (pink bar) is skiped, we go directly from the donor site of exon 1 to the acceptor site of exon 3, at the end we only use exon 1, 3, 4 to translate the protein strand.
Alternative splicing enables the producion of may more proteins than the initial DNA might indicate e.g. humans have 24.000 protein coding genes, but they also have around 100.000 different proteins.

(d)Why does the problem of domain adaptation emerge in machine learning? Describe two approaches to domain adaptation.(3P)

The problem occurs because there can also be more than one source domain, it exists to deal with multiple sources to improve prediction on the target domain.
To generate a model out of the convex combination of SVM_S and SVM_T is a simple approach of domain adapation. SVM_S uses the source data for the training, for the optimazation of the hyper-parameter is uses the available target data. The SVM_T uses for training and model selection the target data. In the convex combination approach the parameter α is optimized by using the evaluation set of the target domain.
Another approach is to use the weight vector, which is used by multi-task learning here the source domain and the target domain are considers as a task. There for each task learning a separate model and connecting at the same time the parameter learning.

(e) Explain the multitask learning approach of the lecture and discuss whether multitask learning with many source domains or dualtask learning lead to better results. What can you say about the position of two really similar tasks? (3P)

Multitask learning as discussed in the lecture is extending the dual task learning approach by regularizing the pairwise distances of the weight vectors of each task (see lecture 4, slide 11). If the additional source organisms are further apart from the target organism that the original source organism, multitask learning performs worse than dualtask learning. If the source organisms have similar distance however, then the multitask learning performs better than dualtask learning.

(f) What is the difference between the Major Histocompatibility Complex (MHC) and human leukocyte antigens (HLA) molecules in the immune system? What is the main purpose of MHC I?(1P)

Major Histocompatibility Complex (MHC) is part of the immune system, in the human (MHC) molecules are called human leukocyte antigens (HLA) molecules. So there is no differenc exept the naming in the human immune system. There are two different major classes: MHC class I / HLA class I and MHC class II / HLA class II. The main purpose of MHC I is to display peptide fragments of within the cell to T-cells $\Longrightarrow$ this triggers an immediate response from the immune system.

(g) Describe what leveraging is. Heckerman et al. might help. (1P)

Leveraging describes the process of encoding of the data and thus enabling data sharing. It can be performed by including task information into the features. This features than

can be used in several learning approaches such as SVM, logistic regression, etc..

Problem 2 (T, 8 Points)

Show, how we can calculate the squared euclidean distance between two samples in this Hilbert space$||\phi(xi) - \phi(xj)||^2$ without using the mapping function ϕ.
Squared euclidean distance:

$$d(x, y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$$

Normed vector space:

$$d(x, y) = ||x - y||$$

Inner product space:The inner product space refers to the dot product with scalars in Euclidean space. $\longrightarrow$Euclidean norm:

$$[x_1 x_2 ... x_m] * \begin{matrix}[x_1 \\ x_2 \\ ... \\ x_m]\end{matrix} = \sqrt{x_1^2 + x_2^2 + ... + x_m^2}$$

Taking the equations from above into account $||\phi(xi) - \phi(xj)||^2$ can also be written as:

$$\begin{aligned} d(x_i, xj)^2 &= ||\phi(xi) - \phi(xj)||^2 \\ &=< \phi(xi) - \phi(xj), \phi(xi) - \phi(xj) > \\ =< \phi(xi), \phi(xi) > + < \phi(xj), \phi(xj) > -2 < \phi(xi), \phi(xj) > \\ &= k(x_i, x_i) + k(x_j, x_j) - 2 * k(x_i, x_j). \end{aligned}$$

Problem 3 (P, 9 Points)

Implement the weighted degree kernel (without shifts) in python and compute and visualize kernel matrices. The data can be found in sequencesMSA.fasta.

(a) Implement the weighted degree kernel as a function in python that takes two sequences as well as the parameter d, and the β parameters. The output should be the kernel value as defined in slide 21 of lecture 4. (7P)

You can see our result in Listing 1.

```python
def weighted_degree_kernel(s1,s2,d,betas):
  K= 0
  for k in range (1,d+1):
      I= 0
      beta = betas[k-1]
      for l in range (0,(len(s1)-k+1)):
          if (s1[l:l+k]==s2[l:l+k]):
              I = I+1
      K= K+ (beta * I)
  return K
```

Listing 1: Function for a weighted degree kernel without shifts

(b) Visualize the kernel matrix for d = 3 and $\beta_k = 2(d-k+1)/(d(d+1))$ for the 8 sequences from sequencesMSA.fasta. The visualization should show a 8 x 8 matrix representing the kernel values in a heat map.(2P)

You can see our result in Figure 2.

