SPIRE: Supporting Parameter-Driven Interactive Rule Mining and Exploration*

Xika Lin[1], Abhishek Mukherji[2]†, Elke A. Rundensteiner[1] and Matthew O. Ward[1]
Computer Science Department, Worcester Polytechnic Institute, 100 Institute Road, Worcester MA, USA.[1]
Samsung Research America - Silicon Valley, 1732 North 1st Street, San Jose CA, USA.[2]
xika|rundenst|matt@wpi.edu[1] a.mukherji@samsung.com[2]

ABSTRACT

We demonstrate our SPIRE technology for supporting interactive mining of both positive and negative rules at the speed of thought. It is often misleading to learn only about positive rules, yet extremely revealing to find strongly supported negative rules. Key technical contributions of SPIRE including *region-wise abstractions of rules*, *positive-negative rule relationship analysis*, *rule redundancy management* and *rule visualization* supporting novel *exploratory queries* will be showcased. The audience can interactively explore complex rule relationships in a visual manner, such as comparing negative rules with their positive counterparts, that would otherwise take prohibitive time. Overall, our SPIRE system provides data analysts with rich insights into rules and rule relationships while significantly reducing manual effort and time investment required.

1. INTRODUCTION

1.1 Motivation

Mining of associations and correlations from huge data sets is critical for applications ranging from market basket analysis [2] and bioinformatics [11] to intrusion detection and web usage mining [7]. Traditionally, association rule mining algorithms [2, 5, 13] were developed to find only positive associations between items. Yet positive rule mining based on the support-confidence framework has been criticized for its misleading results [3]. A negative association rule (e.g., Coke → ¬ Pepsi), on the other hand, considers the presence (coke) as well as absence (pepsi) of an item and mines for negative implications between items, meaning, if customers buy coke products, they probably won't buy pepsi products.

Yet mining for negative association rules, which are considered complement to positive rule mining, received surprising little attention due to the challenge in discovering these rules. Computation for both positive and negative rules requires the examination of an exponentially large search space and thus renders their approaches unfit for interactive analysis. Mining systems with huge response delays risk losing a user's attention and, more importantly, are impractical in time critical applications.

Besides having unacceptable high response times, a mixture of positive and negative rules as result poses another challenge. Analysts have to manually compare and contrast the positive rules with negative ones to identify genuine rules. For instance, while a positive rule that has a reasonable high support and confidence could be considered a significant rule, its opposite rules (negative ones) that might be more valuable may be hidden in the large ruleset. Thus they may go unnoticed due to possibly lower support or confidence value. For discovering such opposite rules, the analyst has to go about a tedious and time-consuming manual search for all of its opposite rules from the ruleset. The task of analyzing and exploring the relationship between negative and positive association rules, as part of sense-making of the mined rules, requires much manual effort with little or no help from existing systems.

Visually identifying similarities or differences among rules based on their attributes is yet another desired feature that existing systems lack. Beyond manual sifting, analysts using existing systems cannot quickly gain an intuitive insight about mined rules. Yet an interactive visualization solution fitting both positive and negative association rules is desired for advanced sense-making of rulesets.

Therefore, an interactive data mining system, capable of not only answering mining queries for rules efficiently on huge datasets, but also providing positive-negative rule relationship exploration together with support for visualizing the result at near real-time speed is important for decision making applications as motivated by the example below.

Motivating Example: The study of space shuttle landing control is important in aeronautics. The analyst may want to study the association between landing conditions (such as stability, wind, visibility, etc.) and the choice of manual or automatic control from the shuttle landing control dataset [4]. A rule miner could be used to generate rules for determining the conditions under which manual control would be preferable to auto landing of the spacecraft. An example rule would be R1 = {Stability:stab → Class:manual} with support = 27% and confidence = 53%. However, its opposite rule such as R2 = {¬ Stability:stab → Class:manual} with support = 6% and confidence = 100% won't be generated by such a traditional rule miner. Even though R2 might be detected by a negative rule miner using an extreme low threshold, the computation might take hours or possibly days to finish. Plus a mass of insignificant rules would be generated as well due to low threshold. This conflicts with the near real-time responsiveness required for effective interactive analysis. Even if R2 is generated, the analyst probably won't notice either rule R2 or the relation between R1 and R2.

