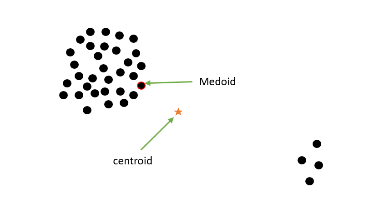
**Machine Learning in Healthcare  
Homework #3**

1. **Clustering:**
2. K-medoid clustering is a more robust algorithm to noise compared with K-means clustering, as both try to find clusters in the data in an unsupervised manner based on the geometry and the determined metric used to define vicinity. As we learnt, K-means algorithm iteratively: computes centroids, assign points based on their closest computed centroid and repeats again till convergence. The centroids therefore most definitely will not be included in the set of given points, so it is a generated parameter that might be influenced critically by the nature of a cluster, mainly when the data is considerably noisy and/or contains significant outliers, so the centroid, which is the reference point of each cluster, might be influenced by noise. On the other hand, K-medoid’s cluster reference must be a sample in the cluster. Even though there might be noise and outliers in a given dataset, an obvious cluster will not be affected severely by extreme samples, outliers will not bias sharply the center of interest of a cluster (outliers are quantitively less than standard data, otherwise outliers will not be considered outliers but another distinct cluster) because the medoid is a part of the cluster, so in order to lower the ‘Energy’ of the entire system, the medoid of a noisy cluster will be among the main the ‘main’ subgroup of the cluster and never will be away of this main subgroup, thus not affected severely even by an extreme outlier. Qualitative visual elaboration:



If a centroid is biased due to outliers, the main cluster might be partitioned if there is another adjacent cluster. On the other hand, we can see that the medoid although be biased, but still remains in the adjacency of the main cluster because it can’t obtain a value that is not among the data values, and it shall be in the adjacency of the main cluster not the noise/outliers to minimize the cost function because grouped outliers are sparse, otherwise they wouldn’t be outliers but rather a distinct cluster. Thus k-medoid is more robust to noise. One more perspective, L1 penalty gives less weight to distant points than L2 penalty, so distant points less affect the loss function, in a way we can look at it as ‘less overfitting’ in a case of distant points.

1. Pertaining the 1D special case of this algorithm, we will try to prove that the centroid that minimizes the loss term is the mean of m examples:
2. **SVM:**

Classification based on SVM algorithm is a very potent approach, and can be optimized by altering the kernel which is the ‘metric’ of the system. The algorithm will try to maximize the margins between the different classes, based on the kernel and the hyper-parameter C which defines the user’s tolerance for misclassification (degree of overfitting). Now we will do matching between the figures and the listed settings.

|  |  |
| --- | --- |
| Figure | Setting |
| *A* | *1* |
| *B* | *6* |
| *C* | *3* |
| *D* | *2* |
| *E* | *5* |
| *F* | *4* |

* Elaborations:
  + For setting 1 and 2 the corresponding figures are A and D accordingly as stated above. It is obvious that the border applied by the algorithm between the two groups is a linear border, thus the kernel for both cases is a linear kernel. Now in order to differentiate and match correctly, it is possible to notice that in figure A., the defined border is set with higher proximity to points, mainly the purple points. A border that is not deterred by the adjacent points even though it is possible to set another ‘conservative’ border (i.e. in figure D.) indicates that the algorithm is relatively more tolerable to misclassifications in the training set thus there is less ‘overfitting’ and more generalizing, meaning that between figure A and D., the hyper-parameter C of the classifier in figure A. is lower than the hyper-parameter C of the classifier in figure D. as we see in figure D., the border is more complied to every point and tries to maximize the margins even for distant points of the main cluster, thus it over-fits the data meaning that the hyper-parameter C is relatively large.
  + For setting 3 and 4 the corresponding figures are C. and F. accordingly as stated above. For polynomial kernels, we expect to get segregation of the data by non-linear borders, but also we expect not to get closed loops/enclosed areas. For figure C. It is relatively straight forward to recognize that the separator is 2nd order polynomial due to its simple non-linearity, for figure F. we can see that the borders are quite complex, segregating without obtaining enclosed areas, so it is not a section of a Gaussian plot thus the kernel is not RBF kernel. The remaining setting is 10th order polynomial kernel which implies a complex behavior of border (setting 4) a behavior that we observe in figure F.
  + For setting 5 and 6 the corresponding figures are E. and B. accordingly as stated above. Other than elimination of figures A., C., D. and F., we are searching for borders that are obtainable by doing a ‘section’ for a 3D Gaussian mixture, which dictates to get a form of enclosed spaces or ‘islands’ due to the morphology of a Radial basis function/Gaussians. Figure E. and B. apply for this criteria. We can see that in figure B. the blue cluster is more tightly enclosed and overly-fitted to the training set (the observed data). On the other hand, figure E. represents a more general border, less tightly-fitted to the data. Relative over-fitting is dictated by relatively higher value of the hyper-parameter , because is inversely related with the variance of the Gaussian distribution. Meaning that higher implies lower variance thus less generalizability and more over-fitting. For that, setting 6 (higher ) is matched with figure B. and therefore setting 5 is matched with figure E.

1. **Capability of generalization:**