ABSTRACT

Association rules have been widely used for detecting relationships 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 propose to 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 matrix-based visualization technique is employed to encode the measure computation value, the data distribution and the intermediate results. We also design a set of visual exploration tools to support interactively inspection and manipulation of association measures, constraints of different types, and the results of intermediate steps. The effectiveness of our approach is demonstrated with various scenarios.

Keywords
Categorical Data, Association Rules, Visual Analysis

1. INTRODUCTION

1.1 Background

The association rules mining [1] is one of the most commonly used method to find relationships between different attribute-value pairs in different datasets. For example, there is an online transaction dataset which contains about 10,000 customers of a company. Each data instance records a set of commodities in a customer’s basket. The association rule mining helps the manager to find co-occurrence relationships among different purchased items (e.g., if a customer buys bread, he is likely to purchase butter). Interesting rules are critical for the sales managers to understand the behavior patterns of the customers.

However, strong rules found by the automatic rule mining methods (e.g., support-confidence framework) are not necessarily interesting [10]. The judgement of whether a given rule is interesting may vary from 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 in it 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 headphones after buying a digital audio player. Association rules with a desirable sequence are interesting in such case.

- 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 [8], or narrow the interesting rules by setting various constraints for certain attributes, itemsets, and rules [25] (e.g., specifying the interesting subset in database). Before defining measures and constraints, the process and results of association rule mining need to be understood. However, the results are actually hard to understand [13] using existing automatic mining methods. Meanwhile, most of these works do not provide methods for modulating the mining process interactively.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

VINCI '16 September 24-26, 2016, Dallas, Texas, USA
DOI: 10.1145/2992254.2992256
Existing visualization methods can be employed to either the final rules [9], or the intermediate results [28] of rule mining. However, only basic interactions are supported, 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.

The scheme proposed in this paper is a user-driven visual analysis pipeline for iteratively discovering association rules. We identify the main contribution of this paper as a visual-assisted rule mining pipeline that reinforces the iterative mining loop and makes it intuitive and effective for the user to understand, evaluate and modulate the process of mining interesting rules. 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 design a set of interactions that supports intuitive inspection and manipulation of the measures, constraints, and the rule results of the intermediate steps.

## 2. RELATED WORKS

### 2.1 Automatic Analysis of Association Rules

Association Rule [1] was proposed to identify relationships between items in very large databases. It used support-confidence framework to find frequent itemsets in the market basket analysis.

**Interestiness Measures** Typically, different patterns are 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. [8] 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. [25] integrated boolean constraints on items into the association rules. Ng et al. [20] characterized various constraints (contains, minimum, maximum, count, sum) according to anti-monotonicity and succinctness.

**Frequent Itemsets Computation** Most algorithms find frequent itemsets iteratively in a bottom-up manner (e.g., Apriori [2] and hash-based method [22]). However, most of the existing works do not allow the user to modulate the itemsets found in the intermediate steps. Our approach advances these iterative algorithms by integrating the visual analysis with automatic frequent itemset computation to find the interesting itemsets.

### 2.2 Visualization of Association Rules

Existing visualization methods mainly visualize the mining results subject to different tasks [26], [19]. ARVis [4] was proposed for rummaging and validating association rules. Sekhavat et al. [24] 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 [12].

Various visualization methods can be used for exploring and explaining the association rules, such as mosaic plot [13], scatter plot [18], node-link graph [17], and matrix [21]. Other works can be found in [9]. Most of these works focus on the explanation of the final association rule result. Zhao et al. [28] proposed to show the result of intermediate steps. In either case, the user is neither able to propose self-defined measures for various tasks, nor to modify the constraints of different types during the rule mining algorithm.

### 2.3 Visual Analysis of Categorical Data


## 3. VISUAL ASSOCIATION RULE MINING

Before introducing our approach, we define the problem 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 \( A_j \) has a domain \( D_j \). Mining an interesting association rule is finding the implication of the form \( X \Rightarrow Y \) in which \( X \) and \( Y \) are conjunction of attribute-value pairs (items):

\[
X = \{A_{i_1} = v_{i_1}, A_{i_2} = v_{i_2}, \ldots, A_{i_k} = v_{i_k}\} \quad \text{and} \quad Y = \{A_{j_1} = v_{j_1}, A_{j_2} = v_{j_2}, \ldots, A_{j_l} = v_{j_l}\}, \quad 1 \leq k, l \leq n, \quad X \cap Y = \emptyset.
\]

\( X \) is the Left Hand Side (LHS) itemset of a rule and \( Y \) is the Right Hand Side (RHS) itemset. For example, in the rule \{“Milk” = “yes”, “Bread” = “yes”\} \( \Rightarrow \{“Butter” = “yes”\} \), \{“Milk” = “yes”, “Bread” = “yes”\} is the LHS itemset and \{“Butter” = “yes”\} is the RHS itemset.

