Visual Analysis of User-Driven Association Rule Mining

Wei Chen\textsuperscript{a,1}, Cong Xie\textsuperscript{b,1,∗}, Pingping Shang\textsuperscript{a}, Qunsheng Peng\textsuperscript{a}

\textsuperscript{a}State Key Lab of CAD\&CG Zhejiang University, Hangzhou, China
\textsuperscript{b}Stony Brook University, New York, USA

Abstract

Association rules have been widely used for detecting relations between attribute-value pairs of categorical datasets. Existing solutions of mining interesting association rules are based on the support-confidence theory. However, it is non-trivial for the user to understand and modify the rules or the results of intermediate steps in the mining process, because the interestingness of rules might differ largely for various tasks and users. In this paper we reinforce conventional association rule mining process by mapping the entire process into a visualization assisted loop, with which the user workload for modulating parameters and mining rules is reduced, and the mining efficiency is greatly improved. A hierarchical matrix-based visualization technique is proposed for the user to explore the measure value and the intermediate results of association rules. We also design a set of visual exploration tools to support interactively inspection and manipulation of mining process. The effectiveness and usability of our approach is demonstrated with two scenarios.

Keywords: Categorical Data, Association Rules, Visual Analysis

1. Introduction

1.1. Background

The association rules mining [1] is an effective method to find relationships between different attribute-value pairs in different datasets. For example, an online commercial company records a set of commodities in a customer’s basket. It is critical for a sales manager to learn the co-occurred purchased items, such as a customer who buys bread is also likely to purchase butter. The sales manager can use the rules for decision making, recommendation and other marketing activities.

However, strong rules found by the automatic rule mining methods (e.g., support-confidence framework) are not necessarily interesting [2]. The judgment of whether a given rule is interesting may vary based on different factors:

- Interestingness is based on domain-specific knowledge and mining tasks. The interesting patterns of rules can differ largely under different usage scenarios. For example, diamond can have a very low sales amount, but analysts are still interested because of its high price. However, the traditional support-confidence framework will exclude this item due to its low purchasing frequency.

- Interestingness is based on structure of the data. Association may involve specific structures and sequences. For example, customers tend to buy headsets after buying a digital audio player. Rules with a desirable sequence are interesting in such cases.

- Interestingness can be subjective. The user assesses a rule based on his or her knowledge, experiences, and understanding of their tasks, which may differ from one to another.

To identify interesting rules, previous mining methods either allow the user to extend the existing association rules by developing different measures [3], or narrow the interesting rules by setting various constraints for certain attributes, itemsets, and rules [4] (e.g., specifying the interesting subset in database). However, the automatic mining process is actually hard to understand [5], which makes it challenging for defining measures and constraints. Meanwhile, most of these works do not provide methods for modulating the mining process interactively.

Existing visualization methods [6] [7] for rule mining only support basic interactions, such as searching and deleting rules. They are incapable of various tasks that require user expertise and domain knowledge, such as defining measures or modulating different types of constraints.

To deal with the above problems, a user-driven visual analysis pipeline [8] is proposed. It employs a visual-assisted rule mining pipeline that reinforces the iterative mining loop for understanding the learning process, which also enables the modulation of the process of mining interesting rules. However, scalability is one of the major challenges because it is difficult to visualize a large amount of rules in the screen space. Another problem is its usability since this method only deal with online transaction dataset.

In this paper, we extend the existing visual analysis of rule
mining approach [8] by dealing with the scalability and usability problems. In particular, we introduce a matrix-based visualization method to present the measure computation, the distribution of interesting itemsets and the intermediate results of rule mining. We organized the matrix visualization of association rules with a hierarchical structure by aggregation. The exploration and inspection of the rules in different granularities are allowed. The interactions are also designed for manipulation of the measures, constraints, and the rule results of the intermediate steps.

The rest of our paper is structured as follows. Section 2 reviews related work. Section 3 defines the tasks and and Section 4 gives an overview of our approach. Section 5 introduces our model and algorithm of the user-driven association rule mining. Section 6 and 7 describe the visualization and interactive exploration of the mining process, respectively. Two real-world case studies with different datasets were conducted to verify the usability of our method in Section 8. Section 9 ends with conclusions and future work.

2. Related Work

Association Rule [1] is proposed to identify relationships between items in very large databases. It finds frequent itemsets iteratively in a bottom-up manner, such as Apriori [9]. The support-confidence framework is commonly used to find frequent itemsets in the market basket analysis.

2.1. Automatic Analysis of Association Rules

In a real world association mining problem, the mining process usually was modulated according to the specific scenarios. Existing work dealt with this by choosing different measures and constraints.

**Interestingness Measures** Typically, different patterns were focused based on different datasets and domain-specified semantics. Some existing works used alternative interesting measure to extend or replace the support-confidence framework. Geng et al. [3] described these interesting measures in details.

**Constraint-based Mining** Constraint-based strategies are used to target at interesting rules and reduce the search space when dealing with large datasets. Srikant et al. [4] integrated binary constraints on items into the association rules. Ng et al. [10] characterized various constraints (contains, minimum, maximum, count, sum) according to anti-monotonicity and succinctness.

