HOMEWORK 1

TDT4173: MACHINE LEARNING

Jamal Sharifi

17. september 2022

K-means

How does the algorithm work on a technical level and what kind of machine learning problems is it suited for?

K-means clustering is an unsupervised learning method that is well suited for identifying similar groups within the dataset. This method splits the data into k clusters of data points, where each datapoint is assigned to the nearest centroid (center of the cluster). The number of clusters assigned to a dataset should be subject to maximizing the silhouette and minimizing the distortion. The algorithm can be implemented using the following steps:

1. Choose k centroids at random
2. Distribute the data points to the nearest centroids
3. Choose new k centroids by adjusting the current clusters
4. Repeat step 2 and 3 until the difference between old and new centroids is acceptable

What is its inductive bias, i.e., what assumptions does it make about the data in order to generalize?

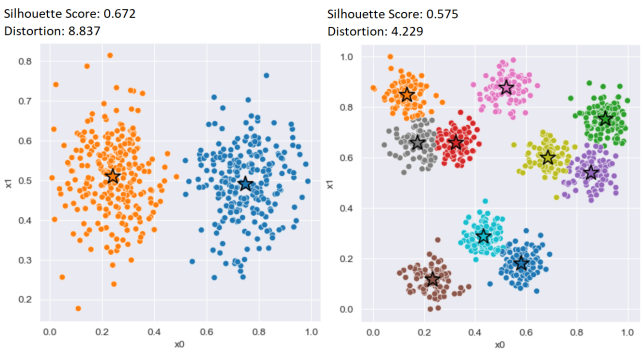
It is assumed that the dataset has clusters with similar distances. Transforming the coordinate system or using non-euclidean distance as measurement should not affect the results. However, in this assignment I have used euclidean distance and cartesian coordinates. It is then assumed that the dataset have k clusters that have d similar euclidean distances, and therefore has identifiable clusters.

What happens in the second dataset that makes it harder than the first and how does this problem relate to the algorithm's inductive bias?

In the second dataset, data points are scattered differently than in the first dataset. Here, the feature x0 is around 10 times feature x1. Computing the Euclidean distance without scaling the coordinates will give you wrong centroids, and therefore wrong clusters.

What modifications did you do to get around this problem?

What modifications did you do to get around this problem? I have scaled the coordinates so that the features x0 and x1 have the same weights, and implemented k-means++ algorithm to reduce the time it takes to find all the clusters correctly. By using the k-means++ algorithm, first centroid is chosen at random from all the datapoints and then the remaining centroids are chosen based on the farthest point of all clusters. This is repeated until k initial centroids are found and the final clusters are determined using the regular k-means algorithm.



(a) First dataset (b) Second dataset

Fig. 1: Possible solutions for the two datasets.

Logistic Regression

How does the algorithm work on a technical level and what kind of machine learning problems is it suited for?

Logistic Regression is well suited for handling binary problems, such as whether something is true or not. It is a supervised learning method that tries to predict the outcome of a variable. In this exercise we have used the activation function sigmoid function to make prediction for the variable. The sigmoid function gives out a number between 0 and 1, which is then used to calculate the cost function and this function tells us how close the predictions are to the true values. The models parameters are w and b, and these parameters are updated so that they produce least error in the model. This is done by using Gradient descent.

What is its inductive bias, i.e., what assumptions does it make about the data in order to generalize?

It is assumed that the dataset has groups or classes that can be linearly separated. Further, the dataset should consist of a large sample size and the data to be independent.

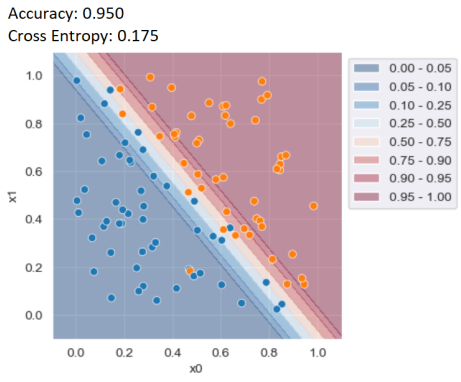


Fig. 2: Possible solution for the first dataset.

What happens in the second dataset that makes it harder than the first and how does this problem relate to the algorithm's inductive bias?

The second dataset has a class of data inside the other class, so by using the same algorithm without separating the data properly we get bad accuracy results.

What modifications did you do to get around this problem?

I have solved this problem using two methods. The first is to take advantage of symmetry and rotate all the datapoints by $\pi/4$ clockwise, then changing the sign of all x-coordinates $x \leq 0$. This way we get a clean line separating the classes (fig. 3).

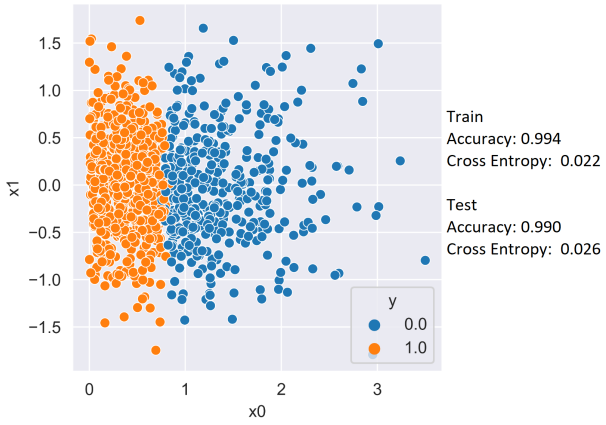


Fig. 3: Possible solution for the second dataset using rotation.

The second solution is found when separating the classes by mapping the feature vectors into 3d

$$\theta(x, y) = \begin{bmatrix} x^2 \\ \sqrt{2}xy \\ y^2 \end{bmatrix}$$

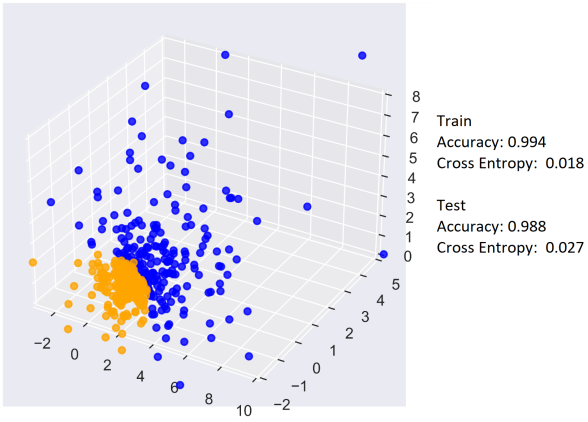


Fig. 4: Possible solution for the second dataset using the kernel trick.