16
Association Rule Mining I

16.1 Introduction

Classification rules are concerned with predicting the value of a categorical attribute that has been identified as being of particular importance. In this chapter we go on to look at the more general problem of finding *any* rules of interest that can be derived from a given dataset.

We will restrict our attention to IF ... THEN ... rules that have a conjunction of 'attribute = value' terms on both their left- and right-hand sides. We will also assume that all attributes are categorical (continuous attributes can be dealt with by discretising them 'globally' before any of the methods discussed here are used).

Unlike classification, the left- and right-hand sides of rules can potentially include tests on the value of any attribute or combination of attributes, subject only to the obvious constraints that at least one attribute must appear on both sides of every rule and no attribute may appear more than once in any rule. In practice data mining systems often place restrictions on the rules that can be generated, such as the maximum number of terms on each side.

If we have a financial dataset one of the rules extracted might be as follows:

IF Has-Mortgage = yes AND Bank_Account_Status = In_credit
THEN Job_Status = Employed AND Age_Group = Adult_under_65

Rules of this more general kind represent an *association* between the values of certain attributes and those of others and are called *association rules*. The

M. Bramer, *Principles of Data Mining*, Undergraduate Topics in Computer Science, DOI 10.1007/978-1-4471-7307-6_16

process of extracting such rules from a given dataset is called *association rule mining* (ARM). The term *generalised rule induction* (or GRI) is also used, by contrast with classification rule induction. (Note that if we were to apply the constraint that the right-hand side of a rule has to have only one term which must be an attribute/value pair for a designated categorical attribute, association rule mining would reduce to induction of classification rules.)

For a given dataset there are likely to be few if any association rules that are exact, so we normally associate with each rule a *confidence* value, i.e. the proportion of instances matched by its left- and right-hand sides combined as a proportion of the number of instances matched by the left-hand side on its own. This is the same measure as the predictive accuracy of a classification rule, but the term 'confidence' is more commonly used for association rules.

Association Rule Mining algorithms need to be able to generate rules with confidence values less than one. However the number of possible Association Rules for a given dataset is generally very large and a high proportion of the rules are usually of little (if any) value. For example, for the (fictitious) financial dataset mentioned previously, the rules would include the following (no doubt with very low confidence):

IF Has-Mortgage = yes AND Bank_Account_Status = In_credit
THEN Job_Status = Unemployed

This rule will almost certainly have a very low confidence and is obviously unlikely to be of any practical value.

The main difficulty with association rule mining is computational efficiency. If there are say 10 attributes, each rule can have a conjunction of up to nine 'attribute = value' terms on the left-hand side. Each of the attributes can appear with any of its possible values. Any attribute not used on the left-hand side can appear on the right-hand side, also with any of its possible values. There are a very large number of possible rules of this kind. Generating all of these is very likely to involve a prohibitive amount of computation, especially if there are a large number of instances in the dataset.

For a given unseen instance there are likely to be several or possibly many rules, probably of widely varying quality, predicting different values for any attributes of interest. A conflict resolution strategy of the kind discussed in Chapter 11 is needed that takes account of the predictions from all the rules, plus information about the rules and their quality. However we will concentrate here on rule generation, not on conflict resolution.

16.2 Measures of Rule Interestingness

In the case of classification rules we are generally interested in the quality of a rule set as a whole. It is all the rules working in combination that determine the effectiveness of a classifier, not any individual rule or rules.

In the case of association rule mining the emphasis is on the quality of each individual rule. A single high quality rule linking the values of attributes in a financial dataset or the purchases made by a supermarket customer, say, may be of significant commercial value.

To distinguish between one rule and another we need some measures of rule quality. These are generally known as *rule interestingness measures.* The measures can of course be applied to classification rules as well as association rules if desired.

Several interestingness measures have been proposed in the technical literature. Unfortunately the notation used is not yet very well standardised, so in this book we will adopt a notation of our own for all the measures described.

