

Course name: Machine Learning in Healthcare
HW: HW3

Machine Learning in Healthcare
Winter 2020-2021

HW3 -Theoretical Part

Mor Ventura: 313177412

Date: 15.01.2021

Course name: Machine Learning in Healthcare

HW: HW3

MLH- HW3**1. Clustering**

- a. K-medoid is more robust to noise or outliers than K-means algorithm. In general, median is less sensitive to outliers because the “punishment” is only the absolute of the error (l1 norm, distance) and not squared as in mean (l2 norm).

- b. Proof:

$$\begin{aligned}
 \sum_i^m (x_i - c)^2 &\stackrel{\pm \bar{x}}{=} \sum_i^m (x_i - \bar{x} + \bar{x} - c)^2 \\
 &= \sum_i^m (x_i - \bar{x})^2 + 2 \cdot \sum_i^m (x_i - \bar{x}) \cdot (\bar{x} - c) + \sum_i^m (\bar{x} - c)^2 \\
 &= \sum_i^m (x_i - \bar{x})^2 + 2 \cdot (\bar{x} - c) \cdot \underbrace{\sum_i^m (x_i - \bar{x})}_{** \text{average} = 0} + m \cdot (\bar{x} - c)^2 \\
 &\rightarrow \sum_i^m (x_i - \bar{x})^2 + m \cdot (\bar{x} - c)^2 \stackrel{\text{minimum}}{\Rightarrow} \sum_i^m (x_i - \bar{x})^2
 \end{aligned}$$

We will get minimum when $m \cdot (\bar{x} - c)^2 = 0$, ie $c_{min} = \bar{x}$ ■

** proof of $\sum_i^m (x_i - \bar{x}) = 0$:

$$\begin{aligned}
 \sum_i^m (x_i - \bar{x}) &= \sum_i^m x_i - \sum_i^m \bar{x} = \sum_i^m x_i - m \cdot \bar{x} = \sum_i^m x_i - m \cdot \frac{1}{m} \sum_i^m x_i \\
 &= \sum_i^m x_i - \sum_i^m x_i = 0 \quad \blacksquare
 \end{aligned}$$

- c. Bonus:

$$L_1 = \sum_i^m |x_i - c|$$

$$\frac{\partial L_1}{\partial c} = \sum_i^m -\text{sign}(x_i - c)$$

Course name: Machine Learning in Healthcare

HW: HW3

$$\begin{cases} x_i < c \rightarrow -sign = -(-1) = +1 \\ x_i > c \rightarrow -sign = -(+1) = -1 \\ x_i = c \rightarrow -sign = 0 \end{cases}$$

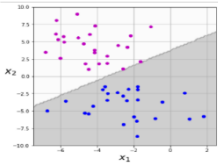
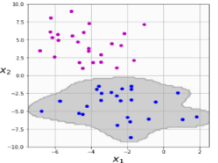
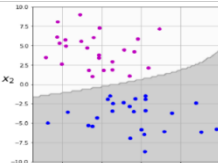
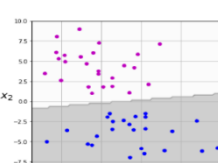
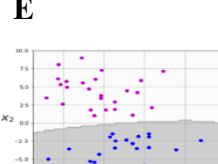
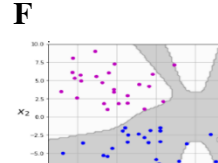
The minimum: When half of the measurements are bigger than c and half are smaller than $c \rightarrow$ the result is 0.

It is exactly the meaning of median. ■

Course name: Machine Learning in Healthcare

HW: HW3

2. SVM

Visualization #	Settings #	Explanation
A 	2 – Linear, $C=1$	<ul style="list-style-type: none"> Linearly separable, $C = \frac{1}{\lambda}$ Large $C \rightarrow$ small $\lambda \rightarrow \downarrow$ Regularization $\rightarrow \uparrow$ Fidelity \rightarrow small margins
B 	6 – RBF, $\gamma = 1$	<ul style="list-style-type: none"> Radially separable Large $\gamma \rightarrow \rightarrow \downarrow$ Regularization $\rightarrow \uparrow$ Fidelity
C 	3 – Polynomial, 2nd	<ul style="list-style-type: none"> Non-linearly separable \sim parabolic shape
D 	1 – Linear, $C=0.01$	<ul style="list-style-type: none"> Linearly separable, $C = \frac{1}{\lambda}$ small $C \rightarrow$ large $\lambda \rightarrow \uparrow$ Regularization $\rightarrow \downarrow$ Fidelity \rightarrow large margins
E 	5 – RBF, $\gamma = 0.2$	<ul style="list-style-type: none"> Radially separable Small $\gamma \rightarrow \rightarrow \uparrow$ Regularization $\rightarrow \downarrow$ Fidelity
F 	4 – Polynomial, 10nd	<ul style="list-style-type: none"> Non-linearly separable overfitting

Course name: Machine Learning in Healthcare

HW: HW3

Additional explanation:

First, I classified the separable hyperplanes to the kernels that were offered in the settings:

$\{A, D \in \text{Linear kernel}, C, F \in \text{Polynomial kernel}, B, E \in \text{RBF kernel}\}$.

Afterwards, I matched each pair by the hyper-parameters.

- $C = \frac{1}{\lambda}$, controls the trade-off between increasing the distance between the hyperplane and the support vectors, and decreasing the number of samples which are misclassified by this hyperplane.
- γ is the inverse of the std ($= \sqrt{\text{variance}}$) of the RBF kernel, so the higher the variance (for example, wider gaussian), the lower the γ .

Course name: Machine Learning in Healthcare

HW: HW3

3. Capability of generalization

- a. The scientific term of the balance that Einstein meant in a machine learning aspect, is the “**over-fitting**” principle or the “**variance-bias trade-off**”. As Albert Einstein stated, “Everything should be made as simple as possible but not simpler”, which also applies to ML: the learning phase on the training dataset should create the simplest estimator, that has an accurate balance between the resemblance to the measurements and the generalization of the learning so it will give good results when tested. Put simply, this means not overfitting or underfitting the data (too simple model).
- b. The way each of the terms in AIC affect the terms of balance: the model with the minimal AIC value, is the preferred one.

- $2p - 2$ (the total number of learned parameters): this parameter penalizes on increasing number of parameters that the model uses; it actually means that it discourages overfitting (high variance).
- $2 \ln(\hat{L})$ – minus $2 \ln$ (the estimated likelihood): this parameter penalizes on low likelihood value - underfitting or encourages high likelihood value, i.e., it encourages a good fit between the model and the data.

By this combination of those parameters, AIC metric mathematically represents exactly what we defined in the last section – it demands having the simple model that achieves the “sweet-spot”, the balance, of the variance-bias trade-off.

- c. The 2 possible options for violated balance:
 - **Overfitting** – too close or exact corresponding between the model and the training data, so it may probably fail to fit additional data or predict future observations correctly (test dataset) – high p.
 - **Underfitting** – lack of complexity of the model so it does not fit correctly, characterized by a high cost-function value - low $\ln(\hat{L})$.
- d. We aim for a low value of the AIC metric. Meaning a good model is the one that has a minimal AIC among all other models. Logically, it makes sense because we want to find a model that has a similarity to the measurements’ distribution (likelihood) and does not use too many parameters.