Effects of attribute noise on K-Means clustering

Petros Mitseas[1]

[1]National Centre for Scientific Research "Demokritos"

December 21, 2021

Abstract

This paper examines the effect of attribute noise when clustering with K-Means, compared to other algorithms. The performance of the algorithms is tested under three cases: unnormalized features, features with added gaussian noise and dataset containing random, uniformly distributed points. We conclude that K-Means is more sensitive than other density-based algorithms, in the presence of such noise.

Keywords
kmeans, noise, clustering, dbscan

1 Introduction

K-Means is arguably the most simple, yet fundamental algorithm used in unsupervised learning for clustering data [1]. Given an arbitrary number of clusters, the algorithm uses an iterative process during which the centroid of each cluster is calculated and the data points are assigned to the cluster with the closest centroid.

Although the algorithm is easy to implement and usually the first option when it comes to clustering, it suffers from limitations, namely suboptimality, manually choosing the number of clusters and subjection to noise. Regarding the first two issues, there have been methods to mitigate these effects such as choosing random starting centers with specific probabilities [2] and picking the number of clusters based on the Silhouette score [3]. In the following sections we emphasize on the noise aspect and compare K-Means to other popular clustering algorithms.

Attribute noise [4] refers to impurities in the data, that may happen due to inaccurate measurements, e.g. noisy or limited precision sensors, missing values or outliers. The type of noise can be Gaussian or uniform distributed. Through our experiments we can conclude that K-Means is highly sensitive to attribute noise compared to other, more resilient density-based algorithms, such as DBSCAN. Therefore special care should be given in preprocessing the data in order to filter out outliers and abnormalities, before attempting to cluster the dataset.

2 K-Means Algorithm

The algorithm aims to calculate the centroids and assign each data point to a cluster, in a manner that minimizes the sum of squared euclidean distances between the clusters' centroids and their data point. Finding the optimal solution to this problem is known to be NP-hard. K-Means uses an iterative process with polynomial complexity to determine a solution to the problem, which is not guaranteed to be optimal. However given enough iterations and using techniques such as K-Means++, the algorithm converges on sufficient solutions.

The algorithm starts with a random set of k center-points μ. During each update step, all observations x are assigned to their nearest center-point. Afterwards, the center-points are re-positioned by calculating the mean of the assigned observations to the respective center-points.

The update process reoccurs until all observations remain at the assigned center-points and therefore the center-points would not be updated anymore.

Algorithm 1 K-Means

initialize data points D
initialize cluster centroids $\mu_1, \mu_2, ..., \mu_k$
while not convergence **do**:
  for i in D **do**
    $c^{(i)} := \arg\min_j ||x^{(i)} - \mu_j||^2$
  end for
  for j in k **do**
    $\mu_j := \frac{\sum_{i=1}^{m} 1\{c^{(i)}=j\}x^{(i)}}{\sum_{i=1}^{m} 1\{c^{(i)}=j\}}$
  end for
end while

3 Attribute Noise

In the following experiments we consider three types of attribute noise.

3.1 Gaussian noise

Gaussian noise (or white noise) is a type of interference that arises often in real-world environments. The probability density function of such noise is equal to that of the normal distribution. In our experiments, in order to simulate noise, we use a random number generator that follows such distribution to generate noise values, which are added to each individual data point of the dataset.

Algorithm 2 Adding gaussian noise to data

initialize σ
generate dataset D
for d in D **do**
  draw number n from $N(0, \sigma)$
  $d \leftarrow d + n$
end for

3.2 Scaling differences

The raw features (dimensions) of the data are commonly measured in different scales. Therefore, normalizing the dataset is a core process in cluster analysis, in order to prevent features with large numerical values from dominating in distance-based objective functions [5]. Two most known methods are Standardization (Z-score Normalization) and Re-scaling (min-max normalization)

3.3 Outliers and missing values

Outliers are data that behave much differently than the majority, e.g. extreme values. These values may indicate a fault in the data collection process and should be discarded before the clustering/classification process. Otherwise the performance of these processes may be affected. Different algorithms have different sensitivity to outliers, for example DBSCAN is able to identify and ignore such values. Furthermore, methods such as K-Means– have been proposed, which aim to simultaneously cluster data and discover outliers [6].

4 Comparison with other clustering methods

We compare K-Means to other popular clustering algorithms.

4.1 DBSCAN

DBSCAN [7] (Density-based spatial clustering of applications with noise) is a clustering algorithm that groups data points together, based on their distance from their neighbour points. Unlike K-Means it doesn't use a predefined number of clusters. The algorithm is also capable of identifying outliers, e.g. points that are far apart from their neighbours.

4.2 Spectral Clustering

Spectral Clustering [8] uses the eigenvectors of matrices derived from the data, to form clusters. One benefit over K-Means is that it can create clusters of arbitrary shapes.

