Towards the Application of Association Rules for Defeasible Rule Discovery

Guido Governatori
Cooperative Information Systems Research Centre,
Queensland University of Technology,
Brisbane, Queensland, Australia
g.governatori@qut.edu.au

Andrew Stranieri
Donald Berman Laboratory for Information Technology and Law,
La Trobe University,
Bundoora, Victoria, Australia
stranier@cs.latrobe.edu.au

Abstract. In this paper we investigate the feasibility of Knowledge Discovery from Databases (KDD) in order to facilitate the discovery of defeasible rules that represent the ratio decidendi underpinning legal decision making. Moreover we will argue in favour of Defeasible Logic as an appropriate formal system in which the extracted principles should be encoded.

1 Introduction

One of the main and most controversial issues in the legal domain is the identification of the ratio decidendi. In other words, given a collection of similar cases we want to determine the principles (rules) leading to the adjudication of the cases. On the other hand the defeasibility of normative reasoning is a very well established phenomenon, so the rules (principles) extracted from the cases should be defeasible and we have to identify the appropriate non-monotonic reasoning mechanism.

The central aim of this paper is to demonstrate that the field of knowledge discovery from databases (KDD) can be applied to facilitate the discovery of defeasible rules that represent the ratio decidendi underpinning legal decision making. According to [13] KDD techniques can be grouped into four categories;

- Classification. The aim of classification techniques is to group data into predefined categories such as 'pro-plaintiff' or 'pro-defendant'. KDD classification techniques have been applied to law by [33], [32], [35], [20] and [5].
- Clustering. The aim of clustering techniques is to analyze data in order to group the data meaningfully. Clustering techniques have been applied to the law by [29], [27], [19], [31], [8], [34], [24] and [10].
- Series Analysis. The aim of series analysis to discover sequences within the data. Very few studies have been performed that analyze sequences of data in law. The study by [28] is an exception.

Guido Governatori and Andrew Stranieri, 'Towards the Application of Association Rules for Defeasible Rule Discovery' in Bart Verheij, Arno R. Lodder, Ronald P. Loui and Antoinette J. Muntjewerff (eds.), *Legal Knowledge and Information Systems. Jurix 2001: The Fourteenth Annual Conference*. Amsterdam: IOS Press, 2001, pp. 63-75.

- Association. The objective of association techniques is to discover ways in which data elements are associated with other data elements. For example, an association between the gender of litigants and the outcome of their cases may surprise analysts and stimulate hypotheses to explain the phenomena. Association techniques have not been applied to law to the same extent as classification and clustering techniques. [30] have illustrated that association rule generators can highlight interesting associations in a small dataset in family law. In that study, the Apriori algorithm advanced by [1] was applied to suggest hypotheses for future investigation. The present study differs in that it applies association rules generated directly from data to facilitate the discovery of defeasible rules.

One of the main applications of the KDD process is the discovery of rules (principles) from a set of relevant cases. It is indeed customary in Common Law to examine precedents in order to extract the ratio decidendi, and very often the principles leading to the adjudication can be expressed as a set of defeasible rules. However, it is seldom the case that precedents consist only of facts; more often the facts are supplemented by some principles. In some cases the explicitly listed principles are not enough to justify the conclusions. Thus the proposed methodology can be used to identify hidden principles.

In this study we represent legal reasoning using defeasible logic. First of all Defeasible Logic has been developed by Nute [25, 26] over several years with a particular concern about computational efficiency (indeed, its efficiency is linear cf. [22]) and ease of implementation (nowadays several implementations exist [9, 23] and some of them can deal with theories consisting of over 100,000 propositional rules [23]). In [3] it was shown that Defeasible logic is flexible enough to deal with several intuitions of non-monotonic reasoning, and it has been applied to legal reasoning [4] and legal negotiation [16]. Moreover various variants of Defeasible Logic have been characterised in terms of argumentation semantics [17, 18]; thus it is possible to establish correspondences with other non-monotonic systems for legal reasoning.

Defeasible rules are sourced from regulations, statutes, precedents and expert heuristics. This is a very manually intensive task that involves skilled knowledge engineers and experts. As a consequence, the knowledge acquisition phase is typically extensive. The objective of the present study is to apply KDD so as to automatically suggest plausible defeasible rules. This will facilitate knowledge engineer-expert interaction and lead to the development of improved knowledge bases in less time.

Often, in the knowledge acquisition phase, the focus is not on the discovery of rules. The principles are known, but we have to identify the most appropriate rules for the case at hand, especially when the rules are conflicting. In such cases we have to establish a preference order among available rules. KDD techniques explored in this study can help to identify rule preferences.