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

†This work was done when Abhishek was a graduate student at the Computer Science Department of Worcester Polytechnic Institute.

Proceedings of the VLDB Endowment, Vol. 7, No. 13
Copyright 2014 VLDB Endowment 2150-8097/14/08.

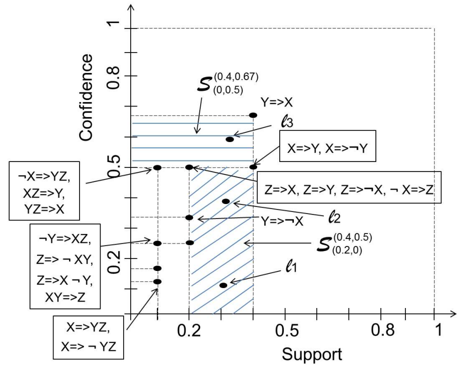


Figure 1: SPIRE Parameter Space

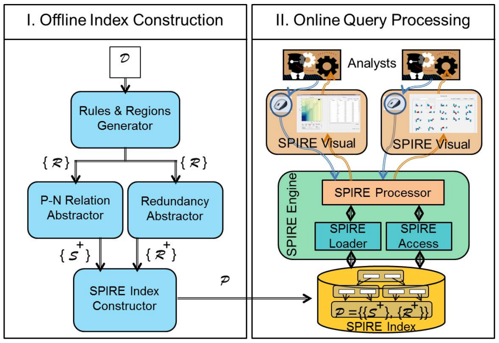


Figure 2: SPIRE Framework

Then, the analyst may miss R2 (a rule with 100% confidence) that is more beneficial than R1. We will demonstrate that our SPIRE system can instantaneously provide useful insights to the analyst about the rules relationships between positive and negative rules.

The SPIRE Contributions: In this demonstration, we present **SPIRE (Support for Parameter-driven Interactive Rule Mining and Exploration)** system that overcomes the above challenges by offering real-time responsiveness and enhanced sense-making of mined rules. Over the past 15 years the XMDV team at WPI, composed of visualization, HCI and database experts, supported by a series of six NSF grants, has developed a freeware visual tool suite XmdvTool [12] to facilitate interactive data exploration. We currently focus on extending XmdvTool to support interactive parameter space exploration for mining association rules.

Through the simple yet effective SPIRE interactive visualizations, the audience will be able to experience the benefits of using the SPIRE model over using other existing mining technologies [2, 5, 13]. First, for benchmark datasets such as Chess [4] and Mushroom [4] datasets, SPIRE demonstrates 6 orders of magnitude or more improvement in response times over the state-of-the-art techniques[1] [2, 5, 13]. In particular, our experiments confirm that even for large datasets (e.g. webdocs [7] $\sim$ 1.5 GB) SPIRE responds in less than a second while state-of-the-art technologies [2, 5, 13] tend to take several hours to compute the results. Second, SPIRE provides a competitive advantage to the analysts by linking positive rules with their negative rules. Thus, they help the analysts to make sense of rule relationships and extract the most desirable associations. Third, SPIRE enables analysts to perform real-time in-depth investigation of association rules via a rich set of novel *exploratory mining queries*. We demonstrate the powerful interactive capabilities of SPIRE using several real datasets.

2. THE SPIRE FRAMEWORK

As depicted in Fig. 2, the SPIRE framework consists of two phases (a) offline index construction and (b) online query processing. In the offline stage, to construct SPIRE index, *Rule & Regions Generator* generates all rules and construct regions. *P-N Relation Abstractor* summarizes positive-negative rule relationships to enable users to navigate through the rule relationships. *Redundancy Abstractor* captures rule redundancy relation such that, if desired by the user, non-redundant rules can be efficiently generated upon demand. *Index Constructor* creates our *SPIRE Index*.

