Machine Learning assignment 1

In this assignment I chose to implement the k-means and logistic regression algorithms.

K-means

K-means is one of the most popular unsupervised machine learning algorithms. The algorithm manages to group data in different groups based on their attributes. It is an unsupervised algorithm, meaning that it does not receive any information concerning if the results are correct or incorrect. This algorithm is therefore suited for problems like spam mail sorting.

K-means works by first choosing k different centroids, this can be done randomly or by more advanced initialization methods like k-means++. Afterwards a distance function is chosen to calculate the distance between each datapoint and centroid, thereafter all data points are assigned to the centroid with the closest distance. In this assignment I chose to use the euclidean distance to do so. The next step is to calculate the mean value for all the data points assigned to each centroid, and then re-assign the centroid to the mean value until convergence. By implementing this we get the following results on dataset 1.

Modifications

K-means is of course not perfect, and has a couple of inductive biases. Firstly it assumes that the variance is spherical and the same for each variable. Secondly, the prior probability for each of the k clusters are equal.

In the second data set the first assumption about spherical variance is broken, as the clusters have a mean height of x1 = 0.2, and mean width of x2 = 2.0. So we have to transform the data set to where the clusters are more circulare (See the transform(X) function).

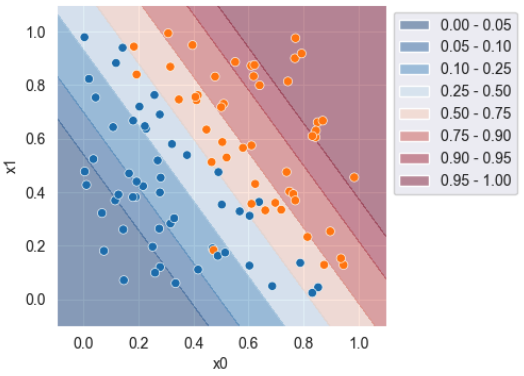
But still the algorithm won't find all the 10 clusters every time, this is because there are multiple local minima, and prior probability for all the clusters are not equal. So I implemented the k-means++ initialization method where spreading out the k initial cluster centers is a good thing. I also implemented a loop with 10 epochs which compares the different local minima distortion scores, and chooses the centroids with the lowest distortion. This resulted in the algorithm consistently finding the 10 clusters.

Logistic Regression

Logistic regression is a very useful supervised machine learning algorithm for handling binary classification. It is a supervised algorithm, meaning that for each prediction, it also receives the correct answer and uses this to improve its parameters. Thereby using training sets with expected results to learn, and thereafter predicting results on testing sets where the correct result is “unknown”. The algorithm works firstly by assuming that the data set is linearly separable, meaning that for n features, there exists a linear equation

$h_\theta(x) = \theta_0 + \theta_1 x_1 + \ldots + \theta_n x_n$

which separates the data correctly (linear regression). To find this equation we need a cost function that tells us how far off we are, we use the ordinary least squares method to do so. The next step is to use a learning algorithm which based on the correct answer, minimizes the cost function by updating the θ_i-values.

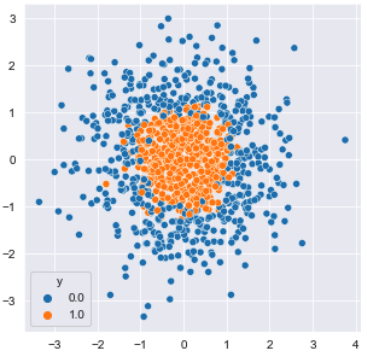


So if we take the partial derivative of the cost function we can iteratively work us towards the local minima (gradient descent). The last step is to implement this into a logistic regression where $h_\theta(x)$ in the cost function, is instead defined as the logistic sigmoid function $h_\theta(x) = \frac{1}{1+e^{-(\theta_0+\theta_1 x_1+\ldots+\theta_n x_n)}}$. Now we can finally logistically predict our data with $h_\theta(x)$, and update out parameters with $\theta_j := \theta_j + \alpha(y^{(i)} - h_\theta(x^{(i)}))\, x_j^{(i)}$, although I vectorized my data and used batch gradient descent. This gives us the following result on dataset 1.

```
Accuracy: 0.920
Cross Entropy: 0.287
```

Modification

Logistic regression has a couple of inductive biases. Firstly logistic regression requires all the observations to be independent, and a large sample size. Secondly it assumes linearity of independent variables. If we plot the second data set it becomes clear that the second mentioned inductive bias is not fulfilled, since the data is spread out in circular fashion. We can use the Kernel trick and map the data to another space where the independent variables are linearly separable. In polar coordinates this logic applies for visual obvious reasons where the only separating feature between the two groups is the radius. The transformation involves first to find the orgio in the new space, and thereafter to calculate the radius of each point relative to the new origo. This results in the algorithm being successful over 90% of the time as seen to the right.




```
Train
Accuracy: 0.906
Cross Entropy:  0.217

Test
Accuracy: 0.902
Cross Entropy:  0.199
```