Small is Beautiful: Discovering the Minimal Set of Unexpected Patterns

Balaji Padmanabhan
The Wharton School, University of Pennsylvania
1310 Steinberg-Dietrich Hall
Philadelphia, PA 19104
Tel: +1-(215)573-9646
balaji@wharton.upenn.edu

Alexander Tuzhilin
Stern School of Business, New York University
44 West 4th Street
New York, NY 10012
Tel: +1-(212)998-0832
atuzhili@stern.nyu.edu

ABSTRACT
A drawback of most traditional data mining methods is that they do not leverage prior knowledge of users. In many business settings, managers and analysts have significant intuition based on several years of experience. In prior work [11, 12] we proposed methods that could discover unexpected patterns in data by using this domain knowledge in a systematic manner. In this paper we continue our focus on discovering unexpected patterns and propose new methods for discovering a *minimal set* of unexpected patterns that discover orders of magnitude fewer patterns and yet retain most of the truly interesting ones. We demonstrate the strengths of this approach experimentally using a case study application in a marketing domain.

1. INTRODUCTION
A well-known criticism of many rule discovery algorithms in data mining is that they generate too many patterns, many of which are obvious or irrelevant. It stands to reason that more effective methods are needed to discover fewer and more relevant patterns from data and KDD researchers have addressed this issue extensively. One way to approach this problem is by focusing on discovering *unexpected* patterns [4, 5, 6, 7, 11, 12, 13, 14, 17, 18], where unexpectedness of discovered patterns is usually defined relative to a system of prior expectations. In particular, we proposed in our prior research [11, 12] a characterization of unexpectedness of a discovered pattern relative to the system of prior beliefs and developed efficient algorithms for the discovery of these unexpected patterns.

Although these algorithms generate significantly fewer and more relevant patterns, still many of the generated unexpected patterns are redundant in the sense that they can be derived from other discovered unexpected patterns. Therefore, this paper focuses on *minimality of unexpected patterns* and on efficient algorithms that discover such minimal patterns. The power of the proposed approach lies in combining two independent concepts of unexpectedness and minimality of a set of patterns into one integrated concept that provides for the discovery of small but important sets of interesting patterns.

The concept of minimality has been studied in AI for a long time and more recently in KDD. In particular in an early influential work [9], Mitchell addresses the problem of learning generalizations of a set of objects and presents a unifying approach to the problem of generalizing knowledge by viewing the generalization task as a search problem. In the context of discovering a minimal set of rules in data mining, the approach presented in [9] has the limitation that in most cases it may not be possible to have training examples that are classified into *known* generalizations. Therefore, rather than *learning* these generalization relationships among different objects, it is necessary to *define* them. Recent characterizations of various notions of minimality in the KDD literature take this approach and we describe them below.

In the KDD literature [2, 3, 8, 15, 16, 20] provide alternate approaches to characterizing a minimal set of discovered rules. In particular, [2] presents an approach that finds the "most interesting rules", defined as rules that lie on a support and confidence frontier. Further, [2] proves that these rules necessarily contain the strongest rules discovered using several objective criteria other than just confidence and support.

In [16] several heuristics for pruning large numbers of association rules has been proposed. One of these heuristics prunes out certain refinements of rules, thus, hinting at the concept of minimality of a set of rules. However, [16] focuses exclusively on heuristics that prune redundant rules from a discovered set of rules and does not explore the concept of minimality formally, nor proposes any algorithms for discovering a minimal set of patterns.

In [8] a technique is presented to prune and then summarize an already discovered set of association rules. In particular, [8] defines the concept of direction-setting rules and demonstrates how non-direction-setting rules can be inferred from them. Therefore, the set of direction-setting rules constitutes a set of rules that are "minimal" in some sense. This work is related to [16] in the sense that certain rule refinements are pruned out in the approach in [8] and therefore, they are not direction-setting. However, the method presented in [8] is different from [16] and

KDD 2000, Boston, MA USA

from our approach in the sense that not all refined rules are non-direction setting according to [8]. Moreover, [8] focuses on pruning already discovered rules and does not address the issue of direct discovery of minimal sets.

An approach to eliminating redundant association rules is presented in [20]. In particular, [20] introduces a concept of the "structural cover" for association rules and presents post-processing algorithms to find the structural cover. In this paper we, present an alternative formal characterization of the minimal set of patterns that corresponds to structural covers of [20] for the association rules but is also broader and applicable to more general classes of rules. Moreover, [20] focuses on pruning already discovered rules and does not address the issue of direct discovery of minimal sets.

Finally, the work of [3] and [15] is also related to the problem of discovering minimal sets of rules. In particular, [3] and [15] provide methods for eliminating rules such that the support and/or confidence values of these rules are not unexpected with respect to the support and confidence values of previously discovered rules. However, this work is only marginally related to our approach because we focus on a more general definition of minimality that does not directly depend on confidence and support of discovered rules. In this paper we present a new approach for characterizing minimality of a set of unexpected patterns and present efficient methods to discover minimal unexpected patterns.

In Section 2 we present a few preliminaries regarding unexpectedness, followed by a characterization of minimality of unexpected patterns in Section 3. We then present two algorithms (one in detail and a sketch of the second, for lack of space) for discovering minimal unexpected patterns in Section 4. We present experimental results and discussion in Section 5.

2. PRELIMINARIES - UNEXPECTEDNESS

We define an atomic condition to be a proposition of the form $value_1 \leq attribute \leq value_2$ for ordered attributes and *attribute* = *value* for unordered attributes where *value*, $value_1$, $value_2$ belong to the set of distinct values taken by *attribute* in dataset *D*. In this paper we consider rules and beliefs defined as extended association rules of the form $X \rightarrow A$, where *X* is the conjunction of atomic conditions (an itemset) and *A* is an atomic condition.

We follow the definition of unexpectedness from [11] and define the rule $A \rightarrow B$ to be *unexpected* with respect to the belief $X \rightarrow Y$ on dataset *D* if the following conditions hold:

(a) *B AND Y* |= FALSE, i.e., *B* and *Y* logically contradict each other.