In this section we will write a rule in the form

if LEFT then RIGHT

We start by defining four numerical values which can be determined for any rule simply by counting:

N_{LEFT} Number of instances matching LEFT
N_{RIGHT} Number of instances matching RIGHT
N_{BOTH} Number of instances matching both LEFT and RIGHT
N_{TOTAL} Total number of instances

We can depict this visually by a figure known as a *Venn diagram.* In Figure 16.1 the outer box can be envisaged as containing all N_{TOTAL} instances under consideration. The left- and right-hand circles contain the N_{LEFT} instances that match LEFT and the N_{RIGHT} instances that match RIGHT, respectively. The hashed area where the circles intersect contains the N_{BOTH} instances that match both LEFT and RIGHT.

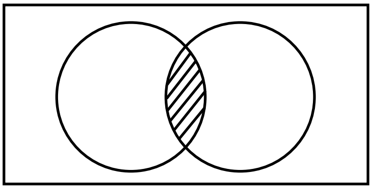

Figure 16.1 Instances matching LEFT, RIGHT and both LEFT and RIGHT

The values N_{LEFT}, N_{RIGHT}, N_{BOTH} and N_{TOTAL} are too basic to be considered as rule interestingness measures themselves but the values of most (perhaps all) interestingness measures can be computed from them.

Three commonly used measures are given in Figure 16.2 below. The first has more than one name in the technical literature.

Confidence (Predictive Accuracy, Reliability)
N_{BOTH} / N$_{LEFT}$
The proportion of right-hand sides predicted by the rule that are correctly predicted

Support
N_{BOTH}/N_{TOTAL}
The proportion of the training set correctly predicted by the rule

Completeness
N_{BOTH}/N_{RIGHT}

The proportion of the matching right-hand sides that are correctly predicted by the rule

Figure 16.2 Basic Measures of Rule Interestingness

We can illustrate this using the financial rule given in Section 16.1.

IF Has-Mortgage = yes AND Bank_Account_Status = In_credit
THEN Job_Status = Employed AND Age_Group = Adult_under_65

Assume that by counting we arrive at the following values:

$N_{LEFT} = 65$
$N_{RIGHT} = 54$
$N_{BOTH} = 50$
$N_{TOTAL} = 100$

From these we can calculate the values of the three interestingness measures given in Figure 16.2.

Confidence = $N_{BOTH}/N_{LEFT} = 50/65 = 0.77$
Support = $N_{BOTH}/N_{TOTAL} = 50/100 = 0.5$
Completeness = $N_{BOTH}/N_{RIGHT} = 50/54 = 0.93$

The confidence of the rule is 77%, which may not seem very high. However it correctly predicts for 93% of the instances in the dataset that match the right-hand side of the rule and the correct predictions apply to as much as 50% of the dataset. This seems like a valuable rule.

Amongst the other measures of interestingness that are sometimes used is *discriminability*. This measures how well a rule discriminates between one class and another. It is defined by:

$1 - (N_{LEFT} - N_{BOTH})/(N_{TOTAL} - N_{RIGHT})$

which is

1 − (number of misclassifications produced by the rule)/(number of instances with other classifications)

If the rule predicts perfectly, i.e. $N_{LEFT} = N_{BOTH}$, the value of discriminability is 1.

For the example given above, the value of discriminability is

$1 - (65 - 50)/(100 - 54) = 0.67.$

16.2.1 The Piatetsky-Shapiro Criteria and the RI Measure

In an influential paper [1] the American researcher Gregory Piatetsky-Shapiro proposed three principal criteria that should be met by any rule interestingness measure. The criteria are listed in Figure 16.3 and explained in the text that follows.

Criterion 1
The measure should be zero if $N_{BOTH} = (N_{LEFT} \times N_{RIGHT})/N_{TOTAL}$
Interestingness should be zero if the antecedent and the consequent are statistically independent (as explained below).

Criterion 2
The measure should increase monotonically with N_{BOTH}

Criterion 3
The measure should decrease monotonically with each of N_{LEFT} and N_{RIGHT}

For criteria 2 and 3, it is assumed that all other parameters are fixed.

Figure 16.3 Piatetsky-Shapiro Criteria for Rule Interestingness Measures

The second and third of these are more easily explained than the first.

