Near knowledge:

Inductive learning systems in law

Dan Hunter*

> To be fond of learning is to be near knowledge.
>
> Tze-Sze, *The doctrine of the mean*, (5th C BC)

Abstract

Induction is an interesting model of legal reasoning, since it provides a method of capturing initial states of legal principles and rules, and adjusting these principles and rules over time as the law changes. In this Article I explain how Artificial Intelligence-based inductive learning algorithms work, and show how they have been used in law to model legal domains. I identify some problems with implementations undertaken in law to date, and create a taxonomy of appropriate cases to use in legal inductive inferencing systems. I suggest that inductive learning algorithms have potential in modeling law, but that the artificial intelligence implementations to date are problematic. I argue that induction should be further investigated, since it has the potential to be an extremely useful mechanism for understanding legal domains.

* Assistant Professor of Legal Studies, The Wharton School, University of Pennsylvania (Monash University BSc 1987; Monash University Law School LLB (Hons) 1989; University of Melbourne Law School LLM 1996; University of Cambridge PhD 1999). Thanks to John Zeleznikow, with whom I created the initial ID3 worked example explained in Part II; and to the following people who read and commented on previous incarnations of this Article: Michael Aikenhead, Mark Gawler, David Hamer, Richard Ingleby, Richard Johnstone, Andrew Stranieri, and Richard Wright. The mistakes are, like my prejudices, mine alone. Research for this Article was supported by funds from the University of Melbourne Law School, the Herchel Smith Research Fellowship in Intellectual Property Law, and Emmanuel College, Cambridge. My profound thanks for their support.

Contents

I. Introduction

Induction is the process where we humans take a set of experiences and, from these experiences, derive a general principle. Thus, my expectation that the sun will rise tomorrow is inductively derived; as is my knowledge that apples are good to eat, that women are different from men, that black is not white, that birds can fly, and so on. Each one of these general principles can be derived from a series of individual instances of experience: I see a blackbird flying, a robin flying, a hawk flying, and derive a principle that 'Most birds can fly'. A huge corpus of our understanding of the world comes from inductively learnt principles.

Induction is not confined to the examples given above, which all involve real-world experiences using sense data. Humans also use induction in deriving principles about abstract domains, such as law. For example, lawyers are able to take the individual experiences of precedents and derive general principles that we call 'rules' or 'principles' or 'legal arguments'. Any case-law domain necessary relies on inductive learning as a prerequisite for fashioning some generally applicable sets of rules. Without this ability to generalize from precedents, we would be like a legal version of Jose Luis Borges' character 'Funes the Memorious'. Ireneo Funes had a prodigious memory for specific instances and details, but he was incapable of generalizing from these instances:

> Not only was it difficult for him to comprehend that the generic symbol "dog" embraces so many unlike individuals of diverse size and form; it bothered him that the dog at three fourteen (seen from the side) should have the same name as the dog at three fifteen (seen from the front).[1]

---

1 Jose Luis Borges LABYRINTHS: SELECTED STORIES AND OTHER WRITINGS 59 -65 (1964).

I have argued in a separate Article[2] that inductive inference is one of the most overlooked -- and worst understood -- forms of legal reasoning, and I do not wish to rehearse these arguments here. Instead, in this Article I wish to show that artificial intelligence techniques for inductive learning are useful, important, and relevant techniques in studying and modeling legal reasoning. Inductive learning algorithms provide a way of using computer systems to model the process of reasoning from precedent. I will, however, argue that these algorithms cannot be applied to law without a serious understanding of the way that both the algorithms and the law operate. This Article will show how we can intelligently use these algorithms in modeling legal reasoning.[3]

To show these main features, this Article is divided into five parts, including this Introduction. In the next part I provide an overview of inductive learning generally, in order to define terms and set the context for the Article. In this part I also show how induction may be understood to operate in legal reasoning. Here I also provide a basic explanation of inductive learning algorithms from artificial intelligence, and give a worked example of the use of a learning algorithm in law. With this background, Part III reviews the major research done in applying inductive algorithms to legal reasoning. Part IV analyses the benefits and problems with these applications, and concludes with some ideas about how inductive algorithms should be used in law. Part V concludes the analysis and makes some suggestions for further research.

To begin then, let us look at how induction operates.

---

2 Hunter, D., *No Wilderness of Single Instances: Inductive Inference in Law*, 48 JOURNAL OF LEGAL EDUCATION 365-401 (1998) [hereinafter Hunter, *No Wilderness*].

II. Induction

Induction is the process of taking a number of instances from experience, classifying them into categories, and deriving from them one or more generally applicable rules.[4] That is, we take a number of isolated experiences and attempt to generalize a category into which all of these examples fall. Once the classification has been made, it is possible to derive a rule which defines each classification. And once these rules have been induced, they may be applied to new situations in a purely deductive manner. Often the initial stage of the process is called 'inductive learning,' while the two stage process of induction and then deduction is usually called 'inductive reasoning.' Generally I will use the term 'induction' or 'inductive reasoning' to indicate the two stage process. I do so because in law the deductive stage of the process is as important as the earlier inductive learning phase, as any information derived in learning must be applied to a new case.

