
# Project Title: A Comparative Analysis of Gaussian Mixture Model and K-means Clustering Algorithms With Applications in Time-of-Use (TOU) Clustering

**Project Module Code: EG2605**

**Student Name: Huang Xiao**

**Student Number: A0244823E**

**Dept Name: Department of Electrical & Computer Engineering, NUS**

**Project link:** https://github.com/AlbertHX86/UROP.git

**AY2023/2024**

# Abstract

Clustering algorithms are powerful unsupervised machine learning techniques used to group similar data points together based on their attributes and features. Clustering algorithms also have many application scenarios, such as the rise in popularity of electric vehicles and smart grids, where the utilization patterns of electric vehicle users in smart grids and consumer behaviour become topics that need to be enhanced in pricing models [1]. In order to classify user behaviour, clustering can be applied to the raw charging data to find the charging behaviour of different users.

In this study, we present a comprehensive comparison between two widely-used clustering algorithms, Gaussian Mixture Model (GMM) and K-means, in terms of their accuracy, time and space complexity, and suitability for various application scenarios. Specifically, we evaluate the accuracy of both algorithms using two well-established metrics, the Davies-Bouldin Index (DBI) and Error Rate, to provide a quantitative assessment of their performance. Our analysis further investigates the time and space complexity of GMM and K-means, offering insights into their efficiency and scalability.

To demonstrate the practical implications of these algorithms, we applied both GMM and K-means to a real-world dataset of electric vehicle (EV) charging data from the Colorado Boulder workspace. Our objective was to uncover hidden trends, identify peak charging hours, and analyze customers' behaviours. The results from both algorithms provide valuable insights into the charging patterns and user preferences, with each method highlighting specific aspects of the data.

# Preface and Acknowledgements

Embarking on my first formal research journey at the university, I am profoundly grateful to my advisor, Prof. Xiang Cheng, for his unwavering support and guidance throughout this endeavour. I vividly recall the numerous challenges I faced in the project's initial stages, stemming from my inexperience and the absence of a robust methodology. Yet, it was Prof. Xiang Cheng's steadfast mentorship and expertise that enabled me to grow and mature as a researcher during this year-long opportunity. The invaluable counsel I received from him will forever serve as a guiding principle in my future research endeavours: to truly comprehend what I know and to understand the rationale behind it.

Furthermore, I extend my heartfelt gratitude to the National University of Singapore (NUS) for offering undergraduates an unparalleled research experience that bridges the gap between theoretical knowledge and practical application. This exposure has not only enriched our academic pursuits but also laid a solid foundation for our future growth and development. I am truly appreciative of the opportunities provided and the insights gained, which will undoubtedly shape my professional trajectory in the years to come.

## DECLARATION

I hereby declare that the project is my original work and it has been written by me in its entirety. I have duly acknowledged all the sources of information which have been used in the project.

*Xiao Huang*

# Contents

# 1 Introduction

## 1.1 Introduction to GMM and Kmeans

Common types of clustering include partitioning-based methods, such as K-Means clustering, which makes attempts to minimize the within-cluster sum of squares, and model-based clustering (MBC), such as GMM, which belongs to the family of generative probabilistic model-based clustering and assigns a probability score to each data point for each cluster, indicating the degree of belonging to a particular cluster. Each type of clustering algorithm has its strengths and weaknesses in different clustering scenarios.

Gaussian Mixture Model (GMM) and K-means are two widely-used clustering algorithms in machine learning and data mining. Both methods aim to partition a dataset into distinct groups (clusters) based on similarity or distance measures [2]. However, they are based on different underlying principles and assumptions, leading to varying performance and suitability for specific problem domains.

## 1.2 Related Works

As the two most prevalent clustering algorithms, K-Means and GMM have been utilized extensively across many disciplines. However, choosing between the two clustering methods for different applications has been a challenge due to disparities in their implementation methods and clustering results. GMM, which is a type of Bayesian clustering, inherits from the fundamental principles of Bayesian clustering [2]. The introduction of AutoClass as a clustering technique in the 1990s moved Bayesian clustering to the forefront of discussion due to its simplicity and precision in application [3]. AutoClass has been extensively implemented in the intrusion detection, biological information clustering, and GIS clustering, and has produced favourable outcomes [3, 4]. Even though the AutoClass application is no longer accessible, GMM can still be accessed through the MATLAB toolkits. Because the implementation principles of the two methods are distinct, the commonly employed evaluation indicators, such as BIC and SI, cannot be applied directly to the evaluation, and additional experiments are required for comparison [2].

Time-of-Use rates (TOU) on distribution power grids are extensively discussed in smart grids because they can effectively balance peak load and the pricing of the electricity rates[1]. In addition, electric vehicle charging stations are a significant component of smart infrastructure. The operation of the electricity grid is impacted by the daily actions of users. Currently, the pricing model of researchers primarily incorporates the demand situation of the real-time power network as well as the real-time cost and price into the dynamic pricing model, and uses system engineering simulation for modeling. User behavior can be further incorporated into the model in future

model enhancements [5].

## 1.3 Report Organization

The remainder of this report is organized as follows:Chapter 2 presents the process of a novel clustering algorithm's development based on Bayes' rule and conditional probability. Together, their application to a mix of 1D Gaussian distributions In the section, it is shown that the algorithm works notably well at clustering the dataset. The results derived from the experiments show that model-based clustering algorithms have advantages in clustering distributed data following calculable distributions.

As research goes on further, The clustering model based on the normal distribution in the above article was found to be a special case of the GMM model in the research. When creating a GMM model, in order to apply it to more complex data, such as high-dimensional data and data consisting of a mix, to multiple models, such as the GMM model, which is more complex compared to our 1D model. However, in the case of GMM clustering of low-dimensional data sets, the coefficients in the GMM model often affect the clustering results, resulting in weaker clustering results with the GMM than with our simple 1D clustering method. In Chapter 2, the comparison of the GMM and K-Means algorithms is described, and evaluation metrics such as DBI are introduced as a basis for evaluating the performance of the two algorithms. Additionally, based on a breast cancer data [6], GMM and K-Means are applied to better illustrate the differences in clustering data with different dimensions and complexities in features.

In Chapter 3, based on the public data of Colorado Boulder's workplace EV charging data [7], GMM and Kmeans are performed to do temporal clustering to identify the peak charging hours and users' behaviors during the charging period. Followed with discussions about applications in TOU pricing model developments in Chapter 4.