Criterion 2 states that if everything else is fixed the more right-hand sides that are correctly predicted by a rule the more interesting it is. This is clearly reasonable.

Criterion 3 states that if everything else is fixed

(a) the more instances that match the left-hand side of a rule the less interesting it is.

(b) the more instances that match the right-hand side of a rule the less interesting it is.

The purpose of (a) is to give preference to rules that correctly predict a given number of right-hand sides from as few matching left-hand sides as possible (for a fixed value of N_{BOTH}, the smaller the value of N_{LEFT} the better).

The purpose of (b) is to give preference to rules that predict right-hand sides that are relatively infrequent (because predicting common right-hand sides is easier to do).

Criterion 1 is concerned with the situation where the antecedent and the consequent of a rule (i.e. its left- and right-hand sides) are independent. How many right-hand sides would we expect to predict correctly just by chance?

We know that the number of instances in the dataset is N_{TOTAL} and that the number of those instances that match the right-hand side of the rule is N_{RIGHT}. So if we just predicted a right-hand side without any justification whatever we would expect our prediction to be correct for N_{RIGHT} instances out of N_{TOTAL}, i.e. a proportion of N_{RIGHT}/N_{TOTAL} times.

If we predicted the same right-hand side N_{LEFT} times (one for each instance that matches the left-hand side of the rule), we would expect that purely by chance our prediction would be correct $N_{LEFT} \times N_{RIGHT}/N_{TOTAL}$ times.

By definition the number of times that the prediction actually turns out to be correct is N_{BOTH}. So Criterion 1 states that if the number of correct predictions made by the rule is the same as the number that would be expected by chance the rule interestingness is zero.

Piatetsky-Shapiro proposed a further rule interestingness measure called RI, as the simplest measure that meets his three criteria. This is defined by:

$RI = N_{BOTH} - (N_{LEFT} \times N_{RIGHT}/N_{TOTAL})$

RI measures the difference between the actual number of matches and the expected number if the left- and right-hand sides of the rule were independent. Generally the value of RI is positive. A value of zero would indicate that the rule is no better than chance. A negative value would imply that the rule is less successful than chance.

The RI measure satisfies all three of Piatetsky-Shapiro's criteria.

Criterion 1 RI is zero if $N_{BOTH} = (N_{LEFT} \times N_{RIGHT})/N_{TOTAL}$

Criterion 2 RI increases monotonically with N_{BOTH} (assuming that all other parameters are fixed).
Criterion 3 RI decreases monotonically with each of N_{LEFT} and N_{RIGHT} (assuming that all other parameters are fixed).

Although doubts have been expressed about the validity of the three criteria and much research in this field remains to be done, the RI measure remains a valuable contribution in its own right.

There are several other rule interestingness measures available. Some important ones are described later in this chapter and in Chapter 17.

16.2.2 Rule Interestingness Measures Applied to the *chess* Dataset

Although Rule Interestingness Measures are particularly valuable for association rules, we can also apply them to classification rules if we wish.

The unpruned decision tree derived from the *chess* dataset (with attribute selection using entropy) comprises 20 rules. One of these (numbered rule 19 in Figure 16.4) is

IF inline = 1 AND wr_bears_bk = 2 THEN Class = safe

For this rule

$N_{LEFT} = 162$
$N_{RIGHT} = 613$
$N_{BOTH} = 162$
$N_{TOTAL} = 647$

So we can calculate the values of the various rule interestingness measures as follows:

Confidence $= 162/162 = 1$
Completeness $= 162/613 = 0.26$
Support $= 162/647 = 0.25$
Discriminability $= 1 - (162 - 162)/(647 - 613) = 1$
RI $= 162 - (162 \times 613/647) = 8.513$

The 'perfect' values of confidence and discriminability are of little value here. They always occur when rules are extracted from an unpruned classification tree (created without encountering any clashes in the training data). The *RI* value indicates that the rule can be expected to correctly predict 8.513 more correct classifications (on average) than would be expected by chance.