Another benefit in the application of KDD to facilitate the generation of defeasible rules involves the validation of decision making practice against explicitly given normative codes. Both precedents and statutes are frequently subject to opposite interpretations; thus a method to determine rules from cases can be used to see whether an interpretation corresponds to actual legal practice. Finally the same approach can be applied to the measure the degree of legal efficacy.

[21] describes the application of KDD in the form of inductive logic programming (ILP) to learning in two settings, predictive and descriptive. The aim of an ILP approach for predictive discovery is to induce a theory that explains positive and negative examples and background knowledge. This has been applied to learning default rules by [12]. The descriptive setting usually includes only positive examples and relaxes the strict notion of explanation. This enables ILP to be applied to describe patterns in a dataset that meet criteria such as associativity by enhancing algorithms for generating association rules [11]. Although ILP has been applied to discover default rules and also to enhance the discovery of association rules, ILP is not used in this study. Instead, association rules that are discovered using conventional algorithms are applied to help identify plausible groups of defeasible rules.

The KDD technique known as Association Rules is described in the next section of this paper. Following that, Defeasible Logic is described. In the subsequent section the application of association rules to facilitate the identification of useful defeasible rules will be discussed with examples from a hypothetical data set constructed for the purpose.

2 Association Rules

An association rule identifies a link between two or more attributes in a dataset. A famous, yet unsubstantiated example of an association rule that is generated from a supermarket database of purchases is:

$$Nappies \rightarrow Beer\ (Support = 30\%,\ Confidence = 75\%)$$

The support represents the number of times Nappies and Beer are purchased together as a proportion of the total transactions. The support for Nappies and Beer is the proportion of itemsets (Nappies, Beer) in the dataset, i.e., frequency $(Nappies, Beer)/n$ where an itemset is a group of items such as (Nappies, Beer) within a transaction.

The confidence is interpreted as the percentage of transactions where Beer is purchased given that Nappies are purchased. For example, of the 10 transactions four involved Nappies and, of those 3 also purchased Beer then the confidence is 75%. The confidence of a rule $A \rightarrow B$ is the conditional probability that a transaction contains B given that it contains A and is calculated as $Support(A \cup B)/Support(A)$

An association rule is drawn directly from data. It is not a generalization from the data but merely identifies an association between the purchase of features. As such, association rules are typically used to identify patterns in data-sets that can be used to suggest hypotheses.

Difficulties in the generation of association rules in large datasets involve the combinatorial explosion in the number of rules to be assessed. For example, if a database has only 2 boolean attributes A, B then the support of 6 itemsets representing the following rules need to be calculated:

$$A \rightarrow; A \rightarrow B; A \rightarrow A; B \rightarrow; B \rightarrow A; B \rightarrow B;$$

If the the database has 3 boolean attributes the support of 25 itemsets need to be calculated. In a real world database containing many multi-valued attributes, and large numbers of records, the support calculations become extremely demanding of memory and processor resources.

[1] first described the Apriori algorithm for discovering association rules from large databases. This algorithm reduced the computational complexity by avoiding the exploration

of all rules. This was achieved by inviting the user to specify threshold confidence and support values. If a rule $A, B \rightarrow D$ did not meet the threshold values then the Apriori algorithm avoids an analysis of the rule $A, B, C \rightarrow D$ and all other rules that have A, B in the antecedent.

The application of association rules to facilitate the discovery of defeasible rules advanced in this paper, relies on the calculation of a support for every itemset in a database. For the trials performed in this study, the Apriori implementation by Christian Borgelt (http://www.cs.uni-magdeburg.de/∼borgelt) was used with the support and confidence thresholds set to 0. This did not present a resource problem because the data set was small. However, for the method advanced here to scale to large datasets an approach that can generate the support of all itemsets in real time must be adopted. Fortunately, [15] has recently advanced brute force algorithms that are tractable in large datasets.

3 Defeasible Logic

In this section we describe Defeasible Logic formally following the presentation of [6]. Defeasible logic is a sceptical formalism, meaning that it does not support contradictory conclusions. Instead it seeks to resolve differences. In cases where there is some support for concluding A but also support for concluding $not\ A$ ($\neg A$), the logic does not conclude either of them (thus the name "sceptical"). If the support for A has priority over the support for $\neg A$ then A would be concluded. Sceptical reasoning, in general, is appropriate for the study of normative reasoning.

A set of norms (rules) will be represented as a defeasible theory. A defeasible theory, i.e., a knowledge base in Defeasible Logic, consists of five different kinds of knowledge: facts, strict rules, defeasible rules, defeaters, and a superiority relation defined over the rules.

Facts denote simple pieces of information that are deemed to be true regardless of other knowledge items. A typical fact is that John is a minor: $minor(John)$.

Briefly, strict and defeasible rules are represented, respectively, by expressions of the form $A_1, \ldots, A_n \rightarrow B$ and $A_1, \ldots, A_n \Rightarrow B$, where $A_1, \ldots, A_n$ is a possibly empty set of prerequisites and B is the conclusion of the rule.

