Exploratory Data Analysis Using Alternating Covers of Rules and Exceptions

S. Saikia*, G. Shroff*, P. Agarwal*, A. Srinivasan+, A. Pandey*, G. Anand*
sarmimala.saikia, gautam.shroff, puneet.a, aditeya.pandey, gaurangi.anand@tcs.com; ashwin.srinivasan@iiitd.ac.in
*TCS Research, +Indraprastha Institute of Information Technology Delhi

ABSTRACT

During exploratory data analysis in a variety of domains we have found it useful to summarize the available data via sets of rules and their exceptions. Often, a practitioner is interested in exploring more than one such summary before settling on any single explanation. It is important for each summary to be succinct, cover a significant fraction of the data as well as highlight rules enjoying high confidence, support and lift. It is often difficult to achieve all these objectives simultaneously, and retain a meaningful dialogue with the user. To help achieve these goals, we have found it useful to combine some existing tools for data analysis: association rule-mining to find broad patterns in the data; clustering to group these patterns into similar sets; exception-mining to find deviations from the patterns; and a coverage-based extraction of explanations for the data. We demonstrate how a combination of these methods can be used for exploratory data analysis on both synthetic and industrial data sets for which we derived interesting insights.

1. INTRODUCTION

During exploratory analysis we often seek the causes or effects of specific situations, such as the kind of consumers who often use a particular product feature, the situations in which a vehicle heats up, or how an engine behaves when under heavy load. In such cases rule-based explanations are easy to understand and therefore desirable. Practitioners often want:

Alternatives: To automatically identify alternative descriptions for a subset of data; (or at least, different patterns that roughly describe the same subset of data);

Comprehensibility: To ensure that the descriptions found are meaningful, given what is already known;

Explanations: To extract from the descriptions, one or more combinations that can adequately "explain" the data;

Deviations: To understand outliers to these explanations.

With rules as a representation language, most coverage-based methods do not allow a user access to alternative descriptions. Instead, a greedy procedure is usually adopted, in which the next best rule is found, the corresponding data explained ("covered") is removed from further consideration, and the process is repeated. This results in a single explanation of the data. So, while they satisfy the "Explanation" requirement by constructing one explanation for the data, they typically do not present any solutions to finding alternatives or deviations. Comprehensibility may follow if the rules do not represent overly complex concepts.[1]

Although association rule mining [1] is particularly a widely used approach for finding patterns in data, by itself it will not satisfy any of the requirements of a practitioner, as described above, since it is simply aimed at finding patterns that exist in the data. Nevertheless, in conjunction with other techniques, some useful headway can be made. For example, clustering of association rules has been proposed to group similar sets of rules [7, 8, 11, 16, 18, 20], thus addressing the requirement of "Alternatives" above. Exception mining in conjunction with association rules has been proposed as a way of identifying comprehensible association rules and deviations from them [4, 5, 12, 17, 20].

It would appear that a combination of the methods described may result in a tool that could satisfy each of the requirements listed. In this paper we describe a tool of such a kind. ACRE (**A**lternating **C**overs of **R**ules and **E**xceptions), uses a combination of association-rule mining, clustering, and exception-mining to allow us to address the "Alternatives", "Comprehensibility" and "Deviations" requirement. In addition, it allows multiple rule-sets to be extracted, using a coverage-based approach (the "Explanation" need).

The basic procedure is as follows. ACRE first computes a rule set via association rule mining. Many of these rules usually overlap with each other, i.e., they cover many of the same data instances. ACRE clusters the rules based on their mutual overlap. Each cluster of rules thus roughly covers the same subset of the data. Whenever an explanation is sought for the entire dataset, a rule is selected in turn from each cluster until no further rules are required, or possible. By altering the choice of the rule chosen from each cluster, ACRE is able to provide alternative explanations for the same dataset. For each rule, ACRE is also able to identify exceptions in the data. These data instances can then be analyzed by ACRE in the same manner as before, namely using association rule-mining, clustering, explanation and, if needed, further levels of exception-mining.

From the user's viewpoint, we expect ACRE to be used

The 20th International Conference on Management of Data (COMAD), 17th-19st Dec, 2014 at Hyderabad, India.

