Biclustering Based on FCA and Partition Pattern Structures for Recommendation Systems

Nyoman Juniarta[1], Victor Codocedo[2], Miguel Couceiro[1], and Amedeo Napoli[1]

[1] Université de Lorraine, CNRS, Inria, LORIA, F-54000 Nancy, France
{nyoman.juniarta, miguel.couceiro, amedeo.napoli}@loria.fr
[2] Universidad Técnica Federico Santa María, Santiago, Chile
victor.codocedo@inria.cl

Abstract. This paper focuses on item recommendation for visitors in a museum within the framework of European Project CrossCult about cultural heritage. We present a theoretical research work about recommendation using biclustering. Our approach is based on biclustering using FCA and partition pattern structures. First, we recall a previous method of recommendation based on constant-column biclusters. Then, we propose an alternative approach that incorporates an order information and that uses coherent-evolution-on-columns biclusters. This alternative approach shares some common features with sequential pattern mining. Finally, given a dataset of visitor trajectories, we indicate how these approaches can be used to build a collaborative recommendation strategy.

Keywords: biclustering, FCA, pattern structures, recommendation

1 Introduction

CrossCult (http://www.crosscult.eu) is a European project whose idea is to support the emergence of a European cultural heritage by allowing visitors in different cultural sites (e.g. museum, historic city, archaeological site) to improve the quality of their visit by using adapted computer-based devices and to consider the visit at a European level. Such improvement can be accomplished by studying, among others, the possibility to build a dynamic recommendation system. This system should be able to produce a relevant suggestion on which part of a cultural site may be interesting for a specific visitor.

Here, our objective is to study a dynamic recommendation system for visitors in a museum. Given a new visitor V_n, the task is to suggest a museum item that may be interesting for him/her. Based on how a suggestion is made to a new visitor V_n, a recommendation system can be classified into one of the three following categories [1]:

- *Content-based recommendations*: The system makes a suggestion based only on the previous visited items of V_n. For example, if V_n visited mostly the items from prehistoric era, then the system recommends another item from that era.

- *Collaborative recommendations*: The system looks for previous users who have similar interest to V_n, and makes a suggestion based on their visited items. For example, if many of V_n's similar users have visited item I, then the system recommends this item.
- *Hybrid approaches*: The combination of content-based and collaborative approaches.

Our method belongs to the second category (collaborative recommendation). First we group all previous users based on their visit trajectories using biclustering. When V_n arrives, we try to find a G_s, i.e. a group of visitors who shares a similar interest to V_n. Then, based on the behavior of the visitors in G_s, we can suggest one item that may be interesting for V_n.

In this paper we will recall an approach in [6] that uses partition pattern structures to obtain biclusters with constant (or similar) values on the columns. Then we will propose an alternative approach that relies on this approach to mine another type of biclusters: those with coherent evolution on the columns (`CEC` biclusters). This bicluster type is useful when we are dealing with a dataset of trajectories where each trajectory corresponds to an ordered list of items. Furthermore, the mining of `CEC` biclusters can be related to sequential pattern mining, which we will explore in this paper.

This paper is organized as follows. First, we mention some related works about recommendation in Section 2. Then the basic background on biclustering is given in Section 3. Section 4 explains how to perform biclustering using partition pattern structures. The application of biclustering to recommendation systems will be presented in Section 5. Finally, we conclude our paper and outline some future works in Section 6.

2 Related work

In this section, we will describe related work about recommendation systems, biclustering, and Formal Concept Analysis (FCA).

FCA has been studied in collaborative movie recommendations for a user by looking at the ratings given by other users. In [5], FCA is used to generate a lattice from a binary matrix (with users as rows and movies as columns) as the formal context. This matrix is derived from a rating dataset which is binarized, such that the matrix contains only the information whether a user has rated a movie. The lattice is then drawn to select some neighbors – i.e. users who have rated the same movies as the new user – regardless of the rating values. In this way, the exhaustive search of neighbors can be avoided. The neighbors' ratings can be then studied to recommend movies rated by the neighbors but not yet rated by the new user.

Pattern structures [8,13] are a generalization of FCA, where the objects have more complex descriptions (e.g. sequence, graph, etc.). FCA was also extended into Triadic Concept Analysis, and it was shown in [13] that triadic concepts are in 1-1-correspondence with maximal biclusters of similar values.

