Jared Frazier | CSCI 4350 | OLA 4 | Unsupervised Learning with K-Means | Due 2021/11/30

Introduction

In pattern recognition, image processing, and many other problems, the task of finding groups of data points that have similar features is known as clustering [1], [2]. More broadly, such a task is referred to as unsupervised learning because there are no explicit labels associated with the features of a dataset [2]. To elucidate this concept, consider several features that are relevant to the identification of a particular flower: petal length, petal width, stem length, and stem diameter. Different types of flowers will naturally have different features (e.g., a rose is quite different than a sunflower). If the only information that one possesses are the aforementioned measurements, the objective of a clustering algorithm for such a dataset is to determine a point or series of points that captures features for a particular type of flower. Such a trained clustering algorithm could be used to classify different flowers based only on flower features, though the kind of flower (e.g., whether it's a rose or sunflower) will not be known. Intuitively, this problem is unsupervised because features are given for different categories (or labels) of flowers, but the categories themselves may not be given.

Methods

One such method of clustering data is the (naive) k-means algorithm (aka Lloyd's algorithm) [3], [4]. This algorithm assigns N data points into K clusters. A cluster is composed typically of two parts, but a third part is included based on the specifications of OLA 4: (1) the centroid; (2) the set of data points that are closest to the centroid; and (3) a label that describes the relationship of the centroid to some category in the dataset. The centroid is simply a vector containing a single continuous value per feature in the dataset. The centroid represents mean of the set of data points that are determined during the iterations of the k-means algorithm. The addition of the label is unique to this OLA, and it is used to measure how well the clustering algorithm can performs as a supervised learning problem. The computation of the centroids is the objective of the k-means algorithm, and such a computation proceeds as below [5]:

```
k_random_centroids = random_sample_without_replacement(dataset)
cur_centroids = k_random_centroids
converged = False
while not converged:
      prev_iter_centroids = cur_iter_centroids
      cur_iter_centroids = assign_and_update_centroids(centroids, dataset)
      if prev_iter_centroids == cur_iter_centroids:
            converged = True
```

The assignment and update step of the k-means algorithm occurs by calculating the straight line (or Euclidean distance) for all data points to each of the K clusters. Data points are then assigned to a cluster and the centroid of a given cluster is subsequently updated by calculating the mean along the row axis of a N x M matrix that represents the dataset of N examples and M features. The algorithm converges when the centroids of each cluster no longer change as a result of subsequent iterations. Hence, the clusters for the n-1 and n iteration of the

algorithm are tracked in order to check whether the previous iteration's clusters and the current iteration's clusters are the same.

To validate the performance of the k-means algorithm, leave-10 repeated random subsampling cross validation was performed. The repeated random subsampling size was 100—meaning 100 different random seeds were used—and for each such subsample, the performance of the clustering algorithm was computed using the iris flower and breast cancer diagnostic dataset using several cluster sizes [6]. The cluster size range for the iris dataset was 1 to 140, while the cluster size range for the cancer dataset was 1 to 95. The ranges were determined by determining the total number of examples N for each dataset and subtracting 10 because 10 such examples are used to test the performance of the algorithm.