[1]The work of Srinivasan et al [15] is an example of a covering procedure that also highlights deviations.

as follows. The user identifies a data set for exploratory analysis. Data analysis is selection of rules iteratively from clusters found by ACRE, and examination of exceptions until a satisfactory set of patterns are found.

In the sections that follow we summarize the related work on clustering of association rules as well as exception mining and then describe ACRE. We present our results qualitatively using our previous technique [14] for visualizing interactively, rules and exceptions as well as quantitatively in terms of F1-measure, both for ACRE as well as standard rule-based classifier (CN2, [3]).

2. RELATED WORK

Rule Summarization: Pruning and grouping association rules using clustering was introduced in [18]. A rule cover is a subset of rules that covers almost all data instances as the original set. In [16] a normalized distance metric (different from ours) is used to cluster association rules along with self organizing feature maps. These two approaches are similar to our approach, however, they do not augment the rules with exceptions. Rule-templates [6] can be used to select interesting rules from a larger set, based on user-defined criteria. [11] prunes a rule set by removing the insignificant ones, and then finds a special subset called Direct Setting (DS) rules to form a summary; however, DS-rules do not include rule-exceptions as we do. A technique similar to ACRE was used in [19] for summarizing a collection of frequent itemsets, using K representatives by clustering them to create a pattern profile representing sets of similar frequent itemsets. [8] introduces a technique for clustering association rules using a geometric-based algorithm. However, both these approaches filter rules based on the pattern of the clustered frequent itemsets, rather than coverage.

Exception Mining: Deviation analysis was used in [12] to identify interesting exceptions and explore reliable ones; [5] used information theory measures and evaluated exceptions based on the common sense rules, negative association rules were used in [4] to find exceptions and [17] used an autonomous probabilistic model to find out exceptions and common sense rule pairs with high confidence. However, mining of exceptions, and pruning of association rules have always been separate areas of research. We have combined both these techniques so as to derive a hierarchical summary that alternates between small sets of covering rules and their exceptions to provide a succinct summary for a desired consequent of interest. An approach similar to ours has also been used in [20] to find useful association rules and exceptional association rules for each of them, however, they do not cluster rules or exceptions to offer choices as we do. Finally, [13, 10] find 'unexpected' rules that are exceptions to *user-supplied* existing knowledge about the domain.

3. ACRE ALGORITHM

Each data instance is called a transaction that contains one or more items from a set of items $I = \{i_1, i_2 ...., i_n\}$: For example each survey response is a transaction, in which items are the customer's responses to each question asked. In multi-sensor data, each time step is a transaction with the individual values of different sensors forming items, after suitable discretization. A subset of I is referred to as an *itemset*; frequent itemsets are those that occur more often than others. Each frequent itemset, say $\{X, y\}$, may form a rule r for a pre-determined 'consequent of interest' (COI) y, with the subset of items (X) as antecedent, i.e., $X \rightarrow y$.

The *support of a rule* $S(r)$ is the percentage of transactions that contain all items in $(X \cup y)$. The *confidence of a rule* $C(r)$ is $P(y|X)$. Further, *lift* of a rule is a measure of its interestingness and is the ratio of its confidence and the probability of consequent, i.e., $L(r) = P(y|X)/P(y)$.

Since confidence is not usually equal to one, we also attempt to find patterns in the subset of transactions satisfying the antecedents of a rule. Such rules, i.e., $X, Z \rightarrow \neg y$, are called *exceptions* of the main rule $X \rightarrow y$.

The *coverage of a rule* is $\rho(r) = P(X \cup y)/P(y)$, indicating the percentage of transactions where the rule is satisfied, out of those that contain the consequent of interest y. However, rules can overlap in the data, so the coverage of multiple rules may be far less than the sum of their coverages.

Given a set of rules $R = \{r_1, r_2, ..., r_N\}$ having common consequent (y), we define *Rule cover* $R_{co} = \{r_1, r_2, ..., r_k\}$ as a subset of R, which cover almost the same set of transactions as covered by R. We quantify the *degree of overlap* O_{ij} between two rules $\{r_i, r_j\}$ as $O_{ij} = \frac{S(r_i \bigcap r_j)}{min(S(r_i), S(r_j))}$. A *distance measure* d_{ij} between a pair of rules $\{r_i, r_j\}$ is defined as $d_{ij} = \frac{1}{(O_{ij}+\kappa)}$ ($\kappa = .01$ is a small constant).