(b) *A AND X* holds on a statistically large subset of tuples in *D*[1]. We use the term "*intersection of a rule with respect to a belief*" to refer to this subset. This intersection defines the subset of tuples in *D* in which the belief and the rule are both "applicable" in the sense that the antecedents of the belief and the rule are both true on all the tuples in this subset.

(c) The rule $A, X \rightarrow B$ holds (with the same level of threshold support and confidence). Since condition (a) constrains *B* and *Y* to logically contradict each other, it logically follows that the rule $A, X \rightarrow \neg Y$ holds.

A key assumption in this definition, motivated in [10, 11], is that of the *monotonicity of beliefs*. In particular, if we have a belief $Y \rightarrow B$ that we expect to hold on a dataset *D*, then monotonicity assumes the belief should also be expected to hold on any statistically large subset of *D*.

Given the definition of unexpectedness, [11] presents algorithm ZoomUR that discovers all the unexpected rules with respect to a set of beliefs. In the first phase of ZoomUR, ZoominUR discovers all unexpected patterns that are *refinements* to any belief. More specifically, given any belief $X \rightarrow Y$, ZoominUR discovers all unexpected rules of the form $X, A \rightarrow B$ such that *B AND Y* |= FALSE. We refer to such rules as "unexpected refinements". In the second phase of ZoomUR, starting from all the unexpected refinements, ZoomoutUR discovers more general rules (*generalizations*) that are also unexpected. As demonstrated in [10, 11], this approach generated far fewer and more interesting patterns than traditional approaches.

3. MINIMAL SET OF PATTERNS

Though ZoomUR discovers only unexpected rules and also far fewer rules than Apriori [1],[2] it still discovers large numbers of rules many of which are redundant in the sense that they can be inferred from other discovered rules under the monotonicity assumption stated in Section 2. In this sense, some of the discovered unexpected patterns are *expected* with respect to other discovered patterns and, thus, can and should be eliminated. For example, consider belief *diaper* → *beer* and two unexpected patterns *diaper, weekday* → *not_beer* and *diaper, weekday, male* → *not_beer*. Then the second unexpected pattern can be inferred from the first one under the monotonicity assumption. To address this issue, we formally characterize minimality of a set of unexpected patterns based on the monotonicity assumption. In order to do this, we first need to define inference of one rule from another under monotonicity.

3.1 Inference Under Monotonicity Assumption

Before introducing minimal rules, we need to define formally which rules can be inferred to hold on a dataset due to the monotonicity assumption.

Definition. Rule $(A \rightarrow B) \models_M (C \rightarrow D)$ if
1. $C \models A$, and
2. $D = B$. □

In this definition, rule $C \rightarrow B$ can be inferred from rule $A \rightarrow B$ under the monotonicity assumption because if rule $A \rightarrow B$ holds on some data and $C \models A$ then, by the monotonicity assumption, $A \rightarrow B$ should hold on the subset of data defined by *C*.

[1] One of the ways to define "large subset of tuples" is through the user-specified *support* threshold value.

[2] This is not surprising since the objective of Apriori is to discover all strong rules, while ZoomUR discovers only unexpected rules.

Example. Consider the rules *diaper, weekday* ® *not_beer* and *diaper, weekday, male* ® *not_beer*. Since *diaper, weekday, male* |= *diaper, weekday* it follows that the rule *diaper, weekday, male* ® *not_beer* is implied from the rule *diaper, weekday* ® *not_beer* under the monotonicity assumption (*diaper, weekday* ® *not_beer* $|=_M$ *diaper, weekday, male* ® *not_beer*), and therefore is redundant in that sense. □

We next present a definition for the minimal set of rules, followed by the definition of the minimal set of unexpected patterns.

3.2 Minimal Set of Rules

Definition. *Y* is the *minimal set* of *X* if and only if the following conditions hold:

(1) $Y \subseteq X$.

(2) $\forall x_i \in X, \exists y_i \in Y \mid y_i \models_M x_i$.

(3) $\forall y_1, y_2 \in Y, y_1 \not\models_M y_2$.

Proposition 3.1. For any set of rules *X*, the minimal set of *X* is unique.

Proof. To prove this proposition, we define a directed graph $G=(V, E)$ as follows. The set of nodes *V* consists of all the rules from *X*. Given two nodes $n_1 = A \rightarrow B$ and $n_2 = C \rightarrow D$ from *V*, there is an edge from node n_1 to node n_2 in *E* if *A* ® *B* $|=_M$ *C* ® *D*. Then it is easy to see that the minimal set of rules for *X* consists of all the nodes of *G* having no incoming edges (in-degrees of these nodes are 0). Then the claim follows from the observation that the set *G* is unique. □

We would like to reiterate that we use $|=_M$ instead of classical logical implication |= in the definition above because the concept of mininmality, as defined in this paper, is based exclusively on the inference under the monotonicity assumption as specified in Section 3.1. Given the above definition, we introduce the minimal set of unexpected patterns as follows.

Definition. If *B* is a belief and *X* is the set of all unexpected patterns with respect to *B*, the *minimal set of unexpected patterns* with respect to *B* is the minimal set of *X*. □

4. DISCOVERING THE MINIMAL SET OF UNEXPECTED PATTERNS

In Section 4.1 we present an algorithm MinZoominUR that discovers the minimal set of unexpected refinements. We describe this algorithm because in many applications we are interested only in refinements of beliefs and also because MinZoominUR illustrates some important points used in MinZoomUR. We then present in Section 4.2 an overview of MinZoomUR, an efficient algorithm for discovering the minimal set of unexpected patterns.

The inputs to algorithms MinZoominUR and MinZoomUR are: (1) a set of beliefs, *B*, (2) the dataset *D*, (3) minimum support and confidence values *minsup* and *minconf* and (4) minimum and maximum *width* for all ordered attributes. In the case of ordered attributes the width of any condition of the form $value_1$ £ *attribute* £ $value_2$ is defined to be $value_2$ - $value_1$. We take as user inputs the minimum and maximum width for all ordered attributes. Note that this is *not* a restrictive assumption in any way since the default can be the smallest width and largest width respectively for these two parameters.