Partition pattern structures are an instance of the pattern structure framework. They were used in a collaborative movie recommendation [7] by identifying similar-column biclusters within the rating matrix. Such a bicluster corresponds to a set of users with similar rating behavior (hence similar interest) across a set of movies. Therefore, to recommend a movie to a new user, there is a search for biclusters whose users have similar interest to him/her. Then, a recommendation is given by looking at the movies rated by users in the biclusters but not yet rated by the new user. Using the real MovieLens data, a study was also conducted based on Boolean matrix factorization [2].

Moreover, recommendation systems based on FCA and/or biclustering have been applied to other real world problems such as detection of future advertising terms for a company [10], educational orientation of Russian school graduates [11], and idea recommendation at a crowdsourcing project of Witology company [9]. Furthermore, a unified taxonomy of biclustering methods was proposed in [12].

3 Biclustering

In this section, we will recall the basic background and discuss illustrative examples of the different types of biclusters as described in [14]. We consider a dataset composed of a set of objects, each of which has values over a set of attributes. This dataset can be represented as a numerical matrix, where each cell ij indicates the value of object i w.r.t. attribute j.

One may be interested in finding which subset of objects possesses the same values w.r.t. a subset of attributes. Regarding the matrix representation, this is equivalent to the problem of finding a submatrix that has a constant value over all of its elements (example in Table 1). This task is called biclustering with constant values, which is a simultaneous clustering of the rows and columns of a matrix.

Table 1. A bicluster with constant value (shaded)

1	1	4	3	5
1	1	2	5	1
3	3	4	2	1

Other than constant values, the bicluster approach also focused on finding other types of submatrices, as shown in Table 2. A bicluster with constant columns (rows) is a submatrix where each column (row) has the same value, as illustrated in Table 2a (Table 2b, resp.).

In a bicluster with additive coherent values, the value of each cell ij follows the equation $\gamma + \alpha_i + \beta_j$, where γ is a constant, α_i is a constant value for row i, and β_j is a constant value for column j. For example, if $\gamma = 1$, $(\alpha_1, \alpha_2, \alpha_3, \alpha_4) = (3, 2, 4, 6)$, and $(\beta_1, \beta_2, \beta_3, \beta_4) = (0, -2, 1, -1)$, then we can obtain the bicluster

Table 2. Examples of some types of biclusters. (a) Constant columns, (b) constant rows, (c) additive coherent values, (d) multiplicative coherent values, (e) coherent evolution on the columns, and (f) coherent evolution on the rows.

4	2	5	3
4	2	5	3
4	2	5	3
4	2	5	3

(a)

1	1	1	1
2	2	2	2
4	4	4	4
3	3	3	3

(b)

4	2	5	3
3	1	4	2
5	3	6	4
7	5	8	6

(c)

4	2	5	3
2	1	2.5	1.5
8	4	10	6
6	3	7.5	4.5

(d)

1	2	4	3
3	5	7	6
2	3	8	4
4	5	9	8

(e)

1	2	4	3
0	1	1	2
5	4	6	4
6	5	7	5

(f)

in Table 2c. Similarly, we can obtain a bicluster with multiplicative coherent values as shown in Table 2d using a constant for each row and each column. The main difference is that, instead of adding, we multiply them.

Another interesting type is the CEC bicluster, also known as order-preserving submatrix [4]. In this type of bicluster, each row induces the same linear order across all columns. For example, in the bicluster in Table 2e, each row follows *column1* $\leq$ *column2* $\leq$ *column4* $\leq$ *column3*. Moreover, a bicluster with coherent evolution on the rows can be defined similarly, as shown in Table 2f.

Those different types of biclusters are useful when we are interested in identifying a group of people who behave similarly according to a set of attributes. This group identification is necessary in the task of collaborative recommendation, because in the process of making a suggestion to a person, we first identify the people who are similar to him/her. As suggested in [3] the bicluster-based recommendation may give a better performance than state-of-the-art recommendation algorithms.

4 Biclustering Using Partition Pattern Structures

Biclustering has many common elements with FCA. In FCA, from a binary matrix we try to find a maximal submatrix whose elements are 1. In other words, the objective is to identify maximal constant-value biclusters (but only for biclusters whose values are 1). Hence, a formal concept can be considered as a bicluster of objects and attributes. Furthermore, formal concepts are arranged in a concept lattice, that can describe the hierarchical relation among all biclusters.

