MaxMin clustering for historical analogy

Yasunobu Sumikawa¹ · Ryohei Ikejiri² · Ryo Yoshikawa³

Received: 26 February 2020 / Accepted: 11 July 2020 / Published online: 28 July 2020
© Springer Nature Switzerland AG 2020

Abstract

Historical analogy is the ability to use historical knowledge to consider solutions for a present event, and it can be promoted by group learning. However, group creation for promoting the ability has been unexplored. This study proposes a novel clustering algorithm, named MaxMin clustering (MMC), to enhance discussions of group learning toward promoting historical analogy. The key concept is group formation by aggregating similar and different users. MMC uses aspects provided by users for the same present event. Subsequently, it solves maximum and minimum optimization problems to find similar and different users by counting the number of aspects shared by them. MMC is implemented and evaluated through comparison with other clustering algorithms; the comparison is based on the degree to which the generated clusters satisfy conditions for enhancing discussions of group learning toward promoting historical analogy. The experimental results prove that only MMC can generate suitable groups.

Keywords Clustering · Historical analogy · Collaborative learning · History education

1 Introduction

The benefits of studying history are manifold, e.g., enhanced understanding of the past and discovery of meaningful connections or analogies over time. In fact, history can provide both information regarding the past and several solution candidates for similar modern issues [15]. Hence, several ongoing educational studies and institutions propose the learning of history, followed by application of the acquired knowledge to the development of creative solutions for solving present issues [38]. Therefore, the ultimate goal is to develop an ability to apply such solutions to modern issues [43]. This ability is called *historical analogy*.

According to [23], two important aspects must be considered to effectively utilize analogy. First, incompleteness exists when generating a plausible inference from a source to a target. Second, the analogy is effective if explicit thinking ability pertaining to higher-order relations exist

between the source and target. Hence, discussions regarding a present event based on past events having common aspects with the former can promote historical analogy, with the awareness that no perfectly similar events occur over different time periods. However, when historical analogy is applied to temporal events, each person's sense of similarity between past and present events is subjective [23]. In addition, historical analogy may yield misused analogies. Therefore, learners must be cautious when adopting such analogies in their discussions [18]. Previously, Ikejiri et al. discovered that the validity of historical analogies can be effectively verified through group discussion-based history learning [27]. Subsequently, they performed experimental evaluations to study how group discussions promote historical analogy for high-school students [30].

Contributions In this paper, we consider the following research question:

✉ Yasunobu Sumikawa, ysumikawa@acm.org; Ryohei Ikejiri, ikejiri@iii.u-tokyo.ac.jp; Ryo Yoshikawa, yoshikawa.ryo@nagoya-bunri.ac.jp | ¹University Education Center, Tokyo Metropolitan University, Hachioji, Tokyo, Japan. ²Interfaculty Initiative in Information Studies, The University of Tokyo, Bunkyo City, Tokyo, Japan. ³Department of Information and Media Studies, Nagoya Bunri University, Inazawa, Aichi, Japan.

- How can groups of users having similar and different aspects of the same present events be formed?

To answer this question, we propose a novel clustering algorithm, named MaxMin clustering (MMC), to promote the identification of historical analogies by group discussions. MMC forms groups in two steps. First, it finds users who have the same aspects of an event; these users are classed as a subgroup. Second, it aggregates subgroups into a group, which contains users having different aspects of the same event. The objective of this process is to enhance discussions, i.e., users in the same subgroups are able to confirm whether their ideas and claims are correct, whereas users in different subgroups can exchange ideas within the same group. It is note worthy that the second step, where different users are aggregated into a single group, is the key concept of MMC. Notably, the objectives of existing clustering algorithms are essential for forming groups composed of similar data only.

To demonstrate the use of MMC in a real-world scenario, we investigate a situation where students of a high-school history class are required to predict future implications of the information technology (IT) revolution. It is assumed that the students are knowledgeable regarding the Industrial Revolution and its historical context. Hence, some students may focus on the positive effects of the IT revolution (present event) by recognizing that the Industrial Revolution (past event) enhanced the economic growth of Great Britain through increased usage of steam power and development of machine tools and factories. By contrast, other students may be concerned regarding work–life balance because working hours increased during the Industrial Revolution. If the two effects are assigned “economy” and “literature and thought,” respectively, MMC can distinguish the different subgroups based on their selections. In this study, we regard the categories selected by students as the source and target presented in [23]. Therefore, to cluster both similarities and differences to enhance the resultant analogy, MMC combines students into subgroups and groups according to the number of categories they have in common and the differences between them, respectively. This is achieved by solving maximum and minimum optimization problems.

This paper extends our FICC2019 paper [31]. First, we compare MMC with a broader range of related studies, to provide clear understanding of the relationships between this study and other studies pertaining to machine learning, data mining, Histoinformatics, and history education. Subsequently, we extend the clustering algorithm proposed in [31], which assumes only 2 persons in each subgroup and only 2 subgroups in each group. By contrast, the algorithm presented in this paper generalizes the numbers of persons/subgroups into subgroups/groups.

This generalization allows more flexible application scenarios than the FICC2019 algorithm. Finally, more extensive evaluations (five cases) were performed in this study compared with [31], in which the algorithm was evaluated in one case only.

The remainder of this paper is organized as follows: Sect. 2 provides the definitions used herein, whereas Sect. 3 summarizes several related studies. Section 4 details our data collection methods. Section 5 describes the group creation method. Section 6 provides the experimental results, and Sect. 7 presents our conclusions.

2 Problem definitions

Assumption This study assumes that each user first selects a past event they perceive as analogous to the present event according to the similarity between the present and past events. In addition, it is assumed that all past events have suitable event categories before MMC is applied. In the experimental evaluation reported herein, we regarded the categories as aspects of the events for users to create feature vectors.

Input and output Let C be a set of event categories and C' be a power set of C . For user-selected event categories $C'' = \{c'_1, c'_2, \dots, c'_m\}$ (where m represents the number of users and c'_i is the element of C' corresponding to the i th user), MMC outputs clusters of users C_u . This study uses event categories (C'') to create feature vectors that are used to create groups.

3 Related works

3.1 Analogy-based information retrieval

Temporal information retrieval (T-IR) is becoming one of the most important topics in IR research owing to the increasing sizes of digital archives containing items such as historical images and documents. Related studies mainly propose algorithms to obtain desirable data incorporating temporal expressions, e.g., to detect temporal expressions or information [24], retrieve history-related images [13], organize information by creating timelines [2, 16, 25], or perform future-related IR [3, 32, 50]. A detailed survey of T-IR is provided in [8].