4.3 Gaussian Mixture Models

Gaussian mixture models implement the Expectation Maximization algorithm [9] to fit a number of Gaussian distributions to the data, under the assumption that the original data are produced by such distributions. The process is iterative: At first, the parameters of the distributions are selected at random. Then at each step the probability of each point being generated by the distribution is calculated, and the parameters are updated to maximize the log likelihood of those probabilities.

5 Experiments

In this section we present the results of the tests performed on the training dataset, for the cases of noise described in section 3.

The dataset was constructed using two-dimensional data, where each dimension is sampled from Gaussian distributions.

- Cluster 1: $X \sim N(0, 0.8), Y \sim N(0, 0.8)$
- Cluster 2: $X \sim N(-6, 1), Y \sim N(-3, 1)$
- Cluster 3: $X \sim N(2, 1.2), Y \sim N(-5, 1.2)$
- Cluster 4: $X \sim N(9, 1.6), Y \sim N(3, 1.6)$

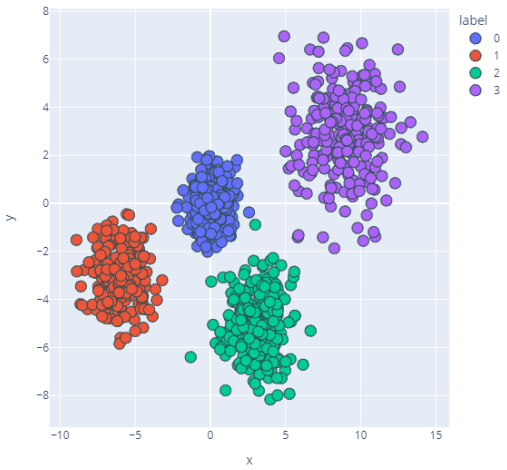

Figure 1: Dataset containing four clusters produced by sampling normal distributions.

Since we know the original cluster labels (ground truth), we can use this information to evaluate the performance of the algorithms. The metric used is the Normalized Mutual Information (NMI) [10].

5.1 Unnormalized Data / Incorrect scaling

As K-Means is based on the distance between the points and the cluster's centroid, measuring each dimension at different scale has a negative impact on the algorithm's performance. To illustrate this, we multiply one dimension by a factor of 5, and apply different clustering algorithms. It is shown that K-Means was affected the most, since the formed clusters do not resemble the actual ones.

Figure 2: Effect of unnormalized data when clustering with different algorithms.

5.2 Data with added Gaussian noise

In this experiment, we add Gaussian noise to each data point, and evaluate the performance of the algorithms. We alter the standard deviation of the noise distribution and observe the effects on the clustering process.

It is shown that K-Means, GMM and Spectral Clustering perform relatively close, in the case of added gaussian noise. DBSCAN has the worst performance, since the clusters are becoming more and more sparse, as the standard deviation of the noise increases. This leads to many points being labeled as noise.

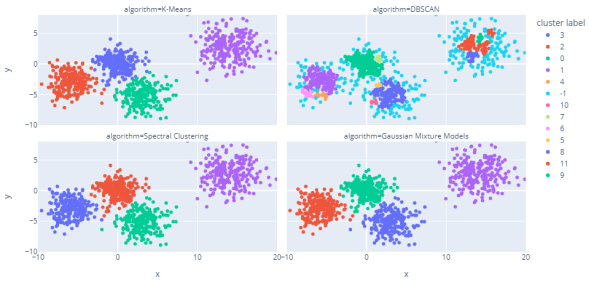

Figure 3: Clustering with added noise from Gaussian distribution $N(0, 1.0)$. DBSCAN labels many data points as outliers.

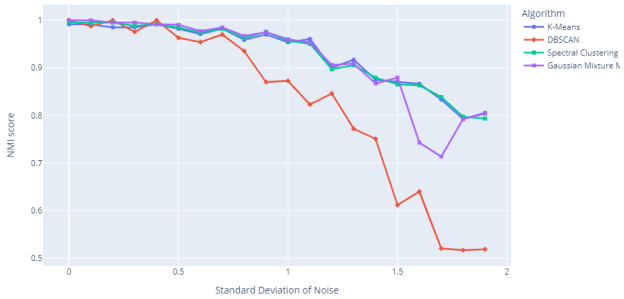

Figure 4: The Normalized Mutual Information score for different standard deviation of added noise

5.3 Dataset containing random uniform noise

This experiment considers the case where the dataset contains random noise data, uniformly distributed over the feature space. These points should be filtered out by the clustering algorithm or the data preprocessing step.

As expected DBSCAN outperformed the other algorithms, due to its ability to identify and exclude outliers. K-Means performed consistently at different noise levels, although it included the outliers in the produced clusters. GMM performed at similar levels and Spectral Clustering had the worst performance when a few noise data points were present.

In the presence of such noise, deciding the appropriate number of clusters for K-Means is challenging. A usual rule of thumb is to use the Silhouette method. However, as the count of the noise points increases, it becomes difficult to determine the value that maximizes the silhouette score.