Results and Discussion

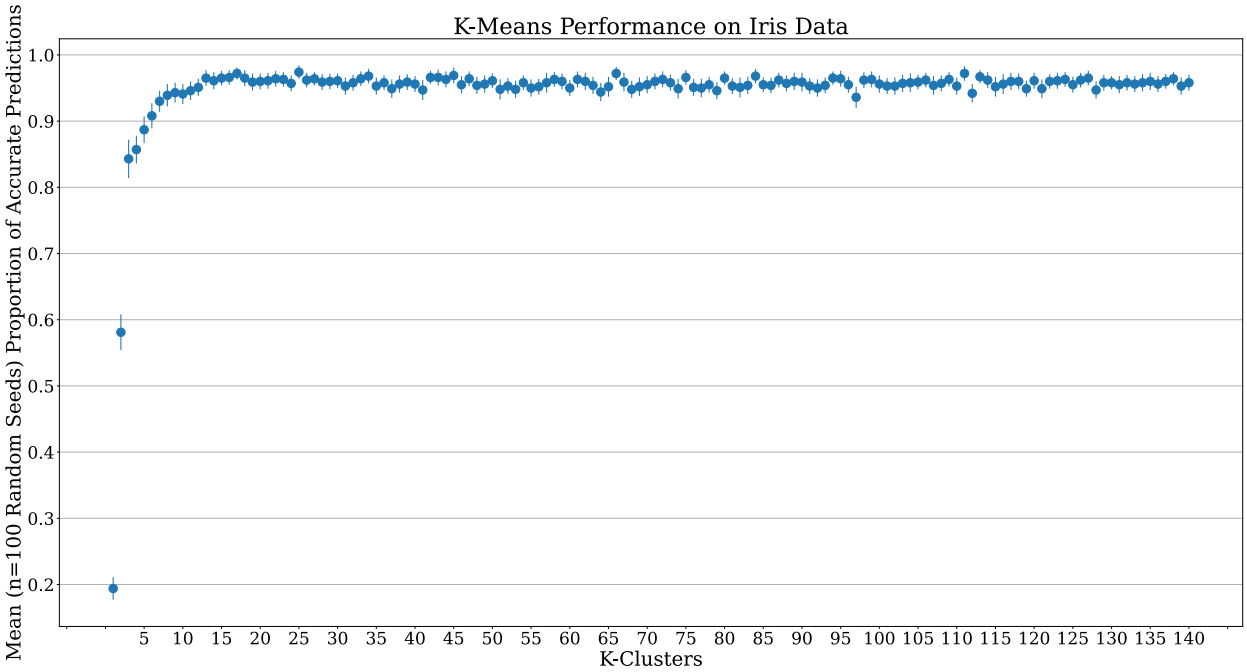

Figure 1. Performance on iris dataset. Error bars determined by 1.96 * standard error (95% confidence interval).

Figure 1 demonstrates my algorithm's performance on the iris dataset. The figure shows that as number of clusters increases, the performance on the dataset also increases. However, the trend depicted is clearly not linear but rather adopts a more logarithmic shape. Such a description is an approximation since, unlike a logarithmic function $f(x) = log(x)$ which is increasing for all x, the trend has sporadic local for K-Clusters on the interval of 20 to 140. This naturally introduces a problem for a clustering algorithm that learns a function via the generation of basis vectors (i.e., the centroids for each cluster). That problem is how many clusters should be used in order to best group data points with distinct features into discrete categories. Since the labels are given for this dataset but not used during the k-means training, one might intuit that if there are 3 labels (categories) of iris flowers, then the best number of clusters is equal to the number of

labels in the dataset. However, figure 1 shows this is clearly not the case! When there are 3 clusters, the mean percentage of accurate predictions that the trained algorithm can make on the test set is ca. 85%. While this is prediction accuracy reasonably high, it is certainly not the best performance for the iris dataset. The optimal number of clusters for the iris dataset is 25, with mean percentage of accurate predictions of ca. 97%.

