

FIRE: Interactive Visual Support for Parameter Space-Driven Rule Mining*

Abhishek Mukherji, Xika Lin, Jason Whitehouse, Christopher R. Botaish,
Elke A. Rundensteiner and Matthew O. Ward
Computer Science Department, Worcester Polytechnic Institute
100 Institute Road, Worcester MA, USA.
mukherab|xikalj|whitehouse|cbotaish|rundenst|matt@wpi.edu

ABSTRACT

While significant strides have been made on efficient association rule mining, the usability of mining systems woefully lags behind. In particular, the usability of rule mining systems is limited by the lack of support for interactive exploration of the 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)* addresses the usability shortcoming. FIRE features innovative visual displays and effective interactions that enable analysts to conduct rule exploration at the speed of thought. Particularly, the parameter space view (PSpace) displays the distribution of rules produced for diverse parameter settings. This not only facilitates user parameter selection but also empowers analyst's to understand rule relationships in the parameter space context. Our user study with 22 subjects establishes the usability and effectiveness of the proposed features and interactions of FIRE using benchmark datasets. Overall, this research encompasses significant contributions at the intersection of data mining, knowledge management and visual analytics.

1. INTRODUCTION

1.1 Motivation

Mining of associations and correlations from huge data sets is critical for applications ranging from market basket analysis [2], bioinformatics to intrusion detection and web usage mining [17]. Existing rule mining algorithms [2, 10, 22] tend to be compute-intensive, rendering even their fast implementations [4] unfit for interactive analysis. Mining systems with delayed response times risk losing a user's attention and, more importantly, are often unacceptable in mission critical applications. Over the years, significant focus [1, 4, 15, 18, 22] has thus been placed on addressing the performance challenge. Recent experiments [15] using IBM Quest [2], webdocs [17] and other benchmark data sets demonstrate that the

*This patent pending work was partly supported by the National Science Foundation under grants IS-0812027, CCF-0811510 and IIS-1117139.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
CIKM'13, Oct. 27–Nov. 1, 2013, San Francisco, CA, USA.
Copyright 2013 ACM 978-1-4503-2263-8/13/10 ...\$15.00.
<http://dx.doi.org/10.1145/2505515.2505631>.

preprocess-once-query-many solutions [15, 18] can offer near real-time responsiveness due to preprocessing and indexing. This near real-time responsiveness lays the foundation for offering speedups sufficient for interactive rule exploration.

While significant strides have been made on performance, usability of rule mining systems has received little attention [3, 11]. In particular, the usability of these systems is limited due to the lack of support for interactive exploration of relationships among rule results produced with various parameter settings. Unfortunately, sense-making of rulesets extracted from a data set is not the focus of existing rule mining systems [3, 9, 11, 15, 18]. Thus, **usability** of rule mining systems continues to be the limiting bottleneck hindering their adoption for sense-making in real-world applications as further described below.

1.2 Challenges for Effective Rule Exploration

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

Lack of parameter recommendations. The distribution of frequent itemsets majorly differs from data set to data set. Yet analysts wanting to explore different data sets may not be aware of the appropriate parameter settings required to obtain the rulesets of interest from each and every data set. While too high parameter settings may result in no rules being discovered, if the parameters are set too low, then analysts may be faced with an overwhelmingly large number of rules being extracted. While some rule visualization techniques have been proposed [3, 9, 11], they do not provide a comprehensive parameter space-driven interactive paradigm for rule mining. Neither do they provide recommendations to guide analysts through appropriate parameter selection and refinement.

Lack of insights into rule relationships. Redundancy relationships [1] among rules provide analysts with insights about which rules are rich in significance or information content. For instance, those rules that cover other rules (rendering them redundant) based on their antecedents and/or consequents are typically considered more important than redundant rules. Beyond manual sifting, analysts using existing systems [3, 9, 11] cannot gain such insights into rule relationships.

Lack of pareto-optimal parameter selections. Finding the top-k rules from a dataset is a commonly sought after feature. Existing systems [3, 9, 11] can only extract top-k rules based on *one* parameter (either support or confidence) at a time. However, certain rules may have high support yet low confidence, and vice-versa. Thus, employing a 2-dimensional combination of support and confidence for top-k rule extraction is yet another useful feature that has the potential to improve the usability of interactive mining systems.