Search methods for analogous items is also a T-IR research topic. Previously, Zhang et al. proposed an algorithm for detecting counterparts of entities over time, which functions via matrix transformations bridging two different vector spaces [65]. This algorithm first constructs vector spaces for different time-ranges, e.g., [1800–1850]

and [1950–2000]. It then maps an entity from one vector space onto another one by considering the top-k similar words on the two spaces. Zhang et al. subsequently extended this algorithm to consider hierarchical cluster structures [66].

It is note worthy that the IR algorithms mentioned above do not output groups; therefore, their objectives are differ from that of the present study.

3.2 History education

History education researchers have studied effective and efficient methods for enhancing historical analogy. Drie and Boxtel have discovered the components of historical reasoning [56]. Mansilla studied how students successfully applied their knowledge of history to current problems [5]. Lee proposed the definition of usable history to connect past and present events [38]. Ikejiri et al. designed learning tools for identifying causal relationships within modern societal problems using references to historical causal relations [26] and for creating new policies that can stimulate the Japanese economy [27].

Through experimental evaluations of high-school students having different aspects of the same events, Ikejiri et al. discovered that group discussions are beneficial for promoting historical analogy [30]. However, no algorithm that automatically creates groups containing both similar and different data has been reported to date.

3.3 Clustering

In natural language processing and machine learning studies, clustering algorithms are widely used; therefore, several types of clustering algorithms have been developed. The key purpose of a clustering algorithm is to identify similarities between data and to cluster them into groups [1, 19]. As several surveys presenting a broad overview of clustering have been published, e.g., [17, 59, 60], this study compares previously proposed partitioning-, hierarchy-, distribution- and graph-based algorithms with MMC.

First, we review partitioning-based algorithms. These types of algorithms segment data into groups based on two assumptions. The first is that all groups must contain at least one data element, whereas the second is that each data element must belong to one group. The k-means algorithm [40] is one of the most popular algorithm in this category. It first randomly assigns a cluster number to each data element and then calculates the cluster centers by averaging all the coordinates of the data within the same clusters. After this calculation, it reassigns cluster numbers to each data element. These processes are iteratively performed until certain criteria are satisfied. Other proposed

partitioning-based algorithms include CLARA [34], PAM [35], and CLARANS [44].

Next, we explain hierarchy-based algorithms. This type of algorithm defines hierarchical relationships between data, with the relationships typically represented by dendrograms. Two different approaches can be used to define the required dendrograms. The first is a bottom-up approach that creates clusters by merging data recursively. The second is a top-down approach that splits a node recursively into sub-nodes. Both approaches are terminated if certain criteria are satisfied. Representative algorithms in this category include Birch [64], CURE [21], and ROCK [22].

If the data are considered to be generated from a probability distribution, statistical methods can be applied through a distribution-based algorithm. One of the most famous algorithms in this category is the Gaussian mixture model (GMM) [51]. This algorithm assumes that all data are generated from several Gaussian distributions. As another example, DBCLASD [61] operates under the assumption that data in a cluster are uniformly distributed.

Finally, we summarize graph-based algorithms. This type of algorithm uses the data to define graphs whose nodes and edges represent the data and the similarity scores between them. Spectral clustering [54] is one of the most famous graph-based algorithms that functions by creating clusters on a graph.

The algorithms above create groups by aggregating similar data. However, the objective of the present study is to combine not only similar users, but also different users as a group. Among the various algorithms considered herein, the experimental results (Sect. 6) reveal that MMC alone satisfies the objective. In other words, we used the algorithms above as baselines in the evaluation and then confirmed that none of the baselines can satisfy the purpose of this study.

Mautz et al. proposed an algorithm that discovers multiple mutually orthogonal subspaces by finding both shape and color spaces for objects and the corresponding clusters [41]. Similar to previous studies regarding clustering study, this algorithm finds only similar objects, whereas MMC creates groups from dissimilar subgroups.

In addition, studies comparing algorithms or generated clusters have been performed [37]. Cazals et al. proposed a framework to analyze the stability of clustering algorithms and compare clusters by introducing meta-clusters [10]. They defined the family-matching problems on an intersection graph. As the objectives of above mentioned studies are to compare between algorithms or clusters, they are therefore orthogonal to this study.

3.4 Classification

3.4.1 Single- and multi-label classification

Single-label classification is one of the most important topics in classification research, as many algorithms proposed as multi-labeled classification (MLC) and semi-supervised learning (SSL) are based on single-label classification. A special case of single-label classification is a binary classification that assigns one of two categories to each data element. Random forest and support vector machine are popular algorithms that perform binary classification. If two or more categories exist, and we apply classifiers to evaluate categories as one vs. rest, the classification problem can be extended from binary to multi-class classification, where labels from various categories are assigned to data. Such research has been fundamental in the development of natural language processing, IR, machine learning, and other research fields; therefore, several researchers have published related literature surveys [4, 6, 49, 53].

MLC is the extension of single-label classification and allows one or more categories to be assigned to data elements. MLC algorithms can be categorized into two types of approaches: transformation and algorithm adaptation [55]. The former approach transforms data into a form suitable for the application of the traditional single-label classifier. In this approach, several classifiers are independently trained for each label. Subsequently, they are used to predict labels by combining [12] or chaining [52] them. Another transformation approach, i.e., label powerset transformation, is also popular in MLC. In this method, the label representation is transformed to consider all label combinations for the application of multi-class classifiers. The algorithm adaptation approach modifies an existing single-label classifier to treat multi-label data. MLkNN is one of the most famous algorithms employing this approach [63]. Both the transformation and algorithm adaption approaches are used as ensemble-style approaches, as in random k-labelsets [39] and classifier chain ensembles [52], which combine results from several classifiers based on either problem transformation or algorithm adaptation. Typically, the results are combined via a voting scheme, where every category is predicted using the probability of votes from individual classifiers [42]. Regarding single-label classification, several researchers have presented overviews of MLC studies in literature surveys [6, 49, 62].