ACRE: consists of data processing steps that are performed for data understanding at two levels, to generate alternative yet 'coverage equivalent' rule-exception covers.

Data Processing:

Step-1: Rule Generation, we generate a set of rules R for a COI with support $> \tau_s$, and confidence $> \tau_p$ using frequent itemsets derived by PFP-growth algorithm [9].

Step-2: Rule Cover: We find a rule cover from R, such that the cumulative coverage of the rule cover R_{co} is at-least τ_r%. For this we scan the rules in R in descending order of support, and add them to rule cover R_{co} until τ_r% of the transactions containing COI are covered, or top-K rules are included in R_{co}, whichever comes first.

Step-3: Rule Clustering: The rules in R_{co} are clustered based on overlap using the distance measure d_{ij} using DBSCAN (optimizing parameters using gradient descent).

Multilevel Data Understanding:

At **Level 1**, steps 1-3 are performed on the entire set of transactions, yielding a set of rules R. Using an interactive rule visualization such as VARC [14], the user *chooses one rule from each cluster*. At **Level-2**, we repeat steps 1-3 for the set of transactions covered by the antecedents of each rule in R, however, this time we use $\neg y$ as the COI. This yields a set of exceptions for every rule in R. Note that the confidence threshold used for mining exceptions for rule $r : X \rightarrow y$ is $\tau_e = (100 + \Delta c - C(r))$, where Δc is called the confidence *gap*: For example, consider using a gap of 20% - if r has 85% confidence, 15% of the time we anyway expect $\neg y$ when X is true, so we would consider any exception implying $\neg y$ as significant if it had a confidence of even 35%.

4. RESULTS AND EVALUATION

We distinguish between two ways of using ACRE: in **batch mode**, at each level, we choose the highest support rule from each cluster, in **interactive mode**, the user *interactively chooses alternative rules from each cluster (at each level)* thereby allowing many equivalent alternating rule-exception covers to be examined. We expect batch mode to be more useful for data sets that have relatively few items. In higher