# 2 Comparative Experiments With the Clustering Algorithms

## 2.1 K-Means

K-Means is a commonly used unsupervised learning algorithm that focuses on clustering data sets using iterative solving. The steps are: pre-dividing the data into K groups, then randomly selecting K objects as the initial cluster centers, then calculating the distance between each object and each seed cluster center, and assigning each object to the cluster center closest to it. The clustering centers and the objects assigned to them represent a cluster [2]. A common process of K-Means is described as follow:

1. Specify the number of clusters K
2. Initialize centroids by first shuffling the dataset and then randomly selecting K data points for the centroids without replacement
3. Keep iterating until there is no change in the selectiojn of centroids. i.e assignment of data points to clusters isn't changing
   (a) Compute the sum of the squared distance between data points and all centroids.
   (b) Assign each data point to the closest cluster.
   (c) Compute the centroids for the clusters by taking the average of all data points that belong to each cluster

The process explained is a normal Expectation Maximization (E-M) method of clustering [8]. Within the process, the E-steps show assigning of the data points to the closest cluster, and the M-step represents the re-computing and iteration of finding the centers of each cluster. A mathematical explanation to the process as be found as follows:

$$J = \sum_{i=1}^{m} \sum_{k=1}^{K} w_{ik} \left\| x^i - \mu_k \right\|^2 \tag{1}$$

K-means presupposes that clusters are spherical and of equal dimensions, according to the K-means experiment results . As the noise in the data becomes smaller, then the spherical nature of the data becomes more obvious, and K-Means' results become much better than when the sigma is large. In addition, K-means assigns data points to a single cluster based on their proximity to the centroid, resulting in difficult assignments. In situations where data elements belong to multiple clusters with differing degrees of membership, this may not be optimal. Updating the hard assignment models with probabilistic models that employ soft assignments to cluster may therefore help resolve the issues.

| | Sigma_0_2... | Sigma_0_2... | Sigma_0_4... | Sigma_0_4... | Sigma_0_6... | Sigma_0_6... | Sigma_0_8... | Sigma_0_8... | Sigma_1_0... | Sigma_1_0... |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | -0.0220 | 0.5057 | -0.1476 | 0.6216 | -0.2914 | 0.7788 | -0.3478 | 1.0408 | -0.4677 | 1.2600 |
| 2 | -0.0220 | 0.5057 | -0.1476 | 0.6216 | -0.2914 | 0.7788 | -0.3868 | 0.9962 | -0.4677 | 1.2600 |
| 3 | -0.0220 | 0.5057 | -0.1476 | 0.6216 | -0.2914 | 0.7788 | -0.3478 | 1.0408 | -0.4677 | 1.2600 |
| 4 | -0.0220 | 0.5057 | -0.1476 | 0.6216 | -0.2914 | 0.7788 | -0.5116 | 0.8844 | -0.7192 | 1.0102 |
| 5 | -0.0220 | 0.5057 | -0.1476 | 0.6216 | -0.2914 | 0.7788 | -0.5116 | 0.8844 | -0.7192 | 1.0102 |
| 6 | -0.0220 | 0.5057 | -0.1476 | 0.6216 | -0.2914 | 0.7788 | -0.3478 | 1.0408 | -0.4677 | 1.2600 |
| 7 | -0.0220 | 0.5057 | -0.1476 | 0.6216 | -0.2914 | 0.7788 | -0.5116 | 0.8844 | -0.4677 | 1.2600 |
| 8 | -0.0220 | 0.5057 | -0.1476 | 0.6216 | -0.2914 | 0.7788 | -0.3478 | 1.0408 | -0.7192 | 1.0102 |

Figure 1: K-Means clustering results with varying noises

## 2.2 Algorithm development

### 2.2.1 1-D Gaussian Model-Based Clustering Algorithm Development

**Bayes' Rule** Bayes' Rule is used to describe the relationship between two conditional probabilities.

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \tag{2}$$

where A and B are events and P(B)$\neq$ 0

**Bayesian Classifier** In contrast to assigning data into a certain cluster or category, the Bayes' Rule-based clustering algorithm clusters through calculation. Suppose there are k different clusters, with a given data point X. The cluster is the one in which the data has the largest rate. i.e:

$$if\ P(C_1|X) > P(C_2|X)\ then\ assign\ X\ to\ C_1 \tag{3}$$

Additionally, the probability of each of the clusters is supposed to be the same. i.e: P($C_1$)=P($C_2$). Therefore, in order to find P(A—Z=z), given the value of Z=z, the probability that it is an element of cluster A can be derived as:

$$P(A|Z=z) = \frac{P(Z=z|A)P(A)}{P(Z=z|A)P(A)+P(Z=z|B)P(B)} \tag{4}$$

Note that P(Z=z—A)=P(X=z), for small $\varepsilon$,

$$P(z<X<z+\varepsilon) \approx \varepsilon f_x(z) \tag{5}$$

where $f_x$ is the density of X. Suppose all the clusters are following Gaussian distribution

$N(\mu, \sigma^2)$ with probability density function as:

$$\frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}(\frac{x-\mu}{\sigma})^2} \tag{6}$$

Therefore, P(A—Z=z) can be further expresses as:

$$P(A|Z=z) = \frac{f_X(z)p}{f_X(z)p + f_Y(z)(1-p)} \tag{7}$$

as discussed in the previous sections, the probability of occurrence of each cluster should be the same. Thus, the equation can be further simplified to:

$$P(C1|X) = \frac{e^{-\frac{1}{2}(\frac{x-\mu_1}{\sigma})^2}}{e^{-\frac{1}{2}(\frac{x-\mu_1}{\sigma})^2} + e^{-\frac{1}{2}(\frac{x-\mu_2}{\sigma})^2}} \tag{8}$$

Therefore, the iteration process of labelling the data is changed from labelling in numbers (i.e 0 and 1) to using the conditional probability calculated from (10). The detailed implementation procedures are:

1. Determine the number of clusters K and the parameters used in the algorithm.

```
function [mu1,mu2]=p_method(data,sigma) %input the parameters to
   the algorithm
```

2. Start the algorithm and update the labelling in the data to the conditional probability calculated using equation (8).

```
        for i=1:m

            PZA=(exp(-(cluster(i)-mu1)^2/(2*sigma^2)))/
                ((exp(-(cluster(i)-mu1)^2/(2*sigma^2)))
                +exp(-(cluster(i)-mu2)^2/(2*sigma^2)));
            PZAset(end+1)=PZA;

            PZB=(exp(-(cluster(i)-mu2)^2/(2*sigma^2)))/
                ((exp(-(cluster(i)-mu2)^2/(2*sigma^2)))
                +exp(-(cluster(i)-mu1)^2/(2*sigma^2)));
            PZBset(end+1)=PZB;
        end
```

