Advance Access publication on 27 July 2022 https://doi.org/10.1093/iwc/iwac022

Information Architecture: Using K-Means Clustering and the Best Merge Method for Open Card Sorting Data Analysis

SIONE PAEA[1,*], CHRISTOS KATSANOS[2] AND GABIRIELE BULIVOU[1]

[1]*School of Information Technology, Engineering, Mathematics and Physics, The University of the South Pacific, Suva, Fiji*
[2]*Department of Informatics, Aristotle University of Thessaloniki, 54124 Thessaloniki, Greece*
**Corresponding author: sione.paea@usp.ac.fj*

Open card sorting is a well-established method for discovering how people understand and categorize information. This paper addresses the problem of quantitatively analyzing open card sorting data using the K-means algorithm. Although the K-means algorithm is effective, its results are too sensitive to initial category centers. Therefore, many approaches in the literature have focused on determining suitable initial centers. However, this is not always possible, especially when the number of categories is increased. This paper proposes an approach to improve the quality of the solution produced by the K-means for open card sort data analysis. Results show that the proposed initialization approach for K-means outperforms existing initialization methods, such as MaxMin, random initialization and K-means++. The proposed algorithm is applied to a real-world open card sorting dataset, and, unlike existing solutions in the literature, it can be used with any number of participants and cards.

RESEARCH HIGHLIGHTS

- This paper proposes a new algorithm, called BMK-means, that combines for the first time partitional category with insights from hierarchical categories to analyze open card sorting data.
- The proposed BMK-means algorithm identifies the optimal number of categories, creates the initial core categories using the best merge method (BMM), identifies the initial centers and finally applies the K-means to create categories from open card sorting data.
- The rationale for the proposed algorithm is that the category results heavily depend on the goodness of the initialization technique and the chosen number of categories.
- The paper shows that the proposed algorithm is better for analyzing open card sort data compared to three known K-means variations: MaxMin, random initialization, and K-means++.

Keywords: information architecture; dendrograms; category; hypertext/hypermedia; user-centered design; HCI design and evaluation methods

Handling Editor: Dr. Helen Petrie

Received 6 September 2021; Revised 25 April 2022; Accepted 9 June 2022

1. INTRODUCTION

Open card sorting is a method used to derive an information architecture (IA) based on users' groupings of the content (Rosenfeld *et al.*, 2002). IA represents the underlying structures that give shape and meaning to the content and functionality of an interactive system (Kalbach, 2007; Katsanos *et al.*, 2019). User-centered IA aims to increase the findability of information (Morville and Rosenfeld, 2006) and enhance the user

experience and information integration processes (Bitan *et al.*, 2019). The most widely adapted method to support the design of user-centered IAs is card sorting (Katsanos *et al.*, 2019; Paea and Baird, 2018;Paea *et al.*, 2020).

Quantitative analysis of open card sorting data can be viewed as a clustering problem. The latter is defined as the problem of finding homogeneous groups of data points in a dataset. These groups are called clusters or categories and can be defined as a region in which the density of objects is locally higher than in other regions (Likas *et al.*, 2003). In the context of open card sorting, the aim is to form groups (i.e. categories) of the provided cards (i.e. content items). There are two primary branches of clustering algorithms: hierarchical and partitional (Jain *et al.*, 1999). Open card sorting datasets are typically analyzed using hierarchical clustering algorithms (Paul, 2014; Katsanos *et al.*, 2019). Using partitional clustering algorithms to analyze open card sorting datasets is still in its infancy and therefore remains an open problem.

The K-means algorithm is one of the most popular partitional clustering methods (Likas *et al.*, 2003; Redmond and Heneghan, 2007). Although it has the great advantage of being easy to implement, it still has some drawbacks (Shukla and Naganna, 2014; Fränti and Sieranoja, 2019). One important drawback is that poor initialization can cause the iterations to get stuck into an inferior local minimum (Fränti and Sieranoja, 2019). The K-means result, therefore, depends a lot on the initialization. This paper proposes an improved K-means algorithm that can resolve this problem. In specific, a method based on the best merge method (BMM) is used to generate the initial category centers to replace the random way in the original K-means algorithm. The BMM is derived from hierarchical cluster analysis, which is widely used in the industry to see the patterns of users' open card sorting datasets (Paea and Baird, 2018; Paea *et al.*, 2020). Thus, our approach combines for the first time partitional clustering with insights from hierarchical clustering to analyze open card sorting data.

This paper presents a new algorithm for quantitative analysis of open card sort data, named best merge K-means (BMK-means) algorithm. The BMK-means algorithm uses a new initialization method for the K-means algorithm in open card sorting data analysis. We compare the BMK-means algorithm with three existing initialization methods (random initialization, MinMax and K-means++) to analyze a real-world open card sorting dataset. We found that the proposed algorithm produces categories of better quality and converges faster compared to the existing methods. Recently, Paea and Baird (2018) proposed a promising method that also uses K-means clustering to analyze open card sort data. However, they found that their method works well for up to 30 participants and 30 cards. Our method solves this limitation, as it is expected to work equally well for any number of participants and cards.

In the following section, we start with a brief description of the background and related works. Section 3 describes the study methodology. Section 4 explains the proposed algorithm. Section 5 presents the results. Section 6 compares the proposed algorithm with three existing K-means initialization methods. Finally, Section 7 provides conclusions and describes directions for future research and the limitations of this study.

2. BACKGROUND AND RELATED WORK

2.1. Card sorting

Card sorting applies to a wide variety of activities involving ordering, grouping and/or naming objects or concepts. It can provide insight into users' mental models, illuminating the way that they often tacitly group, sort and label tasks and content within their own heads (Morville and Rosenfeld, 2006; Katsanos *et al.*, 2019). Card sorting is based on the assertion that different participants categorize the content differently but with enough commonality to understand each other (Harper *et al.*, 2003). The card sorting method is typically used to understand how users classify and structure the content of interactive systems, particularly websites. The aim is to produce an IA that supports findability.

There are two primary alternatives, open and closed card sorting. In an open card sorting, each participant is given a stack of cards. The participants are then asked to group those cards together in any way they want. Finally, they create labels for the groups that they chose. In a closed card sorting, the researchers create the labels for their respective groups. The participants are given a stack of cards and are asked to put each card into a group. The focus of this paper is on the analysis of open card sort data.

A typical step-by-step roadmap to effectively apply the open card sorting method is described in Paea *et al.* (2020). Various methodological issues related to conducting an open card sort have been explored in the literature. Previous research has shown that open card sorts provide reliable results with 20 to 30 participants (Tullis and Wood, 2004) or even with 10 to 15 participants (Lantz *et al.*, 2019). The number of cards to be sorted should be between 30 and 100 so that the participants can have enough items to form groups and the sorting is not tiring for them (Spencer, 2009). For a large set of cards (e.g. 100), Tullis and Wood (2005) propose a technique in which each participant only sorts a randomly selected subset of the full set of cards. If each participant sorts 60% of the full set and 30–40 participants are involved, then the obtained results are highly similar to sorting done on the full set of cards. Chaparro *et al.* (2008) examined the usability of electronic card sorting programs from the perspective of the researcher and the end-user.

Recently, the reliability of the open card sorting method has been studied. Katsanos *et al.* (2019) presented an empirical evaluation of the method's cross-study reliability. Six card sorts involving 140 participants were conducted: three open sorts for a travel website and three for an e-shop. Their findings support the cross-study reliability of card sorting. A recent

study (Pampoukidou and Katsanos, 2021) found that open card sorting also has a high test–retest reliability. The study involved the same participants performing open card sorts twice with a time interval of 15–20 days for three websites: an e-shop, a travel and tourism website and a university website. The results showed that the participants provided significantly and highly similar groupings and labels between the two card sorts per domain.

The main quantitative data from an open card sort is a similarity score per pair of cards (Righi *et al.*, 2013). Two cards would have 100% similarity if all participants grouped them together, whereas they have 0% similarity if no users categorized them together. These scores are usually organized in an N × N similarity or dissimilarity/distance matrix, where N is the number of cards, and each cell contains a similarity or dissimilarity/distance score. Methods to quantitatively analyze this matrix, such as hierarchical cluster analysis (Spencer, 2009), k-means clustering (Spencer, 2009; Paea and Baird, 2018) and multidimensional scaling (Paea and Baird, 2018), are typically used to make sense of the card sort groupings. Each method claims to determine an optimal solution in its own way.

Nawaz (2012) work draws attention to the choice of card sorting analysis and techniques and shows how it impacts the results. His research focuses on three analysis methods: actual merge method (AMM), BMM and edit distance. His study concludes that it is important to understand the methodological issues for tools supporting card sort analysis. Katsanos *et al.* (2008) introduced a computational tool, AutoCardSorter, that uses the semantic similarity between words, phrases and passages of content items and hierarchical cluster analysis to develop a website IA. In three validation studies, they found that AutoCardSorter proved approximately 17 times faster compared to an open card sort without expense in the quality of results.

Paea and Baird (2018) applied the combination of the K-means algorithm and multidimensional scaling to derive an IA from an open card sorting dataset. They found that their algorithm worked well for small card sorts of less than 30 participants and 30 cards. Their method was compared with BMM, AMM and participant-centric analysis. One of the contributions and motivations of the new algorithm proposed in this paper is that it overcomes the limitation of Paea and Baird (2018) algorithm by providing a more robust algorithm that works well for any number of participants and number of cards. The new proposed algorithm also provides strong quantitative meanings for open card sorting datasets.

2.2. K-means method

K-means is a popular partitional clustering algorithm (Likas *et al.*, 2003). K-means intends to partition n objects into k categories in which each object belongs to the category with the nearest centroid (mean) (Fränti and Sieranoja, 2019). Each category is represented by an adaptively changing centroid, starting from some initial values named seed points. K-means computes the squared distances between the input data points (inputs) and centroids and assigns inputs to the nearest centroid. The Euclidean distance is selected as the similarity index, and the produced categories minimize the sum of the squares of the various types (Huang, 1998; Likas *et al.*, 2003).

The algorithm starts by partitioning the input points into k initial sets, either at random or using some heuristic data. It then calculates the centroid of each set and constructs a new partition by associating each point with the closest centroid. Then, the centroids of each set are recalculated, and the algorithm is repeated by alternate application of these two steps until convergence, which is obtained when the points no longer switch categories or centroids are no longer changed. One important parameter of the algorithm is determining the central point of each category, which depends on the choice of the number of categories k. The best number of categories k leading to the most significant separation (distance) is not prior known and must be computed from the dataset. The next sections discuss existing techniques to determine the optimal number of categories k and algorithm initialization variations.

