



# 336546 - Machine Learning in Healthcare

## Homework number 3

Omri Magen – 302301718

Dates: 13/1/2021-15/1/2021

## 1) Clustering (10%):

a.

K-medoids is more robust to noise and outliers than k-means because k-medoids minimizes the sum of pairwise dissimilarities instead of a sum of squared Euclidean distances.



Where the "*medoid*" of a cluster is defined as the object in the cluster whose average dissimilarity to all the objects in the cluster is minimal, that is, it is a most centrally located point in the cluster. And the "*squared Euclidean distances*" is just the sum of squares:  $d^2(x, y) = \|x - y\|^2 = \sum_{i=1}^d (x_i - y_i)^2$ .

b.

To find the minimum we will take the derivative:

$$\frac{d}{d\mu} \sum_{i=1}^m (x_i - \mu)^2 = 0$$



After taking the derivative we get

$$-2 \sum_{i=1}^m x_i + 2 \cdot m \cdot \mu = 0$$

$$m \cdot \mu = \sum_{i=1}^m x_i$$

And finally:

$$\mu_{min} = \bar{x} \blacksquare$$

c. (bonus)

Using the assumption that  $\mu \in [x_1, x_m]$  and defining  $f(\mu)$  to be:

$$f(\mu) = \sum_{i=1}^m |x_i - \mu|$$

If we look at the equation above, we notice that:

$$\sum_{i=1}^m |x_i - \mu| = \sum_{i=2}^{m-1} |x_i - \mu| - (x_1 - \mu) + (x_m - \mu) = \sum_{i=2}^{m-1} |x_i - \mu| + (x_m - x_1)$$

First of all, let us assume  $m$  is odd, we can apply the above identity repeatedly:

$$f(\mu) = \sum_{i=1}^m |x_i - \mu| = \left| \left( x_{\frac{m+1}{2}} - \mu \right) \right| + (x_m - x_1) + (x_{m-1} - x_2) + \dots + \left( x_{\frac{m+3}{2}} - x_{\frac{m+1}{2}} \right)$$

And then we see that  $f(\mu)$  is:

$$f(\mu) = \left| \left( x_{\frac{m+1}{2}} - \mu \right) \right| + \text{constant}$$

This is just the absolute value function with its vertex being at  $(x_{\frac{m+1}{2}}, \text{constant})$ , the minimum of the absolute value function occurs at its vertex, therefore  $x_{\frac{m+1}{2}}$  (which is the median) minimizes  $f(\mu)$ .

Now suppose  $m$  is even, again by using our identity, we get:

$$f(\mu) = \sum_{i=1}^m |x_i - \mu| = \left| \left( x_{\frac{m}{2}} - \mu \right) \right| + \left| \left( x_{\frac{m+2}{2}} - \mu \right) \right| + \text{constant}$$

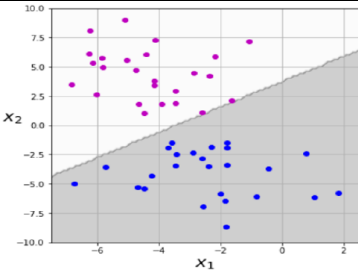

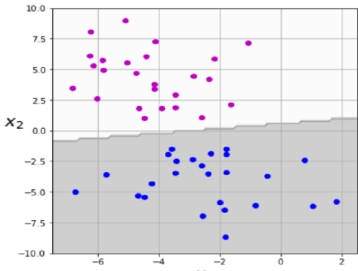

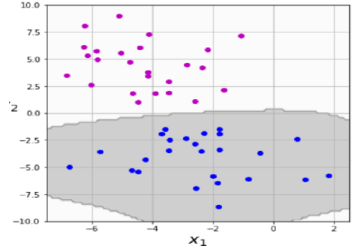
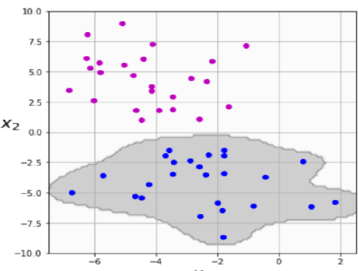
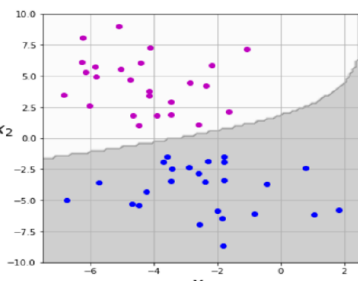
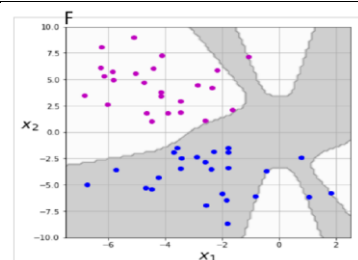
Where the minimum occurs at  $f'(\mu) = 0$  (or when not defined), therefore by differentiating and setting  $f'(\mu)$  to zero we get:

