Nat Manley, u1906046

Machine Learning Coursework Report

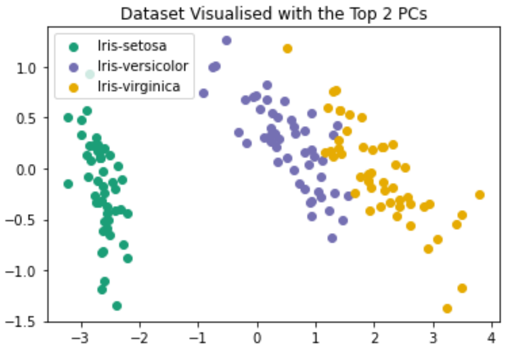

Figure 1 Flower Dataset Coloured by ground truth values

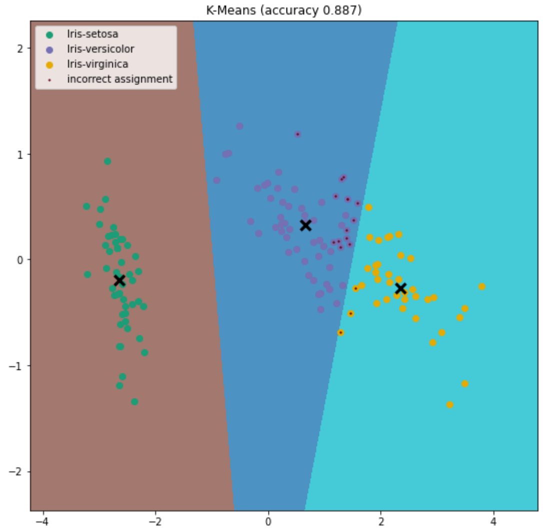

Figure 1 Original K-Means Clusters

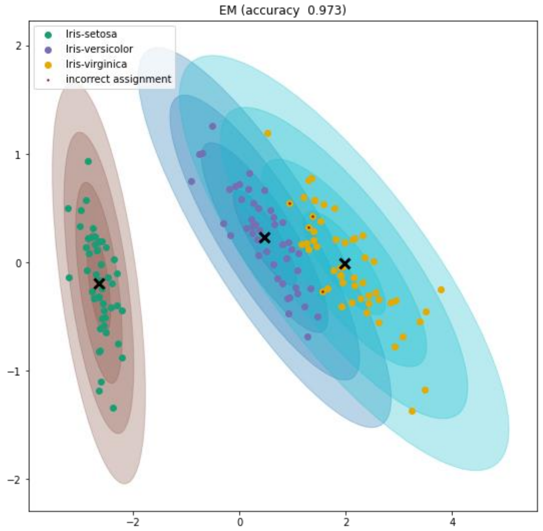

Figure 3 GMM Clusters after EM algorithm

The original cluster centres were calculated with `sklearn.clusters.KMeans`, then the starting theta values specified as follows:

```
self.theta.append([Cluster(1/3, mean, [[1,0],[0,1]])
                              for mean in self.kmeans.cluster_centers_])
```

The initial prior value was set to 1/3, which is the assumption that a point is equally likely to belong to any cluster, ‘mean’ is the centroid calculated via K-Means, and the initial sigma value is set to the identity matrix, meaning that the clusters of the initial theta are all circular.

The final accuracy of the GMM clusters after the EM algorithm is significantly higher than that of base K-Means. This is because K-Means works under the assumption that all clusters are circular, and so only has to consider the Euclidean distance to cluster centroids to decide which cluster a datapoint is assigned to, whereas the EM-Algorithm models clusters as gaussian distributions, with the algorithm trying to maximize the likelihood of those distributions producing the data. This allows more complex cluster arrangements such as non-circular clusters (although still Gaussian).

Although not shown well in this dataset, the EM algorithm also allows for differing cluster sizes as seen in the following figure:

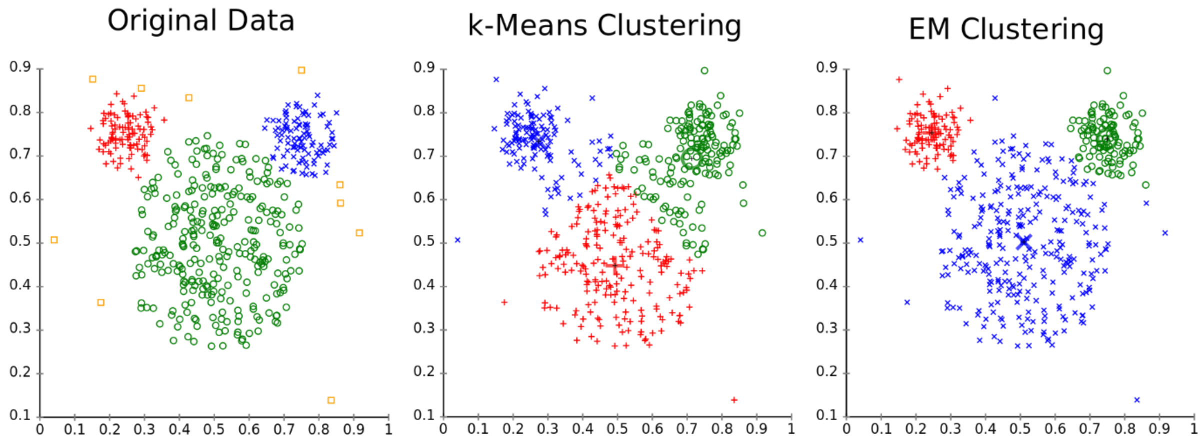

Figure 2 differing cluster assignments *[www.mghassany.com/MLcourse/gaussian-mixture-models-em.html]*