2.3. Techniques to determine the number of categories k

Although K-means has been widely used in data analysis and pattern recognition, it has three major limitations (Žalik, 2008). One of them is that the number of categories must be predetermined and fixed. The number of categories should match the data. Various methods have been proposed to determine the number of categories, and they are briefly described in the following sections: (i) eigenvalue-one criterion, (ii) scree plot (eigenvalue and percentage of variance), (iii) elbow method, (iv) gap statistic method, (v) silhouette method and (vi) 3D category view (3DCV)-average method.

2.3.1. *Eigenvalue-one criterion*

One of the most commonly used criteria for determining the number of categories is the eigenvalue-one criterion, also known as the Kaiser criterion (Kaiser, 1960). Katsanos *et al.* (2008) used the eigenvalue-one criterion to identify the optimal number of categories while analyzing open card sort datasets. This method identifies the optimal number of categories in terms of variance explained by implementing an eigenvalue analysis of the similarity matrix and keeping only the eigenvalues greater than one.

2.3.2. *Scree plot (eigenvalue and percentage of variance)*

Another method for determining the number of factors to retain is the scree plot (Cattell, 1966). A scree plot provides a good graphical representation of the ability of the principal component analysis to explain the variation in the data (Cattell, 1966). The scree plot can be produced either by plotting the eigenvalue against the number of categories or by plotting

the percentage of variance explained against the number of categories. According to this criterion, the significant factors are disposed like a cliff, having a big slope while the trivial factors are disposed at the base of the cliff. Nevertheless, this method is considered to be very subjective because the curve's cut-off point is sometimes not clear.

2.3.3. *Elbow method*

The elbow method (Bholowalia and Kumar, 2014; Syakur *et al.*, 2018) is a method that looks at the sum of the squared error (SSE) explained as a function of the number of categories. This method relies on the idea that one should choose a number of categories so that adding another category does not give much better modeling of the data. The percentage of variance (within-category sum of squared errors) explained by the categories is plotted against the number of categories. The first category will add much information, but the marginal gain will drop dramatically and give an angle in the graph. The k number of categories is chosen at this point according to this method.

2.3.4. *Gap statistic method*

Tibshirani ***et al.*** (2001) proposed another method for deciding the number of categories called gap statistic. The gap statistic method compares the total within intra-category variance for different values of k (number of categories) with the expected values under the dataset's null reference distribution. After categorizing the dataset for different values of k, we get the intra-category variance for the observed dataset as well as the reference dataset (uniform random reference datasets over the range of the observed data are generated) and then calculate the gap statistic as shown in Appendix 1.

2.3.5. *Silhouette method*

The silhouette method (Rousseeuw, 1987; Kaufman and Rousseeuw, 1990) is another well-known method with decent performance to estimate the potential optimal category number. This method uses the average distance between one data point and others in the same category and the average distance among different categories to score the category result. This technique provides a graphical representation of how well each object lies within its category. For every item or point i, its silhouette $S(i)$ is calculated as shown in Appendix 1.The $S(i)$ value lies between -1 and 1. A value closer to 1 indicates that an object is better categorized, and if it is closer to -1 the object should be categorized into another neighboring category. If there are too many or too few categories, as may occur when a poor choice of k is used in the K-means algorithm, some categories will typically display many narrow silhouettes than the rest. Given that the silhouette width provides an evaluation of category validity, silhouette plots and averages may be used to determine the natural number of categories within a dataset.

2.3.6. *3DCV-average method*

The 3DCV algorithm used by OptimalSort, a well-known online card sorting tool, simply uses the average (mean) of the number of categories created by participants in the card sorts. This average A is calculated as shown in Appendix 1.

2.4. K-means algorithm initialization variations

Fränti and Sieranoja (2019) compared nine initialization techniques for K-means. They found that the maxmin heuristics (MaxMin and K-means++) were the best initialization techniques for K-means. In this paper, we compare the proposed algorithm against these two techniques. The traditional K-means algorithm that generates initial category centroids randomly is also compared with the proposed algorithm. Thus, three existing initialization methods are compared against the proposed algorithm: (i) random centroids (RC), (ii) furthest point heuristic (MaxMin) and (iii) K-means++. All the existing initialization methods used in this paper were repeated 100 times to reduce the errors. Then, we chose the lowest total within the SSE. These existing initialization methods are briefly discussed in the following and are delineated in the work of Fränti and Sieranoja (2019):

- RC: By far, the most common technique is to select k random data objects as the set of initial centroids (MacQueen, 1967). The rationale behind this method is that random selection is likely to pick points from dense regions, that is, points that are good candidates to be centered.
- K-means++: The K-means++ (Arthur and Vassilvitskii, 2007) is usually reported as an efficient approximation algorithm in overcoming the poor clustering problem with the standard K-means algorithm. K-means++ initializes the category centroids by finding the data objects that are farther away from each other in a probabilistic manner. In K-means++, the first category centroid is randomly assigned, and the next ones are selected such that the probability of choosing a point as centroid is directly proportional to its distance from the nearest, previously chosen centroid.
- Furthest point heuristic (MaxMin): The main idea of the MaxMin algorithm is to isolate the category centroids that are farthest apart. The algorithm selects an arbitrary point as the first centroid and then adds new centroids one by one. At each step, the next centroid is the point that is furthest (max) from its nearest (min) existing centroid (Gonzalez, 1985; Katsavounidis *et al.*, 1994). This method was originally developed as a 2-approximation to the K-center clustering problem. It should be noted that motivated by a vector quantisation application, Katsavounidis *et al.*'s (1994) variant takes the point with the greatest Euclidean norm as the first center.

2.5. The need for a K-means variation for open card sort data analysis

Quantitative analysis of open card sort data using partitional clustering algorithms is underexplored in the existing literature. One of the most common partitional clustering algorithms is the K-means algorithm (Likas *et al.*, 2003; Redmond and Heneghan, 2007). In the background, we discussed the K-means algorithm, briefly explained the techniques to determine the number of categories and discussed the existing initialization techniques. These are the gaps we identify in the existing literature:

1) The usage of the K-means algorithm to analyze card sorting datasets has yet to be widely applied.
2) The existing K-means algorithm variations have shown that the final category quality depends on the goodness of the initialization technique.
3) The challenge that arises in quantitative analysis of open card sort data is deciding the number of categories.

These gaps lead us to propose a new K-means algorithm variation that is appropriate for open card sort data analysis. In the following sections, we first describe a real-world open card sort dataset, and then compare the results of analyzing this dataset with existing K-means variations and our proposal.

3. METHODOLOGY

Section 3.1 explains the open card sort dataset used in this study. Section 3.2 presents the application of the selected known K-means initialization methods to our dataset.

3.1. Open card sort dataset

The open card sorting was used to redesign the IA of the learning content items of a university course on mathematics available through the Moodle page. Fifty content items were chosen for the card sort. The moodle team and the researchers had close consultation and decided on the 50 cards. Examples of the cards chosen are the following: 'Course tour', 'Assignments', 'Practice quizzes', 'Case studies', 'Labs', 'Class News and Announcements', 'Contact us', 'Campus life', 'Course syllabus' and '"Past test papers'. The titles of all 50 cards are provided in Appendix 2. The research office of the university gave ethical approval for the study. The participants were selected from third-year students, postgraduate students and staff doing and teaching mathematics courses at a regional university. This study recruited 112 (56 men and 56 women) participants currently working and/or studying at the university. The sample age range varies from 21 to 48 years ($M = 28$ and $SD = 6$). All participants had at least 2 years of experience using the mathematics Moodle page.

Due to COVID-19 restrictions, an online card sorting was used using the Desmos Card Sorting Activity tool. A demonstration video and the information sheet were shared with the participants beforehand for ease of reference regarding the purpose and process of card sorting. The participants signed the consent form on the day of the actual card sorting, and they performed open card sorting with the researcher(s) online presence for meaning-making and clarification purposes. This was arranged through Zoom and Big Blue Button (BBB) sessions. A link was sent to the participants for the online card sorting. There was no limitation set on the number of cards for each group, and participants were free to create as many groups as they wanted.

Some participants created three categories only, while others created more complex classifications involving up to 15 categories ($M = 6$, $SD = 2$). There were no significant differences between the number of categories formed by males ($M = 6$) and females ($M = 6$), $t(6) = 0.687$, *ns*, and the number of categories formed were unrelated to age ($r = -0.29$, *ns*). Figure 1 shows the distribution of the number of categories sorted by 112 participants. The participants created a total of 749 categories with a median of 6 categories and a mean of 6 categories.

3.2. Known initialization methods for K-means applied to our dataset

We applied three known initialization methods for K-means to our open card sorting dataset: (i) RC, (ii) MaxMin and (iii) K-means++. We used a sampling without replacement method to guarantee that we do not select the same card twice regarding the centroids' initialization methods. The selection is independent of the order of the data.

For all three K-means variations, we took the following steps to get our results:

1. Randomly choose a specific number of cards from the data to be the initial centroids. This number is equal to the optimal number of categories detected by the proposed algorithm (see Section 4.1).
2. Determine the initial centroids. The initial centroid is randomly assigned from the existing data, and the number of categories is equal to the number of initial centroids.
3. Perform the K-means variation using these initial centroids. From the results, we then find out the total within-category sum of squared errors (TWSSE).
4. Repeat steps 1 to 3 for another 99 times (total runs is 100) and get their respective TWSSE.
5. Compare all the TWSSE and choose the one that is the smallest as the best category result.
6. Once we know the best category result from the previous step (5), we can then check which card was its initial centroid in each of the categories.
7. Finally, plot the best category result, the final centroids and the initial centroids as well.