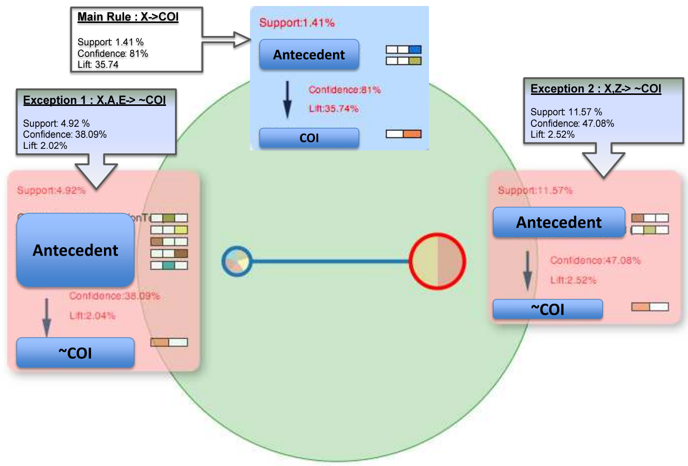

Figure 1: VARC depiction of a Sensor-I rule

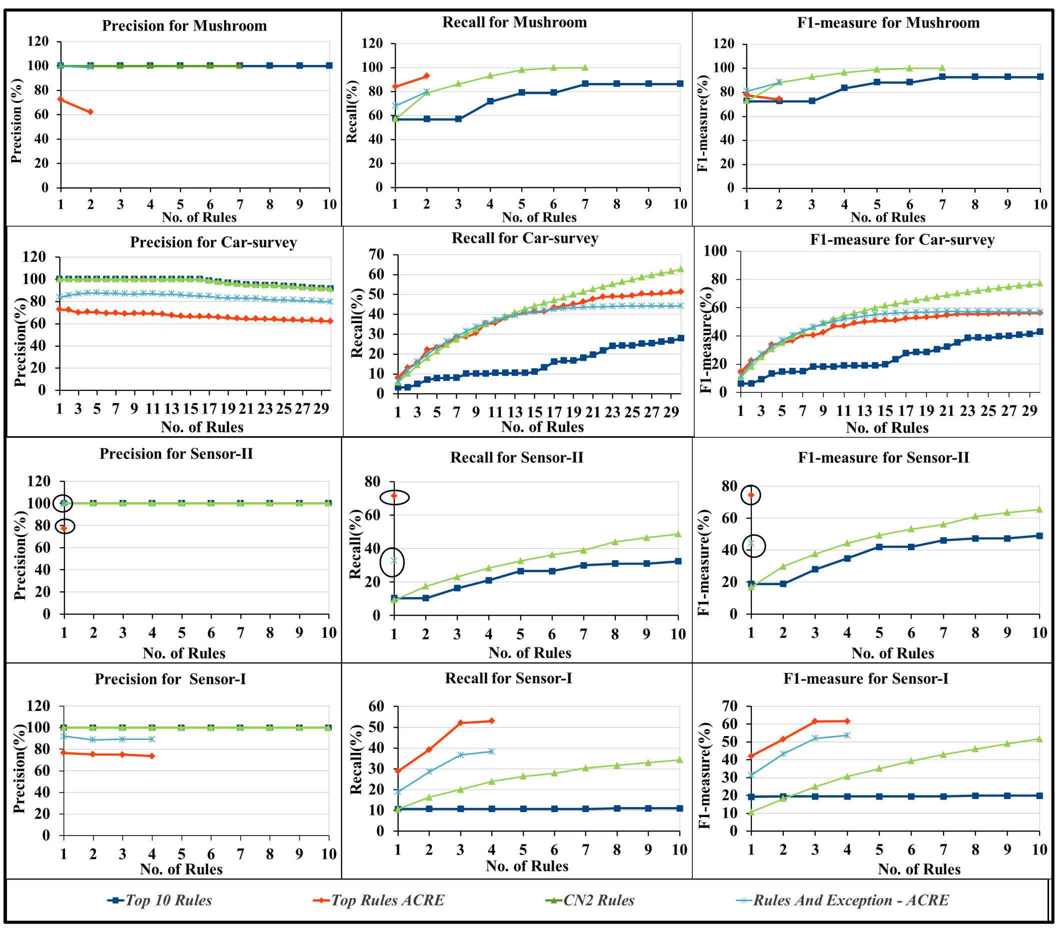

Figure 2: Precision, Recall and F1-measure graph for Mushroom, Car-Survey, Sensor-I, and Sensor-II datasets

dimensional spaces, an interactive mode is expected to be more useful because one is more likely to find relatively disjoint rules that nevertheless overlap in the data. However, in such cases frequent itemset mining itself has performance challenges (which, to our knowledge, remains an open problem). Here we present the results for batch mode on relatively narrow data.

We used the standard public Mushroom dataset [2] (poisonous vs non-poisonous mushrooms), as well as three real-life datasets: Car-survey, which captures customer driving patterns as well as usage of special vehicle features, and two vehicular sensor datasets: Sensor-I and Sensor-II. The Car-survey dataset has 1,153 transactions with 17 attributes while Sensor-I has 1,603 transactions with 18 attributes and Sensor-II has 54,565 transactions with 14 attributes.

ACRE requires input parameters such as minimum support threshold τ_s, minimum confidence threshold τ_p, coverage percentage τ_r, top-K rules K, and Δc. We used τ_p as 70% in general, unless the class-probability of the COI is greater, in which case $\tau_p = COI_{classProbability} + 10$. For top rules $\tau_r = 90\%$ and $K = 200$ in case if the required coverage is not achieved. However, for discovering exceptions we set $\tau_r = 75\%$ and $\Delta c = 20\%$.

Figure 1 depicts one of the rules discovered for Sensor-I data using the VARC visualization [14]. Our real-life datasets are proprietary, so the actual items are masked. Using 15 car-survey rules we described 40% of customers who used a particular enhanced mode based on their driving. Four Sensor-I rules described the effect of highly loading an engine, and one Sensor-II rule (with four exceptions) causally explained high-temperature situations.

Even though our goal is data-understanding rather than classification accuracy; still we can use F1-measure for quantitative evaluation from an IR perspective, i.e., balancing precision vs. recall of the consequent of interest. Figure 2 shows the F1-measure obtained by ACRE, both for top-level rules alone as well as rules+exceptions. These are compared with the top association rules as well as rules obtained using CN2 [3]. (Rule-sets are ordered similarly for a fair comparison, i.e., the best n rules in terms of F1-measure are chosen.)