3. Iterate the algorithm until it converges.

```
% the max_iteration numbers is set to 1000 by default, and this
   can be updated according to data sizes
    while iter<1000
        p1new=zeros(1,m);
        p2new=zeros(1,m);
        PZAset=[];
        PZBset=[];
        cluster1=[];
        cluster2=[];
        % re-selection of the labels
        iter=iter+1;
        result1=0;
        result2=0;
        for j=1:m
            result1=result1+PZAset(j)*cluster(j);
            result2=result2+PZBset(j)*cluster(j);
        end
        mu1=result1/sum(PZAset);
        mu2=result2/sum(PZBset);
    end
```

The implementation of the code is shown in the appendix A and the following comparative experiments are made based on the algorithms described above.

### 2.2.2 Comparison of 1D Gaussian model-based clustering algorithm with K-Means

The performance of the experiments was assessed by cross-comparing the results of clustering from two algorithms: the built-in K-Means in the MATLAB tools kit, re-written K-Means, and the model-based 1-D clustering algorithm. In the experiments, the same dataset was used for all three methods (with identical sigma), and fifty trials were conducted accordingly. The experiments mix two groups of data with a Gaussian distribution, N(0, $1^2$) and N(0.5,$1^2$), each containing 100 data points (theoretically K = 2). The two groups of data were combined and clustered using the three methods described above, and the centers of the clustered data were observed.

The clustering result in the appendix C: shows an example clustering result based on the two algorithms. As demonstrated by the results, the model-based clustering algorithm works better in clustering 1D data using soft assignment, and it is more accurate and robust than K-Means.

Therefore, it is advantageous to generalize the idea to higher dimensional cases so that it can be utilized in practical settings.

## 2.3 GMM

Gaussian Mixture Models (GMM) are probabilistic models for representing complex data distributions that are composed of multiple subpopulations or clusters. GMM assumes that the data is generated from a mixture of several Gaussian distributions with different means and variances. Each Gaussian distribution corresponds to a different cluster or subpopulation within the data. The GMM then models the probability density function (PDF) of the data as a weighted sum of the PDFs of the individual Gaussian distributions [2]. Compared with the probability density function for 1D Gaussian distributions, the PDF of multivariate Gaussian distributions is as follows:

$$f(\mathbf{x}|\mu,\Sigma) = \frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x}-\mu)^{\top}\Sigma^{-1}(\mathbf{x}-\mu)\right) \tag{9}$$

in which the covariance matrix is:

$$\Sigma = \begin{bmatrix} \sigma_1^2 & \sigma_{1,2} & \cdots & \sigma_{1,d} \\ \sigma_{2,1} & \sigma_2^2 & \cdots & \sigma_{2,d} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{d,1} & \sigma_{d,2} & \cdots & \sigma_d^2 \end{bmatrix} \tag{10}$$

In the Gaussian mixture model, the probability for a sample from a given distribution is:

$$f(\mathbf{x}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}|\mu_k,\Sigma_k) \tag{11}$$

where $\mathbf{x}$ is a data point, $K$ is the number of clusters or components in the GMM, $\pi_k$ is the weight or mixing coefficient of the $k$-th Gaussian distribution (with $\sum_{k=1}^{K}\pi_k = 1$), $\mathcal{N}(\mathbf{x}|\mu_k,\Sigma_k)$ is the Gaussian distribution with mean $\mu_k$ and covariance matrix $\Sigma_k$, and $|\Sigma_k|$ denotes the determinant of the covariance matrix.

The likelihood of the GMM given a set of data points $\mathbf{x}_1,\mathbf{x}_2,\ldots,\mathbf{x}_N$ is given by:

$$L(\theta|\mathbf{x}n) = \prod n = 1^N \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}_n|\mu_k,\Sigma_k) \tag{12}$$

where $\theta = \mu_1,\ldots,\mu_K,\Sigma_1,\ldots,\Sigma_K,\pi_1,\ldots,\pi_K$ are the parameters of the GMM.

The goal of fitting a GMM to data is to find the parameters $\theta$ that maximize the likelihood $L(\theta|\mathbf{x}_n)$. This can be done using the Expectation-Maximization (EM) algorithm, which alternates

between computing the expected responsibilities of each component for each data point (the E-step) and updating the parameters based on these responsibilities (the M-step).

### 2.3.1 Comparison of the 1D model-based algorithm with GMM

For Gaussian Mixture Models in 1D spaces, the covariance matrix is still considered in the algorithm, but it is reduced to a scalar value (variance). In the case of 1D spaces, the GMM algorithm estimates the parameters of each Gaussian component to model the underlying probability distribution of the data. Consequently, it can be concluded that the 1D model-based clustering algorithm can be considered as a special case of GMM in 1-D clustering. A detailed experiment result comparing the performances of 1-D model-based clustering, GMM and K-Means is shown in the appendix Comparison.

The only distinction in the implementation of the 1D model-based clustering algorithm discussed in the previous section is that the sigma was presumed to be known and input into the algorithm. In contrast, GMM estimates the sigma, which may not be as accurate as explicitly entering the parameters. In 1D cases, GMM does not function as well as the 1D algorithm for this reason. As it is nearly impossible to know the parameters in real-world applications, the estimation phase in GMM is required when putting the model-based algorithm into practice.

## 2.4 Comparison between GMM and K-Means implementation in MATLAB

### 2.4.1 Davies-Bouldin Index

In order to evaluate the quality and performance of clustering algorithms, Davies-Bouldin Index is used in the experiments [9].

$$DBI = \frac{1}{k}\sum_{i=1}^{k}\max_{i\neq j}\left(\frac{S_i + S_j}{d_{ij}}\right), \tag{13}$$

where $k$ denotes the total number of clusters, $S_i$ and $S_j$ represent the average intra-cluster distances of clusters $i$ and $j$ respectively, and $d_{ij}$ is the inter-cluster distance between clusters $i$ and $j$. The average intra-cluster distance, $S_i$, can be computed as:

$$S_i = \frac{1}{n_i}\sum_{x\in C_i} d(x, c_i), \tag{14}$$

where $n_i$ is the number of data points in cluster $C_i$, $x$ is a data point within the cluster, and $c_i$ is the centroid of cluster $i$. The inter-cluster distance, $d_{ij}$, is typically measured as the Euclidean distance between the centroids of the respective clusters:

$$d_{ij} = |c_i - c_j|. \tag{15}$$

DBI score ranges from 0 to infinity, with a lower score signifying better clustering quality, meaning that corresponds to higher intra-cluster compactness [2].

