Machine Learning in Healthcare – HW3

Theoretical section

Submitted by: Sapir Gershov, 204471577

1. Clustering

a. Both the K-means and K-medoids algorithms are partitional - technique of clustering the data set of n objects into k clusters with k known a priori.

   While the K-means algorithm minimizes the total squared error to find clusters, the K-medoids randomly chooses datapoints as centers (medoids) and minimizes the sum of dissimilarities (distances) between points labeled to be in a cluster and the points designated as the center of that cluster.

   The K-medoid could be more robust to noise and outliers, as compared to K-means, because it minimizes a sum of general pairwise dissimilarities instead of a sum of squared Euclidean distances. The possible choice of the dissimilarity function is very rich, and not necessarily has to be the Euclidean distance.

   A medoid of a finite dataset is a data point from the set, whose average dissimilarity to all data points is minimal (i.e., it is the most centrally located point in the set).

b. Consider 1-D data, $x \in \mathbb{R}^1$, whose proximity measure is the Euclidean distance. For our objective function, which measure the quality of a clustering, we use the sum of square error (SSE). In other words, we calculate the error of each data point, i.e., its Euclidian distance to the closet centroid (μ), and then compute the total sum of the squared errors.

   The SSE is formally defined as follows:

$$SSE = \sum_{i=1}^{m} dist(x_i, \mu)^2 = \sum_{i=1}^{m} (x_i - \mu)^2$$

   where $dist$ is the standard Euclidian (L_2) distance between two objects in the Euclidian space, μ is the cluster centroid and x_i is a data point in the dataset.

   We can minimize the equation above by differentiating the SSE, setting it equal to 0 and solving, as indicated below:

$$\frac{\partial}{\partial \mu} SSE = \frac{\partial}{\partial \mu} \sum_{i=1}^{m} (x_i - \mu)^2 = \sum_{i=1}^{m} \frac{\partial}{\partial \mu} (x_i - \mu)^2 = \sum_{i=1}^{m} \frac{\partial}{\partial \mu} (x_i^2 - 2 \cdot x_i \cdot \mu + \mu^2) = \sum_{i=1}^{m} (-2 \cdot x_i + 2\mu)$$

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

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

   Thus, the best centroid for minimizing the SSE of a cluster is the mean of the points in the cluster.

c. We consider how to partition the data into a cluster such that the sum of the Manhattan (L_1) distances of points from the center of their cluster is minimized. We are seeking to minimize the sum of the L_1 absolute errors (ASE) as given by the following equation:

$$SAE = \sum_{i=1}^{m} dist_{L_1}(x_i, \mu) = \sum_{i=1}^{m} |x_i - \mu|$$

where $dist_{L_1}$ is the L_1 Manhattan distance, μ is the cluster centroid and x_i is a data point in the dataset. We can minimize the equation above by differentiating the SAE, setting it equal to 0 and solving, as indicated below:

$$\frac{\partial}{\partial \mu} SAE = \frac{\partial}{\partial \mu} \sum_{i=1}^{m} |x_i - \mu| = \sum_{i=1}^{m} \frac{\partial}{\partial \mu} |x_i - \mu| = 0 \rightarrow \sum_{i=1}^{m} sign(x_i - \mu) = 0$$

If we solve for μ, we find that $\mu = median\{x_i \in C\}$, the median of the points in the cluster.

2. SVM

Support Vector Machine (SVM) is a supervised machine learning algorithm for classification tasks. SVM separates data points that belong to different classes with a decision boundary. When determining the decision boundary, the SVM model tries to solve an optimization problem with the following goals:

- Increase the distance of decision boundary to classes (or support vectors)
- Maximize the number of points that are correctly classified in the training set

There is a trade-off between these two goals which is controlled by the hyperparameters:

Kernel

The kernel hyperparameter chooses the transformation we perform on our data. Choosing the right kernel is crucial, because if the transformation is incorrect, then the model can have very poor results. We can choose between linear kernels and non-linear, depending on our data separability.

C

The C hyperparameter adds a penalty for each misclassified data point.

If C is small, the penalty for misclassified points is low so a decision boundary with a large margin is chosen at the expense of a greater number of misclassifications. If C is large, SVM tries to minimize the number of misclassified examples due to the high penalty which results in a decision boundary with a smaller margin.

Gamma

Gamma is a hyperparameter used with non-linear SVM.

Low values of gamma indicate a large similarity radius which results in more points being grouped together. For high values of gamma, the points need to be very close to each other in order to be considered in the same group (or class). Therefore, models with very large gamma values tend to overfit.

Setting	Image	Explanation
1. Linear kernel with C = 0.01	D	Images A and D have a linear kernel, and If the hyperparameter C is **small**, the decision boundary will have a much **larger** margin
2. Linear kernel with C = 100	A	Images A and D have a linear kernel, and If the hyperparameter C is **big**, the decision boundary will have a much **smaller** margin
3. 2nd order polynomial kernel	C	2nd order polynomial kernel will have a decision boundary with the shape of a quadric function
4. 10th order polynomial kernel	F	10nd order polynomial kernel will have a decision boundary with the shape of a function in the power of 10
5. RBF kernel with $\gamma = 0.2$	E	Low values of gamma indicate a **large** similarity radius in the cluster
6. RBF kernel with $\gamma = 1$	B	High values of gamma indicate a **small** similarity radius in the cluster

3. Capability of generalization

a. The scientific term of the balance that Einstein mean is 'Generalization'. Generalization is a term used to describe a model's ability to react to new data. That is, after being trained on a training set, a model can digest new data and make accurate predictions. A model's ability to generalize is central to the success of a model. A proper generalized model will assure **balance** between goodness-of-fit and complexity.

b. The best-fit model according to AIC is the one that explains the greatest amount of variation using the fewest possible independent variables. Thus, AIC rewards goodness of fit (as assessed by the likelihood function), but it 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.

c. If a model has been trained too well on training data it will be unable to generalize, since too many parameters generate a too complex model. Too complex model will make inaccurate predictions when given new data, making the model useless even though it is able to make accurate predictions for the training data. This is called **overfitting**. The inverse is also true. **Underfitting** happens when a model has not been trained enough on the data, since few parameters generate a too simple model. In the case of underfitting, it makes the model just as useless and it is not capable of making accurate predictions, even with the training data.

d. Lower AIC scores are better, and AIC penalizes models that use more parameters. Hence, if two models explain the same amount of variation, the one with fewer parameters will have a lower AIC score and will be the better-fit model.