

Article

Design of an Unsupervised Machine Learning-based Movie Recommender System

Debby Cintia Ganesha Putri ^{1,*}, Jenq-Shiou Leu ¹, and Pavel Seda ^{1,2}

¹ Department of Electronic and Computer Engineering, National Taiwan University of Science and Technology, Taiwan

² Department of Telecommunications, Brno University of Technology, Czech Republic

³ Institute of Computer Science, Masaryk University, Brno, Czech Republic;

* Correspondence: Debby Cintia Ganesha Putri (e-mail: M10602822@mail.ntust.edu.tw).

Abstract: This research aims to determine the similarities in groups of people to build a film recommender system for users. Users often have difficulty in finding suitable movies due to the increasing amount of movie information. The recommender system is very useful for helping customers choose a preferred movie with the existing features. In this study, the recommender system development is established by using several algorithms to obtain groupings, such as the K -Means algorithm, birch algorithm, mini-batch K -Means algorithm, mean-shift algorithm, affinity propagation algorithm, agglomerative clustering algorithm, and spectral clustering algorithm. We propose methods optimizing K so that each cluster may not significantly increase variance. We are limited to using groupings based on Genre and, Tags for movies. This research can discover better methods for evaluating clustering algorithms. To verify the quality of the recommender system, we adopted the mean square error (MSE), such as the Dunn Matrix and Cluster Validity Indices, and social network analysis (SNA), such as Degree Centrality, Closeness Centrality, and Betweenness Centrality. We also used Average Similarity, Computational Time, Association Rule with Apriori algorithm, and Clustering Performance Evaluation as evaluation measures to compare method performance of recommender systems using Silhouette Coefficient, Calinski-Harabaz Index, and Davies-Bouldin Index.

Keywords: affinity propagation, agglomerative spectral clustering, association rule with Apriori algorithm, average similarity, birch, clustering performance evaluation, computational time, Dunn Matrix, mean-shift, mean squared error, mini-batch K -Means, recommendations system, K -Means, social network analysis.

1. Introduction

The explosion of information on the internet is developing following the rapid advancement of internet technology. The recommender system is a simple mechanism to help users find the right information based on the wishes of internet users by referring to the preference patterns in the dataset. The purpose of the recommender system is to automatically generate proposed items (web pages, news, DVDs, music, movies, books, CDs) for users based on historical preferences and save time searching for them online by extracting worthwhile data. Some websites using the recommender system method include yahoo.com, ebay.com, amazon.com [1–6]. A movie recommender is an application most widely used to help customers select films from a large capacity film library. This algorithm can rank items and show users high-level items and good content to provide a movie recommended based on customer similarity. Customer similarity means collecting film ratings given by individuals based on genre or

tags and then recommending films that promise to target customers based on individuals with identical tastes and preferences.

Traditional recommender systems always suffer from several inherent limitedness, such as poor scalability and data sparsity [7]. Several works have evolved a model-based approach to overcome this problem and provide the benefits of the effectiveness of the existing recommender system. In the literature, many model-based recommender systems were developed by partitioning algorithms, such as *K*-Means, and Self-Organizing Maps (SOM) [8–12].

Other methods that can be used in the recommender system include the clarification method, association rules, and data grouping. The purpose of grouping is to separate users into different groups to form neighbors who are “like-minded” (closest) substitutes of searching the entire user space to increase system scalability [13]. In essence, making high-quality film recommendations with good groupings remains a challenge, and exploring those following efficient grouping methods is an important issue in the recommended system situation. A very useful feature in the recommender system becomes the ability to guess user preferences and needs in analyzing user behavior or other user behaviors to produce a personalized recommender [14].

To overcome the challenges mentioned above, several methods are used to classify performance for the movie recommender system, such as *K*-Means algorithm [15–17], birch algorithm [18], mini-batch *K*-Means algorithm [19], mean-shift algorithm [20], affinity propagation algorithm [21], agglomerative clustering algorithm [22], and spectral clustering algorithm [23]. In this article, we develop a grouping that can be optimized with several algorithms, then obtain the best algorithm in grouping user similarities based on genre, tags, and ratings on movies with the MovieLens dataset. Then, the proposed scheme optimizes *K* for each cluster so that it can significantly reduce variance. To better understand this method, when we talk about variance, we are referring to mistakes. One way to calculate this error is by extracting the centroids of each group, and then squaring this value to remove negative terms. Then, all of these values are added to obtain total error. To verify the quality of the recommender system, we use mean squared error (MSE), Dunn Matrix as Cluster Validity, and social network analysis (SNA). It also uses Average Similarity, Computational Time, Rules of Association with Apriori algorithms, and performance evaluation grouping as evaluation measures to compare the performance for recommender systems.

1.1. Prior Related Works

Zan Wang, X. Y. (2014) presented research on an improved collaborative movie recommender system to develop CF-based approaches for hybrid models to provide movie recommendations that combine dimensional reduction techniques with existing clustering algorithms. In a sparse data environment, “like-minded” selection based on the general ranking is a function of producing high-quality recommended films. Based on the MovieLens data set, an experimental evaluation approach can prove that it is capable of producing high predictive accuracy and more reliable film recommendations for existing user preferences compared to existing CF-based clustering [13]. This study also applies the clustering method to find the nearest cluster and recommends a list of movies based on similarities among users. Our recommender system dataset refers to this research by using MovieLens dataset to establish the experiments, including 100,000 ratings by 943 users on 1682 movies, with a discrete scale of 1–5. Each user has rated at least 20 movies. Then, the dataset was randomly split into training and test data at an 80 % to 20 % ratio. Md. Tayeb Himel, M. N. (2017) researched the Weight Based Movie Recommender System using *K*-Means algorithm [14]. This research uses the *K*-Means algorithm and explains the results. This research motivates us to use other methods as a comparison to identify the algorithm with better performance.

1.2. Problem Formulation

Information overload is a problem in information retrieval, and the recommendation system is one of the main techniques to deal with problems by advising users with appropriate and relevant

80 items. At present, several recommendation systems have been developed for quite different domains,
81 however, this is not appropriate enough to meet user information needs. Therefore, a high-quality
82 recommendation system needs to be built. When designing these recommendations an appropriate
83 method is needed. This paper investigates several appropriate clustering methods for research
84 in developing high-quality recommendation systems with a proposed algorithm that determines
85 similarities to define people group to build a movie recommender system for users. Next, experiments
86 are conducted to make performance comparisons with evaluation criteria on several clustering
87 algorithms using the K -Means algorithm, birch algorithm, mini-batch K -Means algorithm, mean-shift
88 algorithm, affinity propagation algorithm, agglomerative clustering algorithm, and spectral clustering
89 algorithm. The best methods are identified to serve as a foundation to improve and analyze this movie
90 recommender system.

91 1.3. Main Contributions

92 This study investigates several appropriate clustering methods to develop high-quality
93 recommender systems with a proposed algorithm for finding the similarities within groups of people.
94 Next, we conduct experiments to make comparisons on several clustering algorithms including
95 K -Means algorithm, birch algorithm, mini-batch K -Means algorithm, mean-shift algorithm, affinity
96 propagation algorithm, agglomerative clustering algorithm, and spectral clustering algorithm. After
97 that, we find the best method from them as a foundation to improve and analyze this movie
98 recommender system. We limit to using 3 tags and 3 genres because to analyze performance and get
99 good visualization, the most stable results are 3 tags and 3 genres, before we have tried more than 3
100 but the visualization results obtained are not so good with some of the methods used in this study.
101 We start losing the ability to visualize correctly when analyzing three or more dimensions. Then we
102 limit it by using favourite genres and tags and more details on the algorithm comparison. The main
103 contributions of this study are as follows:

- 104 • Performance comparison of several clustering methods to generate a movie recommender system.
- 105 • To optimize the K value in K -Means, mini-batch K -Means, birch, and agglomerative clustering
106 algorithms.
- 107 • To verify the quality of the recommender system, we employed SNA. We also used the Average
108 Similarity to compare performance methods and association rules with the Apriori algorithm of
109 recommender systems.

110 The remainder of this paper is organized as follows. In Sect. 2, we present an overview of the
111 recommender system and review the clustering algorithm and system design of the recommender
112 system. We detail the algorithm design of the K -Means algorithm, birch algorithm, mini-batch K -Means
113 algorithm, mean-shift algorithm, affinity propagation algorithm, agglomerative clustering algorithm,
114 and spectral clustering algorithm. Additionally, we proposed a method to optimize K in some of the
115 methods. This session also explains the evaluation criteria. The experiments, dataset explanation, and
116 results are illustrated in Sect. 3. Evaluation results of algorithms via various test cases and discussions
117 are shown in Sect. 4. Finally, Sect. 5 concludes this work.

118 2. Recommender Systems

119 2.1. Overview

120 A recommender system is a simple algorithm to provide the most relevant information for users
121 to find patterns in the dataset. This algorithm rates the item and indicates the user who is rated high.
122 It attempts to recommend items that best suit customer needs (in the form of products or services). A
123 very useful feature in a recommender system is the ability to guess the preferences and user needs in
124 analyzing user behavior or other user behavior to generate a personalized recommender [14]. The chief
125 purpose of our system is to identify movies based on users' viewing histories and ratings provided in

126 the MovieLens system and datasets [24], and to use a specific algorithm to predict movies. The results
 127 are returned to the user as a recommender item with the parameters of the user. The illustration for
 128 the recommender system is presented in Fig. 1, which explains similarities within people to build a
 129 movie recommendation system for users.

