CS 6604: Data Mining Fall 2007

Lecture 3 — Monday, August 27, 2007

Lecture: Naren Ramakrishnan *Scribe: Wesley Tansey*

1 Overview

In the last lecture we covered the basics of itemset mining. Most pertinent to this lecture, three major classes of itemsets were introduced:

1. A *frequent* itemset is simply a set of items occurring a certain percentage of the time.
2. A *closed* itemset is set of items which is as large as it can possibly be without losing any transactions.
3. A *maximal* frequent itemset is a frequent itemset which is not contained in another *frequent* itemset. Unlike *closed* itemsets, maximal itemsets do not imply anything about transactions.

(Although the notion of closed itemset is defined above without using the notion of frequency or support, in practice we will mine closed frequent itemsets, i.e., those closed itemsets that sastisfy minimum constraints on support.)

In this lecture we first introduce two closure operators: f and g. These operators are then used to formally define a *closed set*, a *generator*, and a *minimal generator*. The *A-Close* [3] algorithm is presented as a way to discover closed sets by identifying the generators of an itemset. Finally, the *Pincer Search* [2] algorithm is briefly discussed as a means to discover maximal sets.

2 Foundational Terms and Operators

We begin by defining two useful functions: $t(x)$ and $i(y)$. The function $t(x)$, where $x \subseteq I$, I being the set of all items, yields the set of transactions that contain x. Similarly, $i(y)$, where $y \subseteq T$, T being the set of all transactions, yields the set of items that are contained in every transaction of y.

Essentially, this means that t is a mapping from the power set[1] of all items, $P(I)$, to the power set of all transactions, $P(T)$. This produces a sort of cycle or "dance" where you map between the different sets. Fig. 2 illustrates this mapping.

Question: Starting from a given itemset (or transaction set), will iterating on these operators ultimately expand to the entire set?

[1] A power set is a set of all subsets.

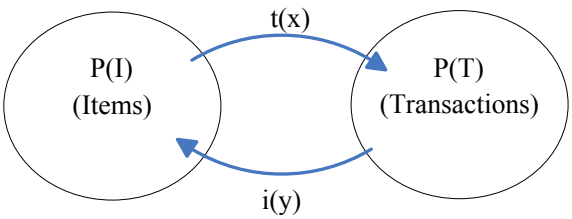

Figure 1: A mapping between powersets. $P(I)$ is the power set of all items. $P(T)$ is the power set of all transactions.

Students and Courses Analogy: Consider a set of students, S, and a set of courses, C. For any subset $s \subseteq S$, if we enter this cycle, we are not guaranteed to arrive at the set of all students or all classes. For instance, CS 6604 may only have one doctoral student, and that student may not currently be enrolled in any other classes. If this student is not a member of s, then the cycle will never expand to include CS 6604 nor the doctoral student. The same is true of the $t(x)$ and $i(y)$ cycle.

2.1 Example Mappings

Consider the following example:

transaction ID	items
1	{A,C,T,W}
2	{C,D,W}
3	{A,C,T,W}
4	{A,C,D,W}
5	{A,C,D,T,W}
6	{C,D,T}

Example 1: $t(\{C\})$ The itemset $\{C\}$ is found in all transactions, so the result is $t(\{C\}) = \{1, 2, 3, 4, 5, 6\}$. Taking this result and calling i on it produces the original input set (i.e., $i(\{1, 2, 3, 4, 5, 6\}) = \{C\}$).

Example 2: $t(\{C, D\})$ The itemset $\{C, D\}$ is found in four transactions (i.e., $t(\{C, D\}) = \{2, 4, 5, 6\}$). Similarly to the first example, calling i on this result produces the original input set (i.e., $i(\{2, 4, 5, 6\}) = \{C, D\}$).

Example 3: $t(\{A\})$ The itemset $\{A\}$ is found in four transactions (i.e., $t(\{A\}) = \{1, 3, 4, 5\}$). However, unlike the first two examples, calling i **does not** produce the original set, as $i(\{1, 3, 4, 5\}) = \{A, C, W\}$.

2.2 Closure Operators

A key insight of the t and i mapping functions is that only two things can happen when calling them successively on a subset, x:

1. The subset remains the same (e.g., $i(t(x)) = x$).
2. The subset increases slightly, but will never increase again thereafter (e.g., if $i(t(x)) = y$ and $x \subset y$, then $i(t(y)) = y$).

To make such dances easy, we define two new functions, $f(x) = i(t(x))$ and $g(x) = t(i(y))$. These two new functions are called *closure operators*. As can be found in most discrete math textbooks, to be a *closure operator* requires satisfying three properties:

Idempotence For all inputs, if the operator is applied twice, it must produce the same result as if it were applied once (e.g., $f(f(x)) = f(x)$).

Extension The original input subset must be contained in the resulting subset (e.g., $x \subseteq f(x)$).