**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.

Existing methods on association rule mining commonly employ an iterative pipeline to discover strong association rules (See black arrows in Fig. 1). The rule mining process 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 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 further make the itemsets, measures, constraints, rules and the mining process visible to the user, a set of visual representations are proposed (Fig. 1 (c)). Meanwhile, 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 4), visualization (Section 5) and exploration (Section 6) of association rule mining.

4. USER-DRIVEN REPRESENTATION OF ASSOCIATION RULES

We use a 5-tuple to describe our association rules mining process: \((M, C, A, L, R)\), 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.

4.1 Interestingness Measures

The user can define new measures for various tasks before the mining process. 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 \(c(X \Rightarrow Y) = P(X|Y)\) 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 \(Y\). More measures for frequent patterns have been discussed in Section 2.1.

**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 \(s(X \Rightarrow Y) < \maxsup\) and confidence \(c(X \Rightarrow Y) > \minconf\), then \(X \Rightarrow Y\) is rare.

**Measures of Negative Association Rules** The user may be interested in negative associations between itemsets [23], for example, customers who buy Coke are unlikely to purchase Pepsi. Different measures are used here for mining negative patterns, e.g., negative \(n(X \Rightarrow Y) = (P(X|Y) + P(Y|X))/2\). \(X \Rightarrow Y\) is a negative association rule if \(P(X) > \minp, P(Y) > \minp\) and \(n(X \Rightarrow Y) < \maxneg\).

**Self-defined Measures** To deal with more complex patterns in the datasets, the user can define various measures by themselves. For example, in the transaction dataset, each customer has multiple attributes, including age group, gender, constellation, and account registration date. Measures for gender difference. In market basket analysis, sales managers need to find the feature of the population distributions of different commodities. They may focus on the commodities which have more popularity in female customers than in male customers. Therefore, they can define a new measure called Gender Difference Measure: \(g(X \Rightarrow Y) = |P(X \cup Y(Gender = \text{“male”}) - P(X \cup Y(Gender = \text{“female”}))|\), whose value ranges from 0 to 1. These measure can be used to evaluate the items’ popularity difference between different genders.

4.2 Constraints

When mining the association rules, the user might have some intuition with the datasets and would like to set various constraints. 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 whose price is higher than 5000 is not frequent, he or she can 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 \(v_c\) 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 \(A_c = \{A_1 = \text{“Gender”}, A_2 = \text{“Age”}\}\). When exploring the association between \(X = \{\text{“Video Game” = “yes”}, Y = \{\text{“Online Game” = “yes”}\}\), 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.

**Measures 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 \(\minsup \geq 0.001\).

**Other Constraints** More constraints can be proposed by the user to consume 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).

4.3 Association Rule Mining Algorithm

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

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

4.3.1 Interesting Itemset Finding

Most existing methods find interesting itemsets iteratively in a bottom-up manner. Here we use Apriori [2] algorithm. The reasons for using it are two folds. First, it can be extended to mining various patterns, such as rare association rules [15] and negative association rules [27] mentioned in Section 4.1; 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.
Right Hand Side (RHS) itemset

5.1 Rule Matrix: Visualizing Measure Computation Values and Association Rules

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.

5. VISUAL REPRESENTATION

The interesting itemsets list generated in every intermediate step are all presented with a list view (Fig. 2). The list generated in every intermediate step. (d) Measure modulation view. (e) Attribute selection panel of the distribution matrices. The user sets the attribute constraint by dragging the desired region in a distribution matrix. (g) The history view allows the user to go back to any previous analysis state.

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 6). Therefore, our approach works well with an arbitrary itemset finding algorithm in a bottom-up manner.

4.3.2 Rule Generating

Suppose an association rule set \( R_k \) 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 \( R_k \) if confidence \( c(X \Rightarrow Y) > \text{min}_\text{conf} \) and support \( s(X \Rightarrow Y) > \text{min}_\text{supp} \).