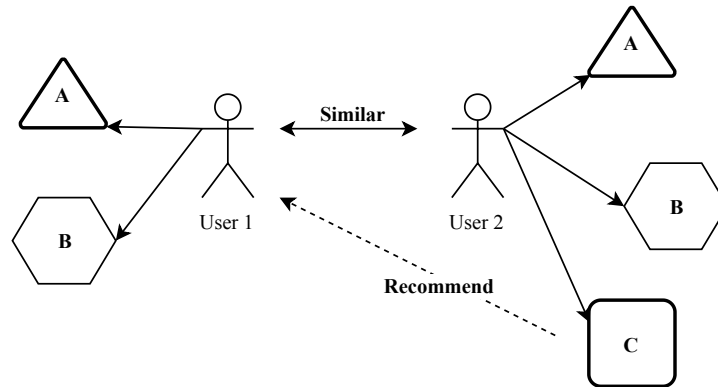


Figure 1. Similarities within people to build a movie recommendation system for users.

130 Fig. 1 shows how two users have a similar interest in items A and B. When this occurs, the
 131 similarity index of both users will be calculated. Furthermore, the system can recommend items C to
 132 other users because the system can detect that both users have similarities in terms of the items.

133 2.2. System Design

134 In this recommender system, *K*-Means algorithm, birch algorithm, mini-batch *K*-Means algorithm,
 135 mean-shift algorithm, affinity propagation algorithm, agglomerative clustering algorithm, and spectral
 136 clustering algorithm are used to determine the best performing algorithms in movie recommendations
 137 based on optimized *K* values. After applying several algorithms, all spaces are searched to obtain the
 138 user nearest neighbor in the same cluster and Top-*N* List of recommender movies. Fig. 2 shows an
 139 overview of the flow of seven existing algorithms.

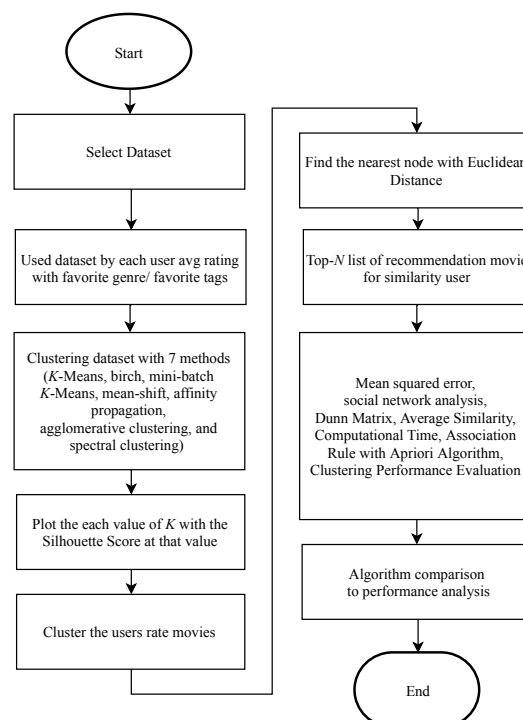


Figure 2. Flowchart configure a recommendation system for movies.

140 2.3. Clustering Algorithm

141 clustering is an analytical method that was used as early as 1939 by Tryon, R. C [25]. Clustering
142 was first used in psychology and then rapidly expanded to other fields. Since the explosion of the
143 amount of information available on the internet, many efforts have been made to reduce the problem
144 of information overload. This overload can be resolved by using a clustering method. clustering is
145 a classification of the same objects from different groups with partitions using existing data into a
146 new group, and each group of data is identified with a certain degree of distance. In cluster analysis,
147 there is also a contrast between parametric and nonparametric approaches [26]. Data clustering is a
148 technique commonly used in various fields, such as data mining, pattern recognition, image analyze,
149 and artificial intelligence. Data clustering is employed to reduce a large amount of data by providing
150 the categories or classifying the data that have a high degree of similarity.

151 2.3.1. K-Means Clustering Algorithm and Optimize K Number Cluster

152 K-Means clustering is a method that automatically divides datasets into k groups [27]. Results
153 selected the initial central cluster k to be iteratively refined by being assigned to the nearest central
154 cluster. Each center of the C_j cluster is updated to an average sample of its constituents.

155 K-Means is used to group approaches given its simplicity, efficiency, and flexibility in calculations
156 especially considering a large amount of data. K-Means calculate the cluster center in assigning objects
157 to the closest cluster based on distance. When the midpoint does not change, the clustering algorithm
158 seeks convergence. However, K-Means lacks the ability to choose the right initial seeds and could
159 cause classification inaccuracies. Selecting a random starting seed can produce a locally good solution
160 that is quite inferior to finding the direct K value. The various initial seeds that run on the same dataset
161 might deliver different partition results. Given a set of objects (x_1, x_2, \dots, x_n) , where each object is
162 an m -dimensional vector, the K-Means algorithm aims at separating these objects to form k groups
163 automatically. Alg. 1 provides the K-Means procedure.

164 Optimize K Number Cluster

165 To overcome the limitations above, we optimized K to choose the correct number of K clusters.
166 Choosing the best number of K clusters is the key point of the K-Means algorithm. To find K , we
167 calculate the clustering error with MSE and silhouette score. First, we select a dataset and choose the
168 range of k values to test. Then, we define the function to calculate the clustered errors and error values
169 for all k values. Finally, we plot each value of k with the silhouette score at that value. A detailed
170 explanation of MSE and silhouette score is provided in [28,29].

171 **(1) Silhouette Score to determine the K number cluster.** The silhouette was first introduced by
172 Peter J. Rousseeuw in [30] in 1986. This is a method of interpretation and validation for clear data
173 clusters. This technique provides a graphical representation of how well each object fits inside the
174 group. The silhouette value for an attribute is given by the equation below:

$$S_i = \frac{a(i) \cdot b(i)}{\max \{a(i), b(i)\}}, \quad (1)$$

175 where $a(i)$ is the average dissimilarity of data point i with other data within the one cluster. Here, $b(i)$
176 is the minimum average dissimilarity of data point i with any other cluster in which i is not inside
177 member a .

178 2.3.2. Birch Clustering Algorithm

179 Balanced Iterative Reducing and clustering using Hierarchies (birch clustering) uses a hierarchical
180 data structure that calls CF-tree for increment and dynamically clusters data points [31]. The birch
181 algorithm uses an input set of data points N , which is represented as a vector of real value, and the

Algorithm 1 K-Means algorithm and optimize k number cluster.**Input:** Selecting dataset and using dataset by each user avg rating with favourite genre/tags;**Output:** Finishing with release Top- N list of recommendation movie for similarity user;

```

1: function K-MEANS()
2:   Choosing  $k$  initial cluster centers
3:    $C_j, j = 1, 2, 3, \dots, k;$ 
4:   Each  $x_i$  is assigned to its closest cluster
5:   center based on the distance metric
6:    $J = \sum_{j=1}^k \sum_{i \in C_{temp}} \|x_i - M_j\|^2,$ 
7:   where  $M_j$  denotes the mean of data points
8:   in  $C_{temp};$ 
9:   Choosing the right  $K$  number of Clusters;
10:  clustering the user's rate movies;
11:  Finding the nearest node to search similarity with
12:  user with Euclidean distance;
13:  if there is not change then
14:    The algorithm has converged and clustering task
15:    is ended, also recalculate the  $M$  of  $K$  clusters
16:    as the new cluster centers and go to;
17:  end if
18: end function

```

182 desired number of cluster K . The first phase builds a CF tree from a data point. This can be defined
 183 with given a set of N d-dimensional data points, and the clustering feature (CF) of the set is used to
 184 establish the triple- $CF = (N, LS, SS)$, where

$$\vec{LS} = \sum_{i=1}^N \vec{x}_i, \quad (2)$$

185 is the linear sum and,

$$\vec{SS} = \sum_{i=1}^N \overrightarrow{(x_i)^2}, \quad (3)$$

186 is the sum of the data points.

187 Clustering features are organized on a CF-tree, a height-balanced tree within two parameters,
 188 including branching factor B and threshold T . Each non-leaf node contains most B entries inside the
 189 form $[CF_i, child_i]$, in which child i is one pointer to its i the child node and CF_i is the clustering feature
 190 representing the associated subcluster.

191 2.3.3. Mini-Batch K-Means Clustering

192 The algorithm Mini-batch K-Means was developed as a modification of the K-Means algorithm.
 193 It uses mini-batch to reduce time in very complex and large-scale calculations of datasets. The efforts
 194 to optimize grouping results could be used with this method. Mini-batch K-Means are randomly used
 195 as input, which is a subset of the entire dataset. Mini-batch K-Means is faster than K-Means and is
 196 usually used for large datasets. For dataset $T = \{x_1, x_2, \dots, x_n\}$, $x_i \in R^{m \cdot n}$, x_i represents a network
 197 record with an n -dimensional real vector. In addition, m indicates the number of records inside the
 198 dataset T . The objective of the clustering problem is to uncover the set C of cluster centers $c \in R^{m \cdot n}$
 199 to minimize the dataset T of records $c \in R^{m \cdot n}$ in function [32]. In contrast to K-Means, mini-batch

200 *K*-Means randomly selects a subset of records from the dataset. Mini-batch *K*-Means greatly reduces
 201 the clustering time and the convergence time. The sum of squared distances is computed in one cluster
 202 as follows:

$$\min \sum_{x \in T} \|f(C, x) - x\|^2, \quad (4)$$

203 where $f(C, x)$ returns the closest of cluster center $c \in C$ to record x , and $|C| = K$ and K is the number
 204 of clusters to obtain.