Rule	N_{LEFT}	N_{RIGHT}	N_{BOTH}	Conf	Compl	Supp	Discr	RI
1	2	613	2	1.0	0.003	0.003	1.0	0.105
2	3	34	3	1.0	0.088	0.005	1.0	2.842
3	3	34	3	1.0	0.088	0.005	1.0	2.842
4	9	613	9	1.0	0.015	0.014	1.0	0.473
5	9	613	9	1.0	0.015	0.014	1.0	0.473
6	1	34	1	1.0	0.029	0.002	1.0	0.947
7	1	613	1	1.0	0.002	0.002	1.0	0.053
8	1	613	1	1.0	0.002	0.002	1.0	0.053
9	3	34	3	1.0	0.088	0.005	1.0	2.842
10	3	34	3	1.0	0.088	0.005	1.0	2.842
11	9	613	9	1.0	0.015	0.014	1.0	0.473
12	9	613	9	1.0	0.015	0.014	1.0	0.473
13	3	34	3	1.0	0.088	0.005	1.0	2.842
14	3	613	3	1.0	0.005	0.005	1.0	0.158
15	3	613	3	1.0	0.005	0.005	1.0	0.158
16	9	34	9	1.0	0.265	0.014	1.0	8.527
17	9	34	9	1.0	0.265	0.014	1.0	8.527
18	81	613	81	1.0	0.132	0.125	1.0	4.257
19	162	613	162	1.0	0.264	0.25	1.0	8.513
20	324	613	324	1.0	0.529	0.501	1.0	17.026

$N_{TOTAL} = 647$

Figure 16.4 Rule Interestingness Values for Rules Derived from *chess* Dataset

The table of interestingness values of all 20 classification rules derived from the *chess* dataset, given as Figure 16.4, is very revealing.

Judging by the RI values, it looks as if only the last five rules are really of any interest. They are the only rules (out of 20) that correctly predict the classification for at least four instances more than would be expected by chance. Rule 20 predicts the correct classification 324 out of 324 times. Its support value is 0.501, i.e. it applies to over half the dataset, and its completeness value is 0.529. By contrast, Rules 7 and 8 have RI values as low as 0.053, i.e. they predict only slightly better than chance.

Ideally we would probably prefer only to use rules 16 to 20. However in the case of classification rules we cannot just discard the other 15 much lower quality rules. If we do we will have a tree with only five branches that is unable to classify 62 out of the 647 instances in the dataset. This illustrates the general point that an effective classifier (set of rules) can include a number of rules that are themselves of low quality.

16.2.3 Using Rule Interestingness Measures for Conflict Resolution

We can now return briefly to the subject of conflict resolution, when several rules predict different values for one or more attributes of interest for an unseen test instance. Rule interestingness measures give one approach to handling this. For example we might decide to use only the rule with the highest interestingness value, or the most interesting three rules, or more ambitiously we might decide on a 'weighted voting' system that adjusts for the interestingness value or values of each rule that fires.

16.3 Association Rule Mining Tasks

The number of generalised rules that can be derived from a given dataset is potentially very large and in practice the aim is usually either to find all the rules satisfying a specified criterion or to find the best N rules. The latter will be discussed in the next section.

As a criterion for accepting a rule we could use a test on the confidence of the rule, say 'confidence > 0.8', but this is not completely satisfactory. It is quite possible that we can find rules that have a high level of confidence but are applicable very rarely. For example with the financial example used before we might find the rule

IF Age_Group = Over_seventy AND Has-Mortgage = no
THEN Job_Status = Retired

This may well have a high confidence value but is likely to correspond to very few instances in the dataset and thus be of little practical value. One way of avoiding such problems is to use a second measure. One frequently used is support. The value of support is the proportion of the instances in the dataset to which the rule (successfully) applies, i.e. the proportion of instances matched by the left- and right-hand sides together. A rule that successfully applied to only 2 instances in a dataset of 10,000 would have a low value of support (just 0.0002), even if its confidence value were high.