The rule set \( R_k \) generated in the \( k \)th step is visualized using matrix-based visualization (Section 5.1).

Figure 2: The main interface of our approach. (a) The rule matrix view visualizes the measure computation of the rules generated in intermediate steps of mining process. (b) The distribution matrix view provides context information of rules. (c) Interesting itemsets list generated in every intermediate step. (d) Measure modulation view. (e) Attribute selection panel of the distribution matrices. The user sets the \( A_C \) by adding or removing the attributes in the panel. (f) The user sets the attribute constraint by dragging the desired region in a distribution matrix. (g) The history view allows the user to go back to any previous analysis state.

Each rule \( r \) in \( R_k \) has a Left Hand Side (LHS) itemset \( X_i \) and Right Hand Side (RHS) itemset \( Y_j \), where \( X_i \) and \( Y_j \) are disjoint

items. A graph structure of \( R_k \) 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 [9]. 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 set.

5.1.1 Matrix Layout

Let \( X \) be the set of all the LHS itemsets of a rule \( r \) in \( R_k \), and \( Y \) be the set of all RHS itemsets of a rule \( r \) in \( R_k \). \( X \) 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 \( c_{ij} \) 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.
5.1.2 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 $c_{ij}$ 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 or user-specified color scales can be used for the color mapping. If the measure is zero, the color of the cell $c_{ij}$ is set to grey. For a cell $c_{ij}$ 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” = “yes”, “Make Up” = “yes” -> (“Make Up” = “yes”).

A cell $c_{ij}$ 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 6.1.

5.2 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 the a set of attributes $A_C$ (Section 4.2).

For each LHS (or RHS) itemset $I$, 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.

5.2.1 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 $A_C = \{A_1, A_2, \ldots, A_m\}$ to split the matrix. The distribution matrix is generated using the Algorithm 1.

Fig. 4 (a) shows a distribution matrix which is split by a set of attributes $A_C = \{”Age Group”, ”Gender”, ”Constellation”\}$. Each cell stands for a customer group whose “Age group” and “Gender” and “Constellation”. (b) A distribution matrix of $I = ”Skin Care” = ”yes”, ”Make Up” = ”yes”, the color of cells encodes the frequency of purchasing “Skin Care” and “Make Up”. The commodities of “Skin Care” and “Make Up” are mainly purchased by woman customers. (b) The corresponding contingency table of the distribution matrix in (c).

5.2.2 Color Encoding

The distribution matrix can be regarded as the visualization of a contingency table (Fig. 4 (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 = ”Skin Care” = ”yes”, ”Make Up” = ”yes”) 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. 4 (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 (6.3) and set the data constraints to confine the search space.

6. INTERACTIVE EXPLORATION

In our approach, a set of user interactions is designed to modulate the association rules mining process.

6.1 Itemset Constraint Modulation

Even though the itemset finding algorithm (Section 4.3.1) 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 \( C_k \), 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.

6.2 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 rows and columns 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.

The user can filter the rules by selecting the attributes in the attribute view. The rule matrix will only show the rules which are generated by the selected attributes.

Combining Rules The user can combine two different LHS itemsets \( X_i \) and \( X_j \) by dragging one row into another (Fig. 5). A new row of \( \{X_i \cup X_j\} \Rightarrow Y_k \) 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.

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

6.3 Data Constraint Modulation

For the data constraint in Section 4.2, 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.

Figure 5: (a) One row \( X_3 \) is dragged into another row \( X_3 \) in the rule matrix. (b) A new row, which stands for a new LHS itemset \( X_3 \cup X_3 \), is added in the rule matrix.

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

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.

6.4 Measure Constraint Modulation

The user is allowed to add or deleting 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 find the proper threshold quickly.

A history view (Fig. 2 (g)) is provided to support the undo operation. The user can go back to a previous analysis state by clicking the desired history state in the view.

7. CASE STUDY: ONLINE TRANSACTION DATASET

We implemented an integrated system in Java. A case study is presented in this section to show the features of our approach.