Applying the same dataset as introduced in the previous experiments, which has two clustering centroids 0 and 0.5. The error rate and DBI score for both algorithms with sigmas ranging from 0.1 to 1 can be seen in the graphs.

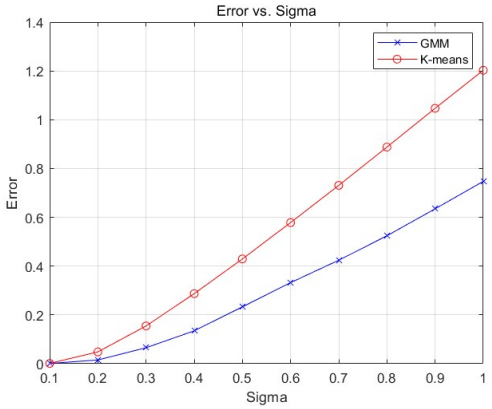



Figure 2: Error Rate

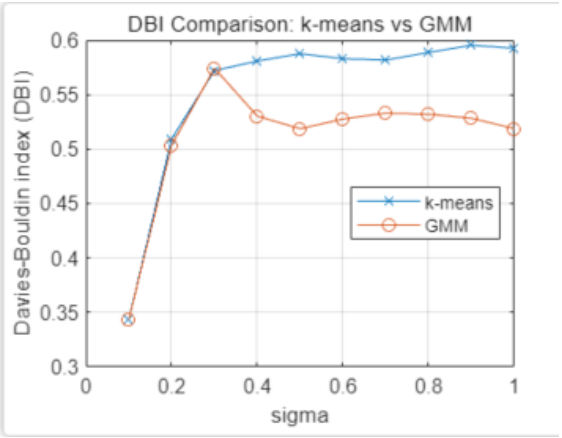



Figure 3: DBI Calculation

With basic data constructed using the Gaussian mixture model, it can be concluded that GMM performs better than K-Means in identifying clusters.

However, it is notable that GMM requires more iterations as the magnitude of the data increases. As the quantity of the data increases from 200 to 1000, 3000 iterations may no longer be sufficient for convergence; 10000 iterations are required for GMM to converge, resulting in a significant increase in running time.

### 2.4.2 Running time, Time and Space Complexity

Assume n is the number of data points, k is the number of clusters, I is the number of iterations, and d is the number of dimensions. The k-means algorithm has a time complexity of O(n * k * I * d), the space complexity of the GMM algorithm is O(n * d + k * $d^2$). GMM has a time complexity of O(n * k * 1 * $d^2$) and the space complexity of the GMM algorithm is O(n * d + k * $d^2$) [10].

Apply GMM and K-Means on the same 1000-data dataset, and vary the dimension (features) of the dataset, the run-time for GMM and K-Means can be seen in the following graph. As the dimensionality of the dataset increases, the volume of the space grows exponentially, causing the data points to become increasingly sparse. This sparsity makes it difficult for GMM to accurately

estimate the underlying probability distributions, especially the covariance matrices, which may lead to Curse of dimensionality [10], and result in poor performance.

GMM is a more flexible model compared to k-means, as it can model more complex, non-spherical clusters. However, this flexibility can also lead to overfitting, particularly in high-dimensional spaces with a limited number of data points. Overfitting occurs when the model captures noise or random fluctuations in the data, leading to a poor generalization to unseen data [10].

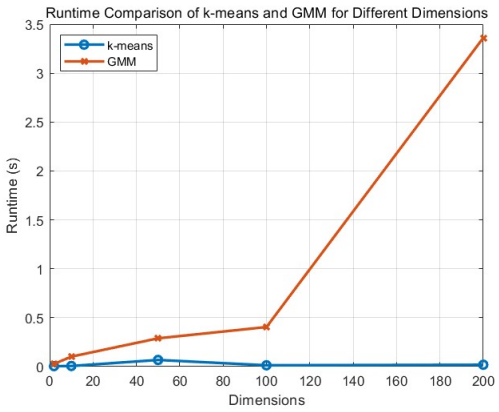



Figure 4: Runtime Comparison

Therefore, for clustering missions with large sizes and high dimensions, it is necessary to consider between two algorithms or use dimension deduction methods for GMM clustering to ensure maximized accuracy.

### 2.4.3 Different Shapes

In the following experiments, we compared the performance of GMM and K-Means in clustering data with different shapes. By applying DBI to evaluate the performances of GMM and K-Means on 50 distinct Gaussian distributed data and spherical data. It can be noticed that GMM works better in clustering Gaussian distributed data, whereas K-Means works better in spherical data. An example comparison result is shown below:

| | Dataset | KMeans DBI | GMM DBI |
|---|---|---|---|
| 0 | Gaussian 1 | 0.922116 | 0.713710 |
| 1 | Spherical 1 | 0.886994 | 0.891077 |

Figure 5: GMM and K-Means with different shapes

Breast Cancer Wisconsin (Diagnostic): contains information about breast cancer tumors diagnosed in patients in Wisconsin, USA [6]. The data consists of 569 samples, each with 30 numerical features representing characteristics of the tumor, such as radius, texture, and area. The data contains much more complex mixture of data with different shapes and distributions, and the large size and dimensions might be a challenge to both algorithms [6].

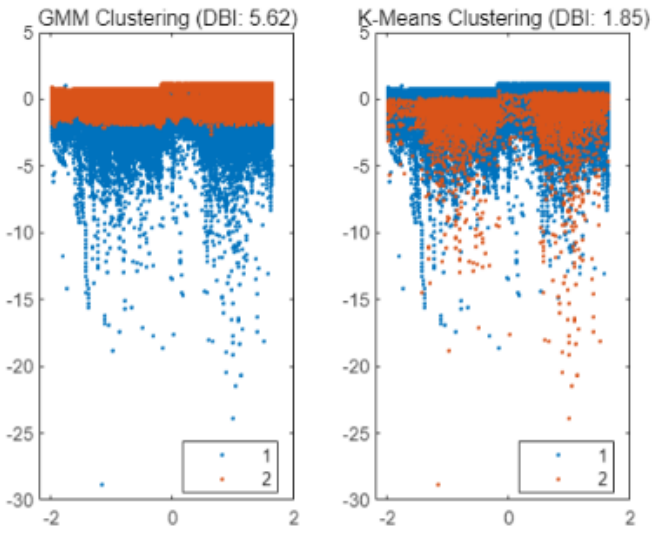



Figure 6: GMM and K-Means with complex data

# 3 Temporal Clustering of EV charging data

## 3.1 Introduction to the purpose and the dataset

Electric vehicles are becoming a major type of vehicle in many countries recently, and the availability of large-scale electric vehicle charging loads has posed new challenges for the power grid [5]. Peak domestic electricity demand coincides with the EV charging load, which is problematic because it can easily overwhelm grid components, generate voltage fluctuations, increase line loss, and exacerbate the three-phase imbalance in the distribution network [5].