3.4.2 Semi-supervised learning style classification

High-quality labeled datasets must be prepared to train both single- and multi-label classifiers. However, the preparation is expensive. In many real applications, the

available labeled datasets are small and assigning suitable labels to unlabeled data is time-consuming. If numerous unlabeled data are obtained, then SSL style classification is useful for reducing the cost of preparing labeled data. This is because the SSL-style approach incrementally adds labeled data from unlabeled data by applying classifiers trained on the labeled data. The classifiers are then retrained on the new labeled data, which include the results already provided by the classifiers [9].

One of the most popular implementations of SSL classification is the use of single- and multi-label classifiers with the expectation-maximization algorithm for classifier training [14, 20, 45]. The details of this approach are available in [48, 69].

As an alternative type of SSL-based classification, LP has been proposed [68]. The objective of this algorithm is to spread labels from a small-sized labeled dataset to a large-sized unlabeled dataset. This procedure is performed on a graph whose nodes and edges represent labeled and unlabeled data and their similarities, respectively. This algorithm is based on two fundamental assumptions: 1) the values of the initial labeled dataset are not affected by the spreading from the unlabeled data, and 2) similar data are assigned to the same label.

The original algorithm is designed for single-label classification, especially for multi-class classification; however, this algorithm has been extended to MLC and has overcome several issues in MLC. For example, methods have been developed for label correlation recognition [33, 58], transductive algorithm establishment [36], and implementation of smoothing effects for incorrect labels [11, 57, 67]. Additional details are available in [70].

4 Data collection

4.1 Event categories

Thirteen event categories were used in this study: Reign (Rg), Diplomacy (Dp), War (Wr), Production (Pr), Commerce (Cr), Study (St), Religion (Rl), Literature and Thought (LT), Technology (Tc), Popular Movement (PM), Community (Cn), Disparity (Ds) and Environment (En). These event categories are described in [28, 29] as an event category list to define a history education curriculum that associate past and present events. These categories are based on definitions obtained from the Encyclopedia of Historiography [47]. Sample events corresponding to the 13 categories are listed in Table 1.

Table 1 Example events. This table uses abbreviated category names: Reign (Rg), Diplomacy (Dp), War (Wr), Production (Pr), Commerce (Cr), Study (St), Religion (Rl), Literature and Thought (LT), Technology (Tc), Popular Movement (PM), Community (Cn), Disparity (Ds), and Environment (En)

Event	Categories
Bloc economy	Dp, Cr, Cn
First opium War	Dp, Wr, Cr, Tc
Industrial revolution	Pr, Cr, St, Tc, Cn, En
Literary encouragement of Louis XIV	Rg, LT, Cn
Commercial rules in guild	Pr, Cr, Tc, Cn

Fig. 1 Obtaining event categories of past events

4.2 Past event

MMC uses event categories assigned to past events. It is assumed that past events and their categories are defined before an algorithm is applied.¹ Hence, each user only needs to select a past event before discussing about it.

Figure 1 illustrates the manner in which each user selects a past event. First, the user reads about a present event described in the newspapers or any other articles types, such as those on Wikipedia. Subsequently, he/she selects multiple categories applicable to the present event,

and then searches past events according to the input event categories for the present event. It is note worthy that all events can be assigned to more than one category. For example, if the user reads a Wikipedia article² regarding the 2014 West Africa Ebola outbreak to determine the outbreak's widespread effects, the following event categories can be assigned: *En*, as there were many deaths, both human and nonhuman; *Tc*, as a vaccine was developed; *St*, as research was performed regarding the details and relevant statistics. After searching for past events, the user selects the past event that they consider as having the most similar effects to those of the present event. MMC regards the event categories of the selected past events as C'' , which is defined in Sect. 2.

5 Methodology

5.1 Overview

As the objective of this study is to stimulate group discussions by students, the algorithm was designed to satisfy the following requirements.

1. Each group should have at least two users with the same aspects of the same event.
2. Each group should have users with different aspects of the same event.

As discussed in Sect. 1, grouping users with different aspects is effective for stimulating discussions; hence, the second requirement is the key concept of MMC. The first requirement aids the discussion if one user neglects to mention some important ideas (as the other user with the same aspects can then mention them).

Figure 2 presents an overview of our algorithm, which first uses information on the perceived aspects selected by each user for a specified event. Subsequently, to satisfy the first requirement, the algorithm creates subgroups by aggregating users having similar aspects of the same event according, as indicated by their selected event categories for the selected past events. Finally, MMC combines these subgroups to satisfy the second requirement.

In the remainder of this section, we describe each step in MMC.

¹ This paper uses the dataset available on <https://zenodo.org/record/3601707>.

² https://en.wikipedia.org/wiki/West_African_Ebola_virus_epidemic.

Fig. 2 Overview of MMC

5.2 Feature vector creation

Initially, MMC uses event categories and converts them into feature vectors, the elements of which are represented by 0 or 1. This feature vector creation for the i th user can be formally defined as follows:

$$f_{i_k} = \delta(c_k, c'_i), c_k \in C, 1 \leq k \leq |C| \quad (1)$$

where the δ function returns 1 if the first argument is included in the second argument; otherwise, it returns 0. C and c'_i represent the set of all event categories and of event categories for past events that are selected by the i th user, respectively. Eq. 1 defines the k th element of the feature vector.

5.3 Combining similar data

After creating feature vectors from the event categories, MMC creates user subgroups according to their similarities. MMC measures the similarity between feature vectors by counting the number of common categories; a higher number corresponds to a greater similarity.

The formal definition of the subgroup similarity measurement is as follows:

$$DataSim(\mathbf{f}) = \sum_{k=1}^{|C|} And(f_{1_k}, f_{2_k}, \dots, f_{m_k}), 1 \leq |\mathbf{f}| \leq m \quad (2)$$

where the function *And* applies the AND logical operator to the arguments, and f_i represents the feature vector for the i th user. As each element of the feature vector is binary, as described in the previous section, Eq. 2 represents the number of common categories selected by the users.

MMC solves the following maximum problem.

$$\begin{aligned} \max \{ & DataSim(fv(SG_i)) \mid SG_i \in \mathbf{SG} \} \\ \text{s.t. } & \bigcap SG_i = \emptyset \\ & |SG_i| \geq 2 \end{aligned}$$

where $fv(\cdot)$ is a function that outputs feature vectors for the given argument using Eq. 1, $\mathbf{SG}$ is a set of subgroups, and $SG_i \in \mathbf{SG}$ is a set of users.

