

1.

a)

Random initialization

I generate a random vector of integers in the range 0 to number of points and initialised the means to the corresponding examples to make sure the initialised means are among the points we are classifying and are not far away from the clusters.

I have kept a maximum of 50 iterations to terminate. If however after an iteration the **class values remain same** then i terminate it right there as the next u values would be same and there is no point running it again.

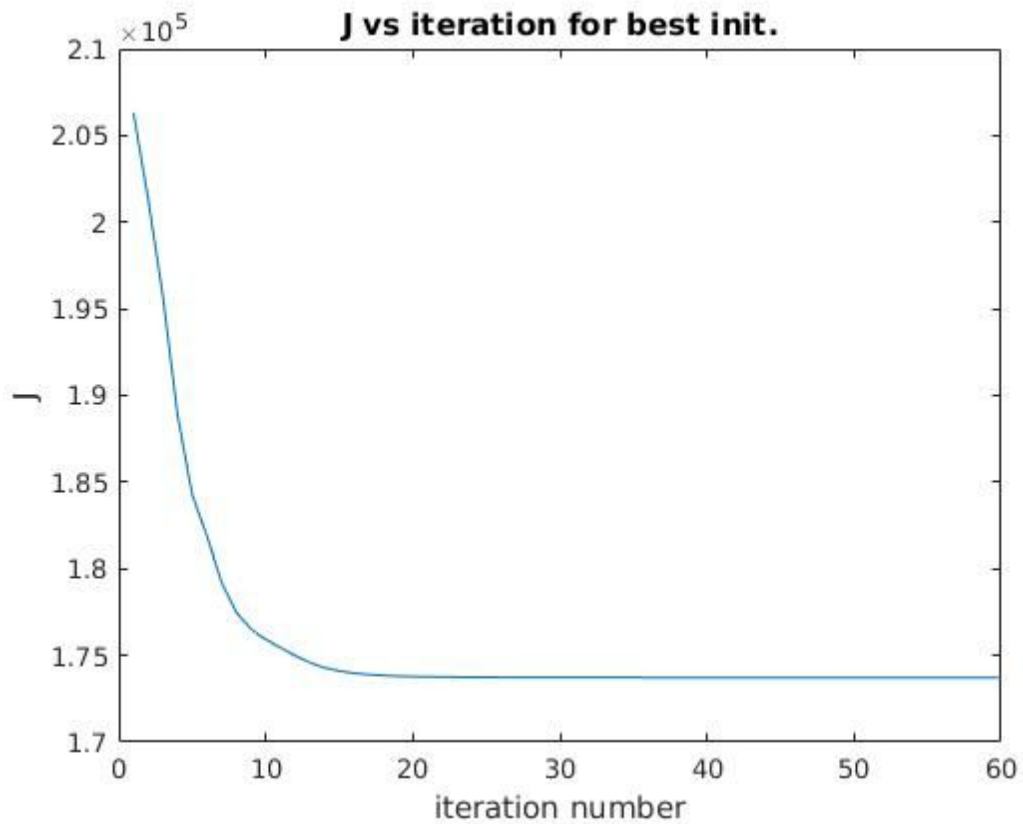
The average number of iterations to converge is around 27.

This number depends upon the initialisation.