However, most of the existing works did not allow the user to modulate their measures or the itemsets found in the intermediate steps. Our approach advances these algorithms by integrating the visual analysis to set interesting measures and find the desired itemsets.

2.2. Visual Analysis of Association Rules

Various visualization methods [6] were used for exploring and explaining the association rules, such as mosaic plot [5], scatter plot [11], node-link graph [12], and matrix [13] [14]. Other visualization approaches for categorical data [15] [16] and high-dimensional data [17] [18] [19] can also be employed since the itemsets can be regarded as a large set of categorical variables.

Existing visualization methods mainly visualized the mining results subject to different tasks [20], [21]. ARVis [22] was proposed for rummaging and validating association rules. Sekhatvat et al. [23] provided multiple views to inspect the entire association rule set visually. Some works visualized the association rules with constraints, such as soft maximal association rules [24]. Zhao et al. [7] proposed to show the result of intermediate steps.

Most of the above methods focused on the explanation of the final association rule result. The user was neither able to propose self-defined measures for various tasks, nor to modify the constraints of different types during the rule mining algorithm with existing methods. The user-driven visual analysis of rule mining [8] was proposed to mapping the mining process to a visual analysis pipeline. Our paper extends this method by dealing with scalability and usability problems.

3. Problem and Tasks

The association mining problem is as defined as follows: Let \( R = \{A_1, A_2, \ldots, A_n\} \) be a set of categorical attributes (e.g., (“Milk”, “Bread”) in transaction dataset). Each attribute \( A_i \) has a domain \( D_i \). Mining a strong association rule is finding the implication of the form:

\[
X \Rightarrow Y
\]

in which \( X \) and \( Y \) are conjunction of attribute-value pairs (items): \( X = \{A_{x_1} = v_{x_1}, A_{x_2} = v_{x_2}, \ldots, A_{x_k} = v_{x_k}\} \) and \( Y = \{A_{y_1} = v_{y_1}, A_{y_2} = v_{y_2}, \ldots, A_{y_l} = v_{y_l}\} \), where \( A_{x_i}, A_{y_j} \in R, v_{x_i} \in D_{x_i} \), \( v_{y_j} \in D_{y_j} \), \( 1 \leq k, l \leq n \), and \( X \cap Y = \emptyset \). \( X \) is the Left Hand Side (LHS) itemset of a rule and \( Y \) is the Right Hand Side (RHS) itemset. For example, in the rule \( r_1: \{\text{“Milk” = “yes”, “Bread” = “yes”}\} \Rightarrow \{\text{“Butter” = “yes”}\} \). \{“Milk” = “yes”, “Bread” = “yes”\} \Rightarrow \{“Butter” = “yes”\} is the LHS itemset and \{“Butter” = “yes”\} is the RHS itemset. To simplify the binary attributes, we use \{“Milk”, “Bread”\} to represent the occurrence of items in itemsets. So \( r_1 \) can be rewritten as \( \{\text{“Milk”, “Bread”}\} \Rightarrow \{\text{“Butter”}\} \).

This paper focuses on finding association relationships in two different datasets: online transaction and MovieLens dataset. We use the online transaction data as an example to illustrate our approach in the following paper.

**Online Transaction Dataset**: An online E-transaction dataset records about 10,000 customers of a company. Every commodity contains a binary attribute that denotes presence or absence in every customers basket. Each customer has multiple attributes, including age group, gender, constellation, and account registration date.

**MovieLens Dataset**: It records around one million ratings given by more than 6,000 viewers on about 4,000 movies [25]. For each viewer in the dataset, he or she rates a part of the movies between 1 to 5 stars. The viewers also have information such as location, age group, gender, and occupation.
To understand how interesting rules can be found based on the strong rules, we have discussions with the target users (See section 8). We summarize that they are interested in the following tasks in analysis of the interesting rule mining:

- **T1** Examining the strong rules found by the mining process.
- **T2** Identifying interesting rules with specific constraints, such as filtering the strong rules by domain knowledge.
- **T3** Defining customized measures of interestingness.

![Figure 1: Black arrows: Conventional association rule mining pipeline. Orange arrows: Our pipeline reinforces this process by integrating a suite of visual representations and interactions.](image)

4. Approach Overview

To fulfill the tasks, we firstly use an iterative pipeline for discovery of the strong association rules (See black arrows in Fig. 1). It consists of two main stages. The first stage (Fig. 1 (a)) iteratively searches itemsets from the raw data. The second stage (Fig. 1 (b)) generates a set of interesting rules based on the given measures and constraints. While the user can interfere the mining process by tuning some parameters or specifying measures or constraints, it is incapable of interpreting the complexity of versatile measures, constraints and itemsets. In addition, only the final input, e.g., the mined rules, are visualized in the entire rule mining process. As such, finding interesting rules by tuning parameters in the “black-box” process can be extremely time-consuming.

