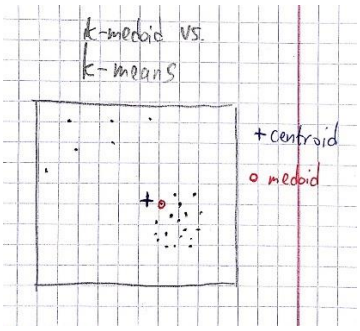
Machine Learning in Healthcare - HW3

Shaked Ron 305048142

1. Clustering

a. K-medoid is an algorithm which is used to cluster data into different groups while the labeling of these proposed groups is unknown (unsupervised learning). Because of this issue (labels are unknown), this algorithm tries to group the data by using specific parameters of their geometry (or distribution). While K-means tries to do the same thing by iteratively calculating geometrical centroids, an n-dimensional expectation of each cluster, K-medoid iteratively seeks for the best representative of each cluster which is the one sample relatively closer to all its other cluster members. As a result of the above, noisy data may cause K-means to converge when a given cluster's expectation was set far from most samples in the cluster, thus can lead to misclassification of samples. On the other hand, K-medoid will overcome the same issue given the fact it is data dependent – cluster's representative will always be defined as one of the samples – specifically one of those closer to most of them. In this aspect K-medoid would be more robust to noise (or outliers).



b. Let us look next on the 1D case of K-means algorithm and to check whether the data mean is the centroid in this specific case:

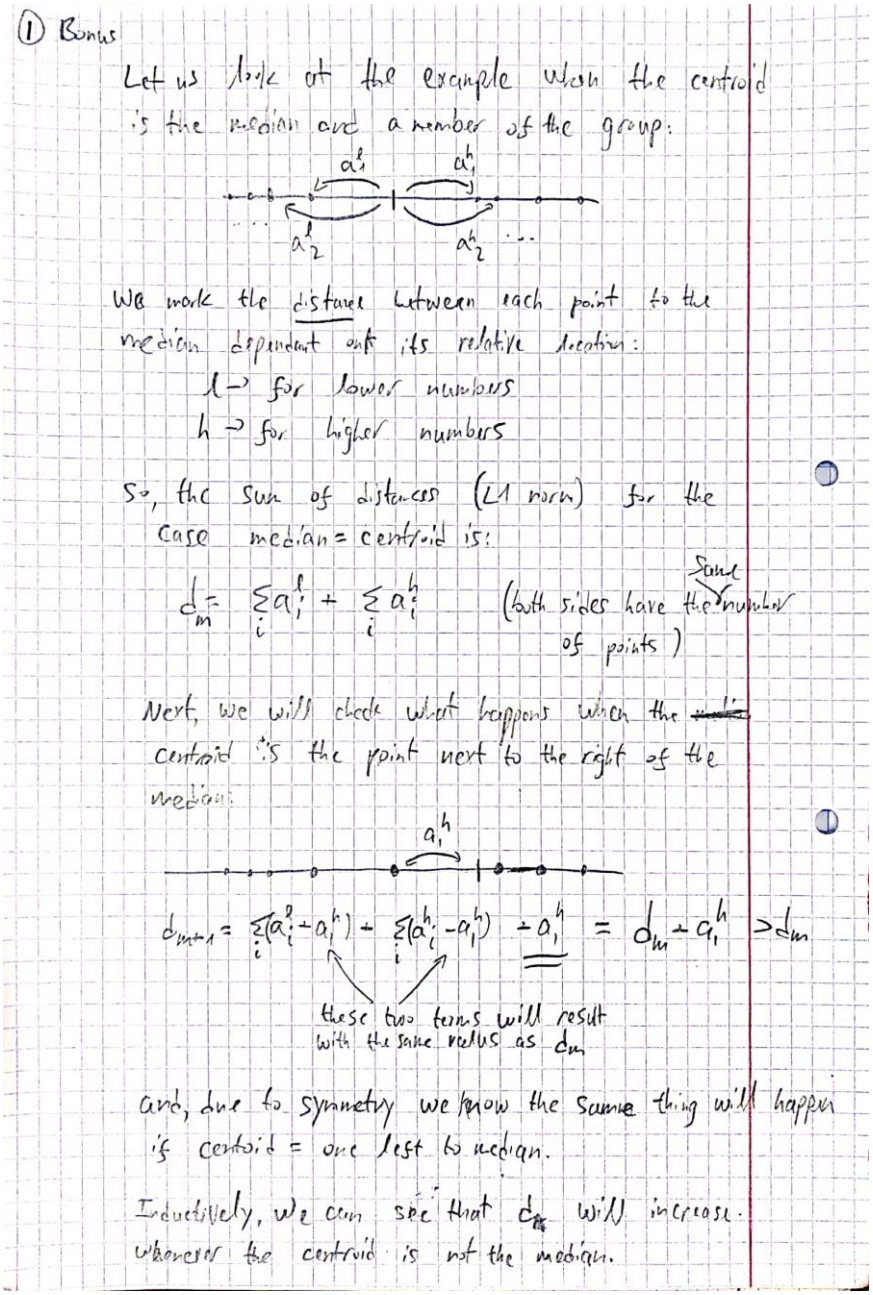
$$minimize \sum_{i=1}^{m}(x_i-\mu)^2 \ for\ \mu$$

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

$$now\ we\ will\ compare\ the\ term\ above\ to\ zero\ in\ order\ to\ find\ the\ minima$$

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

c. (bonus)



While this is the case for an odd number of group members (m=2k+1, k is an integer), in the case of an even number, any point between the two points closest to the middle will be good as the median and as the best centroid.

