Medical Data Science, WS 2021/2022
Prof. Dr. Nico Pfeifer
Chair for Methods in Medical Informatics
University of Tuebingen

2021-11-09

Assignment 2

Deadline: Tuesday, November 23, 7:59 p.m.

This problem set is worth 50 points. You should submit in groups of two people. Submit your solutions digitally by uploading to the ILIAS webpage (none of the other students can see the files you upload). Just upload a zipped folder containing all necessary files and name the folder by your last names. The folder should be named according to the following scheme:

cs[MDS][Assignment 2]_lastname1_lastname2

Problem 1 (T, 15 Points)

Kernel methods, domain adaptation and multitask learning

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

(b) (3P) 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.

(c) (3P) 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?

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

(e) (3P) 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?

(f) (1P) 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*?

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

Problem 2 (T, 8 Points)

Kernel functions use an implicit mapping of the data points to a potentially high-dimensional Hilbert space $\phi : x_i \rightarrow \phi(x_i)$, meaning that $k(x_i, x_j) = \left\langle \phi(x_i), \phi(x_j) \right\rangle$. Show, how we can calculate the squared euclidean distance between two samples in this Hilbert space $||\phi(x_i) - \phi(x_j)||^2$ without using the mapping function ϕ.
Hint: Express $||\phi(x_i) - \phi(x_j)||^2$ as a combination of $k(x_a, x_b)$ where $a, b \in \{i, j\}$. This can be achieved by 'reshaping' the formula according to the given information and mathematical rules.

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 here.

(a) (7P) 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.

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

Problem 4 (P, 18 Points)

In this exercise, you will 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 cross-validation to estimate performances (running over different values of the cost parameter C of the SVM). The data can be found here. The binding class label is in the last column.

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

(b) (7P) 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 $C \in \{10^{-4}, 10^{-3}, ..., 10^{4}\}$. For R, use the **ksvm()** function of the **kernlab** package. For MATLAB you can either use the libsvm package with a custom kernel or the built in SVM functionality from the statistics and machine learning toolbox (using a global variable K as the kernel matrix and custom kernel functions).

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

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

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