To address these challenges, we have designed a new visual-assisted rule mining pipeline (See orange arrows in Fig. 1) in which the user is embedded into the mining process and equipped with tools to explore complex itemsets and specify constraints. We believe that the key to reinforce the automatic mining process with a visual analysis approach is to construct a user-driven representation that enables a better understanding of association rule mining, the underlying dataset, and the relevant parameters. Such a comprehensive representation should contain all datasets, and visualization and interaction tools. To support **T1**, a set of visual representations are proposed (Fig. 1 (c)) for itemsets, measures, constraints, rules in the mining process. For **T2** and **T3**, interactive exploration tools (Fig. 1 (d)) are designed to allow for intuitive data exploration, pattern extraction and insight generation on the basis of the visual representation. In this way, not only the input and output of the underlying mining process can be visually depicted and enhanced, but also a deep understanding of automatic rule mining process can be visually disclosed. The following three sections elaborate three main components of the proposed pipeline: representation (Section 5), visualization (Section 6) and exploration (Section 7) of association rule mining.

5. User-Driven Representation of Association Rules

We use a 5-tuple to describe our association rules mining process: \( \langle M, C, A, L, R \rangle \), in which \( M \) is a set of interest measures, \( C \) is a set of constraints, \( A \) is the association rules mining algorithm, \( L \) is a list of interesting itemsets, \( R \) is the set of strong association rules.

5.1. Interestingness Measures

Since interestingness varies from different applications, the user is allowed to define new measures for various tasks (T3). New measures can be used to either extend or replace the conventional support-confidence framework.

**Measures of Frequent Association Rules** Traditionally, the support-confidence framework is used to find frequent co-occurrence of different itemsets. If support \( s(X \Rightarrow Y) = P(X \cup Y) \) and confidence \( c(X \Rightarrow Y) = P(Y|X) \) are larger than given thresholds, then \( X \Rightarrow Y \). Here \( P(X \cup Y) \) means the probability of co-occurrence of \( X \) and \( Y \). The user can use alternative measures besides support and confidence to evaluate the correlation of \( X \) and \( Y \). For example, \( \chi^2 \) measure can be used to find the dependence of \( X \) and \( Y \).

**Measures of Rare Association Rules** Relatively infrequent associations of itemsets are also likely to be of great interest as they might indicate crucial information (e.g. identifying rare symptoms of a kind of rare disease). Support and confidence can be used to measure whether a rule is rare: if support \( s(X \Rightarrow Y) < \text{max}_{sup} \) and confidence \( c(X \Rightarrow Y) > \text{min}_{con} \), then \( X \Rightarrow Y \) is rare.

**Measures of Negative Association Rules** The user may be interested in negative associations between itemsets [26], for example, customers who buy Coke are unlikely to purchase Pepsi. Different measures are used here for mining negative patterns, e.g., negative \( n(X \Rightarrow Y) = (P(XY) + P(YX))/2 \). \( X \Rightarrow Y \) is a negative association rule if \( P(X) > \text{min}_n, P(Y) > \text{min}_n, \) and \( n(X \Rightarrow Y) < \text{max}_{neg} \). Other negative measures can be self-defined (e.g., the measure defined in Section 8.3).

**Self-defined Measures** To deal with various patterns in real-world applications, more customized measures can be defined. In marketing analysis, sales managers may focus on the commodities with specific customer groups. For example, there are some kinds of the commodities which are popular in female customers than in male customers. Measures for gender difference are defined for the analysis of those commodities: \( g(X \Rightarrow Y) = |P(X \cup Y)|^\langle\text{"Gender"} = \langle\text{"male"}\rangle - P(X \cup Y)|^\langle\text{"Gender"} = \langle\text{"female"}\rangle \rangle \). This measure ranges from 0 to 1. It can be used to evaluate an item’s popularity difference between different genders.
5.2. Constraints

The user might want like to set various constraints to filter the rules based on the domain knowledge (T2). This can reduce the search space and specify the interesting data. Following constraints are supported by our approach:

Itemset and Rule Constraint The user can specify the interestingness of itemsets during the mining process. An itemsets \( I \) can be deleted from the interesting list \( L \) if it is undesirable, and can be added into \( L \) if they are identified as interesting by the user. For example, the user focuses on the high-priced commodities when mining frequent association rules. Even if a commodity with high price (e.g., 5000) is not frequent, he or she can still add it into \( L \). The rules can be identified as interesting/uninteresting as well.

Data Constraint The user can specify the values of certain attributes to narrow the interesting subset of the data. By setting a set values \( v_i \) to a set of attributes \( A_C \), the user can remove the data with uninteresting values. Computation of association rules’ measures will be performed on the selected subset.

In market basket analysis, sales managers focus on customers’ age and gender as well. These attributes help them to understand the associations between commodities. Let \( A_C = \{ A_1 \equiv \text{“Gender”}, A_2 \equiv \text{“Age”} \} \). When exploring the association between \( X = \{ \text{“Video Game”} \}, Y = \{ \text{“Online Game”} \} \), they can explore their target customers by setting “Gender” = “male”, and “Age” \( \leq 30 \). The support measure of \( X \Rightarrow Y \) is calculated based on the selected subset.

Measure Constraint The user is allowed to change the measures during the mining process. The user can specify the threshold of measure computation under different scenarios. An adjustable threshold can help the user to filter out uninteresting rules. For example, if commodities are rarely purchased in a dataset, the user can set a very low minimum support threshold \( \text{min sup} = 0.001 \).

Other Constraints More constraints can be proposed by the user to confine the search space and find interesting rules. For example, when dealing with large datasets, the user is allowed to limit the number of the rules (e.g. finding the top-k most frequent rules).