Lack of support for ruleset comparisons. For a data set of interest, analysts would like to discover parameter settings that produce rules deemed interesting by the analysts. Yet when using

existing systems [3, 9, 11], analysts must go about a tedious and time-consuming trial-and-error process of parameter selection interleaved with sifting through the extracted rules to discover genuinely interesting ones. Given two distinct parameter settings, support for evaluating the relationship between the rulesets extracted using these two settings can be tremendously useful in this context.

Insufficient support for interactive rule filtering. Interactive rule filtering is essential for analysts wanting to explore the rules containing the itemsets matching their interest, e.g., about a particular product or a stock ticker. Further, filtering is also useful in situations when even high parameter settings produce overwhelmingly large numbers of rules. Existing constraint-based mining solutions [5, 14] take the first important step towards usability by pushing constraints into the rule mining algorithms. Unfortunately, these solutions are ill-suited for interactive mining, as even the fastest among these current online mining algorithms [4] have unacceptably high response times for even moderately sized data sets.

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

Based on a novel parameter space-driven approach, we now introduce our proposed *Framework for Interactive Rule Exploration (FIRE)*. FIRE features innovative visual displays and interactions to enable analysts to conduct rule exploration at the speed of thought. Overall, this research work encompasses contributions at the intersection of data mining, knowledge management and visual analytics as described below.

- We propose a novel parameter space-driven rule exploration framework, called **FIRE**. FIRE abstracts rule exploration at two layers of abstractions, namely, the overall parameter space view (**PSpace**) and the detailed rule space view (**RSpace**). Linkage between PSpace and RSpace permits drill-down and ruleset comparison (Sec.2).

- The key innovation here is the **PSpace** view which 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; (d.) rule cardinality skyline. The **RSpace** view enables analysts to explore the detailed information about particular rulesets within the selected regions (Sec. 3).

- Our user study evaluates the usability and effectiveness of various proposed features of the FIRE framework over several benchmark datasets. The user study provides clear evidence that our proposed FIRE framework is efficient and effective in helping analysts to understand rule distribution over the parameter space by providing parameter recommendations and to gain rich insights into the rule relationships (Sec. 4).

2. INTERACTIVE RULE MINING MODEL

Here, we describe the parameter space model that forms the foundation for our interactive mining framework and then overview the FIRE framework.

2.1 Foundation: The PSpace Model

The core principle behind our interactive rule exploration framework is the *preprocess-once-query-many* (POQM) paradigm [1], i.e., in an offline step, (a.) extracting all rules from a dataset that

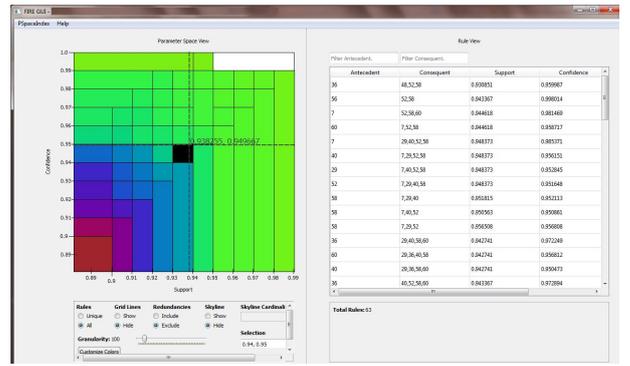


Figure 1: The FIRE Visualizer

satisfy a low primary support¹ and, then (b.) compactly indexing the large number of extracted rules for subsequent interactive rule exploration by analysts. This parameter space-driven approach that we adopt from [15] forms the foundation for the effective management and exploration of mined rulesets with near real-time responsiveness.

In the context of rule mining, the parameter space consists of a two-dimensional space of *support* and *confidence*. A *parametric location* is a point within this space. One key insight is that even for a huge number of rules, many rules may map to the same location and thus can be compactly indexed by their location.

Stable Region Abstractions. An important observation that drives our parameter space-driven approach is that for many real-world datasets several settings on the parameter space either contain no rules at all or vice versa the same set of rules may be valid across a large range of diverse parameter settings. In fact, the parameter space can be divided into several regions, which we henceforth call *stable regions*. The ruleset valid for any possible parametric location within a stable region remains unchanged, whereas rulesets valid for two locations not in the same stable region are guaranteed to be distinct. *Stable regions* form our *coarse granularity abstractions* for storing and managing rules. In an offline step, we partition the *parameter space* into a finite number of *non-overlapping stable regions* meeting the above properties.

