HW3 - Machine Learning in Healthcare – 336546

Clustering

a. K-medoid is more robust to noise or outliers than K-means.

K-medoid and K-means are algorithms for clustering. **K-medoid** minimizes the dissimilarities between two data examples, while finding the best data examples to represent K clusters - those points are called medoids.
On the other hand, **K-means** minimizes the squared Euclidian distance between the examples and a random point, called centroid. The algorithm finds the best centroids to represent the center for each cluster.

K-medoid is less sensitive to noise or outliers than K-means because the medoid is a point from the data, as opposed to the centroid that is a calculated point and not necessarily a point in the data set. As they both try to represent the center of the cluster in different ways, we can compare it to the differences between median and mean: For example, if we calculate the mean and median of the array [1, 2, 3, 4, 100000] we will get that the mean is 20002 and the median is 3. We can see that the median metric is less sensitive to the outlier.

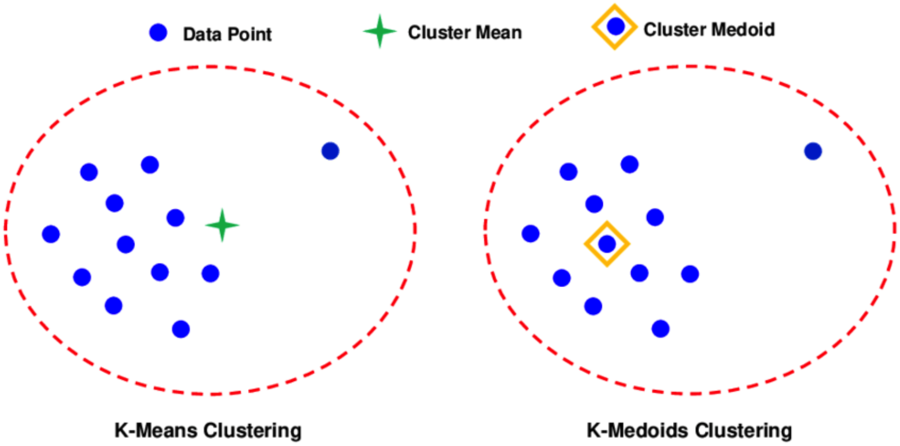


Figure 1 - Graphical representation of the difference between the k-means and k-medoids clustering methods

b. For 1D case the centroid (μ) which minimizes the term $\sum_{i=1}^{m}(x_i-\mu)^2$ is the mean of m examples. **Proof:**

$$Definition:\ f(x)=\sum_{i=1}^{m}(x_i-\mu)^2$$

$$f'(x)=\sum_{i=1}^{m}2(x_i-\mu)$$

$$To\ achive\ minimum\ we\ compare\ f'(x)\ to\ zero:\ \sum_{i=1}^{m}\cancel{2}(x_i-\mu)=0$$

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

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

$$\rightarrow \mu=\frac{1}{m}\sum_{i=1}^{m}x_i=mean\ of\ m\ examples$$

c. For 1D case the medoid (μ) which minimizes the term $\sum_{i=1}^{m}|x_i-\mu|$ is the median of m examples given that μ belongs to the dataset. **Proof:**

$$Definition:\ g(x)=\sum_{i=1}^{m}|x_i-\mu|$$

$$g'(x)=\sum_{i=1}^{m}sign(x_i-\mu)$$

$$To\ achive\ minimum\ we\ compare\ g'(x)\ to\ zero:\ \sum_{i=1}^{m}sign(x_i-\mu)=0$$

$$g'(x)=0\ if\ the\ signs\ of\ the\ difference\ cancle\ each\ other$$

$$\rightarrow Same\ number\ of\ positive\ and\ negative\ samples.$$

$$That\ will\ happen\ if\ \mu\ is\ the\ median\ of\ the\ samples,$$

$$because\ it\ turnes\ the\ x_i<\mu\ to\ negative\ values,$$

$$keeps\ the\ x_i>\mu\ positive\ and\ zeros\ x_i=\mu.$$

$$\rightarrow When\ m\ is\ an\ uneven\ number,\mu=x_{\frac{m+1}{2}}.$$

$$When\ m\ is\ even, the\ median\ supposed\ to\ be\ \frac{x_{\frac{m}{2}}+x_{\frac{m+1}{2}}}{2}, but\ it\ can't\ be\ the\ medoid\ because$$

