EXAM 11.02.20

Question 1

1). In classification we have a target function $f: X \to Y$ with $X \subseteq \mathbb{R}^d$ and $Y = \{y_1 \dots y_k\}$

In regression we have a target function $f: X \to Y$ with $X \subseteq \mathbb{R}^d$ and $Y = \mathbb{R}$.

2). In linear regression we have a model $y(x,w) = \sum_{j=1}^{M} w_j \phi_j(x) = w^T \phi(x)$. The target value is $t = y(x,w) + \varepsilon$ with ε additive gaussian noise. Since the noise is gaussian

$p(\varepsilon|\beta) = N(\varepsilon|0, \beta^{-1}) \Rightarrow p(t|x_1 \dots x_N, w, \beta) = N(t|y(x,w), \beta^{-1})$.

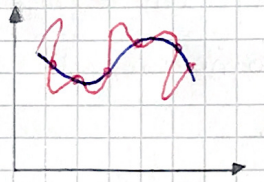
If we use the iid hypothesis: $p(\{t_1 \dots t_n\}|x_1 \dots x_N, w, \beta) = \prod_{n=1}^{N} N(t_n|w^T\phi(x_n), \beta^{-1}) =$

$$= \sum_{n=1}^{N} \ln N(t_n|w^T\phi(x_n), \beta^{-1}) = -\beta \cdot \frac{1}{2} \sum_{n=1}^{N} (t_n - w^T\phi(x_n))^2 - \frac{N}{2} \ln(2\pi\beta^{-1})$$

We want to find $w^* = \operatorname{argmin}_w E_D(w)$ with $E_D(w) = \frac{1}{2} \sum_{n=1}^{N} (t_n - w^T\phi(x_n))^2$

We can find w^* with least squares $\Rightarrow E_D(w) = \frac{1}{2}(t - \phi w)^T(t - \phi w) \Rightarrow w^* = \phi^\dagger t$.

3).



It's possible to see that this solution is overfitted.

We can avoid overfitting applying regularization.

$\operatorname{argmin}_w E_D(w) + \lambda E_W(w)$ with $\lambda > 0$ regularization factor

and $E_W(w) = \frac{1}{2} w^T w$.

Question 2

1). Logistic regression is a probabilistic discriminative model based on maximum likelihood.

$p(t|\tilde{w}) = \prod_{n=1}^{N} y_n^{t_n} (1 - y_n)^{(1-t_n)}$. We want to minimize the error function (cross-entropy):

$E(\tilde{w}) = -\sum_{n=1}^{N} t_n \log(y_n) + (1 - t_n) \log(1 - y_n)$. We can apply Newton-Raphson to find

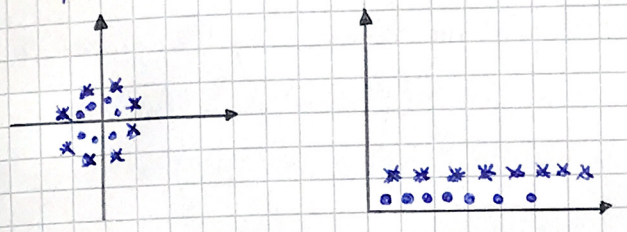
$\tilde{w}^* = \operatorname{argmin}_{\tilde{w}} E_D(\tilde{w}) \Rightarrow \tilde{w} \leftarrow \tilde{w} + H(\tilde{w})^{-1} \nabla E(\tilde{w})$ with $\nabla E(\tilde{w}) = \sum_{n=1}^{N} (y_n - t_n)\tilde{x}_n$

2). The second solution fits the data better. We have to do the product $w_i x_i$ and then we apply the sigmoid function. If $\sigma(w_i x_i) = t_i$ the prediction is correct otherwise is wrong. It's possible to see that in the first case we have 0 correct predictions, in the second case 3.

Question 3

1). If the input vector appears only in the form of an inner product $x^T x'$, we can replace the inner product with a kernel function $k(x, x')$. For example if we have $\phi^T(x)\phi(x)$ we can replace this product with $k(\phi(x), \phi'(x))$.

2).



I use a polynomial kernel function: $k(x, x') = (\beta x^T x' + \gamma)^d$, selecting for example $d = 3$.

Question 5

1). In supervised learning we have a target function $f: X \to Y$ and a dataset $D = \{(x_i, y_i)\}_{i=1}^{n}$ and we want to find an approximated function $\hat{f}$ that returns values similar to f, specially for samples not in the dataset.

In reinforcement learning we have a dataset $D = \{x_1, a_1, r_1, \dots x_n, a_n, r_n\}$ and we have an agent that acts in an environment, the goal of RL is to find a behaviour function $\pi: X \to A$.

2). In MDPs we have the property of full observability, states are fully observable. If we have non-deterministic actions we cannot predict the outcome of one action but we can observe the resulting state after the execution of this action.

Question 6

1). K-Means takes as input the dataset $\{x_n\}$ and the number of clusters K and returns as output $\mu_1 \dots \mu_K$. The algorithm works in the following way:

① We select the value of K = # clusters.

② We have to split our samples in clusters. We can do this randomly or sistematically as follow:

- We take the first K samples and we assign each sample to a single element cluster;
- We take the remaining $N-K$ samples and we assign each sample to the cluster with the nearest centroid.

③ We select each sample in sequence and we compute the distance from all the clusters. If the sample is not correctly classified we move it to the correct cluster and we recompute the centroids of the modified clusters. We repeat this step until convergence