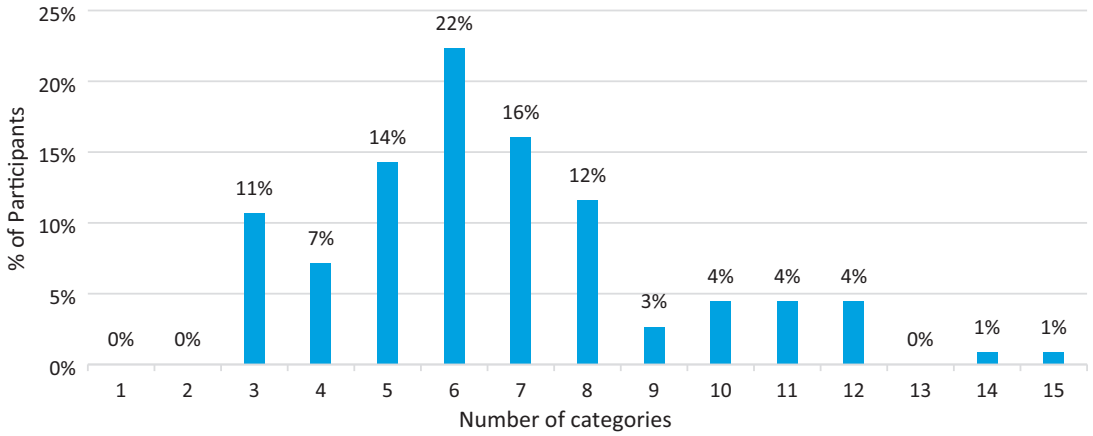


FIGURE 1. Number of categories created by 112 participants in our open card sort dataset.

4. PROPOSED ALGORITHM

The K-means algorithm is very popular because of its ability to categorize any kind of data and outliers quickly and efficiently. However, as it has been aforementioned, it mainly suffers from two challenges: (i) the optimal number of categories k should be known as an input and (ii) the quality of the results is susceptible to the designated initial starting points as category centroids. This section describes the proposed algorithm called the BMK-means algorithm, which provides a solution to these challenges.

4.1. Determine the number of categories k

First, the proposed algorithm uses six methods to determine the optimal number of categories (see Section 2.3 for a brief discussion of the methods) for the given open card sort dataset: (i) eigenvalue-one criterion, (ii) scree plot (eigenvalue and percentage of variance), (iii) elbow method, (iv) gap statistic method, (v) silhouette method and (vi) 3DCV-average method. Then, the algorithm chooses the k number of categories that was most often found by the aforementioned approaches. The application of this step on our card sorting dataset is discussed in the following:

1) Eigenvalue-one criterion

Table 1 shows that only the first six components have eigenvalues greater than one. So based on this proposal, six categories explaining 66.27% of the total variance are retained for this dataset.

2) Scree plot (eigenvalue and percentage of variance)

Figure 2 presents scree plots of our open card sort data. The scree plot (see Fig. 2a) shows that there is one break when the number of categories is six and then the line begins to flatten out. Figure 2(a,b) suggests that 6 categories should be used based on this method.

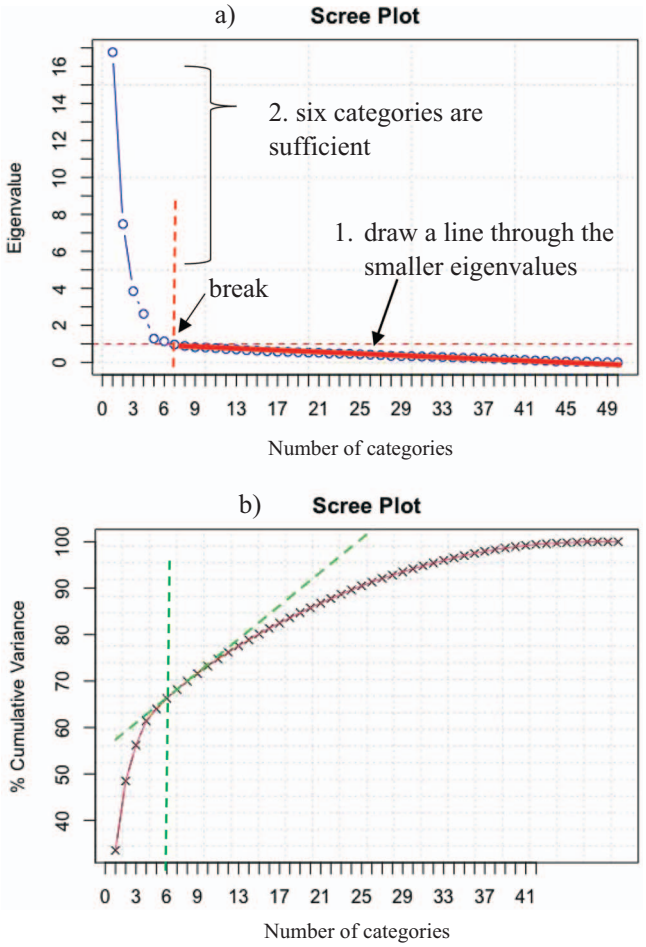


FIGURE 2. Determining the optimal number of categories for our card sort dataset. (a) The scree plot for the initial variables. (b) The scree plot for the cumulative variance

TABLE 1. Eigenvalues, percentage of the variance and cumulative percentage of variance for our card sort dataset.

Component/category	Initial eigenvalues			Extraction sums of squared loadings		
	Total	% of variance	Cumulative %	Total	% of variance	Cumulative %
1	16.766	33.531	33.531	16.766	33.531	33.531
2	7.474	14.948	48.479	7.474	14.948	48.479
3	3.849	7.698	56.177	3.849	7.698	56.177
4	2.621	5.242	61.419	2.621	5.242	61.419
5	1.287	2.574	63.993	1.287	2.574	63.993
6	1.138	2.276	66.269	1.138	2.276	66.269
7	0.96	1.92	68.189			
8	0.888	1.776	69.965			
9	0.818	1.637	71.601			
10	0.807	1.614	73.216			
11	0.764	1.527	74.743			
12	0.722	1.443	76.187			
13	0.697	1.394	77.58			
14	0.654	1.309	78.889			
15	0.63	1.26	80.149			
16	0.6	1.199	81.349			
17	0.572	1.144	82.493			
18	0.564	1.128	83.621			
19	0.543	1.086	84.707			
20	0.523	1.046	85.754			
21	0.517	1.034	86.788			
22	0.482	0.964	87.751			
–	–	–	–			
–	–	–	–			
–	–	–	–			
50	0	0	0			

3) Elbow method

Figure 3 presents the results of the elbow method for our card sort dataset. A sharp decrease is observed at $k = 6$, which is the optimal number of categories according to this method.

4) Gap statistic method

Figure 4 presents the results for the gap statistic method on our open card sort dataset and shows that the optimal number of categories k is two.

5) Silhouette method

Using the silhouette width method, Figure 5 shows that the optimal number of categories k is 18.

6) 3DCV-average method.

Figure 1 shows that the participants created a total of 749 categories with a mean of 6 categories. Therefore, the number of categories is six by using the 3DCV-average method.

TABLE 2. Optimal number of categories from six methods used on our open card sort dataset.

Number	Method name	k-value
1	Eigenvalue-one criterion	6
2a	Scree plot (eigenvalue)	6
2b	Scree plot (percentage of variance)	6
3	Elbow method	6
4	Gap statistic method	2
5	Silhouette sethod	18
6	3D category view (3DCV)—average method	6

7) Summary—determine the number of categories k

Table 2 summarizes the number of categories provided by all the methods mentioned above. Most methods chose six categories; thus, the proposed BMK-means algorithm chooses six categories as the optimal number k for the dataset in this study.

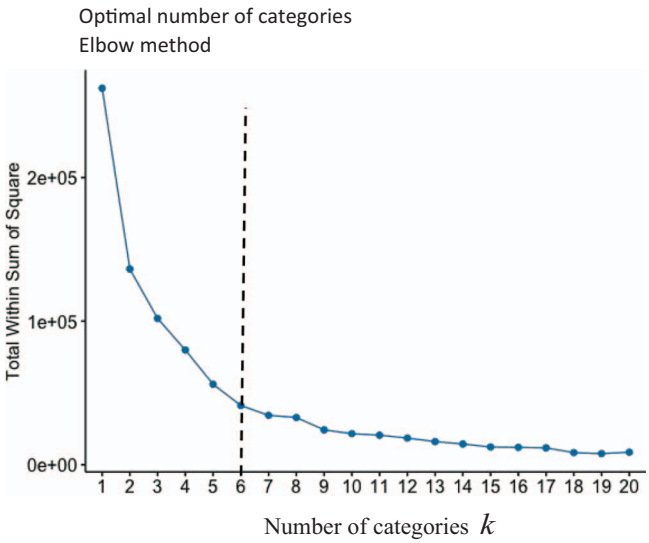


FIGURE 3. Determining the optimal number of categories k for our open card sort dataset using the elbow method.

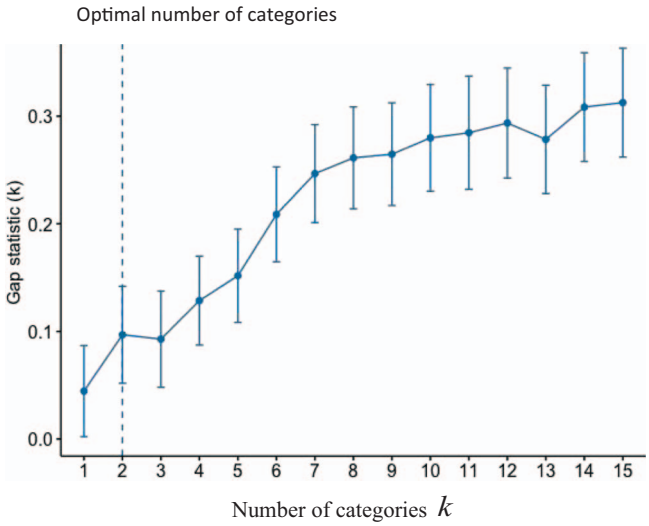


FIGURE 4. Determining the optimal number of categories k for our open card sort dataset using the gap statistic method.

4.2. BMM initialization

BMM (OptimalSort) is a method that can be used to examine how categories are merged in hierarchical category analysis. Appendix 3 presents the main steps of the BMM algorithm along with two experiment scenarios that explain the BMM steps and calculations.

The BMK-means algorithm locates the initialization centers based on two design values, the categories' size, the participant's agreement % and how the card merges. The two design values that depend on the dataset are the following: (i) k the optimal number of categories for the K-means algorithm

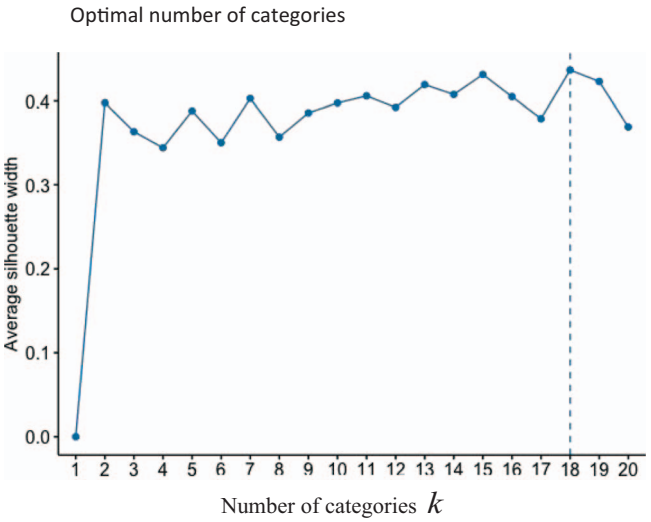


FIGURE 5. Determining the optimal number of categories k for our open card sort dataset using the silhouette method.