Based on the distribution of charging hours and the corresponding energy consumption, Temporal Clustering can be done to the data, in order to identify the charging patterns. So as to identify the peak periods and plan to avoid overloading in the EV charging network [1]. The selected data involves the data from public chargers in Colorado Boulder city, showing the energy use, length of charging time, gasoline savings and greenhouse gas emission reductions from all city-owned electric vehicles. The overall size of the data is 43659 [7].

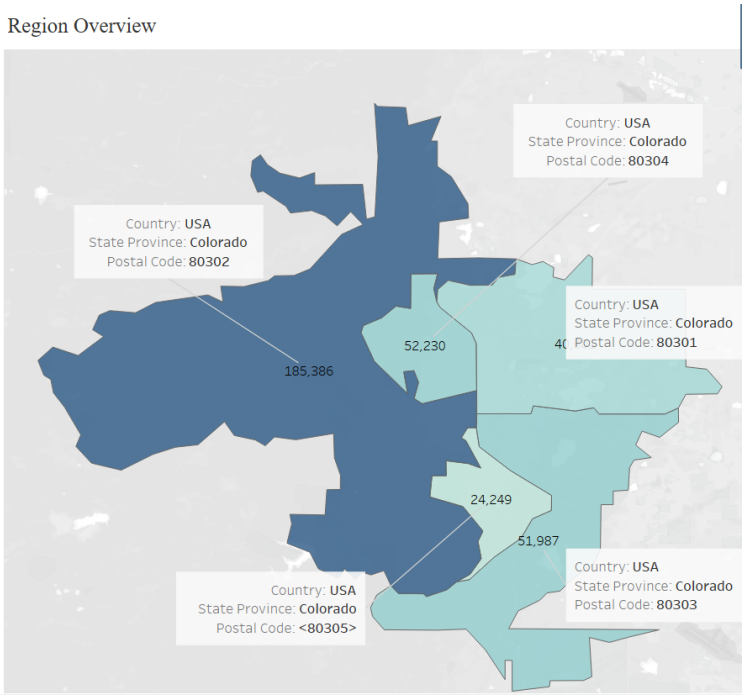



Figure 7: Region Overview

## 3.2 Data Manipulation

Over 40,000 data points may make the computational complexity extremely large, and result in poor performance in algorithms.

| | Energy__kWh_ | time_only_start | time_only_end | Total_Duration_seconds | Charging_Time_seconds |
|---|---|---|---|---|---|
| 0 | 4.608 | 07:29:00 | 09:22:00 | 6779 | 6761 |
| 1 | 8.786 | 11:42:00 | 15:27:00 | 13539 | 8535 |
| 2 | 4.940 | 11:51:00 | 12:54:00 | 3781 | 3770 |
| 3 | 0.000 | 13:55:00 | 13:57:00 | 120 | 0 |
| 4 | 3.023 | 14:54:00 | 15:52:00 | 3513 | 3505 |
| ... | ... | ... | ... | ... | ... |
| 43654 | 2.401 | 18:51:00 | 21:31:00 | 9598 | 9569 |
| 43655 | 6.587 | 19:08:00 | 20:14:00 | 3987 | 3963 |
| 43656 | 7.130 | 19:25:00 | 20:38:00 | 4370 | 4241 |
| 43657 | 11.828 | 21:44:00 | 23:49:00 | 7466 | 7450 |
| 43658 | 14.048 | 23:39:00 | 11:10:00 | 41484 | 8427 |

43659 rows × 5 columns

Figure 8: Original Data

1. Reduce the dimension of the dataset, and only remain the columns pertinent to the objective: After clustering, the most common charging time and charging period will be identified, as well as the amount of energy will also be identified.

2. Convert the start time to seconds after midnight, so that K-Means and GMM can be used with our further changes. The starting time was set to 0:00 am and the subsequent time is displayed in seconds after the origin point.

3. For people who plug in the charger but did not charge are also enumerated; therefore, those rows with 0 energy consumption are removed.

4. Over 40,000 data points may make the computational complexity extremely large, and result in poor performance in algorithms. Random sampling is used in the data to collect 5000 data points so that the size will be optimal for clustering. The random state is set to 42.

5. Shapiro-Wilk tests are done on the data to identify the distribution of the data and predict the algorithm to use in the process [11]. It can be identified that the data may not be a perfect Gaussian mixture model, therefore GMM may not be the best algorithm to give the optimal result if no further manipulations are done.

```
Shapiro-Wilk Test for Energy__kWh_: W-statistic = 0.7932
Shapiro-Wilk Test for Total_Duration_seconds: W-statistic = 0.1324
Shapiro-Wilk Test for Charging_Time_seconds: W-statistic = 0.8184
Shapiro-Wilk Test for start: W-statistic = 0.9862
```

Figure 10: w-statistic results

| | Energy__kWh_ | Total_Duration_seconds | Charging_Time_seconds | start |
|---|---|---|---|---|
| 37384 | 9.155 | 5538 | 5459 | 48420 |
| 1430 | 4.069 | 4364 | 4340 | 59520 |
| 10187 | 8.256 | 14774 | 9859 | 55080 |
| 19611 | 4.289 | 10736 | 4965 | 47880 |
| 38122 | 4.988 | 5660 | 5632 | 38880 |
| ... | ... | ... | ... | ... |
| 26576 | 1.980 | 1340 | 1330 | 45420 |
| 43165 | 4.033 | 2481 | 2466 | 42900 |
| 16866 | 8.424 | 8335 | 8034 | 23160 |
| 8740 | 6.290 | 4082 | 4069 | 44940 |
| 7543 | 8.389 | 9662 | 9639 | 66060 |

5000 rows × 4 columns

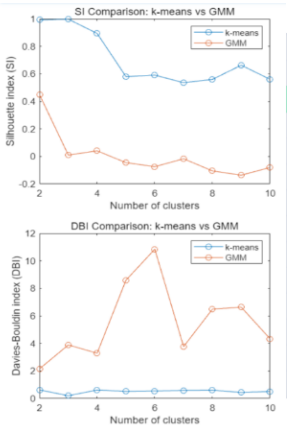
Figure 9: sampled data

## 3.3 Temporal Clustering using Kmeans and GMM

The machine used in the experiment was Intel I7 8750U, MATLAB 2022a, Python 3.9.1 (Pandas, Numpy, Sklearn). The max iteration number in GMM was set to 100000, and tolerance for posterior probabilities was set to 1e-10.