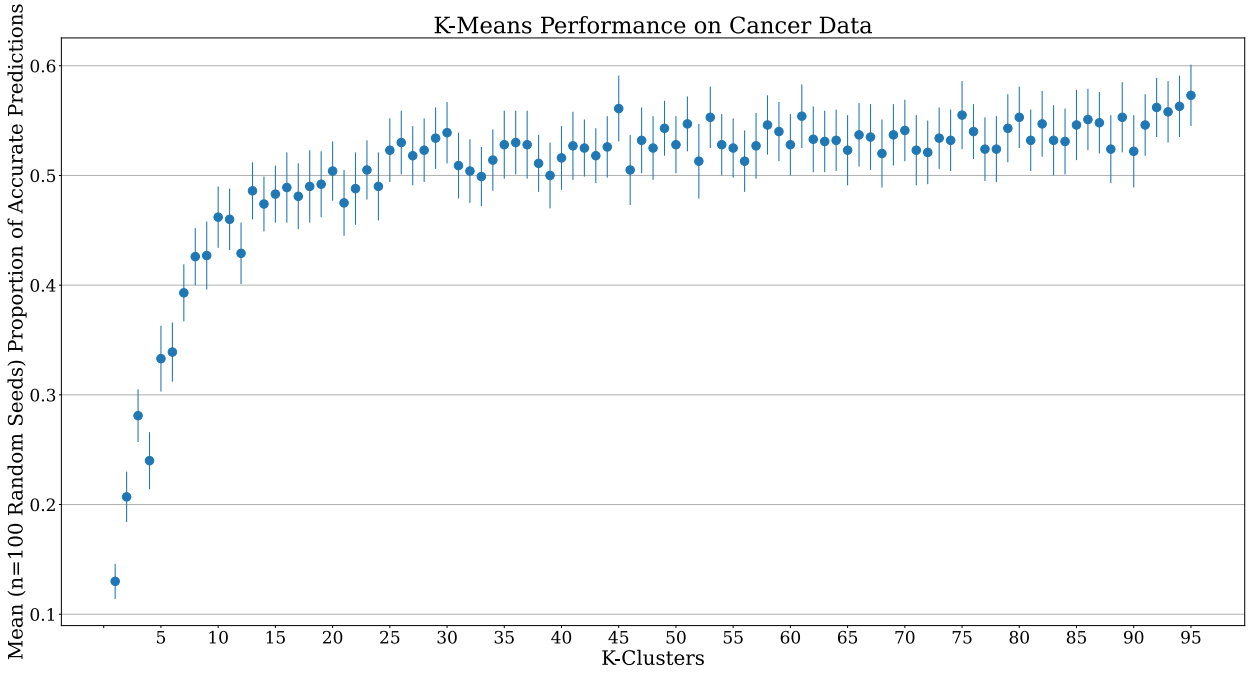

Figure 2. Performance on cancer dataset. Error bars determined by 1.96 * standard error (95% confidence interval).

Figure 2 demonstrates my algorithm's performance on the cancer dataset. Similar to the iris dataset, as the number of clusters increases the performance of the algorithm also increases. Again, the intuition for performance might be that with 6 categories of breast tissue classification, 6 clusters would lead to optimal performance; however, it is clear that this is not the case. When 6 clusters are used, the mean percentage of accurate predictions is ca. 34%, while the optimal number of clusters is 95 with prediction accuracy of ca. 57%. Naturally, overfitting is a significant problem for such unsupervised problems because as K approaches N, the cluster centroids will become the data points themselves, thus revealing no more meaningful information about features of data.

Incidentally, this by far was my most well-organized OLA, and I believe this is reflected in the codebase. The coding process went *far* smoother when I just wrote the names of methods and attributes, and then wrote the corresponding comment blocks and parameter/return types for the methods. Design and organization are such an obvious way to improve throughput, and on this particular project I genuinely enjoyed focusing on these two pillars of AI/software design.

References

[1] A. Likas, N. Vlassis, and J. J. Verbeek, “The global k-means clustering algorithm,” Pattern Recognition, vol. 36, no. 2, pp. 451–461, 2003, doi: https://doi.org/10.1016/S0031-3203(02)00060-2.

[2] Wikipedia contributors, “Unsupervised learning — Wikipedia, The Free Encyclopedia.” 2021. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Unsupervised_learning&oldid=1055916874

[3] D. Natingga, “Data Science Algorithms in a Week: Top 7 Algorithms for Computing, Data Analysis, and Machine Learning,” in Chapter 5: Clustering into K Clusters, Packt Publishing, 2017, pp. 102–108.

[4] Wikipedia contributors, “K-means clustering — Wikipedia, The Free Encyclopedia.” 2021. [Online]. Available: https://en.wikipedia.org/w/index.php?title=K-means_clustering&oldid=1056937903

[5] S. Shukla and S. Naganna, “A review on K-means data clustering approach,” International Journal of Information and Computation Technology, vol. 4, no. 17, pp. 1847–1860, 2014.

[6] D. Dua and C. Graff, “UCI Machine Learning Repository.” 2017. [Online]. Available: http://archive.ics.uci.edu/ml