



FIRE: a two-level interactive visualization for deep exploration of association rules

Abhishek Mukherji¹ · Xika Lin² · Ermal Toto² · Christopher R. Botaish² · Jason Whitehouse² · Elke A. Rundensteiner² · Matthew O. Ward²

Received: 14 May 2017 / Accepted: 19 May 2018 / Published online: 16 June 2018
© Springer International Publishing AG, part of Springer Nature 2018

Abstract

While rule mining is critical for decision-making applications, rule mining systems still lack support for interactive exploration of multitude of generated rules and understanding of relationships among rule results produced with various parameter settings. Based on a novel parameter space-driven approach, our proposed *Framework for Interactive Rule Exploration* [FIRE (PARAS/FIRE homepage: <http://paras.cs.wpi.edu/>)] addresses this usability shortcoming. FIRE features innovative visual displays and interactions to enable interactive rule exploration. We propose two linked interactive displays, namely the parameter space view (PSpace) and the rule space view (RSpace) that together enable enhanced sense-making of rule relationships. The PSpace view visualizes the distribution of rules produced for diverse parameter settings. This not only facilitates user parameter selection for rule mining but also enhances an analyst's understanding of rule relationships in the parameter space context. The RSpace view provides a detailed display of the rules using a novel rule glyph visualization to facilitate interactive visual rule comparisons. We evaluate the usability and effectiveness of our FIRE framework with two studies. First, in a case study a researcher explored a dataset of interest using the FIRE paradigm as well as the state-of-the-art rule visualization techniques from the ARulsViz R package. Further, our user study with 22 subjects establishes the usability and effectiveness of the proposed visual displays and interactions of FIRE using several benchmark datasets. Overall, this research encompasses significant contributions at the intersection of data mining and visual analytics.

Keywords Interactive data discovery · Rule space exploration · Data visualization · Rule mining

1 Introduction

1.1 Motivation

Mining of associations and correlations from huge datasets is critical for applications ranging from market basket analysis [2], bioinformatics [33] to intrusion detection and web usage mining [22]. However, even the most advanced rule mining approaches [3,13,38] are faced with two-key challenges, namely (a) unacceptably high response times that are not suitable for interactive analysis (performance); and

(b) lack of support for sense-making of rule relationships (usability). Existing rule mining algorithms [2,13,38] tend to be compute-intensive, rendering even their fast implementations, such as [3], inadequate for interactive analysis. Mining systems with delayed response time risk losing a user's attention and, more importantly, are thus unacceptable in mission-critical applications.

Over the years, significant focus has been placed on addressing the performance challenge [1,3,19,23,38]. Recent experiments [19] using IBM Quest [2], webdocs [22] and other benchmark datasets demonstrate that the *preprocess-once-query-many* (POQM) solutions [19,23] can offer near real-time responsiveness due to preprocessing and indexing. This near real-time responsiveness lays the foundation for offering speedups essential for interactive rule exploration. While significant strides have been made on this performance challenge, the usability of rule visualization systems has received little attention [14,15].

Supported by National Science Foundation under Grants IIS-0812027, CCF-0811510 and IIS-1117139.

✉ Abhishek Mukherji
abhishek.mukherji@gmail.com; amukherji@acm.org

¹ Cisco Systems Inc., San Jose, CA, USA

² Department of Computer Science, Worcester Polytechnic Institute, Worcester, MA, USA

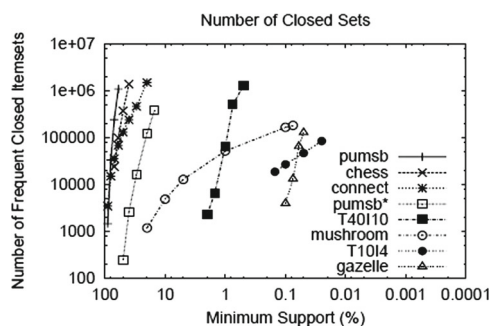


Fig. 1 Distribution of CFIs by datasets

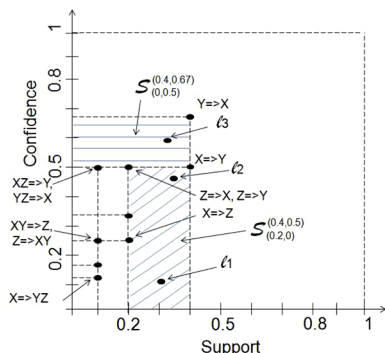


Fig. 2 The rule parameter space

The usability of rule mining systems suffers from the fact that a large number of rules is typically generated. A detriment is the lack of support for interactive exploration of relationships among rule results produced with various parameter settings. Such guided exploration of rulesets and their relationships, also referred to as sense-making, tends not to be the focus of existing rule mining systems [12,14,15,19,23].

On the other hand, recent works on rule relationships [5,6,31] and actionable high-utility rules [28] have begun to make significant advances in defining functions for measuring the utility of rules and complex rule relationships other than the traditional frequency-based measures. Qin et al. [27] study rule relationships in the specific domain of multidrug adverse reactions. However, analysts using these advanced techniques would still need to sift through the generated list of rules manually. That is, the focus of any of these advanced approaches is not the visual support of the rule discovery and exploration process. Our proposed work on designing visualizations and interactions thus complements advanced high-utility and rule relationship techniques, such that together they can enable sense-making of actionable rules in real-world applications as further described below.

1.2 Challenges

The challenges hindering the usability of rule mining systems can be summarized as below.

1. Numerous ways of sense-making One key challenge of mining interesting association rules is that there are numerous "interestingness" definitions and parameters; at times domain-dependent ones [27]. Sense-making means not only discovering co-occurring itemsets with high support and confidence (see Appendix A.2 for definitions), but also distinguishing interesting rules from obvious ones. For example, in banking data, if the restriction is that those maintaining a balance below the minimum limit will be charged a monthly maintenance fee, then a low account balance and fine paid will have high support and/or confidence yet be an obvious rule. Instead, an interesting rule to learn may be "mortgage payment in the beginning of every month causes the account to go low balance". However, its support and/or confidence may not be as strong as those for the obvious rule, because other reasons for the low balance may exist. Overall, there is no fixed rule of thumb for sense-making of association rules. Interestingness depends on several aspects including the domain, the dataset and the user’s perspective. Thus, a tool that allows users to systematically explore the mined rules is needed instead of simply presenting the rules with high scores.

2. Lack of interactive parameter space view Several rule visualization techniques have been proposed [12,14,15], yet none provide a broad view of the parameter space for rule mining with parameter selections or refinements. In the absence of parameter space insights, analysts may not be aware of the appropriate thresholds of support and confidence parameters required to obtain the rulesets of interest from any particular dataset. Figure 1 (taken from [37]) depicts how the distribution of closed frequent itemsets (CFIs) differs significantly from dataset to dataset. While gazelle and T10I4 benchmark datasets [30] have most CFIs concentrated around only 0.1% support, chess and mushroom datasets instead feature ≥ 2000 CFIs at 94 and 50% support, respectively. Thus, for a dataset, automated learning of parameter ranges and interactive presentation on a parameter space is desired. Moreover, existing systems [12,14,15] can only extract top-k rules based on one parameter at a time. However, certain interesting rules may have high support yet low confidence, and vice-versa. Such a two-dimensional combination of support and confidence (or, recently proposed interestingness measures [27,28]) for top-k rule extraction is not yet available. This feature has the potential to provide interactive mining over a learned parameter space.

3. Limited insights into rule relationships A set of rules may consist of identical itemsets, yet items may be distributed differently in antecedent and consequent, few dominating rules from the set may implicitly imply the others, defined as redundancy relationship among rules [1] (Appendix A.2). These relationships could be leveraged to represent the complete ruleset with just a subset of rules, thus reducing the

clutter. However, in [19] we discover that redundancy is a query-time phenomenon, i.e., redundancy among rules must be resolved based on the user selected parameters. Unfortunately, existing association rule tools [12,14,15] lack a mechanism to manage these dynamic rule relationships. Additionally, graphical representation of other rule relationships [5,27] are also required.

4. Lack of support for ruleset comparison When existing systems are used for discovering interesting rules in a given dataset, analysts must go about a tedious and time-consuming trial-and-error process of parameter selection interleaved with rule generation and sifting through the extracted rules to discover interesting ones. Ability to compare rulesets across different parameter values with minimal clicks in real-time will truly enable interactive rule exploration.

5. Rule tabular view is inadequate Identifying similarities and differences among rules based on their attributes is yet another desired feature. Beyond manual sifting through list of rules, analysts using existing systems [12,14,15] cannot gain such insights into rules. Further, grouped rule view to discover clusters and outliers among rulesets is also desirable.

Therefore, development of an interactive data mining technology, capable of not only answering mining requests but also providing parameter tuning recommendations together with support for improved sense-making of rules to overcome the above challenges, is imperative for effective support for decision-making applications.

1.3 Contributions of the FIRE framework

Our proposed *Framework for Interactive Rule Exploration (FIRE)* for rule exploration successfully tackles the above challenges. While the backend innovations are detailed in our prior work PARAS [19] targeting mining performance optimizations, data modeling and indexing; here we introduce complementary visual paradigm that features innovative visual displays and interactions to enable analysts to conduct rule exploration in real-time. While we introduced the notion of rule distribution across parameter settings in our short paper [24], in this work, we extend these preliminary ideas by including the dual-space interactive rule visualization paradigm along with its comprehensive evaluation with conclusive results, as detailed below:

- We propose a novel visual rule exploration framework, called **FIRE**. **FIRE** supports rule exploration at two layers of abstractions, namely, the overall parameter space view (**PSpace**) and the detailed rule space view (**RSpace**). Both layers are supplemented with innovative features and interactions.
- The **PSpace** view displays the overall distribution of rules within the space of interestingness parameters (such as

support and confidence). Salient features of the PSpace view include (a) parameter recommendations via stable region abstractions; (b) rich insights into region-wise rule cardinality; (c) capture of rule redundancy relationships; and (d) the rule cardinality skyline to explore alternative results. The PSpace interactions, described in Sect. 4, focus on resolving the challenges 1 to 3 in Sect. 1.2.

- The PSpace-RSpace hierarchical relationship (Sect. 4.3) enables real-time exploration using the two views. The stable regions are selected via the **PSpace** view and the **RSpace** view shows detailed information about particular rulesets within the selected region. To enable interactive filtering of rules, the RSpace view includes antecedent/consequent auto-fill filters and parameterwise sorting features. This addresses challenge 4 in Sect. 1.2.
- While the tabular RSpace view [24] lists rules with detailed information, to facilitate visual sense-making of rulesets (challenge 5), we now introduce an effective visualization technique called **rule glyphs**, adapted from [32], for *graphically* representing association rules. To visually capture key properties of rules and their interestingness measures, we design three variants of the RSpace glyph views, namely, *lined*, *connected* and *filled* glyphs. We also explore various glyph placement strategies that enable analysts to gain insights into clusters of similar rules as well as to detect outliers of rules that deviate significantly from the norm (Sect. 5).
- In this journal paper we present a case study comparing **FIRE** with the state-of-the-art rule visualization techniques in ARulesViz R package [14]. The case study is qualitative in nature where a researcher learns to use FIRE for the first time, and is tasked with documenting his interactions while exploring a new dataset of interest. The researcher also explored the same dataset using a combination of 10 rule visualization techniques in the ARulesViz R package. He concluded that FIRE enabled him to discover patterns in the dataset that were either undiscovered or cumbersome to derive using state-of-the-art techniques (Sect. 6).
- We also conducted an extensive user study to evaluate the diverse capabilities of our FIRE visual paradigm. 22 participants were used to evaluate the usability and effectiveness of various features of the FIRE framework over several benchmark datasets [30]. The user study is comprised of two evaluations, namely, of the PSpace view (first presented in [24]) and the new RSpace glyph view, respectively. This extensive user study provides evidence that our proposed FIRE visualizations are efficient and effective in helping analysts to understand the rule distribution over the parameter space and to gain rich insights into the rule relationships via the RSpace glyph view and the glyph placement strategies (Sect. 7).

2 Foundation of parameter-driven rule mining

The core principle we adopt for our interactive rule exploration framework corresponds to the *preprocess-once-query-many* (POQM) paradigm [1]. In an offline step, we extract all rules from a dataset that satisfy a minimum *primary support*. Then we compactly index the large number of extracted rules for subsequent interactive rule exploration by analysts. In particular, we adopt the *parameter space-driven approach* [19] which, in the context of rule mining, consists of a two-dimensional space of *support* and *confidence* (Fig. 2). A *parametric location* ℓ_1 is a point within this space, denoted by $(\ell_1.\text{supp}, \ell_1.\text{conf})$. Several rules may map to the same location, e.g., in Fig. 2 rules $(XZ \Rightarrow Y)$ and $(YZ \Rightarrow X)$ both map to $(0.1, 0.5)$.

Stable region abstractions In [19] we describe an important observation that, for many real datasets, several regions of the parameter space contain no rules at all. Additionally, the same set of rules may recur in many different regions across a large range of diverse parameter settings. The parameter space can thus be divided into several disjoint regions, which we henceforth call *stable regions*. The key idea of a stable region is that the ruleset valid for any possible parametric location within this region remains unchanged. On the other hand, rulesets valid for two locations not in the same stable region are guaranteed to be distinct. For example, consider the shaded region $\mathcal{S}_{(0.2,0)}^{(0.4,0.5)}$ in Fig. 2 where $(0.4, 0.5)$ and $(0.2, 0)$ are the upper and lower bounds of the region's support and confidence values. *Stable regions* form our *coarse granularity abstractions* for storing and managing rules. Lastly, while a rule $R = (Y \Rightarrow X)$ first appears in region $\mathcal{S}_{(0,0.5)}^{(0.4,0.67)}$, it may also be valid for region $\mathcal{S}_{(0.2,0)}^{(0.4,0.5)}$. In that case, $\mathcal{S}_{(0,0.5)}^{(0.4,0.67)}$ is said to be the *leading neighbor stable region (l-neighbor)* for $\mathcal{S}_{(0.2,0)}^{(0.4,0.5)}$.