Applying the silhouette method and knee point detection, the determined K is 5. After applying GMM and K-Means to the data, the clusters can be obtained. Comparing the DBI of GMM and K-Means, it can be seen that K-Means is a better algorithm to use in this example. The proportion of mixture of Gaussian models is shown in the table in the appendix A:

In the figure below are the clustering results for GMM and K-Means respectively. It can be seen that the two clustering methods give different five clusters. However, it can be seen that in GMM, the five clusters given in the ”start” column are similar. However, the K-Means clustering gives about three different start times periods.




| Energy | pluginhours | actual charging hours | start time | |
|---|---|---|---|---|
| 7.274142483 | 6226.276224 | 5451.574301 | 47432.04545 | K-Means |
| 7.625837687 | 6052.526119 | 5592.578358 | 25995 | |
| 7.760710214 | 6542.668646 | 5697.844418 | 68571.73397 | |
| 18.19978675 | 16698.18634 | 13614.14079 | 34389.56522 | |
| 6.291786638 | 5242.135776 | 4709.75 | 37304.61207 | |
| 6.7048388 | 5504.588566 | 4921.168697 | 58261.06842 | |
| | | | | |
| 7.923139163 | 8038.149999 | 6470.455435 | 47494.64785 | GMM |
| 6.279908542 | 3756.152716 | 3736.287716 | 49898.39007 | |
| 7.821429066 | 7086.630657 | 7042.853714 | 47510.30391 | |
| 13.26860443 | 17574.96841 | 10335.31934 | 43067.92355 | |
| 4.726129828 | 8452.497651 | 5560.500716 | 47084.37158 | |
| 9.7284968 | 6785.703661 | 6767.308113 | 47511.70847 | |

Figure 11: K-Means and GMM clusteirng results

Based on the clustering results, the users' charging behaviours can be identified as follows:

| | Time | Plug in hrs (hrs) | Power consumption (kWh) |
|---|---|---|---|
| Group 1 | Around 8-9 in the morning | 1-2 hrs / plug in for whole morning | Avg 6.955 kWh (1-2h)<br>18 kWh morning |
| Group 2 | Around 13:17 (noon) | 1 hrs short charging | Avg 6.985 kWh |
| Group 3 | Around 19:04 (evening) | 1.5 hrs | Avg 7.76 kWh |

Figure 12: Clustered user groups

Therefore, EV charger users in Boulder can be categorized into three categories. Daily predictable and repeating patterns can be used as components in time series and to forecast the future value of EV charging prices.

According to the previously discussed clustering effects of GMM and K-means for various data sizes and dimensions. In addition, additional validation was performed in the experiments. Initially, random sampling was employed to reduce the number of samples from 5,000 to 1,000 and 500, and the DBI comparison graph is presented below.

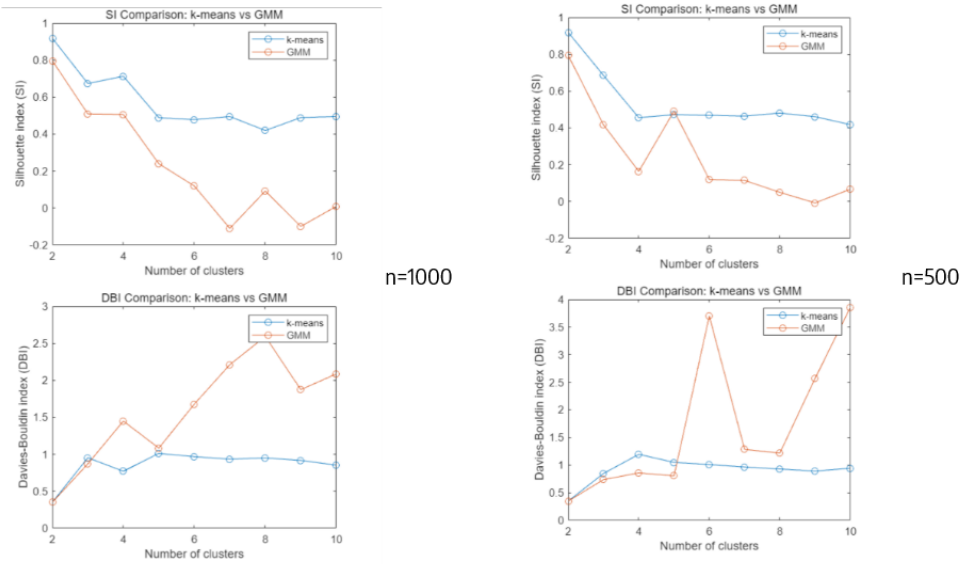



Figure 13: K-Means and GMM clustering results

Although the size of 500 or 1000 in the data above is not representative of the overall data in our data, GMM can be considered to obtain more accurate results for other clustering missions of EV charging facility data, if the data is smaller and less dimensional.

# 4 Conclusion and Future Work

## 4.1 Conclusion

K-Means and GMM, the two most commonly used clustering methods, have different implementation principles and clustering effects. In this paper, we compare the clustering effects of the two methods on data of different sizes, dimensions and shapes, and introduce DBI as an evaluation metric to evaluate them. However, for high dimensional and high data volume data, GMM is sometimes unable to analyse the internal structures of the data, and the final clustering result is not as good as K-Means, and the running time increases quadratically with the increase in dimensions of the data.

At the application level, GMM and K-Means both have their own strengths. GMM demonstrates its accuracy in clustering small data with a Gaussian mixture distribution. In this paper, the problem of clustering the behaviour of users of electric vehicle charging facilities is also discussed. Based on the clustering results, it evident that the usage behaviour of EV charging facility users changes seasonally on a daily basis. There are three different peak times in a day, and the charging behaviour of users shows different trends at different peak times.