(see Section 4.1), and (ii) t the threshold on the dendrogram produced by the BMM. Figure 6 presents an example of the BMM dendrogram with a fixed threshold t value (red dash line), the initialization centroid cards (open black circles), the category size and names (blue box), the participant's agreement (0–100%) and the card merge.

Given a dendrogram (Fig. 6) produced using the BMM, the BMK-means algorithm calculates the optimal number of categories using the following formula at a particular t value:

$$k = \sum_{i=0}^{n} k_{n-i}, \tag{1}$$

where n is the highest number of cards in a category, k_n represents one category with maximum n cards and k is the optimal number of the categories at a t value. For instance, if $t = 48\%$ in Fig. 6, then there are 10 categories, with each category containing two or more cards. Starting from the top of Fig. 6, examples of these are: [Questionnaire, Surveys] with two cards, [Read me first, Read an article, Required Readings] with three cards, [Discussion forums, Chat room] with two cards, [Web conferencing/webinar, Youtube, Software Mobile applications] with four cards etc. Thus, Equation (1) calculates the $k$value as

$$k = \sum_{i=0}^{n} k_{n-i} = k_2 + k_3 + k_2 + k_4 + k_7 + k_2 + k_4 + k_9 + k_2 + k_5 = k_3 + k_4 + k_7 + k_4 + k_9 + k_5 = 6.$$

One of the algorithm criteria is to count the number of categories starting from the category containing the highest number of cards ($k = k_9 + k_7 + k_5 + k_4 + k_4 + k_3 = 6$). When $k$the

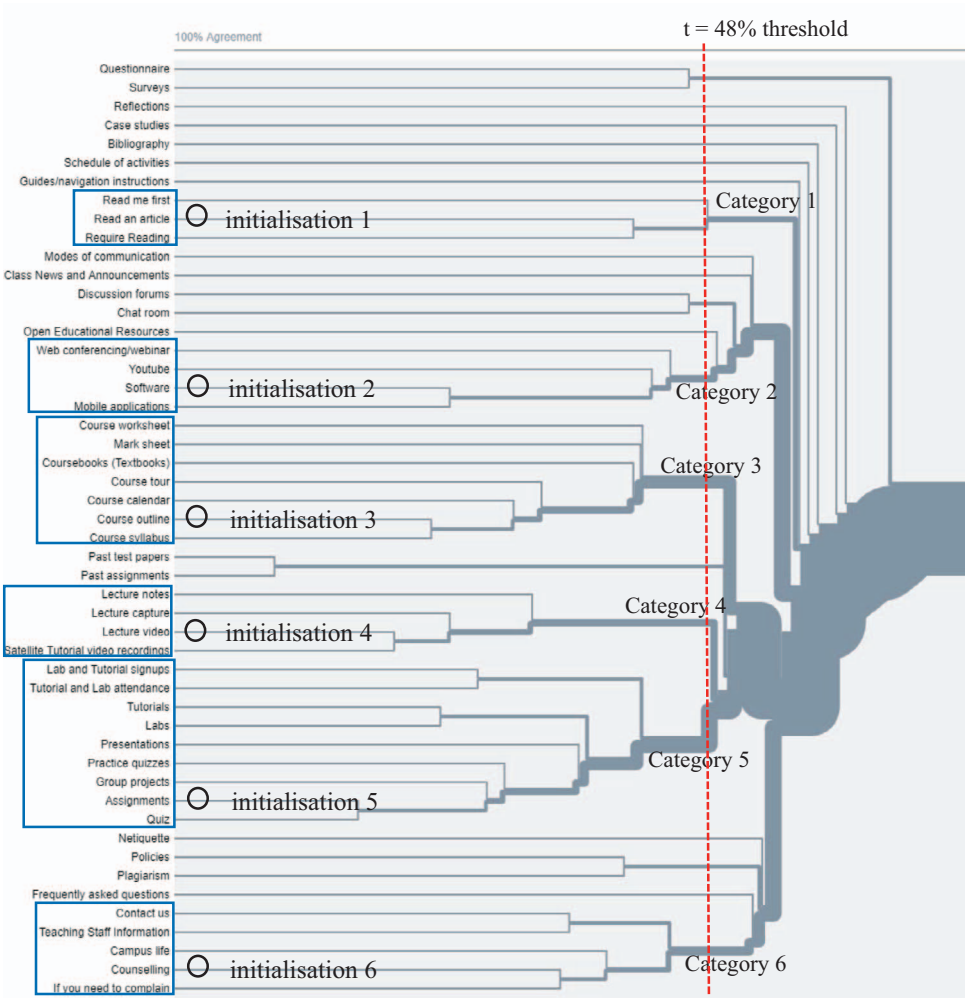


FIGURE 6. The BMM dendrogram (OptimalSort) and the result of BMK-means algorithm. The thicker the lines, the more cards are merged together. The results are from the dataset of this study involving 112 participants who completed an open card sort with 50 cards. BMK-means found that the optimal number of categories k for this dataset is 6, and this is why there are six initialization cards. BMK-means set the threshold t on the dendrogram to 48% participant's agreement, which results in initial categories that contain at least three cards.

value from Equation (1) equals the calculated k value in Section 4.1, the algorithm stops counting and disregards the rest of the categories (in this example, it disregards four categories with two cards). For reading purposes, the categories are numbered based on their order of appearance from top to bottom in Fig. 6.

Table 3 sketches the steps of the BMK-means algorithm applied when selecting the initialization center cards in Fig. 6 (black open circle) and Table 4.

Table 4 shows the six initialization center cards, the number of cards and the card names in each chosen category of Fig. 6.

It is argued that the proposed algorithm can make K-means more efficient and, more importantly, to provide categories that are more congruent to participants' groupings when analyzing open card sorting datasets. In addition, BMK-means overcomes the limitation reported in Paea and Baird (2018), who proposed an algorithm that combines Multidimensional Scaling (MDS) and K-Means clustering to analyze open card sorting datasets. They mention that their algorithm does not work well when the number of participants and the number of cards are greater than 30. In addition, the BMK-means algorithm has more advantages than Fred and Jain's (2002) method because the proposed algorithm calculates the optimal number $k$of categories before using the dendrogram to locate the initial centers.

5. EXPERIMENTS METHODOLOGY

In the next sections, we study the overall performance of the different K-means algorithm variations using the following metrics: (i) the participant's agreement score (PAS), (ii) category validity score (CVS) and (ii) Initial and final centroids distance (IFCD). This section presents these metrics, and Sec-

TABLE 3. The steps of the BMK-means algorithm used when selecting the initialization center cards.

1) Begin with the choosing of a threshold t value:
   The algorithm will look at a t value (Fig. 6) that contains the optimal number of categories k presented in Section 4.1 for the dataset using the following steps:
   i. The algorithm starts by moving the dashed vertical line (threshold) from the right (0% agreement) to the left (100% agreement) side of the BMM dendrogram (see Fig. 6 for an example);
   ii. For a specific t value, the algorithm searches for the categories that contain two or more cards using Equation (1);
   iii. Repeats the process in steps (i) and (ii) and stores the results in memory until the dashed line reaches 100% participant's agreement;
   iv. Compares all the calculated k values in (iii) and chooses the k value that each category contains the highest number of cards (see the example above). Following steps (i) to (iv) in this study's dataset, six categories lead to a threshold $t = 48\%$ participant's agreement with three or more cards in each category (categories 1 to 6 in Fig. 6);
   v. Suppose step (iv) contains more categories than the calculated k value due to more similar categories of the same lowest number of cards. In this case, the algorithm will include a category with the closest next merge to t value (dashed vertical line) from the right side. The new category must not be part of any previously chosen categories.
2) The chosen categories must equal the number of k where the initialization centers are located;
3) Pick the strongest pair with the highest participant's agreement in a chosen category (the pair that is closest to 100% participant's agreement in Fig. 6).
4) Select one card from the strongest pair in (3) that is grouped with the next strongest card positioned along the right edge of the similarity matrix (Fig. 7) as a starting center of the category. For instance, the black open circle initialization 3 of category 3 in Fig. 6 and the initialization 3 highlighted by the box in Fig. 7. The strongest pair in category 3 is 'Course syllabus' and 'Course outline'. The algorithm chose 'Course outline' as the starting center of category 3 because 'Course outline' pairs with the next related strongest card 'Course calendar' (see Fig. 7). This process repeats for the rest of the categories to find the initialization cards.

TABLE 4. Number of cards from the $k = 6$ categories and the six initial card centers

Category number	Initialization card name	Number of cards
1	Read an article	3
2	Software	4
3	Course outline	7
4	Lecture video	4
5	Assignments	9
6	Counseling	5

tion 6 presents the results of the comparative analysis of the proposed BMK-means algorithm and the three existing K-means initialization methods (RC, MaxMin and K-means++).

5.1. Metrics used to compare the algorithms

All three metrics used to compare the proposed algorithm against existing ones are related to the similarity matrix. As previously mentioned, the similarity matrix is a simple representation of how frequently two cards were placed together by open card sort participants. Figure 7 shows one way to visualize the similarity matrix for our card sorting dataset. The darker the blue, where two cards intersect, the more often the participants paired them together. Figure 7 shows that the strongest pair is placed in the top left corner, grouping them with the next related strongest pair that either of those cards have, and the process repeats for the new pair. This way, categories of cards that are strongly related to each other appear together in the same shade of blue on the matrix. Paea and Baird (2018) and Paea *et al.* (2020) discuss the similarity matrix in more detail.

5.2. Participant's agreement score

PAS measures the degree of participants' agreement between pairs. We calculate the PAS of each category by summing all the percentages from the combination of cards (cells) in a category and then dividing by the number n of combination cells. However, this is probably biased by the number of elements within a category. The elements belonging to the smallest categories tend to have a lower agreement. This bias could be corrected by including 100% into the denominator of the formula. The 100% indicates that all participants agree to pair two cards together.