Rule Redundancy Resolution. Redundancy relationships among rules can be leveraged to filter out redundant rules for presenting succinct query results to users. Two types of redundancies are defined in [1], namely, *simple* and *strict*. In [15] we designed algorithms that effectively precompute rule redundancies in the context of the parameter space. To facilitate interactive redundancy resolution, we abstract rules into stable regions and store compact rule redundancy information for the rules.

2.2 The FIRE Framework

Our FIRE framework proposed in this work manages the rulesets using the *FIRE index* that compactly stores stable regions, rules and their redundancy information. A rich variety of analytical queries can be processed over the FIRE index. The queries are broadly classified as rule mining, stable region and redundancy resolution queries [15].

The interactive user requests are supported via the FIRE visualizer (Fig. 1) that enables analysts to explore the stable region abstractions of the parameter space model and the corresponding rulesets with ease - thus supporting effective visual analytics. The framework is composed of two views, namely, the *PSpace* view (left of Fig. 1) and the *RSpace* view (right of Fig. 1). The *PSpace*

¹Assuming that analysts are not interested in rules with support lower than the *primary support* [1].

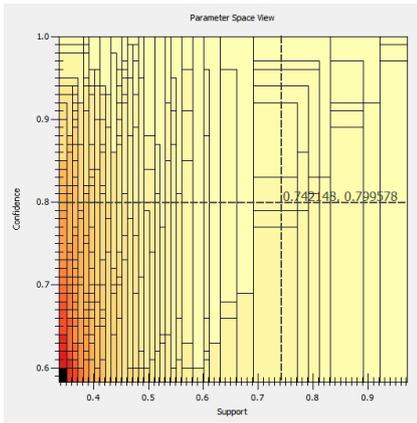


Figure 2: PSpace (All Rules)

view displays the overall distribution of rules within the space, facilitating parameter tuning and exploration at a higher level of abstraction. The *RSpace* view provides a detailed view of each rule including the itemsets in its antecedent and consequent together with its support and confidence values. The requests are passed from the visualizer to the FIRE Engine for efficient execution.

3. FIRE INTERACTIVE VISUALIZATION

Below we describe the visual interactions for the two views.

Overview of the PSpace View. FIRE provides analysts with a novel abstract view of the distribution of rules within the parameter space called the *PSpace* view. As depicted on the left hand side of Fig. 1, the *PSpace* view presents 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. 1 (left) depicting the rule distribution for the Chess data set and Fig. 2 depicting that for the Mushroom data set. Both are benchmark data sets taken from UCI Machine Learning Repository [19]. This offers an overview of the complete rule space driven by a parameter-centric perspective.

Stable Region Abstractions For Parameter Recommendations.

For a dataset with a sparse distribution of rules in the parameter space, often despite submitting several successive mining requests with distinct (minsupp,minconf) input parameter values, a rule miner may repeatedly return the same set of rules due to the sparse population of rules. When using an existing rule miner, the analyst must progress through a frustrating trial-and-error to finally get a new set of rules. When using the *PSpace*, the analyst can instead explore the parameter space by clicking through different regions and every time be guaranteed to receive a distinct ruleset for investigation. Thus, FIRE saves time and effort by recommending which next parameter settings will cause a difference in the output. In FIRE, analysts can navigate through regions by both indirectly entering (support, confidence) values in the control panel textbox (Fig. 1) or by directly clicking on the *PSpace* view.

Rich Insights into Region-wise Rule Cardinality. To provide rich insights into the density of rules within different regions, we visually mark regions with colors denoting cardinality. Fig. 1 (left) and Fig. 2 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 depicts high count. For this FIRE offers a variety of color palettes including variants of sequential, diverging and qualitative ramps [20].