5.3. Association Rule Mining Algorithm

The process of mining association rules is usually divided into two parts: 1) Find all interesting itemsets. 2) Generate strong association rules from the interesting itemsets. The pipeline is illustrated in the black arrows in Fig. 1.

The first step is much more complex and less straightforward than the second step [1]. Therefore, the overall performance of rule mining is determined by the interesting itemset finding.

5.3.1. Interesting Itemset Finding

Most existing methods find interesting itemsets iteratively in a bottom-up manner. Here we use Apriori [9] algorithm. The reasons for using it are two folds. First, it can be extended to mining various patterns, such as rare association rules [27] and negative association rules [28] mentioned in Section 5.1; Second, it is easy to understand. Apriori iteratively finds interesting itemsets.
• **Initialize** All the frequent items are added into the interesting itemsets list \( L_1 \).

• **Construct** The candidate \( k \)-itemset list \( C_k \) is generated from \( L_{k-1} \). A \( k \)-itemset \( I \) is in the candidate itemset \( C_k \) only if all of \((k - 1)\)-subset of it are in \( L_{k-1} \).

• **Filter** A candidate \( k \)-itemset \( I (I \in C_k) \) is added into \( L_k \) only when its frequency is higher than the threshold.

The construct and filter steps are repeated until no further interesting itemsets are found.

Note that we do not allow the user to modify the itemset finding algorithm itself. However, the user can interactively adjust the results in each intermediate step (Section 7). Therefore, our approach works well with an arbitrary itemset finding algorithm in a bottom-up manner.

5.3.2. Rule Generating

Suppose an association rule set \( \mathcal{R} \) is deduced from the interesting itemsets list \( L_k \) found in the \( k \)th step. For a non-empty subset \( X \) of an interesting itemset \( I \) in \( L_k \), let \( Y = I - X \). If all the measure values of rule \( X \Rightarrow Y \) is in the given range, this rule is strong. For example, for finding frequent association rules, \( X \Rightarrow Y \) is a strong rule and is added into \( \mathcal{R} \) if confidence \( c(X \Rightarrow Y) > \min_{\text{conf}} \) and support \( s(X \Rightarrow Y) > \min_{\text{supp}} \).

The rule set \( \mathcal{R} \) generated in the \( k \)th step is visualized using matrix-based visualization (Section 6.1).

6. Visual Representation

The interesting itemset list \( L_k \), the candidate itemset list \( C_k \) that are generated in the \( k \)th step of algorithm as well as the categorical attribute set are all presented with a list view (Fig. 2).

During the association rules mining, understanding of the measure values and the association patterns are necessary (T1). In short, we use the rule matrix to visualize the association rules \( \mathcal{R} \) generated in the \( k \)th intermediate step of the rule mining, and use the distribution matrices to disclose the data patterns and association rules.

6.1. Rule Matrix: Visualizing Measure Computation Values and Association Rules

Each rule \( r \) in \( \mathcal{R} \) has a Left Hand Side (LHS) itemset \( X_r \) and Right Hand Side (RHS) itemset \( Y_r \), where \( X_r \) and \( Y_r \) are disjoint itemsets. A graph structure of \( \mathcal{R} \) can be constructed in which the LHS and RHS itemsets are the nodes and a rule is a link connecting two nodes of its LHS and RHS itemsets.

The graph-based and matrix-based visualization were among the most widely used methods for visualizing this graph structure [23] [6]. We argue that the node-link graph visualization is not suitable for our problem:

• Measure computation value cannot be visualized effectively in the graph. Most existing approaches use an attribute (e.g., the color) of the links to encode the measure values of the rules. However, these visual encodings of the links are not easily perceived.

• Visual clutter may be produced when a graph contains a very large sets of rules.

6.1.1. Matrix Layout

Let \( X \) be the set of all the LHS itemsets of a rule \( r \) in \( \mathcal{R} \), and \( Y \) be the set of all RHS itemsets of a rule \( r \) in \( \mathcal{R} \). \( X \) and \( Y \) may have common itemsets because an itemset could occur on the left hand side of a rule and on the same time occur in the right hand side of another rule. Assuming \( X \) has \( m \) itemsets and \( Y \) has \( n \) itemsets, a \( m \times n \) matrix is created with each row for a LHS itemset \( X_i (1 < i < m) \) and each column for a RHS itemset \( Y_j (1 < j < n) \). Each cell \( c_{i,j} \) stands for the rule of \( X_i \Rightarrow Y_j \).

The matrix is updated after each step in the mining process, which provides the following two benefits:

• Shows only the most updated rules generated because old rules have been already checked before.

• Reduced amounts of rules and itemsets lead to a relatively small matrix, which can be visualized and understood effectively because of the limited working memory of the human.

6.1.2. Color Encoding

Making sense of the measure computation in each intermediate step is critical for evaluating the mining process. In our approach, the color of a cell \( c_{i,j} \) encodes the value of a selected measure of \( X_i \Rightarrow Y_j \), e.g., the support measure (Fig. 3). A pre-computed color scale [29] or user-specified color scales can be used for the color mapping. If the measure is zero, the color of the cell \( c_{i,j} \) is set to grey. For a cell \( c_{i,j} \) in the matrix, if \( X_i \cap Y_j \neq \emptyset \), its color is set to grey because \( X_i \Rightarrow Y_j \) has no meaning in practical applications (e.g., (“Skin Care”, “Make Up”) ⇒ (“MakeUp”)).