## 4.2 Future Work

In this paper only two models, GMM and K-Means, were tested. For model-based clustering methods, as well as for classification-based clustering methods, more clustering methods can be evaluated. In addition, for the clustering of EV charging facility users behaviors, clustering methods such as DBSCAN, which are insensitive to the shape, size and dimension of the data, can be adopted.

Although the dynamic pricing model of TOU pricing is not discussed in this report, the dynamic simulation CLD model for the TOU pricing model can be optimised by incorporating time as a new parameter, thereby improving the accuracy of the pricing model with respect to time. Time-series forecasting of the pricing of smart grid charging facility usage can be performed using the daily trends found in this paper to optimise the pricing of EV charging facility usage in a smart grid working environment.

# References


[1] P. K. Wesseh and B. Lin, "A time-of-use pricing model of the electricity market considering system flexibility," *Energy Reports*, vol. 8, p. 1457–1470, 2022.

[2] Q. Fernando A, "A predictive view of bayesian clustering," *Journal of Statistical Planning and Inference*, vol. 136, no. 8, pp. 2407–2429, 2004.

[3] J. Stutz and P. Cheeseman, "Autoclass — a bayesian approach to classification," *Maximum Entropy and Bayesian Methods*, p. 117–126, 1996.

[4] C. Xiang, P. C. Yong, and L. S. Meng, "Design of multiple-level hybrid classifier for intrusion detection system using bayesian clustering and decision trees," *Pattern Recognition Letters*, vol. 29, no. 7, p. 918–924, 2008.

[5] C. B. Jones, W. Vining, M. Lave, H. Thad, C. Neuman, J. Bennett, and D. R. Scoffield, "Impact of electric vehicle customer response to time-of-use rates on distribution power grids," *SSRN Electronic Journal*, Nov 2022.

[6] O. L. Mangasarian and K. P. Bennett, "Breast cancer wisconsin (diagnostic) data set," 1992. [Online]. Available: https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+28Diagnostic29

[7] May 2022. [Online]. Available: https://open-data.bouldercolorado.gov/datasets/39288b03f8d54b39848a2df9f1c5fca2_0/explore

[8] C. B. Do and S. Batzoglou, "What is the expectation maximization algorithm?" *Nature Biotechnology*, vol. 26, no. 8, p. 897–899, 2008.

[9] J. Rojas Thomas, M. Mora Cofre, and M. Santos, "New version of davies-bouldin index for clustering validation based on hyper rectangles," *6th Chilean Conference on Pattern Recognition (CCPR)*, 2014.

[10] B. van Stein, H. Wang, W. Kowalczyk, M. Emmerich, and T. Bäck, "Cluster-based kriging approximation algorithms for complexity reduction," *Applied Intelligence*, vol. 50, no. 3, p. 778–791, 2019.

[11] A. P. King and R. J. Eckersley, "Inferential statistics iv: Choosing a hypothesis test," *Statistics for Biomedical Engineers and Scientists*, p. 147–171, 2019.

# Appendix

## Code for 1-D model-based clustering

```
function [mu1,mu2]=p_method(data,sigma)
%initialize
    [m,~]=size(data);
    m1=round(rand(m,1));
    m2=1-m1;
    result1=0;
    result2=0;
    for i=1:m
        result1=result1+m1(i)*data(i);
        result2=result2+m2(i)*data(i);
    end
    c1new=result1/sum(m1);
    c2new=result2/sum(m2);
    while c1new==c2new
        [c1new,c2new]=inictr(data);
    end
    mu1=c1new;
    mu2=c2new;
    iter=1;
    cluster1=[];
    cluster2=[];
    cluster=data;
    while iter<1000
        p1new=zeros(1,m);
        p2new=zeros(1,m);
        PZAset=[];
        PZBset=[];
        cluster1=[];
        cluster2=[];

        for i=1:m

            PZA=(exp(-(cluster(i)-mu1)^2/(2*sigma^2)))/
```

```
                ((exp(-(cluster(i)-mu1)^2/(2*sigma^2)))
                +exp(-(cluster(i)-mu2)^2/(2*sigma^2)));
            PZAset(end+1)=PZA;

            PZB=(exp(-(cluster(i)-mu2)^2/(2*sigma^2)))/
                ((exp(-(cluster(i)-mu2)^2/(2*sigma^2)))
                +exp(-(cluster(i)-mu1)^2/(2*sigma^2)));
            PZBset(end+1)=PZB;
        end
        iter=iter+1;
        result1=0;
        result2=0;
        for j=1:m
            result1=result1+PZAset(j)*cluster(j);
            result2=result2+PZBset(j)*cluster(j);
        end
        mu1=result1/sum(PZAset);
        mu2=result2/sum(PZBset);
    end
end
```

## Proportion of Guassian mixture

```
Gaussian mixture distribution with 5 components in 4
Component 1:
Mixing proportion: 0.226039
Mean:   1.0e+04 *

    0.0013    0.8494    0.8462    4.6772

Component 2:
Mixing proportion: 0.135374
Mean:   1.0e+04 *

    0.0006    0.7738    0.6098    4.6591

Component 3:
Mixing proportion: 0.197988
Mean:   1.0e+04 *

    0.0005    0.4894    0.4875    4.9519

Component 4:
Mixing proportion: 0.103381
Mean:   1.0e+04 *

    0.0011    1.4405    0.8624    4.4429

Component 5:
Mixing proportion: 0.337219
Mean:   1.0e+04 *

    0.0008    0.4490    0.4472    4.8888
```