A common requirement is to find all rules with confidence and support above specified threshold values. A particularly important type of association rule application for which this approach is used is known as *market basket analysis*. This involves analysing very large datasets of the kind collected by supermarkets, telephone companies, banks etc. about their customers' transactions (purchases, calls made, etc.) to find rules that, in the supermarket case, find associations between the products purchased by customers. Such datasets are generally handled by restricting attributes to having only the values true or false (indicating the purchase or non-purchase of some product, say) and restricting the rules generated to ones where every attribute included in the rule has the value true.

Market basket analysis will be discussed in detail in Chapter 17.

16.4 Finding the Best N Rules

In this section we will look at a method of finding the best N rules that can be generated from a given dataset. We will assume that the value of N is a small number such as 20 or 50.

We first need to decide on some numerical value that we can measure for any rule which captures what we mean by 'best'. We will call this a *quality measure.* In this section we will use a quality measure (or measure of rule interestingness) known as the *J-measure.*

Next we need to decide on some set of rules in which we are interested. This could be all possible rules with a conjunction of 'attribute = value' terms on both the left- and right-hand sides, the only restriction being that no attribute may appear on both sides of a rule. However a little calculation shows that for even as few as 10 attributes the number of possible rules is huge and in practice we may wish to restrict the rules of interest to some smaller (but possibly still very large) number. For example we might limit the rule 'order', i.e. the number of terms on the left-hand side, to no more than four (say) and possibly also place restrictions on the right-hand side, for example a maximum of two terms or only a single term or even only terms involving a single specified attribute. We will call the set of possible rules of interest the *search space.*

Finally we need to decide on a way of generating the possible rules in the search space in an efficient order, so that we can calculate the quality measure for each one. This is called a *search strategy.* Ideally we would like to find a search strategy that avoids having to generate low-quality rules if possible.

As rules are generated we maintain a table of the best N rules so far found and their corresponding quality measures in descending numerical order. If a new rule is generated that has a quality measure greater than the smallest value in the table the Nth best rule is deleted and the new rule is placed in the table in the appropriate position.

16.4.1 The J-Measure: Measuring the Information Content of a Rule

The J-measure was introduced into the data mining literature by Smyth and Goodman [2], as a means of quantifying the information content of a rule that is soundly based on theory. Justifying the formula is outside the scope of this book, but calculating its value is straightforward.

Given a rule of the form **If** $Y = y$, **then** $X = x$ using Smyth and Goodman's notation, the information content of the rule, measured in bits of information, is denoted by $J(X; Y = y)$, which is called the *J-measure* for the rule.

The value of the J-measure is the product of two terms:

- $p(y)$ The probability that the left-hand side (antecedent) of the rule will occur
- $j(X; Y = y)$ The *j-measure* (note the small letter 'j') or *cross-entropy.*

The cross-entropy term is defined by the equation:

$$j(X; Y = y) = p(x|y).\log_2\left(\frac{p(x|y)}{p(x)}\right) + (1 - p(x|y)).\log_2\left(\frac{1 - p(x|y)}{1 - p(x)}\right)$$

The value of cross-entropy depends on two values:

- $p(x)$ The probability that the right-hand side (consequent) of the rule will be satisfied if we have no other information (called the *a priori* probability of the rule consequent)
- $p(x|y)$ The probability that the right-hand side of the rule will be satisfied if we know that the left-hand side is satisfied (read as 'probability of x given y').

A plot of the j-measure for various values of $p(x)$ is given in Figure 16.5.

In terms of the basic measures introduced in Section 16.2:

$p(y) = N_{LEFT}/N_{TOTAL}$
$p(x) = N_{RIGHT}/N_{TOTAL}$
$p(x|y) = N_{BOTH}/N_{LEFT}$

The J-measure has two helpful properties concerning upper bounds. First, it can be shown that the value of $J(X; Y = y)$ is less than or equal to

$p(y).\log_2(\frac{1}{p(y)})$.

The maximum value of this expression, given when $p(y) = 1/e$, is $\log_2 e/e$, which is approximately 0.5307 bits.