A Cell \( c_{i,j} \) is highlighted by a purple border if the corresponding rule \( X_i \Rightarrow Y_j \) is strong. The user can filter or add rules in the matrix by using interaction tools illustrated in Section 7.1.
6.1.3. Hierarchical Visualization

In the online transaction dataset, the size of the rule matrix is small since the number of itemsets $|X|$ and $|Y|$ are limited. However, for other datasets, the generated rule matrix may be huge to be displayed in the screen space in other cases. For example, the MovieLens dataset contains more than 4000 movies (Section 8.3). The generated rule matrix is too big for the user to explore. One solution to deal with this scalability problem is constructing a hierarchical structure of the itemsets and presenting the rules in different granularities.

Different options are provided for constructing the hierarchical structure. A straightforward method will be aggregating the itemsets within the same category, which will result in a 2-level structure. As shown in Fig 4 (a), the movies are aggregated by their major categories, which are given by the labels in the dataset. Automatic methods can also be adopted, such as hierarchical clustering. With the hierarchy constructed, the rule matrix can be used to visualize the itemsets in different levels. For example, a cell in Fig 4 (a) represents a rule of movie genre. The measure value of each aggregated cell is assigned to the average of the values in it.

For the aggregated rows or columns of interest, the user is able to expand them to examine the detailed itemsets in a lower level. Fig 4 (b) shows the exploration in the 2-level structure of the MovieLens dataset. According to a user’s suggestion in the case study, we set the widths or lengths of the non-aggregated cells smaller, which make them distinguishable to the aggregated ones.

6.2. Distribution Matrices: Visualizing Itemset Distribution on Constraint Attributes

The context information helps the user to understand the rules. For example, the handbag and the dress usually have strong associations in woman customer group. The distributions of these commodities in customers’ age and gender help the sales manager to understand these associations better. The distribution matrix is designed to visualize the distribution of an itemset on a set of attributes $A_C$ (Section 5.2), which follows the dimensional stacking approach [30].

For each LHS (or RHS) itemsets $I$, we create a distribution matrix of $I$, which is placed at the right side (or bottom side) of a row (or column) of the rule matrix (Fig. 2 (b)). We use a line to connect the row (or column) of $I$ and the corresponding distribution matrix. In the following, we describe the generation methods of each distribution matrix.

6.2.1. Matrix Division

The initial distribution matrix of $I$ contains only one cell that represents all the data. We use a set of user-specified attributes $A_C = \{A_{c_1}, A_{c_2}, ..., A_{c_m}\}$ to split the matrix. The distribution matrix is generated using the Algorithm 1.

Algorithm 1 Distribution Matrix Splitting

Input: the categorical attribute set $A_C = \{A_{c_1}, A_{c_2}, ..., A_{c_m}\}$, each attribute $A_c(i \leq \ell \leq m)$ has $n_c$ values; the initial cell list $C = \{c_{1,1}\}$ in the matrix; the row number $n_{c_1} = 1$ and the column number of the matrix $n_{c_2} = 1$.

1: for Each $A_c$ in $A_C$ do
2:  for Each Cell $c_{i,j}$ in $C$ do
3:   if Dividing vertically then
4:     Cell $c_{i,j}$ is split vertically into $n_c$ sub-cells;
5:     $n_{c_1} = n_{c_1} \times n_c$;
6:   else
7:     Cell $c_{i,j}$ is split horizontally into $n_c$ sub-cells;
8:     $n_{c_2} = n_{c_2} \times n_c$;
9:   end if
10: update the index of each cell in the matrix;
11: end for
12: end for

Fig. 5 (a) shows a distribution matrix which is split by a set of attribute $A_C = \{\text{“Gender”}, \text{“Age”}, \text{“Constellation”}\}$. Each cell in the matrix stands for a specific combination of attribute values in $A_C$. For example, a cell stands for a customer group whose “Age group” = “15 - 20”, “Gender” = “Male”, and “Constellation” = “Taurus”.

6.2.2. Color Encoding

The distribution matrix can be regarded as the visualization of a contingency table (Fig. 5 (c)) of the attribute set $A_C$. The color of each cell represents the count of itemset in each cell (e.g., the count of transaction records containing $I = \{ \text{“Skin Care”}, \text{“Make Up”}\}$ in a certain customer group whose “Age group” = “15 - 20”, “Gender” = “Male”, and “Constellation” = “Taurus”). The user can specify the color scheme in the distribution matrix. Fig. 5 (b) shows a distribution matrix of the itemset $I$, whose color encodes the count in each cell. A grey and white label on the left (or top) side of the distribution matrix is used to indicate the attribute combination of rows (or columns).
The distribution matrix shows the feature of distribution of $I$ on a set of attribute $A_C$. It allows the user to set the $A_C$ interactively (7.3) and set the data constraints to confine the search space.

7. Interactive Exploration

In our approach, the user can explore the hierarchical matrix in different levels. A set of user interactions is also designed for modulating different kind of constraints (T2).

