CS 6604: Data Mining Fall 2007

Lecture 4 — Wednesday, August 29, 2007

Lecture: Naren Ramakrishnan *Scribe: Joseph Turner*

1 Overview

In the last lecture we examined special classes of itemsets including closed and maximal itemsets, and techniques for mining them.

In this lecture we develop a basic theory and approach for the mining of rules from itemsets. In general, we will highlight some aspects of rules and rule mining that will reinforce the lessons:

1. *Rules are good.*
2. *Rules are too much.*
3. *Rules are bad.*

These are deliberately intended to be contradictory, and supporting examples will be given for each. It is left as an exercise for the reader to relatively weight these lessons.

2 Overview of Rules

What are rules? Given an itemset $X \subseteq I$, a rule (and its confidence) derived from X describes the relationship between items in X. Formally, such a rule can be written as $A \rightarrow B$, where $A, B \subset X$ and $A \cup B = X$. The *confidence* of the rule is written in terms of the support of A and B. Specifically, $\text{confidence}(A \rightarrow B) = \frac{\text{support}(A \cup B)}{\text{support}(A)}$. The more confident the rule, we more we are inclined to 'believe' it, although in some cases it can be misleading. More on this later.

Rules are actionable information. A retailer can tailor its presentation and pricing to address relationships discovered by rules. For instance, a rule indicating that the purchase of diapers implies a purchase of beer could lead a retailer to place these two items at opposite ends of the store, requiring someone purchasing both to walk through all of the other items. The actionable nature of rules makes them valuable. This is why rules are good.

3 Rule Mining

What is the goal of rule mining? Ultimately, we would like to be able to extract a small, meaningful set of rules from each frequent itemset. This can prove challenging, as the number of possible rules for itemset A is $2^{|A|} - 2$. First, we will look at an algorithm for extracting confident rules. Next, we will examine an algorithm for extracting a minimal rule set. Finally, we will introduce an algorithm for finding correlated sets.

3.1 Mining Confident Rules

The first and most basic form of rule mining extracts rules above a certain confidence threshold. As mentioned above, the set of all possible rules is exponential in the size of the itemset. Even so, the problem of extracting confident rules has traditionally taken a back seat to the problem of mining frequent itemsets. Why?

The key idea, as hinted to in previous classes, is that there is a problem similar to finding frequent itemsets embedded in the problem of finding confident rules. First, notice that for all rules derived from a given itemset, the numerator in the confidence calculation is the same. Thus, for a given confidence threshold c and a given rule $A \rightarrow B$, we can calculate a minimum support for the left-hand side of the rule as $\text{support}(A) \leq \frac{\text{support}(A \cup B)}{c}$. Since support cannot increase as an itemset is grown monotonically, the number of possible rules may be pruned by beginning to use the largest proper subsets of the itemset as the antecedent, and following *down* the subset lattice. If a set fails to have a low enough support for the required confidence, its subsets will share that property. Thus, we can exclude any subsets of a failed subset from the search space.

Example: Let the itemset be $X = \{a, b, c, d, f\}$. Then the rule $\{a, b, c, d\} \rightarrow \{f\}$ has confidence equal to $\frac{\text{support}(\{a,b,c,d\} \cup \{f\})}{\text{support}(\{a,b,c,d\})}$. Similarly, the rule $\{a, b, c\} \rightarrow \{d, f\}$ has confidence equal to $\frac{\text{support}(\{a,b,c\} \cup \{d,f\})}{\text{support}(\{a,b,c\})}$. However, we know

$$\text{support}(\{a, b, c, d\} \leq \text{support}(\{a, b, c\},$$

therefore

$$\text{confidence}(\{a, b, c, d\} \rightarrow \{f\}) \geq \text{confidence}(\{a, b, c\} \rightarrow \{d, f\}).$$

Another way to think about the search is by using the right-hand side of the rule rather than the left, so that we can continue to think in terms of bottom-up movement up the lattice. In either case, we can take advantage of the lexicographic candidate generation and other optimizations of frequent itemset mining.

Lets look at an example of the algorithm in action. Consider the database:

1	ACTW
2	CDW
3	ACTW
4	ACDW
5	ACDTW
6	CDT

Suppose we want to find the confident rules derived from the itemset $\{ACW\}$, having a minimum confidence of $c = 100\%$. Figure 1 illustrates the level-wise nature of the algorithm.

At level 1, $\{CW\} \rightarrow \{A\}$ gets eliminated for low confidence. This leaves only one entry at level 2, effectively reducing the calculations from 6 to 4. Since we consider only proper subsets, there is no level zero or level three.

3.2 Mining a Minimal Set of Rules

In a dense database, there will still be many rules of high confidence. This is what is meant by rules are too much. Ideally, only interesting rules would be extracted. One possibility is to examine a *rule cover*,

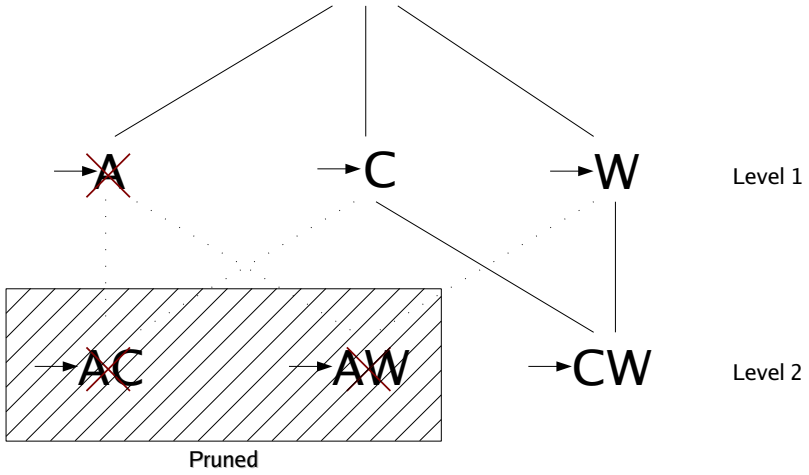

Figure 1: Level-wise view of confident rule search for itemset $\{ACW\}$. The arrows indicate that the proceeding is the right-hand side of the rule.

meaning a set of rules from which all other rules can be derived.

