Machine learning Exam 1 WS 22/23

The exam was written down from memory after taking it. The tasks might have been recalled incorrectly. Dies ist ein Gedächtsnisprotokoll - use on your own risk.

Ex. 1

Multiple choice, pretty much the answers as in other old exams:

(a) Which statement is true: The bayes error is:
... lowest possible error over all models

(b) Which statement is false: The fisher linear discriminant
... can create non linear deciscion boundary

(c) Which statement is true: a biased estimator.
... ?

(d) Which statement is true: K-means algorithm:
... is a non convex algorithm...

Ex. 2

Max likelyhood function, bayes estimator. Function $P(x|\theta) = \theta(1-\theta)^{x-1}$

(a) give the likelihood function $P(D|\theta)$

(b) give the maximum likelihood solution θ for the dataset $D = \{1, 5, 6\}$

(c) We now adopt a bayesian view.

$$p(\theta) = \begin{cases} 1 & 0 < \theta < 1 \\ 0 & \text{otherwise} \end{cases}$$

Compute the posterior $p(\theta|D)$ after a single draw $D = \{2\}$ with *hint:* $\int_0^1 \theta(1-\theta)^A d\theta = \frac{1}{(A+1)(A+2)}$

(d) Evaluate with this posterior the probability of $x > 1$, i.e. $\int P(x > 1|\theta)p(\theta|D)d\theta$

Ex. 3

Kernels. A kernel is positive semidefinite kernel if

$$\sum_i \sum_j c_i c_j k(x_i, x_j) \geq 0$$

a positive semidefinite kernel has

$$\Phi(x) : k(x, x') =< \Phi(x), \Phi(x') >$$

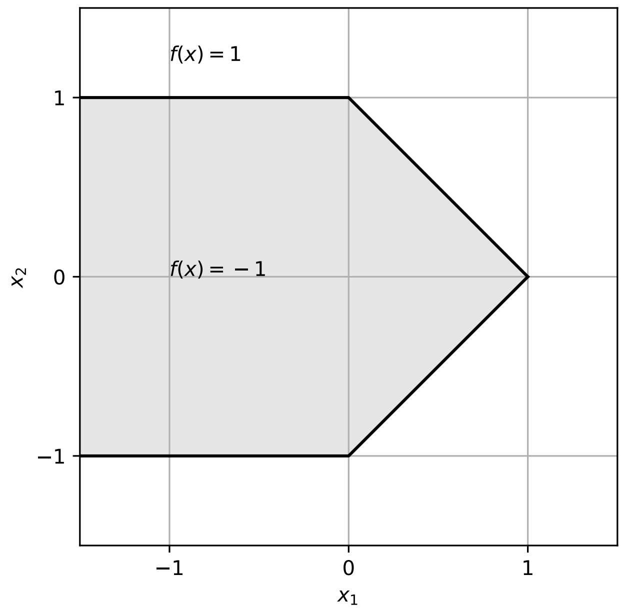
(a) $k(x, x')$ is a kernel. Show that $k_z(x, x') = k(x, x') - k(x, z) - k(z, x') + k(z, z)$ is also a kernel.

(b) We now have $z, x, b \in \mathcal{R}^d, W \in \mathcal{R}^{d \times d}?$. $k(x, x') =< Wx + b, Wx' + b >$. Show that

$$\Phi_z : x- > W(x - z)$$

induces k_z [from the task above]

Ex. 4



(a) [Draw neural network with activation function $a_j = sign(\sum w_{ij}a_i + b_j)$ which outputs matches drawn function.]

(b) Give the activations for input $x = (-2, 2)$

Ex. 5

very loosely
Programming on paper, for ridge regression with function $f(x)$ [something resembling $K(K - \lambda I)^{-1}y$] provided, documentation for np.linalg.inv and scipy.distance.cdist provided, write python code for

(a) Compute some vectorized kernel $k(X_A, X_B) = \frac{1}{0.1 - ||X_A - X_B||^2}$ where X_A, X_B are matrices with one datapoint per row.

(b) Write some function that trains on $X_{\text{train}}, Y_{\text{train}}$ and gives output on X_{test}. Use the kernel function you wrote above.

(c) Using the function written above, write a function from that trains on $Xtrain, Ytrain$ and outputs the mean squared error of the training set.