Strict rules are rules in the classical sense: whenever the premises of a rule are given, we are allowed to apply the rule and get a conclusion. When the premises are indisputable (e.g., facts) then so is the conclusion. An example of a strict rule is "every minor is a person". Written formally: $minor(X) \rightarrow person(X)$.

Defeasible rules are rules that can be defeated by contrary evidence. An example of such a rule is "every person has the capacity to perform legal acts to the extent that the law does not provide otherwise"; written formally: $person(X) \Rightarrow hasLegalCapacity(X)$.

The idea is that if we know that someone is a person, then we may conclude that he/she has legal capacity, *unless there is other evidence suggesting that he/she has not*.

Defeaters are a special kind of rules. They are used to prevent conclusions not to support them. For example: $WeakEvidence \rightsquigarrow \neg guilty$ This rule states that if pieces of evidence are assessed as weak, then they can prevent the derivation of a "guilty" verdict; on the other hand they cannot be used to support a "not guilty" conclusion.

The *superiority relation* among rules is used to define priorities among rules, that is, where

one rule may override the conclusion of another rule. For example, given the defeasible rules

$$\begin{array}{ll} r: & person(X) \Rightarrow hasLegalCapacity(X) \\ r': & minor(X) \Rightarrow \neg hasLegalCapacity(X) \end{array}$$

which contradict one another, no conclusive decision can be made about whether a minor has legal capacity. But if we introduce a superiority relation $>$ with $r' > r$, then we can indeed conclude that the minor does not have legal capacity.

It turns out that we only need to define the superiority relation over rules with contradictory conclusions. Also notice that a cycle in the superiority relation is counter-intuitive from the knowledge representation perspective. In the above example, it makes no sense to have both $r > r'$ and $r' > r$. Consequently, the defeasible logic we discuss requires an acyclic superiority relation.

Now we present formally defeasible logics. A *rule* r consists of its *antecedents* (or *body*) $A(r)$ which is a finite set of literals, an arrow, and its *consequent* (or *head*) $C(r)$ which is a literal. There are three kinds of arrows, $\rightarrow$, $\Rightarrow$ and $\rightsquigarrow$ which correspond, respectively, to strict rules, defeasible rules and defeaters. Where the body of a rule is empty or consists of one formula only, set notation may be omitted in examples.

Given a set R of rules, we denote the set of all strict rules in R by R_s, the set of strict and defeasible rules in R by R_{sd}, the set of defeasible rules in R by R_d, and the set of defeaters in R by R_{dft}. $R[q]$ denotes the set of rules in R with consequent q.

A *defeasible theory* D is a structure $D = (F, R, >)$ where F is a finite set of facts, R is a finite set of rules, $>$ is a binary relation over R.

A *conclusion* of D is a tagged literal, where a tag is either ∂ or Δ, that may have positive or negative polarity.

$+\Delta q$ which is intended to mean that q is definitely provable in D (i.e., using only strict rules).

$-\Delta q$ which is intended to mean that we have proved that q is not definitely provable in D.

$+\partial q$ which is intended to mean that q is defeasibly provable in D.

$-\partial q$ which is intended to mean that we have proved that q is not defeasibly provable in D.

Basically a conclusion B is supported if there is a rule whose conclusion is B, the prerequisites are either supported or given in the case at hand, and a stronger rule whose conclusion is $not\ B$ has prerequisites that fail to be supported.

Provability is based on the concept of a *derivation* (or *proof*) in $D = R$. A derivation is a finite sequence $P = (P(1), \ldots, P(n))$ of tagged literals satisfying four conditions (which correspond to inference rules for each of the four kinds of conclusion). In the following $P(1..i)$ denotes the initial part of the sequence P of length i.

$+\Delta$:
If $P(i+1) = +\Delta q$ then
$\exists r \in R_s[q]$
$\forall a \in A(r) : +\Delta a \in P(1..i)$

$-\Delta$:
If $P(i+1) = -\Delta q$ then
$\forall r \in R_s[q]$
$\exists a \in A(r) : -\Delta a \in P(1..i)$

The definition of Δ describes just forward chaining of strict rules. For a literal q to be definitely provable we need to find a strict rule with head q, of which all antecedents have been definitely proved previously. And to establish that q cannot be proven definitely we must establish that for every strict rule with head q there is at least one antecedent which has been shown to be non-provable.

Now we turn to the more complex case of defeasible provability. Before giving its formal definition we provide the idea behind such a notion. A defeasible proof of a literal p consists of three phases. In the first phase either a strict or defeasible rule is put forth in order to support a conclusion p; then we consider an attack on this conclusion using the rules for its negation $\neg p$. The attack fails if each rule for $\neg p$ is either discarded (it is possible to prove that part of the antecedent is not defeasibly provable) or if we can provide a stronger counterattack, that is, if there is an applicable strict or defeasible rule stronger than the rule attacking p. It is worth noting that defeaters cannot be used in the last phase.

