Song Yang (sy540)
Xin Yang (xy213)
Yi Wu (yw641)
Zhuohang Li (zl299)

Term Paper Pre-report of Data Structure and Algorithms

Analysis of k-means Clustering Algorithm

Introduction

K-means algorithm is a basic clustering algorithm. It's widely used in data mining and machine learning. A lot of optimizing algorithms are available for k-means and can also be included in this project.

Science and Engineering Problem Motivation

K-means algorithm is of great significance in machine learning, especially the unsupervised learning field. Both science and industries implement this algorithm in cases such as clients segmentation and recommendation systems. Since the use cases are closely related to massive data processing, choosing the value of k can crucially vary the performance. Besides, when it comes to non-convex problems, k-means algorithms can be incapable. Thus optimize the performance and make the scope of application more widely, it's necessary to go through this algorithm and find out proper methods.

Description and Explanation of the Algorithm

K-means clustering is a typical unsupervised learning algorithm that aiming at classifying unlabeled data into k clusters. The algorithm works in an iterative way described as follows:

1. Choose k arbitrary points as centroids.

2. Assign every data points to their nearest centroids.

3. Recalculate the centroids.

4. Reassign every data point to its nearest centroid.

5. Iterate from step 2 until no point is reassigned in step 4.

Experimental Configurations and Details.

We will design experimental programs to analyze and verify the efficiency of the algorithm. As k-means is useful in solving clustering problems, we are going to build a classification case. There are variations of implementations that can have impacts on the performance.

As mathematical vector operations are required for this algorithm, we might choose python as our programming language. Thanks to the third-party libraries like Numpy, it's easier to perform matrix computations. As well Python is a commonly acknowledged tool in machine learning and pattern recognition.

Here we will mostly put our efforts on scalar data since the data type varies in the actual environment. If time permits, we might test vector cases as well(using cosine measurement as distance). For scalars, we can regard Euclidean distance, Manhattan distance, and Pearson Correlation Coefficient as dissimilarity degree.