### 7.1. Itemset Constraint Modulation

Even though the itemset finding algorithm (Section 5.3.1) can be used to find most of the interesting itemsets, it is necessary for the user to modify the interesting itemset list $L_k$ manually. Supported operations include adding, deleting and searching the itemsets.

**Adding and Deleting Itemsets** If the user is interested in an itemset $I$ in $C_1$, he or she can add it into $L_k$ in the itemsets view. The user can define a specific threshold for the rules which are generated by $I$. Let $r$ be a strong rule generated by $I$, it will be added into $R_k$ and visualized in the rule matrix visualization.

The user is allowed to remove uninterested itemset $I$ from $L_k$ in the itemsets view. Then all the rules generated by $I$ are removed from $R_k$. $I$ will be no longer involved in the following process and there will be no superset of $I$ generated in the subsequent steps.

**Searching Itemsets** There might be a large amount of itemsets in the interesting itemset list $L_k$. The user can search the itemsets using a search box in the itemsets view. Itemsets which contain the searched attribute will be highlighted. This interaction is helpful when the user wants to select all the itemsets that contain certain attributes. For example, before deleting all the itemsets that contain the attribute “Books”, the user can select all these itemsets by searching “Books”. When an interesting itemset $I$ is selected, the rules generated by $I$ in the rule matrix will be highlighted.

### 7.2. Rule Constraint Modulation

The user can check the measure values of the rules by clicking the corresponding cells. The rules generated in the intermediate steps of the mining process can be modulated using the following interactions:

- **Comparing and Filtering Rules** The columns and rows can be rearranged manually. This interaction is provided for the comparison of the cells in different rows (or columns) in the rule matrix. For the comparison of two rows, the user can place them at adjacent positions by dragging.

- **Combining Rules** The user can combine two different RHS itemsets by dragging one column into another as (Fig. 6). A new row of $X_i \cup X_j$ will be added into the matrix, and the measure value of $X_i \cup X_j \Rightarrow Y_k$ is computed. Although an interesting rule $X_i \cup X_j \Rightarrow Y_k$ can be generated in the subsequent steps of the rule mining process, the user can get and export this rule in advance. This interaction might save the time for the user to get interesting rules. The user is allowed to combine different RHS itemsets by dragging one column into another as well.

![Figure 5: (a) A matrix is split by a set of attributes: $A_C = \{ \text{"Age Group"}, \text{"Gender"}, \text{"Constellation"} \}$. Each cell stands for a customer group, which is grouped by the value combination of "Age Group", "Gender" and "Constellation". (b) A distribution matrix of $I = \{ \text{"Skin Care"}, \text{"Make Up"}, \text{"Color"} \}$ is showed in the distribution matrix visualization. (c) The corresponding contingency table of the distribution matrix in (b).](image)

The user can set a rule $X_i \Rightarrow Y_j$ as (un)interesting by (un)highlighting the cell $c_{i,j}$. The updated rule matrix is then saved at each interaction step of the rule modulation.

### 7.3. Data Constraint Modulation

For the data constraint in Section 5.2, the user is allowed to select the attribute set $A_C$, which is visualized in the distribution matrices. Constraints of $A_C$ can be set to narrow the search of interesting association rules.

- **Adding and Deleting Attributes** In $A_C$, the user is allowed to add/remove the $A_i$ using the attribute selection panel (Fig. 2 (e)). After $A_i$ is added/removed, the distribution matrices will be updated.

### Setting Attribute Constraints

The user can set constraints on different attributes. For example, when selecting the attribute “Age”, the user can set a specific range (e.g., 18-25 years). This constraint will narrow the search space for interesting rules.

![Figure 6: (a) One row $X_3$ is dragged into another row $X_1$ in the rule matrix. (b) A new row, which stands for a new RHS itemset $X_3 \cup X_5$, is added in the rule matrix.](image)
Each cell in a distribution matrix has a specific combination of the attribute-values in $A_C$. By selecting a group of cells in the distribution matrix, the interesting subset of data can be specified. Measures and rules will be recomputed based on the selected subset. For example, by selecting the left half of the distribution matrix, the user chooses all female customers. The support and confidence measures are computed based on the data of selected customers.

The user is allowed to use the following three interactions to select the interesting cells in a distribution matrix:

- **Drag** A group of cells will be selected by dragging a box or any shape in a distribution matrix (Fig. 2 (f)).
- **Eraser** The user can remove cells from the selected cell set.
- **Reset** The user can reset the selection. All cells will be regarded as selected under this state.

The selected cells are highlighted. The rule matrix is updated to show the measure values which are recalculated based on the selected subsets.

### 7.4. Measure Constraint Modulation

The user is allowed to add/delete a measure in the measure selection panel (Fig. 2 (d)). He or she can adjust the threshold of a certain measure. Then the rule matrix is accordingly updated to show the new measure values. For example, when the user sets the support threshold to a higher value, fewer cells in the matrix will be highlighted, indicating less strong rules. The modulation of the thresholds of the measures during the mining process provides a real-time feedback in the rule matrix. It helps the user understand the modulation result and to find the proper threshold quickly.