$+\partial$:
If $P(i+1) = +\partial q$ then either
$+\Delta q \in P(1..i)$ or
(1) $\exists r \in R_{sd}[q] \forall a \in A(r) : +\partial a \in P(1..i)$ and
(2) $-\Delta{\sim}q \in P(1..i)$ and
(3) $\forall s \in R[{\sim}q]$ either
(a) $\exists a \in A(s) : -\partial a \in P(1..i)$ or
(b) $\exists t \in R_{sd}[q]$ such that $\forall a \in A(t) : +\partial a \in P(1..i)$ and $t > s$.

$-\partial$:
If $P(i+1) = -\partial q$ then
$-\Delta q \in P(1..i)$ and
(1) $\forall r \in R_{sd}[q]\ \exists a \in A(r) : -\partial a \in P(1..i)$ or
(2) $+\Delta{\sim}q \in P(1..i)$ or
(3) $\exists s \in R[{\sim}q]$ such that
(a) $\forall a \in A(s) : +\partial a \in P(1..i)$ and
(b) $\forall t \in R_{sd}[q]$ either $\exists a \in A(t) : -\partial a \in P(1..i)$ or $t \not> s$.

Let us work through the condition for $+\partial$, an analogous explanation holds for $-\partial$. To show that q is provable defeasibly we have two choices: (1) We show that q is already definitely provable; or (2) we need to argue using the defeasible part of D as well. In particular, we require that there must be a strict or defeasible rule with head q which can be applied (2.1). But now we need to consider possible "attacks", that is, reasoning chains in support of a complementary of q. To be more specific: to prove q defeasibly we must show that every complementary literal is not definitely provable (2.2). Also (2.3) we must consider the set of all rules for ${\sim}q$ which are not known to be inapplicable (note that here we consider defeaters, too, whereas they could not be used to support the conclusion q; this is in line with the motivation of defeaters. Essentially each such rule s attacks the conclusion q. For q to be provable, each such rule s must be counterattacked by a rule t with head q with the following properties: (i) t must be applicable at this point, and (ii) t must be stronger than s. Thus each attack on the conclusion q must be counterattacked by a stronger rule.

It is worth noting that defeaters can be simulated in term of the other elements of Defeasible Logic [2], thus we can consider theories without defeaters.

To explain the mechanism of defeasible derivations –showing at the same time the appropriateness of Defeasible Logic for normative reasoning– we consider rule 162 of the Australian Civil Aviation Regulations 1988: "When two aircraft are on converging headings at approximately the same height, the aircraft that has the other on its right shall give way, except that (a) power-driven heavier-than-air aircraft shall give way to airships, gliders and

balloons; . . .” This norm can be represented in defeasible logic as follows:

$$r_1 : \neg rightOfWay(Y, X) \Rightarrow rightOfWay(X, Y)$$
$$r_2 : onTheRightOf(X, Y) \Rightarrow rightOfWay(X, Y)$$

The first rule states that, given two aircraft, if one of the aircraft does not have right of way then the other aircraft does, while the second states that the aircraft X has right of way over the aircraft Y if X is on the right of Y;

$$r_3 : \ powerDriven(X), \neg powerDriven(Y) \Rightarrow \neg rightOfWay(X, Y)$$

The idea of the above rules is that a power-driven aircraft does not have right of way over a non-power-driven one.

$$r_4 : balloon(X) \rightarrow \neg powerDriven(X)$$
$$r_5 : glider(X) \rightarrow \neg powerDriven(X)$$

r_4 and r_5 classify balloons and gliders as non-power-driven aircraft and

$$r_6 : \ \Rightarrow powerDriven(X)$$

assumes aircraft to be power-driven unless further information is given. The superiority relation is determined as follows: $r_3 > r_2$ because r_3 is an exception to r_2. By specificity $r_4 > r_6$ and $r_5 > r_6$. Moreover $r_1 > r_3$.

Let us examine the following cases: 1) two aircraft of the same type (power-driven, non-power-driven) are converging 2) a power-driven aircraft and a non-power-driven aircraft are converging. In the first case we can apply r_2 since the prerequisites of r_3 do not hold and we cannot prove the antecedent of r_1. In the second case we can apply r_3 given that both of the two prerequisites hold, and after the conclusion of r_3 has been established we can apply r_1 to derive that the non-power-driven aircraft has the right of way over the power-drive one.

It is worth noting that in this example all the rules have been drawn from the same normative code, and the superiority relation is explicitly given. However, the first is not a serious limitation in so far as it is possible to merge several codes in the same defeasible theory; for the second [7] proposed a variant of Defeasible Logic where the superiority relation is computed dynamically according to some principles encoded as defeasible rules.

