Report Task 1

TDT4172

K-Means

K-means is an unsupervised learning algorithm for clustering problems. It can be used for feature learning and lossy data compression among other things.

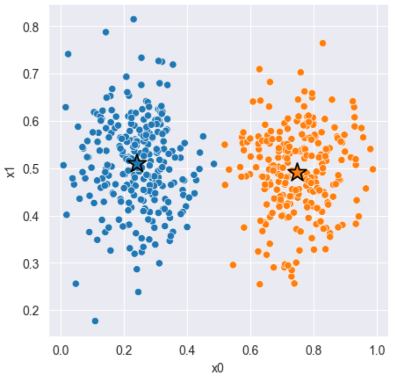
In my implementation we initialize a predefined number, k, of centroids at random. Then for each data point, lable it with the closest centroid in terms of euclidean distance. For each of the centroid classes, calculate the mean and let these be the new centroids. After a number of iterations, the centroids will converge to a stable solution (usually), and the algorithm stops. If we end up in a situation where the centroids does not stabilize, i have a max number of iterations to ensure that the algorithm terminates. In addition the algorithm runs several times and returns the best solution in terms of the distortion in order reduce the chances of returning a local minima.

Inductive bias

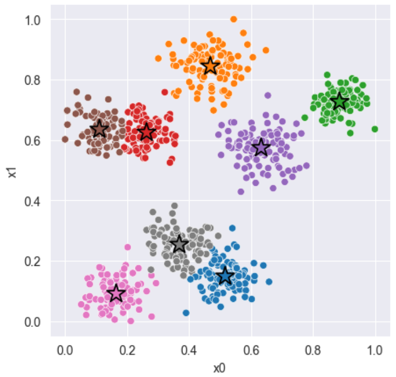
Because k-means uses euclidean distance to classify, the decision boundaries are linear, and in the case of multiple clusters this results in regular and convex areas. The clusters thus end up having roughly equal variance in all dimentions. Additionaly it assumes that a cluster can be represented well by its mean, which further favours regular clusters.

Plots

Data 1 clusters



Data 2 clusters



Results

In problem 2 the scales of the two features were of different magnitude. I worked around this problem by normalizing the data to make both features have $[0, 1]$ as scale. Simply shifted by minimal values and scaled by difference of min and max.

As each centroid is the mean of its closest points the algorithm is sensitive to features of different scales. Naively using k-means on unscaled data has the problem that the inductive bias tends to favor clusters with equal variance i all dimmentions, which in our case in problem 2 was not too different form just clustering with respect to x_0.

	Data 1	Data 2
Distortion	59.082	48.870
Silhouette Score	0.672	0.592
k	2	8

Logistic regression

Logistic regression is suited for binary classification tasks on data which is approximately linearly separable. As with most regression models there are strong assumptions on linear relationships between feature and response.

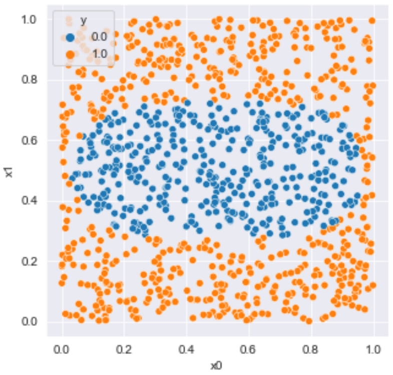
In the training phase we use maximum likelihood estimation to estimate the parameters using gradient decent. In this we assume independent observations to get a tractible likelihood function. We initialize the coefficients, make a prediction using the corrent coefficients and update using the gradient decent update rule $\theta \leftarrow \theta + \alpha \nabla_\theta l(\theta)$.

Inductive bias

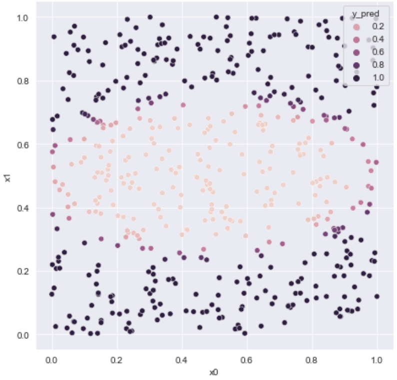
An important part of the inductive bias of logistic regression it that it assumes you can express the decision boundary as a linear combination of the features.

Plots

Data 2



Prediction data 2



Results

In the second data set the classes are not linearly separable in the features. In order to work around the problem i enlarged the features space with squares and cross-terms as the decision boundary is an ellipse.

The test and train results for dataset 2 with learning rate 0.05 and 10000 epochs was:

Data 2	Train	Test
Accuracy	0.984	0.948
Cross Entropy	0.063	0.096