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 Fisher LDA → only linear decision Boundary
... can create non linear deciscion boundary

(c) Which statement is true: a biased estimator. to reduce estimation error for high-dim data
... ?

(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. 2

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

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

$$P(D|\theta) = \prod_i^N P(x_i|\theta) = \prod_i^N \theta(1-\theta)^{x_i-1}$$

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

$$P(D|\theta) = \theta^3(1-\theta)^{4+5} = \theta^3(1-\theta)^9$$

$$\max_\theta P(D|\theta) = \max_\theta \log P(D|\theta)$$

$$= \max_\theta \log \theta^3(1-\theta)^9$$

$$= \max_\theta 3\log\theta + 9\log(1-\theta)$$

$$\frac{\partial}{\partial\theta}\log P(D|\theta) = \frac{3}{\theta} - \frac{9}{1-\theta} = 0 \Rightarrow 3 - 3\theta - 9\theta = 3 - 12\theta = 0$$

$$\hat{\theta} = \frac{1}{4}$$

(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:* $\int_0^1 \theta(1-\theta)^A d\theta = \frac{1}{(A+1)(A+2)}$

$$P(\theta|D) = \frac{P(D|\theta)P(\theta)}{\int P(D|\theta)P(\theta)d\theta} = \frac{\theta(1-\theta)\, 1_{\{0<\theta<1\}}}{\int_0^1 \theta(1-\theta)\, d\theta}$$

$$= 6\theta(1-\theta)\, 1_{\{0<\theta<1\}}$$

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

$$P(x>1|D) = 1 - P(x=1|D)$$

$$= 1 - \int P(x=1|\theta)P(\theta|D)\, d\theta$$

$$= 1 - \int_0^1 \theta\, 6\theta(1-\theta)\, d\theta$$

$$= 1 - \int_0^1 6\theta^2(1-\theta)\, d\theta$$

$$= 1 - \int_0^1 6(\theta^2 - \theta^3)\, d\theta$$

$$= 1 - 6\cdot\left(\frac{1}{3}\theta^3 - \frac{1}{4}\theta^4\right)\Big|_0^1 = 1 - 6\cdot\frac{1}{12} = 0{,}5$$

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.

$$\sum_i \sum_j c_i c_j \left(\Phi(x_i)^T \Phi(x_j) - \Phi(x_i)^T \Phi(z) - \Phi^T(z) \Phi(x_j) + \Phi(z)^T \Phi(z) \right)$$

$$= \sum_i \sum_j c_i c_j \left((\Phi(x_i) - \Phi(z))^T (\Phi(x_j) - \Phi(z)) \right)$$

$$= \sum_i \sum_j c_i c_j \sum_m \left(\Phi_m(x_i) - \Phi_m(z) \right) \left(\Phi_m(x_j) - \Phi_m(z) \right)$$

$$= \sum_m \sum_i c_i \left(\Phi_m(x_i) - \Phi_m(z) \right) \sum_j c_j \left(\Phi_m(x_j) - \Phi_m(z) \right)$$

$$= \sum_m \left(\sum_i c_i (\Phi_m(x_i) - \Phi_m(z) \right)^2 \geq 0$$

(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]

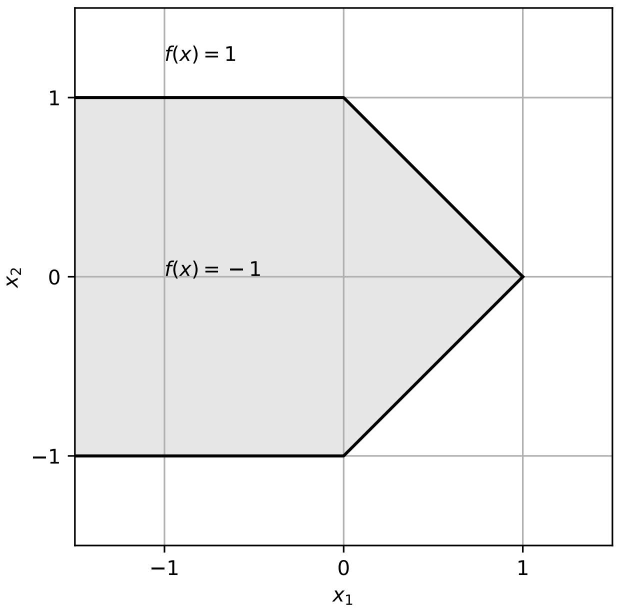
$$k_z(x, x') = \langle Wx + b, Wx' + b \rangle - \langle Wx + b, Wz + b \rangle - \langle Wz + b, Wx' + b \rangle + \langle Wz + b, Wz + b \rangle$$