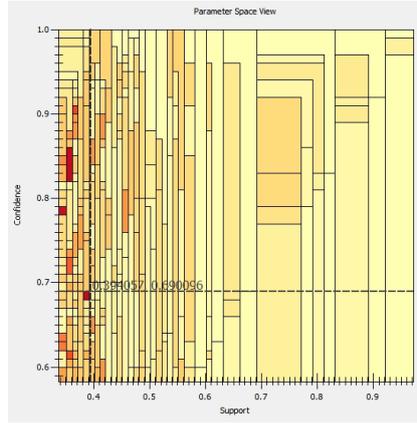


Figure 3: PSpace (Unique Rules)

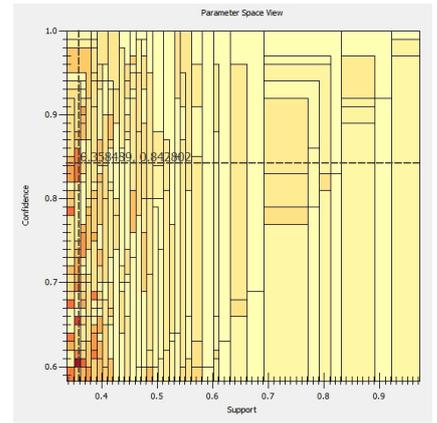


Figure 4: PSpace (Unique + Non-red.)

FIRE is able to display either all rules that appear in a region or only the rules unique to each region. For a dense dataset such as *Chess* or *Mushroom* [19], each parameter setting produces a huge number of rules. Suppose that an analyst changes the query input 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 discern, especially if the analyst has to manually compare the two rulesets to identify the difference. Here, a delta output of rules is desirable which can be achieved in FIRE simply by selecting *Unique* option. While Fig. 2 depicts an *All* rules view for Mushroom, Fig. 3 shows the *Unique* rules view of the same.

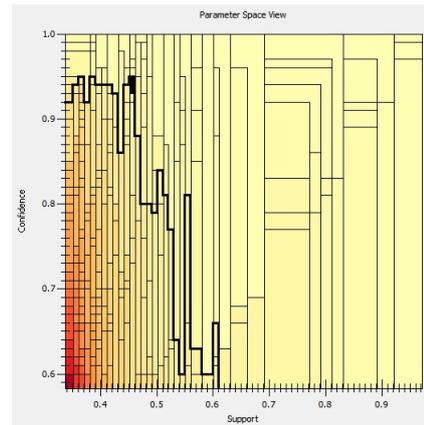


Figure 5: Rule Cardinality Skyline (>100 Rules)

Rule Redundancy Resolution. FIRE allows the analysts to optionally display only the non-redundant rules for a data set in the *PSpace* view. By excluding redundant rules, a succinct set of fewer rules is displayed for ease of analysis. In the context of the stable region abstractions interesting patterns can be observed when redundancies are excluded (Fig. 4) compared to when they are included (Fig. 3). In fact, any combination of unique/all and redundant/non-redundant rules can be displayed to observe different patterns at an overview level and further analyze them.

Rule Cardinality Skyline. Fig. 5 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. Using the skyline selection the analyst can input the desired cardinality in the *skyline*

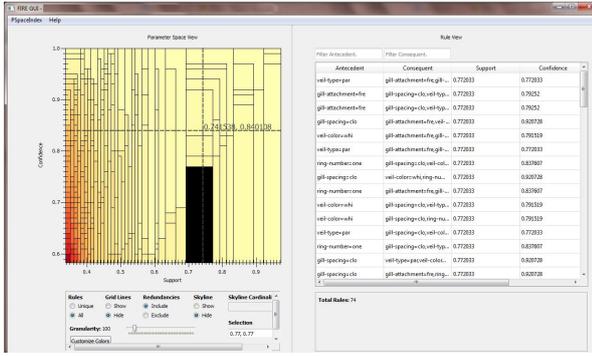


Figure 6: PSpace-RSpace Linkage

cardinality textbox (say, 100). The skyline drawn on the PSpace view marks for each support value (x-axis), the confidence value (y-axis) having ≥ 100 rules. As 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 among a range of support and confidence combinations that will all return the top 100 rules. Furthermore, the analyst can now quickly determine various observations about the data set. For instance, using the rule cardinality skyline in Fig. 5 we observe that no region contains ≥ 100 rules above support = 0.61.