Consider the matrix given by Table 3, where we are interested in finding constant-column biclusters. We recognize that the values of $\mathtt{m}_1$ "break" the objects into two sets: $\{\mathtt{g}_1, \mathtt{g}_2\}$ and $\{\mathtt{g}_3, \mathtt{g}_4\}$. The same "break" is also obtained from the values of $\mathtt{m}_4$. In particular, we can see that the pair $(\{\mathtt{g}_1, \mathtt{g}_2\}, \{\mathtt{m}_1, \mathtt{m}_4\})$ corresponds to a constant-column bicluster. Therefore, it is possible to mine this type of bicluster using this "breaking" – or "partitioning" – technique. Moreover, this technique can be performed using partition pattern structures – an extension of FCA.

In this section, first we will recall the constant-column biclustering approach using partition pattern structures [6]. After that, we will explain the possibility to perform CEC biclustering based on this approach.

4.1 Biclustering with Constant Columns

A partition $\mathtt{d} = \{p_i\}$ of a set $\mathtt{G}$ is a collection of $p_i \subseteq \mathtt{G}$ such that:

$$\bigcup_{p_i \in \mathtt{d}} p_i = \mathtt{G} \quad \text{and} \quad p_i \cap p_j = \emptyset \quad \text{whenever} \quad i \neq j. \tag{1}$$

Notice that when calculating the initial partitions, missing values can produce overlapping partitions (i.e. $p_i \cap p_j \neq \emptyset$). Consider the dataset given by Table 3 that has $\mathtt{G} = \{\mathtt{g}_1, \mathtt{g}_2, \mathtt{g}_3, \mathtt{g}_4\}$ as the set of objects and $\mathtt{M} = \{\mathtt{m}_1, \mathtt{m}_2, \mathtt{m}_3, \mathtt{m}_4, \mathtt{m}_5\}$ as the set of attributes. Here we can define a partition mapping $\delta : \mathtt{M} \rightarrow \mathtt{D}$. The partition is based on the fact that the values of the attribute are equal for all objects in a subset. For example, $\delta(\mathtt{m}_1) = \{\{\mathtt{g}_1, \mathtt{g}_2\}, \{\mathtt{g}_3, \mathtt{g}_4\}\}$ because $\mathtt{G}$ is partitioned as such regarding the value of $\mathtt{m}_1$, whereas $\delta(\mathtt{m}_4) = \{\{\mathtt{g}_1, \mathtt{g}_2\}, \{\mathtt{g}_1, \mathtt{g}_3, \mathtt{g}_4\}\}$. This partition overlaps on $\mathtt{g}_1$ since this object has a missing value on $\mathtt{m}_4$. Intuitively, $\mathtt{g}_1$ can be grouped with either $\{\mathtt{g}_2\}$ or $\{\mathtt{g}_3, \mathtt{g}_4\}$ w.r.t. $\mathtt{m}_4$.

Table 3. A dataset with 4 objects and 5 attributes

	$\mathtt{m}_1$	$\mathtt{m}_2$	$\mathtt{m}_3$	$\mathtt{m}_4$	$\mathtt{m}_5$
$\mathtt{g}_1$	1	5	3	?	7
$\mathtt{g}_2$	1	1	4	2	7
$\mathtt{g}_3$	2	5	4	5	3
$\mathtt{g}_4$	2	5	4	5	7

The space $\mathtt{D}$ of all partitions over $\mathtt{G}$ is a complete lattice, where the meet and join of two partitions $\mathtt{d}_1 = \{p_i\}$ and $\mathtt{d}_2 = \{p_j\}$ are defined as:

$$\mathtt{d}_1 \sqcap \mathtt{d}_2 = \left(\bigcup_{i,j} p_i \cap p_j \right)^+ \tag{2}$$

$$\mathtt{d}_1 \sqcup \mathtt{d}_2 = \left(\bigcup_{p_i \cap p_j \neq \emptyset} p_i \cup p_j \right)^+ \tag{3}$$

where $(.)^+$ is a closure that preserves only the maximal components in $\mathtt{d}$. For example, $\delta(\mathtt{m}_1) \sqcap \delta(\mathtt{m}_4) = \{\{\mathtt{g}_1, \mathtt{g}_2\}, \{\mathtt{g}_1\}, \{\mathtt{g}_3, \mathtt{g}_4\}\}^+ = \{\{\mathtt{g}_1, \mathtt{g}_2\}, \{\mathtt{g}_3, \mathtt{g}_4\}\}$, and $\delta(\mathtt{m}_1) \sqcup \delta(\mathtt{m}_4) = \{\{\mathtt{g}_1, \mathtt{g}_2\}, \{\mathtt{g}_1, \mathtt{g}_2, \mathtt{g}_3, \mathtt{g}_4\}, \{\mathtt{g}_1, \mathtt{g}_3, \mathtt{g}_4\}\}^+ = \{\{\mathtt{g}_1, \mathtt{g}_2, \mathtt{g}_3, \mathtt{g}_4\}\}$.