In an offline step, the *parameter space* is partitioned into a finite number of *non-overlapping stable regions*. For each such *stable region* the following are maintained (a) the rules that are valid within that region and (b) the links to its *l-neighbors*. For details of these algorithms please refer to [19].

Rule redundancy resolution Redundancy relationships among rules can be leveraged to filter out redundant rules for presenting succinct query results to the user. Two types of redundancies are defined in [1], namely, *simple* and *strict* (see appendix for definitions). In [19], we observe that **rule redundancy is a query-time phenomenon and dependent on the user parameter selection**. Thus, rules cannot be tagged as redundant and discarded apriori. In [19], we designed algorithms that effectively precompute rule redundancy relationships in the context of our parameter space.

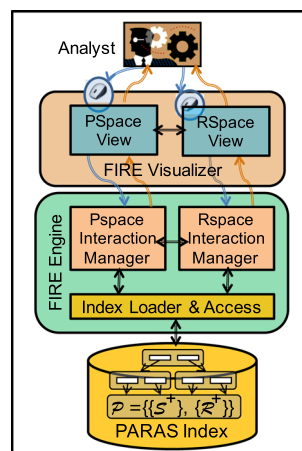


Fig. 3 The FIRE architecture

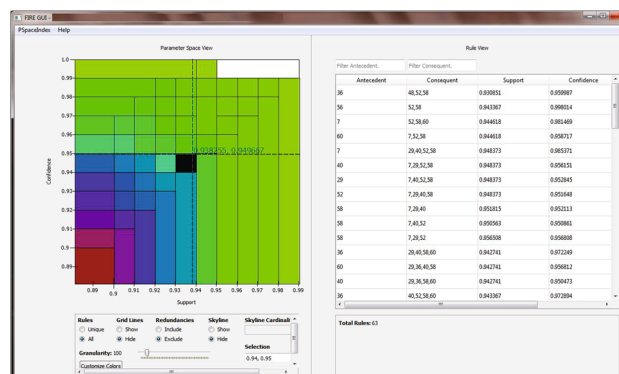


Fig. 4 The FIRE visual interface

3 Overview of FIRE visual paradigm

Our **FIRE** Visualizer (Fig. 3) supports a rich variety of analytical interactions over the PARAS index. The FIRE visual interface¹ (Fig. 4) enables analysts to explore the stable region abstractions of the parameter space model and the corresponding rulesets with ease—thus supporting effective visual analytics. FIRE is composed of a visual paradigm with two layers of interlinked visual interfaces, namely, the *PSpace* view and the *RSpace* view. The *PSpace* view (Sect. 4) displays the overall distribution of rules within the space, facilitating parameter tuning and exploration at a higher level of abstraction. The *RSpace* view (Sect. 5) provides alternate tabular and rule glyph visuals. The tabular view displays the rules in text format, including their itemsets in its antecedent and consequent together with the support and confidence values. The *RSpace* glyph view, which is a novel visualization

¹ The FIRE tool is available at [11] as a web interface for researchers to upload their own datasets, generate association rules on the datasets and visualize the rules.

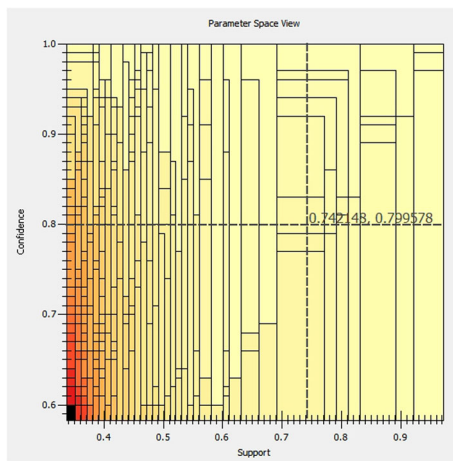


Fig. 5 PSpace (all rules) for the mushroom dataset

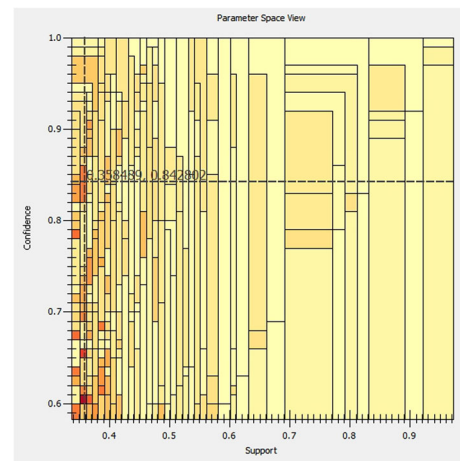


Fig. 7 PSpace (unique + non-red.) for the mushroom dataset

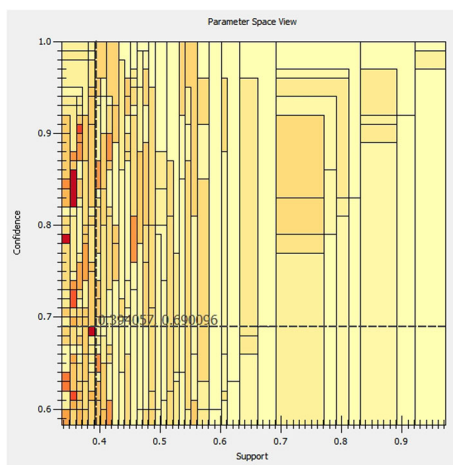


Fig. 6 PSpace (unique rules) for the mushroom dataset

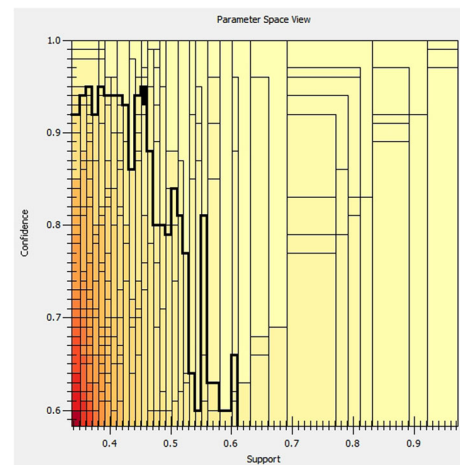


Fig. 8 Rule cardinality skyline (>100 rules)

to show association rules. The glyph view enables analysts to gain rich insights by applying glyph placement strategies to find clusters of similar rules and to detect outliers. The FIRE visualizer is powered by the PARAS backend algorithms [19]. When a dataset is first loaded into FIRE, the PARAS backend generates all rules and organizes the rules into the PARAS index for compact storage. PARAS also includes query processing algorithms to respond to the user visual requests efficiently in real-time. The Index Access module offers an API for accessing the PARAS index that would have been constructed in an offline step using PARAS. When the same dataset is reloaded, the index is directly used as rules are already pre-generated.

4 Interactive visual parameter space design

Below we introduce FIRE's parameter space (PSpace) visual paradigm where rules are distributed over the two-

dimensional space of parameters (here, support and confidence) together with all the visual interactions available to analysts.

4.1 The PSpace visualization

In this work, we design a novel abstract view of the distribution of rules on the parameter space called the *PSpace* visualization. As depicted on the left-hand side (LHS) of Fig. 4, the PSpace view displays rules in a two-dimensional plot of the stable regions within a space of support (x -axis) and confidence (y -axis) dimensions. Depending on the distribution of rules within the two-dimensional space, datasets may differ in number, size and density of the stable regions. Two such examples are shown in Fig. 4 (LHS) depicting the Chess dataset and in Fig. 5 depicting the rule distribution for the mushroom dataset. Both are benchmark datasets taken from the UCI Machine Learning Repository [30]. The PSpace view offers a compact rule space driven by a parameter-centric perspective.

4.2 The PSpace interactions

The following user interactions are provided on the PSpace visual engine.

Stable region display for fast parameter exploration For a dataset with a sparse distribution of rules in the parameter space, even when a user submits several successive mining requests with distinct (minsupp,minconf) input parameter values, a rule miner may often repeatedly return the same set of rules. When using an existing rule miner [12,14], the analyst may have to progress through a frustrating trial-and-error process to finally get a new set of rules. When using the PSpace visualizations, the analyst can instead explore the parameter space by clicking through different regions. Every time she is guaranteed to receive a distinct rule set for investigation. This way, FIRE saves time and effort by laying out the complete distribution of rules in the parameter space. In FIRE, analysts can navigate through regions by either indirectly typing in the support and confidence values in the textbox (Fig. 4) or by directly clicking on the stable regions displayed on the PSpace view.

Rich insights into region-wise rule cardinality To provide rich insights into the density of rules within different regions, a color map is used where different colors denote different cardinality/count of rules. Figures 4 (left) and 5 show two example datasets. Each shade of color denotes the count of rules within the region. Here, a lighter color depicts low count and a darker color depicts high count. FIRE offers a variety of color palettes to choose from including variants of sequential, diverging and qualitative ramps [35]. This tool enables users to select color schemes of interest to customize their displays.

Analysts can use the left bottom panel of the PSpace visualizer for a variety of interactions with the PSpace. For example, users can interactively show either all rules that appear in a region or only the rules unique to each region. For a dense dataset such as *Chess* [30], each parameter setting produces a huge number of rules. Suppose that an analyst changes the parameter values by clicking on the PSpace interactive UI from (minsupp^{old},minconf^{old}) to (lsupp,lconf) such that minsupp^{old} ≥ lsupp and minconf^{old} ≥ lconf. Then the ruleset $\{\mathcal{R}\}^{(lsupp,lconf)}$ would also contain the rules in the original ruleset satisfying (minsupp^{old},minconf^{old}). The change in the ruleset may be difficult to quickly grasp by manual inspection. Here, a delta output of rules is desirable which can be achieved in FIRE simply by selecting the *Unique* option. While Fig. 5 depicts an *All* rules view, Fig. 6 shows the PSpace view for the same dataset when the user selects the *Unique* option via the radio button.

Rule redundancy resolution To the best of our knowledge, FIRE is the first rule visualization system that allows analysts to optionally select to display only the non-redundant rules for a data set. By excluding redundant rules, a suc-

cinct set of fewer rules can be displayed in the PSpace view that covers all rules for ease of analysis. In the context of the stable region abstractions, interesting patterns can be observed when redundancies are excluded (Fig. 7) compared to when they are included (Fig. 6). In fact, any combination of unique/all and redundant/ non-redundant rules can be selected via the radio buttons to observe different patterns of rule distributions over the PSpace view. Further the results displayed can be analyzed using interactions.

Rule cardinality skyline interaction Figure 8 depicts the skyline view that provides recommendations beyond a single stable region boundary. Consider the situation when the analyst wants to find the top-k (say, 100) rules in a dataset. However, at times it is unclear which parameter (support or confidence) to give priority to. By selecting the skyline option on the LHS bottom panel the analyst can input the desired cardinality in the *skyline cardinality* textbox (say, 100). The skyline is then drawn on the PSpace view to mark for each support value (*x*-axis), the confidence value (*y*-axis) having ≥ 100 rules. As a lower confidence value will result in a higher number of rules, the regions below the skyline will contain ≥ 100 rules while those above the skyline will contain < 100 rules. Therefore, the analyst can now select from a range of support and confidence settings that will all return up to the top 100 rules based on particular support and confidence combinations. Furthermore, the analyst can now quickly determine various observations about the dataset. For instance, using the rule cardinality skyline in Fig. 8 one can observe that no region contains ≥ 100 rules above support = 0.61.

Assisted navigation through PSpace visualization Additional features such as cursor positions, optional grid line and zooming are provided to assist the analyst in navigating through the PSpace view. Some of these features can be seen in Fig. 9. In our early user study, we found that while using FIRE, analysts may not be comfortable initially in identifying the support and confidence of desired regions on the PSpace view. Therefore, we have introduced the cursor position feature. Namely, as the analyst moves the cursor over the PSpace, the current cursor position is displayed. In Fig. 9, the current cursor position is (0.74...0.84...).

4.3 Visualizing the PSpace–RSpace relationship

Viewing the rule distribution in the PSpace stable region display is at a level of abstraction higher than the RSpace view of individual rules or rulesets with their respective antecedent and consequent. PSpace-RSpace linkage enables real-time exploration using the two views as described below. By default, the RSpace view loads with all rules mined from the dataset.