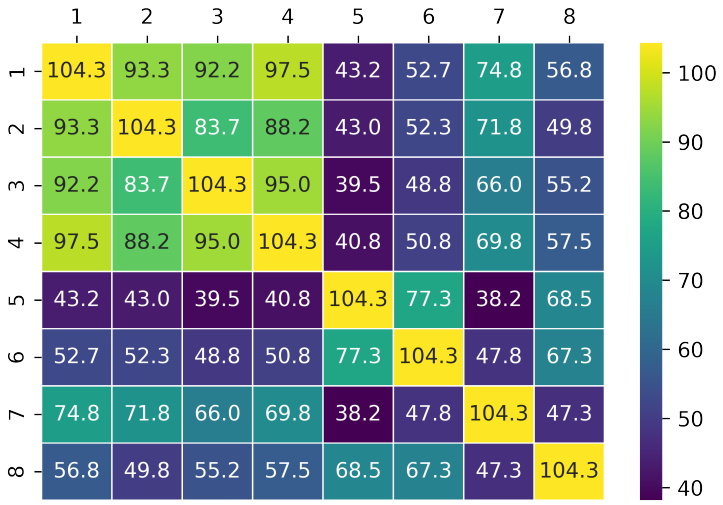

Figure 2: Our kernel matrix filled with values calculated by the function from 3a

Problem 4 (P, 18 Points)

Implement a kernel-based multitask learning approach in Python to predict HLA class I peptide binding. For the peptide kernel use the linseq approach presented in the fifth lecture. Use ten-fold crossvalidation to estimate performances (running over different values of the cost parameter C of the SVM). The data can be found in BindingData.csv. The binding class label is in the last column.

(a) Implement the Dirac, uniform, multitask and peptide kernel functions. (3P)

```
def dirac(a1, a2):
    """ dirac kernel function
    a1(String): allele 1
    a2(String): allele 1
    return K(float): kernel value
    """
    K = 0
    if a1 == a2:
        K = 1
    return K

def uniform(a1, a2):
    """ uniform kernel function
    a1(String): allele 1
    a2(String): allele 1
    return K(float): kernel value
    """
    return 1

def multitask(a1, a2):
    """ multistask kernel function combination of dirac & uniform kernel
    function
    a1(String): allele 1
    a2(String): allele 1
    return K(float): kernel value
    """
    K = dirac(a1, a2) + uniform(a1, a2)
    return K

def peptide(p1,p2):
    """ peptide kernel function
    p1(String):peptide sequence 1,
    p2(String):peptide sequence 2,
    return K(float): kernel value
    """
```

```
    K= 0
    for l in range (len(p1)):
        if (p1[l]==p2[l]):
                K = K+1

    return K
```

Listing 2: Kernel function

(b) Build SVM models using the Dirac kernel, the uniform kernel, and a multitask kernel (consisting of the two former kernels) in combination with the peptide kernel in the cross-validations with $C \in \{10^{-4}, 10^{-3}, ..., 10^{4}\}$.(7P)

Find our code in the attached Jupyter Notebook: `MDS_Assignment02_Auckenthaler_Dittschar`
The results are displayed in table 1.

(c)What happens if you add another Dirac kernel that is based on the supertypes from LANL (where available) to the multitask kernel: supertype.csv? (4P)

The best accuracy increases very slightly to 0.6227. The difference to the dirac and peptide kernels is very slim however.

kernel functions	best C	best score
dirac x peptide	0.1	0.622308
uniform x peptide	0.1	0.618846
multitask x peptide	0.1	0.618846
supertype_multitask x peptide	0.1	0.622692

Table 1: Results of the best score and the best parameter c for our SVM models using different kernel combinations

(d) Generate a ROC curve that shows the performances of the different approaches. Calculate AUCs to compare the different approaches and comment on your findings.(2P)

The AUCs range from 0.647 to 0.673. The highest score belongs to the multitask and peptide kernel, the lowest score to the uniform and peptide kernel.

Table 2: Table for all values including AUCs.

kernel functions	best C	best score	AUC
dirac x peptide	0.1	0.622308	0.666932
uniform x peptide	0.1	0.618846	0.647075
multitask x peptide	0.1	0.618846	0.673292
supertype_multitask x peptide	0.1	0.622692	0.661205

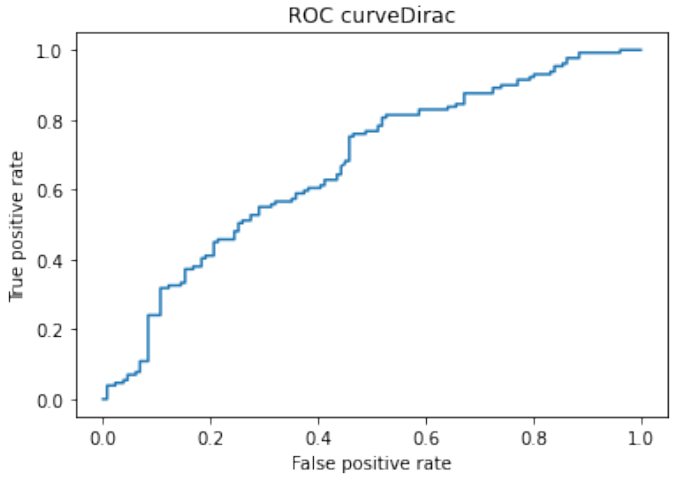

Figure 3: ROC curve for Dirac and peptide kernel.

