CS 6604: Data Mining Fall 2007

Lecture 7 — September 10, 2007

Lecture: Naren Ramakrishnan *Scribe: Patrick Butler*

1 Overview

In this lecture we discuss constraints and constraint mining. We begin by recalling the two main types of constraints we already know so far: monotonic and anti-monotonic. Then we shall define what are called convertible constraints, which are essentially constraints that can be 'converted' into monotonic or anti-monotonic form if we formulate the problem correctly. Finally we shall present 'succinct' constraints and how all four fit into the problem space.

2 Homework Questions Clarifications

2.1 Lattice of booleans

There is a total of 16 possibilities for this problem, in the case of two variables; now just make the lattice and you're done!

2.2 Data Mining Netflix

The database as it is present is not a binary database. It is presented instead as a (people,movie) pairs with ratings. It is left up to you to determine how to convert that into a binary database. For instance, you can think of people as transactions and movies they have rated as items. Alternatively, you can only focus on movies that have been rated highly. There are many possibilities. Also observe that there is likely a skew of really popular movies versus unpopular ones. You can try to take that into account in your data mining.

3 Main Section

3.1 Constraints

$c(\cdot)$: predicate $\rightarrow$ returns true if satisfied false if not.

Definition 1. *Anti-monotone: Given $X \subseteq Y$, if $c(X)$ is not true then $c(Y)$ is not true, i.e., $\neg c(X) \implies \neg c(Y)$.*

Definition 2. *Monotone: Given $X \subset Y$, if $c(X)$ is true then $c(Y)$ is true, i.e., $c(X) \implies c(Y)$.*

Observation 3. *A constraint can be neither, for example: $c(X)$ is true if X contains an even number of elements.*

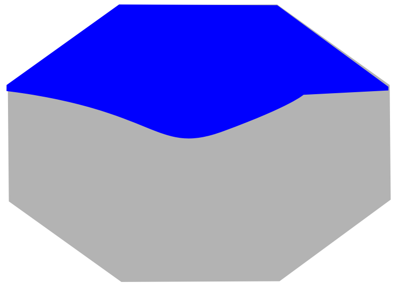

Figure 1: Graphical depiction of a monotonic constraint search.

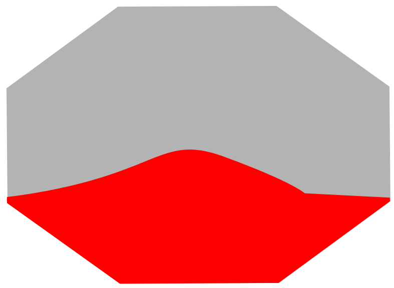

Figure 2: Graphical depiction of a anti-monotonic constraint search.

Anti-monotone (Figure 2) is useful when you are searching bottom up. Monotone (Figure 1 is useful for a top down search, i.e., if you find an itemset violating the monotone property, then everything below it is also violating. An example is the use of the χ^2 constraint for checking correlated occurrences of items.

3.2 Convertible Constraints

There is another type of constraint, namely the convertible constraint. This type of constraint is neither monotonic nor anti-monotonic. However, it can be *made* monotonic or anti-monotonic by surveying the data in a particular way.

Example 4. *Imagine a transaction database:*

Transaction	*items*
t_1	.....
t_2	...
t_3	.

item	*price*
i_1	*20*
i_2	*30*
i_3	*5*
i_4	*1*

Assume the goal is to find $X \subseteq I$ *such that* $avg(X.price) \leq \theta$, *where* $\theta = 10$. *Let us consider some itemsets and whether they satify the constraint:*

- i_1 *doesn't*
- i_1, i_3 *doesn't*
- i_1, i_3, $i_4 < 10$

Thus it is not anti-monotonic.

But then again

- i_3 *does*
- i_1, i_3 *does not*

Thus it is not monotonic either.

Similarly, $avg(X.price) \geq \theta$ *is not either. To make it, say, monotonic, first sort by increasing prices:*

item	*price*
i_4	*1*
i_3	*5*
i_1	*20*
i_2	*30*

To make this work, we must define the notion of a prefix. We now treat itemsets as strings and replace subset relationship by prefix relationship. And we now redefine anti-monotonic and montonic definitions such that: If X *is a prefix of* Y *...*

Next, we make a set of strings in order:

avg(x.price)	*items*
1	i_4
3	i_4, i_3
10.5	i_4, i_1
15.5	i_4, i_2
..	i_4, i_3, i_3
...	...

Presto, observe that it becomes monotonic!

Definition 5. *If a constraint can be converted to an anti-monotonic constraint by the assumption of a suitable order, it is called a* **convertible anti-monotonic** *constraint. Similarly we can define a* **convertible monotonic** *constraint. If we can find orders such that either is possible, we refer to it as just* **convertible**.

Is there a universal order for $var(x, price) <= \theta$? Not really but in the next lecture we will see a small trick that allows us to push the constraint into a levelwise algorithm.

Corollary 6. *A conjunction of monotonic constraints is montonic.*

Corollary 7. *A conjunction of anti-monotonic constraints is anti-monotonic.*

Corollary 8. *A conjunction of monotonic and anti-monotonic constraints is neither.*

In the last case, there are really two borders and we desire to find itemsets that satisfy both (if the borders enclose any itemsets). In the figure here, there are no such itemsets.

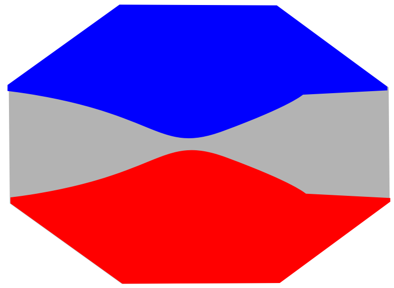

Figure 3: Intersection of a monotonic and an anti-monotonic search.

Example 9. *Now we get a handout woooo! It lists a set of constraints and we are asked to mark whether they are monotonic, anti-monotonic, convertible, etc. Let us take one of them. Is the notion of $v \in S$ monotonic or anti-monotonic?*

Answer: Given $S_1 \subset S_2$
if $v \notin S_1$, $v \notin S_2$ (anti-monotonic) no
if $v \in S_1$, $v \in S_2$ (monotonic) yes

3.3 Succinct Constraints

Definition 10. *$c(\cdot)$ is a succinct constraint if the following holds: If X satisifies c then all others that satisfy C are "derived from" X*

Example 11. $min(S.a) <= v$

3.4 Venn Diagram of all constraints

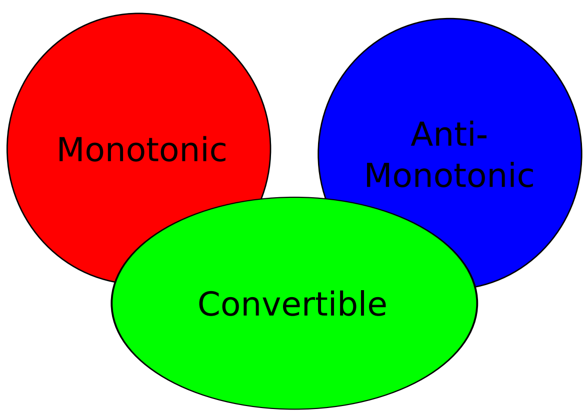

Figure 4: First attempt at the diagram. Where does succinct go?

References

[1] Jian Pei, Jiawei Han. *Can we push more Constraints into Frequent Pattern Mining?*, in KDD '00: Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining, pages 350-354, 2000.