### 7.5. History Navigation

A history view (Fig. 2 (g)) is provided to support the undo operation. The user can go back to the previous analysis state by clicking the desired history state in the view. After modulation of the constraints, the previous state of the mining process is saved (e.g., the interesting itemsets $I_k$ and the association rules $R_k$ of the $k$th step of mining algorithm). The saved itemsets and rules of the $k$th step are restored in the visualization view if the user wants to go back to the $k$th step.

### 8. Case Study

Two case studies are presented in this section to show the features and capacity of our approach.

#### 8.1. Participants and Procedure

Two people who wanted to use association rules for customized recommendation participated our study. The first participant was a sales analyst from the data department of an online retail company. He provided an online transaction dataset (Section 3) in this case. Since there is a sales promotion in his company of “Milk” commodities, he was interested in finding associated commodities for recommendation to his customers.

The second participant was male graduate student who is a science fiction (sci-fi) movie fan. He would like to use the MovieLens dataset [25] to find movies which he might be interested. Both participants had little experience about visual analysis.

Each case study started with a demo to briefly introduce our approach. We then asked each participant to use the system, followed by an interview to gather evaluations and subjective feedback.

#### 8.2. Online Transaction Dataset

The first participant started to use our system after a brief introduction of our approach. He selected the support and confidence measures for the mining process. He set their minimum thresholds to 0.05 and 0.65 respectively after using different values (Fig. 2 (d)). The interesting 1-itemset list $I_1$ and 2-itemset list $I_2$ were found using the automatic algorithm (Section 5.3.1). The strong rule set $R_2$ was generated from $I_2$ and visualized in the rule matrix.

The analyst searched and selected the keyword “Milk” in the itemset view. All the strong rules generated by the itemsets which contain “Milk” were highlighted in the rule matrix. The analyst found some strong rules such as “{“Milk”} ⇒ {“Toys”}” and “{“Milk”} ⇒ {“Baby Products”}” (Fig. 7 (a)). According to his work experience, he thought that these rules were made by people who had children. The customers often buy milk products for their children. However, he did not find the reason for the rule $r_1 = \{ \text{“Milk”} \} \Rightarrow \{ \text{“Digital Accessories”} \}$. By checking this rule, he found the support and confidence of this rule were 0.12 and 0.653 separately. After examining the entire matrix, he learned that “Digital Accessories” occurs in the RHS itemsets of many other rules. He concluded that the rule $r_1$ was recognized by the automatic algorithm as strong because the “Digital Accessories” had a large sales amount. He was not interested in the rules and thought the result of the mining process could be misleading.

He decided to use additional information to find the target customer of the “Milk” commodities. He added “Age”, “Gender”, “Constellation” to generate the layout of the distribution matrices. He checked the distribution matrix of “Milk”. The density of the region in the left of the distribution matrix was high. This region represented the customer groups whose “Gender” was “Female” and the “Age” was between 25 and 40. He thought that one group of the most important target customer of “Milk” are woman who have children. To explore the association rules of these target customers, he set the attribute constraint by dragging the corresponding region in the distribution matrix using a box tool (Fig. 7 (b)). The measure values were recomputed based on the selected subset of data and the rule matrix was updated immediately.

In the rule matrix (Fig. 7 (b)), the analyst found that the cell of rule $r_1$ was no more highlighted, indicating the rule was removed from the strong rule set $R_2$. The support confidence of $r_1$ were updated to 0.15 and 0.641. In addition, he found some rules turned to be highlighted after setting the attribute constraint, such as “[“Milk”] ∪ (“Underwear”), (“Milk” ] ∪ (“Skin Care”). The analyst thought that these rules showed an interesting transaction pattern of the selected target customer
The analyst continued his exploration. The interesting 3-itemset list $I_2$ and strong rule set $R_3$ were generated (Fig. 7 (c)). He found that there was still a rule that contains “Milk” and “Digital Accessories”: $r_2 = \{ \text{"Underwear"}, \text{"Milk"} \} \Rightarrow \{ \text{"Digital Accessories"} \}$. He was surprised about the fact that $r_2$ can be strong under the attribute constraint even though $r_1$ was not. By checking the cell of $r_2$, he found that confidence $c(r_2) = 0.651$, and $c(r_2) > \text{min}_c > c(r_1)$.

The analyst thought the rules which contain “Digital Accessories” were not interesting. He went back to step 2 using the history view and removed all the itemsets in $I_2$ which contain “Digital Accessories” in the interesting itemset view. When he generated $I_3$ again from $I_2$, there had no more itemsets containing “Digital Accessories”. The modulated result of $R_3$ was shown in Fig. 2 (a). Finally, the analyst saved and exported all interesting rules.

8.3. MovieLens Dataset

The second participant started with finding the frequent associations between the sci-fi movies. After trying different measure thresholds, he set the thresholds of support and the confidence measures to 0.03 and 0.65, respectively. The hierarchical rule matrix of movie genres was generated by aggregating the movies in the same genre. After the expanding the Sci-Fi genres and exploring the movies, he told us that people who give some scientific fiction movies (e.g., “Superman”, “StarTrek”) a high rating (4 - 5 stars) will also likely rate “StarWarsIV(77)” as 5 stars (Figure 4).