Drill-down via PSpace-RSpace Linkage. In Fig. 6, when the analyst clicks on a single region (highlighted in *black*) the actual rules within that region can be viewed in the RSpace View via cross links between the two views. This supports rapid drill-down into individual rules while still maintaining the global context via the PSpace view.

Two Region Ruleset Comparison. Analysts can also select two regions at a time to compare their rulesets. In Fig. 7 comparing two stable regions facilitates the analysis of how the change in parameter settings affects the output. Region A is selected with a click (highlighted in *black*) and region B is selected with shift+click (highlighted in *grey*). 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 regions A and B have 3 rules in common, whereas region A has 71 and region B has only 2 unique rules, respectively.

Assisted Navigation through PSpace Abstractions. 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. 6. In our early user interviews, we found that analysts may not be comfortable initially in identifying the support and confidence of desired regions on the PSpace view. Therefore, we introduced the cursor position feature, namely, as the analyst moves the cursor over the PSpace, the current cursor position is displayed. In Fig. 6, the current cursor position is (0.741538, 0.840108).

Tabular RSpace View. Rules can be displayed using a traditional *tabular* RSpace view common to most mining tools as depicted on the right hand side of Fig. 6. This tabular view provides detailed information about each rule. The antecedents and the consequents of each rule are separated into two columns. The support and confidence values are also displayed for each rule. The total number of rules within the selected region on the PSpace view is displayed at the bottom of the RSpace table. In case of an overwhelmingly large number of rules being displayed in the RSpace view, the analyst can filter the rules based on the antecedent and/or

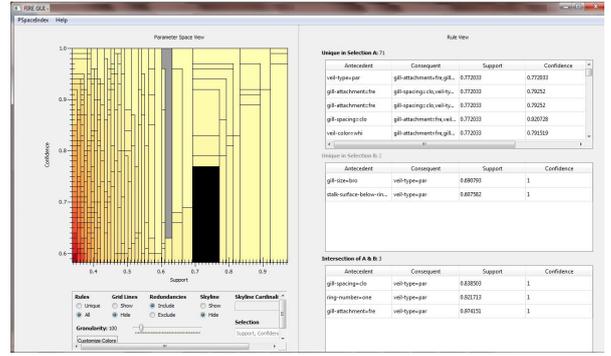


Figure 7: Comparing Two Regions

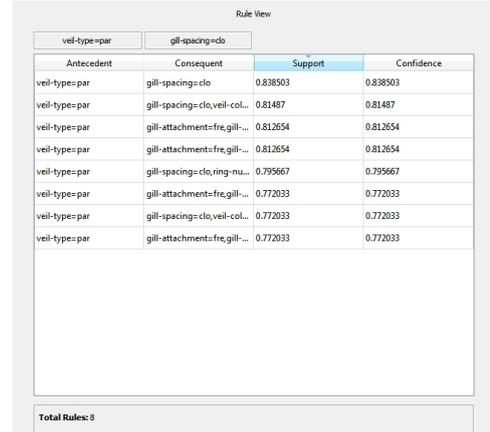


Figure 8: RSpace Filtering and Sorting

the consequent values using a conventional auto-fill control. In general this allows the analyst to determine which rules are prominent for a given attribute value. For example, in Fig. 8, 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. 6) satisfy the filter.

4. EVALUATION

4.1 Evaluation Methodology

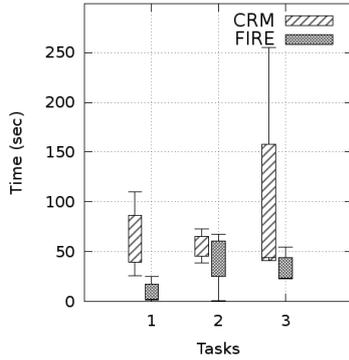
We conducted a user study with 22 subjects to evaluate usability and effectiveness of our FIRE framework as described below.

User Study Procedure. The overall process was as follows: First, we explain association rule mining principles and our visualizations to the subject. Then the subject performs a series of tests. The study on average took between 30 and 50 minutes per subject.