This problem can be regarded as the *Knapsack problem* because each subgroup contains feature vectors as well as a knapsack. In our problem, it is assumed that there are several knapsacks. MMC determines feature vector combinations that should be included in each knapsack to yield the maximum total variance of the subgroups. As the Knapsack problem is NP-complete, a polynomial algorithm that provides a solution does not exist. MMC solves this problem by traversing a tree that represents subgroup candidates to determine the best subgroups.

We present the pseudo-code for solving the maximum problem in Algorithm 1.

Algorithm 1 subgroup creation

Input: A set of feature vectors F
Output: A set of subgroup SG

```

1: Function CombineSimData( $F$ )
2:    $SG \leftarrow \emptyset$ 
3:    $WorkList \leftarrow F$ 
4:   while  $|WorkList| > 0$ 
5:      $s \leftarrow FindBestSimSubGroup(WorkList)$ 
6:      $SG.add(s)$ 
7:     for  $s_k \in s$ 
8:        $WorkList.remove(s_k)$  // Removing all members of this subgroup
9:   return  $SG$ 

```

Algorithm 2 Determining the best subgroup

Input: A set of feature vectors $WorkList$
Output: A subgroup of users $BestSubGroup$

```

1: Function FindBestSimSubGroup( $WorkList$ )
2:    $val \leftarrow 0$ 
3:    $BestSubGroup \leftarrow None$ 
4:   for  $sg \in PowerSet(WorkList)$ 
5:     if  $|sg| \geq 2$ 
6:        $sval \leftarrow DataSim(sg)$ 
7:       if  $val < sval$ 
8:          $val \leftarrow sval$ 
9:        $BestSubGroup \leftarrow sg$ 
10:  return  $BestSubGroup$ 

```

Algorithm 2 defines the function *FindBestSimSubGroup*, which returns a subgroup. This function uses *PowerSet*, which returns a power set of the arguments, corresponding to traversing a tree. After verifying the scores of all combinations, the function returns the subgroup with the highest similarity score.

5.4 Group creation

Following subgroup creation, MMC creates groups by combining subgroups that are dissimilar to each other. To combine dissimilar subgroups, MMC counts the common event categories of subgroup combinations. To perform this process, MMC defines a feature vector for subgroup as follows:

$$SGf_i = \sum_{j=1}^{|SG_i|} f_{j_k}, 1 \leq k \leq |C| \quad (3)$$

This indicates that a feature vector for a subgroup is determined by voting of the member.

Subsequently, MMC measures the similarity between different subgroups based on the following equation:

$$DiffData(SGf) = \sum_{k=1}^{|C|} \prod_{SGf_i \in SGf} SGf_{i_k} \quad (4)$$

where SGf represents a set of feature vectors for subgroups. This measurement is used for solving the following minimum problem.

$$\begin{aligned}
 & \min \{DiffData(SGfv(G_i)) \mid G_i \in \mathbf{G}\} \\
 & \text{s.t. } \bigcap G_i = \emptyset \\
 & \quad |G_i| \geq 2
 \end{aligned}$$

where $SGfv(\cdot)$ is a function that outputs feature vectors for the given argument using Eq. 3, $\mathbf{G}$ and $G_i \in \mathbf{G}$ represent sets of groups and subgroups, respectively.

The method to solve the minimum problem is shown in Algorithm 3. As the minimum problem is also the Knapsack problem, this algorithm performs a tree traverse. In this algorithm, a tree represents group candidates; therefore, traversing this tree is analogous MMC comparing the scores of all candidates.

Algorithm 3 Group creation.**Input:** A set of feature vectors for subgroup FG **Output:** A set of group G

```

1: Function CreateGroups( $FG$ )
2:    $G \leftarrow \emptyset$ 
3:    $WorkList \leftarrow FG$ 
4:   while  $|WorkList| > 0$ 
5:      $SG \leftarrow FindWorstSimGroup(WorkList)$ 
6:      $G.add(SG)$ 
7:     for  $SG_k \in SG$ 
8:        $WorkList.remove(SG_k)$ 
9:   return  $G$ 

```

Algorithm 4 defines the function *FindWorstSimGroup*, which returns a group. This function uses *PowerSet*, which corresponds to traversing a tree. After verifying the scores of all combinations, the function returns the group with the lowest similarity score.

is a partitioning algorithm, as a baseline. In addition, we employed the GMM as another baseline because it performs partitioning by modeling data according to Gaussian distributions. As MMC creates groups after creating subgroups, the output data can be presented as a graph.

Algorithm 4 Determining the best group.**Input:** A set of feature vectors $WorkList$ **Output:** A group of subgroups $BestGroup$

```

1: Function FindWorstSimGroup( $WorkList$ )
2:    $val \leftarrow 0$ 
3:    $BestGroup \leftarrow None$ 
4:   for  $g \in PowerSet(WorkList)$ 
5:     if  $|g| \geq 2$ 
6:        $sval \leftarrow DiffData(g)$ 
7:       if  $val > sval$ 
8:          $val \leftarrow sval$ 
9:          $BestGroup \leftarrow g$ 
10:  return  $BestGroup$ 

```

6 Experimental evaluation

6.1 Setup

6.1.1 Data collection

In the experimental evaluation, we used three types of datasets (Cases 1 & 2, Cases 3 & 4, and Case 5). The first type of dataset was produced by randomly assigning categories to each learner, for 16 assumed learners. For the second type of dataset, we produced 40 learners in a dataset because there are typically 40 students in a class in Japanese public schools. The last type of dataset includes 9 types of datasets for analyzing the scalability of MMC.

6.1.2 Baselines

We compared MMC with four existing algorithms: k-means [40], Birch [64], GMM [51] and Spectral [54]. As MMC can be regarded as a two-step (aggregating similar- and dissimilar-data) partitioning algorithm, we used k-means, which

Hence, Birch and Spectral were used as baselines as they create graphs for creating groups.

6.1.3 Parameters

According to research regarding argumentation-based computer-supported collaborative learning, collaborative learning is effective if each group contains two learners [46]. Therefore, in this study, we created subgroups and groups by combining two users and two subgroups, respectively. Hence, we combined four users in one group.

6.1.4 Measurements

Six measures were used in this study: group size, MinDist, inner-group similarity, user similarity, subgroup similarity, and quality of groups to evaluate the clustering algorithms. Group size corresponds to the number of data in each group. MinDist indicates the minimum Euclidean distance between all data element pairs in the same cluster.