The online query processing phase is performed by *SPIRE Engine*. The interactive user requests are supported via the *SPIRE visualizer* allowing analysts to interact with three innovative views of rules, to submit mining requests and navigate through responses in a visual manner. The requests are passed to the *SPIRE Processor* for efficient execution. The *SPIRE Access* module offers the API for accessing the *SPIRE index*. The *index* compactly stores regions along with their association rules, relation abstractions and redundancy abstractions as explained below.

3. KEY INNOVATIONS OF SPIRE SYSTEM

The SPIRE system encompasses several innovations that form the foundation for effective exploration of rules as explained below.

3.1 SPIRE Region-Wise Abstractions

The core principle behind our interactive rule exploration framework is the *preprocess-once-query-many* paradigm [1]. SPIRE extracts all positive and negative rules satisfying a low primary threshold and then compactly indexes these extracted rules for subsequent interactive rule exploration by analysts in an offline phase. We extend the notion of the PSpace [6, 8, 9] to managing, retrieving and exploring both *positive* and *negative* association rules extracted from a dataset. In the context of rule mining, the parameter space consists of an n-dimensional space, with each dimension representing an interestingness measure. For simplicity, we henceforth work with a two-dimensional PSpace using *support* and *confidence* as dimensions. A key observation is that even for a huge number of rules, many rules may map to the same location. In particular, positive and negative rules tend to share the same set of parameter settings. In the parameter space example shown in Fig. 1, negative rules $\{\neg Y \rightarrow XZ\}$, $\{Z \rightarrow \neg XY\}$, $\{Z \rightarrow X\neg Y\}$ and positive rule $\{XY \rightarrow Z\}$ co-locate in (0.1,0.25). Thus the generated rules can be compactly indexed by their location. An important discovery is that for many realworld 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. Thus, the parameter space can be divided into several regions, which we call stable regions.

The ruleset valid for any possible parametric location within a stable region remains unchanged ($\mathcal{L}1$ and $\mathcal{L}2$ in Fig.1), whereas rulesets valid for two locations not in the same stable region are guaranteed to be distinct ($\mathcal{L}2$ and $\mathcal{L}3$ in Fig.1). Stable regions form our coarse granularity abstractions for storing and managing both positive and negative rules. In an *offline* step, we partition the parameter space into a finite number of non-overlapping stable regions and we index rules by stable regions in our rule base.

3.2 Positive-Negative Rule Relation Analysis

[1] modified to mine both positive and negative rules.

Rule relationship management empowers analysts' understanding of the connection between positive and negative rules. A negative rule is a rule that contains at least the negation of one item (i.e. a rule for which its antecedent or consequent can be formed by a conjunction of presence or absence of terms). An example would be: *Wind=tail* $\wedge$ *Visibility=yes* $\wedge$ $\neg$ *Error=XL* $\rightarrow$ *Landing=manual control*. Each (positive or negative) rule has its opposite rule(s). For a positive rule, opposite rules are the rules composed of the same items, but with at least one negative item. For a negative rule, the opposite rule is the rule composed of same items without any negative item. E.g., the opposite rules of rule R1 = {X$\rightarrow$Y} are R2 = {X$\rightarrow$ $\neg$Y}, R3 = {$\neg$X$\rightarrow$Y} and R4 = {$\neg$X$\rightarrow$ $\neg$Y}. There is a one-to-many relation between the positive rules and negative rules. A positive rule R = ((A_1:A_n) $\Rightarrow$ (C_1:C_m)) has $2^{n+m} - 1$ negative rules while a negative rule has only one positive rule. This leads to the challenge that we must design a corresponding strategy to maintain and query all opposite rules for every rule.

An important observation that drives our location-aware approach is that within the set of opposite rules for rule R, many rules share the same or similar (i.e., same support, different confident or vise versa) location. By summarizing and indexing the opposite rules by their locations, SPIRE provides a solution for offline relationship abstraction and efficient online relationship reconstruction.