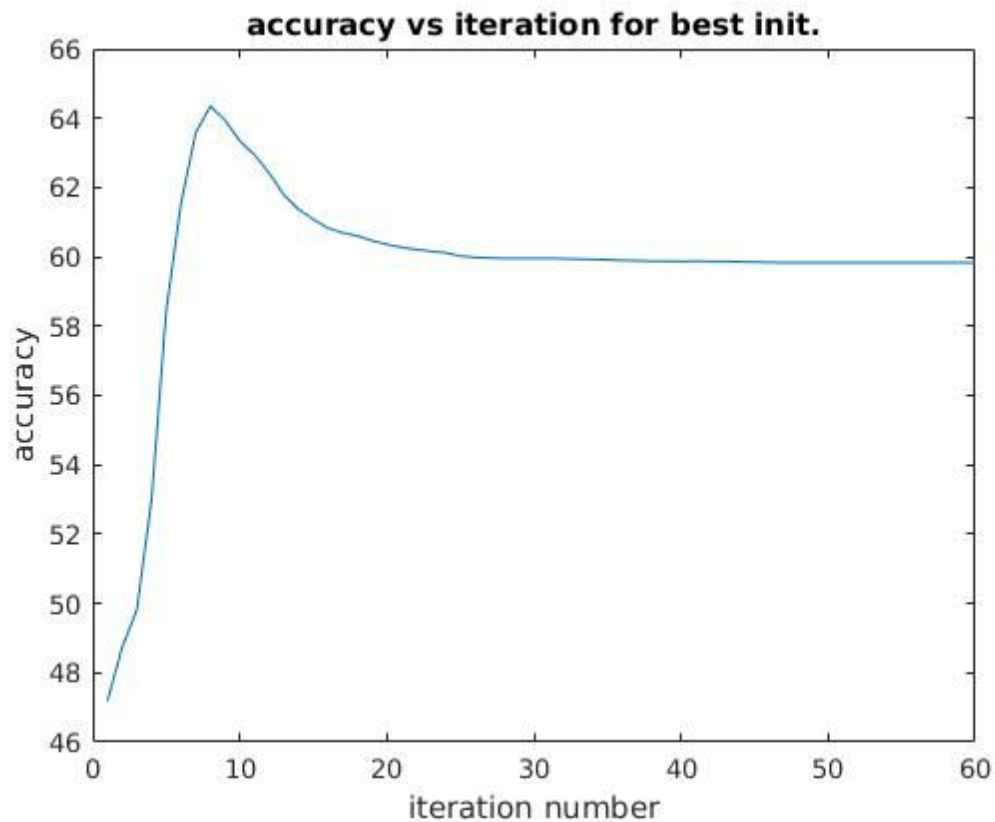
b)

Iteration	J
1	1.7437e+05
2	1.7372e+05
3	1.7466e+05
4	1.7437e+05
5	1.7956e+05
6	1.7372e+05
7	1.7372e+05
8	1.7956e+05
9	1.8006e+05
10	1.7372e+05

Apart from slight variations the Jvalue remains almost constant as they are the case where the individual cases converge.



As we can see initially the error is maximum as the means are assigned randomly. As the number of iterations increase the value of error decreases as the means get better estimated and the points are rightly formed in a group. After around 40 iterations the error value saturates. The final value of error is 1.737×10^5



c)

Initially due to random initialisation the accuracy is quite low. As the number of iterations increases the accuracy increases, reaches a maximum value and then decreases on further running. This may be due to the fact the classification of some points is not in the domain which they are closest. Having a probabilistic approach would have helped here.

d)

Cross validation accuracies

```
[ 0.96360153  0.94439118  0.88771593  0.97216891  0.98655139  0.98174832
  0.97982709  0.96346154  0.96246391  0.9894027 ]
```

These accuracies are very high as compared to k-means as this is supervised learning. We are providing both x and y to the classifier to learn and so we expect the accuracy to be high as observed. There is no need for clustering and more data in the form of labels is available for learning.

2.

a)

Dataset1



Dataset2



b) Done

c)

Dataset1

1



2.



3.



4.



5.

DATASET2



1.



2.



3.



4.



5.

e)

Dataset1

Original accuracy : 79.58%

Accuracy on projected attributes: 59.7%

The number of attributes is reduced drastically and the decreasing the dimension leads to less information being captured. So the accuracy decreases. Time taken is considerably less as number of attributes is quite low

DataSet2

Original Accuracy : 98.75%

Accuracy remains the same on projected set of attributes but the time taken is considerably less. This is because of the low number of attributes to learn for the SVM

f) The projected image is somewhat similar the original case. In dataset1 the size is too small to observe closely. But in dataset2 we can clearly identify the similarity. Due to normalization the images becomes somewhat dusky otherwise it would have been much better.