Monotonicity If one input subset is contained in another input subset, then the first resulting subset must be contained in the second resulting subset (e.g., if $x \subseteq y$ then $f(x) \subseteq f(y)$).

Next we use these closure operators to build some formal definitions.

2.3 Formal Definitions

There are three terms which we define here, namely *closed set*, *generator*, and *minimal generator*.

2.3.1 Closed Set

A set, x, is considered *closed* w.r.t. closure operator f if it satisfies the property $f(x) = x$, i.e., its closure is itself. Conversely, a set is considered not closed if $x \subset f(x)$ (i.e., if the set gains elements when calling $f(x)$). Note that $x \subseteq f(x)$ holds always.

Example: Looking back at the example in Section 2.1, $\{C\}$ is *closed* since $f(\{C\}) = i(t(\{C\})) = \{C\}$. However, $\{A\}$ is not closed, as $f(\{A\}) = i(t(\{A\})) = \{A, C, W\}$. But $\{A, C, W\}$ is closed.

2.3.2 Generator

A set x is called a *generator* of y if the *closure* of x is y.

Example: From the example in Section 2.1, both $\{A\}$ and $\{A, C, W\}$ are generators of $\{A, C, W\}$.

2.3.3 Minimal Generator

A set, x, is called a *minimal generator* of y if it satisfies two properties:

1. The *closure* of x is y.
2. No proper subset of x generates y.

Example: From the example in Section 2.1, $\{A\}$ is a *minimal generator* of $\{A, C, W\}$. However, $\{A, C, W\}$ is not a *minimal generator* as $\{A\}$ is a subset of $\{A, C, W\}$ which is itself a generator.

It is important to note that for a given set, there could be multiple *minimal generators*. Consider the following simple example:

transaction ID	items
1	{A,W}
2	{A,W}

In this example, both $\{A\}$ and $\{W\}$ are *minimal generators* of $\{A, W\}$.

3 Finding Closed Itemsets

Now that we have introduced the idea of *closed sets*, the next step is to determine a way to find the closed sets. The approach we discuss in this section is the *A-Close* algorithm [3].

3.1 The A-Close Algorithm

There are two main steps to the *A-Close* algorithm. First, a level-wise search is performed to find the generators which have sufficient support. The generators are then used as inputs to f, and the resulting outputs are the closed sets.

3.1.1 Example

Since example from the Section 2.1 is not sparse enough to fit our needs here, consider the following example:

transaction ID	items
1	{A,C,D}
2	{B,C,E}
3	{A,B,C,E}
4	{B,E}
5	{A,B,C,E}

This is the example we will use throughout the description of the *A-Close* algorithm. Additionally, we assume a support of $\theta = 3$.

3.1.2 Level-wise Search

The level-wise search is a modified version of the *AS* algorithm [1]. Working from our example, we get the following:

Level 1:

sets	support
{A}	3
{B}	4
{C}	4
{D}	1
{E}	4

Since $\{D\}$ has a support of only 1, we can eliminate it from the search. Applying the *AS* strategy to the remaining four elements gives us the following subsets for level two:

Level 2:

sets	support
{A,B}	2
{A,C}	3
{A,E}	2
{B,C}	3
{B,E}	4
{C,E}	3

It is obvious that sets $\{A, B\}$ and $\{A, E\}$ can be eliminated, as they both have insufficient support. However, it is also possible to remove sets $\{A, C\}$ and $\{B, E\}$. This is because both have the same support as one of their subsets (e.g., $\{A, C\}$ has a support of 3 and so does $\{A\}$). If a set has the same support as as one of its subsets, it cannot be a minimal generator (why?).

At this point there are only two remaining sets, $\{B, C\}$ and $\{C, E\}$. The *AS* method requires that the next level has sets composed of two sets from the previous level which match in all but their last letters. Since there can be no such sets, the step ends.

3.1.3 Deriving Closed Sets

The second step of the *A-Close* algorithm involves taking the generators found in the first step and inputting them into $f(x)$ to obtain the closed sets. The results of this step (broken down into each inner function call of $f(x)$ for clarity) for our running example are shown below:

Level 2:

generators	$i(generators)$	$t(i(generators))$
{A}	{1,3,5}	{A,C}
{B}	{2,3,4,5}	{B,E}
{C}	{1,2,3,5}	{C}
{E}	{2,3,4,5}	{B,E}
{B,C}	{2,3,5}	{B,C,E}
{C,E}	{2,3,5}	{B,C,E}

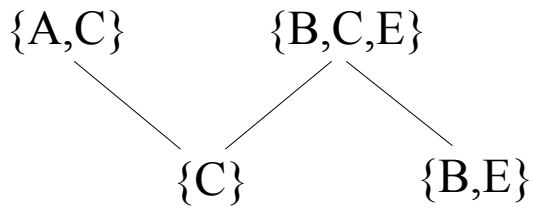

Figure 2: The relationships between the discovered closed itemsets.