A sales analyst from the data department of an online retail company participated in this case. He had little experience about visual analysis. He was interested in finding commodities which have frequent association with “Milk” commodities. After a discussion with the user, we learned that he was interested in the following tasks: 1. Examining the strong rules found by the mining process. 2. Filtering the strong rules by his domain knowledge. 3. Identify-
Figure 6: The visual analysis of the online transaction dataset. (a) The rule matrix shows strong association rules of online transaction dataset, such as \{“Milk”\} \Rightarrow \{“Digital Accessories”\}. The cell color encodes the support value (b) By setting an attribute constraint, the rule \{“Milk”\} \Rightarrow \{“SkinCare”\} becomes strong while the rule \{“Milk”\} \Rightarrow \{“Digital Accessories”\} turns to be not strong. (c) The user still observes strong rules that contain “Milk” and “Digital Accessories” (e.g., \{“Underwear”, “Milk”\} \Rightarrow \{“Digital Accessories”\}) in the subsequent mining process of (b). The user could go back to (b) and delete the itemset of “Digital Accessories”.

After a brief introduction of our approach, he started to use our system. He selected the support and confidence measures for the mining process and set their minimum thresholds to 0.05 and 0.65 separately through the threshold panel (Fig. 2 (d)). The interesting 1-itemset list \(I_1\) and 2-itemset list \(I_2\) were found using the automatic algorithm (Section 4.3.1). 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”\} \Rightarrow \{“Toys”\} and \{“Milk”\} \Rightarrow \{“Baby Products”\} (Fig. 6 (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 \(r_1 = \{“Milk”\} \Rightarrow \{“Digital Accessories”\}\). 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 set \(A_C\) of the distribution matrices to \{“Age”, “Gender”, “Constellation”\} and checked the distribution matrix of “Milk”. The density of the region in the left of the distribution matrix was high. This region stooded for 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 women 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. 6 (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. 6 (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 and 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”\} \Rightarrow \{“Underwear”\}, \{“Milk”\} \Rightarrow \{“Skin Care”\}. The analyst thought that these rules showed an interesting transaction pattern of the selected target customer group. In this case, he might not find these interesting rules and filter out the undesired ones without our visual analysis approach.

The analyst continued his exploration. The interesting 3-itemset list \(I_3\) and strong rule set \(R_3\) were generated (Fig. 6 (c)). He found that there was still a rule that contains “Milk” and “Digital Accessories”: \(r_2 = \{“Underwear”, “Milk”\} \Rightarrow \{“Digital Accessories”\}\). 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 \(c(r_2) > min_{\text{conf}} > c(r_1)\).

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.

### 7.1 User Feedback and Discussion

During the post-study interview, we asked the user to give subjective feedback about the visualization and the interactions. The user showed that our system did not take him much time to understand our approach (his training session was less than 8 minutes).
And he mentioned that the matrix-based visualization was intuitive and helped him with task 1. Especially, he said that “The distribution matrix visualization and the interaction for attribute constraint are helpful”. He noted that task 2 and 3 were done by the visual interactions of the system effectively. He also told us that “The function of filtering itemsets and attributes enabled me to focus on interesting commodities.”

As for comparison of the existing automatic association rule algorithms and our approach, he mentioned that in his daily analysis, it was always hard to target interesting rules. He found that the interactions for modulating of 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 results of our system are precise because they are based on the well-studied mining algorithms. Our system can 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.

8. 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. Our system’s capacity of finding interesting association patterns is verified with a real world case study.

For future work, we plan to support the user to define new measures interactively during the visual analysis. We also expect to improve the matrix-based visualization by using a level-of-detail visualization of matrix to show rules in different granularities.

9. ACKNOWLEDGMENTS
Supported by National 973 Program of China (2015CB352503), Major Program of National Natural Science Foundation of China (61232012), National Natural Science Foundation of China (61422206), Major Program of National Natural Science Foundation of China (61520132001), and Zhejiang Provincial Natural Science Foundation (61532012), National Natural Science Foundation of China (61422206). The authors would like to thank the anonymous reviewers for their constructive comments.

8. 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. Our system’s capacity of finding interesting association patterns is verified with a real world case study.

For future work, we plan to support the user to define new measures interactively during the visual analysis. We also expect to improve the matrix-based visualization by using a level-of-detail visualization of matrix to show rules in different granularities.

9. ACKNOWLEDGMENTS
Supported by National 973 Program of China (2015CB352503), Major Program of National Natural Science Foundation of China (61232012), National Natural Science Foundation of China (61422206), Major Program of National Natural Science Foundation of China (61520132001), and Zhejiang Provincial Natural Science Foundation (61532012), National Natural Science Foundation of China (61422206). The authors would like to thank the anonymous reviewers for their constructive comments.

REFERENCES