An efficient k' means clustering algorithm

Žalik, Krista Rizman

Pattern recognition letters, 2008, Vol.29 (9), p.1385-1391

Contents

- Abstract and Introduction
- Cost function, data metric and membership function
- Proposed algorithm
- Experimental results and performance

Abstract

- This paper outlines an additional algorithm that can be performed upon a k-Means clustered dataset.
- This algorithm optimises the value of k to approach k' by minimising a new cost function.

Introduction

- Clustering techniques enable the identification of hidden patterns within datasets.
- Clustering has applications many areas, including data analysis, pattern recognition, image processing and information retrieval.
- k-Means clustering is a typical clustering function, due to its speed and simplicity.
- However, it requires some user input to define k, which is not guaranteed to be correct / identifiable.

Cost function

- The value of k is optimised by a new function, seeking to minimise the cost function.
- The cost function has three assumptions / characteristics:

1. Dense sampling indicates clusters.
2. A given datapoint is more likely to belong to a dense cluster.
3. Once a given centroid is driven away from the dataset / has no children, it can be ignored.

Data metric

$$dm(x_t, C_i) = ||x_t - c_i||^2 - E\log_2(p(C_i))$$

- For a given datapoint x_t and cluster C_i.
- Calculate the Euclidean distance between x_t and centroid c_i.
- Modified by the probability of x_t belonging to cluster C_i.

- The modifying term gives greater weight to clusters that have a greater density, and reduces the perceived relationship between weaker clusters and data points.
- In this way, clusters can form dominance over neighbouring clusters, and take ownership of more datapoints.

Membership function

$$I(x_t, i) = \begin{cases} 1 \quad \text{if} \quad i = \arg\min(dm(x_t, j)) \qquad j = 1, \dots, N \\ \qquad\qquad\qquad 0 \quad \text{otherwise} \end{cases}$$

- For a given datapoint x_t and cluster C_i.
- Calculate the Euclidean distance between x_t and centroid c_i.
- Modified by the probability of x_t belonging to cluster C_i.

- The modifying term gives greater weight to clusters that have a greater density, and reduces the perceived relationship between weaker clusters and data points.
- In this way, clusters can form dominance over neighbouring clusters, and take ownership of more datapoints.

Parameter E

$$E \in [a, 3a]$$

$$a = \text{average}(r) + \text{average}(d/2)$$

- E effects how strongly the modifying term “pushes” centroids away from the dataset.
- E is defined from the average radius of initial clusters, and the average shortest distance between clusters (greater than $3r$).
- E is an experimental value, and optimising it can fine-tune the performance of the function on a dataset.

k' clustering

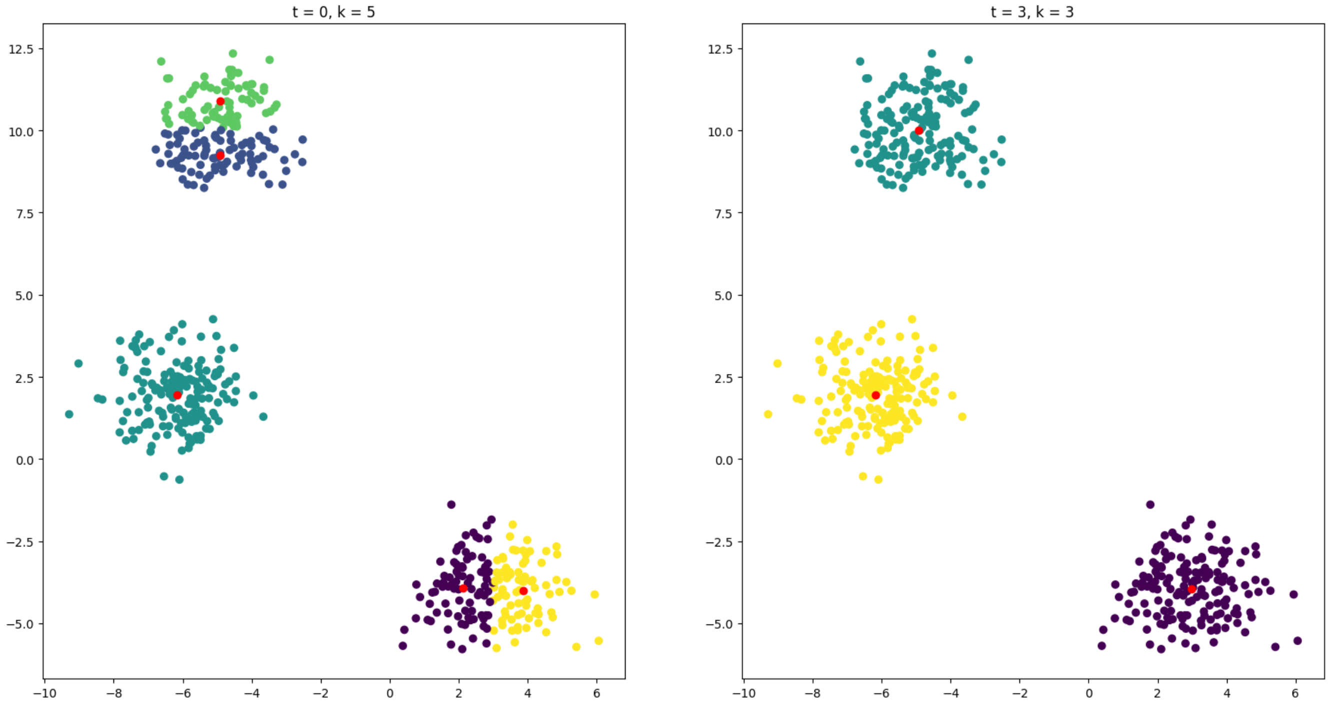
- The k'-means clustering algorithm assumes a initially clustered dataset, where $k > k'$.