3.3 Positive and Negative Rule Redundancy

Redundancy relationships among rules, which so far in the literature have only been defined for positive rules, are used to filter out redundant rules for presenting succinct query results to the user [1]. In particular, two types of redundancies exist among rules, namely, *simple* and *strict*. For the first time we apply the redundancy relationship to both positive and negative rules. In Table 1, rule X$\Rightarrow$Y$\neg$ Z *simple dominates* the rules XY$\Rightarrow$$\neg$Z and X$\neg$ Z$\Rightarrow$Y and *strict dominates* the rules X$\Rightarrow$Y and X$\Rightarrow$$\neg$Z. In general, a rule may be dominated by several *dominating rules* and may in turn dominate several other *dominated rules*. Moreover, we observe that the property of redundancy being a *query-time phenomenon* from PARAS model [6] can be naturally applied to this new negative rule context. Thus, rules cannot be tagged as redundant and discarded apriori.

Rule	Support	Confidence
X$\Rightarrow$ Y $\neg$ Z	S(X $\cup$ Y $\cup$ $\neg$ Z) = 0.3	S(X $\cup$ Y $\cup$ $\neg$ Z) / S(X) = 0.375
X Y$\Rightarrow$ $\neg$ Z	S(X $\cup$ Y $\cup$ $\neg$ Z) = 0.3	S(X $\cup$ Y $\cup$ $\neg$ Z) / S(X $\cup$ Y) = 0.75
X $\neg$ Z$\Rightarrow$ Y	S(X $\cup$Y $\cup$ $\neg$ Z) = 0.3	S(X $\cup$ Y $\cup$ $\neg$ Z) / S(X $\cup$ $\neg$ Z) = 0.5
X$\Rightarrow$ Y	S(X $\cup$ Y) = 0.4	S(X $\cup$ Y) / S(X) = 0.5
X$\Rightarrow$ $\neg$ Z	S(X $\cup$ $\neg$ Z) = 0.6	S(X $\cup$ $\neg$ Z) / S(X) = 0.75

Table 1: Redundancy among Positive and Negative Rules

We examine how such rule redundancies can be identified. Rule R = ((A_1:A_n) $\Rightarrow$ (C_1:C_m)) is *simple dominated* by all rules with template (((A_1:A_n)-(A_v:A_w)) $\Rightarrow$ ((A_v:A_w) $\cup$ (C_1:C_m))), whereas rule R is *strict dominated* by all rules with template (((A_1:A_n)-(A_t:A_u)) $\Rightarrow$ ((A_t:A_u) $\cup$ (C_1:C_{m+e}))). Maintaining all dominating rules for every rule is memory and compute-intensive. Fortunately, we have discovered that the surprisingly compact representation of rule redundancies from PARAS model [6] also applies in the context of our negative rule model. It is sufficient to compare each candidate rule R with only two dominating locations instead of the large number of dominating rules. Thus, for rule R = ((A_1:A_n) $\Rightarrow$ (C_1:C_m)) in the output ruleset, while state-of-the-art online redundancy resolution takes $\mathcal{O}(2^n)$ time, our newly proposed online redundancy resolution solution takes $\mathcal{O}(1)$ time by performing a $\mathcal{O}$(n) time offline redundancy abstraction step.

3.4 SPIRE Exploratory Queries

Our SPIRE framework supports a rich variety of analytical queries broadly classified as (a.) Positive and Negative Rule Mining (RM), (b.) Positive-Negative Rule Relation Exploration (RE) and (c.) Rule Cardinality (RC) Queries. Below, we briefly present sample queries in each category.

Positive and/or Negative Rule Mining (RM) Queries: For a given dataset $\mathcal{D}$, query Q1 finds the set of rules that satisfy query parameters (*minsupp*, *minconf*). The *WITH Rule Included* clause provides users the choice to include *Positive*, *Negative* or *All* rules into output. The *WITH Redundancy Elimination* clause gives users the option to output only non-redundant rules. In case of an overwhelmingly large number of rules valid for parameter settings (*minsupp*,*minconf*), SPIRE offers analysts the choice of viewing only the non-redundant rules for the setting. The RM query takes $\mathcal{O}$(N) time for N rules valid for input (*minsupp*, *minconf*).

```
Q1:  OUTPUT RuleSet $\{\mathcal{R}\}^{(minsupp,minconf)}$
FROM $\mathcal{D}$
HAVING minsupport=minsupp, minconfidence=minconf
WITH Rule Included = Positive/Negative/All;
WITH Redundancy Elimination = T/F;
```

