Machine learning in healthcare- HW3

Asrar Khaldey-209122506

Q1- Clustering

Section A

The K-Medoids algorithm is used to find Medoids in a cluster which is center located point of a cluster. K-Medoids is more robust as compared to K-Means as in K-Medoids we find k as representative object to minimize the sum of dissimilarities between points labeled to be in a cluster and a point designated as the center of that cluster, whereas, K-Means attempts to minimize the total squared error and uses sum of squared Euclidean distances for data objects. And this distance metric reduces noise and outliers. Thus K-Medoids is more robust to noise and outliers as compared to K-Means.

Section B

$in\ order\ to\ find\ the\ minimum\ we\ can\ differentiate\ the\ term\ \sum_{i=1}^{m}(x_i - \mu)^2\ \ by\ \mu\ and\ equal\ it\ to\ zero$:

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

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

$$= \sum_{i=1}^{m}\frac{\partial}{\partial\mu}({x_i}^2 - 2x_i\mu + \mu^2) = \sum_{i=1}^{m}(-2x_i + 2\mu) = 0$$

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

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

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

$therefore, the\ centroid\ \mu\ which\ minimizes\ the\ term\ \sum_{i=1}^{m}(x_i - \mu)^2\ \ is\ the\ mean\ of\ m\ examples.$

Bonus

$in\ order\ to\ find\ the\ minimum\ we\ can\ differentiate\ the\ term\ \sum_{i=1}^{m}(|x_i - \mu|)\ \ by\ \mu\ and\ equal\ it\ to\ zero$:

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

$if\ we\ solve\ for\ \mu\ we\ get\ that\ \mu\ equal\ to\ the\ median\ of\ m\ examples.$

Q2- SVM

Linear kernel is used when the data is linearly separable, in linear kernels the decision boundaries are only curved in the input space, in the implicit, higher-dimensional feature space they are straight lines or planes.

The C parameter tells the SVM optimization, how much we want to avoid misclassifying each training example.

For large values of C, the optimization will choose a smaller-margin hyperplane if that hyperplane does a better job of getting all the training points classified correctly. (2-A)

Conversely, a very small value of C will cause the optimizer to look for a larger-margin separating hyperplane, even if that hyperplane misclassifies more points. (1-D)

Polynomial kernel allows us to learn patterns in our data as if we had access to the interaction features, which are the features that come from combining pre-existing features (a^2, b^2, ab, etc).

The degree parameter controls the flexibility of the decision boundary. Higher degree kernels yield a more flexible decision boundary. (C-3, F-4)

with **Radial Basis Function kernel** we can utilize the kernel to build very complex decision boundaries. The more dimensions of feature space, the better chance we will find a hyperplane that neatly separates our data.
The γ parameter acts as a regularizer — the smaller it is, the smoother the decision boundary, which prevents overfitting, in other words, the gamma parameter defines how far the influence of a single training example reaches, with low values meaning 'far' and high values meaning 'close'. The gamma parameters can be seen as the inverse of the radius of influence of samples selected by the model as support vectors. (E-5, B-6)

Q3- Capability of generalization

Section A

In machine learning we require that the model learn from known examples and generalize from the learned examples to new examples, to do that we use methods like a train/test split or k-fold cross-validation only to estimate the ability of the model to generalize to new data.

When the model is exposed to very little amount of data it will perform poorly on the training dataset and on new data. The model will under fit the problem. In the other hand too much data and the model will perform well on the training dataset and poorly on new data, the model will over fit the problem. In both cases, the model failed to generalize.

Therefore, what we need is to find the balance between model's complexity and performance.

Section B

In estimating the amount of information lost by a model, AIC deals with the trade-off between the goodness of fit of the model and the simplicity of the model.

By looking at AIC value : AIC=2p-2ln(L), we can say that:

'p' is the number of independent variables used to build the model (The more complex the model is more variables needed to get the estimate or prediction). 'L' the maximum likelihood estimate of the model (how well the model reproduces the data).The function 2ln(L) contributes to the goodness of function, when it gets higher the AIC gets lower value and it means that the model is fitted better.

Section C

AIC deals with both the risk of overfitting- expanding the complexity of the model will not appreciably improve its representation of the data, adding more terms in AIC equation will eventually result in the model being over fit or excessively tuned to the data used for parameter estimation- and the risk of under fitting- having a model that is so simple and not trained enough, and by that it is not possible to get accurate predictions for the new data sets.

Section D

AIC estimates the relative amount of information lost by a given model (the less information a model loses, the higher the quality of that model). Therefore, when given several models for the data, the preferred model is the one with the minimum AIC value.

In other words, goodness of fit is preferred (as assessed by the likelihood function), but it also includes a penalty that is an increasing function of 'p' which discourages overfitting, and in the end improves the goodness of the fit.