4.1 Discovering Minimal Unexpected Refinements of Beliefs

In this section we present MinZoominUR, an algorithm for discovering the minimal set of unexpected refinements to a set of beliefs.

Consider the belief *body* ® *head*, having the structure specified in Section 2. We use the term *"CONTR(head)"* to refer to the set of atomic conditions that contradict the atomic condition specified by *head*. Assume that $v_1, v_2, ..., v_k$ are the set of unique values (sorted in ascending order if the attribute *a* is ordered) that *a* takes on in *D*. *CONTR(head)* is generated as follows:

(1) If the head of the belief is of the form "$value_1$ £ *attribute* £ $value_2$" (*attribute* is ordered), any condition of the form "$value_3$ £ *attribute* £ $value_4$"∈ *CONTR(head)* if the ranges [$value_1$, $value_2$] and [$value_3$, $value_4$] do not overlap.

(2) If the head of the belief is of the form "*attribute = val*" (*attribute* is unordered), any condition of the form "*attribute* = v_p"∈ *CONTR(head)* if $v_p \in \{ v_1, v_2, ..., v_k \}$ and v_p ¹ *val*;

Algorithm MinZoominUR is based on Apriori [1] and ZoomUR [11] with several major differences. First, unlike in Apriori, generation of large itemsets starts with a set of beliefs that seed the search. Second, unlike in Apriori and ZoomUR, MinZoominUR does not generate those itemsets that are guaranteed to produce non-minimal rules. Third, rule generation process is integrated into the itemset generation part of the algorithm – this process is immaterial for Apriori and ZoomUR but results in significant efficiency improvements for MinZoominUR.

Before presenting MinZoominUR, we first present a broad overview of the algorithm. Each iteration of MinZoominUR generates itemsets in the following manner. In the *k*-th iteration we generate itemsets of the form {*C,body,P*}, where *C* Î *CONTR(head)* and *P* is a conjunction of *k* atomic conditions. Observe that to determine the confidence of the rule *body, P* ® *C*, the supports of both the itemsets {*C,body,P*} and {*body,P*} will have to be determined. Hence in the *k-th* iteration of generating large itemsets, two sets of candidate itemsets are considered for support determination:

(1) The set C_k of candidate itemsets. Each itemset in C_k (e.g. {*C,body,P*}) contains

- (i) a condition that contradicts the head of belief, (i.e. any condition *C* Î *CONTR(head)),*
- (ii) the body {*body*} of the belief, and
- (iii) *k* other atomic conditions (*P* is a conjunction of *k* atomic conditions).

Inputs: Beliefs *Bel_Set*, Dataset *D*, *minwidth* and *maxwidth* for all ordered attributes *ORD* and thresholds *min_support* and *min_conf*
Outputs: For each belief, *B*, MinUnexp(B)

```
1 forall beliefs B ∈ Bel_Set {
2       MinUnexp(B) = {}
3     C0 = { {x,body(B)} | x ∈ CONTR(head(B)) };
4     C0' = {{body(B)}};
5     k=0
6     while (Ck != ∅ ) do {
7         forall c ∈ Ck ∪ Ck', compute support(c)
8         Lk = {x| x ∈ Ck, support(x) ≥ min_support }
9         Lk' = {x| x ∈ Ck', support(x) ≥ min_support}
10        forall (x ∈ Lk) {
11             Let a = x ∩ CONTR(head(B)) /* this intersection is a single element */
12               rule_conf = support(x)/support(x-a)
13                 if (rule_conf > min_conf) {
14                   MinUnexp(B) = MinUnexp(B) ∪ {x – a →a}
15                     Lk = Lk - x
16                  }
17            }
18              k++
19              Ck = generate_new_candidates(Lk-1, B)
20              Ck' = generate_bodies(Ck , B)
21       }
22       forall x ∈ MinUnexp(B) {
23          Other_unexp = MinUnexp(B)-x
24        if (∃ y ∈ Other_unexp | y |=M x) {
25              MinUnexp(B) = MinUnexp(B) - {x}
26            }
27        }
28 }
```

Figure 4.1 ***Algorithm MinZoominUR***

(2) A set C_k' of additional candidates. Each itemset in C_k' (e.g. {*X*,*P*}) is generated from an itemset in C_k by dropping the condition, *C*, that contradicts the head of the belief.

In each iteration, minimal unexpected rules are generated from the set of large itemsets. The main idea in MinZoominUR is that if an itemset generates an unexpected rule, *it is deleted from consideration* and therefore no superset of this itemset is even considered in subsequent iterations. As we prove in Theorem 4.1, this step avoids generation of itemsets producing non-minimal rules and significantly improves the efficiency of the algorithm.

We explain the steps of MinZoominUR in Fig. 4.1 now. The following is a list of notations that are used in describing the algorithm:

- *UNORD* is the set of unordered attributes.
- *ORD* is the set of ordered attributes.
- *minwidth(a)* and *maxwidth(a)* are minimum and maximum widths for ordered attribute *a*.
- *Attributes(x)* is the set of all attributes present in any of the conditions in itemset *x*.
- *Values(a)* is the set of distinct values the attribute *a* takes in the dataset *D*.

First, given a belief, *B*, the set of atomic conditions that contradict the head of the belief, *CONTR(head(B))*, is computed (as described previously). Then, the first candidate itemsets generated in C_0 (step 3) will each contain the body of the belief and a condition from *CONTR(head(B))*.

Steps (6) through (20) in Fig. 4.1 are iterative: Steps 7 through 9 determine the supports in dataset *D* for all the candidate itemsets currently being considered and selects the large itemsets L_k and L_k'. Each itemset in L_k contains the body and the head of a potentially unexpected rule, while each itemset in L_k' contains only the body of the potentially unexpected rule. Steps 10 through 17 generate unexpected rules such that large itemsets that contribute to unexpected rules are subsequently deleted in Step 15. Specifically, for each large itemset in L_k, if the unexpected refinement rule that is generated from the itemset has sufficient confidence, then two actions are performed:

1. Step 14 adds this rule to the set of potentially minimal unexpected refinements.
2. Step 15 deletes the corresponding itemset from L_k since any itemset that is a superset of this itemset can only generate unexpected refinements that can be monotonically inferred from the new rule generated in step 14.