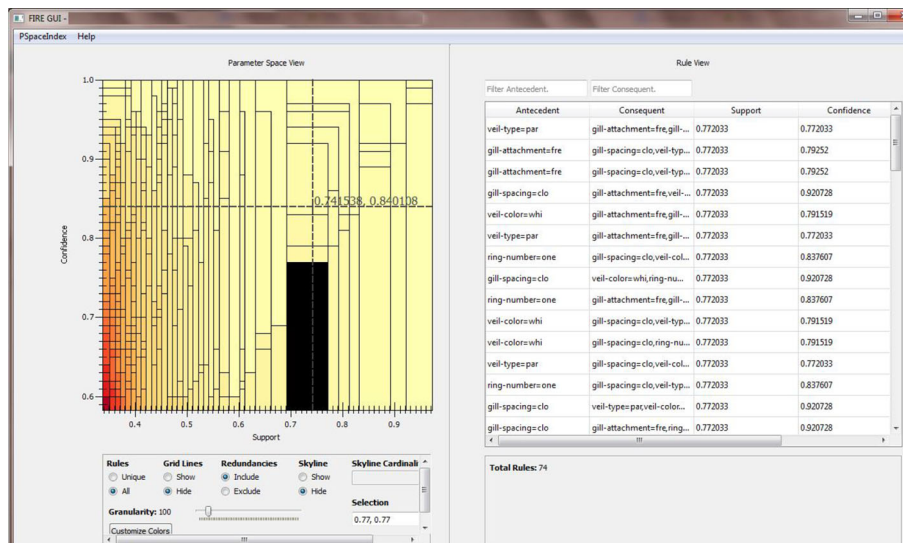


Fig. 9 PSpace-RSpace linkage

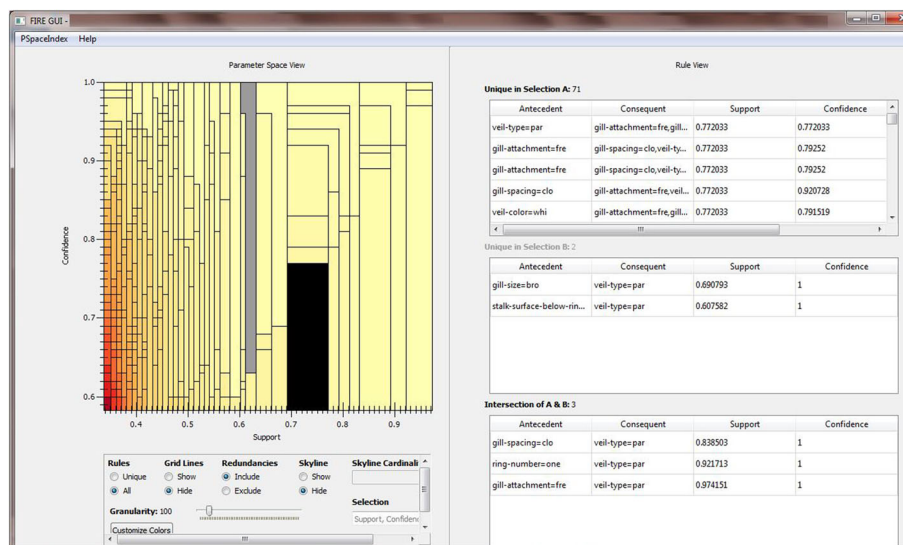


Fig. 10 Comparing two regions

Drill-down via PSpace-RSpace linkage As shown in Fig. 9, when the analyst selects a single region on PSpace view (highlighted in *black*) the rules valid within that region can be viewed in the RSpace view via cross-links between the two views. This supports instant drill-down into individual rules while still maintaining the global context via the PSpace view.

Visual region ruleset comparison Analysts can also select two regions at a time to compare their respective rulesets. In Fig. 10 comparing two stable regions facilitates the analysis of how the change in parameter settings effects the output. Region A is selected with a left click (highlighted in *black*) and region B is selected with shift + click (highlighted in *gray*). Through cross-links, the RSpace view then

will present a comparative display of unique rules within each region and also the common rules shared among these two regions A and B, if any. Here, we see that region A 71 unique rules and region B has 2 unique rules, with 3 common rules.

5 Interactive visual rule space design

Here, we describe the design of the two RSpace views, namely, *tabular* and *glyph* views along with their respective interactions.

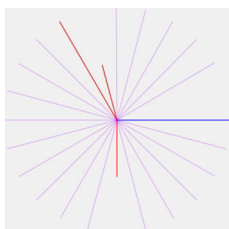


Fig. 11 Lined glyph 1

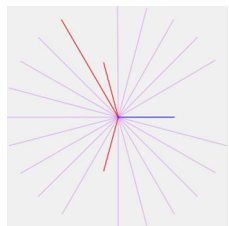


Fig. 12 Lined glyph 2

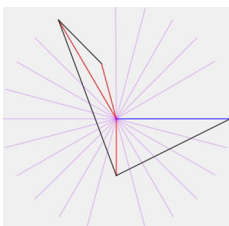


Fig. 13 Connected glyph

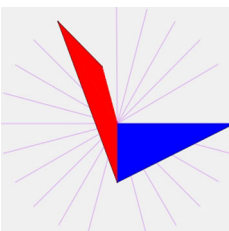


Fig. 14 Filled glyph

5.1 The RSpace tabular view

In most mining tools, rules are listed in a *tabular* RSpace view as depicted on the right-hand side (RHS) of Fig. 9. Tabular view provides detailed information such as antecedents and consequents of each rule together with support and confidence values. The total number of rules within the selected region on the PSpace view is displayed at the bottom of the RSpace table.

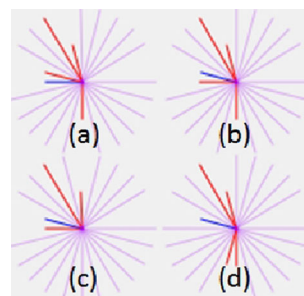


Fig. 15 Lined glyphs

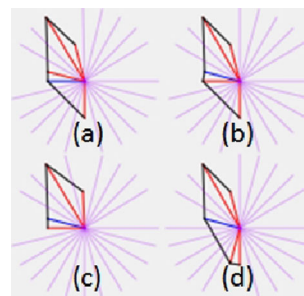


Fig. 16 Connected glyphs

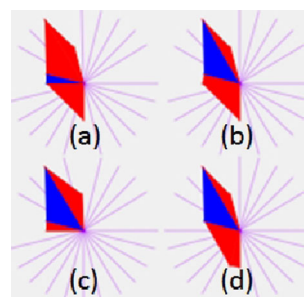


Fig. 17 Filled glyphs

5.2 The RSpace glyph view

The purpose of the detailed RSpace view is for the analysts to visually analyze similarities or differences between the rules being displayed. However, as confirmed by initial user testing, this task is difficult to accomplish by using only the tabular view due to the overload of textual information. Beyond the straightforward tabular view described above, we thus designed a novel RSpace glyph view for graphically representing association rules to facilitate efficient visual analysis of rulesets. A glyph is known to be an effective visualization technique for displaying multivariate data [32]. Glyphs are effective for visual shape comparisons as well as finding clusters or outliers by applying glyph placement strategies. However, to the best of our knowledge, glyphs have never been used to visualize association rules before. Below we describe three variants of our proposed RSpace glyph views.

Rule View

veil-type=par
gill-spacing=clo

Antecedent	Consequent	Support	Confidence
veil-type=par	gill-attachment=fre,gill-...	0.772033	0.772033
veil-type=par	gill-attachment=fre,gill-...	0.772033	0.772033
veil-type=par	gill-spacing=clo,veil-col...	0.772033	0.772033
veil-type=par	gill-spacing=clo,ring-nu...	0.795667	0.795667
veil-type=par	gill-attachment=fre,gill-...	0.812654	0.812654
veil-type=par	gill-attachment=fre,gill-...	0.812654	0.812654
veil-type=par	gill-spacing=clo,veil-col...	0.81487	0.81487
veil-type=par	gill-spacing=clo	0.838503	0.838503

Total Rules: 8

Fig. 18 Tabular RSpace

Lined glyph design A lined glyph (Fig. 11) resembles a 360 degree clock dial with multiple hands. Given a dataset with n attributes, we represent each attribute with a hand on the dial. Attribute hands are placed at equal angles to each other within the total of 360 degree dial. In Fig. 11, the mushroom data set [30] containing 22 attributes is represented with 22 hands. The lined glyph represents the rule $\{poisonous? = edible\} \rightarrow \{gill-attachment = free, veil-type = partial, veil-color = white\}$ from the mushroom dataset. The attributes that participate in a rule are highlighted while the rest of the attributes are displayed in a faded manner.

For each attribute the distinct values are displayed using different hand lengths. For example, the attribute *poisonous?* has 2 distinct values, namely, $\{edible, poisonous\}$. Thus, the hand lengths are encoded such that *poisonous? = edible* is represented by a full length hand (blue hand in Fig. 11), whereas a half length hand would represent *poisonous? = poisonous* (blue hand in Fig. 12).

Further, in order to distinguish the antecedents from the consequents, we propose to draw them using two different colors. In Fig. 11, the single antecedent *poisonous? = edible* is represented with a blue hand, whereas the three consequents $\{gill-attachment = free, veil-type = partial, veil-color = white\}$ are each represented with a red hand.

Connected glyph design The intuition for the connected glyph design is that it is easier to visually comprehend the similarities and differences between different shapes rather than those of the combination of hand positions. For the same example rule discussed above using the lined glyph shown in Fig. 11, we now depict the connected glyph in Fig. 13. The connected glyph is a simple modification of the line

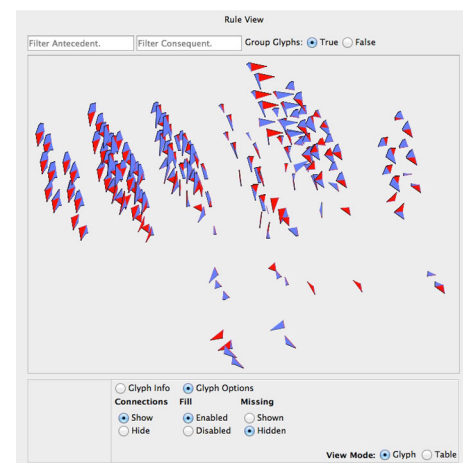


Fig. 19 PCA placement of rules

glyph with the outside ends of the highlighted attribute hands connected to each other so as to give it a shape. Two adjoining hands are connected only if they are within less than 180 degrees of each other in the clockwise direction. Otherwise, this will introduce ambiguity. Further, we use a distinct color for the connection lines (here, black) to distinguish them from the antecedents and the consequents.

Filled glyph design Initial user trials revealed that the connected glyphs were not effective for certain tasks such as distinguishing between antecedents and consequents. Thus, we propose a third glyph design called the filled glyph. The filled glyph display further fills colors inside the shapes created by connecting adjoining hands. The space between two adjoining highlighted attribute hands is filled with the color

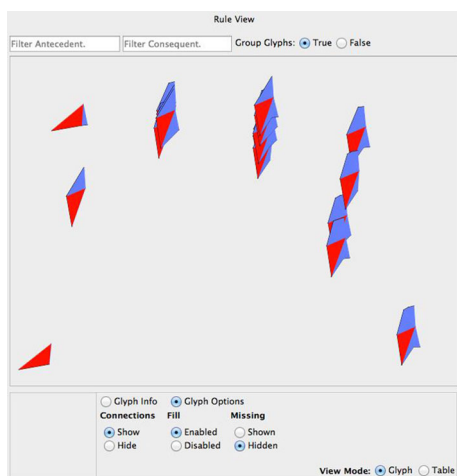


Fig. 20 MDS placement of rules

of the first attribute hand in a clockwise manner. In Fig. 14, the space between hands representing attribute *stalk-color-above-ring = white* and *stalk-color-below-ring = white* is filled with *blue*, i.e., the color of the antecedent. Namely, in this case the antecedent is *stalk-color-above-ring = white*. The space between the attribute hands *stalk-color-below-ring = white* and *veil-type = partial* is filled with *red*, i.e., the color of the consequent *stalk-color-below-ring = white*. Again, the space between two adjoining hands is filled only if the hands are within less than 180 degrees of each other in the clockwise direction.

Comparison of glyph designs The purpose of these three glyph representations is to enable the analysts to visually comprehend the similarities and differences between the rules displayed in the glyph view. Our intuition is that these graphical representations are easier to comprehend and work with than the tabular display. Further, the purpose of providing multiple glyph options is that for different tasks, different glyph displays may be more effective, as confirmed by our evaluation. In Figs. 15, 16 and 17, a set of 4 rule glyphs are shown using lined, connected and filled glyph designs, respectively. Our hypothesis is the following, based on initial user trials. If a task involves counting of hands such as "to find the rule with the minimum number of consequents (*red hands*)", the lined glyphs are most effective. On the other hand, if a task involves similarity detection such as "to find the rules containing the same antecedents (*blue hands*)", then the filled glyphs can effectively reveal the most prominent pattern. Connected glyphs, however, will be efficient for tasks that may involve both counting hands and requiring some shape information. A formal user study (Sect. 7) examines the glyph designs and their relative effectiveness in detail.

5.3 The RSpace interactions

Using different interactions designed for the RSpace view, analysts can drill-down to gain rich insights into rule subsets as described below.

Filtering and sorting of rulesets In case of an overwhelmingly large number of rules being displayed in the RSpace view, the analyst can filter the rules based on antecedent and/or consequent values using an auto-fill control. In general, this allows the analyst to determine which rules are prominent for a given item/itemset. For example, in Fig. 18, the antecedent is filtered on *veil-type = partial* and the consequent is filtered on *gill-spacing = close*. We note that only 8 rules out of the original 74 rules (Fig. 9) satisfy the filter. This is a more manageable number for human analysis. The antecedent and consequent filters are available for both the RSpace views, namely tabular and glyph. As shown in Fig. 18, the rules can also be sorted by *descending/ascending* support or confidence. This is achieved by clicking on the *support* or *confidence* column header, respectively. This is particularly useful if a set contains some rules that have high support yet low confidence and others have high confidence yet low support.

Customizable glyphs. Lastly, the ability to customize colors for distinguishing between antecedent and consequent provides a powerful visualization as certain patterns can be visualized with contrasting color schemes. Further, analysts can choose among any of the three glyph displays; each facilitating easy discovery of different pattern types.

5.4 RSpace glyph placement

Yet another important capability in information visualization is the placement or layout of glyphs on a display to communicate significant information regarding the values of individual glyphs themselves as well as relationships between the objects represented by the glyphs [32]. Here, we explore various placement strategies in the context of our proposed RSpace glyph view. The explored methods range from *data-driven* strategies that use data dimensions as positional attributes to *structure-driven* strategies that base the placement on implicit or explicit structure inherent within the dataset. A comprehensive taxonomy of placement strategies has been developed in [32] to assist the visualization designer in selecting the technique most suitable to his or her data and task. In our context, this feature enables analysts not only to gain insights about clusters of similar rules (e.g., rules with identical antecedents) but also to detect outliers that are separated from the rest of the rules.