205 2.3.4. Mean-Shift Clustering

206 The mean-shift algorithm is proposed as a method for cluster analysis [33]. However, given
 207 that the mean-shift determines the gradient ascent, the convergence of the process needs verification,
 208 and its relation with similar algorithms needs clarification. The mean-shift algorithm is part of a
 209 nonparametric grouping technique that does not require prior knowledge of the number of clusters
 210 and does not constrain the shape of the clusters. Mean-shift clustering is used to discover blobs in
 211 a smooth density of samples, which works by updating the candidates for centroids to be the mean
 212 points within a given region. These candidates are then filtered in a postprocessing stage to eliminate
 213 near-duplicates to form the final set of centroids. Given a candidate centroid x_i followed by iteration t ,
 214 the candidate is updated by the following equation:

$$x_i^{t+1} = m(x_i^t), \quad (5)$$

215 where $N(x_i)$ is the neighborhood of samples within a given distance around x_i and m is the
 216 mean-shift vector for each centroid that points against a region of the maximum increase in the density
 217 of points. This is computed using the following equation:

$$m(x_i) = \frac{\sum_{x_j \in N(x_i)} K(x_j - x_i) x_j}{\sum_{x_j \in N(x_i)} K(x_j - x_i)}. \quad (6)$$

218 The algorithm can automatically determine the number of clusters, using bandwidth parameters.
 219 This information can determine the size of the region to search through. This algorithm is unreachable
 220 because it requires several close neighbour searches during its execution.

221 2.3.5. Affinity Propagation Clustering

In the affinity propagation method all data points are considered as possible exemplars. It exchanges real-values between exemplars until high-quality exemplars and corresponding clusters are not provided. The particular messages are further updated based on a simple formula that ratiocinate a sum-product. At any selected point in time, the magnitude in each message represents the current affinity that one point has for choosing another data point as its exemplar, hence the name “affinity propagation” [34]. The messages sent between points belong to one of the two categories. The accumulated evidence $r(i, k)$ of sample k should be the exemplar for sample i . In addition, regarding availability $a(i, k)$, the accumulated indicates that sample i should choose sample k to be an exemplar. The exemplar chosen by samples is similar enough to many samples that are representative of themselves. The responsibility of a sample k to be the exemplar of sample i is given by the following formula:

$$r(i, k) \leftarrow s(i, k) - \max[a(i, k') + s(i, k') \forall k' \neq k]. \quad (7)$$

222 The similarity between samples i and k , $s(i, k)$ is assessed. The availability of sample k to be an
 223 exemplar to sample i is given by the following formula:

$$a(i, k) \leftarrow \min[0, r(k, k) + \sum_{i', t, i' \in \{i, k\}} r(i', k)]. \quad (8)$$

We define a cluster with update $r(i, k)$ and $a(i, k)$. All the values for r and a were set to zero, and each iterate is calculated until convergence is found. To avoid numerical oscillations, the iteration process requires the damping factor γ as follows:

$$r_{t+1}(i, k) = \lambda \cdot r_t(i, k) + (1 + \lambda) \cdot r_{t+1}(i, k), \quad (9)$$

$$a_{t+1}(i, k) = \lambda \cdot a_t(i, k) + (1 + \lambda) \cdot a_{t+1}(i, k). \quad (10)$$

224 2.3.6. Agglomerative Clustering

225 Agglomerative clustering can scale large numbers of samples when used together with the
 226 connectivity matrix and all possible merges are considered at each step. Ward's method is one of the
 227 agglomerative clustering methods based on a classical sum-of-squares criterion, producing groups
 228 that minimize within-group dispersion at each binary fusion. This method uses Euclidean distance as
 229 the distance metric.

$$\|a - b\|_2 = \sqrt{\sum_i (a_i - b_i)^2}. \quad (11)$$

230 2.3.7. Spectral Clustering

231 Spectral clustering uses information from the eigenvalue (spectrum) of a special matrix that will
 232 be built from a graph or data set. This matrix will be built and interpret its spectrum using eigenvectors
 233 to assign data to clusters. An eigenvector is an important object of linear algebra and helps illustrate the
 234 dynamics of the system represented by the matrix. Specific grouping uses the concept of eigenvalues
 235 and eigenvectors.

$$L = D^{-1/2} \cdot AD^{-1/2}. \quad (12)$$

236 Its closest cluster center is assigned based on the distance metric. The affinity matrix is formed,
 237 and the diagonal matrix is defined. The matrix is formed, and the normalized Laplacian matrix and
 238 eigenvectors are computed.

239 2.4. Evaluation Criteria

240 We utilized the training data to develop the offline model, and the remaining data are used to
 241 analyze and provide the movie recommendation. To verify the quality of the recommender system,
 242 we employed the MSE, SNA, Dunn Matrix as Cluster Validity Indices, and evaluation measures
 243 with Average Similarity, Computational Time, Association Rule with Apriori algorithm, Clustering
 244 Performance Evaluation. The following is verified and evaluated.

245 2.4.1. Mean Squared Error

246 MSE is used to facilitate training, and some overall error measure is often used as a performance
 247 metric or an objective function [35].

$$\text{MSE} = \frac{1}{M} \cdot \frac{1}{N} \sum_{m=1}^M \sum_{j=1}^N (d_{mj} - y_{mj})^2, \quad (13)$$

248 where d_{mj} and y_{mj} represent the desired (target) value and output at the m the node for the j the
 249 training pattern respectively, M is the number of output nodes, and N is the number of the training
 250 patterns. The purpose of training is to detect the set of weights that minimize the objective function.

251 MSE is very clever in providing information about this artificially built model. By minimizing the
 252 MSE value, the variant model is minimized. This can provide relatively consistent results as input
 253 data compared to models with large variants (large MSE).

254 2.4.2. Clustering Validity Indices: Dunn Matrix

255 The Dunn Index (DI) is a metric for evaluating clustering algorithms with internal evaluation
 256 schemes, with results being based on the cluster data itself. Dunn's index is the ratio of within and
 257 between cluster separation (Malay K. Pakhira, S. B., 2004). Similar to all other indices, the purpose of
 258 the Dunn index is to identify a set of compact clusters with small variants between cluster members,
 259 that are well separated, and within which the average cluster differs, significantly compared to the
 260 cluster variants.

261 The higher the Dunn index value is, the better the grouping. The number of clusters maximizing
 262 the Dunn index will be taken as the optimal number of clusters k . It also has several shortcomings. As
 263 the number of clusters and data dimensionality increase, the cost of computing also increases. The
 264 Dunn index for c number of clusters is defined as follows:

$$\text{Dunn index}(U) = \min_{1 \leq i \leq c} \left\{ \min_{1 \leq i \leq c} \left\{ \frac{\delta(X_i, X_j)}{\max_{1 \leq i \leq c} \{\Delta(X_k)\}} \right\} \right\}. \quad (14)$$

265 2.4.3. Social Network Analysis

266 To understand relations between one user and another user in clustering, we used SNA. SNA
 267 is a collection of relational methods used to understand and identify relationships between actors
 268 systematically [36], which is widely utilized in the system to determine the level of relation or similarity
 269 between users or actors. User relationships in the clustering could be known through centrality. There
 270 are three general centrality measurements, as reported below [35]:

271 Degree

Degree is the number of relations. An actor with the most relationships is the most important actor. To search density, the most varied grouping with similarity users in one cluster, can be calculated using the following equation:

$$C_D = d(n_i) = \sum_j x_{ij}, \quad (15)$$

272 where C_D is centrality degree, $d(n_i)$ is degree of node i , and X_{ij} is edge $i - j$.

273 Closeness

274 Closeness is the proximity of actors with other actors. The actor is critical of its close relation to
 275 other actors. Both clusters have a high linkage relationship. It can be calculated using the following
 276 Equation:

$$C_c(n_i) = \left[\sum_{j=1}^g d(n_i, n_j) \right]^{-1}, \quad (16)$$

277 where $C_c(n_i)$ is centrality closeness of node I and $d(n_i, n_j)$ is edge $i - j$

$$C'_c = (n_i) = (C_c(n_i)) \cdot (g - 1). \quad (17)$$

278 **Betweenness**

Betweenness is used to calculate the number of shortest paths between actors j and k where actor i is located. The shortest distance has the highest relation between clusters. It can be calculated using the following Equation:

$$C_B(n_i) = \sum_{j < k} \frac{g_{jk} \cdot (n_i)}{g_{jk}} \quad (18)$$

279 where $C_B(n_i)$ is betweenness actor centrality (node) i , $CHF(n_i)$ is the number of actors path where
280 i is, and CHF is the number of the path that connects actors j and k .