4 Identifying Defeasible Rule Sets using Association Rules

Association rules automatically generated from a dataset can facilitate the discovery of strict rules, defeasible rules and rule preferences. We must assume that the data set is a high quality set in that there are no missing values and all values present are correctly recorded (i.e., no noise). The extent to which these assumptions can be violated is left to future research.

In order to illustrate the application of association rules to the discovery of defeasible rule, we adopt a hypothetical sample dataset that relates to the aircraft example illustrated in the previous section. Table 1 represents three records from 36 situations that were regarded as plausible recordings of air traffic data. For example, Case 1 illustrates a situation in which

Case	Right of way	Y power driven	X power driven	position	x Type	y Type
1	rightOfWayX	yPowerDriven	xPowerDriven	X_on_right	Xairship	Yairship
2	rightOfWayY	yPowerDriven	xPowerDriven	Y_on_right	Xairship	Yairship
3	rightOfWayX	yPowerDriven	-xPowerDriven	X_on_right	Xglider	Yairship

Table 1: Sample data for aircraft data set

Itemset	Attribute	Attribute	Attribute	Support
1.	X_on_right			0.49
2.	X_on_right	¬ yPowerDriven		0.22
3.	rightOfWayX	X_on_right		0.405
4.	rightOfWayX	X_on_right	¬ yPowerDriven	0.135
5.	¬ yPowerDriven	X_on_right	rightOfWayX	0

Table 2: Sample itemsets for aircraft data set

aircraft X was granted right of way over Y. In that situation both X and Y were power driven and X was on the right of Y.

As discussed above, association rules are typically deployed to suggest hypotheses with datasets so large that the generation of all possible rules is not normally feasible in real time. Instead, a threshold level of support and confidence are set by the user, so that only those rules that exceed the thresholds are sought. However, for the purposes of this study, the confidence and support thresholds are set to 0 in order to generate all itemsets.

Table 2 illustrates a sample of the 1057 itemsets with up to four antecedent and one consequent attribute (rightOfWay) generated from the aircraft dataset. Itemset 1 in that table indicates that 49% of records in the set included the attribute X_on_right; Itemset 4 reflects that 13.5% of records contains the itemset triplet (rightOfWayX, X_on_right, -yPowerDriven).

Recall that the confidence of a rule is the conditional probability that the consequent will occur given the antecedent has been observed. The confidence of the rule $A \Rightarrow B$ is

$$\frac{Support(A, B)}{Support(A).}$$

Table 3 lists two rules along with their confidence, support and a measure called interest.

According to [14] the "interestingness" of a rule refers to the degree to which a discovered pattern is of interest to the user. The interest metric aims to highlight those rules that are more interesting than others. Various metrics have been advanced but a conventional metric for the calculation of interest for the rule $A \rightarrow B$ is the

$$\frac{Support(A, B)}{Support(A) * Support(B)}.$$

Although the mere generation of association rules may arguably be a useful tool for the engagement of a domain expert in the process of eliciting defeasible rules, the sheer number of rules generated mitigates against this. A closer analysis of the itemsets and rules may go some way toward facilitating the automatic generation of strict and defeasible rules from association rules.

Rule	Rule	Confidence	Support	Interest
ar1.	$X\_on\_right \Rightarrow rightOfWayX$	0.68	0.405	1.15
ar2.	$X\_on\_right, -yPowerDriven \Rightarrow \neg rightOfWayX$	0.62	0.135	1.63

Table 3: Sample rules from aircraft data set

For the association rule $A \rightarrow B$ generated from a non-noisy data-set, to be interpreted as a Strict rule in defeasible reasoning, we assume the confidence of the rule to be 100%. Every time A is observed B is also observed. For example, itemsets $(A, \neg B)$, $(A, C, \neg B)$, are not present in the data-set. We assume the closed world assumption and conclude that these itemsets do not exist because their presence could represent exceptions to the $A \rightarrow B$ rule and render it defeasible.

For the association rule $A \rightarrow B$ generated from a non-noisy data-set, to be interpreted as a defeasible rule the confidence of the rule $A \rightarrow B$ is greater than 0 but less than 100%. On some occasions that A is also observed, B is observed and on others $\neg B$ is observed.

There are typically many rules with a confidence between 0 and 100 so additional effort is required in order to identify rules that may be considered meaningful in a defeasible rule set. To identify plausible defeasible rules we look for groups of rules that represent the following situations: Simple Exceptions; Separate Vs. Aggregate rules; General Vs. Specific rules; and Apparent irrelevance.