Table 2 Scores of sizes, minimum distances in a group, and total distances between data for each group of Cases 1 & 2. Each group is labeled by a unique name from C1 to C4

		Case 1						Case 2					
		C1	C2	C3	C4	Ave.	SD	C1	C2	C3	C4	Ave.	SD
kmeans	Size	4	5	5	2	4	1.4142	3	6	3	4	4	1.4142
	MinDist	1.73	1.73	2.0	1.0	1.61	0.42	2.0	1.41	2.0	1.41	1.70	0.33
	Inner-group	2.17	2.25	2.22	1.0	1.91	0.61	2.29	2.24	2.15	2.0	2.17	0.12
Birch	Size	6	3	4	3	4	1.4142	5	4	4	3	4	0.8164
	MinDist	1.0	1.73	2.0	1.73	1.61	0.42	1.41	1.41	1.73	2.0	1.64	0.28
	Inner-group	2.36	1.98	2.14	1.98	2.12	0.17	2.20	2.0	2.29	2.15	2.16	0.12
GMM	Size	6	3	4	3	4	1.4142	5	5	2	4	4	1.4142
	MinDist	2.0	1.0	1.73	1.73	1.61	0.42	1.41	1.73	2.0	1.41	1.64	0.28
	Inner-group	2.29	1.89	2.10	2.26	2.14	0.18	2.28	2.29	2.0	2.0	2.14	0.16
Spectral	Size	3	2	4	7	4	2.1602	6	6	2	2	4	2.3094
	MinDist	1.73	1.0	1.73	2.0	1.61	0.42	1.73	2.0	1.41	1.41	1.64	0.28
	Inner-group	1.98	1.0	2.07	2.37	1.85	0.59	2.25	2.47	1.41	1.41	1.89	0.55
MMC	Size	4	4	4	4	4	0	4	4	4	4	4	0
	MinDist	1.73	1.73	1.41	2.0	1.71	0.24	1.41	2.0	1.73	2.0	1.78	0.27
	Inner-group	2.65	2.50	2.50	2.50	2.54	0.07	2.52	2.46	2.32	2.33	2.41	0.09

Table 3 Scores of sizes, minimum distances in a group, and total distances between data for each group of Case 3. Each group is labeled by a unique name from C1 to C10. “–” indicates that the corresponding term cannot be measured

		C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	Ave.	SD
kmeans	Size	2	9	3	7	4	6	2	2	3	2	4	2.49
	MinDist	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2.23	0.0	0.0	0.22	0.70
	Inner-group	0.0	0.55	1.33	1.21	0.94	0.53	0.0	2.23	0.0	0.0	0.68	0.75
Birch	Size	7	10	3	5	2	1	4	1	1	6	4	3.01
	MinDist	0.0	0.0	0.0	0.0	0.0	–	0.0	–	–	0.0	0.0	0.0
	Inner-group	1.22	0.81	1.15	0.56	0.0	–	0.94	–	–	0.53	0.74	0.42
GMM	Size	7	4	9	3	5	3	4	2	1	2	4	2.44
	MinDist	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	–	0.0	0.0	0.0
	Inner-group	1.21	0.94	0.55	1.15	1.35	0.0	0.0	0.0	–	0.0	0.58	0.59
Spectral	Size	6	6	2	7	5	3	2	4	3	2	4	1.88
	MinDist	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0	0
	Inner-group	0.53	1.58	0.0	0.83	0.69	1.49	0.0	0.0	0.0	0.0	0.51	0.62
MMC	Size	4	4	4	4	4	4	4	4	4	4	4	0
	MinDist	1.414	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.14	0.44
	Inner-group	2.21	1.57	1.98	1.78	1.65	1.72	2.23	2.19	1.33	1.48	1.82	0.32

Table 4 Scores of sizes, minimum distances in a group, and total distances between data for each group of Case 4. Each group is labeled by a unique name from C1 to C10. “–” indicates that the corresponding term cannot be measured

		C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	Ave.	SD
kmeans	Size	7	9	5	4	3	4	2	2	1	3	4	2.45
	MinDist	0.0	0.0	0.0	0.0	0.0	0.0	1.41	1.41	–	0.0	0.31	0.62
	Inner-cluster	1.45	0.38	0.0	0.66	0.0	0.86	1.41	1.41	–	0.0	0.69	0.63
Birch	Size	4	11	2	5	4	5	3	1	4	1	4	2.87
	MinDist	0.0	0.0	1.41	0.0	0.0	0.0	0.0	–	0.0	–	0.18	0.50
	Inner-cluster	0.86	0.73	1.41	1.02	0.70	0.0	0.0	–	0.90	–	0.70	0.49
GMM	Size	3	2	3	5	3	4	9	5	3	3	4	2.00
	MinDist	1.73	1.41	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.31	0.67
	Inner-cluster	1.82	1.41	0.66	0.0	0.0	0.70	0.57	1.02	0.0	1.15	0.73	0.63
Spectral	Size	2	7	2	3	5	6	3	4	3	5	4	1.70
	MinDist	1.41	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.14	0.45
	Inner-cluster	1.41	0.0	0.0	0.0	0.0	1.29	0.0	1.42	0.0	1.56	0.57	0.74
MMC	Size	4	4	4	4	4	4	4	4	4	4	4	0
	MinDist	1.41	1.41	1.41	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.42	0.68
	Inner-cluster	2.17	2.31	2.09	2.11	1.49	1.96	1.89	1.55	1.65	1.70	1.79	0.51

Table 5 Qualities of all clusters

	k-means	Birch	GMM	Spectral	MMC
Case 1	2.85	2.54	2.56	2.85	0.92
Case 2	2.81	2.89	2.67	2.46	1.23
Case 3	13.22	12.28	12.93	11.83	1.19
Case 4	16.81	17.06	15.86	15.17	1.02

Table 6 Average MinDists of all clusters

	k-means	Birch	GMM	Spectral	MMC
Case 1	2.26	2.20	2.28	2.44	2.54
Case 2	2.20	2.18	2.19	2.26	2.41
Case 3	0.74	0.84	0.78	0.79	1.82
Case 4	0.65	0.68	0.60	0.61	1.89

Table 7 Results of Inter-group for all clusters

	k-means	Birch	GMM	Spectral	MMC
Case 1	2.62	2.63	2.58	2.61	2.53
Case 2	2.64	2.66	2.63	2.63	2.48
Case 3	2.14	2.16	2.14	2.11	2.00
Case 4	2.14	2.16	2.13	2.11	1.98

Inner-group similarity is the average of all Euclidean distances between pairs in the same cluster. User similarity indicates the average similarity between two users who are allocated as the same subgroup. The similarity was measured using Eq. 2. Subgroup similarity is the average score calculated from Eq. 4 for each group. Finally, group quality measures the cluster quality. If all data in a group are close to each other, and the data between different groups are far from each other, then it can be concluded that the group quality is high. This measurement was proposed by Calinski and Harabaz (CH) [7] and is widely used in clustering studies.