The similarity matrix is a square $m \times m$ matrix, where m represents the number of cards. Each cell $C_{i,j}$ represents the number of times the card i and the card j have been categorized into the same group by participants. Given a partition of the elements (calculated, for instance, with the K-means algorithm), we can calculate for each category the participant's agreement using the following formula:

$$\mathrm{PAS}(A) = \frac{\sum_{I=1}^{I \subset A} C_{i,j}}{n \times 100\%},$$

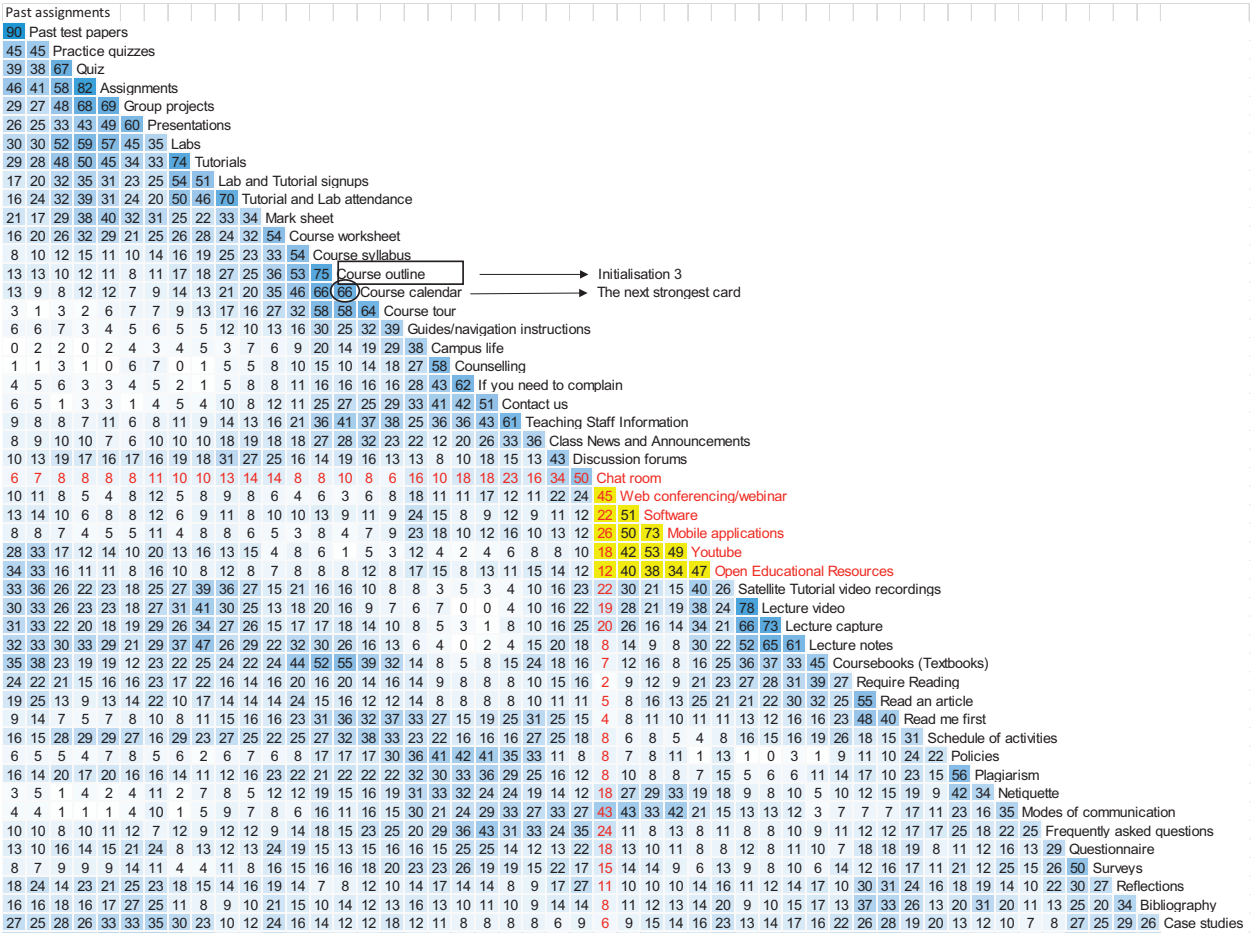


FIGURE 7. The similarity matrix (OptimalSort) displays how many participants agree with each pair from the combination of cards. The results are from the 112 participants who submitted card sorts for a total of 50 cards.

where PAS(A) is the PAS of a category A, $I \subset A$ are the elements that belong to the A category and n is the number of the combination cards (cells) in$A$category. The algorithm, therefore, sums all the cells of the elements which belong to the same category of k, then divides by $n \times 100\%$. The final step is to sum all the category participant's agreements. An example is provided to help to explain the calculation better in Appendix 4.

5.3. Category validity score

Given a partition of the cards (e.g. calculated using the k-means algorithm), we can calculate for each element the category validity using the following formula:

$$\mathrm{CVS}\,(k \subset A) = \frac{\sum_{i \neq k}^{I \subset A} C_{k,i}}{n \sum_{i \neq k}^{I \subset M} C_{k,i}},$$

where CVS(k) is the category validity of the card k, $I \subset A$ are the elements that belong to the same A category of k (except k itself), $I \subset M$ are all the elements (except, again, k) and n is the number of cards in$A$category (Bussolon, 2009). The algorithm sums all the cells of the elements that belong to the same category k (except the diagonal value $C(k,k)$) divided by the sum of all the given k row cells (except the diagonal value $C(k,k)$). We include the size of the category (n) into the denominator of the formula to overcome the bias by the number of elements within a category especially if a category contains a small size. An example is provided to help explain the calculation better in Appendix 5.

5.4. Initial and final centroids distance

The distance between the initial and final centroids was calculated using the distance formula $d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$. It shows how far the final centroid moved away from the initial center. For the three existing methods, we chose the lowest SSE found after 100 repetitions. The proposed algorithm requires

TABLE 5. The proposed categories for our card sorting dataset using the BMK-means algorithm.

Level 1	Card numbers	Category PAS	CVS	IFCD
Group 1	9 Cards20, 25, 42, 7, 48, 16, 24, 21, 47	833/3600 = 0.231	0.211	15.33
Group 2	6 Cards28, 22, 10, 50, 31,37	600/1500 = 0.4	0.26	7.92
Group 3	7 Cards13, 14, 38, 4, 27, 3, 1	917/2100 = 0.437	0.262	18.20
Group 4	6 Cards36, 11, 44, 43, 19, 49	635/1500 = 0.423	0.214	11.02
Group 5	9 Cards8, 18, 45, 26, 39, 23, 15, 46, 12	1626/3600 = 0.452	0.36	16.72
Group 6	13 Cards35, 29, 33, 34, 32, 2, 41, 5, 6, 30, 9, 17, 40	2345/7800 = 0.301	0.443	24.27
		Total = 2.244	Total = 1.75	Total = 92.47

The card names that correspond to the card numbers in the second column are presented in Appendix 2.

TABLE 6. The proposed categories for our card sorting dataset using the K-means random initialization algorithm.

Level 1	Card numbers	Category PAS	CVS	IFCD
Group 1	5 Cards13, 14, 38, 4, 3	523/1000 = 0.523	0.21	8.23
Group 2	13 Cards35, 29, 33, 34, 32, 2, 41, 5, 6, 30, 9, 17, 40	2345/7800 = 0.301	0.443	21.71
Group 3	6 Cards27, 42, 45, 48, 1, 46	486/1500 = 0.324	0.163	23.05
Group 4	6 Cards28, 37, 22, 10, 50, 31	600/1500 = 0.4	0.26	20.19
Group 5	13 Cards8, 18, 26, 36, 39, 11, 44, 43, 23, 19, 15, 49, 12	3096/7800 = 0.397	0.477	34.01
Group 6	7 Cards20, 25, 7, 16, 24, 21, 47	546/2100 = 0.26	0.184	46.81
		Total = 2.205	Total = 1.737	Total = 154.00

The card names that correspond to the card numbers in the second column are presented in Appendix 2.

only one iteration due to its centroid points being already identified. This is one additional advantage of the proposed algorithm compared to the existing methods running times.

6. EXPERIMENTAL RESULTS

Four simulated experiments were carried out to test how the K-means algorithm with four different initialization methods (proposed one v three existing ones) perform on a real-world card sorting dataset. The results per experiment are reported in the following. To this end, a table is used per experiment that includes these columns: (i) the primary level group number in the first column, (ii) a list of the card numbers in the second column. The card names that correspond to these card numbers are presented in Appendix 2, (iii) the category PAS in the third column, (iv) the CVS in the fourth column and (v) the distance between the final and initial centers in the fifth column. The findings from all four simulated experiments are summarized and discussed at the end of this section.

6.1. Experiment 1: BMK-means algorithm result

The key findings from using the BMK-means algorithm on our open card sort dataset are presented in Table 5.

6.2. Experiment 2: K-means random initialization

Table 6 shows the results from using the K-means random initialization algorithm on our open card sort dataset.

6.3. Experiment 3: K-means MaxMin initialization

Table 7 shows the results from using the K-means MaxMin algorithm on our open card sort dataset.

6.4. Experiment 4: K-means++ initialization

Table 8 shows the results from using the K-means++ algorithm on our open card sort dataset.

6.5. Summary and comparisons

We summarize our main findings in Table 9 with the method names in the first column, the number of algorithm repetitions of each method in the second column, the total category PAS in the third column, the total CVS in the fourth column and the total distance score in the fifth column.

This section compares the proposed algorithm (BMK-means) with the three existing methods to provide valuable insights into which method obtains better category quality.

6.5.1. Participant's agreement score

Figure 8 shows the total value of the PAS in each method. Measuring each method's strength depends on the total value of the PAS—the larger the total score, the better the technique. Figure 8 shows that the BMK-means algorithm has the highest total score (2.244) compared to the others. This finding

TABLE 7. The proposed categories for our card sorting dataset using the K-means MaxMin algorithm.