In step (19), function *generate_new_candidates*(L_{k-1}, B) generates the set C_k of new candidate itemsets to be considered in the next pass from the previously determined set of large itemsets, L_{k-1}, with respect to the belief B ("*x* ® *y*") as described in ZoomUR [11]. In general we generate C_1 from L_0 by adding additional conditions of the form *attribute* = *value* for unordered attributes or of the form *value1* £ *attribute* £ *value2* for ordered attributes to each of the itemsets in L_0. Incremental generation of C_k from L_{k-1} when $k > 1$ is similar to the *apriori-gen* function described in [1]. In step (20), as described previously, we would also need the support of additional candidate itemsets in $C_{k'}$ to determine the confidence of unexpected rules that will be generated. The function *generate_bodies*(C_k, B) generates $C_{k'}$ by considering each itemset in C_k and dropping the condition that contradicts the head of the belief and adding the resulting itemset in $C_{k'}$.

Steps (22 – 27) are needed to detect any remaining non-minimal rules that arise due to the following special case of certain itemsets containing unordered attributes. To illustrate this special case, consider the following two itemsets: {{*a=1*}, {*5 £ b £ 10*}} and {{*a=1*}, {*7 £ b £ 8*}}. The special case is that neither of these sets is a "superset" of the other, yet (*5 £ b £ 10* → *a=1*) $|=_M$ (*7£b £ 8* → *a=1*) since (*7 £ b £ 8*) |= (*5 £ b £ 10*). Therefore, the rule *7 £ b £ 8* → *a=1* should be eliminated in order to produce the minimal set of unexpected rules. Since Steps (6 – 21) of the algorithm do not eliminate such rules, the additional Steps (22 – 27) do this. Note that in the case of only unordered attributes in the itemsets, Steps (22 – 27) of the algorithm *are not needed* since MinUnexp(B) after Step 21 is guaranteed to be minimal (see the proof of Theorem 4.1).

The computational complexity of Steps (1 – 21) is determined by the total number of candidate itemsets *K* generated in Steps (19 - 20) taken over all the iterations of the While-loop. The computational complexity of the elimination procedure in Steps (22 – 27) is $O(n^2)$, where *n* is the size of the set MinUnexp(B). In practice $K >> n^2$. Therefore, the bottleneck of MinZoominUR algorithm lies in Steps (6 – 21). Moreover, the complexity of MinZoominUR in the worst case is comparable to the worst-case complexity of Apriori that is bounded by $O(\|C\| * \|D\|)$, where $\|C\|$ denotes the sum of the sizes of candidates considered, and $\|D\|$ denotes the size of the database [1]. However, in the average case, the computational complexity of MinZoominUR is significantly lower than that of Apriori. This is the case because the average number of candidates considered in MinZoominUR is significantly lower than that for Apriori due to (a) minimality-based elimination procedure, and (b) presence of the initial set of beliefs that seed the search process.

Observe that a key strength of MinZoominUR, compared to ZoomUR [11] and Apriori [1], is that rule discovery is *integrated* into the itemset generation procedure in such a way that it can greatly reduce the number of itemsets generated in subsequent iterations.

Theorem 4.1. For any belief, *B*, MinZoominUR discovers the minimal set of unexpected rules that are refinements to the belief.

Sketch of the Proof. We will first show that for the case where there are unordered attributes only, MinZoominUR generates the minimal set of unexpected patterns *without needing to apply the minimal filter* (Steps 22 through 27 of Figure 4.1). For unordered attributes only, it is easy to see that a rule $X_1=x_1, X_2=x_2, \ldots, X_n=x_n$ ® $Y = y_1$ is non-minimal if and only if there is a rule of the form Z ® $Y = y_1$, where $Z \subset \{X_1=x_1, X_2=x_2, \ldots, X_n=x_n\}$[3]. From this observation it can be shown that, as done in MinZoominUR, itemset deletion immediately following the generation of an unexpected rule from the itemset is adequate to guarantee the generation of the minimal set of unexpected refinements. However there is a special case involving ordered attributes that cannot guarantee *only* minimal rules before Steps 22-27. This special case arises since a syntactic subset check cannot capture containment when dealing with ranges of values for ordered attributes. An example of this special case was given above in Section 4.2. Hence the filter in Steps 22-27 removes any non-minimal rules remaining. A detailed proof of this theorem is in [10]. □

In this section we focused on discovering minimal set of unexpected refinements of beliefs. In the next section we present the main ideas of MinZoomUR, an algorithm that discovers the minimal set of unexpected patterns.

4.2 Discovering Minimal Unexpected Patterns

Due to the space limitation, we present only an overview of the discovery algorithm. The complete description can be found in [10]. First we present a few preliminaries. We use the term *parents(x)* to denote the set of all subsets of *x* that contain the body of the belief and one condition that contradicts the head of the belief considered in previous iterations[4] during the candidate generation phase of the algorithm. Specifically,

$parents(x) = \{ a \mid a \subset x, body(B) \subset a, c \in CONTR(head(B)), c \in a\}$

An itemset *y* is said to be a parent of *x* if $y \in parents(x)$.

We use the term *zoomin rules* to denote unexpected rules that are refinements to beliefs and *zoomout rules* for unexpected rules that are more general unexpected rules. The large itemset *x* is said to generate a zoomin *rule* if confidence $(x - c \rightarrow c) > min\_conf$, where $c \in CONTR(head(B))$. The large itemset *x* is said to generate a zoomout rule if *x* generates a zoomin rule $x - c \rightarrow c$ *and* confidence$(x - c - d \rightarrow c) > min_conf$, where $c \in CONTR(head(B))$, $d \subseteq body(B)$ and *d* is not empty.

