Christopher Morgan
CS478 : Brother Christophe
Oct 30, 2013

Use the Apriori algorithm on the Binarized Lenses problem.

Set *minconf* = 0.8. Start with *minsup* = 0.9 and report your results (i.e., number of rules found and sample rules) for decreasing values of *minsup* using decrements of 0.05, down to 0.05 (i.e., this means you should run the algorithm 18 times).

Minimum Support Value	Number of Rules Found	Rules
.9	0	
.85	0	
.8	0	
.75	0	
.7	0	
.65	0	
.6	0	
.55	0	
.5	2	1. norm-tearp-rate=0 12 ==> no-lenses=1 12 conf:(1) 2. no-lenses=1 15 ==> norm-tearp-rate=0 12 conf:(0.8)
.45	2	“
.4	2	“
.35	10	1. norm-tearp-rate=0 12 ==> no-lenses=1 12 conf:(1) 2. no-lenses=0 9 ==> norm-tearp-rate=1 9 conf:(1) 3. pre-presbyopic=1 8 ==> young=0 8 conf:(1) 4. presbyopic=1 8 ==> young=0 8 conf:(1) 5. young=1 8 ==> pre-presbyopic=0 8 conf:(1) 6. young=1 8 ==> presbyopic=0 8 conf:(1) 7. presbyopic=1 8 ==> pre-presbyopic=0 8 conf:(1) 8. pre-presbyopic=1 8 ==> presbyopic=0 8 conf:(1) 9. pre-presbyopic=0 presbyopic=1 8 ==> young=0 8 conf:(1) 10. young=0 presbyopic=1 8 ==> pre-presbyopic=0 8 conf:(1)
.3	28	… 11. young=0 pre-presbyopic=0 8 ==> presbyopic=1 8 conf:(1) 12. presbyopic=1 8 ==> young=0 pre-presbyopic=0 8 conf:(1) 13. pre-presbyopic=1 presbyopic=0 8 ==> young=0 8 conf:(1) 14. young=0 presbyopic=0 8 ==> pre-presbyopic=1 8 conf:(1) 15. young=0 pre-presbyopic=1 8 ==> presbyopic=0 8 conf:(1) 16. pre-presbyopic=1 8 ==> young=0 presbyopic=0 8 conf:(1) 17. young=0 norm-tearp-rate=0 8 ==> no-lenses=1 8 conf:(1) 18. pre-presbyopic=0 presbyopic=0 8 ==> young=1 8 conf:(1) 19. young=1 presbyopic=0 8 ==> pre-presbyopic=0 8 conf:(1) 20. young=1 pre-presbyopic=0 8 ==> presbyopic=0 8 conf:(1)

.25	41	…
.2	53	…
.15	310	…
.1	1252	…
.05	2769	…

Summarize your findings.

The data suggests that highest degree of support that this data exhibit is 0.5. In other words, once support is increased passed 0.5 there are no rules that can be generated that have support that high. The data also suggests that the number of rules with a given minimum support increase exponentially as the minimum support decreases.

The data also suggest that one of the best class predicting rules is norm-tearp-rate=0 => no-lenses=1. The results also show a high correlation between pre-presbyopic=1, presbyopic=1, and young=0.

Run the original Lenses problem against ID3 in Weka (or your own implementation if you prefer). Compare the rules you obtained with Apriori with the tree (or rules) induced by ID3.

```
tear-prod-rate = reduced: none (12.0)
tear-prod-rate = normal
|   astigmatism = no: soft (6.0/1.0)
|   astigmatism = yes
|   |   spectacle-prescrip = myope: hard (3.0)
|   |   spectacle-prescrip = hypermetrope: none (3.0/1.0)
```

At the main branch of the tree, the ID3 tree and Apriori algorithms appears to very strongly agree. The tree suggests that with a tear-rod-rate=0 the classification should be none or no lenses. The Apriori algorithm agrees by producing the rule norm-tearp-rate=0 => no-lenses=1. Both saying that for a reduced tear production rate the classification should be no lenses.

For here the algorithms seem to divert. ID3 continues classification by looking at tear production rate with astigmatism and spectacle prescription, where as the Apriori algorithm made no more class rules and created rules concerning young, pre-prebyopic, and prebyopic.

Use your algorithm on the Mirror Symmetry problem.

Run Apriori for various combinations of *minsup* and *minconf* values.

Minsup	Minconf	Rules
.4	.85	1. Lmost7=0 47 ==> Rmost7=0 40 conf:(0.85)
.35	“	“
.3	“	1. Rmost14=0 Symm=1 31 ==> Lmost14=0 31 conf:(1) 2. Lmost14=0 Symm=1 31 ==> Rmost14=0 31 conf:(1)