In this work, we employ derived data-driven placement techniques that generate glyph positions using analytics applied to the data values as a whole input. Thus, instead

of a location reflecting only one, two, or three of the data dimensions, it reflects a combination of all the dimensions in an attempt to convey N -dimensional relational information in the smaller number of dimensions. Common dimensionality reduction techniques [32] include Principal Component Analysis (PCA), Multidimensional Scaling (MDS), Self-Organizing Maps (SOMs), spring-based models and so on.

We have adapted two of these layout techniques, namely, *PCA-based* placement (Fig. 19) and *MDS-based* (Fig. 20) placements in our FIRE visualization. PCA finds linear combinations of the dimensions that best explains the largest variation in the multivariate dataset. The first two principal components are then used to determine the position of a glyph in a 2-D space as they capture the most prominent combinations of the original attributes that distinguish the data. In contrast, MDS is an iterative refinement process that attempts to adjust weights or positions until a certain criterion is met. In the context of rule glyphs the criteria would be common antecedents and/or consequents. In our case, the distances (or similarities) between glyphs in 2-D is a good approximation of the similarity of the rules based on the participating itemsets as antecedents and consequents.

6 Case study of a bike sharing dataset

We evaluated the usability and effectiveness of our FIRE framework in two stages. In this first stage, we introduce a case study² during which a researcher explored a dataset of interest. The case study is qualitative in nature. The researcher independently explored the bike sharing dataset [29] from the UCI machine learning repository using (a) FIRE and (b) ARulesViz as described below.

6.1 Exploring the dataset using FIRE

Dataset description The bike sharing dataset contains two years of bike usage data. Each data instance contains the counts of casual (walk-ins) and registered users in a given day and information about weather conditions (temperature, humidity) and holiday status (weekday, weekend, holiday). The contributors of the dataset claim that "most of the important events in the city could be detected by monitoring these data" [29]. The aim is to mine rules that link bike usage to holidays, workday status and weather conditions. While a brief preamble is given below, detailed description of the preprocessing performed on the dataset can be found in Appendix A.

Generating rules and loading them into FIRE The data was loaded into FIRE with a minimum support of 5% and

minimum confidence of 60%. As the dataset contains 732 instances, each representing a day, 5% support means a rule would be mined only if it is present in at least 36 days, or in more than 1 month out of 24 months. Therefore, the researcher believes this value constitutes a good primary support. Given these parameters FIRE generated the rules and loaded them within a few seconds. The total number of rules generated was 9673. In the ALL rules setting (Sect. 4.1), the set of 9673 rules can be listed in the RSpace view by clicking on the lowest stable region (0.05, 0.6) on the PSpace view. The highest rules are located in the upper right corner (Fig. 21). However, from the PSpace view, we find that the stable region with maximum support and confidence is empty. The two neighboring stable regions are as follows. The region (0.683,1) contains a rule with confidence = 1 and support < 1 and the region (0.807, 0.976) contains a rule with maximum possible support (80.7%) and confidence < 1. These rules are listed below:

1. {workingday = yes \rightarrow holiday = no} (support = 0.683, confidence=1): a common knowledge rule, correctly derived by the data, yet uninteresting.
2. {adjusted_casual = low \rightarrow holiday = no} (support = 0.807, confidence = 0.976): this rule can be interpreted as the number of casual users being low during non-holidays.

As holidays are rare (~ 11 days per year, or less than 5% of the data), the primary support of 5% does not cover rules with "holiday = yes". The PSpace view for the bike sharing dataset in Fig. 21 clearly lets us learn with just one glance that the rules with support > 50% are rare in this dataset. We were able to quickly explore all such regions. One interesting rule we found in this space is: {workingday = yes \rightarrow adjusted_casual = low} (support: 0.671, confidence = 0.982). This means that overall walk-ins are low on working days. It is common knowledge that 5 out of 7 days are working days, which gives an expected maximum support level of 71%. However, 22 of the working days are holidays for the duration the bike sharing dataset was collected, or approximately 3% of data. Thus, working days make up approximately 68% of the data. Dividing support by confidence serves as a sanity check, and arrives at the same number without the need to have prior knowledge about the data: $0.671/0.98 \simeq 0.68$ (68%). In other words, this rule implies that working days are $\sim 68\%$ of the instances, and for $\sim 98\%$ of those working days, the casual user count is low.

Using rule filtering and sorting features Having explored all regions with support > 50%, the next step was to explore rules with support $\leq 50\%$. The rule mentioned in the previous section is: {workingday = yes \rightarrow adjusted_casual = low} (support: 0.671, confidence = 0.982). This rule has a

² This case study was performed by an avid bike user with an interest in data mining.

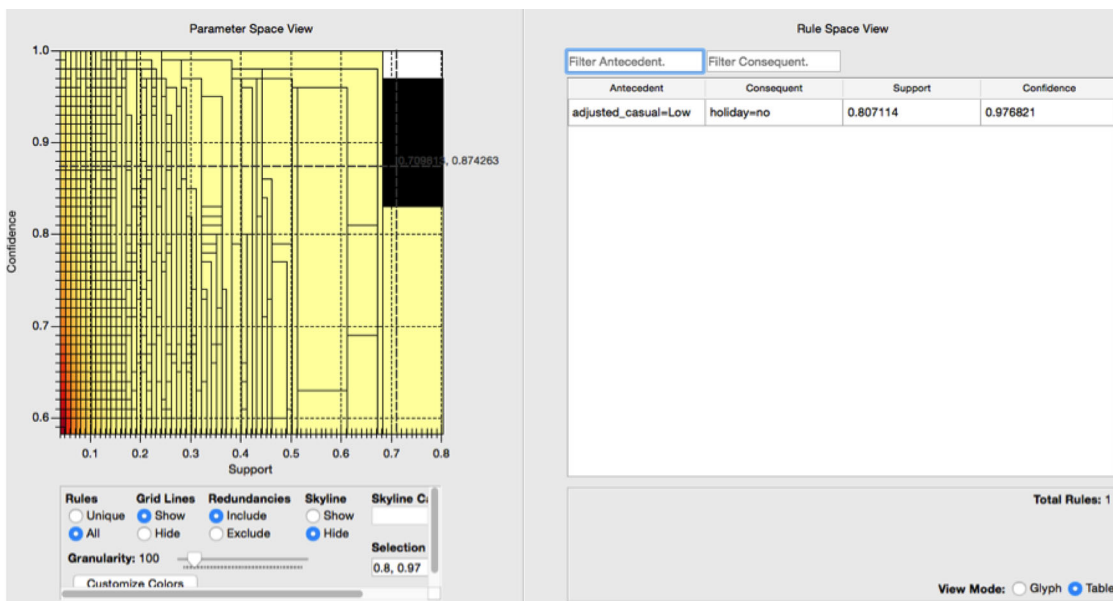


Fig. 21 Finding the highest rules: the highest no common knowledge rule

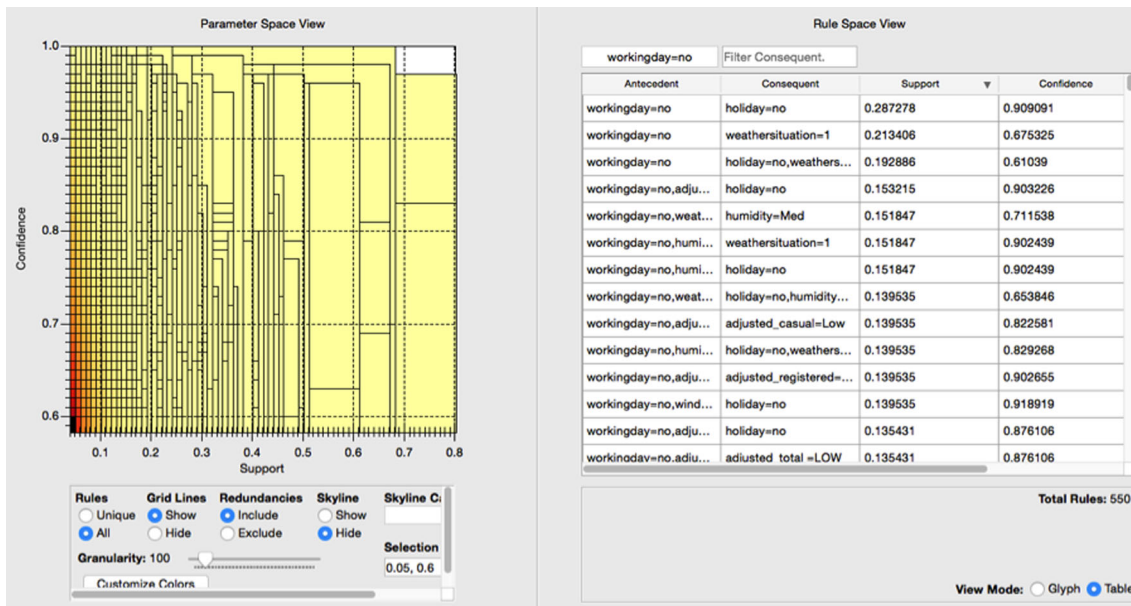


Fig. 22 Filtering for weekends

strong support value due to the large number of instances that contain working days. What about rules when “workingday = no”? In the bike sharing dataset “workingday = no” includes all weekends as well as holidays. To find these rules The researcher took the following steps:

1. In the PSpace view, he clicked on the stable region with the lowest coordinates (0.05, 0.6). This then resulted in 9673 rules being listed,
2. Then in the RSpace view, he filtered for rules with “workingday = no” in the antecedent. This resulted in 550 rules being listed,

3. Lastly in the RSpace view, he sorted rules by descending support values.

The same three steps can be repeated by filtering for “workingday = no” in the consequent. These features of FIRE are described in Sect. 4.1.

The highest support value for a rule containing “workingday = no” was 28.7%. However, this rule represents common knowledge (Fig. 22) {workingday = no → holiday = no} (support = 0.287, confidence = 0.909). Thus, in other words, 90.9% of the non-working days are weekends and the rest are

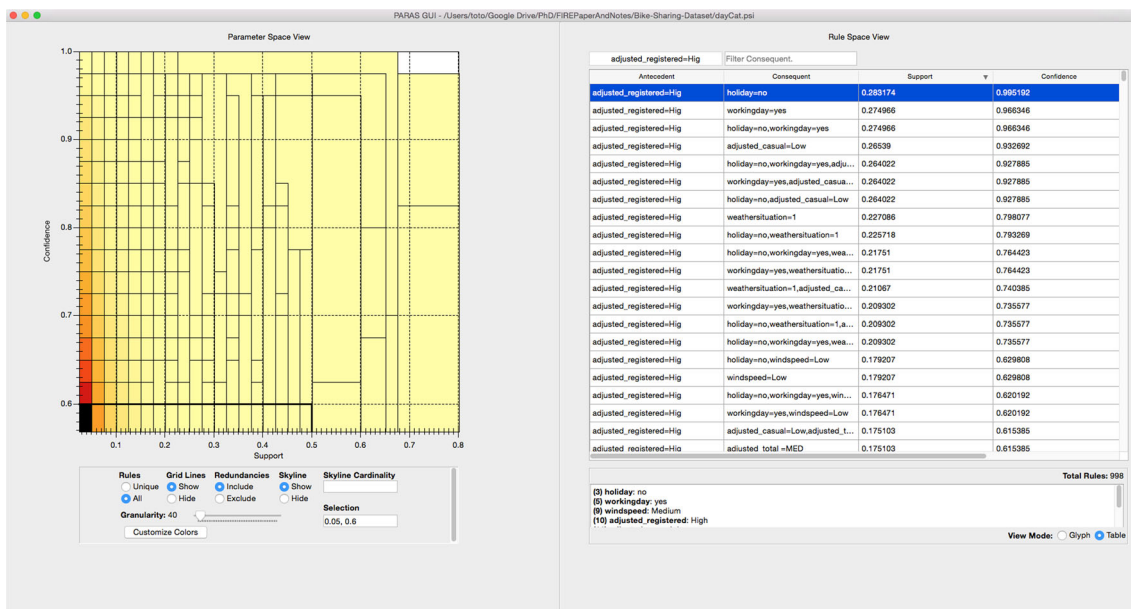


Fig. 23 Filtering for registered users

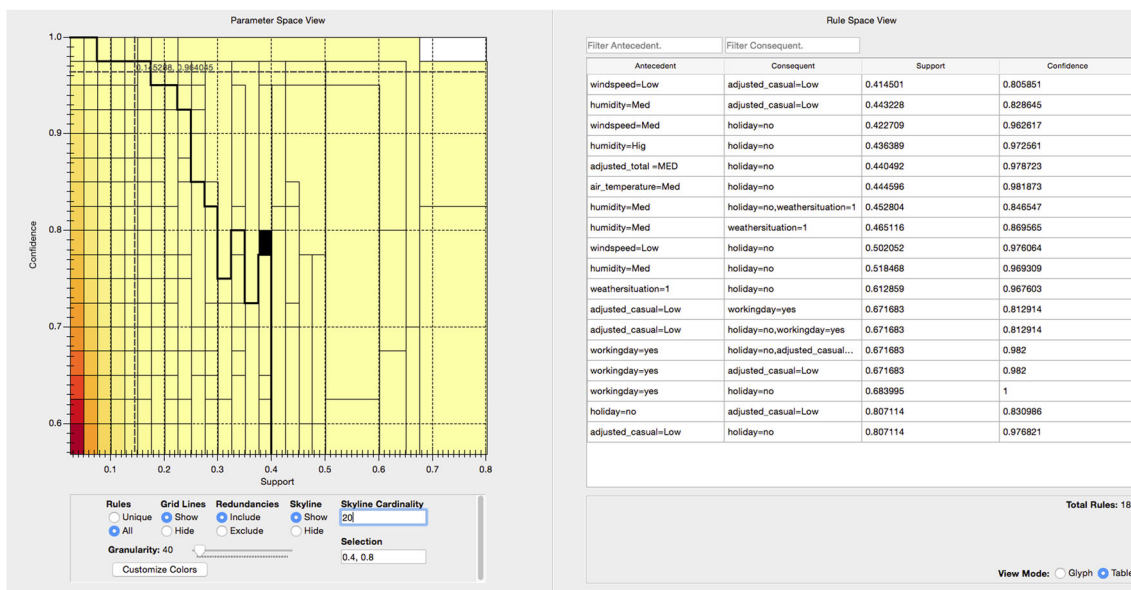


Fig. 24 Skyline cardinality: distinguishing regions with > 20 rules and ≤ 20 rules

holidays. However, a non-common knowledge rule found in this space was: {adjusted_casual = High → workingday = no} (support = 0.0519, confidence = 0.974). This rule indicates that bike rentals by walk-in users were high 6% of the days (0.0519/0.974) and that in 97% of these instances it was not a working day. This rule makes sense as walk-in users have other means of transport for their daily lives and they are instead much more likely to rent bikes on weekends and holidays.