6.2 Discussion of cluster shape analysis

Q. How much data do each group form using each algorithm containing data?

Q. Which clustering algorithms most effectively accomplish group creation for the research question posed in this study?

A. All baselines often aggregated less than three or more than five data into one group. However, MMC placed four data in all groups.

A. MMC outputs appropriate groups in the context of the research question. All baselines fail to create satisfying clusters that are required in this experimental evaluation, whereas MMC can create them.

Table 8 User and subgroup similarities of MMC for Case 3. Dist(sg1) and Dist(sg2) indicates the user similarities. Dist(sg1, sg2) indicates the subgroup similarity

	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10
Dist(sg1)	1.41	0.0	1.73	0.0	0.0	1.41	0.0	2.23	0.0	0.0
Dist(sg2)	1.73	1.0	0.0	2.23	2.0	0.0	1.41	0.0	0.0	2.0
Dist(sg1, sg2)	4.58	4.12	4.79	3.60	3.46	4.24	5.83	5.0	4.0	2.82

Table 9 User and subgroup similarities of MMC for Case 4. Dist(sg1) and Dist(sg2) indicates the user similarities. Dist(sg1, sg2) indicates the subgroup similarity

	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10
Dist(sg1)	1.41	1.41	1.41	0.0	0.0	2.0	0.0	1.41	2.0	2.23
Dist(sg2)	2.23	2.0	1.41	1.41	0.0	0.0	2.44	0.0	0.0	2.0
Dist(sg1, sg2)	3.87	4.69	4.47	5.47	4.47	4.47	3.74	3.74	3.46	1.73

Fig. 3 CH scores **Fig. 4** MinDist scores

Tables 2, 3, and 4 list the results of three measures: group size, MinDist, and inner-group similarity. Regarding group size, all baselines failed to create groups having four data, whereas MMC included four data in all clusters. As the aim of this experimental evaluation was to combine four users, MMC was the best algorithm in this study.

Q. How similar were the data in each subgroup created by MMC on average?

Q. How different were the subgroups in each group created by MMC on average?

A. MMC aggregated similar data in all subgroups as effectively as the baselines, because the MinDist scores were similar for all algorithms.

A. MMC can effectively combine dissimilar subgroups in all groups, with generally higher inner-cluster scores than those achieved by the baselines.

Next, we analyzed the qualities of all subgroups and groups. With regard to the first and second cases shown in Table 2, it is apparent that the MinDist scores of MMC were almost identical to those of the baselines because the average MinDist scores of all algorithms were 1.6 or 1.7. Subsequently, we compared the inner-group similarity scores of all algorithms. As MMC combines dissimilar subgroups, it is natural that its MinDist score was the highest among the five algorithms. These tendencies also appear in Cases 3 & 4, as shown in Tables 3 and 4.

Q. What were the qualities of the groups output by each algorithm?

A. The groups created by MMC had the lowest CH scores among those of the five algorithms.

Table 5 lists the whole-cluster qualities of various algorithms. It is apparent that MMC achieved the lowest CH score for all cases. This is natural because MMC combines dissimilar subgroups by solving the minimum problem. In fact, Table 6 shows that the average minimum inner-cluster distances of MMC are the largest for all cases.

Meanwhile, Table 7 shows that the average total distances between data in different groups are the smallest.

-
- Q.** How close were the two data in each subgroup created by MMC?
- Q.** How far apart were the two subgroups in each subgroup created by MMC?
- A.** All subgroups tended to contain two similar data, whereas all groups tended to contain two distantly spaced subgroups.
-

Next, we measured the MMC's group creation performance by analyzing the similarities between the two users in each subgroup and the differences between the two subgroups in each group; the results are reported in Tables 8 and 9, respectively. It is apparent that all subgroups tended to contain two similar data. By contrast, almost all distances between the subgroups were relatively larger than those between the data of the subgroups.

-
- Q.** How much will the quality of the clusters of the algorithms change as the number of students increases?
- Q.** How did the MinDist value of each method change as the number of data increases?
- A.** All the baselines improved the quality of the clusters and MinDist scores as the number of data increased.
- A.** MMC sustained the quality of the clusters and MinDist score because it combines dissimilar subgroups as a group.
-

Finally, we evaluated the five algorithms with CH and MinDist scores when they were applied on the third dataset type (Case 5), which includes 9 datasets: 40, 100, 200, 300, 400, 500, 600, 700, and 800 artificial student data. These data were produced by randomly assigning categories to each data element. Figure 3 shows the CH scores for all algorithms when they were applied on the 9 datasets. Based on the four baselines, as the number of data increased, the CH scores of the baselines increased as well. However, the CH score of MMC did not change. This is because MMC combines dissimilar subgroups by solving the minimum problem. Solving the minimum problem enable all groups to include two dissimilar subgroups; therefore, the CH score was unaffected. Meanwhile, the four baselines created groups by aggregating only similar data, thereby resulting in a higher CH score.

To understand the results better, Fig. 4 shows the MinDist scores that were produced when the five algorithms produced the CH scores shown in Fig. 3. As shown, MMC sustained the MinDist score, whereas the four baselines improved the MinDist scores. In MMC, as all subgroups were created by two users, MinDist sustained the score. Meanwhile, the baselines were not restricted; hence, closer data were obtained and the CH score improved.

7 Conclusions

The benefits of learning history are manifold. In this paper, we introduced a novel clustering algorithm, named MMC, for creating a new collaborative learning platform specialized for history. MMC solves two optimization problems to combine users having similar aspects of a particular event into one subgroup and to combine subgroups to form one group with user pairs having different aspects of the same event. We evaluated MMC with 4 baselines on three types of datasets to demonstrate that only MMC can output appropriate clusters for the research question proposed in this study, whereas all baselines failed to create the clusters. In addition, we confirmed that MMC aggregated similar and dissimilar data appropriately in all clusters into one group by measuring the quality of clusters and the minimum distance value of each cluster. In all baselines, these values improved as the number of data increased; however, MMC remained constant regardless of the number of data. It was confirmed that this properly solved the maximization and minimization problems pertaining to the similarity between clusters.