Notice that ACRE has been always as good as CN2 (for Mushroom and Car-survey), at least for the succinct explanation it provides. Further, ACRE provides a markedly better F1-measure than CN2 for Sensor-I and Sensor-II. Next, notice how exceptions improve precision while reducing coverage somewhat. Thus, if coverage is more important top-level ACRE rules can be used without exceptions if their precision is deemed adequate. At the same time, their exceptions are always available for a deeper explanation that also improves precision, albeit affecting recall to some extent.

5. CONCLUSIONS

We have described ACRE, a technique for computing a set of rules and their exceptions that achieves high coverage of a desired consequent of interest as well as reasonable precision. ACRE clusters association rules at two levels to minimize inter-rule overlaps. Experimental results show that ACRE provides a succinct explanation with an adequately higher F1-measure, and is found to be as good or better than CN2 especially on our real-life data. Further, ACRE naturally enables qualitatively alternative yet quantitatively equivalent rule-exception covers by allowing for choosing different rule sets from each cluster of rules, at each level, thus allowing for more interactive and effective use in practice.

6. REFERENCES

[1] R. Agrawal, T. Imieliński, and A. Swami. Mining association rules between sets of items in large databases. In *SIGMOD 1993*.

[2] C. Blake and C. J. Merz. UCI repository of machine learning databases. 1998.

[3] P. Clark and T. Niblett. The CN2 induction algorithm. *Machine learning*, 1989.

[4] O. Daly and D. Taniar. Exception rules mining based on negative association rules. In *ICCSA 2004*.

[5] F. Hussain, H. Liu, E. Suzuki, and H. Lu. Exception rule mining with a relative interestingness measure. In *Knowledge Discovery and Data Mining*. Springer Berlin Heidelberg, 2000.

[6] M. Klemettinen, H. Mannila, P. Ronkainen, H. Toivonen, and A. I. Verkamo. Finding interesting rules from large sets of discovered association rules. In *CIKM 1994*.

[7] S. Kotsiantis and D. Kanellopoulos. Association rules mining: A recent overview. *GESTS Intl Trans on Computer Science and Engineering*, 2006.

[8] B. Lent, A. Swami, and J. Widom. Clustering association rules. In *ICDE 1997*.

[9] H. Li, Y. Wang, D. Zhang, M. Zhang, and E. Y. Chang. Pfp: Parallel fp-growth for query recommendation. In *RecSys 2008*.

[10] B. Liu, W. Hsu, and S. Chen. Using general impressions to analyze discovered classification rules. *KDD 1997*.

[11] B. Liu, W. Hsu, and Y. Ma. Pruning and summarizing the discovered associations. *KDD 1999*.

[12] H. Liu, H. Lu, L. Feng, and F. Hussain. In *Methodologies for Knowledge Discovery and Data Mining*. Springer Berlin Heidelberg, 1999.

[13] B. Padmanabhan and A. Tuzhilin. A belief-driven method for discovering unexpected patterns. *KDD 1998*.

[14] G. Sharma, G. Shroff, A. Pandey, P. Agarwal, and A. Srinivasan. Interactively visualizing summaries of rules and exceptions. In *EuroVis Workshop on Visual Analytics*, 2014.

[15] A. Srinivasan, S. Muggleton, and M. Bain. Distinguishing exceptions from noise in non-monotonic learning. In *Proc of 2nd Inductive Logic Programming Workshop*, 1992.

[16] A. Strehl, G. K. Gupta, and J. Ghosh. Distance based clustering of association rules. In *ANNIE 1999*.

[17] E. Suzuki. Autonomous discovery of reliable exception rules. *KDD 1997*.

[18] H. Toivonen, M. Klemettinen, P. Ronkainen, K. Htnen, and H. Mannila. Pruning and grouping discovered association rules. In *ECML 1995*.

[19] X. Yan, H. Cheng, J. Han, and D. Xin. Summarizing itemset patterns: A profile-based approach. *KDD 2005*.

[20] A. Zhou, L. Wei, and F. Yu. Effective discovery of exception class association rules. *Journal of Computer Science and Technology*, 2002.