Thus far the researcher found rules mostly related to casual users. He further explored rules related to the registered users. For this purpose, he followed the same three steps as in the

case of filtering for “workingday = no”. Instead, in step 2, he then filtered for rules with “adjusted_registered = High” in the antecedent as shown in Fig. 23. There are 998 such rules. He then sorted them by their descending support. The highest support possible for a rule with “adjusted_registered=High” is 28.3% with a 99.5% confidence (highlighted in blue color). The top 3 rules indicate that registered bike users are high in numbers during working days. However, the fourth rule: {adjusted_registered = High → adjusted_casual = Low} (support = 0.264, confidence = 0.932) shows an interesting inverse relationship between the count of registered and

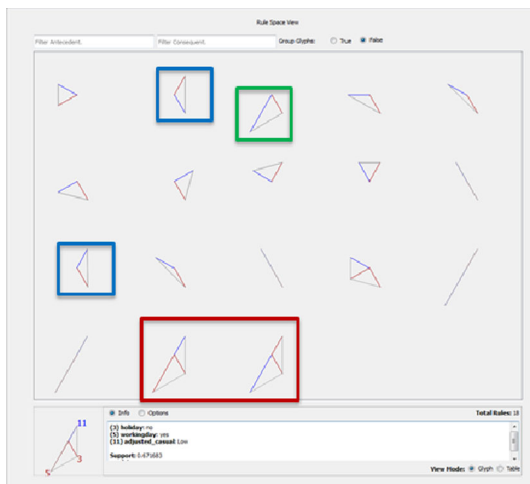


Fig. 25 Comparing rule glyphs



Fig. 26 Clustered rule glyphs

casual users. Specifically, whenever “adjusted_registered = High”, 93% of those days “adjusted_casual = Low”.

Utilizing the Skyline feature over the PSpace view for retrieving the regions with a certain cardinality The case study thus far involved exploring the different stable regions and going through the list of rules in each region. In the *ALL* rules view, the count of rules cumulatively increases as the researcher moves toward lower support or confidence settings. Further, he wanted to list the top-k (say, 20) rules. However, as there are two rule ranking criteria, namely, support and confidence, he employed the skyline cardinality feature that allowed him to separate stable regions with more than *k* rules from those with less than *k* rules (Fig. 24).

The regions adjacent to the skyline were the most interesting, because they had a high number of rules with good support and confidence value. For example, the highlighted region (0.4, 0.8), which is above the skyline, contains 18 rules. In addition to the rules explored thus far, several new rules involving temperature, humidity, weather situation can be found in this region. One of these rules is {adjusted_total = MEDIUM → holiday = no} with a support of 44% and a high confidence of 97.8%. Here, the total rentals are discretized into three values {LOW,MEDIUM,HIGH}.

Comparing rules using the glyph view For the stable region (0.4, 0.8) that the researcher explored above, he noticed from the tabular view that several of the 18 rules had common attributes in the antecedent and/or the consequent. Thus, he next wanted to compare the rules and see which ones are similar, i.e., have common attribute values. In order to compare all rules, he needed to manually compare $C(n, r) = n!/(r!(n - r)!)$ possible combinations of rules. In our concrete example $n = 18$ and the $r = 2$, we have 153 possible comparisons to make. This problem becomes increasingly complex with a large number of rules and is difficult to do



Fig. 27 ARulesViz scatterplot UI

Table 1 Comparison of association rule visualization techniques

Technique	Rule set	Measures	Interactive	Reordering	Ease of use
Scatterplot	Large	3	✓		++
Two-Key plot	Large	2 + order	✓		++
Matrix-based	Medium	1		✓	0
Matrix-based (2 measures)	Medium	2		✓	-
Matrix-based (3D bar)	Small	1		✓	+
Grouped matrix	Large	1	✓	✓	0
Graph-based	Small	2			++
Graph-based (external)	Large	2	✓	✓	+
Parallel coordinates	Small	1		✓	-
Double-decker	Single rule	(2)			-
<i>FIRE PSpace</i>	Large	2	✓	✓	+++
<i>FIRE Rule Glyph</i>	Large	2	✓	✓	+++

in the tabular list of rules. While the glyph view does not reduce the number of comparisons he had to make, he found it easier to look for the similarity among shapes using the rule glyph view (rule glyphs are defined in Sect. 5.2). Figure 25 shows several such examples. The two rules depicted using the blue boxes are opposite to each other, i.e., one with “holiday = no” in the antecedent and “adjusted_casual=Low” in the consequent, and the other vice versa. Similarly, the two rules within the red box contain three attributes each and their antecedent/consequent sequence (“adjusted_casual = Low”/“workingday = yes”) is swapped. Further, we see that the rules depicted in the red box can be obtained by combining the attributes of the rules in the blue box and the rule highlighted within the green box. Overall, the researcher found the glyph representation convenient for visual shape comparisons among rules and rulesets.

Glyph clustering functionality The next step in the exploration was to enable clustering for the group of rules as above. The goal was to look for outliers that might contain interesting rules. When enabling clustering, as expected, the rules described in the previous section grouped together based on the commonality of attributes (see Fig. 26). Further, several rules that have common set of attribute-value pairs are grouped together such that the common attribute-value pairs are depicted by a shared line or lines close to each other.

6.2 Exploring the dataset using ARulesViz

ARulesViz is a popular R package that contains a total of 10 state-of-the-art association rule visualization techniques [14]. The visualizations include: (a) scatterplot (2 variants), (b) matrix-based (4 variants), (c) graph (2 variants), (d) parallel coordinates, and, (e) double-decker. Details of each visualization technique can be found in [14]. Inside the R environment, the researcher typed in R commands to

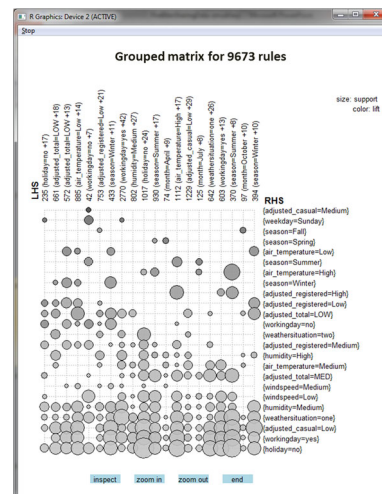


Fig. 28 ARulesViz grouped matrix UI

load the Bike Sharing dataset [29]. Then using the ARules package association rules were generated. Finally, using the ARulesViz package the rules were visualized using the different rule visualization techniques available in the ARulesViz package. The overall comparison of these visualization techniques is shown in Table 1. This comparison extends the original comparison given in [14] by adding the two primary visualization techniques of FIRE, namely, (a) FIRE PSpace stable regions view, and (b) FIRE RSpace rule glyph view.

Three of the matrix-based visualizations, graph-based, parallel coordinates and double-decker visualizations support a medium to a small number of rules at a time. On the other hand, scatterplot variants, grouped matrix, and graph-based (external) as well as FIRE PSpace and Glyph views can support a large rule set. In the interactive scatterplot view (Fig. 27), one can select an arbitrary region (shown as a red shaded box) and show the list of rules that qualify for the

selected support and confidence in the region in the console output (here, a total of 43 rules). This interaction in effect is equivalent to the *unique* rules view in the FIRE PSpace visualization. The limitation of the scatterplot view is that for an arbitrarily chosen region that includes several rules, a high number of rules will be listed in the console view. Thus several rules may be hidden unless the rule list is explored exhaustively. Moreover, no reordering support is available in the scatterplot visualizations.

The FIRE PSpace view can be considered as a layer of the stable region abstraction over the scatterplot view. Additional features of the FIRE PSpace view such as unique rules, redundancy exclusion, and skyline provide semantic filters based on support and confidence measurements, rule redundancy definitions as well as the cardinality of rules, respectively. Further, all these techniques (Table 1) can be categorized by the number of measures (e.g., support, confidence and lift) that can be simultaneously visualized. While the scatterplot allows three measures (two on the axes, one using color/shade), most other approaches allow two measures at a time. The FIRE PSpace view utilizes color mapping schemes to denote the density of rules in the stable regions.

As described in [14], to explore large sets of rules with graph-based visualization, advanced interactive features like zooming, filtering, grouping and coloring nodes are needed. Such features are available in interactive visualization and exploration platforms for networks and graphs like Gephi. From the ARulesViz package [14], graphs for sets of association rules can be *exported* in the GraphML format or as a Graphviz dot-file to be explored in tools like Gephi. This process of exporting the rule graphs is cumbersome for interactive exploration. On the other hand, the FIRE RSpace tabular view is enabled with antecedent and consequent auto-fill filters as well as support and confidence ordering for enhanced exploration through list of rules. The FIRE rule glyph view utilizes color schemes to differentiate antecedents from consequents. The details of each rule represented by a glyph, such as the antecedent and consequent values of the rule can be seen at the bottom of the RSpace view by hovering over or selecting the glyph.

The grouped matrix view (Fig. 28) is a variant of matrix-based visualization technique such that the rules are grouped based on common antecedents and consequents (see [14] for details). The view utilizes a K-mean clustering algorithm for the same, where the user needs to provide the value of K (default value of $K = 20$). In Fig. 28, all 9673 rules are shown with $K = 20$. The appropriate value of K for any dataset needs to be learnt using trial-and-error. Moreover, the LHS, shown in the top x -axis, consists of clusters of multiple antecedent values grouped together. This made it difficult for the researcher to comprehend the items other than the single one listed in each column. Similar in flavor to the grouped matrix view is the FIRE rule glyph clustering

Table 2 User study schedule

Tested feature	Tasks	Duration (min)
S1. PSpace stable region and interaction	T1	2–4
	T2	2–5
	T3	3–6
S2. PSpace-redundancy/RSpace-tabular-filter	T4	3–6
S3. PSpace skyline exploration	T5	4–6
S4. RSpace glyph rule analysis	T6	2–3
	T7	2–3
S5. RSpace glyph placement analysis	T8	2–3
	T9	4–8
	T10	2–3
Total		26–47

approach, where the researcher utilized PCA and MDS layout (see Sect. 5.4 for details). However, the full details of the attributes in the antecedent and the consequent can be viewed in the RSpace view by hovering over the group.

Conclusions Overall, the FIRE PSpace view together with its rich diversity of features effectively supports interactive exploration for a high number (~ 9673) of rules for the bike sharing dataset [29]. In addition the RSpace view, in particular, the rule glyph visualizations enables effective comparison of rules. Having graphical displays and interactions on otherwise static sets of rule enable novel interactions with the data and a rapid exploration of the rule space. Moreover, compared to the state-of-the-art association rule visualization techniques in [14], that required the researcher to understand and type in syntactically correct R command line inputs or scripts, FIRE is a completely graphical visualization tool as every feature is available through intuitive clicks through labeled interactions.

7 Evaluation using a user study

7.1 Evaluation methodology

Here, we further present the second stage of our evaluation of the FIRE system. We conducted a controlled user study to compare the features of FIRE to that of the state-of-the-art systems such as Weka [12] and measured the effectiveness of different visual representations compared to the list of rules provided by Weka.

7.1.1 User study procedure

The overall process was as follows: The subjects perform a series of 5 studies listed in Table 2. As the studies progressed,

the study administrator explained the purpose and process for each task with examples. Lastly, the subject fills out an exit questionnaire. On average the study took between 26 and 47 minutes per subject.

7.1.2 Tools compared

Our user study compares our FIRE visualizer to the cached association rule miner (CRM). CRM is an association rule miner based on the APRIORI algorithm [2] but with instant response time due to the cached rules. CRM provides users with a tabular view of rules and all functions offered by existing rule mining systems (e.g., WEKA [12]).

7.1.3 Metrics of evaluation

We measured both efficiency and accuracy of the subjects in accomplishing the tasks. For efficiency, we measured the time consumed by each subject for each task. For accuracy, we measured the percentage of correctly answered tasks by the subjects.

7.1.4 Datasets

We chose two datasets from the UC Irvine Machine Learning Repository [30], namely, chess and mushroom. The chess dataset is derived from the game step. The mushroom dataset contains characteristics of various species of mushrooms. Chess and mushroom datasets have ≥ 2000 closed frequent itemsets at 94% and 50% support, respectively (see Fig. 1).