The order between any two partitions is given by the subsumption relation:

$$\mathtt{d}_1 \sqsubseteq \mathtt{d}_2 \iff \mathtt{d}_1 \sqcap \mathtt{d}_2 = \mathtt{d}_1 \tag{4}$$

Given a set of attributes $\mathtt{M}$, a partition space $\mathtt{D}$, and a mapping δ, a partition pattern structures for constant-column biclustering is determined by the triple

$(\mathtt{M}, (\mathtt{D}, \sqcap), \delta)$. A pair $(\mathtt{A}, \mathtt{d})$ is then called a partition pattern concept (pp-concept) iff $\mathtt{A}^{\Box} = \mathtt{d}$ and $\mathtt{d}^{\Box} = \mathtt{A}$, where:

$$\mathtt{A}^{\Box} = \bigsqcap_{\mathtt{m} \in \mathtt{A}} \delta(\mathtt{m}) \qquad \mathtt{A} \subseteq \mathtt{M} \tag{5}$$

$$\mathtt{d}^{\Box} = \{\mathtt{m} \in \mathtt{M} | \mathtt{d} \sqsubseteq \delta(\mathtt{m})\} \qquad \mathtt{d} \in \mathtt{D} \tag{6}$$

For any partition component $p \in \mathtt{d}$, each pair $(p, \mathtt{A})$ corresponds to a constant-column bicluster. For example, from the concept $(\{\mathtt{m}_1, \mathtt{m}_4\}, \{\{\mathtt{g}_1, \mathtt{g}_2\}, \{\mathtt{g}_3, \mathtt{g}_4\}\})$, two biclusters can be obtained: $(\{\mathtt{g}_1, \mathtt{g}_2\}, \{\mathtt{m}_1, \mathtt{m}_4\})$ and $(\{\mathtt{g}_3, \mathtt{g}_4\}, \{\mathtt{m}_1, \mathtt{m}_4\})$.

4.2 Biclustering with Coherent Evolution on the Columns

In a dataset of movie ratings, bicluster with constant columns is useful to identify a set of users with the same taste regarding a set of movies. Another interesting problem arises, e.g. when the dataset contains watching order. In that case, we may be interested in finding a set of users who watch a set of movies in the same order. This problem corresponds to CEC biclustering, where the objective is to find a set of rows which has coherent evolution over a set of columns, as previously described in Section 3. In the current section, we will explain the possible application of partition pattern structures to discover CEC biclusters.

Table 4. A dataset with 5 objects and 5 attributes

	$\mathtt{m}_1$	$\mathtt{m}_2$	$\mathtt{m}_3$	$\mathtt{m}_4$	$\mathtt{m}_5$
$\mathtt{g}_1$	1	2	3	4	5
$\mathtt{g}_2$	4	2	1	?	3
$\mathtt{g}_3$	2	3	4	1	1
$\mathtt{g}_4$	5	4	2	3	1
$\mathtt{g}_5$	2	1	5	4	3

Consider the dataset given by Table 4, with the set of attributes $\mathtt{G} = \{\mathtt{g}_1, \mathtt{g}_2, \mathtt{g}_3, \mathtt{g}_4, \mathtt{g}_5\}$. First, we have to list each pair of attributes and the partition according to the pair's evolution. For the pair $\mathtt{p}_{1,2} = (\mathtt{m}_1, \mathtt{m}_2)$, the partition is $\{\{\mathtt{g}_1, \mathtt{g}_3\}, \{\mathtt{g}_2, \mathtt{g}_4, \mathtt{g}_5\}\}$ because in $\mathtt{g}_1$ and $\mathtt{g}_3$, $\mathtt{m}_1$ is less than $\mathtt{m}_2$, whereas in $\mathtt{g}_2$, $\mathtt{g}_4$, and $\mathtt{g}_5$, $\mathtt{m}_1$ is greater. As in Subsection 4.1, missing values generate an overlapping partition. For instance, the pair $\mathtt{p}_{1,4}$ gives rise to the partition is $\{\{\mathtt{g}_1, \mathtt{g}_2, \mathtt{g}_5\}, \{\mathtt{g}_2, \mathtt{g}_3, \mathtt{g}_4\}\}$. Furthermore, two columns with the same value (e.g. $\mathtt{g}_3$ in $\mathtt{m}_4$ and $\mathtt{m}_5$) can also produce an overlapping partition because, by our definition of CEC bicluster in Section 3, they satisfy $\mathtt{m}_4 \leq \mathtt{m}_5$ and $\mathtt{m}_5 \leq \mathtt{m}_4$. Therefore, the partition for $\mathtt{p}_{4,5}$ is $\{\{\mathtt{g}_1, \mathtt{g}_2, \mathtt{g}_3\}, \{\mathtt{g}_2, \mathtt{g}_3, \mathtt{g}_4, \mathtt{g}_5\}\}$. Some pairs and their partitions are listed in Table 5.

