

Theoretical Questions – HW3

Fatima Abbas -208808154

1. Clustering

A.

k-medoid is more robust to noise and outliers compared to k-means because it minimizes a sum of pairwise dissimilarities instead of a sum of squared Euclidean distances. In other words, k-means selects the cluster center, which is mostly just a "virtue point" towards the outliers. On the other hand, k-medoid chooses the "actual object" from the cluster, the "most centered". For example, suppose we have five 2D points in one cluster with the coordinates of (1,1), (1,2), (2,1), (2,2), and (100,100). If we do not consider the object exchanges among the clusters, with k-means we will get the center of cluster (21.2, 21.2) which is pretty distracted by the point (100,100). However, with k-medoid will choose the center among (1,1), (1,2), (2,1), and (2,2) according to its algorithm.

B.

In order to prove that the centroid μ , which minimizes the term $\sum_{i=1}^m (x_i - \mu)^2$ is the mean of m examples, we can differentiate the $\sum_{i=1}^m (x_i - \mu)^2$ and equaling it to zero:

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

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

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

Thus, the best centroid for minimizing the term is the mean of m examples.

Bonus:

In order to prove that the centroid (practically, the medoid) which minimizes the term $\sum_{i=1}^m (|x_i - \mu|)$ is the median of m examples (given that μ belongs to the dataset). We can differentiate the term and equaling it to zero:

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

And that's means if we solve for μ we find that $\mu = \text{median}\{x_i\}$; the median of m examples.

2. SVM

Linear kernels compute similarity in the input space. They do not define a transformation to higher dimensions. Because of this, each of the hyperplanes in the figure must be straight lines (suitable for figures A & D). The C parameter tells the SVM optimization, how much we want to avoid misclassifying each training example.

The larger values of C, the more penalty SVM gets when it makes classification → the optimization will choose a smaller-margin hyperplane and for that, since A have bigger line slope, we can conclude that 2(C=1) goes for 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. So we can conclude that 1(C=0.01) goes for D.

→ 1-D 2-A

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). SVM with Polynomial kernel can generate a non-linear boundary (suitable for figures C & F). The degree parameter controls the flexibility of the decision boundary. Higher degree kernels yield a more flexible decision boundary. Second order polynomial kernel will have a decision boundary with the shape of a quadratic function (one curvature). So we can conclude that 3 (2nd) goes for C, 4(10nd) goes for F.

→ 3-C 4-F

The RBF feature space has an infinite number of dimensions. This means that we can utilize the kernel to build very complex decision boundaries. The more dimensions, the better chance we will find a hyperplane that neatly separates our data.

The radial basis, part of the name, comes from the fact that this function decreases in value as it moves away from the center. This explains why the decision boundaries are bell-curve shaped.

The γ parameter acts as a regularizer, the smaller it is the smoother the decision boundary, which prevents overfitting. In other words, it defines how far the influence of a single training example reaches. 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. So we can conclude that 5 ($\gamma=0.2$) goes for E, 6($\gamma=1$) goes for B.

→ 5-E 6-B

3. Capability of generalization

- a. The scientific term of the balance that Einstein meant to in machine learning aspect is generalization (model's ability to react to a new data). "A proper generalized model will assure balance between goodness of fit and complexity". In other cases, generalization of the model by testing it on test group (new data) will result in large error. The model must be trained on an enough training set (not too much data cause this will give a complex model), and by then can digest new data and make accurate predictions.
- b. The higher the number of learned parameters (p), the more complex the model and the more probability of overfitting. The estimated likelihood function $2 \ln(\hat{L})$, contributed to the goodness of fit of the model, the larger it gets the lower the AIC value is ($AIC = 2p - 2\ln(\hat{L})$), and by that the model is fitted better to the training set.
- c. This balance could be violated by overfitting, which happens when we have too complex model (trained on too much training data), that's mean it is too accurate for the training data and by that it can't give accurate predictions for new data. The balance could be violated also by underfitting, which happens when we have too simple model (not enough training data), and by that, it can't give accurate predictions for the training data itself so for sure not for a new data.
- d. The preferred model is the one with the minimum AIC value (indicate a better fit), since AIC estimates the relative amounts of information lost by the model. The AIC is given by ($AIC = 2p - 2\ln(\hat{L})$), therefore, in order to get the minimum AIC we need to find the balance between reducing the number of parameter(p) and increasing the log likelihood function.