7.1.5 General method

Each subject was asked to perform all of the five studies (S1–S5) described in Sect. 7.2. To avoid carryover effects and learned knowledge about a dataset, we counter-balanced the order of tasks, datasets and tools. For S1, we switched datasets and tools. For example, half of the subjects performed the task T1 on the chess dataset using CRM, and T1 on the mushroom dataset using FIRE. On the other hand, the other half of the subjects performed the task T1 instead on the mushroom dataset using CRM and T1 on the chess dataset using FIRE. For S2 and S3, we switched both the questions and the tools. Particularly, we asked subjects to find characteristics of edible mushrooms using CRM and characteristics of poisonous mushrooms with FIRE. This way addressed the "pre-knowledge" problem. For S4, we randomized the order of showing different glyph displays for each subject. For S5, we randomized the order of applying different glyph placement strategies for each subject. In general, we avoided practice and fatigue effects by randomizing the order of tools and tasks. In these task assignments, no carryover problems

arose, as each subject was asked to only finish a particular task on a given dataset using the tools in a random order.

7.1.6 Environment setup

We conducted our experiments on a Windows 7 PC with Intel(R) Core(TM)i5-2410M CPU@2.3 GHz processor and 4 GB of RAM, with a display resolution of 1600 by 900. Our visualizations displayed in a 1000 by 600 window.

7.1.7 Study population

We performed the user study with a population of 22 subjects (10 undergraduate students and 12 graduate students). They were either from computer science, computer engineering or mathematical sciences programs. The user study was conducted on a one-to-one basis, i.e., a tester to subject test.

7.2 Design of our user study

7.2.1 S1: Stable region usage study

In our stable region usage tests, we asked the subjects to perform three different tasks (T1–T3) by varying tools and datasets, such that each dataset was tested for each visualization in a random order. The three tasks were designed to verify the ability of the subjects to explore the parameter space, to utilize the stable region abstractions and to compare rule sets. The questions were as follows:

- T1 What are the most prominent rules by support and/or confidence?
- T2 Which setting (out of 4 choices) gives a different set of rules than the given setting?
- T3 Find the common and unique rules for two different parameter settings.

7.2.2 S2: Filter/redundancy study

In this study, we used only the mushroom dataset. We asked our subjects to first filter the antecedents of the rules and then to remove redundant rules. Some users used FIRE first and CRM next, and vice-versa. The goal was to test the ability of our subjects to use filter and redundancy removal features by asking them to perform the following task.

- T4 Find the most frequent characteristics of edible/ poisonous mushrooms.

7.2.3 S3: Skyline view study

In the skyline view study, we asked the subjects to find the top- k rules from the mushroom dataset by varying the tools

(FIRE and CRM). The goal was to test if our subjects can make use of the rule skyline cardinality. For this, we presented our subjects with the following task.

T5 Find the parameter settings that produce top- k rules in the dataset, where $k = 20, 50,$ or 100 .

7.2.4 S4: Glyph display study

In our glyph view study, we showed the subjects a set of 6 glyphs using different glyph designs, namely, lined, connected and filled. We told the subjects that the antecedent(s) is/are represented by the *blue* color and the consequent(s) is/are represented by the *red* color. We verified the hypothesis that different glyph designs may be more effective for different tasks. We presented our subjects with the following tasks.

T6 Given a set of 6 glyphs, find the rules with the same antecedents. Three questions were asked, each using a different glyph design.

T7 Given a set of 6 glyphs, find the rule(s) with the greatest number of consequents. Three questions were asked, each using a different glyph design.

7.2.5 S5: Glyph placement study

In this study, three glyph placement strategies were presented using the glyphs generated from the mushroom dataset. The goal was to test if the subjects are able to leverage glyph placement strategies to identify cluster or outlier among a set of glyphs. In addition, we verified the hypothesis that different glyph placement strategy may be more effective for different tasks. In these tests, the connected glyph design was chosen to present the questions due to the fact that the connected glyph gives a visual shape to the glyph together with serving the purpose of showing each hand (attribute) clearly.

T8 Identify outliers within a given set of glyphs using two different glyph placement strategies (i.e., the unclustered layout versus the clustered layout). Two questions were asked—each using a different glyph placement strategy.

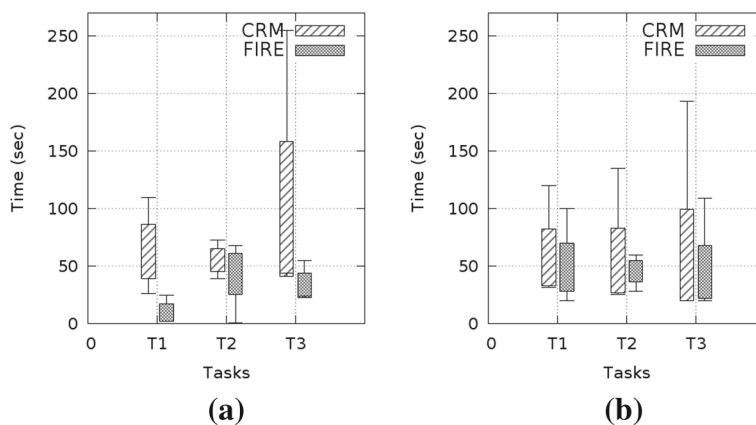


Fig. 29 Time spent on tasks 1, 2 and 3. **a** Mushroom. **b** Chess

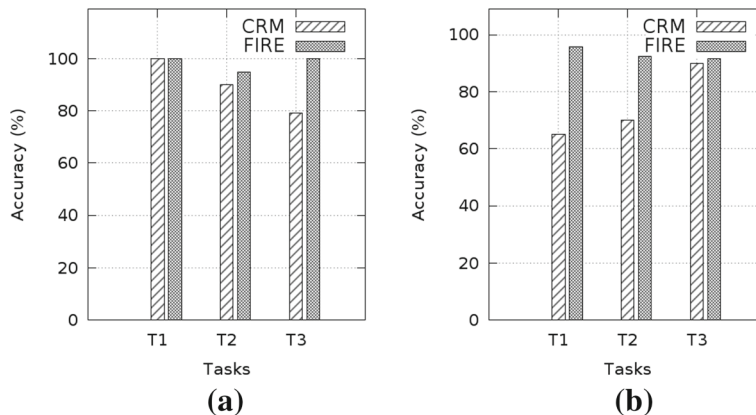


Fig. 30 Accuracy of tasks T1, T2 and T3. **a** Mushroom. **b** Chess

- T9 Given a set of glyphs, identify glyph(s) with a certain attribute-value pair using three different glyph layout strategies, i.e., unsorted layout, sorted layout and clustered layout. Total of six questions were asked, two questions using each of the placement strategies.
- T10 Using the clustered layout, identify groups of similar glyphs and count the groups containing a given attribute-value pair. Two different sets of glyphs were tested.

7.2.6 Exit questionnaire

A survey questionnaire was presented to the subjects at the end of the studies. We asked them to rate the two alternative tools, namely, FIRE and CRM in terms of their ease of use on a scale of 1–5 (where 5 = very easy, 1 = very difficult). We also asked them which tool they preferred for each of the 3 studies (S1–S3). We also asked the subjects to rank the alternate glyph designs (S4) and glyph layouts (S5) by their ease of use. Overall, they were asked the following questions about each task.

- Q1 Which task(s) is/are easier with FIRE than CRM? (list tasks)
- Q2 Which task(s) is/are easier with CRM than FIRE? (list tasks)

7.3 Hypotheses

As FIRE provides several features for interactive rule exploration, we anticipated that conducting certain tasks using FIRE would be faster and more accurate than using CRM. Also, we expected that the glyph designs and glyph layout strategies may vary in their effectiveness for different tasks. This led to the following hypotheses.

- H1 For T1, T2, T3, T4 and T5, subjects perform better using FIRE than CRM in term of both time spent and accuracy.

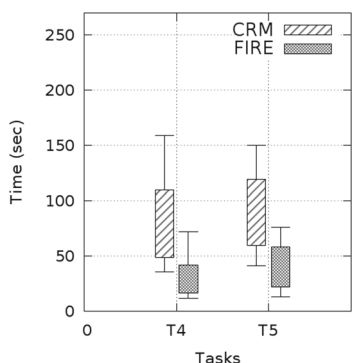


Fig. 31 Time spent on tasks T4 and T5

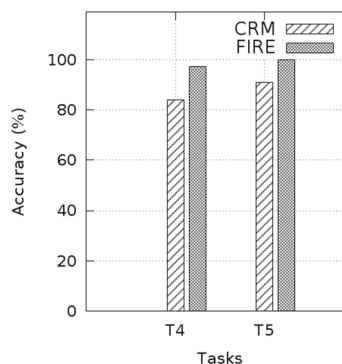


Fig. 32 Accuracy of tasks t4 and t5

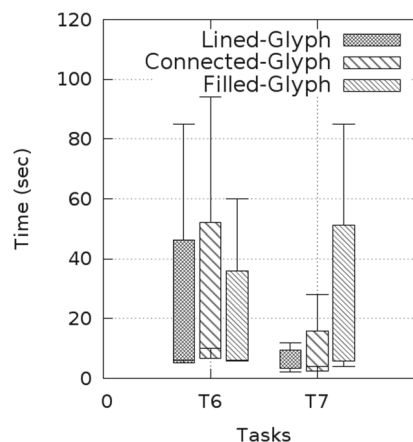


Fig. 33 Time spent on tasks T6 and T7

- H2 For T6, the filled glyph is more effective than other glyph designs, whereas for T7, the lined glyph is more effective than others.
- H3 For T8, the clustered layout is more effective than other glyph placement strategies in detecting outlier glyphs.
- H4 For T9, the sorted layout is more effective than other layouts in aiding the analyst in finding glyph(s) with a certain attribute-value pair.
- H5 For T10, the subjects can easily identify group of similar glyphs using the clustered layout.

7.4 Results and discussion

Stable region usage study As confirmed in Fig. 29, subjects took less time when working with FIRE compared to that while using CRM.

This is because the tabular view in CRM does not provide any aid or intuition for subjects to accomplish the tasks.

As shown in Figs. 29a and 30a, for task T1, subjects spent 9s on average using FIRE to get 100% accuracy while subjects used 62s on average with CRM to achieve the same accuracy. For T2, the minimum time spent was 2 seconds using FIRE while that using CRM required was at least 26

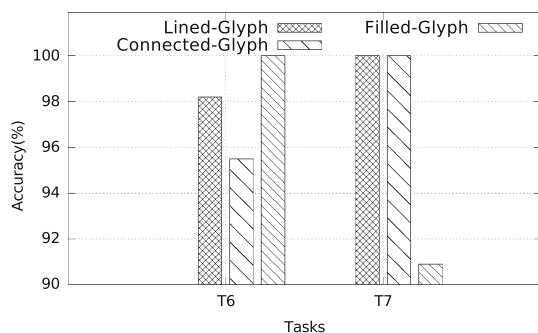


Fig. 34 Accuracy of tasks T6 and T7

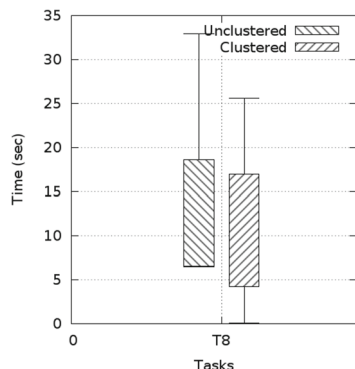


Fig. 35 Time spent on tasks T8

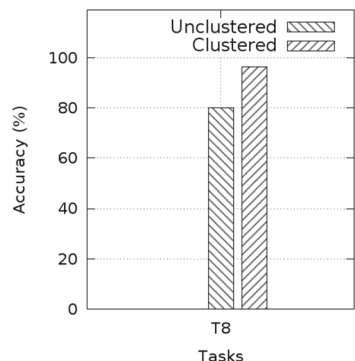


Fig. 36 Accuracy of tasks T8

seconds. Thus for T2, FIRE outperformed CRM in measures of accuracy by 5%. For T3, the maximum time spent with FIRE was 55 s, while in CRM it was 255 s. Subjects using FIRE achieved 100% correctness while in CRM this figure was 80%.

Similarly, in Figs. 29b and 30b, our subjects took less time using FIRE than CRM to complete all three tasks. At the same time, they made fewer mistakes using FIRE than CRM. In particular, the accuracy of T1 using FIRE was 30% higher than the accuracy of CRM. This is because more than one rule existed that satisfied the question in the chess dataset. Subjects tended to omit some rules that resulted in this low accuracy. In contrast, FIRE is able to reveal the full answer with just 1 or 2 clicks.

Filter/redundancy + skyline view studies In Figs. 31 and 32, we show the time spent and accuracy for tasks T4 and T5, respectively. Again, subjects using FIRE spent less time to perform the tasks, yet were able to achieve better accuracy than subjects using CRM for the same task. More specifically, subjects used 29 s on average with FIRE yet achieved near 100% accuracy for T4. The subjects using CRM, on the other hand, took 80 s and reached only 84% accuracy. Overall, the results confirmed our hypothesis H1, i.e., our FIRE technology is a win-win in terms of both efficiency and accuracy.

Glyph view study Figures 33 and 34 show the time spent and the accuracy when using the three glyph designs. The results confirmed our hypothesis H2. For task T6 that asked for antecedent similarity detection, the *filled* glyph indeed is proven to be the most effective among the three glyph designs. In particular, subjects spent 20 s on average to correctly answer this similarity detection question using filled glyphs. Those using other glyph displays took longer time and yet committed several mistakes. For T7 involving counting of the number of consequents, the lined glyph showed an impressive efficiency (avg. 6 s) and 100% accuracy. Most subjects rated the lined glyph as the easiest to use in their exit questionnaire. In T7 subjects using the connected glyph design also achieved 100% accuracy with a slightly higher time spent.

Glyph placement study In Figs. 35 and 36, we show the time spent and the accuracy of task T8 when using unclustered and clustered layouts. The subjects used less time when supported by our clustered layout, while they needed significantly more time using the unclustered layout. The fastest subject took only 1 s to complete this task with the help of our clustered layout. Accuracy-wise, subjects achieved 97% correctness using the clustered layout while only 80% accuracy was achieved by subjects with the unclustered layout. This is because our clustered layout essentially groups similar glyphs together and simultaneously unveils the outliers to subjects. The results confirmed our hypothesis H3. The subjects are able to leverage our clustered layout to recognize the outlier within a set of glyphs effectively.