Positive-Negative Rule Relation Exploration (RE) Queries: This new query class explores positive-negative relationships (see Section 3.2) in the context of association rules. The analyst selects one rule $\mathcal{R}_i$ from the rules generated from Q1, Query Q2 obtains the set of all opposite rules for the chosen rules. The RE query incurs $\mathcal{O}$(N) time complexity for the N rules returned.

```
Q2:  OUTPUT Opposite Rules $\mathcal{R}_i.\{\mathcal{R}\}^{\leftrightarrow}$
FROM $\mathcal{P}$
HAVING rule = $\mathcal{R}_i$;
```

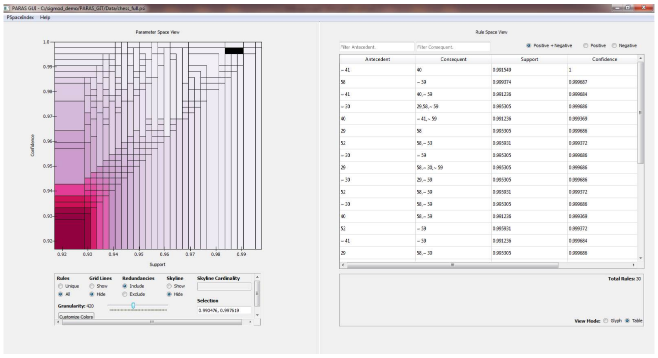
Rule Cardinality (RC) Queries: For the stable region containing input (minsupp,minconf), Query Q3 obtains the number of rules within that region. The query identifies the region and the cardinality of rules within that region through a constant time $\mathcal{O}$(1) lookup over the SPIRE index, while also allowing analysts to specify the preference of redundancy elimination and rule types. The RC query gives analysts a sense of the population of rules within the region.

```
Q3:  OUTPUT Rule Cardinality $|\{\mathcal{R}\}|^{(minsupp,minconf)}$
FROM $\mathcal{P}$
HAVING minsupport=minsupp, minconfidence=minconf
WITH Rule Included = Positive/Negative/All;
WITH Redundancy Elimination = T/F;
```

4. SPIRE DEMONSTRATION

Our demonstration illustrates how analysts can interact with our SPIRE visualizer. Datasets from several domains will be used, including the Shuttle Landing Control and the Chess datasets [4].

Region & Rule Exploration: SPIRE provides analysts with an abstract view of the distribution of rules within the parameter space. As depicted on the left hand side (LHS) of Fig. 3, the *Region view* presents rules in a two dimensional plot of the 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. One example is shown in Fig. 3 depicting the chess dataset. This offers an overview of the complete rule space driven by a parameter-centric perspective. The audience can alternate among the *POSITIVE*, *NEGATIVE* and *POSITIVE+NEGATIVE* radio button options. With *POSITIVE*, the audience will be able to view only the positive rules within each region. With *NEGATIVE*, only the negative rules will be shown. *POSITIVE+NEGATIVE* is



Rules: ○ Unique ● All
Grid Lines: ○ Show ● Hide
Redundancies: ● Include ○ Exclude
Skyline: ○ Show ● Hide
Skyline Cardinality:
Granularity: 420
Selection: 0.990476, 0.997619
Customize Colors

Rule Space View

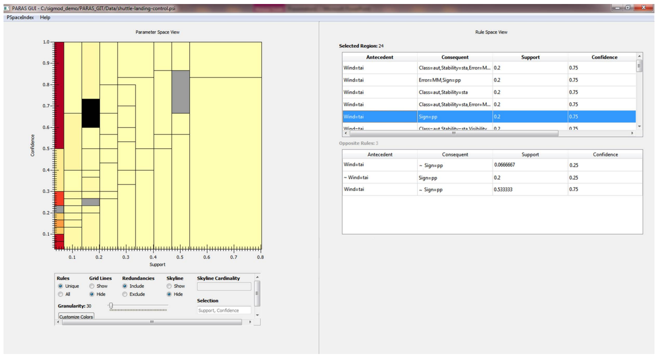
Filter Antecedent. | Filter Consequent. | ● Positive + Negative ○ Positive ○ Negative

