Homework 5

Lucas Cosier

Data Mining I

ETH zürich

December 13, 2022

Problem 1

a) • **Solution.** Given $\boldsymbol{x} = (x_1, x_2), \boldsymbol{x'} = (x'_1, x'_2)$

$$\begin{aligned} k(\boldsymbol{x}, \boldsymbol{x'}) &= (\langle \boldsymbol{x}, \boldsymbol{x'} \rangle + 1)^2 \\ &= \langle \boldsymbol{x}, \boldsymbol{x'} \rangle^2 + 2 \langle \boldsymbol{x}, \boldsymbol{x'} \rangle + 1 \\ &= \sum_{i,j}^{2} (x_i x_j)(x'_i x'_j) + \sum_{i}^{2} (\sqrt{2} x_i)(\sqrt{2} x'_i) + 1 \end{aligned} \tag{1}$$

Hence the feature map is $\phi(\boldsymbol{x}) = \begin{bmatrix} x_1 x_1 \\ x_1 x_2 \\ x_2 x_1 \\ x_2 x_2 \\ \sqrt{2} x_1 \\ \sqrt{2} x_2 \end{bmatrix}$ The dimensionality of the feature space is $\begin{pmatrix} 2+2 \\ 2 \end{pmatrix} = 6$

• **Solution** The Gaussian kernel with $\sigma = 1$ can model potentially infinitely dimensional inputs since the dimensionality of the exponent is $\mathbf{R}^n$ for given inputs $\boldsymbol{x}, \boldsymbol{x}' \in \mathbf{R}^n$

b) Through the dual we are able to rewrite the optimization objective for SVMs in terms of the dot product, which enables us to use the kernel trick in order to work in potentially infinite dimensional feature spaces. This avoids the need to model any feature map explicitly, and instead kernelizes the SVM.

Problem 2

a) **Solution.** Let the Gram matrix K for the linear kernel simply be defined in terms of the inner product $K_{ij} = \langle x_i, x'_j \rangle$. For any vector $\boldsymbol{a}$, we need to prove that $a^T K a \geq 0$. By the above, we have that

$$\sum_{i,j} a_i K_{ij} a_j = \sum_{i,j} a_i \langle x_i, x'_j \rangle a_j$$

And by linearity in the first and second terms:

$$\sum_j \left\langle \sum_i a_i x_i, x_j \right\rangle a_j = \left\langle \sum_i a_i x_i, \sum_j a_j x_j \right\rangle$$

which is ≥ 0 for any $\boldsymbol{a} \in \mathbf{R}^n$ due to the properties of the $\mathcal{L}_2-$norm (term is of form $\langle y, y \rangle$)

b) The same argument as above stands if we replace $\boldsymbol{x}$ for $\phi(\boldsymbol{x})$.

c) - Following the above, we now have that $K_{ij} = k_1(x_i, x_j) + k_2(x_i, x_j)$ where we assume k_1 and k_2 are parameterized by feature maps ϕ, ϕ'. This leads to

$$\begin{aligned}\sum_{i,j} a_i K_{ij} a_j &= \sum_{i,j} a_i(\langle\phi(x_i), \phi(x_j)\rangle + \langle\phi'(x_i), \phi'(x_j)\rangle)a_j \\ &= \sum_{i,j} a_i \langle\phi(x_i), \phi(x_j)\rangle a_j + \sum_{i,j} a_i \langle\phi'(x_i), \phi'(x_j)\rangle a_j\end{aligned} \tag{2}$$

  And by point b) we already know that each term is ≥ 0 and hence k_3 is PSD

- For k_4 we have that $K_{ij} = \lambda k_1(x_i, x_j'), \lambda \in \mathbf{R}^+$, therefore

$$\sum_{i,j} a_i K_{ij} a_j = \sum_{i,j} a_i \lambda \langle x_i, x_j'\rangle a_j = \sum_{i,j} a_i \left\langle \sqrt{\lambda}\phi_1(x_i), \sqrt{\lambda}\phi_1(x_j')\right\rangle a_j \tag{3}$$

  which again is PSD by the above.

Problem 3

a) **Solution.** Identifying polynomial kernel with $p = 4, c = 0$ as k_1, constant kernel as $k_2 = 3$, the linear kernel as k_3, and Gaussian kernel as k_4 yields

$$k(\boldsymbol{x}, \boldsymbol{x}') = 6k_1(\boldsymbol{x}, \boldsymbol{x}') + k_2 + k_3(\boldsymbol{x}, \boldsymbol{x}') + k_4(\boldsymbol{x}, \boldsymbol{x}')$$

Then, by the additivity and scalar multiplicity closure properties of the kernel, the above function is a valid kernel.

b) **Solution.**

$$k_{base}(s, s') = \begin{cases} 0 & \text{if } s_0 \neq G \vee s_0' \neq G \\ \sum_{i=0}^{2} k(s_i, s_i') & \text{otherwise} \end{cases}$$

where $k(s_i, s_i')$ is the dirac kernel operating on the individual characters of each of the 3-mers substrings $s \in S, s' \in S'$. The substructures S, S' are sets of 3-mers substrings composing the amino acid sequences.

c) **Solution.**

$$\begin{aligned}X_1 =' GPAGFAGPPGDA' \rightarrow S_1 = \{&'GPA','GFA','GPP','GDA','PAG', \\ &'FAG','PPG','AGF','AGP','PGD'\}\end{aligned}$$

$$\begin{aligned}X_2 =' PRGDQGPVGRTG' \rightarrow S_2 = \{&'PRG','DQG','PVG','RTG','RGD', \\ &'QGP','VGR','GDQ','GPV','GRT'\}\end{aligned}$$

$$X_3 =' GFPNFDVSVSDM' \rightarrow S_3 = \{'GFP','NFD','VSV','SDM','FPN', 'FDV','SVS','PNF','DVS','VSD'\}$$

$k(X_1, X_2) = 15$
$k(X_1, X_3) = 6$

d) **Solution.** Measures for unequal length sequences need to account for resulting unequal substructure set cardinalities. In this case, a normalization procedure should be added.

The problem stems from considering sequences of size 3. If the length of the amino acid considered is not divisible by 3, then some information will be lost (for example if a 'G' is found on position $N - 2$ where N is the total length of the input X) by not considering the last pair. One solution would be to add as many wildcard characters as needed to form a k kmer sequence