2. SVM

SVM is a machine learning algorithm used for data classification. SVM utilizes the geometrical distances between data samples in a way determined by the user-chosen kernel. The use of such a kernel let us get computationally quicker results of data separation based on more complex dimensions while applying calculations on the features dimensions. Each kernel has pros and cons and uses different parameters to adjust the goodness of fit.

A – linear C=0.01
D – linear C=1

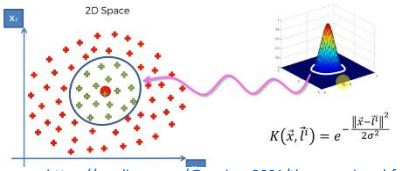
Observing the figures, we can only spot two linear hyper-planes (here: lines). At a closer look, the difference is revealed as in fig A the margins (distances from the decision boundary hyper-plane to support vectors) contain data points (see two purple dots nearly 'on' the line), different than in fig D. This hint tells us that the algorithm in fig D converged to a result more accurately fitting the data, assuming these are the training sets, fig D is more over fitting the data than A, this happens when C is larger. C is a hyper-parameter representing the penalty for misclassification and it is equal to inverse λ. When C is large misclassification is penalized largely thus may result with an overfit and lack of generalization capability. We can see that the margins in D are better fitted to the data – hence based on a higher C. In fig A, C is smaller, we can see that the model 'ignores' the outliers (i.e., the penalty is low) and might be aa better candidate for good generalization.

C – 2nd order polynomial
F – 10th order polynomial

After eliminating the abovementioned two linear machines, we can spot these two figures showing a more complex model, polynomial curves. We should consider that an RBF kernel is also a valid answer, but we can see that the shapes of separation in figs C and F are not presenting a section of Gaussian distribution (elliptic enclosed region). C instantly looks like a parabolic separator, while F is a little more complicated to understand. We can be assisted by the fact that the model seems over-fitting the data in F (a sign for more complex models) and that the classifier seems somehow symmetric with characteristics of an high-degree polynomial function.

E – RBF with γ=0.2
B – RBF with γ=1

After the two steps explained before, let us examine the two figures left, both showing an elliptical and nearly closed shape. This kind of shape can be reached by using a non-linear kernel like RBF (Gaussian) which could be thought of as creating a topographic map of the distances within classes:



https://medium.com/@suvigya2001/the-gaussian-rbf-kernel-in-non-linear-svm-2fb1c822aae0

Gamma is a hyper-parameter used for the measure of "how good do we fit the data?" and is inversely related to the variance of the gaussian distribution used to define the classes. Hence larger gamma will result in a narrower distribution and may lead to an overfit.

3. Capability of generalization

a. The scientific term of the most important balance in machine learning is bias-variance tradeoff. When a model perfectly fits a training dataset (usually a more complex model) it gives a result with low bias and high variance and vice versa for a less perfect fit. A low bias-high variance result is also known as overfitting and would probably result with a bad generalization for the test set and for future input samples. The other end, high bias-low variance (usually a less complicated model) is called underfitting, while making generalization more effective, we risk in learning almost nothing and going close to "coin-toss" decisions. The bottom line, in the perspective of the mentioned Einstein's citation, is we want to find the simplest model required for our purpose but a one that still does the job good enough (satisfying).

b. Let us examine the AIC criterion:

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

2p represents the number of parameters learned during the learning phase of the model (training), as so it represents the complexity of the model. A higher number of parameters means a more complex model and a more probable overfit (low bias-high variance). On the other hand, the likelihood measures how likely it is to get the results that we got. As our results of classification are more likely (goodness-of-fit), L is larger which means the model fits the data better. Note that L is a value in the range between 0 to 1, thus ln(L) will result in a negative number up to zero. A perfect fir will result with a ln(L)=0 while a bad one will increase the AIC value.

c. As mentioned beforehand, violation of this gentle balance would likely cause an over- or under-fitting of the data. Under-fitting would make the model prone to errors due to unsatisfying learning, while over-fitting would result with errors due to lack of capability of generalization of the model (it assumes future data will look exactly like the training set).

d. Usually (and as Einstein commands…) we want our model to best fit the data while it stays as simple as possible. Recall that the likelihood is a measure with values ranging from 0 to 1, thus the natural log of it results with a negative number. For instance, a low likelihood results with a negative value of larger magnitude thus increasing AIC value. additionally, increasing p for instance will also increase AIC value. So, to summarize these phenomena, a simple and accurate model will get low p and high L values, or low p and an almost zero (yet negative) ln(L), hence we would like to minimize our model's AIC (which can be useful in model selection process).