$$it\ is\ not\ one\ of\ the\ points\ in\ the\ data, so\ we\ choose\ the\ \mu\ to\ be\ x_{\frac{m}{2}}\ or\ x_{\frac{m+1}{2}}$$

$$(the\ one\ that\ gives\ better\ results).$$

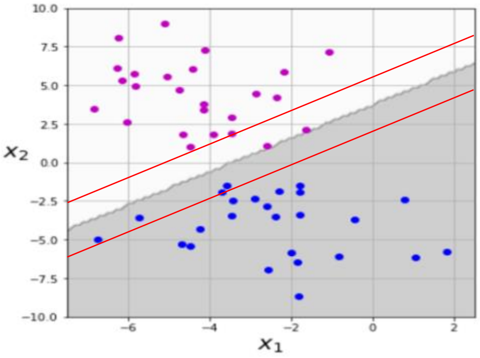
SVM

In SVM we can choose different kernels, they affect the way our data is separated (exp. for linear kernel the data separated by a straight line). Also, we can choose different C (or λ) which is hyperparameter that affects the width of the margins, and by that controls the trade-off between misclassification and overfitting (exp. For large value of C the margin is small, so we get a small number of misclassified samples, but we risk overfitting).

In Addition, for RBF kernel we can choose different γ, which is an hyperparameter that defines how far (close/far from the hyperplane) the influence of a single training example reaches, that way it controls the trade-off between better separation and overfitting (exp. For high values of γ the influence of the closer points is greater, so the separation will be better than with lower γ but we will risk overfitting).

Matching the images to the settings:

A

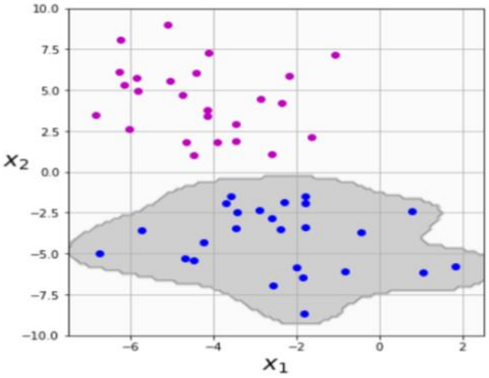


1 – Linear kernel with C=0.01

Linear kernel: Straight line separates the data.

C=1: The margin is large (in red), and we have a risk for misclassifications (there are samples in the margins, very close to the hyperplane), which means that the C is small.

B



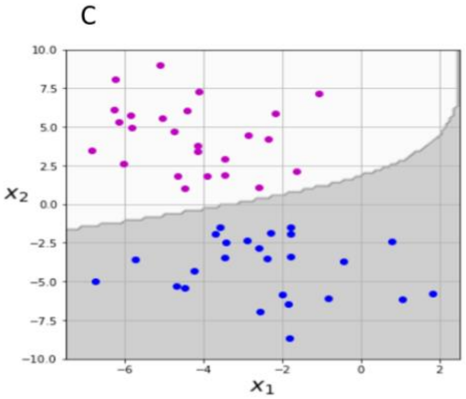
6 – RBF kernel with $\gamma = 1$

RBF kernel: Non-linear (spherical) separation.

$\gamma = 1$: The fitting to the blue data is high (the grey area is circling the blue dots) which means the γ is high.

3 – 2nd order polynomial kernel

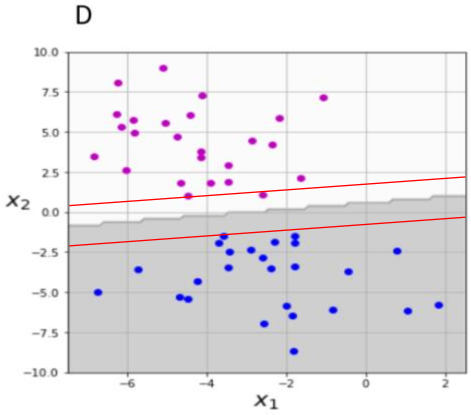
2nd order polynomial kernel: Non-linear separation, the shape of it consists with 2nd order polynomial.



2 – Linear kernel with C=1

Linear kernel: Straight line separates the data.