Antecedent	Consequent	Support	Confidence
~ 41	40	0.991549	1
58	~ 59	0.999374	0.999687
~ 41	40, ~ 59	0.991236	0.999684
~ 30	29,58, ~ 59	0.995305	0.999686
40	~ 41, ~ 59	0.991236	0.999369
29	58	0.995305	0.999686
52	58, ~ 53	0.995931	0.999372
~ 30	~ 59	0.995305	0.999686
29	58, ~ 30, ~ 59	0.995305	0.999686
~ 30	29, ~ 59	0.995305	0.999686
52	58, ~ 59	0.995931	0.999372
~ 30	58, ~ 59	0.995305	0.999686
40	58, ~ 59	0.991236	0.999369
52	~ 59	0.995931	0.999372
~ 41	~ 59	0.991236	0.999684
29	58, ~ 30	0.995305	0.999686

Total Rules: 30

View Mode: ○ Glyph ● Table

Figure 3: Positive and Negative Rule Mining



Rules: ● Unique ○ All
Grid Lines: ○ Show ● Hide
Redundancies: ● Include ○ Exclude
Skyline: ○ Show ● Hide
Skyline Cardinality:
Granularity: 30
Selection: Support, Confidence
Customize Colors

Rule Space View

Selected Region: 24

Antecedent	Consequent	Support	Confidence
Wind=tai	Class=aut,Stability=sta,Error=M...	0.2	0.75
Wind=tai	Error=MM,Sign=pp	0.2	0.75
Wind=tai	Class=aut,Stability=sta	0.2	0.75
Wind=tai	Class=aut,Stability=sta,Error=M...	0.2	0.75
Wind=tai	Sign=pp	0.2	0.75
Wind=tai	Class=aut,Stability=sta,Visibility...	0.2	0.75

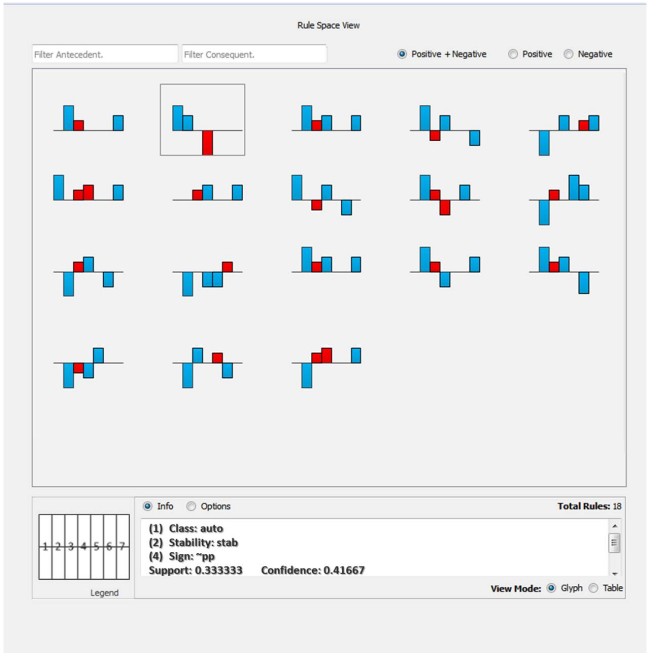
Opposite Rules: 3

Antecedent	Consequent	Support	Confidence
Wind=tai	~ Sign=pp	0.0666667	0.25
~ Wind=tai	Sign=pp	0.2	0.25
Wind=tai	~ Sign=pp	0.533333	0.75

Figure 4: Positive-Negative Linkage

the default setting that displays all rules within each region. The *Rule View* (RHS of Fig. 3) lists the rules valid within the selected region via cross links between the views. When the analyst clicks on a region (in black), a list of 30 rules are returned instantaneously in the Rule View in Fig. 3.