Figure 14: Proportion of Guassian mixture

## 1D clustering results

| dataset1 | 0.012 | 0.5268 | | | |
|---|---|---|---|---|---|
| probability | | mean(0-1) | | MATLAB | |
| -0.11407132 | 0.745909427 | -0.518043101 | 1.235058774 | 1.263678564 | -0.493725647 |
| -0.11407132 | 0.745909427 | 1.235058774 | -0.518043101 | 1.263678564 | -0.493725647 |
| -0.11407132 | 0.745909427 | -0.518043101 | 1.235058774 | 1.263678564 | -0.493725647 |
| -0.11407132 | 0.745909427 | -0.518043101 | 1.235058774 | 1.263678564 | -0.493725647 |
| 0.745909427 | -0.11407132 | -0.518043101 | 1.235058774 | 1.263678564 | -0.493725647 |
| 0.745909427 | -0.11407132 | -0.518043101 | 1.235058774 | 1.263678564 | -0.493725647 |
| 0.745909427 | -0.11407132 | 1.235058774 | -0.518043101 | -0.493725647 | 1.263678564 |
| 0.745909427 | -0.11407132 | 1.235058774 | -0.518043101 | -0.477600974 | 1.283024057 |
| -0.11407132 | 0.745909427 | -0.518043101 | 1.235058774 | -0.493725647 | 1.263678564 |
| -0.11407132 | 0.745909427 | -0.518043101 | 1.235058774 | 1.263678564 | -0.493725647 |
| -0.11407132 | 0.745909427 | 1.235058774 | -0.518043101 | -0.493725647 | 1.263678564 |
| -0.11407132 | 0.745909427 | 1.235058774 | -0.518043101 | 1.263678564 | -0.493725647 |
| -0.11407132 | 0.745909427 | -0.518043101 | 1.235058774 | -0.493725647 | 1.263678564 |
| -0.11407132 | 0.745909427 | 1.235058774 | -0.518043101 | 1.263678564 | -0.493725647 |
| 0.745909427 | -0.11407132 | 1.235058774 | -0.518043101 | 1.263678564 | -0.493725647 |
| -0.11407132 | 0.745909427 | -0.518043101 | 1.235058774 | -0.493725647 | 1.263678564 |
| -0.11407132 | 0.745909427 | -0.518043101 | 1.235058774 | 1.263678564 | -0.493725647 |
| 0.745909427 | -0.11407132 | 1.235058774 | -0.518043101 | -0.493725647 | 1.263678564 |
| 0.745909427 | -0.11407132 | -0.518043101 | 1.235058774 | 1.263678564 | -0.493725647 |
| -0.11407132 | 0.745909427 | -0.518043101 | 1.235058774 | -0.493725647 | 1.263678564 |
| 0.745909427 | -0.11407132 | 1.235058774 | -0.518043101 | -0.493725647 | 1.263678564 |
| 0.745909427 | -0.11407132 | 1.235058774 | -0.518043101 | -0.493725647 | 1.263678564 |
| 0.745909427 | -0.11407132 | 1.235058774 | -0.518043101 | 1.263678564 | -0.493725647 |
| -0.11407132 | 0.745909427 | -0.518043101 | 1.235058774 | 1.283024057 | -0.477600974 |
| 0.745909427 | -0.11407132 | -0.518043101 | 1.235058774 | 1.263678564 | -0.493725647 |
| 0.745909427 | -0.11407132 | 1.235058774 | -0.518043101 | 1.263678564 | -0.493725647 |
| 0.745909427 | -0.11407132 | -0.518043101 | 1.235058774 | -0.493725647 | |

## Comaprison of 3 methods

| GMM | | Own algorithm | | KMeans | |
|---|---|---|---|---|---|
| 0.433222941 | -0.078938583 | 0.512945976 | -0.001522268 | -0.531685796 | 1.209151932 |
| 0.433222017 | -0.078939595 | 0.512945976 | -0.001522268 | -0.531685796 | 1.209151932 |
| -0.07893941 | 0.433222186 | -0.001522268 | 0.512945976 | -0.531685796 | 1.209151932 |
| -0.49227559 | 0.615386385 | -0.001522268 | 0.512945976 | -0.531685796 | 1.209151932 |
| -0.078940059 | 0.433221594 | 0.512945976 | -0.001522268 | -0.531685796 | 1.209151932 |
| -0.078940011 | 0.433221638 | 0.512945976 | -0.001522268 | -0.531685796 | 1.209151932 |
| 0.433222219 | -0.078939374 | 0.512945976 | -0.001522268 | -0.531685796 | 1.209151932 |
| -0.078938845 | 0.433222702 | 0.512945976 | -0.001522268 | 1.209151932 | -0.531685796 |
| -0.078938631 | 0.433222898 | -0.001522268 | 0.512945976 | 1.199559281 | -0.539648241 |
| -0.078939655 | 0.433221962 | -0.001522268 | 0.512945976 | 1.209151932 | -0.531685796 |
| 0.433222025 | -0.078939587 | 0.512945976 | -0.001522268 | 1.199559281 | -0.539648241 |
| -0.49227559 | 0.615386385 | -0.001522268 | 0.512945976 | 1.199559281 | -0.539648241 |
| -0.078939256 | 0.433222327 | 0.512945976 | -0.001522268 | 1.209151932 | -0.531685796 |
| 0.433221663 | -0.078939983 | 0.512945976 | -0.001522268 | -0.531685796 | 1.209151932 |
| -0.078938844 | 0.433222703 | -0.001522268 | 0.512945976 | 1.199559281 | -0.539648241 |
| -0.078939649 | 0.433221968 | 0.512945976 | -0.001522268 | -0.531685796 | 1.209151932 |
| 0.432548567 | -0.079678272 | -0.001522268 | 0.512945976 | 1.199559281 | -0.539648241 |
| -0.49227559 | 0.615386385 | 0.512945976 | -0.001522268 | -0.531685796 | 1.209151932 |
| -0.078939073 | 0.433222494 | 0.512945976 | -0.001522268 | -0.531685796 | 1.209151932 |
| 0.433222192 | -0.078939404 | 0.512945976 | -0.001522268 | -0.531685796 | 1.209151932 |
| -0.078938681 | 0.433222852 | -0.001522268 | 0.512945976 | -0.539648241 | 1.199559281 |
| 0.432547773 | -0.079679145 | 0.512945976 | -0.001522268 | 1.199559281 | -0.539648241 |
| -0.078938538 | 0.433222982 | -0.001522268 | 0.512945976 | 1.199559281 | -0.539648241 |
| -0.078939108 | 0.433222462 | -0.001522268 | 0.512945976 | 1.199559281 | -0.539648241 |
| -0.49227559 | 0.615386385 | 0.512945976 | -0.001522268 | -0.531685796 | 1.209151932 |
| -0.078939928 | 0.433221713 | -0.001522268 | 0.512945976 | -0.539648241 | 1.199559281 |
| 0.433221615 | -0.078940036 | -0.001522268 | 0.512945976 | 1.209151932 | -0.531685796 |
| -0.078938761 | 0.433222779 | 0.512945976 | -0.001522268 | -0.531685796 | 1.209151932 |
| 0.432547566 | -0.079679372 | -0.001522268 | 0.512945976 | -0.531685796 | 1.209151932 |
| -0.078939418 | 0.433222179 | -0.001522268 | 0.512945976 | -0.531685796 | 1.209151932 |
| 0.433222433 | -0.07893914 | -0.001522268 | 0.512945976 | -0.531685796 | 1.209151932 |