Since a CEC partition is defined by at least two attributes, the partition mapping becomes $\gamma : \mathtt{P} \to \mathtt{D}$. For instance, $\gamma(\mathtt{p}_{1,2}) = \{\{\mathtt{g}_1, \mathtt{g}_3\}, \{\mathtt{g}_2, \mathtt{g}_4, \mathtt{g}_5\}\}$.

Table 5. Some examples of partitions over Table 4

Pair	Partition
$\mathtt{p}_{1,2}$	$\{\{\mathtt{g}_1,\mathtt{g}_3\},\{\mathtt{g}_2,\mathtt{g}_4,\mathtt{g}_5\}\}$
$\mathtt{p}_{1,3}$	$\{\{\mathtt{g}_1,\mathtt{g}_3,\mathtt{g}_5\},\{\mathtt{g}_2,\mathtt{g}_4\}\}$
$\mathtt{p}_{1,4}$	$\{\{\mathtt{g}_1,\mathtt{g}_2,\mathtt{g}_5\},\{\mathtt{g}_2,\mathtt{g}_3,\mathtt{g}_4\}\}$
$\mathtt{p}_{2,3}$	$\{\{\mathtt{g}_1,\mathtt{g}_3,\mathtt{g}_5\},\{\mathtt{g}_2,\mathtt{g}_4\}\}$
$\mathtt{p}_{2,5}$	$\{\{\mathtt{g}_1,\mathtt{g}_2,\mathtt{g}_5\},\{\mathtt{g}_3,\mathtt{g}_4\}\}$

As in Subsection 4.1, given a set of attribute pairs $\mathtt{P}$, a partition space $\mathtt{D}$, and the mapping function γ, a partition pattern structures for coherent-evolution biclustering is determined by the triple $(\mathtt{P},(\mathtt{D},\sqcap),\gamma)$. A pp-concept is a pair $(\mathtt{B},\mathtt{d})$ such that $\mathtt{B}^{\square}=\mathtt{d}$ and $\mathtt{d}^{\square}=\mathtt{B}$, where:

$$\mathtt{B}^{\square}=\prod_{\mathtt{p}\in\mathtt{B}}\gamma(\mathtt{p}) \qquad \mathtt{B}\subseteq\mathtt{P} \tag{7}$$

$$\mathtt{d}^{\square}=\{\mathtt{p}\in\mathtt{P}|\mathtt{d}\sqsubseteq\gamma(\mathtt{p})\} \qquad \mathtt{d}\in\mathtt{D} \tag{8}$$

Here, the extent of a pp-concept is a set of attribute pairs. We can obtain a CEC bicluster in a pp-concept if there is a clique among the attributes in the pairs. For example, consider the pp-concept $\mathtt{ppc}_1$ with extent $\{\mathtt{p}_{1,2},\mathtt{p}_{1,3},\mathtt{p}_{2,3}\}$ and intent $\{\{\mathtt{g}_1,\mathtt{g}_3\},\{\mathtt{g}_5\},\{\mathtt{g}_2,\mathtt{g}_4\}\}$. Its extent forms a clique among $\mathtt{m}_1$, $\mathtt{m}_2$, and $\mathtt{m}_3$, since all pairings of any two of those attributes are included.

If a pp-concept $(\mathtt{B},\mathtt{d})$ contains a set of attributes $\mathtt{A}$ that forms a clique, then each pair $(p,\mathtt{A})$, for any partition component $p\in\mathtt{d}$, corresponds to a CEC bicluster. For example, from $\mathtt{ppc}_1$, we can obtain bicluster $(\{\mathtt{g}_1,\mathtt{g}_3\},\{\mathtt{m}_1,\mathtt{m}_2,\mathtt{m}_3\})$.