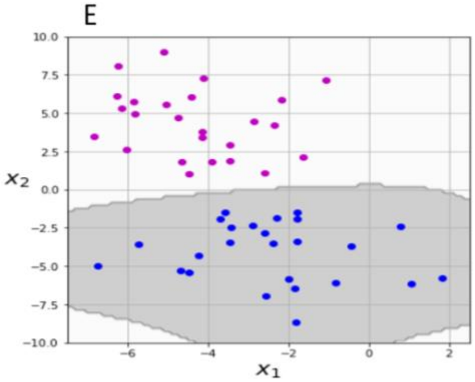
C=1: The margin is smaller than in A (in red), and we have less risk for misclassifications (less samples in the margins), which means that the C is large.



5 – RBF kernel with $\gamma = 0.2$

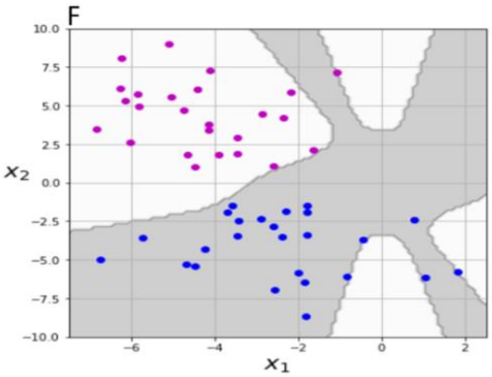
RBF kernel: Non-linear (spherical) separation.

$\boldsymbol{\gamma = 0.2}$**:** The fitting to the blue data is not exact (as opposite to B) which means the γ is low.



4 – 10th order polynomial kernel

10th order polynomial kernel: Non-linear separation, the shape of it consists with high order polynomial.



Capability of generalization

a. Albert Einstein stated that: “Everything should be made as simple as possible but not simpler”. The scientific term of the balance that Einstein meant to in machine learning aspect is generalization. Generalization is a way to estimate the capabilities of a machine learning algorithm, by providing accurate and meaningful predictions on **unseen** data with the concepts learned by the algorithm. If a machine learning model is generalized, it will achieve balance between the capacity to properly represent the training data and providing accurate predictions to testing data.

b. Akaike information criterion (AIC) is composed of the terms $2p, 2\ln(\hat{L})$:

$$AIC = 2p - 2\ln(\hat{L})$$

While p is the total number of learned parameters and $\hat{L}$ is the estimated likelihood given those parameters. The lower the AIC value – the better the model.

The first term ($2p$) represent the variance. Variance is the variability in the model prediction and its adjustment on the given data set, **with low variance we can provide accurate predictions to unseen data.** High variance can lead to noise it the data set, overfitting, and high complexity. The second term ($2\ln(\hat{L})$) represent the bias. Bias is the error between average model prediction and the ground truth, it describes how well the model match the training set, **with low bias we can properly represent the training data**. High bias can lead to high error rate, underfitting and failure to capture data trends.

We can’t have low bias without high variance and vice versa because those parameters are inversely connected. Which means, we need to find balance between the variance and the bias so our model will be generalized.

c. Two options that are likely to happen if this balance was violated are:
Overfitting is when the ML model is too complex and correctly predicts the target data from the input data in the training set but preforms poorly on the testing set. Overfitting caused by high variance and low bias.
Underfitting is when the ML model is not complex enough and unable to correctly predict the target data from the input data on the training and the testing sets. Underfitting caused by low variance and high bias.

d. We are aiming to lower the AIC. By lowering the AIC, we are decreasing the total number of learned parameters while increasing the likelihood. While lowering the AIC, we are searching for balance in the bias-variance trade-off.

References

[1] Entezami, A., Sarmadi, H., & Razavi, B. S. (2020). An innovative hybrid strategy for structural health monitoring by modal flexibility and clustering methods. Journal of Civil Structural Health Monitoring, 10(5), 845-859.

[2] Arora, P., & Varshney, S. (2016). Analysis of k-means and k-medoids algorithm for big data. Procedia Computer Science, 78, 507-512.

[3] Barbiero, P., Squillero, G., & Tonda, A. (2020). Modeling generalization in machine learning: A methodological and computational study. arXiv preprint arXiv:2006.15680.

[4] Hu, S. (2007). Akaike information criterion. Center for Research in Scientific Computation, 93. ISO 690

[5] Yang, Z., Yu, Y., You, C., Steinhardt, J., & Ma, Y. (2020, November). Rethinking bias-variance trade-off for generalization of neural networks. In International Conference on Machine Learning (pp. 10767-10777). PMLR. ISO 690