281 **2.4.4. Average Similarity**

282 Average Similarity represents the similarity between two clusters. High similarity represents a
283 benchmark for how many clusters are clustering. Cosine similarity measures the similarity between
284 two nonzero vectors by taking the cosine of the angle from between vectors that intervene in their
285 dot product space [37]. The measure is independent of vector length, which makes it a measure
286 typically used for high-dimensional spaces. The cosine of two nonzero vectors can be derived using
287 the Euclidean dot product formula:

$$A \cdot B = \|A\| \cdot \|B\| \cdot \cos(\theta) \quad (19)$$

288 Given two vectors of attributes, A and B , the cosine similarity, $\cos \theta$, is represented using a dot
289 product and magnitude as follows:

$$\begin{aligned} \text{similarity} = \cos(\theta) &= \frac{A \cdot B}{\|A\| \cdot \|B\|} \\ &= \frac{\sum_{i=1}^n A_i \cdot B_i}{\sqrt{\sum_{i=1}^n A_i^2} \cdot \sqrt{\sum_{i=1}^n B_i^2}} \end{aligned} \quad (20)$$

290 **2.4.5. Computational Time**

291 Computational Time is the run time, which is the length of time that is needed to perform the
292 Computation process in a program. In this case, the Computation Time (CT) is calculated as the end
293 time(e) of the algorithm program minus the start time(s) to compare the time performance required.
294 Computational Time, in this case, is given by the following formula:

$$CT = e - s. \quad (21)$$

2.4.6. Association Rule: Apriori Algorithm

Association rule mining is a technique for identifying the fundamental relationships between various items. Profits will be gained from the sale of more items. From the user's transaction data set, rules can be made for customer purchasing trends. Based on the selected movie and customer tastes, it can produce recommendations using the association rules mining technique. Association rule mining is used to find correlations between customers and products [38]. The mining technique of multidimensional association rules is used to identify the most accurate recommendation and assess the movie recommender list. Preprocessing techniques are also used for the rules of association with apriori algorithms. Apriori algorithms can be used so that computers can learn the rules of the association to identify patterns of relationships between one or more items in a dataset. Thus, by using association rules and clustering techniques, we can create an efficient recommendation system that provides recommendations in a shorter time. For example, if the user chooses movie A, and movie B has similarities with movie A, then the user has a tendency to choose movie B that has similarities with movie A and movie C that has similarities with movie B.

There are three major components of the Apriori algorithm, including Support, Confidence, and Lift, Item A is represented by X, and Item B is represented by Y, which are described below.

Support

The term support represents the default popularity of an item. It is calculated based on the division between the number of transactions containing a selected item and the total number of transactions. Here, suppose that we want to find support for item B. This is calculated using the following formula:

$$\begin{aligned} \text{Support}(\{X\} \rightarrow \{Y\}) & \\ &= \frac{\text{Transactions containing both } X \text{ and } Y}{\text{Total number of transactions}}. \end{aligned} \quad (22)$$

Confidence

Confidence refers to the possibility that item B is also purchased if item A is purchased. This can be calculated by finding the number of transactions where A and B were bought together and then divided by the total number of transactions where A was purchased. Mathematically, this relationship is shown below:

$$\begin{aligned} \text{Confidence}(\{X\} \rightarrow \{Y\}) & \\ &= \frac{\text{Transactions containing both } X \text{ and } Y}{\text{Transactions containing } X} \end{aligned} \quad (23)$$

Lift

Lift(A → B) refers to the increase in sales ratio B when A is sold. Lift(A → B) can be calculated by dividing Confidence (A → B) divided by Support (B). Mathematically this relationship is shown below:

$$\begin{aligned} \text{Lift}(\{X\} \rightarrow \{Y\}) & \\ &= \frac{\frac{\text{Transactions containing both } X \text{ and } Y}{\text{Transaction containing } X}}{\text{Fraction of transactions containing } Y} \end{aligned} \quad (24)$$

322 2.4.7. Clustering Performance Evaluation

323 In this section, the seven clustering algorithm indices that have been used in this article to evaluate
324 the partition obtained by the above three techniques for different values of Index are described [39].

325 Silhouette Coefficient

326 If the truth label is unknown, the evaluation must be performed using the model itself. This is
327 the case for the Silhouette Coefficient where the higher score is connected to a model with a better
328 group. We defined the coefficient for each sample. This contains two scores. Specifically, a represents
329 the distance between sample A and all other points in the same class, and b represents the distance
330 between sample B and other points in the next nearest cluster. The distance metric could be calculated
331 using Euclidean distance or the Manhattan distance. The coefficient s for a single sample is then
332 retrieved as follows:

$$s = \frac{b - a}{\max(a, b)}. \quad (25)$$

333 Calinski-Harabaz Index

334 If the ground truth labels are not known, the Calinski-Harabasz index can be used to evaluate the
335 model. A higher Calinski-Harabasz score relates to a model with better clusters assignments. For k
336 clusters, the Calinski-Harabasz score is given as the ratio of the between-clusters dispersion mean and
337 the within-cluster dispersion:

$$s(k) = \frac{Tr(B_k)}{TR(w_k)} \times \frac{N - k}{k - 1}, \quad (26)$$

338 where B_k is the between group dispersion matrix, and W_k is the within-cluster dispersion matrix,
339 which is defined as follows:

$$W_k = \sum_{q=1}^k \sum_{x \in c_q} (x - c_q) \cdot (x - c_q)^T, \quad (27)$$

$$B_k = \sum_q n_q \cdot (c_q - c) \cdot (c_q - c)^T, \quad (28)$$

340 where the N entity is the number of points in data, C_q is the set of points in cluster e_q , C_g be the
341 center of cluster q , c be the set center of E , and n_q be the number of points inside cluster q .

342 Davies-Bouldin Index

343 Similar to the Calinski-Harabaz Index case when the ground truth labels are not known, the
344 Davies Bouldin index can be used for the model evaluation. Here, a lower index is related to a model
345 with better disjunction between the clusters. A simple choice to construct R_{ij} is nonnegative and
346 symmetric as follows:

$$R_{ij} = \frac{s_i \rightarrow s_j}{d_{ij}}. \quad (29)$$

347 The index is defined as the Average Similarity between each cluster C_i for $i = 1, k$ and its most
348 similar cluster C_j . Regarding the context inside the index, the similarity is defined as a measure R_{ij} that

349 trades the s_i , the average distance between each point of cluster i and the centroid of that cluster. Also
350 the distance between cluster centroids i and j . Then the Davies-Bouldin Index is established as follows:

$$DB = \frac{1}{k} \sum_{i=n}^k \max_{i \neq j} R_{ij}. \quad (30)$$

351 3. Experiment Results

352 In this part, the experimental design and empirical result of the proposed movie recommender
353 algorithm via K -Means, birch, mini-batch K -Means, mean-shift, affinity propagation, agglomerative
354 clustering, and spectral clustering technique are presented. To verify the quality of the recommender
355 system, the MSE, Dunn Matrix as Cluster Validity Indices, and SNA are used. Average Similarity,
356 Computational Time, Association Rule with Apriori algorithm, and Clustering Performance Evaluation
357 are used as evaluation measures. Finally, the results are analyzed and discussed. These experiments
358 were performed on Intel(R) Core(TM) i5-2400 CPU @ 3.10 GHz, 8.0 GB RAM computer and run Python
359 3 with Jupyter Notebook 5.7.8 version to simulate the algorithm.

360 3.1. Dataset

361 We use the MovieLens dataset to conduct an experiment and this dataset is also available online.
362 The dataset is a stable benchmark dataset within 20 million ratings and 465,000 tag applications applied
363 to 27,000 movies, including 19 genres, by 138,000 users. The tag data with 12 million relevance scores
364 are incorporated across 1,100 tags. Datasets are determined using a discrete scale of 1–5. We limit the
365 use of clustering based on 3 genres and 3 tags to analyze the performance and get a good visualization.
366 High dimensional could not be handled properly, so the visualization results were not suitable when
367 using all tags and genres. Afterwards the dataset was randomly split into training and test data with
368 a ratio of 80 % / 20 %. The goal is to obtain similarities within groups of people to build a movie
369 recommending system for users. Then, we analyze a dataset from MovieLens user ratings to examine
370 the characteristics people share with regard to movie taste and use this information to rate the movie.

371 3.2. Algorithm Clustering Result

372 We try to establish different clustering algorithms using K -Means, birch, mini-batch K -Means,
373 mean-shift, affinity propagation, agglomerative clustering, and spectral clustering techniques to
374 discover similarities within groups of people to develop a movie recommendation system for users.
375 Then, clustering based on 3 genres and 3 tags is used to analyze the performance, obtain good
376 visualization, and sort movie ratings from high to less by assigning the dataset using a discrete scale of
377 1–5. Then, to overcome the above limitations, K was optimized to select the right K number of clusters.
378 Euclidean distance was also used to find the closest neighbor or user in the cluster. Finally, this study
379 recommends the list- N of movie list based on user similarity. The visualization clustering results for
380 seven different algorithms are presented (see Figs. 3,4,5,6,7,8, and 9). In these figures the X-axis for the
381 first subfigures is the average romance rating, and the Y-axis is the average sci-fi rating. For the second
382 subfigures, the X-axis is the average fantasy rating, and average funny rating is reported on the Y-axis.

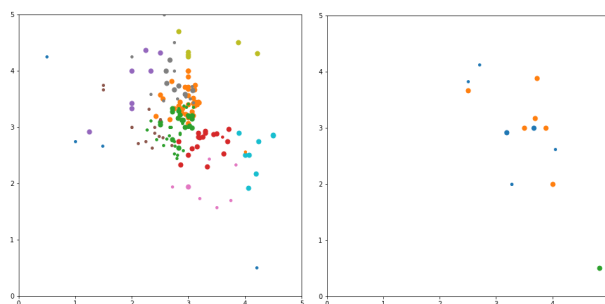


Figure 3. Visualization of K-Means clustering algorithms. K-Means genre (a), K-Means tag (b).