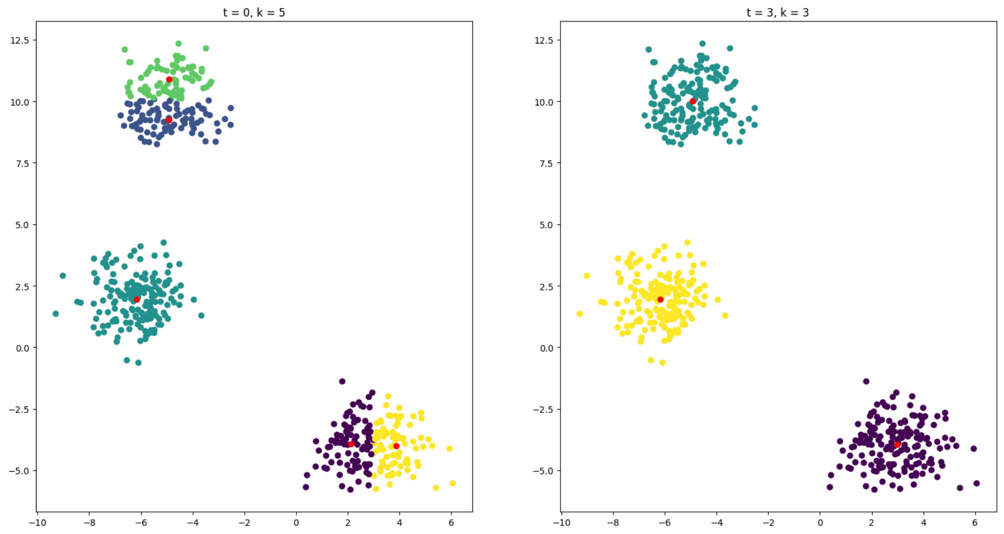
1. For each datapoint and centroid, assign clusters according to $I(x_t, i)$.
2. For all K clusters, set c_i to be the centre of mass of all points in cluster C_i.

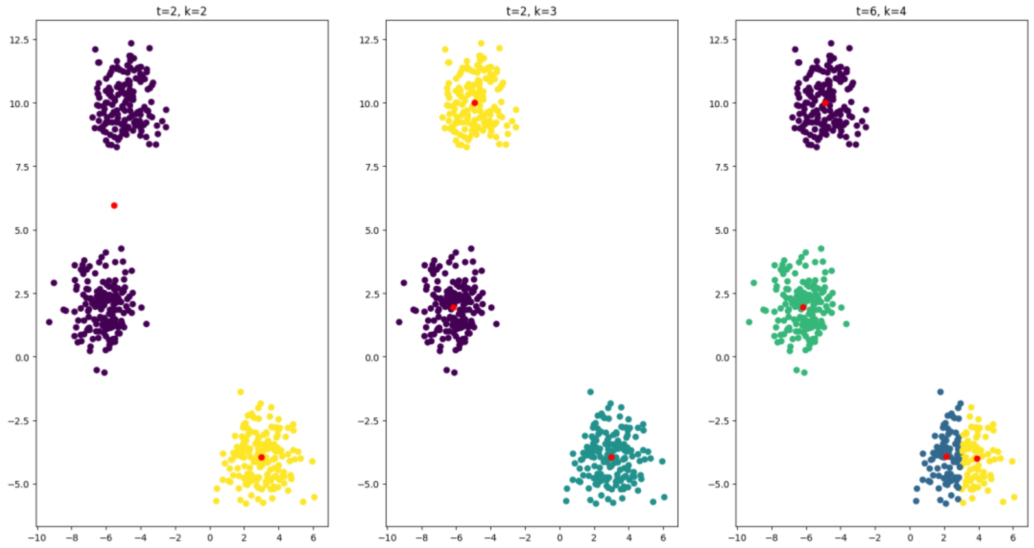
- Repeat until cluster centres remain unchanged between repetitions, or a threshold value is reached.
- Now, k' clusters have been discovered, and all additional centroids have been driven away from the dataset and discarded.

Experimental results and performance

- The proposed algorithm was implemented in Python and tested on a dataset.
- The efficiency of this algorithm was compared to a brute-force method of finding k'.







Dataset:

- $k' = 3$
- N_samples $= 500$

k' clustering:

- $t = 3$ to identify $k' = 3$

k clustering:

- $t = 2 + 3 + 4 = 9$ to identify $k' = 3$

Conclusions

- The proposed algorithm performs well in identifying k' efficiently.
- However, our implementation encountered reproducibility issues, as the algorithm proved highly sensitive to initial conditions, dependent on initial centroids.
- In testing, our implementation predicted $k' = \{1, 3\}$ with a ratio $2:1$.
- This occurred due to the approximation of E, presenting an optimisation problem.