After removing duplicates, we find four closed sets: $\{A, C\}$, $\{B, E\}$, $\{B, C, E\}$, and $\{C\}$.[2]

3.2 The CHARM Algorithm

Although the *A-Close* algorithm does sufficiently find the desired closed sets, it is not as efficient as other algorithms. In particular, the *CHARM* [4] algorithm is a more efficient approach to solving this problem. However *CHARM* is much more complicated than *A-Close* and, as such, is left as optional reading. Maybe we will come back to it later in the course.

3.3 Relationships Between Closed Sets

Once the desired closed itemsets are found, we can construct a lattice representing their relationships. Figure 3.3 shows the lattice for our example.

Question: The example closed itemsets form a connected graph. Will this be true for all closed itemsets?

Reconstructing the subset relationship that exists among the closed itemsets is non-trivial. However, algorithms have been developed which are capable of performing this reconstruction efficiently. Most notable among these algorithms is the *CHARM-L* algorithm [5].[3]

4 Finding Maximal Itemsets

Often it is preferrable to find the maximal itemsets rather than the closed itemsets. In this case, it would be inefficent to simply use the *A-Close* or *CHARM* algorithms to find the closed itemsets and then derive the maximal itemsets. Thus, a new algorithm called *Pincer Search* [2] is introduced as a way to find maximal sets.

[2] $\{D\}$ did not make the list of closed sets since it was eliminated in the first level of the generator search due to lack of support. However, if support were not a concern (as is often the case for mathematicians), we would find that some set involving $\{D\}$ is also a closed set.

[3] While the paper describing the original *CHARM* algorithm is not required reading, the paper describing the more involved *CHARM-L* algorithm is apparently going to be, as one of the co-authors is the instructor, and he is going to arm twist us. How considerate.

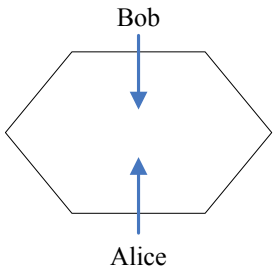

Figure 3: Alice and Bob performing the Pincer Search.

4.1 Pincer Search

We know that the lattice of sets is shaped in a hexagon with the widest portion in the middle, as shown in Figure 4.1. In *Pincer Search*, Alice is searching from the bottom of the lattice (i.e., starting with the smallest sized itemsets and increasing) and Bob is searching from the top of the lattice (i.e., starting with the largest sized itemsets and decreasing). By coordinating their efforts, Alice and Bob can utilize the results of each others' searches to prune their own search and reduce the time it takes to find the border representing the maximal itemsets.

4.2 Example

If Alice is searching the second level and discovers that the set $\{A, B\}$ is frequent, Bob does not benefit. However, if $\{A, B\}$ is infrequent, Bob can immediately prune all sets containing $\{A, B\}$, thereby narrowing his search space significantly.

Conversely, if Bob is searching and discovers that the set $\{A, B, C, D, E\}$ is infrequent, Alice does not benefit as it is not clear which subsets can be ruled out. However, if $\{A, B, C, D, E\}$ is frequent, Alice can prune her search of all subsets of $\{A, B, C, D, E\}$ since they must all be frequent.

5 Food for Thought

There are two parts involved in association rule mining:

1. Find the frequent itemsets that satisfy a given support threshold.
2. For each frequent itemset, identify confident rules.

So far, we have focused purely on finding frequent itemsets and have ignored the second objective. Traditionally in data mining the second problem is not considered as often as the first problem. Why is this so?

Hint: Given an itemset of size k, $\{i_1, i_2, ..., i_k\}$, think of a level-wise algorithm which finds confident rules derived from this itemset.

The next class will discuss *rules* in detail. There is a whole theory for them.

References

[1] Rakesh Agrawal, Ramakrishnan Srikant, *Fast Algorithms for Mining Association Rules in Large Databases*, Proceedings of the 20th International Conference on Very Large Data Bases (VLDB'94), pages 487-499, 1994.

[2] Dao-I Lin, Zvi M. Kedem. *Pincer Search: A New Algorithm for Discovering the Maximum Frequent Set*, in Advances in Database Technology - EDBT'98, 6th International Conference on Extending Database Technology, 1998, pages 105-119, 1998.

[3] Nicolas Pasquier, Yves Bastide, Rafik Taouil, Lotfi Lakhal. *Discovering Frequent Closed Itemsets for Association Rules*, in Proceedings of ICDT'99, pages 398-416, 1999.

[4] Mohammed Zaki, Ching-Jui Hsiao. *CHARM: An Efficient Algorithm for Closed Itemset Mining*, in Proceedings of the Second SIAM International Conference on Data Mining, 2002.

[5] Mohammed Zaki, Naren Ramakrishnan. *Reasoning about Sets using Redescription Mining*, in Proceeding of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining (KDD'05), pages 364-373, 2005.