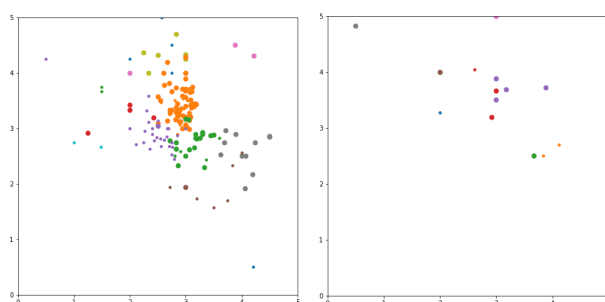


Figure 4. Visualization of birch clustering algorithms. birch genre (c), birch tag (d)

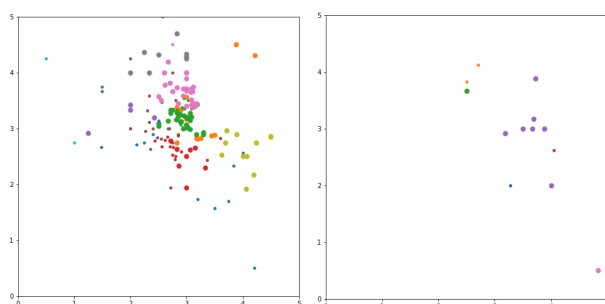


Figure 5. Visualization of mini-batch K-Means clustering. mini-batch K-Means genre (e), mini-batch K-Means tag (f).

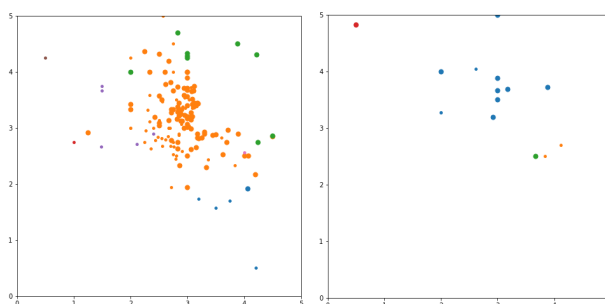


Figure 6. Visualization of mean-shift clustering algorithms. mean-shift genre (g), mean-shift tag (h).

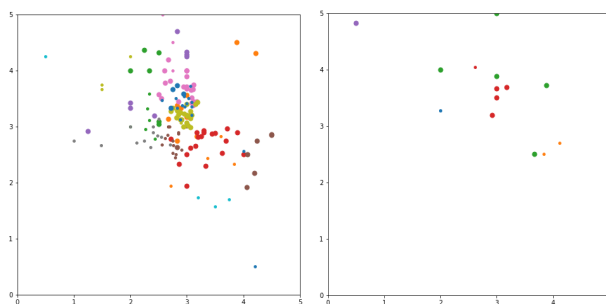


Figure 7. Visualization of affinity propagation clustering algorithms. affinity propagation genre (i), affinity propagation tag (j).

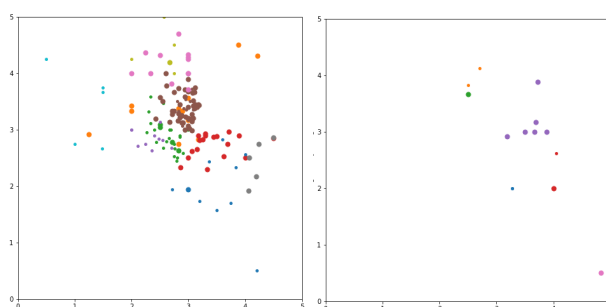


Figure 8. Visualization of agglomerative clustering algorithm. agglomerative clustering genre (k), agglomerative clustering tag (l).

383 To obtain a more delimited subset of people to study, grouping is performed to exclusively obtain
 384 ratings from those who like either romance or science fiction movies. X and Y-axes are romance and
 385 sci-fi ratings, respectively. In addition, the size of the dot represents the ratings of the adventure
 386 movies. The bigger the dot, the higher the adventure rating. The addition of the adventure genre
 387 significantly alters the clustering. The Top N -Movies lists of several clustering algorithms with
 388 K -Means genre n cluster = 12, K -Means tag n cluster = 7, birch genre n cluster = 12, birch tags n cluster
 389 = 12, MiniBatch- K -Means genre n cluster = 12, MiniBatch- K -Means tags n clusters = 7, mean-shift
 390 genre, mean-shift tags, affinity propagation genre, affinity propagation tags, agglomerative clustering
 391 genre n cluster = 12, agglomerative clustering tag n cluster = 7, spectral clustering genre, spectral
 392 clustering tags are reported below.

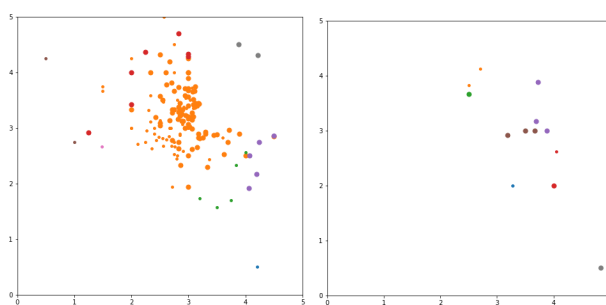


Figure 9. Visualization of spectral clustering algorithm. spectral clustering genre (m), spectral clustering tag (n).

393 Considering a subset of users and discovering what was their favourite genre, we define a function
 394 that would calculate each user's average rating for all romance movies, science fiction movies, and
 395 adventure movies. To obtain a more delimited subset of people to study, we biased our grouping to
 396 exclusively obtain ratings from those users who who like either romance or science fiction movies. We
 397 used the x and y -axes of the romance and sci-fi ratings. In addition, the size of the dot represents the

Casablanca (1942)	4.687500
E.T. the Extra-Terrestrial (1982)	4.550000
There's Something About Mary (1998)	4.500000
Rear Window (1954)	4.500000
One Flew Over the Cuckoo's Nest (1975)	4.406250
Dr. Strangelove or: How I Learned to Stop Worrying and Love the Bomb (1964)	4.333333
Taxi Driver (1976)	4.291667
Chinatown (1974)	4.285714
Dead Man Walking (1995)	4.250000
Clerks (1994)	4.230769
Ferris Bueller's Day Off (1986)	4.227273
Indiana Jones and the Last Crusade (1989)	4.192308
Dark Knight Rises, The (2012)	4.187500
South Park: Bigger, Longer and Uncut (1999)	4.187500
Citizen Kane (1941)	4.187500
Who Framed Roger Rabbit? (1988)	4.166667
Groundhog Day (1993)	4.166667
Seven Samurai (Shichinin no samurai) (1954)	4.166667
Willy Wonka & the Chocolate Factory (1971)	4.150000
Tommy Boy (1995)	4.142857

Figure 10. Example for visualization Genre *K*-Means of Top list-*N* of movies.

398 ratings of the adventure movies (the bigger the dot, the higher the adventure rating). The addition of
 399 the adventure genre significantly affects the clusters. The more data added to our model, the more
 400 similar the preferences of each group are. The final version is Top list of *N* of movies. Additionally, we
 401 considered a subset of users and discovered their favourite tags. We defined a function that calculated
 402 each user's average rating for all funny tag movies, fantasy tag movies, and mafia tag movies. To
 403 obtain a more delimited subset of people to study, we biased our grouping to exclusively obtain ratings
 404 from those users who like either funny or fantasy tags movies. We also results obtained before the
 405 comparison algorithm are Top *N* movies to be given to similar users. The results of Top *N* movies
 406 before the comparison algorithm and Top *N* movies to give to similar users for interest in favourite
 407 genre and tags.

408 3.2.1. Optimize *K* Number Cluster

409 From the results obtained, we can choose the best choices of the *K* values. Choosing the right
 410 number of clusters is one of the key points of the *K*-Means algorithm. We also use mini-batch *K*-Means
 411 algorithm and birch algorithm. We do not apply to mean-shift and affinity propagation because the
 412 algorithm automatically sets the number of clusters. Increasing the number of clusters shows the
 413 range that resulted in the worst clusters based on the Silhouette Score. Optimize *K* is represented by a
 414 Silhouette score. The *X*-axis represents the largest score, so the group is more varied to use and the *Y*
 415 axis represents the number of clusters. So that it can determine the right number of clusters to be used
 416 in displaying visualizations. The results to optimize *K* in several clustering algorithms are presented
 417 below.

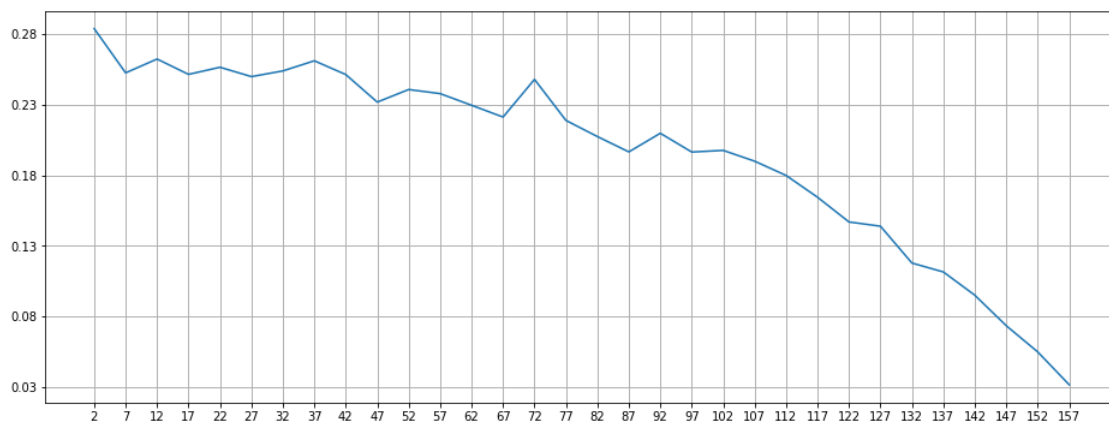


Figure 11. Example of visualization optimization of *K* in *K*-Means genre rating.