For task T9, which asked the user to identify glyphs with a certain attribute-value pair, the sorted view indeed was proven to be most effective among the three placement strategies. As shown in Figs. 37 and 38, the subjects using the sorted layout achieved 99% accuracy and took less time, while the subjects using the unsorted layout achieved 80% correctness and took more time. This is because the sorted view allows subjects to sort glyphs by a single attribute using the “sort-by” function. The set of glyphs is thus classified by the specified attribute and the glyphs with the same value are naturally grouped together to facilitate search. Notable among these three layout strategies was the clustered layout,

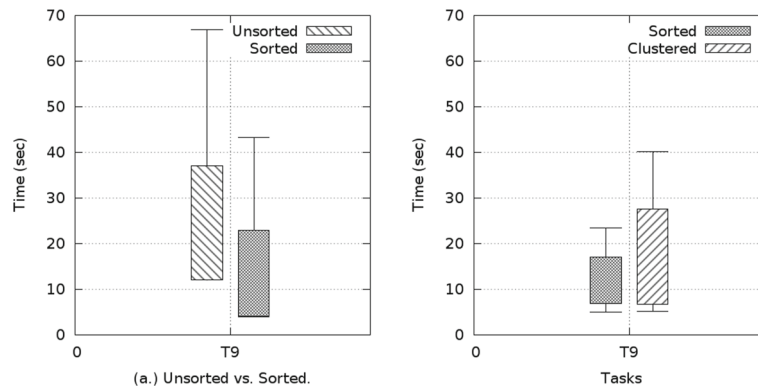


Fig. 37 Time spent on tasks T9

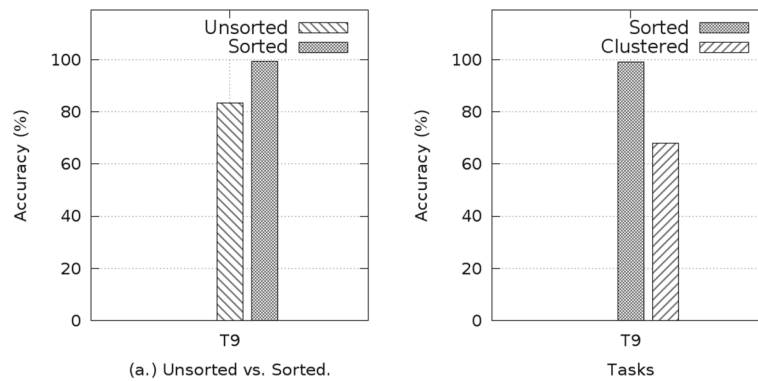


Fig. 38 Accuracy of tasks T9

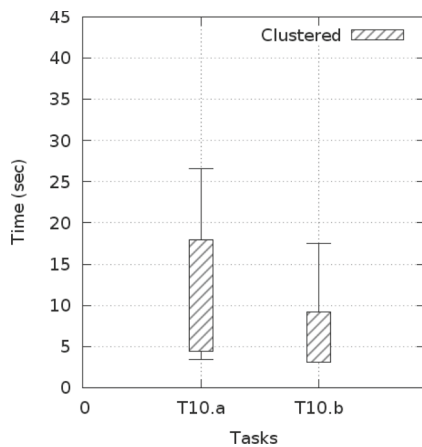


Fig. 39 Time spent on task T10

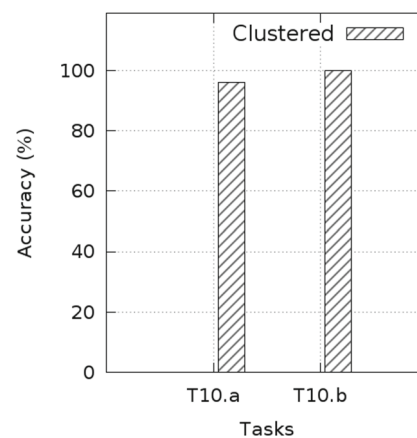


Fig. 40 Accuracy of task T10

which does not behave well in this task. Clustered layout tends to group glyphs using all of their attributes instead of a designated one, which renders it less suitable for this task.

Figures 39 and 40 show the effectiveness of identifying groups of glyphs using the clustered layout. In particular, the subjects used on average 11 and 6 s, respectively, to achieve near 100% correctness on both questions in task T10. Our hypothesis H5 is thus confirmed. In our initial trial on the sub-

jects, they were unable to perform this task well without the help of the clustered layout. The subjects could not group the glyphs correctly within an acceptable response time. Therefore, this task is best suited for the clustered layout.

Exit questionnaire Answers to Q1 and Q2 on the exit questionnaire are shown in Fig. 41. There is a clear endorsement in favor of FIRE versus CRM, especially in test T5 where none of the subjects chose CRM over FIRE. The most common

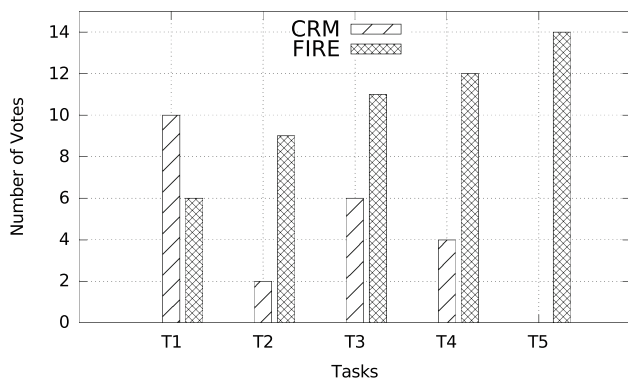


Fig. 41 Votes on the preference of CRM and FIRE in term of tasks

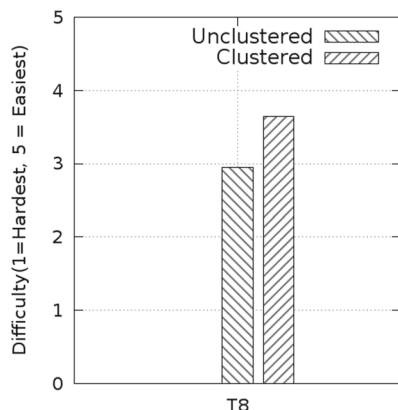


Fig. 42 Survey question on task T8

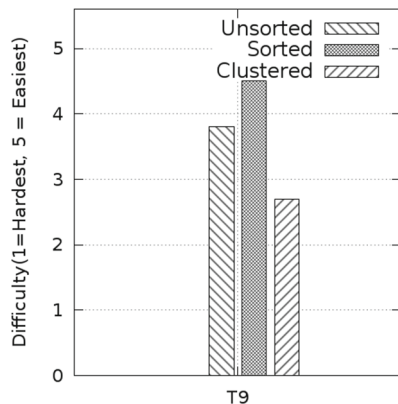


Fig. 43 Survey question on task T9

reason cited for this choice was the facilitated exploration of PSpace. The only exception was T1, as some subjects stated that they are more familiar with the sorted rules in the tabular view. In terms of ease of use, on a scale from 1 to 5, FIRE was rated 4.3 on average and CRM was rated 3. Here, 1 = very difficult and 5 = very easy. Figures 42 and 43 show the results for the glyph placement study that verified the task of finding glyphs with a given attribute-value pair using different layouts. On a scale from 1 to 5, the clustered layout was

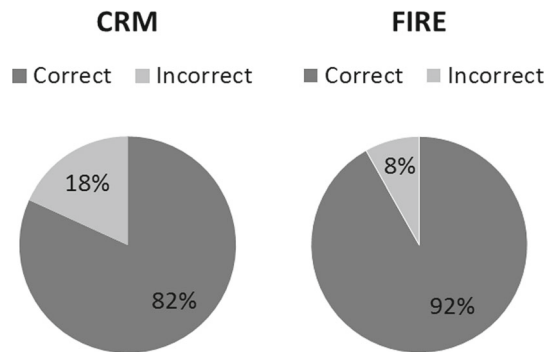


Fig. 44 Overall accuracy using CRM and FIRE

rated 2.7, the sorted layout was rated 4.5 and the unsorted glyph layout was rated 3.8. In terms of identifying dissimilar glyphs, the clustered view was rated 3 and the unclustered view was rated 3.7.

Overall, as shown in Fig. 44, the user study showed that 92% of our subjects could perform the task correctly with FIRE while 82% of them produced correct answers with CRM. In addition, the glyph representation of rules and the glyph layout strategies offered users great benefits in association rule exploration. In conclusion, all hypotheses were confirmed by our user study. Our study shows that FIRE indeed aids human analysts in performing interactive rule exploration tasks efficiently and accurately.

8 Related work

Parameter space exploration Prior research has explored the space of parameters for handling parameterized database queries [7] and tuning database configuration parameters [10]. Most data mining queries are parameterized, which, while making the algorithm flexible and tunable to one’s own problem, often causes huge difficulty as typically the selection of appropriate parameter values is left to the human analysts. Closest to our work, [36] aims to help analysts understand the relationship among clusters produced with different parameter settings to better understand good results for density-based clusters. We instead explore the parameter space for rule mining. Closest to our proposed parameter space display is the recent demonstration called AssocExplorer [21] that proposes a scatterplot of rules on a 2-D space. However, they overlook the visual clutter problem that is common even for a moderate number of rules. We tackle the clutter problem with our proposed stable region abstractions, zoom and granularity features.

Interactive association rule mining Hahsler et al. [14] presented the R-extension package *arulesViz* which implements several visualization techniques to display individual rules. In that sense, these efforts only focus on subset of our problem, namely, on designing displays for visualizing association

rules as in our RSpace view. Analogous to our RSpace view, they work with standard visualizations found in visualization toolkits including variants of scatterplots, histograms and parallel coordinates visualization techniques. In this paper, we instead propose three variants of RSpace glyph views for graphically representing individual rules. We found rule glyphs and associated placement strategies to be well-suited to facilitate exploration and comparison among rules. These core techniques can potentially be integrated into ARulesViz as well.

Couturier et al. [9] proposed an integrated framework covering both rule extraction and visualization steps of the mining process. They provided a guided exploration based on clustering of rules. Neither of these approaches provide support for understanding the distribution of rules within the space of interestingness parameters (such as support, confidence and lift). Last but not least, unlike other efforts on interactive rule mining, a key contribution of our work is its focus on evaluating the usability of our FIRE framework via a formal user study.

Online association rule mining Online mining techniques [1,16,17] typically prestore the intermediate frequent itemsets. Here, we instead adopt the approach of rule prestorage from [19] to achieve the required real-time interactive behavior. [19,23] propose to store the final rule results instead. They achieve near real-time responsiveness, laying the foundation for offering speedups sufficient for interactive rule exploration. However, sense-making of rulesets extracted from a dataset, which is the topic of our current study, is not the focus of these rule mining systems [12,14,15,19,23].

Interestingness measures as parameters Han et al. [34] identify the importance of analyzing the interestingness measures of rules. They compare different null-invariant measures, such as confidence, to provide insights into similarities and differences among them. However, they do not tackle interactive rule mining through precomputation as undertaken by our work. In a more recent work, Cao et al. [5] propose a new interestingness parameter *Max Coverage Gain*. They introduce the *MCGminer* algorithm to handle complex rule interactions and reduce the computational complexity of identifying the globally optimal rule set in a large imbalanced dataset. By extensive evaluation over 13 UCI datasets [30], their metric is proven to be accurate and effective. Our work is orthogonal to [5,34] as we provide an overall framework for interactive rule mining. The parameters and strategies proposed in these works can be added to our framework to provide a richer experiences to analysts.

Rule relationships and actionable high-utility rules Combined mining [6] techniques focus on determining and managing various aspects of patterns such as rules, e.g., relationships among patterns, pattern representation. Works on actionable high-utility itemset mining [28] establish that

itemsets that are frequent may not necessarily be of high-utility. Further, [31] presents a framework that integrates both explicit and hidden item dependencies and an algorithm IRRMiner that captures such implicit relations with implicit rule inference. These works propose utility functions to establish how significant a itemset/rule is, and bridge the gap between research outcomes and business needs. While these two works make significant advances in discovering high-utility rules and defining complex rule relationships, our interactive FIRE engine powers the discovery process itself by presenting rules in an easy explorable manner. We use rule redundancy relationships as an example of rule relationships, the concepts in these relevant works can be adopted into the backend (PARAS) of our visual FIRE engine to then together provide richer insights to the analysts for sense-making of association rules.

9 Conclusion

In this work we designed, implemented and evaluated an innovative visualization technology for interactive rule exploration called the **FIRE** framework. **FIRE** offers parameter recommendations and enhanced sense-making of rule relationships. Particularly, we propose two linked visualizations, namely, the PSpace and RSpace views. Both views are supplemented with innovative visualizations and interactions that enable analysts to effectively conduct visual rule exploration. While PSpace offers a rule distribution abstraction, RSpace facilitates a detailed analysis of rules and their relations. In addition, our novel RSpace glyph display enables visual comparison of rule shapes further augmented by glyph placement strategies [32].

Our case study using the Bike sharing dataset [29] illustrates the capabilities of the FIRE system and compares it with that of the state-of-the-art ARulesViz rule visualization techniques. Further, our user study with 22 subjects demonstrates the usability and effectiveness of the proposed FIRE framework using several benchmark datasets.

FIRE is being maintained as a system [11] and further extended [20,25–27] to include additional capabilities including negative rules, enhanced interactivity, and interestingness measures with respect to specific domains. In the future, new interestingness measures [5,6,28,31] could be incorporated to make **FIRE** more useful.