Exploration via Positive-Negative Rule Linkage: To provide rich insights into the positive-negative rule relationships, we provide explicit linkage analysis support between positive and negative rules. When the analyst clicks on a single positive rule, multiple regions containing its opposite rules (negative ones) are highlighted. The *Rule view* then will present a comparative display of opposite rules along with their parameters. In Fig. 4, the analyst clicks on a rule {Wind:tail → Sign:pp} in the selected region (in black), its opposite rules are then displayed in the box below. The regions in which those opposite rules fall are highlighted (in grey). Similarly, when the analyst selects a negative rule, its opposite rule (positive one) is shown and the related region is highlighted. Analyst can then compare and contrast the results or further explore to determine which rule is genuinely interesting.



Rule Space View

Filter Antecedent. | Filter Consequent. | ● Positive + Negative ○ Positive ○ Negative

● Info ○ Options

Total Rules: 18

(1) Class: auto
(2) Stability: stab
(4) Sign: ~pp
Support: 0.333333 Confidence: 0.41667

View Mode: ● Glyph ○ Table

Figure 5: Profile Glyph Representation of Rules

The Profile Glyph View: Our profile glyph view (Fig. 5) helps analysts to visually comprehend similarities or differences between the rules being displayed. However, this task is difficult to accomplish using only the tabular textual view due to the overload of text information. Beyond the straightforward tabular view described above, we thus designed a novel glyph view for graphically representing both positive and negative rules to facilitate efficient visual analysis of rulesets. We adopt and adapt profile glyph representation [10] to visualize both positive and negative rules. Fig. 5 depicts a profile glyph representation of rules from the shuttle landing control dataset. There are 7 slots corresponding to 7 different attributes, where an upward (downward) bar for an attribute indicates a positive (negative) value for that item and the height of the bar represents the value of that attribute. The antecedent is represented in red and the consequent is represented in blue. The detailed information about the glyph is shown underneath by hovering and clicking the cursor. The selected glyph in Fig. 5 represents the association rule: {¬Sign:pp → Class:auto and Stability:stab} with support=0.33 and confidence=0.42. Users can further filter the glyphs using a rich variety of filters, including content filter (antecedent/consequent) or type filter (positive/negative/all).

Conclusion: In summary, our demonstration will give the audience a rich insight of rules while significantly reducing manual effort and time investment required.

5. 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'94*, pages 487–499, 1994.

[3] M.-L. Antonie and O. R. Zaïane. Mining positive and negative association rules: An approach for confined rules. In *PKDD*, pages 27–38, 2004.

[4] A. Asuncion and D. Newman. UCI machine learning repository. http://www.ics.uci.edu/ mlearn/MLRepository.html, November 2007.

[5] J. Han, J. Pei, and Y. Yin. Mining frequent patterns without candidate generation. In *SIGMOD*, pages 1–12, 2000.

[6] X. Lin, A. Mukherji, E. A. Rundensteiner, C. Ruiz, and M. O. Ward. Paras: A parameter space framework for online association mining. *PVLDB*, 6(3):193–204, 2013.

[7] C. Lucchese, S. Orlando, R. Perego, and F. Silvestri. Webdocs: a real-life huge transactional dataset. In *FIMI'04*, 2004.

[8] 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 Conference*, pages 1017–1020, 2013.

[9] A. Mukherji, X. Lin, J. Whitehouse, C. R. Botaish, E. A. Rundensteiner, and M. O. Ward. Fire: interactive visual support for parameter space-driven rule mining. In *CIKM*, pages 2447–2452, 2013.

[10] M. O. Ward. A taxonomy of glyph placement strategies for multidimensional data visualization. *Information Visualization*, 1(3-4):194–210, 2002.

[11] P.-Y. Wong, T.-M. Chan, M.-H. Wong, and K.-S. Leung. Predicting approximate protein-dna binding cores using association rule mining. In *ICDE*, pages 965–976, 2012.

[12] Xmdvtool home page. http://davis.wpi.edu/ xmdv/, March 2014.

[13] M. J. Zaki, S. Parthasarathy, M. Ogihara, and W. Li. New algorithms for fast discovery of association rules. In *SIG KDD*, pages 283–286, Aug 1997.