418 4. Evaluation and Discussion

419 This section contains the verification and evaluation results of the methodology. The best,
420 performing method is identified, and a discussion is presented.

421 4.1. Evaluation Result

422 The verification and evaluation results are presented below

423 4.1.1. Mean Squared Error

424 Shown in Fig. 12 and Fig. 13 the MSE agglomerative method serves as an example among the
425 seven clustering algorithms.

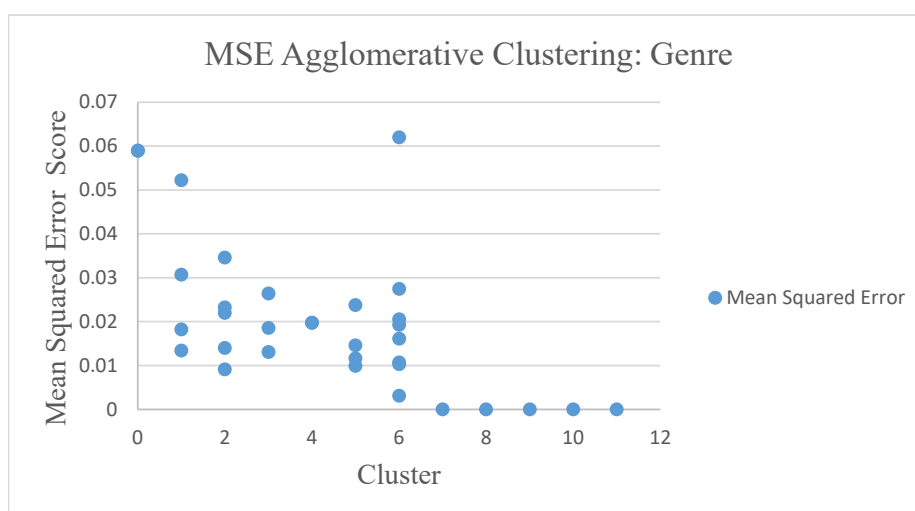


Figure 12. MSE agglomerative clustering genre.

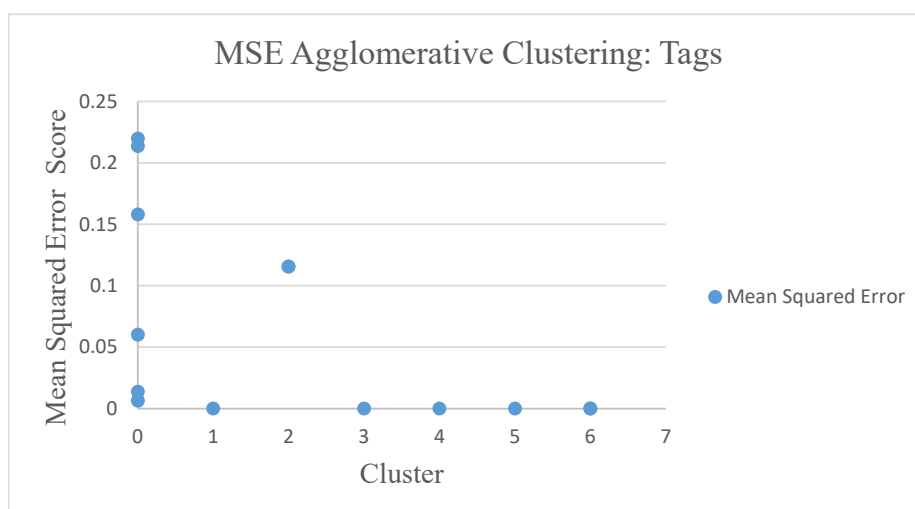


Figure 13. MSE agglomerative clustering tags.

426 4.1.2. Cluster Validity Indices: Dunn Matrix

427 The Dunn matrix is used as a validity measure to compare performance methods of
428 recommendation systems. The following Dunn matrix results are shown in Tab. 1.

Table 1. Dunn Matrix of seven clustering algorithm.

Methods	Amount of Clusters	Score
K-Means algorithm: genre rating	3	0.41
K-Means algorithm: tags rating	3	0.41
birch algorithm: genre rating	3	0.49
birch algorithm: tags rating	3	0.63
mini-batch K-Means algorithm: genre rating	3	0.38
mini-batch K-Means algorithm: tags rating	3	0.37
mean-shift algorithm: genre rating	–	0.39
mean-shift algorithm: tags rating	–	0.63
affinity propagation algorithm: genre rating	–	1.06
affinity propagation algorithm: tags rating	–	0.47
agglomerative clustering algorithm: genre rating	3	0.43
agglomerative clustering algorithm: tags rating	3	0.45
spectral clustering algorithm: genre rating	–	2.09
spectral clustering algorithm: tags rating	–	4.60

429 4.1.3. Average Similarity

430 The Average Similarity birch method example results from seven clustering algorithms are shown
431 in Tab. 2.

Table 2. Average Similarity of birch method.

Methods	Amount of Clusters	Average Similarity
birch algorithm: genre rating	3	0.99
	6	0.97
	12	0.96
birch algorithm: tags rating	4	0.97
	6	0.93
	12	0.94

432 4.1.4. Social Network Analysis

433 Mean-shift results from seven clustering methods for SNA are shown in Tab. 3.

Table 3. Mean-shift social network analysis result.

Methods	Cluster	SNA
mean-shift algorithm: genre rating	Cluster: (0,2,1,4)	Degree, density in cluster 0 (the highest number in result), compare to sequence list of the cluster where the distance result is 5831.49.
		Closeness, the highest in cluster 0 to cluster 1 (both clusters have a high linkage relationship) where the distance result is 1.44.
		Betweenness, the highest in cluster 0 (first) to cluster 2 (end), between 1 (cluster 0 is the most have a relationship where the cluster 1(between) and 2 where the distance result is 1.39.
mean-shift algorithm: tags rating	Cluster: (3, 0, 1, 2)	Degree, the density in cluster 0 (the highest number in result), compare to sequence list of the cluster where the distance result is 148.11.
		Closeness, the highest in cluster 0 to cluster 1 (both clusters have a high linkage relationship) where the distance result is 0.67.
		Betweenness, the highest in cluster 3 (first) to cluster 1 (end), between 0 (cluster 3 is the most have a relationship with cluster 0 (between) and 1 where the distance result is 3.12.

434 4.1.5. Association Rule: Apriori Algorithm

435 The Association Rule with Apriori algorithm is used as an evaluation measure to compare the
 436 method performance of the recommendation systems. An example of results for Association rules
 437 with the apriori algorithm from seven clustering algorithms is shown in the following section.

439 Rule: 12 Angry men (1957) -> Adventures of Priscilla, Queen of the Desert, The (1994)
 440 Support: 0.25
 441 Confidence: 1.0
 442 Lift: 4.0
 443 =====
 444 Rule: 12 Angry Men (1957) -> Airplane! (1980)
 445 Support: 0.25
 446 Confidence: 1.0
 447 Lift: 4.0
 448 =====
 449 Rule: Amadeus (1984) -> 12 Angry men (1957)
 450 Support: 0.25
 451 Confidence: 1.0
 452 Lift: 4.0
 453 =====
 454 Rule: American Beauty (1999) -> 12 Angry Men (1957)
 455 Support: 0.25
 456 Confidence: 1.0
 457 Lift: 4.0
 458 =====
 459 Rule: Austin Powers: International Man of Mystery (1997) -> 12 Angry Men (1957)
 460 Support: 0.25
 461 Confidence: 1.0
 462 Lift: 4.0

465 4.1.6. Computational Time

466 The Computational Time is used as an evaluation measure to compare performance methods of
 467 recommendation systems. Computational Time results are reported below (see Tab. 4).

Table 4. Computational Time of seven clustering algorithm.

clustering method	Computational Time [ms]
<i>K</i> -Means-genre	31.16
<i>K</i> -Means-tags	14.43
birch-genre	24.49
birch-tags	15.34
mini-batch <i>K</i> -Means method-genre	23.82
mini-batch <i>K</i> -Means method-tags	15.79
mean-shift-genre	13.75
mean-shift-tags	10.15
affinity propagation-genre	20.04
affinity propagation-tags	8.53
agglomerative clustering-genre	32.00
agglomerative clustering-tags	10.37
spectral clustering-genre	15.55
spectral clustering-tags	6.22

468 4.1.7. Clustering Performance Evaluation

469 Clustering Performance Evaluation (CPE) result of *K*-Means and birch method examples from
 470 seven clustering algorithm are reported below (see Tab. 5).

Table 5. Clustering Performance Evaluation of K-Means method and birch method.

methods	clustering Performance Evaluation	Score
K-Means algorithm: genre rating	Silhouette Coefficient	0.29
	Calinski Harabaz Index	59.41
	Davies Bouldin Index	1.13
K-Means algorithm: tags rating	Silhouette Coefficient	0.25
	Calinski Harabaz Index	7.47
	Davies Bouldin Index	0.86
birch algorithm: genre rating	Silhouette Coefficient	0.23
	Calinski Harabaz Index	39.03
	Davies Bouldin Index	1.24
birch algorithm tags rating	Silhouette Coefficient	0.25
	Calinski Harabaz Index	5.73
	Davies Bouldin Index	1.16

471 4.2. Discussion

472 A detailed explanation of the above-mentioned experiments is discussed in the subsequent section.
 473 In general, for all of these methods, the higher the value of the lift, support, and confidence, the better
 474 the link is for the recommender system. Further, the higher the Dunn index value, the better the
 475 grouping.