A simple exception occurs when a rule is an exception to a base rule that results in the opposite conclusion. Consider two defeasible rules, $ar3: \ A \Rightarrow P$ and $ar4 : A, B \Rightarrow \neg P$. $ar4$ is an exception to $ar3$ because A is associated with P unless in the presence of B, it is associated with $\neg P$. The necessary conditions for this situation to be identified using association rules are:

- the support for the itemsets (A, P), and $(A, B, \neg P)$ is greater than 0. This indicates that there the rules $A \Rightarrow P$ and $A, B \Rightarrow \neg P$ are represented in the dataset
- the support for the itemset (A, B, P) should be 0. This confirms that A, in the presence of B never leads to P.

The rules illustrated in Table 3 meet the criteria of a simple exception group. The rule $ar1$ in that table is taken to be the base rule and $ar2$ is the exception.

If these conditions are met we identify the base rule and determine a rule preference using the "interestingness" metric where the exception rule has a higher preference than the rule. In the example above $ar1$ is the base rule so the preference relation is $ar1 < ar2$.

A variant of simple exception, refers to the situation where there is a clear defeasible rule say, $ar5 : A \Rightarrow P$ and many different exceptions such as $ar6 : A, B \Rightarrow \neg P$, $ar7 : A, C \Rightarrow \neg P$, $ar8 : \ A, D \Rightarrow \neg P$. Each single exception has a very limited distribution. The necessary conditions above still apply to identify these rules as a group with a base rule and several exceptions.

The procedure for identifying rule pairs (or groups) that meet simple exception criteria is currently performed using a brute force approach on the small hypothetical dataset. This procedure is cumbersome in that any defeasible rule (i.e., those that correspond to association

rules with a confidence between 0 and 100) could be a base rule in an exception group. Therefore, rules that may be exceptions to the base candidate must be sought according to the criteria above. This task explodes combinatorially with large numbers of candidate rules. Future research aims to develop more effective mechanisms than brute force that can be applied to large datasets in real time.

Another situation that association rules can help to automatically identify involves separate and aggregate rules. In this situation A and B are, independently, sufficient reasons for P, but, very often they occur simultaneously. In this case we want to conclude that there are two rules for P, i.e., $A \Rightarrow P$ and $B \Rightarrow P$ instead of $A, B \Rightarrow P$. We identify this situation from the support of itemsets as follows:

$$\frac{Support(A, P) * Support(B, P)}{Support(A, B, P)} > t$$

where t is the aggregate threshold defined by an user.

The next situation addressed involves general vs specific rules. Consider two defeasible rules, $ar8 : A \Rightarrow P$ and $ar9 : A, B \Rightarrow P$. Rule $ar9$ is more specific than $ar8$ because it applies to fewer cases than $ar8$ does. However, in some contexts the more specific (and complex) defeasible rule is preferred and in others the more general (and simpler) rule is preferred. Here again, we include a user defined minimum threshold support for the general rule. If the minimum support is reached then the rule with the highest confidence is selected. For example, the rule $ar10 : X\_on\_right \Rightarrow rightOfWayX$ has a support of 40.5% and confidence 83% the rule $ar11 : X\_on\_right$ and $\neg yPowerDriven \Rightarrow rightOfWayX$ has a support of 13.5% and confidence 62%. If the minimum support for a general rule is set by the user to less than 40% then $ar10$ is preferred because the confidence of $ar10$ is greater than that for $ar11$. However, if the minimum support is greater than 40.5% the general rule does not reach the minimum support for general rules and $ar11$ is selected. A preference for a more general rule over a specific rule underpins the situation we call Apparent irrelevance.

Let us suppose that A alone is not a sufficient reason for D, but together with B it is: the general rule $ar12 : A \Rightarrow D$ does not prevail over the more specific rule $ar13 : A, B \Rightarrow D$. At the same time C alone is not enough to conclude $\neg D$, but $\neg D$ follows from B and C together: $ar13 : C, B \Rightarrow \neg D$ prevails over $ar14 : C \Rightarrow \neg D$.

The use of association rules to facilitate the discovery of defeasible rules is limited in a number of ways. If the antecedent and consequent are not in the same record in the dataset then no association rule corresponding to the desired defeasible rule can be found. For example r_1 above cannot be discovered by association rules because the antecedent and consequent are not separate attributes in the sample dataset:

$$r_1 : \neg rightOfWay(Y, X) \Rightarrow rightOfWay(X, Y)$$
$$r_2 : onTheRightOf(X, Y) \Rightarrow rightOfWay(X, Y)$$

$$r_3 : \ powerDriven(X), \neg powerDriven(Y) \Rightarrow \neg rightOfWay(X, Y)$$

$$r_4 : balloon(X) \rightarrow \neg powerDriven(X)$$
$$r_5 : glider(X) \rightarrow \neg powerDriven(X)$$

The defeasible rules r_2, r_3, r_4, r_5 and r_6 were discovered with confidence levels of 68%, 35%, 100%, 100% and 32%. The rules r_4 and r_5 were discovered as strict rules because their confidence is 100%.