4.3 Comparison with Sequential Pattern Mining

A sequence is an ordered list $\langle s_1s_2\dots s_m\rangle$, where s_i is an itemset $\{i_1,\dots,i_n\}$. A sequence $s=\langle s_1s_2\dots s_m\rangle$ is a subsequence of $s'=\langle s'_1s'_2\dots s'_n\rangle$, denoted by $s\preceq s'$, if there exist indices $1\le i_1<i_2<\dots<i_m\le n$ such that $s_j\subseteq s'_{i_j}$ for all $j=1\dots m$ and $m\le n$. For example, the sequence $\langle\{a\}\{d\}\rangle$ is a subsequence of $\langle\{a,b\}\{a,c,d\}\rangle$, while sequence $\langle\{c\}\{d\}\rangle$ is not.

Notice that the problem of retrieving CEC biclusters can be thought of as a particular type of sequential pattern mining where each itemset is composed by only one item, i.e. the sequences are an ordered list of items. Mining sequential patterns means retrieving frequent subsequences (i.e., subsequences that are present in more than n sequences) and for which there exist many efficient algorithms [16,15].

The CEC biclustering differs from sequential pattern mining when we allow overlaps in the partitions. Consider Table 4 as a sequential dataset. Each number in row x column y corresponds to the itemset when item y appears in the sequence x. For example, the sequence of object $\mathtt{g}_3$ is $\langle\{\mathtt{m}_4,\mathtt{m}_5\}\{\mathtt{m}_1\}\{\mathtt{m}_2\}\{\mathtt{m}_3\}\rangle$.

Let us consider items $\mathtt{m}_1$, $\mathtt{m}_4$, and $\mathtt{m}_5$. According to sequential pattern mining, $\mathtt{g}_3$ is different from $\mathtt{g}_4$, because $\mathtt{m}_4$ and $\mathtt{m}_5$ appear in the same itemset in $\mathtt{g}_3$. On the other hand, according to CEC biclustering with overlaps, $\mathtt{g}_3$ is similar to $\mathtt{g}_4$, because in both objects $\mathtt{m}_5 \leq \mathtt{m}_4 \leq \mathtt{m}_1$.

5 Recommendation

In the context of CrossCult, we are working on a visitor dataset that comprises several trajectories in a museum. Within this project, our main objective is to build a dynamic recommendation system for new visitors. This system should be able to suggest a museum item to visitors based on their trajectories and by looking at the trajectories of previous visitors. Also, it should be able to update the suggestion as they move inside the museum.

5.1 Matrix as order of interest

For each item in the museum, we can measure (e.g. by rating, duration of visit, etc.) their level of interestingness from a set of visitors. An example is shown in Table 6, where the number in cell xy is the ranking of item y according to visitor x. Here we have 3 visitors ($\mathtt{v}_1$, $\mathtt{v}_2$, and $\mathtt{v}_3$) in the database and 1 target visitor ($\mathtt{v}_a$). Among the existing visitors, only $\mathtt{v}_1$ has complete values over all five items. According to this visitor $\mathtt{i}_1$ is the best, followed by $\mathtt{i}_2$, $\mathtt{i}_3$, $\mathtt{i}_4$, and the worst $\mathtt{i}_5$. For $\mathtt{v}_2$ ($\mathtt{v}_3$), the order of interest of $\mathtt{i}_2$ ($\mathtt{i}_4$ resp.) is not known.

The target visitor ($\mathtt{v}_a$) has visited only three items, with the same order of preference in $\mathtt{i}_1$ and $\mathtt{i}_2$. This visitor will be included in the bicluster ($\{\mathtt{v}_1, \mathtt{v}_2, \mathtt{v}_a\}$, $\{\mathtt{i}_1, \mathtt{i}_2, \mathtt{i}_3\}$). The other two members of this bicluster do not agree on the order of interest of $\mathtt{i}_4$ and $\mathtt{i}_5$. Hence, we should suggest $\mathtt{i}_5$ to him/her, since one visitor ($\mathtt{v}_2$) similar to him/her ranked it first.

Table 6. Order of interest of 5 items, observed from certain visitors

	$\mathtt{i}_1$	$\mathtt{i}_2$	$\mathtt{i}_3$	$\mathtt{i}_4$	$\mathtt{i}_5$
$\mathtt{v}_1$	1	2	3	4	5
$\mathtt{v}_2$	3	?	4	2	1
$\mathtt{v}_3$	2	4	3	?	1
$\mathtt{v}_a$	1	1	2	?	?