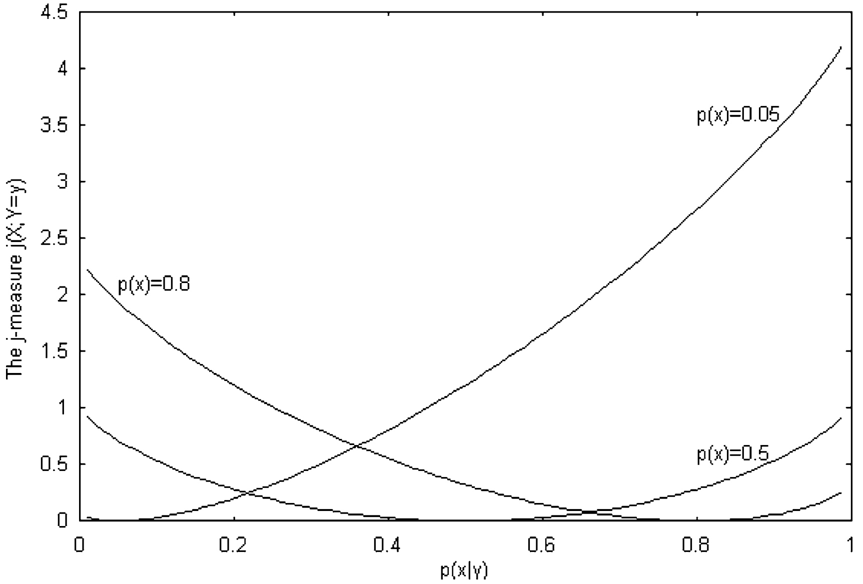

Figure 16.5 Plot of j-Measure for Various Values of $p(x)$

Second (and more important), it can be proved that the J value of any rule obtained by *specialising* a given rule by adding further terms is bounded by the value

$$Jmax = p(y). \max\{p(x|y). \log_2(\tfrac{1}{p(x)}), (1 - p(x|y)). \log_2(\tfrac{1}{1-p(x)})\}$$

Thus if a given rule is known to have a J value of, say, 0.352 bits and the value of $Jmax$ is also 0.352, there is no benefit to be gained (and possibly harm to be done) by adding further terms to the left-hand side, as far as information content is concerned.

We will come back to this topic in the next section.

16.4.2 Search Strategy

There are many ways in which we can search a given search space, i.e. generate all the rules of interest and calculate their quality measures. In this section we will describe a method that takes advantage of the properties of the J-measure.

To simplify the description we will assume that there are ten attributes $a1$, $a2$, ..., $a10$ each with three possible values 1, 2 and 3. The search space comprises rules with just one term on the right-hand side and up to nine terms on the left-hand side.

We start by generating all possible right-hand sides. There are 30 of them, i.e. each of the 10 attributes combined with each of its three values, e.g. $a1 = 1$ or $a7 = 2$.

From these we can generate all possible rules of order one, i.e. with one term on the left-hand side. For each right-hand side, say '$a2 = 2$', there are 27 possible left-hand sides, i.e. the other nine attributes combined with each of their three possible values, and thus 27 possible rules of order one, i.e.

IF $a1 = 1$ THEN $a2 = 2$
IF $a1 = 2$ THEN $a2 = 2$
IF $a1 = 3$ THEN $a2 = 2$
IF $a3 = 1$ THEN $a2 = 2$
IF $a3 = 2$ THEN $a2 = 2$
IF $a3 = 3$ THEN $a2 = 2$

and so on.

We calculate the J-value for each of the 27×30 possible rules. We put the rules with the N highest J-values in the best rule table in descending order of J.

The next step is to specialise the rules of order one to form rules of order two, e.g. to expand

IF $a3 = 3$ THEN $a2 = 2$

to the set of rules

IF $a3 = 3$ AND $a1 = 1$ THEN $a2 = 2$
IF $a3 = 3$ AND $a1 = 2$ THEN $a2 = 2$
IF $a3 = 3$ AND $a1 = 3$ THEN $a2 = 2$
IF $a3 = 3$ AND $a4 = 1$ THEN $a2 = 2$
IF $a3 = 3$ AND $a4 = 2$ THEN $a2 = 2$
IF $a3 = 3$ AND $a4 = 3$ THEN $a2 = 2$

and so on.