Associated with each itemset, *x*, are two attributes: *x.rule*, that keeps track of whether a zoomin rule is generated from *x*, and *x.dropped_subsets*, which keeps track of the subsets of *body(B)* that are dropped during the discovery of zoomout rules.

Unlike what was done in MinZoominUR, an itemset that generates a zoomin rule in MinZoomUR *cannot* always be deleted from subsequent consideration since it is possible for minimal zoomout rules to be derived from non-minimal zoomin rules.

---

[3] Note that this "syntactic" subset property is *not* true when dealing with ordered attributes, which is why the minimal filter in Steps 22-27 are necessary.

[4] Recall that the candidate generation phase of these algorithms (Apriori, MinZoominUR and MinZoomUR) is iterative such that itemsets in subsequent iterations have greater cardinality (number of items).

Consider the following example. For a belief $a, b \rightarrow x$, let $a, b, c \rightarrow y$ and $a, b, c, d \rightarrow y$ be two zoomin rules. Though $a, b, c, d \rightarrow y$ is a non-minimal zoomin rule, the rule may result in a zoomout rule such as $b, c, d \rightarrow y$ which may belong to the minimal set of unexpected rules.

Extending this example one more step, we observe that the zoomout rule $b, c, d \rightarrow y$ can, however, be guaranteed to be non-minimal if the first zoomin rule $a, b, c \rightarrow y$ resulted in a zoomout rule of the form $p, c \rightarrow y$ such that $b, c, d \models p, c$ where p is a proper subset of the body of the belief. Examples of such p are $\{b\}$ and $\{\}$ corresponding to the zoomout rules $b, c \rightarrow y$ and $c \rightarrow y$ respectively (generated from $a, b, c \rightarrow y$). However if the first zoomin rule generated only the zoomout rule $a, c \rightarrow y$, it may still be possible for the zoomout rule $b, c \rightarrow y$ to be minimal since $b, c, d \not\models a, c$.

The discovery strategy of MinZoomUR is based on the following conditions under which some generated rules are guaranteed to be non-minimal and hence can be excluded from the minimal set. These exclusion rules are integrated into the itemset generation phase of the algorithm (similar to the single exclusion rule integrated into MinZoominUR) and thus substantially reduce the number of itemsets considered in subsequent iterations. [10] proves that these conditions do indeed exclude only non-minimal rules, hence we only state these rules here for lack of space. The "exclusion rules" used in MinZoomUR are:

1. If x and y are two large itemsets such that x is a parent of y and *x.rule*=1 rule then the zoomin rule generated from y cannot be minimal. This is the only exclusion rule used previously in algorithm MinZoominUR.
2. If x is a large itemset that generates a zoomin rule and some zoomout rules, then the zoomin rule generated cannot be minimal.
3. If x is a large itemset that generates zoomout rules p and q and *elem_p* $\in$ *x.dropped_subsets(p)* [5] and *elem_q* $\in$ *x.dropped_subsets(q)* and *elem_p* $\subset$ *elem_q* then p cannot be minimal.
4. If x and y are two large itemsets such that x is a parent of y, zoomout rules generated from y generated by dropping any subset, p, from the body of the belief such that p is a subset of some element belonging to *x.dropped_subsets* cannot be minimal rules.

MinZoomUR generates candidate itemsets in the same manner as in MinZoominUR. A main difference in the algorithms is that MinZoomUR considers zoomout rules also for a given itemset immediately after the itemset generates a zoomin rule. This is necessary because some of the "exclusion rules" applied to an unexpected rule generated depends on knowing the zoomout rules generated for that itemset and its parents. After the four exclusion rules are applied, MinZoomUR also applies the minimal filter similar to the one specified in lines (22) – (27) of MinZoominUR. Moreover, as shown in (10), this minimal filter is necessary only when there are ordered attributes. In the case when all the attribures are unordered, the four exclusion rules are also sufficient conditions for generating only minimal rules (hence no minimal filter is necessary at the end).

In summary, the following theorem states, MinZoomUR discovers *all* the minimal unexpected patterns. The proof of this theorem can be found in [10].

Theorem 4.2. For any belief MinZoomUR discovers the minimal set of unexpected patterns.

We would also like to note that the classical notion of "minimality" often assumes that it is possible to reconstruct the set of all objects having certain property from the minimal set of objects having this property. In our case also, the set of all unexpected patterns *can* be reconstructed from the minimal set of unexpected patterns. However, this can be done *only* using a computationally intensive process that requires extensive data manipulation, rather than through an immediate reconstruction procedure that does not require any additional data access. This limitation of our approach is the result of a development of efficient search algorithms that directly discover the minimal set of unexpected patterns without even examining all unexpected patterns. Moreover, this limitation can also be circumvented by letting the domain expert examine the set of minimal unexpected patterns (that is small), select the most interesting minimal patterns, and use the system to automatically refine them to discover all the unexpected patterns obtained from this selected set.

5. EXPERIMENTS

To illustrate the usefulness of our approach to discovering patterns, in this section we consider a case study application of applying the methods to consumer purchase data from a major market research firm. We pre-processed this data by combining different data sets (transaction data joined with demographics), made available to us into one table containing 38 different attributes and 313409 records. For simplicity in generating beliefs and in making comparisons to other techniques that generate association rules in these experiments we restrict our consideration to rules involving discrete attributes only. An initial set of 28 beliefs was generated by domain experts after examining 300 rules generated from the data using methods described in [10]. In this section we present some results from applying MinZoomUR, ZoomUR [11] and Apriori [1] to this dataset starting from the initial set of beliefs where applicable. Specifically we compare these methods in terms of the number of rules generated and provide some guidelines as to when each may be applicable and also present results from scale-up experiments. We refer the reader to [10, 11] for several examples of truly unexpected discoveries from applying our unexpected pattern discovery methods.

5.1 Number of Patterns Generated

For a fixed minimum conf. level of 0.6, Figure 5.1 through 5.3 show the number of patterns generated by Apriori, ZoomUR and MinZoomUR for varying levels of minimum support.