5.2 Matrix as order of visit

Consider the dataset given by Table 7 about 4 visitors in a museum with 7 items. As explained in Subsection 4.3, this table can be regarded as a sequential dataset. The numbers in row x indicate the path of visitor $\mathtt{v}_x$. For example, the path of $\mathtt{v}_2$ is $\mathtt{i}_6 \rightarrow \mathtt{i}_1 \rightarrow \mathtt{i}_4 \rightarrow \mathtt{i}_2 \rightarrow \mathtt{i}_3 \rightarrow \mathtt{i}_7 \rightarrow \mathtt{i}_5$.

Now we have a new visitor $\mathtt{v}_a$ who recently arrives to the museum. He/She visits $\mathtt{i}_3$, followed by $\mathtt{i}_4$. Our task is to recommend a new item to her, by studying

Table 7. Order of visit of 7 items, observed from certain visitors

	$\mathtt{i}_1$	$\mathtt{i}_2$	$\mathtt{i}_3$	$\mathtt{i}_4$	$\mathtt{i}_5$	$\mathtt{i}_6$	$\mathtt{i}_7$
$\mathtt{v}_1$	1	2	3	4	5	6	7
$\mathtt{v}_2$	2	4	5	3	7	1	6
$\mathtt{v}_3$	4	2	1	5	6	3	7
$\mathtt{v}_4$	7	3	1	4	2	6	5
$\mathtt{v}_a$	?	?	1	2	?	?	?

the CEC biclusters over the first four visitors. Some of those biclusters are listed in Table 8. From B7, we can see that all four visitors visit $\mathtt{i}_7$ after $\mathtt{i}_2$. In B3, $\mathtt{v}_1$ and $\mathtt{v}_3$ follow the same order w.r.t. $\{\mathtt{i}_3, \mathtt{i}_4, \mathtt{i}_5, \mathtt{i}_7\}$: $\mathtt{i}_3 \rightarrow \mathtt{i}_4 \rightarrow \mathtt{i}_5 \rightarrow \mathtt{i}_7$. These two visitors also agree in the order of items $\{\mathtt{i}_1, \mathtt{i}_4, \mathtt{i}_5, \mathtt{i}_7\}$, as seen in B4.

Table 8. Some CEC biclusters in Table 7

#	Visitors	Items (in order)
B1	$\mathtt{v}_1$	$\mathtt{i}_1, \mathtt{i}_2, \mathtt{i}_3, \mathtt{i}_4, \mathtt{i}_5, \mathtt{i}_6, \mathtt{i}_7$
B2	$\mathtt{v}_2$	$\mathtt{i}_6, \mathtt{i}_1, \mathtt{i}_4, \mathtt{i}_2, \mathtt{i}_3, \mathtt{i}_7, \mathtt{i}_5$
B3	$\mathtt{v}_1, \mathtt{v}_3$	$\mathtt{i}_3, \mathtt{i}_4, \mathtt{i}_5, \mathtt{i}_7$
B4	$\mathtt{v}_1, \mathtt{v}_3$	$\mathtt{i}_1, \mathtt{i}_4, \mathtt{i}_5, \mathtt{i}_7$
B5	$\mathtt{v}_1, \mathtt{v}_4$	$\mathtt{i}_3, \mathtt{i}_5, \mathtt{i}_6$
B6	$\mathtt{v}_2, \mathtt{v}_3$	$\mathtt{i}_6, \mathtt{i}_1, \mathtt{i}_4, \mathtt{i}_5$
B7	$\mathtt{v}_1, \mathtt{v}_2, \mathtt{v}_3, \mathtt{v}_4$	$\mathtt{i}_2, \mathtt{i}_7$

Those CEC biclusters can be studied to give a recommendation to $\mathtt{v}_a$ by focusing on those visitors that are similar to him/her. Thus, we can propose a recommendation strategy that follows sequential patterns in the dataset. The idea behind is the following: if many visitors have the path $\mathtt{i}_a \rightarrow \mathtt{i}_b \rightarrow \mathtt{i}_c$, then we should recommend item $\mathtt{i}_c$ to a visitor who has done $\mathtt{i}_a \rightarrow \mathtt{i}_b$.

Since $\mathtt{v}_a$ has path $\mathtt{i}_3 \rightarrow \mathtt{i}_4$, we focus on the CEC biclusters that have those two items, i.e. B1, B2, and B3. One of those biclusters (B2) has a different ordering ($\mathtt{i}_3$ after $\mathtt{i}_4$), and thus we filter it out. Then, in B3 for example, the path is $\mathtt{i}_3 \rightarrow \mathtt{i}_4 \rightarrow \mathtt{i}_5 \rightarrow \mathtt{i}_7$. Therefore, we can recommend $\mathtt{i}_5$ to $\mathtt{v}_a$.