476 4.2.1. K-Means Performance

477 Movie recommender quality is evaluated with K-Means. MSE results from K-Means show
 478 different results for genre rating and rating tags. The rating tag results are relatively smaller with
 479 rating genre scores of 0-0.95 and rating tags scores of 0-0.28. K-Means has a Dunn Matrix that tends
 480 to be evenly distributed for genre and tags with values of 0.41. The higher the Dunn index value,
 481 the better the grouping. The Average Similarity in the genre showed that the value increases as the
 482 number of clusters decreases. The Average Similarity in K-Means tags shows that the high similarity
 483 value depends on the number of clusters. The Association Rule with Apriori algorithm in K-Means
 484 clustering approach 13 % support for the genre and 25 % for tags of customers who choose movies A
 485 and B. Support is an indication of how often the itemset appears in linkages. The confidence is 61 %
 486 for the genre and 100 % for tags of the customers who choose movie A and movie B. Lift represents the
 487 ratio of 3.3 for genre and 4.0 for tags of the observed support value. This is a conditional probability.
 488 Clustering Performance Evaluation showed that the K-Means method showed good performance with
 489 the Calinski-Harabaz Index with a score of 59.41.

490 4.2.2. Birch Performance

491 To evaluate the movie recommender quality with birch, MSE results from birch showed relatively
 492 small results with a rating genre score range of 0-0.25 and tag scores of 0-0.17. birch tag ratings have
 493 a Dunn Matrix value that tends to be greater than 0.64. The Average Similarity in the birch genre
 494 showed that the value increases, as the number of clusters decreases. The Average Similarity in birch
 495 tags revealed that the high similarity value depended on the number of clusters. The Association Rule
 496 with Apriori algorithm in the birch clustering approach provides 16 % support for genre and 50 %
 497 support for tags of customers who choose movies A and B. Support is an indication of how often the
 498 itemset appears in linkages. Confidence is 100 % for the genre and 50 % for tags of the customers who
 499 choose movie A and B. Lift represents the ratio of 4.0 for genre and 1.0 for tags of the observed support
 500 value. The performance evaluation showed that this method provides good performance with a score
 501 of 1.24 on the Davies-Bouldin Index.

502 4.2.3. Mini-batch K-Means Performance

503 To evaluate movie recommender quality with mini-batch K-Means, MSE results from mini-batch
504 K-Means showed different results in genre rating and rating tags. The rating tag results were relatively
505 smaller with a rating genre score range of 0-0.69 and rating tag scores of 0-0.19. The mini-batch
506 K-Means with genre rating has a Dunn Matrix value that tends to greater than 0.38. The Average
507 Similarity in the mini-batch K-Means genre showed that the high similarity value depended on the
508 number of clusters. The Average Similarity in the mini-batch K-Means tags showed that the high
509 similarity value depended on the number of clusters. The Association Rule with Apriori algorithm in
510 mini-batch K-Means clustering approach provides 13 % support for genre and 14 % support for tags of
511 customers who choose movies A and B. Support is an indication of how often the itemset appears in
512 linkages. Confidence is 100 % for the genre and 100 % for tags of the customers who choose movie A
513 and movie B. Lift represents the ratio of 3.75 for genre and 7.0 for tags of the observed support value.
514 The evaluation showed that the mini-batch K-Means method performs well with Calinski-Harabaz
515 Index with a score of 48.18.

516 4.2.4. Mean-shift Performance

517 To evaluate the movie recommender quality with the mean-shift, the MSE results from the
518 mean-shift showed relatively larger results with a genre rating score range of 0-1 and a tag score of
519 0-1. The mean-shift algorithm with rating tags has a Dunn Matrix value that tends to be greater than
520 0.63. The Average Similarity in the mean-shift genre showed that the value increases as the number of
521 clusters decreases. The Average Similarity in tags mean-shift showed that the high similarity value
522 depended on the number of clusters. The mean-shift in the genre has the best Computational Time at
523 13.75 ms. The Association Rule with Apriori algorithm in mean-shift clustering approach provides
524 12 % support for genre and 9 % for tags of customers who choose movies A and B. Support is an
525 indication of how often the itemset appears in linkages. Confidence is 81 % for the genre and 100 %
526 for tags of the customers who choose movie A and movie B. Lift represents a ratio of 3.06 for genre
527 and 5.5 for tags of the observed support value. The abovementioned evaluation depicts the affinity
528 propagation method to provide sufficient performance with Calinski-Harabaz Index with a score of
529 20.14.

530 4.2.5. Affinity Propagation Performance

531 To evaluate the quality with affinity propagation movie, the results of the MSE from affinity
532 propagation showed different results for genre rating and rating tags. The rating genre results were
533 relatively smaller with a genre rating score range of 0-0.17 and a rating tag score of 0-0.89. The affinity
534 propagation algorithm with genre rating has a Dunn Matrix which tends to be higher at 1.06. The
535 Average Similarity in the affinity propagation genre showed that the high similarity value depended
536 on the number of clusters. The Average Similarity in affinity propagation tags showed that the high
537 similarity value depended on the number of clusters. The Association Rule with Apriori algorithm in
538 affinity propagation clustering approach provided 10.5 % support for the genre and 20 % support for
539 tags of customers choose movies A and B. Support is an indication of how often the itemset appears
540 in linkages. Here, 66 % confidence is noted for the genre and 100 % for tags of the customers who
541 choose movie A and movie B. Lift represents the ratio of 4.22 for genre and 5.0 for tags of the observed
542 support value. Clustering Performance Evaluation also showed that the affinity propagation method
543 showed good performance with the Calinski-Harabaz Index with a score of 53.49.

544 4.2.6. Agglomerative Clustering Performance

545 To evaluate the movie recommender quality with agglomerative clustering. MSE results from
546 agglomerative clustering showed different results in genre rating and rating tags. The rating genre
547 results were relatively smaller with rating genres scores of 0-0.06 and rating tags scores with 0-0.23.

548 Agglomerative clustering algorithms with rating tags have Dunn Matrix results that tend to be greater
549 than 0.45. The Average Similarity in the agglomerative clustering genre showed that the high similarity
550 value depended on the number of clusters. The Average Similarity in agglomerative clustering tags
551 showed that the high similarity value depended on the number of clusters. The Association Rule
552 with the Apriori algorithm in agglomerative clustering approach provides 22 % support for the genre
553 and 16 % for tags of customers who choose movies A and B. Support is an indication of how often
554 the itemset appears in linkages. In addition, 22 % confidence is noted for the genre and 16 % for tags
555 of customers who choose movie A and movie B. Lift represents a ratio of 1.0 for genre and 1.0 for
556 tags of the observed support value. From the performance evaluation, we see that the agglomerative
557 clustering method performs well with the Calinski-Harabaz Index with a score of 49.34.

558 4.2.7. Spectral Clustering Performance

559 To evaluate the movie recommender quality with spectral clustering. MSE results from spectral
560 clustering showed differences in genre rating and rating tags. The rating tag results were relatively
561 smaller with a rating genre score range of 0-0.62 and rating tags scores of 0-0.17. Spectral clustering
562 algorithm with tag rating has the best Dunn Matrix results at 4.61 and the best spectral clustering
563 algorithm results with genre at 2.09. The Average Similarity in the spectral clustering genre showed
564 that the high similarity value depended on the number of clusters. The Average Similarity in spectral
565 clustering tags showed that the high similarity value depended on the number of clusters. Spectral
566 clustering in tags has the best Computational Time at 6.22 ms. The Association Rule with Apriori
567 algorithm in spectral clustering approach provides 12 % support for the genre and 33 % support for
568 tags of customers choose movies A and B. Support is an indication of how often the itemset appears
569 in linkages. Confidence is 75 % for the genre and 100 % for tags of the customers who choose movie
570 A and movie B. Lift represents the ratio of 3.12 for genre and 3.0 for tags of the observed support
571 values. Clustering Performance Evaluation showed that the spectral clustering method showed good
572 performance with the Calinski-Harabaz Index with a score of 16.39.

573 5. Conclusion

574 In this study, seven clusterings were used to cluster performance comparison methods for movie
575 recommendation systems, such as the *K*-Means algorithm, birch algorithm, mini-batch *K*-Means
576 algorithm, mean-shift algorithm, affinity propagation algorithm, agglomerative clustering algorithm,
577 and spectral clustering algorithm. The developed optimized groupings from several algorithms were
578 then used to compare the best algorithms with regard to the similarity groupings of users on movie
579 genre, tags, and rating using the MovieLens dataset. Then, optimizing *K* for each cluster did not
580 significantly increase the variance. To better understand this method, variance refers to the error. One
581 of the ways to calculate this error is to extract the centroid of its respective groups. Then, this value is
582 squared (to remove the negative terms), and all those values are added to obtain the total error. To
583 verify the quality of the recommender system, the MSE, Dunn Matrix as Cluster Validity Indices and
584 SNA were used. In addition, Average Similarity, Computational Time, Association Rule with Apriori
585 algorithm and Clustering Performance Evaluation measures were used to compare the methods of
586 performance systems.

587 Using the MovieLens dataset, experiment evaluation of the seven clustering methods revealed
588 the following:

- 589 1. The best MSE value is produced by the birch method with a relatively small squared error score
590 in the rating genre and rating tags.
- 591 2. spectral clustering algorithm with tag rating has the best Dunn Matrix results at 4.61 and the
592 spectral clustering algorithm has the best genre results at 2.09. The higher the Dunn index value
593 is, the better the grouping.
- 594 3. The closest distance to the SNA is the mean-shift method, which indicates that the distance
595 between clusters has a high linkage relationship invariance.