Acknowledgements This work was supported by NSF under Grants IIS-0812027, CCF-0811510 and IIS-1117139.

A Appendix

A.1 Preprocessing of the bike sharing dataset

The Bike Sharing dataset [29] was preprocessed before loading into FIRE [11] and R ARulesViz [14], as described below.

1. Data corresponding to three of the attributes was eliminated. The attributes are *instant* (unique identifier), *dteday* (date) and *yr* (contains 2 values: year 1 and year 2).
2. The casual and registered users increased over time. In particular, the casual users increase at 0.895 users per day, whereas the registered users increase at a rate of 4.874 users per day. To cancel the effect of the overall growth, the data was rotated to negate the slope of the trend lines. In Figs. 45 and 46, we show the original user counts, and in Figs. 47 and 48 we show the adjusted user counts for the casual and registered users categories. This processing is similar in flavor to season trend decomposition in [8].
3. Further, the attributes were discretized as shown in Table 3.

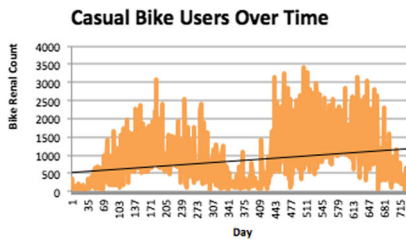


Fig. 45 Original casual user count

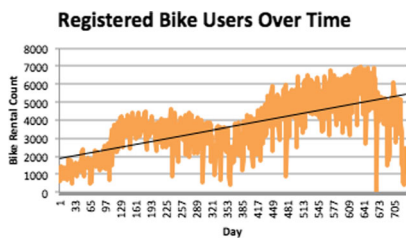


Fig. 46 Original Registered User Count

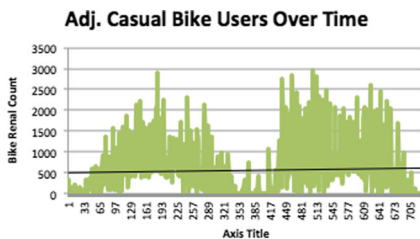


Fig. 47 Adjusted casual user count

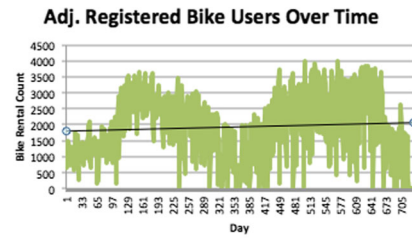


Fig. 48 Adjusted registered user count

Table 3 Discretized attributes: bike sharing dataset

Attribute	Discretization	Category
adjusted_casual	[0,986.006)	LOW
	[986.006,1972.0118)	MEDIUM
	[1972.0118,∞)	HIGH
adjusted_registered	[0,1334.0613)	LOW
	[1334.0613,2668.1227)	MEDIUM
	[2668.1227,∞)	HIGH
air_temperature	[0, 0.333012)	LOW
	[0.333012, 0.586954)	MEDIUM
	[0.586954,∞)	HIGH
humidity	[0, 0.324167)	LOW
	[0.324167,0.648333)	MEDIUM
	[0.648333,∞)	HIGH
windspeed	[0,0.184082324167)	LOW
	[0.184082,0.345773)	MEDIUM
	[0.345773,∞)	HIGH

A.2 Association rules and redundancy

An adjacency lattice (Fig. 49) denotes items such as X , Y and Z . The *support* value of each item (say, X) or itemset (say, XY) indicates the total instances of the item or itemset in the dataset. For example, in a set of 100 records, X occurs in 80 and Y in 60 records. Itemset XY has a support of 40 records. For a rule $R = (X \rightarrow Y)$, its confidence can be represented as confidence $(R) = \frac{\text{support}(XY)}{\text{support}(X)}$.

Aggarwal et al. [1] define rule redundancy relationships, such that redundant rules may be filtered out to present succinct results to the user. The redundant rules could always be derived on demand, if so desired. We examine how these

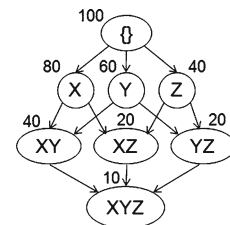


Fig. 49 Adjacency lattice example

Table 4 Redundancy in generated association rules

Rule	Support	Confidence
$X \Rightarrow YZ$	$S(X \cup Y \cup Z) = 0.1$	$S(X \cup Y \cup Z)/S(X) = 0.125$
$XY \Rightarrow Z$	$S(X \cup Y \cup Z) = 0.1$	$S(X \cup Y \cup Z)/S(X \cup Y) = 0.25$
$XZ \Rightarrow Y$	$S(X \cup Y \cup Z) = 0.1$	$S(X \cup Y \cup Z)/S(X \cup Z) = 0.5$
$X \Rightarrow Y$	$S(X \cup Y) = 0.4$	$S(X \cup Y)/S(X) = 0.5$
$X \Rightarrow Z$	$S(X \cup Z) = 0.2$	$S(X \cup Z)/S(X) = 0.25$

redundancy relationships can be identified in the parameter space model. In particular, redundancy can be of two types [1], as defined below.

Definition 1 Simple redundancy Let $A \Rightarrow B$ and $C \Rightarrow D$ be two rules such that the itemsets A, B, C and D satisfy the condition $A \cup B = C \cup D$. The rule $C \Rightarrow D$ is **simply redundant** with respect to the rule $A \Rightarrow B$, if $C \supset A$.

Definition 2 Strict redundancy We consider two rules generated from itemsets X_i and X_j , respectively, such that $X_i \supset X_j$. Let $A \Rightarrow B$ and $C \Rightarrow D$ be rules satisfying $A \cup B = X_i, C \cup D = X_j$, and $C \supseteq A$. Then the rule $C \Rightarrow D$ is **strictly redundant** with respect to the rule $A \Rightarrow B$.

The concept of redundancy can be illustrated using the rules generated from the lattice (Fig. 49) as listed in Table 4. Based on Definitions 1 and 2, if a rule \mathcal{R}_1 is *simple* or *strict redundant* with respect to another rule \mathcal{R}_2 , then \mathcal{R}_2 is said to *simple* or *strict dominate* \mathcal{R}_1 , respectively. In Table 4, the rule $(X \Rightarrow YZ)$ *simple dominates* the rules $(XY \Rightarrow Z)$ and $(XZ \Rightarrow Y)$ (Def. 1). In Table 4, the rule $(X \Rightarrow YZ)$ *strict dominates* rules $(X \Rightarrow Y)$ and $(X \Rightarrow Z)$ (Def. 2). In general, a rule may be dominated by several *dominating rules* and may in turn dominate several other *dominated rules*.

References

- Aggarwal, C.C., Yu, P.S.: A new approach to online generation of association rules. *IEEE Trans. Knowl. Data Eng.* **13**(4), 527–540 (2001)
- Agrawal, R., Srikant, R.: Fast algorithms for mining association rules in large databases. In: *VLDB '94 Proceedings of the 20th International Conference on Very Large Data Bases*, pp. 487–499. Morgan Kaufmann Publishers Inc. San Francisco, CA (1994)
- Borgelt, C.: Efficient implementations of Apriori, Eclat and FP-growth. <http://www.borgelt.net> (2017). Accessed Dec 2017
- Boulicaut, J.F., Jeudy, B.: Constraint-based data mining. In: Maimon, O., Rokach, L. (eds.) *Data Mining and Knowledge Discovery Handbook*, pp. 339–354. Springer, Berlin (2010)
- Cao, L., Li, J., Wang, C., Yu, P.S.: Efficient selection of globally optimal rules on large imbalanced data based on rule coverage relationship analysis. In: *SIAM International Conference on Data Mining*, pp. 216–224 (2013)
- Cao, L.: Combined mining: analyzing object and pattern relations for discovering and constructing complex yet actionable patterns. *WIREs Data Min. Knowl. Discov.* **3**(2), 140–155 (2013)
- Chaudhuri, S., Lee, H., Narasayya, V.R.: Variance aware optimization of parameterized queries. In: *SIGMOD Conference*, pp. 531–542 (2010)
- Cleveland, R.B., Cleveland, W.S., Mcrae, J.E., Terpenning, I.: STL: a seasonal-trend decomposition procedure based on loess. *J. Off. Stat.* **6**(1), 3–73 (1990)
- Couturier, O., Hamrouni, T., Yahia, S.B., Nguifo, E.M.: A scalable association rule visualization towards displaying large amounts of knowledge. In: *International Conference Information Visualization*, pp. 657–663 (2007)
- Duan, S., Thummala, V., Babu, S.: Tuning database configuration parameters with iTuned. *PVLDB* **2**(1), 1246–1257 (2009)
- PARAS/FIRE Home Page. <http://paras.cs.wpi.edu/> (2018). Accessed March 2018
- Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The WEKA data mining software: an update. *SIGKDD Explor. Newsl.* **11**(1), 10–18 (2009)
- Han, J., Pei, J., Yin, Y.: Mining frequent patterns without candidate generation. In: *SIGMOD '00 Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, vol. 29, pp. 1–12. ACM, New York, NY (2000)
- Hahsler, M., Chelluboina, S.: ARulesViz R package. <http://cran.r-project.org/web/packages/arulesViz/vignettes/arulesViz-1.jpg> (2017). Accessed Dec 2017
- Jeudy, B., Boulicaut, J.-F.: Using condensed representations for interactive association rule mining. In: *PKDD*, pp. 225–236 (2002)
- Kaya, M., Reda, A.: Online mining of fuzzy multidimensional weighted association rules. *Appl. Intell.* **29**(1), 13–34 (2008)
- Kubat, M., Hafez, A., Raghavan, V.V., Lekkala, J.R., Chen, W.K.: Itemset trees for targeted association querying. *IEEE Trans. Knowl. Data Eng.* **15**(6), 1522–1534 (2003)
- Leung, C.K.-S.: Constraint-based association rule mining. In: Wang, J. (ed.) *Encyclopedia of Data Warehousing and Mining*, pp. 307–312. Hershey, Information Science Reference (2009)
- Lin, X., Mukherji, A., Rundensteiner, E.A., Ruiz, C., Ward, M.O.: PARAS: a parameter space framework for online association mining. *PVLDB* **6**(3), 193–204 (2013)
- Lin, X., Mukherji, A., Rundensteiner, E.A., Ward, M.O.: SPIRE: supporting parameter-driven interactive rule mining and exploration. *PVLDB* **7**(13), 1653–1656 (2014)
- Liu, G., Suchitra, A., Zhang, H., Feng, M., Ng, S.-K., Wong, L.: AssocExplorer: an association rule visualization system for exploratory data analysis. In: *ACM SIGKDD Demo*, pp. 1536–1539 (2012)
- Lucchese, C., Orlando, S., Perego, R., Silvestri, F.: WebDocs: a real-life huge transac. dataset, FIMI (2004)
- Mukherji, A., Lin, X., Botaish, C.R., Whitehouse, J., Rundensteiner, E.A., Ward, M.O., Ruiz, C.: PARAS: interactive parameter space exploration for association rule mining. In: *ACM SIGMOD*, pp. 1017–1020 (2013)
- Mukherji, A., Lin, X., Whitehouse, J., Botaish, C.R., Rundensteiner, E.A., Ward, M.O.: FIRE: interactive visual support for parameter space-driven rule mining. In: *CIKM*, pp. 2447–2452 (2013)
- Qin, X., Ahsan, R., Lin, X., Rundensteiner, E.A., Ward, M.O.: iPARAS: incremental construction of parameter space for online association mining. In: *BigMine*, pp. 149–165 (2014)
- Qin, X., Ahsan, R., Lin, X., Rundensteiner, E.A., Ward, M.O.: Interactive temporal association analytics. In: *EDBT*, pp. 197–208 (2016)
- Qin, X., Kakar, T., Wunnava, S., Rundensteiner, E.A., Cao, L.: MARAS: signaling multi-drug adverse reactions. In: *ACM SIGKDD*, pp. 1615–1623 (2017)
- Shao, J., Yin, J., Liu, W., Cao, L.: Actionable combined high utility itemset mining. In: *AAAI*, pp. 4206–4207 (2015)

29. Tork, H.F.: Bike sharing dataset. <https://archive.ics.uci.edu/ml/datasets/Bike+Sharing+Dataset> (2017). Accessed Dec 2017
30. UCI Machine Learning Repository. <http://www.ics.uci.edu/~mlern/MLRepository.html> (2017). Accessed 17 March 2017
31. Wang, S., Cao, L.: Inferring implicit rules by learning explicit and hidden item dependency. In: IEEE TSMC, PP(99), pp. 1–12 (2017)
32. Ward, M.O.: A taxonomy of glyph placement strategies for multi-dimensional data visualization. *Inf. Vis.* **1**(3–4), 194–210 (2002)
33. Wong, P.-Y., Chan, T.-M., Wong, M.-H., Leung, K.-S.: Predicting approximate protein-DNA binding cores using association rule mining. In: IEEE ICDE, pp. 965–976 (2012)
34. Wu, T., Chen, Y., Han, J.: Association mining in large databases: a re-examination of its measures. In: PKDD, pp. 621–628 (2007)
35. XmdvTool Home Page. <http://davis.wpi.edu/~xmdv/> (2018). Accessed March 2018
36. Yang, D., Rundensteiner, E.A., Ward, M.O.: A shared execution strategy for multiple pattern mining requests over streaming data. *Proc. VLDB Endow.* **2**(1), 874–885 (2009)
37. Zaki, M.J., Hsiao, C.-J.: CHARM: an efficient algorithm for closed itemset mining. In: SIAM SDM (2002)
38. Zaki, M.J., Parthasarathy, S., Ogihara, M., Li, W.: New algorithms for fast discovery of association rules. In: SIG KDD, pp. 283–286 (1997)