Level 1	Card numbers	Category PAS	CVS	IFCD
Group 1	11 Cards20, 28, 7, 37, 22, 16, 24, 21, 10, 50, 27	1270/5500 = 0.231	0.292	81.99
Group 2	7 Cards8, 18, 26, 39, 23, 15, 12	1053/2100 = 0.501	0.299	10.40
Group 3	8 Cards25, 27, 42, 45, 48, 47, 1, 46	727/2800 = 0.260	0.186	16.05
Group 4	6 Cards36, 11, 44, 43, 19, 49	635/1500 = 0.423	0.214	47.65
Group 5	5 Cards13, 14, 38, 4, 3	523/1000 = 0.523	0.21	56.27
Group 6	13 Cards35, 29, 33, 34, 32, 2, 41, 5, 6, 30, 9, 17, 40	2345/7800 = 0.301	0.443	79.62
		Total = 2.239	Total = 1.644	Total = 291.99

The card names that correspond to the card numbers in the second column are presented in Appendix 2.

TABLE 8. The proposed categories for our card sorting dataset using the K-means++ algorithm.

Level 1	Card numbers	Category PAS	CVS	IFCD
Group 1	8 Cards20, 25, 42, 7, 48, 24, 21, 47	662/2800 = 0.236	0.187	10.36
Group 2	7 Cards29, 32, 30, 16, 3, 17, 40	475/2100 = 0.226	0.158	6.66
Group 3	15 Cards8, 18, 45, 26, 36, 39, 11, 44, 43, 23, 19, 15, 49, 46, 12	3962/10500 = 0.377	0.53	15.14
Group 4	8 Cards35, 33, 34, 2, 41, 5, 6, 9	999/2800 = 0.357	0.275	38.89
Group 5	6 Cards13, 14, 38, 4, 27, 1	727/1500 = 0.485	0.241	18.31
Group 6	6 Cards28, 37, 22, 10, 50, 31	600/1500 = 0.4	0.26	13.65
		Total = 2.081	Total = 1.651	Total = 103.01

The card names that correspond to the card numbers in the second column are presented in Appendix 2.

TABLE 9. Summary of the main findings.

Method name	Number of algorithm repetitions	Total category PAS	Total CVS	Total IFCD
BMK-means	1	2.244	1.750	92.47
Random K-means	100	2.239	1.737	154
MaxMin K-means	100	2.239	1.644	291.99
K-means++	100	2.081	1.651	103.01

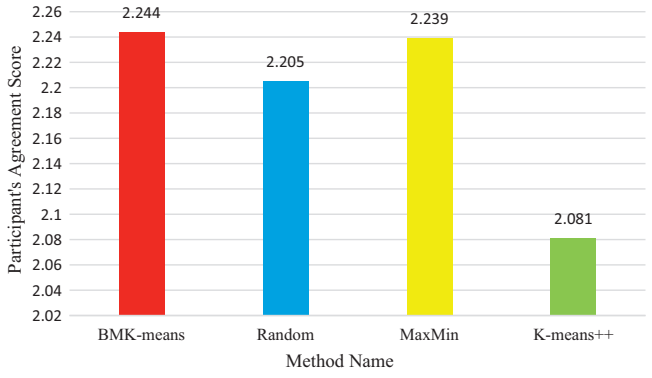


FIGURE 8. Total values of the PAS in the four methods.

shows that the BMK-means algorithm does combine the most similar cards in the resulting categories. Similar cards were selected based on their relationship and closeness depicted by the participant's agreement. The combination of cards in a category plays a role in the PAS, as seen in Fig. 9, which shows the PAS distribution in each discrete category and its corresponding number of cards. Figure 9 indicates that the PAS does depend on the combination of cards and the distribution of the number of cards in a category is important. As seen in Fig. 9, the red color bar is concentrated in the middle compared to the other.

In addition, we calculate the total number of similar cards (Tables 5, 6, 7 and 8) sorted under the same category. For instance, Table 10 shows that BMK-means and MaxMin have the highest number of similar cards (82%) sorted under the same groups. The second highest is BMK-means and Random K-means (78%), and the lowest is BMK-means and K-means++. An example is provided to help explain the calculations better (see Appendix 6). Table 10 further supports the findings of Fig. 8 that the BMK-means algorithm has the highest total PAS, the second is MaxMin, the third is Random and the last is K-means++.

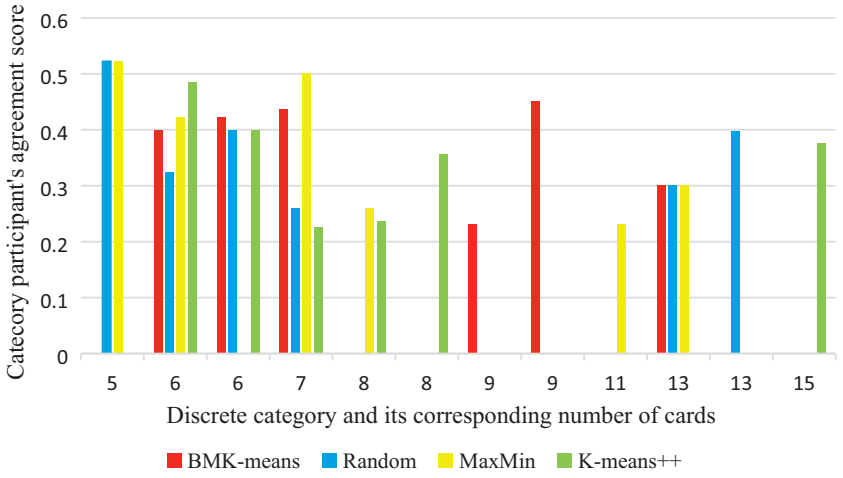


FIGURE 9. Total values of the PAS in six categories of the four methods.

TABLE 10. Number of cards (%) that have been placed in the same groups between two methods.

	BMK-means	Random	MaxMin	K-means++
BMK-means		78	82	74
Random	78		74	72
MaxMin	82	74		56
K-means++	74	72	56	

6.5.2. *Category validity score*

Figure 10 shows the total CVS of each method. The BMK-means algorithm has the highest total CVS (1.75). This means that the BMK-means algorithm categorizes the 50 cards into the resulting groups more cohesively. The CVS indicates the substantial similarity between a card and the combination of the other cards in a category. The combination of cards in a category plays a role in the CVS, as seen in Fig. 11, which shows the CVS distribution in each discrete category and its corresponding number of cards. Figure 11 indicates that the CVS does depend on the combination of cards and the distribution of the number of cards in a category is important. In Fig. 10, the BMK-means algorithm appears to be a robust method for analyzing cards meaningfully in relation to how the open card sorting data collection is being carried out compared to the other methods.

6.5.3. *Total distance score*

Figure 12 shows the total value of the distance between initial and final centroids (IFCD) in each method. Measuring each method's initialization depends on the total value of the IFCD score. This score measures how far the final centroid moved away from the initial center. Figure 12 shows that the BMK-means algorithm has the lowest total score (92.47) compared to

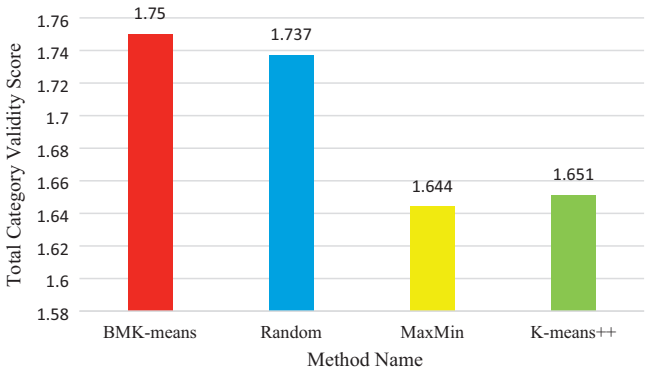


FIGURE 10. Total CVS of the four methods.

the existing methods. This means that BMK-means had the best initialization method compared to the other three approaches.

7. CONCLUSION

This paper presented BMK-means, an algorithm for computing initial category centers for the K-means method in the context of open card sort data analysis. In this algorithm, we first identify the optimal number of categories and then create the initial core categories using the BMM, identify the initial center, and finally apply the K-means to categorize the data. The rationale for the proposed algorithm is that the category results heavily depend on the goodness of the initialization technique. Indeed, study results showed that the quality of initial categories is critical and directly affects the final category quality.

The proposed algorithm is very effective and converges to better category results. In specific, experimental results show that the proposed algorithm compares favorably to other algorithms, obtaining better category quality as operationalized

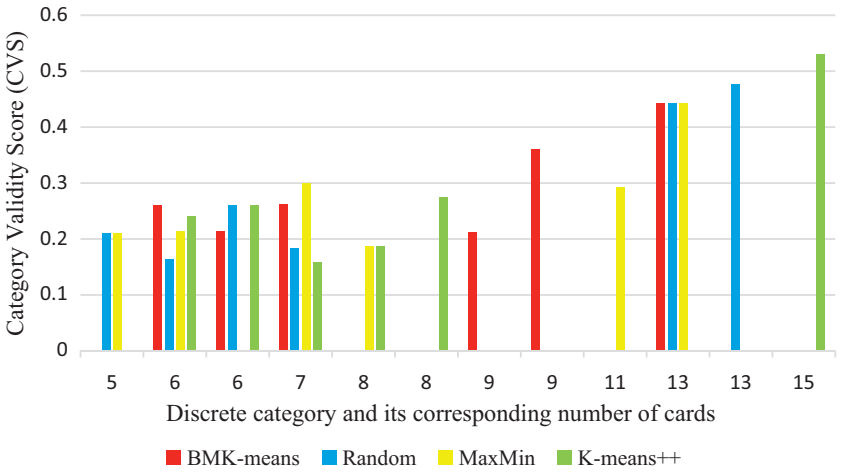


FIGURE 11. CVS in six categories of the four methods.

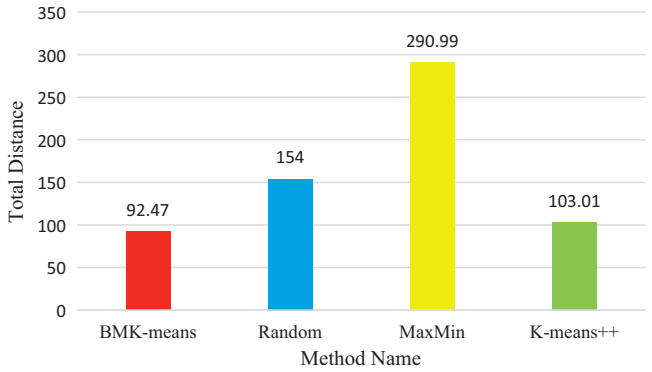


FIGURE 12. Total distance between the initial and final centroids of the four methods.

by the sum of each method total PAS and the total CVS. A method's strength depends on both the total PAS and the total CVS to have the highest scores. We also found that the relationship between CVS and PAS does not show much of anything happening since its correlation $r = 0.0459$ is very close to 0 in this dataset. Our analysis shows that the proposed algorithm has the least total distance between the initial and final category centers. This indicates that the proposed algorithm has the best initialization technique compared to the selected existing methods.