In the future, we plan to analyze *the education effects of our clustering algorithm*. Previously, Ikejiri et al. analyzed [30] group discussions targeting historical analogy in high-school classes. We will invoke this analysis for junior high-school students or university students using the algorithm proposed herein.

Acknowledgements This work was supported by JSPS KAKENHI Grant Numbers 16K16314 and 19K20631.

Compliance with ethical standards

Conflict of interest On behalf of all authors, the corresponding author states that there is no conflict of interest.

References

1. Aggarwal CC, Zhai C (2012) A survey of text clustering algorithms. Springer, Boston, pp 77–128
2. Althoff T, Dong XL, Murphy K, Alai S, Dang V, Zhang W (2015) Timemachine: timeline generation for knowledge-base entities. In: KDD'15. ACM, New York, NY, USA, pp 19–28
3. Baeza-Yates R (2005) Searching the future. In: Proceedings of the mathematical/formal methods in information retrieval workshop associated to SIGIR'05. ACM
4. Barforoush A, Shirazi H, Emami H (2017) A new classification framework to evaluate the entity profiling on the web: past, present and future. ACM Comput Surv (CSUR) 50(3):39:1–39:39
5. Boix-Mansilla V (2000) Historical understanding: beyond the past and into the present. In: Knowing, teaching, and learning history: national and international perspectives, pp 390–418

6. Branco P, Torgo L, Ribeiro RP (2016) A survey of predictive modeling on imbalanced domains. *ACM Comput Surv (CSUR)* 49(2):31:1–31:50
7. Caliński T, Harabasz J (1974) A dendrite method for cluster analysis. *Commun Stat Simul Comput* 3(1):1–27
8. Campos R, Dias G, Jorge AM, Jatowt A (2015) Survey of temporal information retrieval and related applications. *ACM Comput Surv (CSUR)* 47(2):15
9. Cardoso-Cachopo A, Oliveira AL (2007) Semi-supervised single-label text categorization using centroid-based classifiers. In: *SAC'07*. ACM, New York, NY, USA, pp 844–851
10. Cazals F, Mazauric D, Tetley R, Watrigant R (2019) Comparing two clusterings using matchings between clusters of clusters. *J Exp Algorithmics* 24(1):1–41
11. Chapelle O, Weston J, Schölkopf B (2002) Cluster kernels for semi-supervised learning. In: *NIPS'02*. MIT Press, Cambridge, MA, USA, pp 601–608
12. Cheng W, Hüllermeier E (2009) Combining instance-based learning and logistic regression for multilabel classification. *Mach Learn* 76(2):211–225
13. Chew MM, Bhowmick SS, Jatowt A (2018) Ranking without learning: towards historical relevance-based ranking of social images. In: *SIGIR'18*. ACM, New York, NY, USA, pp 1133–1136
14. Cong G, Lee W, Wu H, Liu B (2004) Semi-supervised text classification using partitioned em. In: Lee Y, Li J, Whang KY, Lee D (eds) *Database systems for advanced applications*. Lecture notes in computer science, vol 2973. Springer, Berlin, pp 482–493
15. David JS (2002) *A history of the future*, vol 41. History and theory, theme issue
16. Do QX, Lu W, Roth D (2012) Joint inference for event timeline construction. *EMNLP-CoNLL'12*. ACL, Stroudsburg, PA, USA, pp 677–687
17. Fahad A, Alshatri N, Tari Z, Alamri A, Khalil I, Zomaya AY, Foufou S, Bouras A (2014) A survey of clustering algorithms for big data: taxonomy and empirical analysis. *IEEE Trans Emerg Top Comput* 2(3):267–279
18. Fischer DH (1970) *Historians' fallacies: toward a logic of historical thought*. Harper & Row, Publishers, New York
19. Fu K, Mui J (1981) A survey on image segmentation. *Pattern Recogn* 13(1):3–16
20. Ghani R (2002) Combining labeled and unlabeled data for multi-class text categorization. In: *ICML'02*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, pp 187–194
21. Guha S, Rastogi R, Shim K (1998) Cure: an efficient clustering algorithm for large databases. In: *SIGMOD'98*. ACM, New York, NY, USA, pp 73–84
22. Guha S, Rastogi R, Shim K (1999) Rock: a robust clustering algorithm for categorical attributes. In: *ICDE'99*, pp 512–521
23. Holyoak KJ, Thagard P (1980) *Mental leaps: analogy in creative thought*. MIT Press, Cambridge
24. Holzmann H, Risse T (2014) Named entity evolution analysis on wikipedia. In: *WebSci'14*. ACM, New York, NY, USA, pp 241–242
25. Huet T, Biega J, Suchanek FM (2013) Mining history with le monde. In: *AKBC'13*. ACM, New York, NY, USA, pp 49–54
26. Ikejiri R (2011) Designing and evaluating the card game which fosters the ability to apply the historical causal relation to the modern problems. *Jpn Soc Educ Technol* 34(4):375–386 (in Japanese)
27. Ikejiri R, Fujimoto T, Tsubakimoto M, Yamauchi Y (2012) Designing and evaluating a card game to support high school students in applying their knowledge of world history to solve modern political issues. In: *ICoME'12*. Beijing Normal University
28. Ikejiri R, Sumikawa Y (2016) Developing a mining system to transfer historical causations to solving modern social issues. In: *WHA'16*
29. Ikejiri R, Sumikawa Y (2016) Developing world history lessons to foster authentic social participation by searching for historical causation in relation to current issues dominating the news. *J Educ Res Soc Stud* 84:37–48 (in Japanese)
30. Ikejiri R, Yoshikawa R, Sumikawa Y (2019) Designing and evaluating educational media for collaborative historical analogy. *Int J Educ Media Technol Jpn Assoc Educ Med Stud* 13(1):6–16
31. Ikejiri R, Yoshikawa R, Sumikawa Y (2020) Towards enhancing historical analogy: clustering users having different aspects of events. In: *FICC'19*. Springer International Publishing, pp 756–772
32. Jatowt A, Kanazawa K, Oyama S, Tanaka K (2009) Supporting analysis of future-related information in news archives and the web. In: *JCDL'09*. ACM, New York, NY, USA, pp 115–124
33. Kang F, Jin R, Sukthankar R (2006) Correlated label propagation with application to multi-label learning. In: *CVPR'06*. New York, NY, USA, pp 1719–1726
34. Kaufman L, Rousseeuw PJ (1990) *Finding groups in data: an introduction to cluster analysis*. Wiley, Hoboken
35. Kaufman L, Rousseeuw PJ (2008) *Partitioning around medoids (program PAM)*. Wiley-Blackwell, Hoboken, pp 68–125
36. Kong X, Ng MK, Zhou Z (2013) Transductive multilabel learning via label set propagation. *IEEE Trans Knowl Data Eng* 25(3):704–719
37. Larsen B, Aone C (1999) Fast and effective text mining using linear-time document clustering. In: *KDD'99*. ACM, New York, NY, USA, pp 16–22
38. Lee P (2005) Historical literacy: theory and research. *Int J Hist Learn Teach Res* 5(1):25–40
39. Lo H, Lin S, Wang H (2014) Generalized k-labelsets ensemble for multi-label and cost-sensitive classification. *IEEE Trans Knowl Data Eng* 26(7):1679–1691
40. Macqueen J (1967) Some methods for classification and analysis of multivariate observations. In: *5-th Berkeley symposium on mathematical statistics and probability*, pp 281–297
41. Mautz D, Ye W, Plant C, Böhm C (2018) Discovering non-redundant k-means clusterings in optimal subspaces. In: *KDD'18*. ACM, New York, NY, USA, p 19731982
42. Menc'ia EL, Park S, Fürnkranz J (2010) Efficient voting prediction for pairwise multilabel classification. *Neurocomputing* 73(7–9):1164–1176
43. Ministry of Education Culture SS (2020) *Technology: Japan course of study for senior high schools*. Accessed 13 2014 (in Japanese)
44. Ng RT, Han J (2002) Clarans: a method for clustering objects for spatial data mining. *IEEE Trans Knowl Data Eng* 14(5):1003–1016
45. Nigam K, McCallum AK, Thrun S, Mitchell T (2000) Text classification from labeled and unlabeled documents using em. *Mach Learn* 39(2–3):103–134
46. Noroozi O, Weinberger A, Biemans H, Mulder M, Chizari M (2012) Argumentation-based computer supported collaborative learning (ABCSCCL). A synthesis of fifteen years of research. *Educ Res Rev* 7(2):79–106
47. Ogata I, Kato T, Kabayama K, Kawakita M, Kishimoto M, Kuroda H, Sato T, Minamizuka S, Yamamoto H (1994) *Encyclopedia of historiography*. Koubundou Publishers Inc, Luhmann
48. Pise NN, Kulkarni P (2008) A survey of semi-supervised learning methods. In: *2008 International conference on computational intelligence and security*, vol 2, pp 30–34
49. Qi X, Davison BD (2009) Web page classification: features and algorithms. *ACM Comput Surv (CSUR)* 41(2):12:1–12:31
50. Radinsky K, Horvitz E (2013) Mining the web to predict future events. In: *WSDM'13*. ACM, New York, NY, USA, pp 255–264
51. Rasmussen CE (1999) The infinite gaussian mixture model. In: *NIPS'99*. MIT Press, Cambridge, MA, USA, pp 554–560