We can then go on to generate all rules of order 3 and then all rules of order 4, 5 etc. up to 9. This clearly involves generating a very large number of rules. There are 262,143 possible left-hand sides for each of the 30 possible right-hand sides, making a total of 7,864,290 rules to consider. However, there are two ways in which the process can be made more computationally feasible.

The first is to expand only the best (say) 20 rules of order one with an additional term. The J-values of the resulting rules of order 2 are then calculated and the 'best N rules' table is adjusted as necessary. The best 20 rules of order 2 (whether or not they are in the best N rules table overall) are then expanded by a further term to give rules of order 3 and so on. This technique is known as a *beam search*, by analogy with the restricted width of the beam of a torch.

In this case the *beam width* is 20. It is not necessary for the beam width to be a fixed value. For example it might start at 50 when expanding rules of order one then reduce progressively for rules of higher orders.

It is important to appreciate that using a beam search technique to reduce the number of rules generated is a *heuristic*, i.e. a 'rule of thumb' that is not guaranteed to work correctly in every case. It is not necessarily the case that the best rules of order K are all specialisations of the best rules of order $K-1$.

The second method of reducing the number of rules to be generated is guaranteed always to work correctly and relies on one of the properties of the J-measure.

Let us suppose that the last entry in the 'best N rules table' (i.e. the entry with lowest J-value in the table) has a J-value of 0.35 and we have a rule with two terms, say

IF $a3 = 3$ AND $a6 = 2$ THEN $a2 = 2$

which has a J-value of 0.28.

In general specialising a rule by adding a further term can either increase or decrease its J-value. So even if the order 3 rule

IF $a3 = 3$ AND $a6 = 2$ AND $a8 = 1$ THEN $a2 = 2$

has a lower J-value, perhaps 0.24, it is perfectly possible that adding a fourth term could give a higher J-value that will put the rule in the top N.

A great deal of unnecessary calculation can be avoided by using the *Jmax* value described in Section 16.4.1. As well as calculating the J-value of the rule

IF $a3 = 3$ AND $a6 = 2$ THEN $a2 = 2$

which was given previously as 0.28, let us assume that we also calculate its *Jmax* value as 0.32. This means that no further specialisation of the rule by adding terms to the left-hand side can produce a rule (for the same right-hand side) with a J-value larger than 0.32. This is less than the minimum of 0.35 needed for the expanded form of the rule to qualify for the best N rules table. Hence the order 2 form of the rule can safely be discarded.

Combining a beam search with rule 'pruning' using the *Jmax* value can make generating rules from even quite a large dataset computationally feasible.

In the next chapter we look at the problem of generating association rules for market basket analysis applications, where the datasets are often huge, but the rules take a restricted form.

16.5 Chapter Summary

This chapter looks at the problem of finding any rules of interest that can be derived from a given dataset, not just classification rules as before. This is known as *Association Rule Mining* or *Generalised Rule Induction*. A number of measures of rule interestingness are defined and criteria for choosing between measures are discussed. An algorithm for finding the best N rules that can be generated from a dataset using the J-measure of the information content of a rule and a 'beam search' strategy is described.

16.6 Self-assessment Exercises for Chapter 16

1. Calculate the values of Confidence, Completeness, Support, Discriminability and RI for rules with the following values.

Rule	N_{LEFT}	N_{RIGHT}	N_{BOTH}	N_{TOTAL}
1	720	800	700	1000
2	150	650	140	890
3	1000	2000	1000	2412
4	400	250	200	692
5	300	700	295	817

2. Given a dataset with four attributes w, x, y and z, each with three values, how many rules can be generated with one term on the right-hand side?

References

[1] Piatetsky-Shapiro, G. (1991). Discovery, analysis and presentation of strong rules. In G. Piatetsky-Shapiro & W. J. Frawley (Eds.), *Knowledge discovery in databases* (pp. 229–248). Menlo Park: AAAI Press.

[2] Smyth, P., & Goodman, R. M. (1992). Rule induction using information theory. In G. Piatetsky-Shapiro & W. J. Frawley (Eds.), *Knowledge discovery in databases* (pp. 159–176). Menlo Park: AAAI Press.