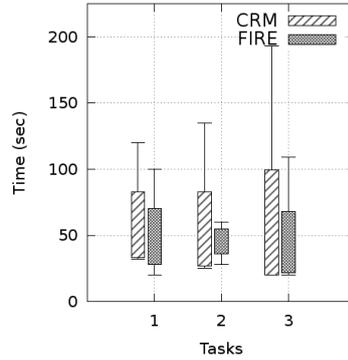
Tools Compared. Our user study compares our FIRE visualizer to the cached association rule miner (CRM). CRM is an association rule miner based on the APRIORI algorithm [2] but with instant response time due to cached rules and meta-knowledge. CRM provides users with a tabular view of rules and all functions offered by existing rule mining systems (e.g., WEKA [9]).

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 user for each task. For accuracy, we measured the % of correctly answered tasks by the subjects.

Datasets. We chose two real datasets from the UC Irvine Machine Learning Repository [19], namely, chess and mushroom. The chess dataset is derived from the game step. The mushroom dataset contains characteristics of various species of mushrooms.



(a) Mushroom



(b) Chess

Figure 9: Time Spent on Tasks 1, 2 and 3.

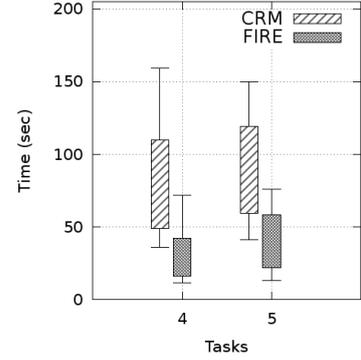
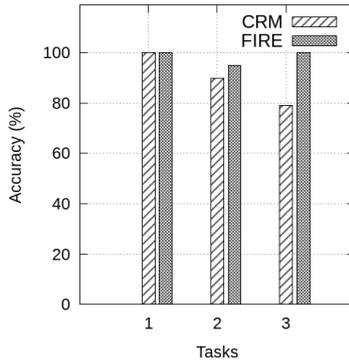
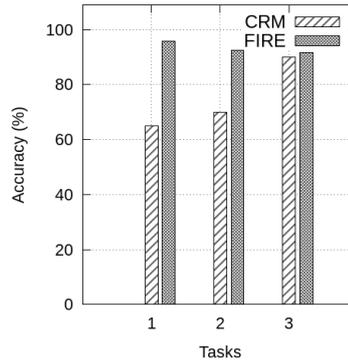


Figure 10: Time Spent on Tasks 4 and 5.



(a) Mushroom



(b) Chess

Figure 11: Accuracy of Tasks 1, 2 and 3.

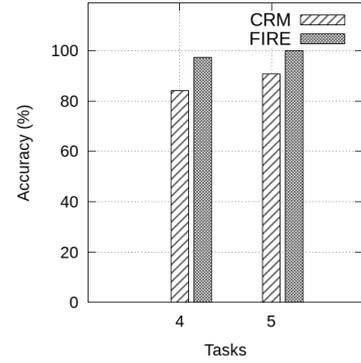


Figure 12: Accuracy of Tasks 4 and 5.

General Method. Each subject was asked to perform all five tasks (T1-T5) described in Section 4.2. To avoid carryover effects and learned knowledge about a dataset, we counter-balanced the order of tasks, datasets and tools. The specific task assignment is shown in Table 1. For example, subjects 1-11 performed T1-T3 on chess dataset using CRM, and T1-T3 on mushroom dataset using FIRE. On the other hand, subjects 12-22 performed T1-T3 on mushroom dataset using CRM and T1-T3 on chess dataset using FIRE. For T4 and T5, we switched questions and tools. For example, we asked subjects to find characteristics of edible mushrooms using CRM and of poisonous mushrooms with FIRE. Thus, no "pre-knowledge" problem will arise. In general, we avoid practice and fatigue effects by randomizing the order of tools and tasks. In these task assignments, no carryover problems arise, as each subject is only asked to finish a particular task on a given dataset using tools in a random order.

Subjects	CRM	FIRE
Subjects 1-11	T1/Chess T2/Chess T3/Chess	T1/Mushroom T2/Mushroom T3/Mushroom
Subjects 12-22	T1/Mushroom T2/Mushroom T3/Mushroom	T1/Chess T2/Chess T3/Chess

Table 1: Task Assignment in User Study.

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

Study Population. We performed the user study with a population of 22 subjects. They have either computer science, computer engi-

neering or mathematical sciences background. The user study was conducted on a one-to-one basis, i.e., tester to subject test.