[5] For belief B and itemset x, if p is a single zoomout rule, then *x.dropped_subsets(p)* will contain only one element which is the subset of *body(B)* that was dropped to create the zoomout rule.

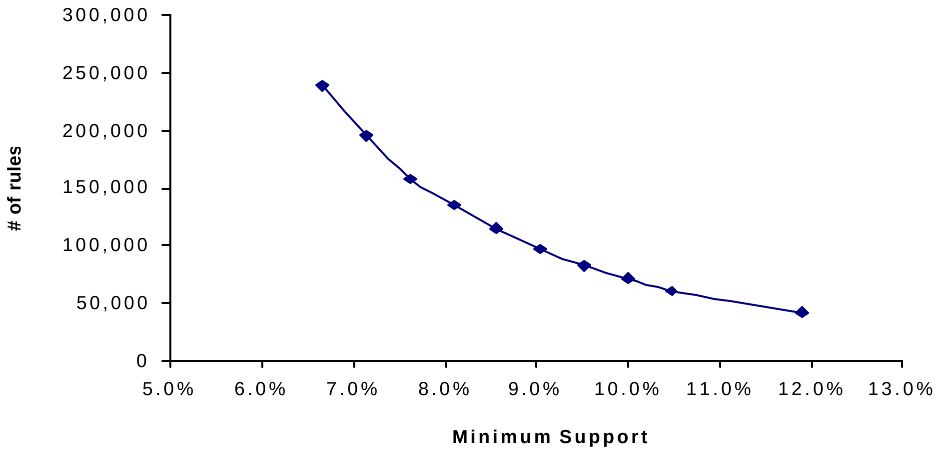

Figure 5.1. Number of rules generated by Apriori

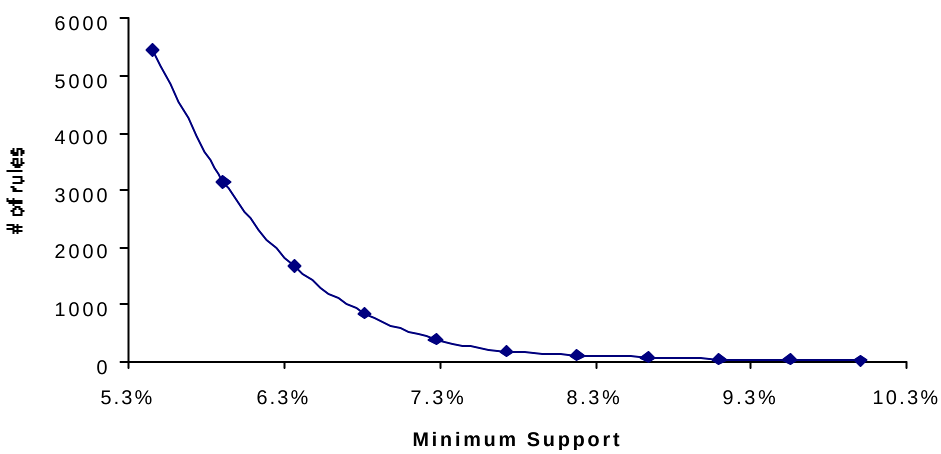

Figure 5.2. Number of unexpected rules generated by ZoomUR

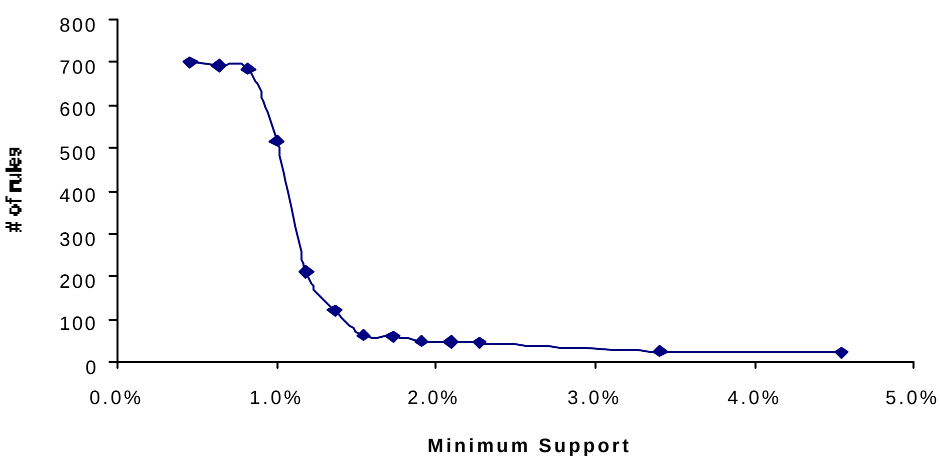

Figure 5.3. Number of unexpected rules generated by MinZoomUR

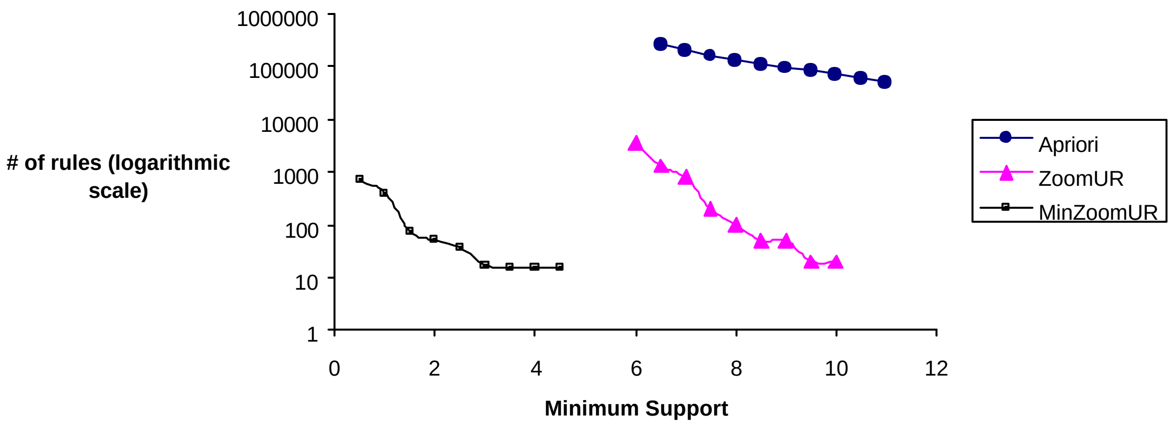

Figure 5.4. Comparison of number of rules generated by Apriori, ZoomUR and MinZoomUR

Apriori generated 50,000 to 250,000 rules even for reasonably high minimum support values. This is not surprising since the objective of Apriori is to discover *all* strong association rules. For reasonable values of support (5 to 10%), ZoomUR generates 50 to 5000 unexpected patterns. MinZoomUR on the other hand generated only 15 to 700 unexpected patterns even for extremely low values for minimum support.