Our analysis shows that the proposed algorithm is closer (82% of cards have been placed in the same category by both techniques) to MaxMin initialization method than the others. The proposed algorithm also solved the limitation of Paea and Baird (2018) work and presented how to find the number of categories k in the open card sort dataset, which are both additional contributions of this work.

There is a need for more in-depth future research of components using qualitative data to provide deep and rich insights into open card-sorting findings. An extension of this research may be to enhance understanding of the topic from the MDS using the 3D and the 2D data points. One limitation of this work is that it currently relies on internal metrics of the category quality. A potential future research direction would be to conduct user testing of the structures produced by the four techniques. We could measure the three dimensions of usability according to ISO 9241: interaction effectiveness (e.g. task success), interaction efficiency (e.g. time on task) and users' perceived satisfaction (e.g. SUS score). If we find statistically significant differences in favor of our method, we can further support our claim that it is the best one for analyzing open card sort data.

ACKNOWLEDGMENTS

A special thank goes out to SC356 Mathematics Team 1 semester 2 2020 and all the participants who have offered their contribution to this study. Thank you for sharing your time and experience. We also thank Optimal Workshop (http://www.optimalworkshop.com) for allowing this work to use their algorithms (Similarity matrix and BMM).

SUPPLEMENTARY MATERIAL

Supplementary data is available at *Interacting with Computers* online.

REFERENCES

Arthur, D. and Vassilvitskii, S. (2007) k-means++: The advantages of careful seeding. In: *Proc. of the 18th annual ACM-SIAM symposium on discrete algorithms*, pp. 1027–1035.

Bholowalia, P. and Kumar, A. (2014) EBK-means: a clustering technique based on elbow method and k-means in WSN. *Int. J. Comput. Appl.*, 105, 17–24..

Bitan, Y., Parmet, Y., Greenfield, G., Teng, S., Cook, R. I. and Nunnally, M. E. (2019) Making sense of the cognitive task of medication reconciliation using a card sorting task. *Hum. Factors*, 61, 1315–1325.

Bussolon, S. (2009) Card sorting, category validity, and contextual navigation. *J. Inform. Arch.*, 1, 5–29.

Cattell, R. B. (1966) The scree test for the number of factors. *Multivar. Behav. Res.*, 1, 245–276.

Chaparro, B. S., Hinkle, V. D. and Riley, S. K. (2008) The usability of computerized card sorting: a comparison of three applications by researchers and end users. *J. Usability Stud.*, 4, 31–48.

Fränti, P. and Sieranoja, S. (2019) How much can k-means be improved by using better initialization and repeats? *Pattern Recogn.*, 93, 95–112.

Fred, A. and Jain, A. K. (2002) Evidence accumulation clustering based on the k-means algorithm. In *Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR)*, pp. 442–451.

Gonzalez, T. F. (1985) Clustering to minimize the maximum intercluster distance. *Theor. Comput. Sci.*, 38, 293–306.

Harper, M. E., Jentsch, F. G., Berry, D., Lau, H. C., Bowers, C. and Salas, E. (2003) TPL—KATS-card sort: a tool for assessing structural knowledge. *Behav. Res. Methods Instrum. Comput.*, 35, 577–584.

Huang, Z. (1998) Extensions to the k-means algorithm for clustering large data sets with categorical values. *Data Min. Knowl. Disc.*, 2, 283–304. https://doi.org/10.1023/A:1009769707641.

Jain, A. K., Murty, M. N. and Flynn, P. J. (1999) Data clustering. *Comput. Surveys*, 31, 264–323.

Kaiser, H. F. (1960) The application of electronic computers to factor analysis. *Educ. Psychol. Meas.*, 20, 141–151.

Kalbach, J. (2007) Designing Web navigation: Optimizing the user experience. O'Reilly Media, Inc., Newton, MA.

Katsanos, C., Tselios, N. and Avouris, N. (2008) Automated semantic elaboration of web site information architecture. *Interact. Comput.*, 20, 535–544.

Katsanos, C., Avouris, N., Stamelos, I., Tselios, N., Demetriadis, S. and Angelis, L. (2019) Cross-study Reliability of the Open Card Sorting Method [Paper presentation]. In *Conference on Human Factors in Computing Systems - Proceedings*, pp. 1–6. New Orleans, LA, USA.

Katsavounidis, I., Kuo, C.-C. J. and Zhang, Z. (1994) A new initialization technique for generalized Lloyd iteration. *IEEE Signal Process. Lett.*, 1, 144–146.

Kaufman, L. and Rousseeuw, P. J. (1990) *Finding groups in data: an introduction to cluster analysis*. John Wiley & Sons, New York.

Lantz, E., Keeley, J. W., Roberts, M. C., Medina-Mora, M. E., Sharan, P. and Reed, G. M. (2019) Card sorting data collection methodology: how many participants is Most efficient? *J. Classif.*, 36, 649–658. https://doi.org/10.1007/s00357-018-9292-8.

Likas, A., Vlassis, N. and Verbeek, J. J. (2003) The global k-means clustering algorithm. *Pattern Recogn.*, 36, 451–461.

Liu, Y. and Wickens, C. D. (1992) Use of computer graphics and cluster analysis in aiding relational judgment. *Hum. Factors*, 34, 165–178.

MacQueen, J. (1967) Some methods for classification and analysis of multivariate observations. *Proc. of the Fifth Berkeley Symposium on Math., Srar. and Prob*, 1, 281–296.

Morville, P. and Rosenfeld, L. (2006) Information architecture for the World Wide Web: Designing large-scale web sites. O'Reilly Media, Inc., Sebastopol.

Nawaz, A. (2012) A Comparison of Card-sorting Analysis Methods. *APCHI '12*. In *Proceedings of the 10th Asia Pacific Conference on Computer-Human Interaction*, pp. 583–592. Shimane, Japan.

Paea, S. and Baird, R. (2018) Information architecture (IA): using multidimensional scaling (MDS) and K-means clustering algorithm for analysis of card sorting data. *J. Usability Stud.*, 13, 138–157.

Paea, S., Havea, R. and Paea, M. K. (2020) Card sorting: practical guidance from a Pacific perspective. It Takes an Island and an Ocean. 76, 78–97.

Pampoukidou, S. and Katsanos, C. (2021) Test-Retest Reliability of the Open Card Sorting Method. In *Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems*, pp. 1–7. Yokohama japan, ACM, New York.

Paul, C. L. (2014) Analyzing card-sorting data using graph visualization. *J. Usability Stud.*, 9, 87–104.

Redmond, S. J. and Heneghan, C. (2007) A method for initialising the K-means clustering algorithm using kd-trees. *Pattern Recogn. Lett.*, 28, 965–973.

Righi, C., James, J., Beasley, M., Day, D. L., Fox, J. E., Gieber, J., Howe, C. and Ruby, L. (2013) Card sort analysis best practices. *J. Usability Stud.*, 8, 69–89.

Rosenfeld, L., Morville, P., Nielsen, J., & ProQuest. (2002). *Information Architecture for the World Wide Web*. O'Reilly, Cambridge. https://books.google.com.fj/books?id=hLdcLklZOFAC

Rousseeuw, P. J. (1987) Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *J. Comput. Appl. Math.*, 20, 53–65.

Shukla, S. and Naganna, S. (2014) A review on K-means data clustering approach. *Int. J. Inf. Commun. Technol.*, 4, 1847–1860.

Spencer, D. (2009) *Card sorting: designing usable categories*. Rosenfeld Media, Brooklyn, NY, USA. https://books.google.com.fj/books?id=-nk3DwAAQBAJ.

Syakur, M. A., Khotimah, B. K., Rochman, E. M. S. and Satoto, B. D. (2018) Integration k-means clustering method and elbow method for identification of the best customer profile cluster. *IOP Conf. Ser. Mat. Sci. Eng.*, 336, 012017.

Tibshirani, R., Walther, G. and Hastie, T. (2001) Estimating the number of clusters in a data set via the gap statistic. *J. R. Stat. Soc. Series B Stat. Methodology*, 63, 411–423.

Tullis, T. and Wood, L. (2004) How many users are enough for a card-sorting study. *Proceedings UPA,* June 7–11. Minneapolis, MN.

Tullis, T. and Wood, L. (2005) How can you do a card-sorting study with LOTS of cards. In *Poster Presented at the Annual Meeting of the Usability Professionals Association,* June 24 – July 1. Montreal, QB, Canada.

Žalik, K. R. (2008) An efficient k'-means clustering algorithm. *Pattern Recognit. Lett.*, 29, 1385–1391.

APPENDIX 1. Techniques to determine the number of categories k

Method	Formulas
Gap Statistic	$\text{Gap}_n(k) = E_n^*\big[\log W_k\big] - \log W_k$, **where** $n_r = \mid C_r \mid$, $W_k = r\sum_{r=1}^{k}\frac{1}{2n_r}D_r = \sum_{r=1}^{k}\frac{1}{2n_r}\sum_{i,i' \in C_r} d_{ii'}$ **is the total intra-category distance** d**, across all** r **categories** C_r**and** $E_n^*\{.\}$ **denote the expectation under a sample of size** n **from the reference distribution. To estimate the gap statistic and find the number of categories via** $\hat{k}_G =$ smallest k such that $Gap(k) \geq Gap(k+1) - S_{k+1}$, **where** S_k **is the standard error from the estimation of** $Gap(k)$**.**
Silhouette	$S(i) = \frac{b(i)-a(i)}{\max\{a(i),b(i)\}}$, **where** $a(i)$ **is the average dissimilarity of** i**th data point with all other data within the same category. The** $b(i)$ **is the minimum average dissimilarity of** i**th data point to any other category which** i **is not a member.**
Category View (3DCV)—Average method	$A = \frac{1}{n}\sum_{i=1}^{n} a_i$, where n is the total number of participants in the study and $a_i$is the number of categories created by a participant.