4.2 Design of User Study

As shown in Table 2, in **stable region usage tests** we asked the subjects to perform three different tasks T1-T3 using different tools and data sets, such that each dataset was tested for each visualization in a random order. The three tasks are designed to verify the ability of the subjects to explore the parameter space, to utilize the stable region abstractions and to compare rulesets. In **filter/redundancy tests**, we asked subjects to first filter on the antecedent of the rules and then remove redundant rules for mushroom dataset. Some users used FIRE first and CRM next, and vice-versa. The goal was to test the ability of subjects to use filter and redundancy removal features. In the **skyline view tests** we asked the subjects to find top-k rules from the mushroom dataset alternatively with FIRE and CRM. The goal was to test if subjects can make use of the rule skyline cardinality.

4.3 Discussion of Results

Stable region usage tests. Subjects took much less time when working with FIRE, whereas they needed a significantly longer time using CRM (Fig. 9). This is because the tabular view does not provide any intuitive mechanism for subjects to accomplish the tasks. As shown in Figures 9.(a) and 11.(a), for test T1, subjects spent 9 secs on average using FIRE to get 100% accuracy while subjects used 62 secs on average with CRM to achieve the same accuracy. For T2, the minimum time spent was 2 secs using FIRE while using CRM required at least 26 secs. Yet FIRE was also better than CRM in accuracy (by 5%). For T3, while the maximum time spent with FIRE was only 55 secs, the maximum time spent

Tests	Tasks
Stable Region Usage Tests	T1 What are the most prominent rules by support and 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.
Filter/Redundancy Tests	T4 Find the most frequent characteristics of edible/poisonous mushrooms.
Skyline View Tests	T5 Find the parameter settings that produce top-k rules in the dataset, where k = 20, 50, or 100.

Table 2: Tasks Description for User Study.

with CRM reaches 255 secs. Accuracy-wise, the subjects using FIRE achieved 100% correctness while CRM only managed 80% correctness. Similarly, in Figures 9.(b) and 11.(b) for chess dataset, subjects took less time using FIRE than CRM to complete all three tasks. At the same time, they made less mistakes using FIRE than CRM. In particular, the accuracy of T1 using FIRE is 30% higher than the accuracy for CRM. This is because more than one rule exists that satisfies the question in the chess dataset. Subjects tend to omit some rules resulting in this low accuracy. In contrast, FIRE is able to reveal the answer with 1 or 2 clicks.

Filter/redundancy + skyline view tests. In Figures 10 and 12 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 secs on average with FIRE yet achieved near 100% accuracy for T4. The subjects using CRM, on the other hand, took 80 secs and reached only 84% accuracy.

In summary, averaged over all tasks, 92% of our subjects could perform the task correctly with FIRE while 82% of them could derive right answers with CRM. The results confirm that our FIRE technology is a win-win in terms of both efficiency and accuracy.

5. RELATED WORK

Parameter Space Exploration. Prior work has explored the space of parameters for handling parameterized database queries [6] and tuning database configuration parameters [8]. While most data mining queries are parameterized, selection of appropriate parameters is often left to the analyst. [21] aims at helping analysts understand the relationship among clusters produced with different parameter settings for density-based clusters. We instead explore the parameter space for data mining, in particular, for rule exploration. The closest to our proposed parameter space display is AssocExplorer [16] that proposes a scatterplot of rules on a 2-D space. However, they overlook the visual clutter problem that is common even if a moderate number of rules needs to be displayed. We tackle the clutter problem with our proposed stable region abstractions, zoom and granularity features.

Interactive Association Rule Mining. Hahsler et al. [3] present the R-extension package *arulesViz* which implements several visualization techniques to display individual rules. We instead focus on the higher-level of the overall space - not just visual display for single rules. Couturier et al. [7] propose an integrated framework covering both rule extraction and visualization of the mining process. It provides 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. Unlike some other work 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, 12, 13] only prestore intermediate values, namely, the frequent itemsets. More recently [15, 18] propose to store the final rule results instead, thus achieving near real-time responsiveness. This lays the foundation for offering speedup sufficient for interactive

rule exploration. However, sense-making of rulesets extracted is not the focus of existing rule mining systems [3, 9, 11, 15, 18].

