Data-driven intelligent systems July 2, 2020

Florian Vahl, Dominik Buchhardt

1 The k-means algorithm

In the Initialization Phase, k random Points are generated, which represent the Centroids of the Clusters. In *digit_KMEANS.ipynb* the k is set to 10, the number of digits in base 10.
In the following Learning Phase the Data Points are assigned to the closest Centroid, the Centroids are reassigned to the means of their cluster. This Learning Phase repeats itself as long the Centroids still move (or in practice do not move more than a certain threshold as floating point numbers have limited accuracy on machines). The KMeans does not only find clusters, but the locally optimal cluster by minimizing the distance of each data point in a cluster to its centroid.
This Cluster Model can be used to assign new data points to the clusters. The *digit_KMEANS.ipynb* example assigns a digit image to the corresponding cluster which each represent a digit in base 10, thus 'recognizing' an image as a certain digit.

2 What is the best k?

The selection of k, meaning the number of classes the data is split into, is a critical and not trivial point while using k-means. The *bestK.ipynb* file generates plots for the *sum of squared distances* and the *silhouette coefficient* for each number of clusters k between 2 and 15.
The *sum of squared distances* describes the sum of the square of the distance from clusters centroid to the data point for each data point. This sum decreases if the data points are closer to their cluster centers. In the extreme case of $k = numberdatapoints$ this distance is zero. This is obviously not the k we want, therefore looking at the rate of which the *sum of squared distances* decreases seems like a better indicator. A big leap downwards can be observed at $k = 3$. The changes afterwards are very small and the clusters seem to over-fit.
The other metric is the *silhouette coefficient.* It is a value that is used to measure the quality of clustering regardless of the cluster count. A higher value is better. The peak of this plot was also at $k = 3$.
Therefore $k = 3$ seems to be the best option for this dataset.

3 Clustering comparisons

The K-Means algorithm creates two clusters on the nested circles with each cluster having half of each circle and their centroids being diametrically opposed on the inner circle. This result is not surprising as the K-Means algorithm finds clusters as the closest data points to a centroid, making the clusters circular and suitable for complex and possible nested clusters.

This behaviour can be also observed on the dataset with interlocked half moons, where each cluster claim part of the other half moon, creating incorrect categorizations. The Centroids are not on the half moons itself but on the hollow space of the moons.
In contrast to K-Means, the Agglomerative Clustering with single linkage (closest distance between two clusters) classifies the half moons correctly. This is not surprising as there is some empty space between each half moon, making the merging of clusters until two are left easy.
The Agglomerative Clustering with average (minimal average distance between two clusters) or complete (minimal maximum distance between two clusters) linkage yield the same result, with one cluster claiming a half moon and part of the other half moon. In this case it is worse than using single linkage because the number of false positives is higher, but it is still lower than using K-Means.
The DBScan algorithm yields for the half moon data set a perfect result, categorizing each half moon as its own cluster and when using more noise, classifying possible noise as noise.
In conclusion K-Means is more suited for classifying clusters with circular shapes, as circles have the minimal distance from their edges to their centers, making it unsuitable for datasets with non-circular/complex and possible nested clusters. Agglomerative Clustering merges clusters in a bottom-up approach in respect to certain distance criteria (like single, average or complete distance) between two clusters until a certain number of clusters are left. Using single linkage yields a result similar to using the DBScan algorithm for this dataset. DbScan uses a density-based approach of clustering, categorizing data points as core points, density-reachable from a core point and outliers, performing good with datasets with dense shapes/clusters even if they are complex. While DBscans performs better on these datasets, it's usage is more complex in *sklearn* than K-Means or Agglomerative Clustering, as the density needs to be set carefully to produce a satisfying result.

4 Self Organizing Maps

- What is the initialization procedure?

  The shape and size of the map are determined.

  The weights of each neuron are initialized randomly.

- How does the algorithm determine the best matching unit (BMU).

  The algorithm picks the best matching by comparing the distance between the input vector and the weight vector for each neuron and picking the neuron with the lowest distance. This distance is often calculated as a cartesian distance between the two. A "the winner takes it all" strategy is the reason why only one best matching unit is picked.

- What is the role of the neighborhood function?

  The neighborhood function describes topological relations between the neurons in the output space. If the weight of a neuron is changed (it is moved towards a specific input) other neurons that are close as described by the neighborhood function are influenced by this movement too. They are also moved in the same direction. This effect decreases over distance. Often a Gaussian function is used to model this behavior.

- What is the role of the learning rate?

  The learning rate describes how much the neurons in the neighborhood of the BMU are influenced. A high learning rate moves neurons that are further away and also moves the close ones by a larger amount.

  If the learning rate is too small the system can get stuck in local minima or be entangled. On the other hand, a large learning rate results in less accurate matches.

  The learning rate needs to be decreased while training to achieve a good macroscopic structure using a high learning rate and a small one for the fine-tuning.

- When does the algorithm stop (termination criterion)?

  The algorithm can be terminated after a certain number of iterations.