Preference relations identified as appropriate for the aircraft defeasible rules do not seem to be readily derived in a general way from confidence, interest or support measures. This illustrates a further limitation. In addition, the application of any KDD technique including association rules is limited by the coverage of the dataset. Currently, datasets that represent key attributes of the ratio decidendi of human decision making do not readily exist in the legal domain. However, as case management systems become prevalent in Courts across many jurisdictions this is likely to change.

5 Conclusion

In this paper we have noted that although considerable research has been done in the application of Knowledge discovery from database (KDD) techniques to law, the majority of attempts have focused on classification and clustering KDD studies. Few applications of KDD have applied association rules to discover potentially interesting patterns from a dataset.

Metrics of confidence, support and interest that association rule algorithms calculate are suited to the task of discovering patterns from a dataset that suggest appropriate rules for the Defeasible Logic formalism. This non-monotonic logic has been shown to have elegant properties that make it very suitable for practical application in modelling legal reasoning.

Strict rules can be discovered by a direct application of the confidence metric. Defeasible rules can similarly be identified. However, typically too many plausible rules emerge. The number of candidate rules can be reduced by applying support, confidence and interest metrics to the discovery of known types of Defeasible rule groupings. The ones explored in the present study include simple exceptions, separate Vs. aggregate rules, general Vs. specific rules and apparent irrelevance. Future research is aimed at identifying additional rule groupings.

Future research is also aimed at the development of a search algorithm that can scan a set of candidate Defeasible rules more effectively than the brute force method adopted in this toy dataset. Once this is achieved a more rigorous empirical trial can be performed to evaluate the approach advanced here.

References

[1] R. Agrawal, T. Imielinski, and A. Swami. Mining association rules between sets of items in large databases. In *Proceedings of ACM Sigmod Conference*, pages 207–216. ACM Press, 1993.

[2] G. Antoniou, D. Billington, G. Governatori, and M.J. Maher. Representation results for defeasible logic. *ACM Transaction on Computational Logic*, 2:255–287, 2001.

[3] G. Antoniou, D. Billington, G. Governatori, M.J. Maher, and A. Rock. A family of defeasible reasoning logics and its implementation. In W. Horn (Ed.) *ECAI 2000. Proceedings of the 14th European Conference on Artificial Intelligence*, Amsterdam, 2000. IOS Press.

[4] G. Antonious, D. Billington, G. Governatori, and M.J. Maher. On the modelling and analysis of regulations. In *AAAI-2000 Proceedings of the Australian Conference on Information Systems*, pages 401–405, 1999.