Mining a Rule Cover. As we have seen previously, if we ignore support, the closed itemsets form a *lattice*. Recall that one definition of a closed itemset is an itemset for which its closure is itself. Since the frequency does not change with the application of the closure operator, it follows that the confidence of a rule would not change with the application of the closure operator to both its left- and right-hand sides.

More formally, given a closure operator $f(x) : P(I) \rightarrow P(I)$, and given a rule $A \rightarrow B$, confidence$(A \rightarrow B) =$ confidence$(f(A) \rightarrow f(B))$. Thus we only need to consider rules among closed itemsets. Additionally, we only need to examine rules among adjacent closed sets in the lattice, since the others may be inferred through transitivity. For a rigorous proof of the preceeding two statements, see [3].

For a given edge of the lattice, each direction will produce two rules; the direction from superset to subset will produce a rule with 100% confidence, while the reverse direction will produce a rule with less than 100% confidence (why?).

However, since closed sets might share items such rules might still contain redundancies. One approach is to cast each rule in terms of their minimal generators, which will produce the most general rule.

Example. Let's examine some rules from the database presented above. The edge between the closed sets $\{ACTW\}$ and $\{ACW\}$. The generators for these itemsets produce the following rules:

- $\{TW\} \rightarrow \{A\}$

- $\{TW\} \rightarrow \{AC\}$
- $\{CTW\} \rightarrow \{A\}$

In this case, $\{TW\} \rightarrow \{A\}$ is chosen as the most general rule.

Note: If the concept of support is reintroduced, the frequent closed itemsets instead form a semi-lattice. However, the method presented above still works.

Finding Redescriptions. A statement of the form $A \Leftrightarrow B$ is called an (exact) *redescription* if confidence$(A \rightarrow B)$ = confidence$(B \rightarrow A) = 100\%$. Redescriptions can be mined directly as well. The salient fact is that, for a given closed set X, the $A \Leftrightarrow B$ is a redescription if A and B are generators of the set X. As you can see, the concept is rather simple, although the rigorous proof is not. For full details of this method and its practical implications, refer to the paper by Zaki and Ramakrishnan [2].

3.3 Mining Correlated Sets

Finally, we discuss the method for mining correlations between itemsets, rather than just their rules. Why would we want to do that? It all boils down to this:

Rules are bad. What does it mean for rules to be bad? Rules by themselves are not bad; they are simply a way of describing the data. However, sometimes rules must be contextualized for their meaning to be understood. Lets look at an example.

Example. Lets examine the purchasing frequency of coffee and tea in a coffee shop. Figure 2 presents the data.

	c	$\bar{c}$	row sum
t	20	5	25
$\bar{t}$	20	5	25
col sum	90	10	100

Rows t and $\bar{t}$ correspond to transactions that do and do not, respectively, contain tea. Similarly, rows c and $\bar{c}$ correspond to transactions that do and do not, respectively, contain coffee. If we examine the rule $t \rightarrow c$, we see that it has a confidence of 80%, fairly high. We would then most likely conclude that it is a valid rule. However, the *a priori* probability that a customer buys coffee is 90%! So in reality, fewer people buy coffee if they buy tea than just buy coffee. This rule, when examined in a vaccuum is hence misleading.

3.4 Correlation Mining

In reality, there is a negative correlation between buying tea and coffee. It would be useful to present this information along with the rule. First, a metric for correlation must be settled on. In the work of Brin,

Motwani, and Silverstein [1], they use the Chi-Squared statistic. Calculating the Chi-Squared statistic is an involved matter. For a full discussion (both theoretical and practical), refer to [1].

An interesting property of dependence (and correlation) is that it is *upward closed*. That means that for a given itemset A, if A is dependent, then its supersets will also be dependent. This is the opposite of support, which is *downward closed*. Upward closure is best, though, if the property being sought is an unwanted one. This is because upward closure generates false negatives. If the property is unwanted, the false negatives merely mean more work, and not lost information. The focus of correlation mining then becomes finding *minimal* correlated sets; that is, correlated sets for which no subset is correlated.

The basic algorithm, then, follows the algorithm for finding frequent itemsets. It examines the data in a levelwise fashion, beginning with the smallest itemsets. For itemsets that are considered dependent, the itemset is recorded, and all of its supersets are pruned. Also observe that sets that are *not* dependent are used to prepare candidates for dependence. This is a point of departure from frequent itemset mining, where frequent subsets are used to prepare candidates that might be frequent. In either case, the goal is the find a *border*, here that between correlated and uncorrelated sets.

References

[1] S. Brin, R. Motwani, and C. Silverstein, *Beyond Market Baskets: Generalizing Association Rules to Correlations*, in Proc. ACM SIGMOD'97, pages 265-276, 1997.

[2] M. Zaki and N. Ramakrishnan, *Reasoning about Sets using Redescription Mining*, in Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'2005), Chicago, IL, pages 364-373, Aug 2005.

[3] M. Zaki, *Generating Non-Redundant Association Rules*, in Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 34-43, 2000.