The EM-algorithm is more flexible than K-Means, and so often has higher accuracy compared to the ground truth, but this is only as long as the data can be modelled by a gaussian distribution. If this is not the case, then the fundamental assumption of the EM-Algorithm is wrong, and so it may not produce the best results.

Despite the advantages of the EM-algorithm over K-Means, it still doesn't provide an optimal solution. As it is an unsupervised learning approach, it doesn't have access the underlying ground truth values, and so outliers can often be misclassified, especially if two clusters are close together, as the boundaries between categories are almost always blurry in real-world data. For example, in our training dataset of flowers, the sepal and petal sizes for the different flowers follow different general trends, but at the extremes, can be quite similar between Iris-Versicolour and Iris-Virginica.

The EM algorithm isn't identifying flowers, it is only identifying the clusters that have the highest probability of modelling the datapoints, and so it cannot differentiate outliers. This means that 100% accuracy is mostly unachievable with this technique, and would be undesirable anyway due to the risk of overfitting, as can be seen in the figure below:

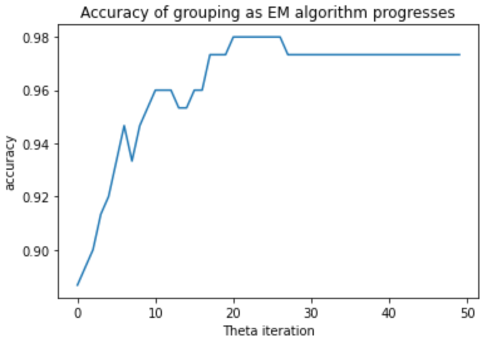

Figure 3 Accuracy of Final Clusters

The Accuracy increases quickly from the initial k-means assignment of spherical GMM clusters. The accuracy of the model peaks for Θ^{21}, but then settles and doesn't change from around Θ^{30}. While the accuracy is less at this point than earlier, the likelihood that the data was produced by these distributions is greater. There are several methods that can be used to decide when to stop iterating including fixed iteration counts [1] [2], or analysing the increase to the likelihood, and stopping when this is sufficiently small [3]. These techniques and others were analysed in [4], which showed that measuring the increase in likelihood, while more applicable to general problems than a fixed iteration count, did not always produce the best result, as the variance of the GMM model is generally the last to converge.

I experimented with using the change of position of the cluster centroids to decide convergence, only stopping iteration once every cluster changed less than a specified epsilon before stopping:

```
oldCentroids = [c.mu for c in theta_o]
newCentroids = [c.mu for c in theta_n]

difference = [mu1-mu2 for mu1, mu2 in zip(oldCentroids, newCentroids)]
converged = all([np.linalg.norm(diff) <= epsilon for diff in difference])
```

However, setting the right epsilon was difficult, as setting it too low meant that there were far more iterations than necessary, as the values in Θ were barely changing, and increasing it meant that it stopped too soon. Due to this, I decided that the difference in centroid position was a poor decider for convergence and so used a different technique.

The final solution uses a fixed iteration count based on the data shown in fig.5 which shows that the difference in accuracy of the model on the dataset does not change after 30 iterations of the EM algorithm. This is the same number of iterations as chosen in [2]. I didn’t think the added complexity of tracking values was worth the benefit, as it would just make the algorithm more generalizable to other datasets which, while useful, is beyond the scope of the project. Using this iteration count to decide convergence gave an accuracy of 0.973, with only 4 datapoints miss-classified. This is far better accuracy than achieved in [2], and so I believe this is an acceptable point to assume convergence.

References

[1] T. B. M. C. J. R. a. P. J. Gilland D.R, “An Evaluation of Maximum Likelihood-Expectation Maximization Reconstruction for SPECT by ROC Analysis,” *The Journal of Nuclear Medicine,* vol. 33, pp. 451-457, 1992.

[2] F. J. a. J. I. Permuter H, “A study of Gaussian mixture models of colour and texture features for image classification and segmentation,” *Pattern Recognition,* vol. 36, pp. 695-706, 2006.

[3] C. C. a. G. H, “Blobworld: Image Segmentation using Expectation-Maximization and its Application to Image Querying,” *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 24, pp. 1026-1038, 2002.

[4] E. E.-D. C. V. P. M. Revlin Abbi, “Analysis of stopping criteria for the EM algorithm in the context of patient grouping according to length of stay,” *4th international IEEE Conference Intelligent Systems,* pp. 3-9, 2008.