		3. Rmost11=0 Symm=1 30 ==> Lmost11=0 30 conf:(1) 4. Lmost11=0 Symm=1 30 ==> Rmost11=0 30 conf:(1) 5. Lmost14=0 Rmost11=0 36 ==> Rmost14=0 32 conf:(0.89) 6. Lmost11=0 Rmost14=0 36 ==> Lmost14=0 31 conf:(0.86) 7. Lmost11=0 Lmost14=0 36 ==> Rmost14=0 31 conf:(0.86) 8. Lmost7=0 47 ==> Rmost7=0 40 conf:(0.85)
.25	"	1. Rmost14=0 Symm=1 31 ==> Lmost14=0 31 conf:(1) 2. Lmost14=0 Symm=1 31 ==> Rmost14=0 31 conf:(1) 3. Rmost11=0 Symm=1 30 ==> Lmost11=0 30 conf:(1) 4. Lmost11=0 Symm=1 30 ==> Rmost11=0 30 conf:(1) 5. Rmost12=1 Symm=1 29 ==> Lmost12=1 29 conf:(1) 6. Lmost12=1 Symm=1 29 ==> Rmost12=1 29 conf:(1) 7. Rmost=0 Symm=1 27 ==> Lmost=0 27 conf:(1) 8. Lmost=0 Symm=1 27 ==> Rmost=0 27 conf:(1) 9. Rmost3=1 Symm=1 27 ==> Lmost3=1 27 conf:(1) 10. Lmost3=1 Symm=1 27 ==> Rmost3=1 27 conf:(1)
.4	.8	1. Lmost7=0 47 ==> Rmost7=0 40 conf:(0.85) 2. Rmost4=1 52 ==> Lmost4=1 43 conf:(0.83) 3. Lmost14=0 60 ==> Rmost14=0 48 conf:(0.8) 4. Rmost5=0 50 ==> Lmost5=0 40 conf:(0.8)
.35	"	1. Lmost7=0 47 ==> Rmost7=0 40 conf:(0.85) 2. Rmost7=1 45 ==> Lmost7=1 38 conf:(0.84) 3. Rmost4=1 52 ==> Lmost4=1 43 conf:(0.83) 4. Rmost1=0 48 ==> Lmost1=0 39 conf:(0.81) 5. Lmost14=0 60 ==> Rmost14=0 48 conf:(0.8) 6. Rmost5=0 50 ==> Lmost5=0 40 conf:(0.8)
.3	"	1. Rmost14=0 Symm=1 31 ==> Lmost14=0 31 conf:(1) 2. Lmost14=0 Symm=1 31 ==> Rmost14=0 31 conf:(1) 3. Rmost11=0 Symm=1 30 ==> Lmost11=0 30 conf:(1) 4. Lmost11=0 Symm=1 30 ==> Rmost11=0 30 conf:(1) 5. Lmost14=0 Rmost11=0 36 ==> Rmost14=0 32 conf:(0.89) 6. Lmost11=0 Rmost14=0 36 ==> Lmost14=0 31 conf:(0.86) 7. Lmost11=0 Lmost14=0 36 ==> Rmost14=0 31 conf:(0.86) 8. Lmost7=0 47 ==> Rmost7=0 40 conf:(0.85) 9. Rmost7=1 45 ==> Lmost7=1 38 conf:(0.84) 10. Lmost4=1 Rmost14=0 36 ==> Lmost14=0 30 conf:(0.83)
.25	"	"
.4	.75	1. Lmost7=0 47 ==> Rmost7=0 40 conf:(0.85) 2. Rmost4=1 52 ==> Lmost4=1 43 conf:(0.83) 3. Lmost14=0 60 ==> Rmost14=0 48 conf:(0.8) 4. Rmost5=0 50 ==> Lmost5=0 40 conf:(0.8) 5. Rmost14=0 61 ==> Lmost14=0 48 conf:(0.79) 6. Rmost3=1 58 ==> Lmost3=1 45 conf:(0.78) 7. Lmost3=1 58 ==> Rmost3=1 45 conf:(0.78) 8. Lmost11=0 53 ==> Rmost11=0 41 conf:(0.77) 9. Lmost5=0 52 ==> Rmost5=0 40 conf:(0.77)

		10. Rmost11=0 54 ==> Lmost11=0 41 conf:(0.76)
.35	“	“
.3	“	“
.25	“	“
.4	.7	1. Lmost7=0 47 ==> Rmost7=0 40 conf:(0.85) 2. Rmost4=1 52 ==> Lmost4=1 43 conf:(0.83) 3. Lmost14=0 60 ==> Rmost14=0 48 conf:(0.8) 4. Rmost5=0 50 ==> Lmost5=0 40 conf:(0.8) 5. Rmost14=0 61 ==> Lmost14=0 48 conf:(0.79) 6. Rmost3=1 58 ==> Lmost3=1 45 conf:(0.78) 7. Lmost3=1 58 ==> Rmost3=1 45 conf:(0.78) 8. Lmost11=0 53 ==> Rmost11=0 41 conf:(0.77) 9. Lmost5=0 52 ==> Rmost5=0 40 conf:(0.77) 10. Rmost11=0 54 ==> Lmost11=0 41 conf:(0.76)
.35	“	“
.3	“	“
.25	“	“

Summarize your findings.

The data suggests that for a truly confident result the minimum support threshold is around 0.3. With higher levels of support we’re unable to obtain rules with confidence levels greater than .85. Therefore as our confidence is held high and the support threshold lowered, the rules become more confident and we can find more rules. Also, as the confidence threshold is lowered we are able to find more rules.

The rules found suggest that when symm == 1 we can find rules that say Lmost(n) == Rmost(n). Without this constraint the rules follow a similar pattern but with less certainty for each rule.

This is an artificial problem. Each attribute represents a bit position in a string of 30 bits: *Lmost, Lmost1, ..., Lmost14, Rmost14, Rmost13, ..., Rmost1, Rmost* and the attribute *Symm* is 1 if the pattern is symmetric about its center, and 0 otherwise. Given this interpretation, do any of the rules discovered by your Apriori algorithm make sense?

The rules that contain a symm=1 on the left hand side of the rule make a lot of sense. Because if the number is symmetric then it is a known fact that Lmost(n) == Rmost(n). However, the rules without this symm variable on the left don’t really make any sense.