[5] T. Bench-Capon. Neural networks and open texture. In *Proceedings of the Fourth International Conference on Artificial Intelligence and Law. ICAIL'93*, pages 292–297. ACM Press, 1993.
[6] D. Billington. Defeasible logic is stable. *Journal of Logic and Computation*, 3:370–400, 1993.
[7] D. Billington and G. Antoniou. A plausible logic with dynamic priorities. In G. Antoniou and G. Governatori (Eds.) *Proceedings of the 2nd Australasian Workshop on Computational Logic*, pages 11–20. QUT Press, 2001.
[8] S. Bruninghaus and K. Ashley. Improving the representation of legal case texts with information extraction methods. In *ICAIL'01. Proceedings of the 8th International Conference on Artificial Intelligence and Law*, pages 42–51. ACM Press, 2001.
[9] M. Covington, D. Nute, and A. Vellino. *Prolog Programming in Depth*. Prentice Hall, 1997.
[10] J. Daniels and E. Rissland. Finding legally relevant passages in case opinions. In *ICAIL'97. Proceedings of the Sixth International Conference on Artificial Intelligence and Law.*, pages 39–46. ACM Press, 1997.
[11] L DeHaspe and L de Raedt Mining Association Rules in Multiple Relations *Proceedings of the 7th International Workshop on Inductive Logic Programming.* Springer-Verlaf. pages 125-132. 1997
[12] B. Duval and P. Nicolas. Learning Default Theories In S. Parsons and A. Hunter (Eds)*Qualitative and Quantitative Approaches to Reasoning with Uncertainty. Lecture Notes in Artificial Intelligence* . Springer-Verlag, pages 148-159.
[13] U. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, editors. *Advances in Knowledge Discovery and Data mining*. 1996.
[14] W.J. Frawley, G. Piatetsky-Shapiro, and C.J. Matheus. Knowledge discovery in databases: an overview. *Knowledge discovery in databases*, pages 1–27, 1991. AAAI/MIT.
[15] G. Goulbourne, F. Coenen, and P. Leng. Algorithms for computing association rules using a partial support tree. *Journal of Knowledge Based Systems*, 13:141–149, 2000.
[16] G. Governatori, M. Dumas, A.H.M. ter Hofstede, and P. Oaks. A formal approach to protocols and strategies for (legal) negotiation. In *ICAIL'01. Proceedings of the 8th International Conference on Artificial Intelligence and Law*, pages 168–177. ACM Press, 2001.
[17] G. Governatori and M.J. Maher. An argumentation theoretic characterization of defeasible logic. In W. Horn (Ed.) *ECAI 2000. Proceedings of the 14th European Conference on Artificial Intelligence*, Amsterdam, 2000. IOS Press.
[18] G. Governatori, M.J. Maher, G. Antoniou and D. Billington. Argumeantation semantics for defeasible logics. In *Pacific Rim Internatinal Conference on Artificial Intelligence*. Springer-Verlag, 2000.
[19] P. Hayes. Intelligent high volume text processing using shallow, domain specific techniques. In P. Jacobs (Ed.) *Text-Based Intelligent Systems: Current Research and Practice in Information Extraction and Retrieval*, pages 227–241. Lawrence Erlbaum, Hillsdale, NJ, 1992.
[20] J. Hobson and D. Slee. Indexing the theft act 1968 for case based reasoning and artificial neural networks. In *Proceedings of the Fourth National Conference on Law, Computers and Artificial Intelligence*, Exeter, 1994.
[21] N. Lavrac *Computational Logic and Machine Learning: A Roadmap for Inductive Logic Programming* Technical Report, J. Stefan Institute, Ljubljana, Slovenia, 1998.
[22] M.J. Maher. Propositional defeasible logic has linear complexity. Technical report, Department of Mathematical and Computer Sciences, Loyola University, Chicago, 2000.
[23] M.J. Maher, A. Rock, G. Antoniou, D. Billignton, and T. Miller. Efficient defeasible reasoning systems. In *Proc. International Conference on Tool in Artificial Intelligence*. EEEI Computer Society Press, 2000.
[24] M. Moens, C. Uyttendaele, and J. Dumortier. Abstracting of legal cases: The salomon experience. In *ICAIL'97. Proceedings of the Sixth International Conference on Artificial Intelligence and Law.*, pages 114–122. ACM Press, 1997.
[25] D. Nute. Defeasible reasoning. In *Proc. 20th Hawaii International Conference on System Science*, pages 470–477. IEEE Press, 1987.
[26] D. Nute. *Handbook of Logic in Artificial Intelligence and Logic Programming*, volume 3, chapter Defeasible Logic, pages 353–395. Oxford University Press, Oxford, 1994.
[27] A. Pannu. Using genetic algorithms to inductively reason with cases in the legal domain. In *ICAIL'95 Proceedings of the Fifth International Conference on Artificial Intelligence and Law*, pages 175–184. ACM Press., 1995.

[28] E. Rissland and T. Friedman. Detecting change in legal concepts. In *ICAIL'95. Proceedings of the Fifth International Conference on Artificial Intelligence and Law*, pages 127–136. ACM Press, 1995.

[29] E. Schweighofer. The revolution in legal information retrieval or: The empire strikes back. *Journal of Information, Law and Technology*, 1, 1999.

[30] A. Stranieri, J. Zeleznikow, and H. Turner. Data mining in law with association rules. In *Proceedings of IASTED International conference on Law and Technology*, pages 129–135, 2000.

[31] P. Thompson. Automatic categorisation of case law. In *ICAIL'01. Proceedings of the 8th International Conference on Artificial Intelligence and Law*, pages 70–77. ACM Press, 2001.

[32] G. Vossos, J. Zeleznikow, A. Moore, and D. Hunter. The credit act advisory system (caas): Conversion from an expert system prototype to a c++ commercial system. In *ICAIOL'93. Fourth International Conference on Artificial Intelligence and Law*, page 180. ACM Press, 1993.

[33] D. Wilkins and K. Pillaipakkamnatt. The effectiveness of machine learning techniques for predicting time to case disposition. In *ICAIL'97. Proceedings of the Sixth International Conference on Artificial Intelligence and Law*, pages 106–113. ACM Press, 1997.

[34] J. Yearwood, A. Stranieri, and J. Zeleznikow. Case-based retrieval of refugee review tribunal text cases. In A. Oskamp, R.V. De Mulder, C. van Noortwijk, C.A.F.M. Grtters, K. Ashley, and T. Gordon, editors, *Legal Knowledge Based Systems. JURIX: The Tenth Conference*, pages 67–83, 1997.

[35] J. Zeleznikow, G. Vossos, and D. Hunter. The ikbals project: Multi-modal reasoning in legal knowledge based systems. *Artificial Intelligence and Law*, 2:169–203, 1994.