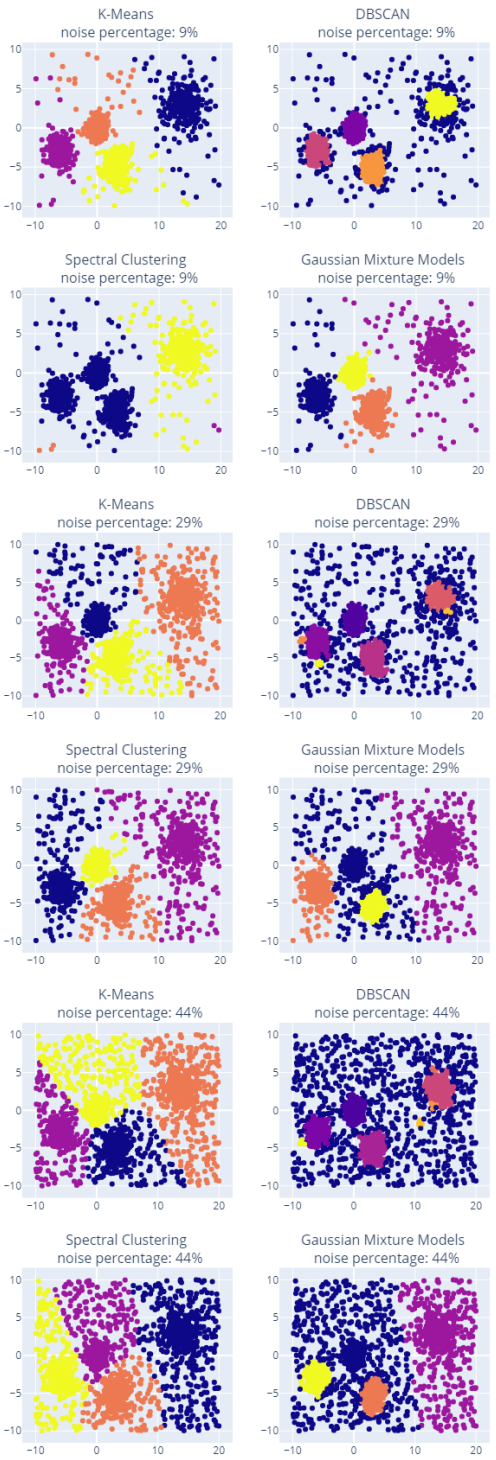

Figure 5: Clustering with different algorithms in the presence of 9%, 29%, and 44% noise, sampled from a random uniform distribution.

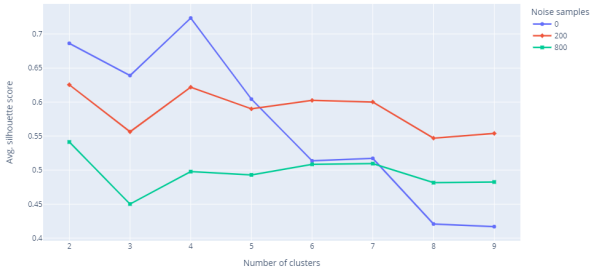

Figure 6: Avg. silhouette score for different number of clusters, in K-Means. The difficulty of finding the optimal K is increasing proportionally to the number of noise points.

Conclusions

We can conclude that K-Means is sensitive to attribute noise. When presented with unnormalized data, the algorithm was unable to form the correct clusters, compared to other density based algorithms. Furthermore, in the presence of randomly distributed noise points, we have shown that DBSCAN outperforms K-Means, due to its ability to identify outliers and exclude them from the actual clusters. The code and data used in this paper can be found at:

References

[1] James MacQueen et al. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. Oakland, CA, USA, 1967.

[2] David Arthur and Sergei Vassilvitskii. k-means++: The advantages of careful seeding. Technical report, Stanford, 2006.

[3] Peter J Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65, 1987.

[4] Xingquan Zhu and Xindong Wu. Class noise vs. attribute noise: A quantitative study. *Artificial intelligence review*, 22(3):177–210, 2004.

[5] Maria M Suarez-Alvarez, Duc-Truong Pham, Mikhail Y Prostov, and Yuriy I Prostov. Statistical approach to normalization of feature vectors and clustering of mixed datasets. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 468(2145):2630–2651, 2012.

[6] Sanjay Chawla and Aristides Gionis. k-means–: A unified approach to clustering and outlier detection. In *Proceedings of the 2013 SIAM International Conference on Data Mining*, pages 189–197. SIAM, 2013.

[7] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *kdd*, volume 96, pages 226–231, 1996.

[8] Andrew Y Ng, Michael I Jordan, and Yair Weiss. On spectral clustering: Analysis and an algorithm. In *Advances in neural information processing systems*, pages 849–856, 2002.

[9] Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1):1–22, 1977.

[10] Nguyen Xuan Vinh, Julien Epps, and James Bailey. Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance. *The Journal of Machine Learning Research*, 11:2837–2854, 2010.