- 596 4. The birch method had a relatively high average similarity to increase the number of clusters,
597 which showed a good level of similarity in clustering.
- 598 5. The best Computational Time is indicated by the mean-shift for genre at 13.75 ms and spectral
599 clustering for tags at 6.22 ms.
- 600 6. Visualization of clustering and optimizing k for movie genre in algorithms is better than movie
601 tags because fewer data are used for movie tags.
- 602 7. Mini-batch K -Means clustering approach is the best approach for the Association Rule with
603 Apriori algorithm with a high score of support, 100 % confidence, and 7.0 ratio of lift for item
604 recommendations.
- 605 8. Clustering Performance Evaluation shows that the K -Means method exhibits good performance
606 with the Calinski-Harabaz Index with a score of 59.41, and the birch algorithm with a score of
607 1.24, on the Davies-Bouldin Index.
- 608 9. Birch is the best method based on a comparison of several performance matrices.

609 Acknowledgments

610 This research was supported by the Ministry of Science and Technology (MOST) under the
611 grant MOST-108-2221-E-011-061- and MIT Laboratory, National Taiwan University of Science and
612 Technology. v2

613 For the research, infrastructure of the SIX Center was used.

614

- 615 1. Isinkaye, F.; Folajimi, Y.; Ojokoh, B. Recommendation systems: Principles, methods and evaluation.
616 *Egyptian Informatics Journal* **2015**, *16*, 261–273.
- 617 2. Nilashi, M.; Salahshour, M.; Ibrahim, O.; Mardani, A.; Esfahani, M.D.; Zakuan, N. A new method for
618 collaborative filtering recommender systems: the case of yahoo! movies and tripadvisor datasets. *Journal*
619 *of Soft Computing and Decision Support Systems* **2016**, *3*, 44–46.
- 620 3. Smith, B.; Linden, G. Two decades of recommender systems at Amazon. com. *Ieee internet computing* **2017**,
621 *21*, 12–18.
- 622 4. Greenstein-Messica, A.; Rokach, L. Personal price aware multi-seller recommender system: Evidence from
623 eBay. *Knowledge-Based Systems* **2018**, *150*, 14–26.
- 624 5. Itmazi, J.; Megías, M. Using recommendation systems in course management systems to recommend
625 learning objects. *International Arab Journal of Information Technology (IAJIT)* **2008**, *5*.
- 626 6. Kumar, M.; Yadav, D.; Singh, A.; Gupta, V.K. A movie recommender system: Movrec. *International Journal*
627 *of Computer Applications* **2015**, *124*.
- 628 7. Lu, J.; Wu, D.; Mao, M.; Wang, W.; Zhang, G. Recommender system application developments: a survey.
629 *Decision Support Systems* **2015**, *74*, 12–32.
- 630 8. Shah, N.; Mahajan, S. Document clustering: a detailed review. *International Journal of Applied Information*
631 *Systems* **2012**, *4*, 30–38.
- 632 9. Yang, M.S.; Sinaga, K.P. A Feature-Reduction Multi-View k -Means Clustering Algorithm. *IEEE Access* **2019**,
633 *7*, 114472–114486.
- 634 10. Wu, J.L.; Chang, P.C.; Tsao, C.C.; Fan, C.Y. A patent quality analysis and classification system using
635 self-organizing maps with support vector machine. *Applied soft computing* **2016**, *41*, 305–316.
- 636 11. Qu, X.; Yang, L.; Guo, K.; Ma, L.; Feng, T.; Ren, S.; Sun, M. Statistics-Enhanced Direct Batch Growth
637 Self-Organizing Mapping for Efficient DoS Attack Detection. *IEEE Access* **2019**, *7*, 78434–78441.
- 638 12. Lv, Z.; Liu, T.; Shi, C.; Benediktsson, J.A.; Du, H. Novel land cover change detection method based on
639 K -means clustering and adaptive majority voting using bitemporal remote sensing images. *IEEE Access*
640 **2019**, *7*, 34425–34437.
- 641 13. Wang, Z.; Yu, X.; Feng, N.; Wang, Z. An improved collaborative movie recommendation system using
642 computational intelligence. *Journal of Visual Languages & Computing* **2014**, *25*, 667–675.
- 643 14. Himel, M.T.; Uddin, M.N.; Hossain, M.A.; Jang, Y.M. Weight based movie recommendation system using
644 K -means algorithm. 2017 International Conference on Information and Communication Technology
645 Convergence (ICTC). IEEE, 2017, pp. 1302–1306.

- 646 15. Hajjar, M.; Aldabbagh, G.; Dimitriou, N.; Win, M.Z. Hybrid clustering scheme for relaying in multi-cell
647 LTE high user density networks. *IEEE Access* **2017**, *5*, 4431–4438.
- 648 16. Dhanachandra, N.; Manglem, K.; Chanu, Y.J. Image segmentation using K-means clustering algorithm and
649 subtractive clustering algorithm. *Procedia Computer Science* **2015**, *54*, 764–771.
- 650 17. Arora, P.; Varshney, S.; others. Analysis of k-means and k-medoids algorithm for big data. *Procedia*
651 *Computer Science* **2016**, *78*, 507–512.
- 652 18. Yang, Y.; Wu, L.; Guo, J.; Liu, S. Research on distributed Hilbert R tree spatial index based on BIRCH
653 clustering. 2012 20th International Conference on Geoinformatics. IEEE, 2012, pp. 1–5.
- 654 19. Peng, K.; Leung, V.C.; Huang, Q. Clustering approach based on mini batch kmeans for intrusion detection
655 system over big data. *IEEE Access* **2018**, *6*, 11897–11906.
- 656 20. Chen, Y.; Hu, P.; Wang, W. Improved K-Means Algorithm and its Implementation Based on Mean Shift.
657 2018 11th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics
658 (CISP-BMEI). IEEE, 2018, pp. 1–5.
- 659 21. Sohn, I.; Lee, J.H.; Lee, S.H. Low-energy adaptive clustering hierarchy using affinity propagation for
660 wireless sensor networks. *IEEE Communications Letters* **2016**, *20*, 558–561.
- 661 22. Zhang, X.; Xu, Z. Hesitant fuzzy agglomerative hierarchical clustering algorithms. *International Journal of*
662 *Systems Science* **2015**, *46*, 562–576.
- 663 23. Shang, R.; Zhang, Z.; Jiao, L.; Wang, W.; Yang, S. Global discriminative-based nonnegative spectral
664 clustering. *Pattern Recognition* **2016**, *55*, 172–182.
- 665 24. Harper, F.M.; Konstan, J.A. The movielens datasets: History and context. *Acm transactions on interactive*
666 *intelligent systems (tiis)* **2016**, *5*, 19.
- 667 25. Robert, C.; others. Cluster analysis: correlation profile and orthometric (factor) analysis for the isolation of
668 unities in mind and personality. *Edwards Brothers* **1939**.
- 669 26. Cox, R.W.; Chen, G.; Glen, D.R.; Reynolds, R.C.; Taylor, P.A. FMRI clustering in AFNI: false-positive rates
670 redux. *Brain connectivity* **2017**, *7*, 152–171.
- 671 27. Wagstaff, K.; Cardie, C.; Rogers, S.; Schrödl, S.; others. Constrained k-means clustering with background
672 knowledge. *Icml*, 2001, Vol. 1, pp. 577–584.
- 673 28. Kodinariya, T.M.; Makwana, P.R. Review on determining number of Cluster in K-Means Clustering.
674 *International Journal* **2013**, *1*, 90–95.
- 675 29. Liang, N.; Zheng, H.T.; Chen, J.Y.; Sangaiah, A.; Zhao, C.Z. TRSDL: Tag-Aware Recommender System
676 Based on Deep Learning–Intelligent Computing Systems. *Applied Sciences* **2018**, *8*, 799.
- 677 30. Massart, D.L.; Kaufman, L.; Rousseeuw, P.J.; Leroy, A. Least median of squares: a robust method for outlier
678 and model error detection in regression and calibration. *Analytica Chimica Acta* **1986**, *187*, 171–179.
- 679 31. Sheikholeslami, G.; Chatterjee, S.; Zhang, A. Wavecluster: A multi-resolution clustering approach for very
680 large spatial databases. *VLDB*, 1998, Vol. 98, pp. 428–439.
- 681 32. Maimon, O.; Rokach, L. *Data mining and knowledge discovery handbook* **2005**.
- 682 33. Fukunaga, K.; Hostetler, L. The estimation of the gradient of a density function, with applications in
683 pattern recognition. *IEEE Transactions on information theory* **1975**, *21*, 32–40.
- 684 34. Dueck, D. *Affinity propagation: clustering data by passing messages*; Citeseer, 2009.
- 685 35. Rukmi, A.M.; Iqbal, I.M. Using k-means++ algorithm for researchers clustering. AIP Conference
686 Proceedings. AIP Publishing, 2017, Vol. 1867, p. 020052.
- 687 36. Freeman, L. The development of social network analysis. *A Study in the Sociology of Science* **2004**, *1*.
- 688 37. Plattel, C. Distributed and Incremental Clustering using Shared Nearest Neighbours. Master's thesis, 2014.
- 689 38. Malik, J.S.; Goyal, P.; Sharma, A.K. A comprehensive approach towards data preprocessing techniques &
690 association rules. Proceedings of The4th National Conference, 2010.
- 691 39. Maulik, U.; Bandyopadhyay, S. Performance evaluation of some clustering algorithms and validity indices.
692 *IEEE Transactions on pattern analysis and machine intelligence* **2002**, *24*, 1650–1654.