Medical Data Science
University of Tuebingen
WS 2023/2024
Karahan Sarıtaş - 6661689
Atahan Özer - 6317973

Assignment II

Problem 1

(a) A symmetric matrix $A \in M_{n\times n}(R)$ is said to be positive semidefinite if $\forall x \in R^n$, $x^T Ax \geq 0$. It is called positive definite when $\forall x \in R^n \setminus \{0\}$, $x^T Ax > 0$. It can be shown that these definitions are equivalent to the following: A symmetric matrix $A \in M_{n\times n}(R)$ is said to be positive semidefinite if all eigenvalues of A, $\lambda_i \geq 0$. Similarly, it is called positive definite when all eigenvalues $\lambda_i > 0$. In the context of complex matrices, the operation of transposition is replaced by the conjugate transpose denoted as $*$.

(b) Both of the approaches are position-aware string kernels. *Weighted degree kernels* compare all subsequences up to a certain length (*k-mers*) at the *exact same position.* The W_D kernel of order d compares two sequences x and x' of equal length l by summing all contributions of k-mer matches of lengths $k \in \{1, ..., d\}$, weighted by coefficients β_k. The ability to identify matching blocks using the W_D kernel is highly sensitive to the sub-sequence's position and cannot accommodate any positional deviation. For example, a slight shift of just one position in a consecutive block within one sequence will cause the W_D kernel to be unable to detect similar blocks, resulting in a reduced similarity score. On the other hand, *weighted degree kernels with shift* can detect signals, which are shifted between two sequences under study. In both of the approaches, matches are weighted by coefficients β_k. Here's a scenario in which both kernels produce identical results: when the sequences are a perfect match to each other:

$$s_1 = AACTG$$
$$s_2 = AACTG$$

In this specific case, the weighted degree kernel fails to account for positional deviation, leading to a distinct result compared to the weighted degree kernel with shift.

$$s_1 = ATCTG$$
$$s_2 = TCTGT$$

(c) For eukaryotic genes that contain *introns* and *exons*, splicing is needed to create an mRNA molecule that will be used to generate a protein. Introns do not carry information to build a protein and they have to be for the mRNA to encode a protein with the right sequence. In *alternative splicing*, one pre-mRNA can be spliced in different ways, depending on which *exons* are kept. When an exon is positioned to

the left with an intron to its right, the splice between them is referred to as a donor splice site. Likewise, when the exon is located to the right and the intron to the left, it is termed an acceptor splice site. In general, humans possess approximately 24,000 protein-coding genes. However, through the mechanism of *alternative splicing*, we are able to generate around 100,000 distinct proteins.

(d) In this context, domain (environment) refers to the underlying distribution from which the samples (train/test) are coming. Having trained on a training set coming from a specific domain (source domain), our machine learning algorithms may be susceptible to generalization issues - where they are not able to achieve high accuracy results when confronted with samples coming from different distributions (target domain) during the deployment. For instance, consider the task of predicting whether an object in an image is a cat or a dog. If our model is exclusively trained on sketch images, it may encounter difficulties in generalizing its predictions to photographs, cartoons, or paintings. One potential approach for addressing the domain adaptation problem is using the training data for training, but using the target data for optimizing the hyperparameters. Another approach would be using a model that consists of the convex combination of classifiers trained on the source and target domain. The convex combination parameter $\alpha \in [0, 1]$ is optimized using the evaluation set of the target domain.

Problem 2

$$\begin{aligned}
\|\phi(x_i) - \phi(x_j)\|_2^2 &= (\phi(x_i) - \phi(x_j))^T(\phi(x_i) - \phi(x_j)) \\
&= (\phi(x_i)^T - \phi(x_j)^T)(\phi(x_i) - \phi(x_j)) \\
&= \phi(x_i)^T\phi(x_i) - \phi(x_i)^T\phi(x_j) - \phi(x_j)^T\phi(x_i) + \phi(x_j)^T\phi(x_j) \\
&= k(x_i, x_i) - k(x_i, x_j) - k(x_j, x_i) + k(x_j, x_j) \\
&= k(x_i, x_i) - 2k(x_i, x_j) + k(x_j, x_j) \quad \text{(kernels are symmetric by definition)}
\end{aligned}$$