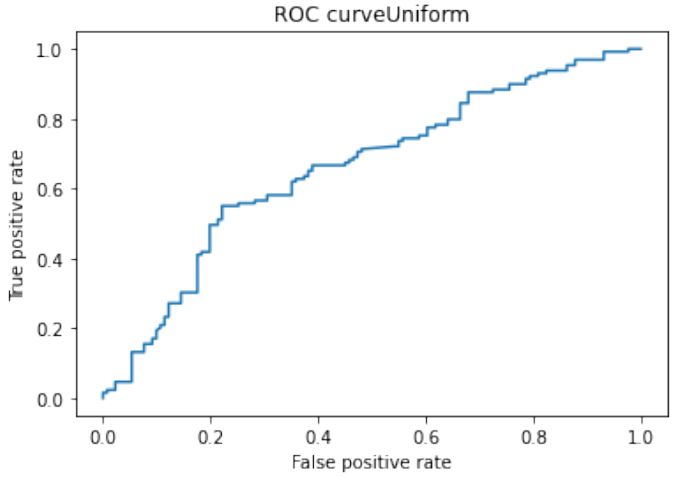

Figure 4: ROC curve for uniform and peptide kernel.

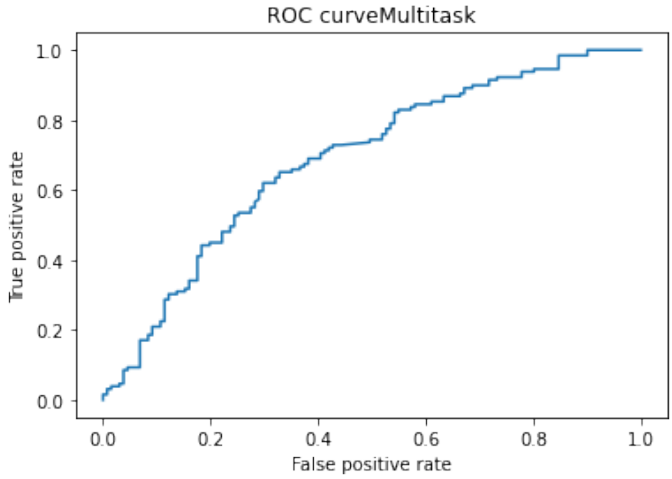

Figure 5: ROC curve for multitask and peptide kernel.

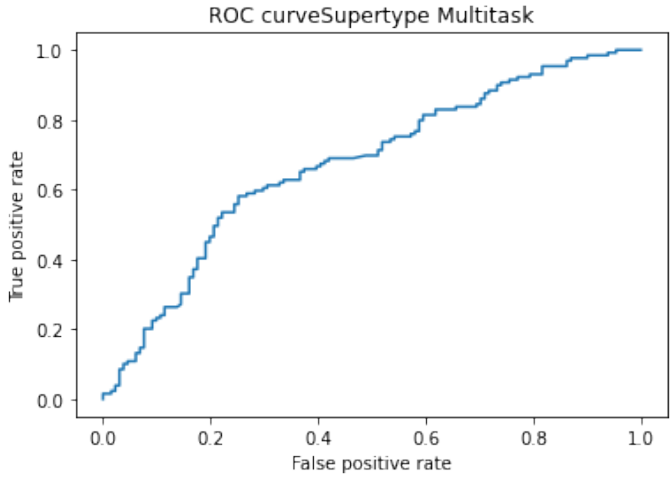

Figure 6: ROC curve for supertype multitask and peptide kernel.

(e) Compare AUCs to accuracy, and discuss possible discrepancies. (2P)

We can see that the AUCs are all generally somewhat higher than the average accuracy scores. This could be because accuracy is determined at the 0.5 threshold only, while the AUCs are determined for all possible thresholds.

References

[1] Bd Editors. Alternative splicing. `https://biologydictionary.net/alternative-splicing/`, 12/03/2018.

[2] Guy Lebanon. Linear algebra: Positive semidefinite matrices. `http://theanalysisofdata.com/probability/C_4.html`, 2022.