Figure 5.4 illustrates the comparison of the three methods in terms of the number of generated rules. Due to the order of magnitude difference in the number of generated rules, the graph plots the number of rules generated using a logarithmic scale for the Y axis.

As we would expect, as the minimum support threshold is lowered, all the methods discover a greater number of rules. Despite this, MinZoomUR discovers orders of magnitude fewer patterns than both ZoomUR and Apriori.

The graphs in Figures 5.1 – 5.3 also demonstrate that a majority of patterns generated by ZoomUR are redundant. Observe that as the support threshold is lowered, the number of patterns generated by both ZoomUR and Apriori increase more than linearly. While this is the case for MinZoomUR in some regions, MinZoomUR plateaus out for lower regions of support. These plateaus signify that very few new minimal unexpected patterns are generated in these experiments despite the fact that in these experiments the number of unexpected patterns generated by ZoomUR keep increasing in that region. This observation coupled with the comparison in the number of rules generated indicate that MinZoomUR is indeed effective in removing redundant patterns, which represent a large majority of the set of all discovered patterns.

5.2 Discussion

Based on these experiments we discuss below some possible tradeoffs between these methods and provide some guidelines to their usage.

The clear advantage of MinZoomUR over ZoomUR is that it generates far fewer patterns and yet retains most of the truly interesting ones as shown in [10]. Since ZoomUR generates all unexpected patterns for a belief and MinZoomUR generates the minimal set of unexpected patterns, MinZoomUR will always generate a subset of patterns that ZoomUR generates. As shown above, this subset can be very small (from 15 to a few hundred patterns for the entire set of beliefs, while ZoomUR can generate an order of magnitude more). Moreover, domain experts can selectively refine some of the patterns in the minimal set to obtain all unexpected patterns that are refinements to the selected pattern.

The drawback of MinZoomUR compared to ZoomUR is that MinZoomUR makes an implicit assumption that minimal unexpected patterns are the "most interesting" patterns. From a subjective point of view this may not be necessarily true. Consider the following example of two unexpected patterns:

- *When coupons are available for cereals, they don't get used (confidence = 60%)*
- *On weekends, when coupons are available for cereals they don't get used (confidence = 98%)*

MinZoomUR will not generate the second unexpected pattern since it is monotonically implied by the first pattern. However, the second unexpected pattern has a much higher confidence and may be considered "more unexpected" by some users. In a more general sense, the criteria implied by monotonicity and confidence are just two methods to rank unexpected patterns. In general there may be other criteria, some of which even depending on other subjective preferences of a user. Hence, since ZoomUR generates all the unexpected patterns, it is guaranteed to contain all the unexpected patterns that are "most unexpected" from any specific definition of the term "most unexpected". In the context of objective measures of interestingness, [2] discuss interesting approaches to finding the "most interesting" patterns. In subsequent work, we will study the issue of generating the "most unexpected patterns" by characterizing the degree of unexpectedness for patterns along the lines of [2, 18].

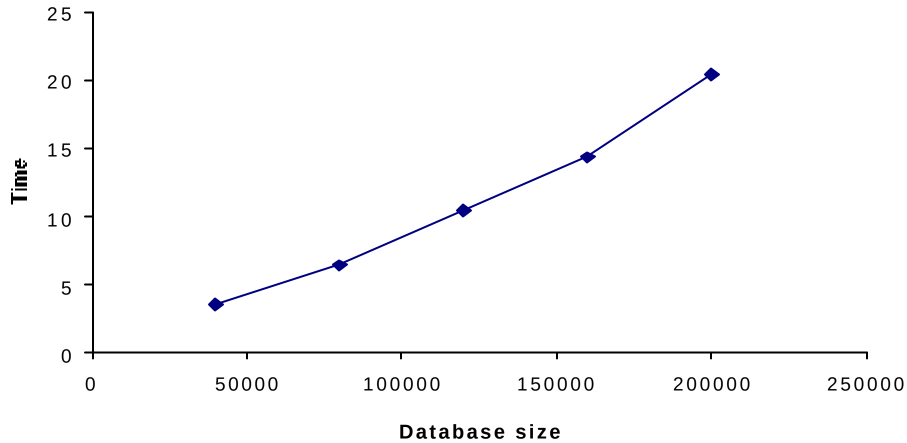

Figure 5.5. Execution time of MinZoomUR as a function of database size

Given the relative advantages of the two methods to discovering unexpected patterns, a practical implication of the above is that ZoomUR can be used to generate unexpected patterns for high levels of support values and MinZoomUR can be used if patterns of very low support need to be generated. As shown in Figure 5.3, MinZoomUR generates a reasonable number of unexpected patterns even for extremely small values of minimum support, as low as even 0.5%. Also the support of some beliefs about a domain may be very low, perhaps reflective of some condition that occurs rarely. In such cases methods such as MinZoomUR that can find patterns at very low support values are *necessary.* As Figure 5.2 shows, for such low values of minimum support most methods may discover tens of thousands of patterns, resulting in a data mining problem of the second order.

Apriori on the other hand has the drawback of generating a very large number of patterns since the objective is to discover all strong rules. As Figure 5.1 shows, for very low support values, this could easily result in a few million rules even on mid-sized problems.