6. CONCLUSION

In this work we designed, implemented and evaluated an innovative technology for interactive rule exploration called the **FIRE** framework. **FIRE** offers parameter recommendations and enhanced sense-making of rule relationships. Particularly, we propose the PSpace views that supplemented with innovative interactions enables analysts to effectively conduct rule exploration. We conducted a user study that demonstrates the usability and effectiveness of the FIRE framework using benchmark datasets.

7. REFERENCES

- [1] C. C. Aggarwal and P. S. Yu. A new approach to online generation of association rules. *IEEE Trans. Knowl. Data Eng.*, 13(4):527–540, 2001.
- [2] R. Agrawal and R. Srikant. Fast algorithms for mining association rules in large databases. In *VLDB*, pages 487–499, 1994.
- [3] *Arulesviz* r package. <http://cran.r-project.org/web/packages/arulesViz/vignettes/arulesViz.pdf>, May 2013.
- [4] C. Borgelt. Efficient implementations of apriori, eclat and fp-growth. <http://www.borgelt.net>, August 2013.
- [5] J.-F. Boulicaut and B. Jedy. Constraint-based data mining. In *Data Mining and Knowledge Discovery Handbook*, pages 339–354, 2010.
- [6] S. Chaudhuri, H. Lee, and V. R. Narasayya. Variance aware optimization of parameterized queries. In *SIGMOD Conference*, pages 531–542, 2010.
- [7] O. Couturier, T. Hamrouni, S. B. Yahia, and E. M. Nguifo. A scalable association rule visualization towards displaying large amounts of knowledge. In *International Conference Information Visualisation*, pages 657–663, 2007.
- [8] S. Duan, V. Thummala, and S. Babu. Tuning database configuration parameters with ituned. *PVLDB*, 2(1):1246–1257, 2009.
- [9] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. The weka data mining software: an update. *SIGKDD Explor. Newsl.*, 11(1):10–18, 2009.
- [10] J. Han, J. Pei, and Y. Yin. Mining frequent patterns without candidate generation. In *SIGMOD*, pages 1–12, 2000.
- [11] B. Jedy and J.-F. Boulicaut. Using condensed representations for interactive association rule mining. In *PKDD*, pages 225–236, 2002.
- [12] M. Kaya and R. Alhajj. Online mining of fuzzy multidimensional weighted association rules. *Applied Intelligence*, 29:13–34, 2008.
- [13] M. Kubat, A. Hafez, V. V. Raghavan, J. R. Leekala, and W. K. Chen. Itemset trees for targeted association querying. *IEEE Trans. Knowl. Data Eng.*, 15(6):1522–1534, 2003.
- [14] C. K.-S. Leung. Constraint-based association rule mining. In *Encyclopedia of Data Warehousing and Mining*, pages 307–312, 2009.
- [15] X. Lin, A. Mukherji, E. A. Rundensteiner, C. Ruiz, and M. O. Ward. PARAS: A parameter space framework for online association mining. In *VLDB*, volume 6 (3), Aug 2013.
- [16] G. Liu, A. Suchitra, H. Zhang, M. Feng, S.-K. Ng, and L. Wong. Assocexplorer: an association rule visualization system for exploratory data analysis. In *KDD Demo.*, pages 1536–1539, 2012.
- [17] C. Lucchese, S. Orlando, R. Perego, and F. Silvestri. Webdocs: a real-life huge transac. dataset. In *FIMI*, 2004.
- [18] A. Mukherji, X. Lin, C. R. Botaish, J. Whitehouse, E. A. Rundensteiner, M. O. Ward, and C. Ruiz. Paras: interactive parameter space exploration for association rule mining. In *SIGMOD*, pages 1017–1020, June 2013.
- [19] UCI machine learning repository. <http://www.ics.uci.edu/~mllearn/MLRepository.html>, July 2013.
- [20] Xmdvtool home page. <http://davis.wpi.edu/~xmdv/>, July 2013.
- [21] D. Yang, E. A. Rundensteiner, and M. O. Ward. A shared execution strategy for multiple pattern mining requests over streaming data. *Proc. VLDB Endow.*, 2(1):874–885, Aug. 2009.
- [22] M. J. Zaki, S. Parthasarathy, M. Ogihara, and W. Li. New algorithms for fast discovery of association rules. In *SIGKDD*, Aug 1997.