The user also found some interesting negative associations between the sci-fi movies. Based on the confidence measure, he proposed a new measure for the negative association rule: $n(X_i \Rightarrow Y_j) = 1 - c(Y|X)$. He defined that a rule $X_i \Rightarrow Y_j$ is strong if $s(X_i) > \text{min}_s$, $s(Y_j) > \text{min}_s$, and $n(X_i \Rightarrow Y_j) > \text{min}_n$. The minimum support threshold and minimum negative threshold were set to 0.1 and 0.9 respectively. The interesting itemsets were found by an Apriori-like algorithm [28]. Thereafter, strong rule set $R_3$ was generated after the interesting 1-itemset list $I_1$ and 2-itemset list $I_2$ were found.

By focusing on the rows and columns of the Sci-Fi movies(Figure 8) in the rule matrix, the user found a negative rule $r_1 = \{ \text{"StarWarsV(80)"} = \text{"5"} \} \Rightarrow \{ \text{"StarWarsIV(77)"} = \text{"4"} \} (m(r_1) = 0.991)$. He checked the rule $r_2 = \{ \text{"StarWarsV(80)"} = \text{"5"} \} \Rightarrow \{ \text{"StarWarsIV(77)"} = \text{"5"} \}$. $r_2$ had a very low negative value $n(r_2) = 0.254$. Thus the confidence value of $r_2$ is $1 - n(r_2) = 0.746$, which is very high in the MovieLens dataset. He concluded that $r_1$ was strong mainly because most people who gave 5 stars to “StarWarsV(80)” would also rate “StarWarsIV(77)” as 5 stars, instead of 4 stars. Although $r_1$ is strong, it could not be explained properly without context information of $r_2$. The user told us that $r_1$ might not be effective to describe the viewers’ behavior. He set $r_1$ as uninteresting.

8.4. User Feedback

During the post-study interview, we asked the users to give subjective feedback about how the system supported the three tasks.

The analyst from the retail company mentioned the matrix-based visualization was intuitive and helped him with T1. Especially, he said that “The distribution matrix visualization and the
Figure 8: The rule matrix of the MovieLens dataset. The color encodes the negative measure value. The user found a negative association rule: 
\{“StarWarsV(80)=5”\} ⇒ \{“StarWarsIV(77)=4”\},
\text{negative}(r_1) = 0.991 
\{“StarWarsV(80)=5”\} ⇒ \{“StarWarsIV(77)=5”\},
\text{negative}(r_2) = 0.254

interaction for attribute constraint are really helpful”. He also told us that “The function of filtering itemsets and attributes enabled me to focus on interesting commodities.” For T2 and T3, he mentioned that in his daily analysis, it was always hard for him to target interesting rules, and explain them. He found that the interactions for setting the constraints were effective for finding interesting associations of commodities. He showed it is time-consuming to understand and check the results of the existing automatic association rule mining methods. Our methods allowed him to set the constraints interactively, which was also helpful for him to explain the patterns of the associations in the datasets directly.

The user participated in the MovieLens case stated that he got useful insights into the rules (T1), which helped him learned the viewers’ preferences of the Sci-Fi movies. He noted that our visual interactions also supported T2 and T3 very well. He appreciated the flexibility of our system for mining various association patterns, such as negative association. He thought that the visualization allowed him to understand the reasons for strong rules. He also suggested that we could constrain the sequence of the movies in the rule, because viewers usually watch movies of an episode sequentially.

8.5. Discussion

Comparing with the automatic methods, the visual analysis of rule mining requires the efforts for training, sense making and interactions. When asked about the training cost, both users showed that our system did not take them much time to understand our approach (both of their training sessions were less than 8 minutes). For the exploration, they mentioned that the interaction with the dataset is necessary for finding interesting measures and constrains, especially for the the first time. Later they can export the customized measures and constraints and use it for an automatic Apriori algorithm.

During the study, we noticed that the users did not know which value were good threshold for finding interesting rules at the beginning of their studies. Both of them mentioned they spent some time finding the measure thresholds by trying different values. The user in the first case noted he aimed to filter roughly about 80% rules related to a target commodity (e.g., “Milk”). The second user mentioned since the threshold value varied among different datasets, the proposed visualization was convenient for him to see the effects of different thresholds.

The results of our system are reliable because they are based on the well-studied mining algorithms. Our system is able to be extended to other applications since the association rules are widely used in categorical dataset. More evaluations in different scenarios can be done in the future to verify the general usability of our approach.

When the data amount is huge, our approach may take a lot of time to run the Apriori algorithm. In this situation, if the measures and thresholds are fixed, we can pre-compute the potential interesting rules. Based on these rules, the user can still set different constraints to get the preferred rules.

9. Conclusion

In this work, we present a visual analysis method for mining association rules in categorical datasets. Our approach incorporates user-defined interesting measures and constraints for different applications, and provides interaction capabilities for modulating the constraints during the iterative mining process. Finally, two different case studies demonstrate the validity of our approach.

For future work, we expect to support more method for constructing the hierarchical structure of the rule matrix. An automatic approach for recommending the measure thresholds will also help the user during the visual exploration. We also would like to focus on temporal structures of the rules (e.g., the sequences of the commodity purchases) by integrating time series analysis [31].

10. Acknowledgments

Supported by Major Program of National Natural Science Foundation of China (61232012), National Natural Science Foundation of China (61772456, 61422211).

References