6 Conclusion

In this work, we have explored an approach to build collaborative recommendation strategy for visitors in a museum. This strategy takes into account the order of interest or the order of visit for each visitor, and we showed how to use CEC biclustering to obtain a set of similar visitors. We also presented a technique for mining CEC biclusters based on FCA using partition pattern structures. This recommendation strategy can be applied to any dataset where the order of items is relevant.

As future work, we intend to explore recommendations based on the order of visit. The problem to be solved is how to model visitors who visit a single item multiple times (for example, $\mathtt{i}_1$, $\mathtt{i}_2$, and back to $\mathtt{i}_1$).

Another noteworthy question is how to measure the "score" of each bicluster in order to rank recommendations for a new visitor. Ranking candidate items was studied in [3] for constant-value biclusters, and it is possible to extend this work to `CEC` biclusters. Moreover, further comparisons of `CEC` biclustering and sequential pattern mining should be investigated, in particular, regarding their complexities and their results. Finally, an implementation of the `CEC` biclustering using partition pattern structures and an empirical study on real-world data should be performed to measure its complexity and efficiency.

References

1. Adomavicius, G., Tuzhilin, A.: Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. IEEE transactions on knowledge and data engineering **17**(6), 734–749 (2005)
2. Akhmatnurov, M., Ignatov, D.I.: Context-aware recommender system based on boolean matrix factorisation. In: CLA. pp. 99–110 (2015)
3. Alqadah, F., Reddy, C.K., Hu, J., Alqadah, H.F.: Biclustering neighborhood-based collaborative filtering method for top-n recommender systems. Knowledge and Information Systems **44**(2), 475–491 (2015)
4. Ben-Dor, A., Chor, B., Karp, R., Yakhini, Z.: Discovering local structure in gene expression data: the order-preserving submatrix problem. Journal of computational biology **10**(3-4), 373–384 (2003)
5. du Boucher-Ryan, P., Bridge, D.: Collaborative recommending using formal concept analysis. Knowledge-Based Systems **19**(5), 309–315 (2006)
6. Codocedo, V., Napoli, A.: Lattice-based biclustering using partition pattern structures. In: Proceedings of the Twenty-first European Conference on Artificial Intelligence. pp. 213–218. IOS Press (2014)
7. Codocedo-Henríquez, V.: Contributions à l'indexation et à la récupération d'information utilisant l'analyse formelle de concepts. Ph.D. thesis, Université de Lorraine (2015), `http://www.theses.fr/2015LORR0143`
8. Ganter, B., Kuznetsov, S.O.: Pattern structures and their projections. In: International Conference on Conceptual Structures. pp. 129–142. Springer (2001)
9. Ignatov, D.I., Kaminskaya, A.Y., Konstantinova, N., Malyukov, A., Poelmans, J.: FCA-based recommender models and data analysis for crowdsourcing platform Witology. In: International Conference on Conceptual Structures. pp. 287–292. Springer (2014)
10. Ignatov, D.I., Kuznetsov, S.O., Poelmans, J.: Concept-Based Biclustering for Internet Advertisement. In: 2012 IEEE 12th International Conference on Data Mining Workshops. pp. 123–130. Brussels, Belgium (Jun 2009), `http://arxiv.org/abs/0906.4982`, arXiv: 0906.4982
11. Ignatov, D.I., Poelmans, J., Zaharchuk, V.: Recommender system based on algorithm of bicluster analysis RecBi. arXiv preprint arXiv:1202.2892 (2012)
12. Ignatov, D.I., Watson, B.W.: Towards a unified taxonomy of biclustering methods. arXiv preprint arXiv:1702.05376 (2017)

13. Kaytoue, M., Kuznetsov, S.O., Macko, J., Napoli, A.: Biclustering meets triadic concept analysis. Annals of Mathematics and Artificial Intelligence **70**(1-2), 55–79 (2014)
14. Madeira, S.C., Oliveira, A.L.: Biclustering algorithms for biological data analysis: a survey. IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB) **1**(1), 24–45 (2004)
15. Wang, J., Han, J.: BIDE: Efficient mining of frequent closed sequences. In: Data Engineering, 2004. Proceedings. 20th International Conference on. pp. 79–90. IEEE (2004)
16. Yan, X., Han, J., Afshar, R.: CloSpan: Mining: Closed sequential patterns in large datasets. In: Proceedings of the 2003 SIAM international conference on data mining. pp. 166–177. SIAM (2003)