52. Read J, Pfahringer B, Holmes G, Frank E (2011) Classifier chains for multi-label classification. *Mach Learn* 85(3):333–359
53. Sebastiani F (2002) Machine learning in automated text categorization. *ACM Comput Surv (CSUR)* 34(1):1–47
54. Shi J, Malik J (2000) Normalized cuts and image segmentation. Technical report
55. Tsoumakas G, Katakis I, Vlahavas I (2010) Mining multi-label data. Springer, Boston, pp 667–685
56. van Drie J, van Boxtel C (2008) Historical reasoning: towards a framework for analyzing students' reasoning about the past. *Educ Psychol Rev* 20(2):87–110
57. Wang F, Zhang C (2006) Label propagation through linear neighborhoods. In: *ICML'06*. ACM, New York, NY, USA, pp 985–992
58. Wang W, Tsotsos J (2016) Dynamic label propagation for semi-supervised multi-class multi-label classification. *Pattern Recogn* 52:75–84
59. Xu D, Tian Y (2015) A comprehensive survey of clustering algorithms. *Ann Data Sci* 2(2):165–193
60. Xu R, Wunsch D (2005) Survey of clustering algorithms. *IEEE Trans Neural Netw* 16(3):645–678
61. Xu X, Ester M, Kriegel HP, Sander J (1998) A distribution-based clustering algorithm for mining in large spatial databases. In: *ICDE'98*. IEEE Computer Society, Washington, DC, USA, pp 324–331
62. Zhang M, Zhou Z (2014) A review on multi-label learning algorithms. *IEEE Trans Knowl Data Eng* 26(8):1819–1837
63. Zhang ML, Zhou ZH (2007) MI-knn: a lazy learning approach to multi-label learning. *Pattern Recogn* 40(7):2038–2048
64. Zhang T, Ramakrishnan R, Livny M (1996) Birch: an efficient data clustering method for very large databases. In: *SIGMOD'96*. ACM, New York, NY, USA, pp 103–114
65. Zhang Y, Jatowt A, Bhowmick S, Tanaka K (2015) Omnia mutantur, nihil interit: connecting past with present by finding corresponding terms across time. In: *ACL/IJCNLP*. ACL, pp 645–655
66. Zhang Y, Jatowt A, Tanaka K (2017) Temporal analog retrieval using transformation over dual hierarchical structures. In: *CIKM'17*. ACM, New York, NY, USA, pp 717–726
67. Zhou D, Bousquet O, Navin LT, Weston J, Scholkopf B (2004) Learning with local and global consistency. In: *NIPS'04*. MIT Press, pp 321–328
68. Zhu X (2005) Semi-supervised learning with graphs. Ph.D. thesis, Pittsburgh, PA, USA
69. Zhu X (2008) Semi-supervised learning literature survey. University of Wisconsin-Madison, Madison, p 2
70. Zoidi O, Fotiadou E, Nikolaidis N, Pitas I (2015) Graph-based label propagation in digital media: a review. *ACM Comput Surv* 47(3):48:1–48:35

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.