APPENDIX 2. The card names and their numbers

Card number	Card name	Card number	Card name
1	Schedule of activities	26	Labs
2	Guides/navigation instructions	27	Course worksheet
3	Read me first	28	Chat room
4	Course tour	29	Class News and Announcements
5	Modes of communication	30	Plagiarism
6	Netiquette	31	Youtube
7	Discussion forums	32	Frequently asked questions
8	Assignments	33	Contact us
9	Policies	34	Counseling
10	Software	35	Campus life
11	Lecture video	36	Lecture capture
12	Tutorials	37	Mobile applications
13	Course calendar	38	Course syllabus
14	Course outline	39	Lecture notes
15	Quiz	40	Teaching Staff Information
16	Questionnaire	41	If you need to complain
17	Surveys	42	Coursebooks (Textbooks)
18	Group projects	43	Past test papers
19	Presentations	44	Past assignments
20	Bibliography	45	Lab and Tutorial signups
21	Reflections	46	Tutorial and Lab attendance
22	Open Educational Resources (OER)	47	Required Readings
23	Practice quizzes	48	Mark sheet
24	Read an article	49	Satellite Tutorial video recordings
25	Case studies	50	Web conferencing/webinar

Appendix 3 The BMM algorithm.

The main steps of the algorithm employed by BMM.

Experiment scenario 1: 5 participants (V, W, X, Y, Z) and 5 cards (a, b, c, d, e)

1) Let $\mu_1, \mu_2, ..., \mu_k$ be the cards to be sorted. Begin with μ_k categories, each with a single card.
2) Produce combinations of two cards in a category (called the based pairs) for all cards. For instance, $[\mu_1, \mu_2]$, $[\mu_1, \mu_3]$, $\ldots$, $[\mu_1, \mu_{k+1}]$, $[\mu_2, \mu_3], \ldots, [\mu_2, \mu_{k+1}]$, $\ldots$, $[\mu_{k-1}, \mu_k]$. The order of the cards is not important, so $[\mu_1, \mu_3] = [\mu_3, \mu_1]$.
3) The based pair with the highest score (frequency of cards placed together by participants) is locked in as a new category.
4) All subsets of this new category are eliminated.
5) The process in step 3) repeats, and when a pair is locked in intersects with an existing locked category, the former is agglomerated with the latter. Repeat step 4).
6) The algorithm stops when all the cards are merged into a single category $[\mu_1, \mu_2, ..., \mu_k]$.

Experiment scenario 1 groupings from 5 participants in open card sorting

Participant	Group	Description
V	$[a], [b], [d], [c, e]$	• 3 groups with a card each • 1 group with 2 cards
W	$[a, b, c], [d, e]$	• 1 group with 3 cards • 1 group with 2 cards
X	$[a, b], [c, d, e]$	• 1 group with 2 cards • 1 group with 3 cards
Y	$[a, b], [c, d], [e]$	• 2 groups with 2 cards each • 1 group with 1 card
Z	$[a, b], [c], [d, e]$	• 2 groups with 2 cards each • 1 group with 1 card

Result of BMM = $4 \times [a, b], 1 \times [a, c], 1 \times [b, c], 2 \times [c, d], 2 \times [c, e], 3 \times [d, e]$.

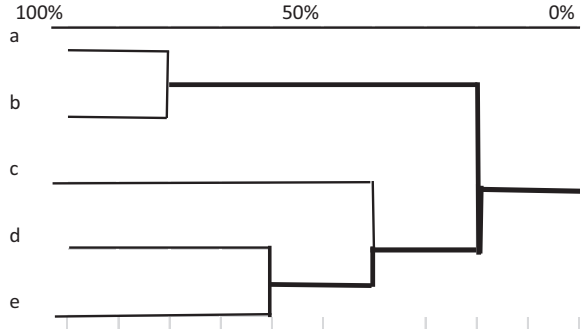
Experiment scenario 2: 5 participants (V, W, X, Y, Z) and 5 cards (a, b, c, d, e)

Experiment scenario 2 groupings from 5 participants in open card sorting

Participant	Group	Description
V	$[a, b,], [c, d, e]$	• 1 group with 2 cards • 1 group with 3 cards
W	$[a, b], [c], [d], [e]$	• 1 group with 2 cards • 3 groups with a card each
X	$[a, b,], [c, d, e]$	• 1 group with 2 cards • 1 group with 3 cards
Y	$[a, b, c], [d], [e]$	• 1 group with 3 cards • 2 groups with a card each
Z	$[a], [b], [c], [d, e]$	• 3 groups with a card each • 1 group with 2 cards

Result of BMM = $4 \times [a, b], 1 \times [a, c], 1 \times [b, c], 2 \times [c, d], 2 \times [c, e], 3 \times [d, e]$.

The following figure presents the result of BMM (experiment scenarios 1 and 2) in a dendrogram plot. The dendrogram presents the grouping or classification process as the formation of nested categories at successive grouping stages (Liu and Wickens, 1992).



The pair reduction process in experiment scenario 1 and experiment scenario 2 has produced identical results for two different experiment scenarios. The BMM works by merging the strongest pairs, so it does not reconstruct the original data.

Appendix 4 Example of how to calculate PAS

Refer to Group 2 of Table 5 and the yellow portion of Fig. 8 that shows the Group 2 (G2) card names. Note that the card names and their numbers are presented in Appendix 2.

$$\mathrm{PAS(G2)} = \frac{\sum_{I=1}^{I \subset A} C_{i,j}}{n \times 100\%} = \frac{C_{27,26} + \ldots + C_{31,26} + C_{28,27} + \ldots + C_{31,27} + C_{29,28} + \ldots + C_{31,28} + C_{30,29} + C_{31,29} + C_{31,30}}{6(100)}$$
$$= \frac{45 + \ldots + 12 + 51 + \ldots + 40 + 73 + \ldots + 38 + 49 + 34 + 47}{15(100)} = \frac{600}{1500} = 0.4$$

Therefore, the participant's agreement score of group 2 is 0.4. This score (0.4) indicates how strong the similarities combination of the cards in group 2. The closer the score to 1, the stronger the similarity between the combination of the cards in a category.

Appendix 5 Example of how to calculate CVS

Refer to Group 2 of Table 5 and the yellow portion of Fig. 8 that shows the Group 2 (G2) card names. To calculate the category

1) Let $\mu_1, \mu_2, ..., \mu_k$ be the cards to be sorted. Begin with μ_k categories, each with a single card.
2) Produce combinations of two cards in a category (called the based pairs) for all cards. For instance, $[\mu_1, \mu_2]$, $[\mu_1, \mu_3]$, $\ldots$, $[\mu_1, \mu_{k+1}]$, $[\mu_2, \mu_3], \ldots, [\mu_2, \mu_{k+1}]$, $\ldots$, $[\mu_{k-1}, \mu_k]$. The order of the cards is not important, so $[\mu_1, \mu_3] = [\mu_3, \mu_1]$.
3) The based pair with the highest score (frequency of cards placed together by participants) is locked in as a new category.
4) All subsets of this new category are eliminated.
5) The process in step 3) repeats, and when a pair is locked in intersects with an existing locked category, the former is agglomerated with the latter. Repeat step 4).
6) The algorithm stops when all the cards are merged into a single category $[\mu_1, \mu_2, ..., \mu_k]$.

validity of card "Chat room" please refer to Appendix 3. Let k=26 = "Chat room" (red row and column in Fig. 8), then

$$\sum_{i\neq 31}^{I\subset A} C_{26,j} = C_{26,27} + C_{26,28} + C_{26,29} + C_{26,30} + C_{26,31}$$
$$= 45 + 22 + 26 + 18 + 12 = 123, \text{and}$$
$$\sum_{i\neq 31}^{I\subset M} C_{26,j} = C_{26,1} + C_{26,2} + C_{26,3} + C_{26,4} + ...$$
$$+C_{26,50} = 719.$$

Thus,

$$\text{CVS}\,(26 \subset \text{G2}) = \frac{\sum_{i\neq 26}^{I\subset A} C_{26,i}}{6\sum_{i\neq 26}^{I\subset M} C_{26,i}} = \frac{123}{6(719)} = 0.0285, \text{where}\, n = 6.$$

Therefore, the category validity score of "Chat room" is 0.0285. In category validity score, the higher the score, the stronger the similarity between a card and the combination of the cards in a category. This technique is also used to measure the findability of an element. There is a correlation between the typicality of an element and its category validity.

APPENDIX 6. Categories created by BMK-means and MaxMin for our open card sort dataset 82% of Cards were placed by two methods in the same categories.

BMK-means	MaxMin
9 Cards	11 Cards
· Bibliography	· Bibliography
· Case Studies	· Chat room
· Coursebooks (Textbooks)	· Discussion forums
· Discussion Forums	· Mobile applications
· Mark sheet	· Open Educational Resources
· Questionnaire	· Questionnaire
· Read an article	· Read an article
· Reflections	· Reflections
· Requires Readings	· Software
6 Cards	· Web conferencing/webinar
· Chat room	· Youtube
· Mobile applications	7 Cards
· Open Educational Resources	· Assignments
· Software	· Group projects
· Web conferencing/webinar	· Labs
· Youtube	· Lecture notes
7 Cards	· Practice quizzes
· Course calendar	· Quiz
· Course outline	· Tutorials
· Course syllabus	8 Cards
· Course tour	· Case studies
· Course worksheet	· Course worksheet
· Read me first	· Coursebooks (Textbooks)
· Schedule of activities	· Lab and Tutorial signups
6 Cards	· Mark sheet
· Lecture capture	· Required Readings
· Lecture video	· Schedule of activities
· Past assignments	· Tutorial and Lab attendance
· Past test papers	6 Cards
· Presentations	· Lecture capture
· Satellite Tutorial video recordings	· Lecture video
9 Cards	· Past assignments
· Assignments	· Past test papers
· Group projects	· Presentations
· Lab and Tutorial signups	· Satellite Tutorial video recordings
· Labs	5 Cards
· Lecture notes	· Course calendar
· Practice quizzes	· Course outline
· Quiz	· Course syllabus
· Tutorial and Lab attendance	· Course tour
· Tutorials	· Read me first
13 Cards	13 Cards
· Campus life	· Campus life
· Class News and Announcements	· Class News and Announcements
· Contact us	· Contact us
· Counselling	· Counselling
· Frequently asked questions	· Frequently asked questions
· Guides/navigation instructions	· Guides/navigation instructions
· If you need to complain	· If you need to complain
· Modes of communication	· Modes of communication
· Netiquette	· Netiquette
· Plagiarism	· Plagiarism
· Policies	· Policies
· Surveys	· Surveys
· Teaching Staff Information	· Teaching Staff Information