$$= \langle Wx + b, W(x' - z) \rangle - \langle Wz + b, W(x' - z) \rangle$$

$$= \langle \underbrace{W(x - z)}_{\Phi_z}, W(x' - z) \rangle$$

$$= \langle \Phi_z(x), \Phi_z(x') \rangle$$

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

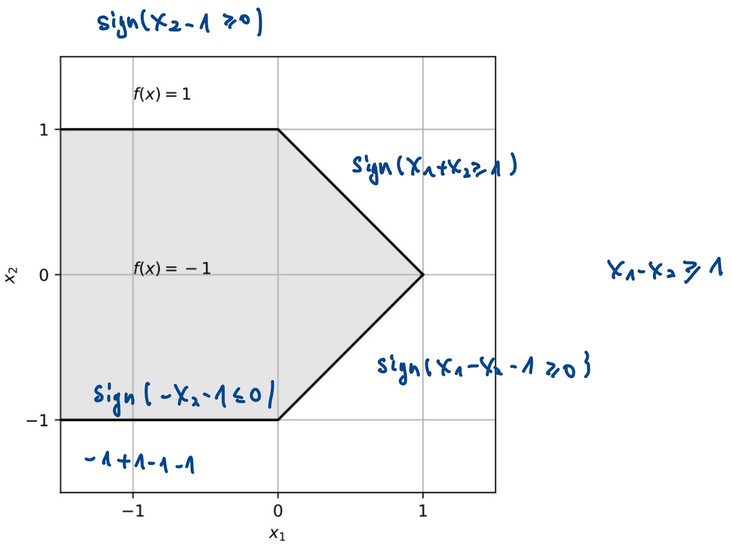
very loosely
Programming on paper, for ridge regression with function $f(x)$ [something resembling $K(K - \lambda I)^{-1} y$] provided, documentation for np.linalg.inv and scipy.distance.cdist provided, write python code for

(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.

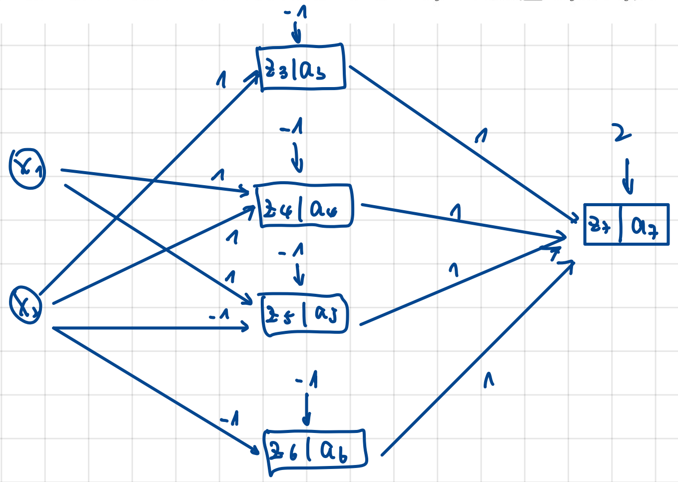
(b) Write some function that trains on $X_{\text{train}}, Y_{\text{train}}$ and gives output on X_{test}. Use the kernel function you wrote above.

(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.

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 $x = (-2, 2)$

$z_7 = a_3 + a_4 + a_5 + a_6 = 1 + 1 - 1 - 1 = 0$

$a_7 = sign(0 + 2) = 1$

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.

```
kernel ( X_A , X_B ):
    sq_dist = cdist ( X_A , X_B , 'sqeuclidean')

    K = 1 / ( 0.1 - sq_dist )

    return K
```

(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.

$\alpha = (K + \lambda I)^{-1} y$

```
train (X_train , Y_train , lam ):

    K = kernel (X_train, Y_train)
    I = np.eye(K.shape[0])
    alpha = np.linalg.inv( K + lam * I ) @ Y_train
    return alpha
```

(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.