$$\frac{\left| \left( x_{\frac{m}{2}} - \mu \right) \right|}{\left( x_{\frac{m}{2}} - \mu \right)} + \frac{\left| \left( x_{\frac{m+2}{2}} - \mu \right) \right|}{\left( x_{\frac{m+2}{2}} - \mu \right)} = 0$$

The median satisfies the above equation, since  $x$  is halfway between  $x_{\frac{m}{2}}$  and  $x_{\frac{m+2}{2}}$


Therefore,  $\mu$  is a median in all cases. ■

**2) SVM (30%):**

	Figure	Match	Explanation
A		 Linear kernel with $C = 1$	We can clearly see that A and D are linear. Now we need to see which C ( $C = 1/\lambda$ ) matches each figure. For large values of C, the hyperplane margins are smaller because the C parameter tells the SVM optimization how much you want to avoid misclassifying each training example and vice versa.
D		 Linear kernel with $C = 0.01$	Thus: (A-2) (D-1)
E		5 RBF kernel with $\gamma = 0.2$	Here we see that E and B are Radial Basis Functions (RBF). Now we need to match the right Y to each figure. The $\gamma$ parameter represents the inverse of the radius of influence of samples selected by the model as support vectors, meaning that the larger the $\gamma$ parameter is the more we fit our model and vice versa.
B		6 RBF kernel with $\gamma = 1$	Thus: (E-5) (B-6)
C		3 2 <sup>nd</sup> order polynomial kernel	We are left with the 2 polynomial kernels. C is a 2 <sup>nd</sup> order polynomial kernel because of the parabolic shape of the graph and F is clearly over fitting the data with a 10 <sup>th</sup> order polynomial kernel.
F		4 10 <sup>th</sup> order polynomial kernel	Thus: (C-3) (F-4)


### 3 Capability of generalization (20%)

a.

The scientific term that describes the balance that Einstein meant in a machine learning aspect in my opinion deals with both the risk of overfitting and the risk of underfitting. Meaning that our model should be as simple as possible while both fitting our data and avoid overfitting/underfitting (variance  as tradeoff). Therefore, once we reached a model that satisfies our needs, we should not try to complicate it resulting in overfitting.

b.

First of all, after looking at the AIC model  $AIC = 2p - 2\ln(\hat{L})$ , given a set of candidate models for the data, the preferred model is the one with the minimum AIC value. Thus, each of the terms ( $2p$  and  $2\ln(\hat{L})$ : where  $P$  is the number of estimated parameters and  $\hat{L}$  is the maximum value of the likelihood function) in AIC affects the balance by reducing and increasing the AIC value.

AIC rewards goodness of fit as assessed by the likelihood function ( $2\ln(\hat{L})$ ) because the larger it gets the lower the AIC value is. The AIC also includes a penalty that is an increasing function of the number of estimated parameters. The penalty discourages overfitting which is desired because increasing the number of  parameters in the model almost always improves the goodness of the fit. Thus there is a balance between the two terms.

c.

As mentioned above, if the balance is violated the two options that are likely to happen due to a large difference in the orders of magnitude of either one of the terms are *overfitting* and *underfitting*. We could get *underfitting* if the penalty function ( $P$ ) is reduced or *overfitting* if the likelihood function is too high.

d.

We are aiming for a low value (minimum) for the AIC that will indicate a better fit. To get the minimum AIC value we need to find the balance between the smallest number of parameter (thus reducing the  $P$  term) and achieving the highest log likelihood (thus increasing the  $\hat{L}$ ). The AIC model allows us to find the optimization for both terms by aiming for the lowest AIC value.