However there are two sides of the coin. Generating a very large number of patterns results in a data mining problem of a second order and is hence avoidable. At the same time it is possible that either of the two methods that seek unexpected patterns could miss other "interesting" patterns that may be unrelated to domain knowledge. However the set of patterns generated by Apriori can, trivially, be guaranteed to have all the interesting patterns since it has all patterns. We believe that this tradeoff is in some sense unavoidable since the problem of generating all interesting patterns (not just "unexpected") is a difficult problem to solve.

Below we experimentally examine the scalability of MinZoomUR with respect to the size of the database.

5.3 Scalability with the size of the database

For a sample of 10 beliefs, we ran MinZoomUR multiple times by varying the number of records in the dataset from 40,000 to 200,000. Figure 5.5 shows the execution times for MinZoomUR.

These experiments indicate that MinZoomUR scales, in the range considered, almost linearly with the size of the database.

In this section we presented results pertaining to the effectiveness of MinZoomUR and compared it to Apriori and ZoomUR. We demonstrated that MinZoomUR can be used to discover far fewer patterns than Apriori and ZoomUR, yet finding most of the truly interesting patterns.

6. CONCLUSIONS

In this paper we presented a definition for the minimal set of unexpected patterns and proposed two algorithms for discovering the minimal set of such patterns. In a real-world application we demonstrated that the main discovery algorithm, MinZoomUR, discovered orders of magnitude fewer patterns than other comparable methods and yet retained most of the truly interesting patterns. We also discussed tradeoffs between various discovery methods and presented some guidelines for their usage.

The power of this approach lies in combining two independent concepts of unexpectedness and minimality of a set of patterns into one integrated concept that provides for the discovery of small but important sets of interesting patterns. Moreover, MinZoominUR and MinZoomUR are efficient since they directly discover minimal unexpected patterns.

7. REFERENCES

[1] Agrawal, R., Mannila, H., Srikant, R., Toivonen, H. and Verkamo,A.I., 1995. Fast Discovery of Association Rules. In Fayyad, U.M., Piatetsky-Shapiro, G., Smyth, P., and Uthurusamy, R. eds., *Advances in Knowledge Discovery and Data Mining.* AAAI Press.

[2] Bayardo, R. and Agrawal, R., 1999. Mining the Most Interesting Rules. In *Proc. of the Fifth ACM-SIGKDD International Conference on Knowledge Discovery and Data Mining.*

[3] Bayardo, R., Agrawal, R. and Gunopulos, D., 1999. Constraint-Based Rule Mining in Large, Dense Databases. In *Proceedings of ICDE, 1999*.

[4] Berger, G. and Tuzhilin, A., 1998. Discovering Unexpected Patterns in Temporal Data Using Temporal Logic. In Etzion, O., Jajodia, S. and Sripada, S. eds, *Temporal Databases: Research and Practice*. Springer, 1998.

[5] Chakrabarti, S., Sarawagi, S. and Dom, B. Mining Surprising Patterns Using Temporal Description Length. In *International Conference on Very Large Databases*, 1998.

[6] Liu, B. and Hsu, W., 1996. Post-Analysis of Learned Rules. In *Proc. of the Thirteenth National Conference on Artificial Intelligence* (AAAI '96), pp. 828-834.

[7] Liu, B., Hsu, W. and Chen, S, 1997. Using General Impressions to Analyze Discovered Classification Rules. In *Proc. of the Third International Conference on Knowledge Discovery and Data Mining (KDD 97)*, pp. 31-36.

[8] Liu, B., Hsu, W., and Ma, Y., 1999. Pruning and Summarizing the Discovered Rules. In *Proc. of the Fifth ACM-SIGKDD Int'l Conference on Knowledge Discovery and Data Mining*.

[9] Mitchell, T., 1982. Generalization as Search. *Artificial Intelligence*, pp. 203-226.

[10] Padmanabhan, B., 1999. Discovering Unexpected Patterns in Data Mining Applications. *Doctoral Dissertation*, New York University, May 1999.

[11] Padmanabhan, B. and Tuzhilin, A., 1998. "A Belief-Driven Method for Discovering Unexpected Patterns." *In Proc. 4th Int'l Conf. on Know. Discovery and Data Mining*, 1998.

[12] Padmanabhan, B. and Tuzhilin, A., 1999. Unexpectedness as a Measure of Interestingness in Knowledge Discovery. In *Decision Support Systems*, (27)3 (1999) pp. 303-318

[13] Suzuki, E., 1997. Autonomous Discovery of Reliable Exception Rules. In *Proceedings of the Third International Conference on Knowledge Discovery and Data Mining*, pp. 259-262.

[14] Subramonian, R. Defining diff as a data mining primitive. In *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining, 1998*.

[15] Srikant, R. and Agrawal, R., 1996. Mining Quantitative Association Rules in Large Relational Tables. In *Proc. of the ACM SIGMOD Conference on Management of Data, 1996.*

[16] Shah, D., Lakshmanan, L.V.S, Ramamritham, K., and Sudarshan, S., 1999. Interestingness and Pruning of Mined Patterns. In *Proceedings of the 1999 ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery (DMKD)*, Philadelphia, 1999.

[17] Silberschatz, A. and Tuzhilin, A., 1995. On Subjective Measures of Interestingness in Knowledge Discovery. In *Proc. of the First International Conference on Knowledge Discovery and Data Mining*, pp. 275-281.

[18] Silberschatz, A. and Tuzhilin, A., 1996. What Makes Patterns Interesting in Knowledge Discovery Systems. *IEEE Transactions on Knowledge and Data Engineering*. Special Issue on Data Mining, vol. 5, no. 6, pp. 970-974.

[19] Srikant, R., Vu, Q. and Agrawal, R. Mining Association Rules with Item Constraints. In *Proceedings of the Third International Conference on Knowledge Discovery and Data Mining (KDD 97)*, pp. 67-73.

[20] Toivonen, H., Klemetinen, M., Ronkainen, P., Hatonen, K. and Mannila, H., 1995. Pruning and Grouping Discovered Association Rules. In *MLNet Workshop on Statistics, Machine Learning and Discovery in Databases*, pp. 47-52.