The rule derived in the inductive learning stage is a generalization of the previous cases or previous evidence.[5] However, the inductively derived rule may be wrong. A quintessential example of this is the situation where early observers in the northern

---

3 I have undertaken a similar task for neural networks in law; see Hunter, D. *Out of their minds: Legal theory in neural networks* to appear in ARTIFICIAL INTELLIGENCE AND LAW (Hereafter Hunter *Minds*)

4 Hunter *No wilderness*, supra note 2. GOLDING, M.P., LEGAL REASONING, 43 (1984) (hereafter GOLDING, LEGAL REASONING), notes that there are many types of inductive arguments, all of which share the feature that they are only evidence of the conclusion reached. He suggests that the type of induction described here is 'induction by enumeration.' It is beyond the scope of this work to examine induction generally, and so we will only examine induction by enumeration since it is the main inductive approach relevant in modeling legal reasoning. For a more detailed analysis, see Hunter *No wilderness*, supra note 1.

hemisphere saw many white swans. They inductively derived a rule that could be applied to future questions about swans. The process of reasoning occurs as follows:

> *A* is a swan and *A* is white.
>
> *B* is a swan and *B* is white.
>
> *C* is a swan and *C* is white.
>
> .
>
> .
>
> .
>
> *n* is a swan and *n* is white.
>
> Therefore, all swans are white[6]

The problem with this, of course, is that the rule is wrong. All swans are not white, as European explorers to Australia discovered in the 1770s. The vulnerability of induction causes concern about its use.[7]

The potential inaccuracy notwithstanding, it is clear that we use induction all the time, and that it is a vital part of the legal reasoning process. As I show in my earlier Article, inductive inference is used in a number of different legal contexts.[8] These include using induction to derive rules from precedents, or using induction to understand whether a

---

5 Hunter *No wilderness*, supra note 2, at 368-70.

6 The example is from GOLDING, LEGAL REASONING supra note 4, at 43.

7 For a resolution of this problem, see Hunter *No wilderness*, supra note 2, at 385-92.

8 Hunter *No wilderness*, supra note 2, at 367-9.

particular judge exhibits consistent bias towards one party or another.[9] However, the best example is perhaps the *ejusdem generis* rule.[10] This canon of statutory interpretation is invoked to determine whether an object, a thing or an action falls within a statutory definition of the form '*x*, *y*, *z* or other.' It is used to define the scope of statutory terms, where general words immediately follow specific words.

The *ejusdem generis* rule clearly relies on inductive reasoning. Courts examine the words that make up the antecedent part of the expression in question, and then decide what the scope of those words incorporate. The courts require there to be a clear genus into which all of the specific words fall, otherwise they will refuse to apply the *ejusdem generis* rule.[11] The mandatory identification of a genus demands that this type of reasoning is reliant on inductive inference. It is necessary to take the specific words, and from them create an inductive generalization about the genus that the legislature wished included in the expression '*x*, *y*, *z* or other.'

Clearly induction is an important part of legal reasoning. Let us look then at how artificial intelligence techniques formalize this induction process.

9 Hunter *No wilderness*, supra note 2, at 392-4.

10 The rule applies also in interpretation of other documents, but is at its most powerful in statutory interpretation.

11 For a fuller discussion of this, see Hunter *No wilderness*, supra note 2, at 392-4.

A. Induction in artificial intelligence and law

> Data isn't information. Information isn't knowledge. Knowledge isn't wisdom.
>
> [Anon]

Induction is a major topic within artificial intelligence theory, and is closely associated with research into learning from examples. Approaches in example-based learning and induction include learning by analyzing differences and similarities in a training set, learning by classification, learning by explaining experience, learning by correcting mistakes, and a number of others.[12] However I shall limit myself to one type of classification learning system—inductive learning by building decision trees. I do this in part for the sake of simplicity in demonstrating the use of artificial intelligence induction algorithms in law. More importantly, however, this type of inductive algorithm generates decision trees, which can then be rendered into rules.[13] These types of algorithms are obviously relevant within a legal framework, since the inductive generation of rules from experience corresponds to our deeply held perception that the derivation and use of rules are central to legal reasoning.[14] Therefore, this type of inductive algorithm is applicable to

---

12 See Patrick .H. Winston ARTIFICIAL INTELLIGENCE (3rd ed, 1992) (hereafter Winston ARTIFICIAL INTELLIGENCE) 347-442.

13 See Part II.B below.

14 By this I do not mean that rules are the only, or indeed the controlling, feature of legal reasoning. However there is a large literature in legal reasoning which identifies rules as one of the primary elements in legal reasoning. See for example, Frederick F. Schauer PLAYING BY THE RULES: A PHILOSOPHICAL EXAMINATION OF RULE-BASED DECISION-MAKING IN LAW AND LIFE (1991); D. Neil MacCormick LEGAL REASONING AND LEGAL THEORY (1978); D. Neil MacCormick *The nature of legal reasoning: A brief reply to Dr Wilson*. 2 LEGAL STUDIES 286 (1982); D. Neil MacCormick *Why cases have rationes and what these are* in Laurence Goldstein. (ed) PRECEDENT IN LAW, 155; Stephen J. Burton. AN INTRODUCTION TO LAW AND LEGAL REASONING. (2nd ed, 1995). Within the artificial intelligence and law movements there

legal reasoning. Finally, my discussion will focus on these types of systems, since induction by building decision trees is the only approach which has been meaningfully applied to law. In any event, this approach is representative of induction generally, and the lessons learnt here may be applied to other inductive paradigms.

B. Inductive learning algorithms in law

> Learning is its own exceeding great reward.
>
> William Hazlitt[15]

The most important algorithm for the inductive classification of a group of cases is called ID3.[16] The ID3 algorithm uses information theory to look for regularities in a set of data, and using these regularities the algorithm then classifies cases into sets which share certain common features. From the regularities it is possible to induce a decision tree which 'explains' all of the cases.

---

have also been a large number of implementations of rule-based reasoning expert systems. See the discussion of rule-based reasoning in John Zeleznikow and Dan Hunter BUILDING INTELLIGENT LEGAL INFORMATION SYSTEMS—REPRESENTATION AND REASONING IN LAW (1994) (Hereafter Zeleznikow and Hunter, BUILDING SYSTEMS).

15 William Hazlitt, *On Old English Writers and Speakers*, THE PLAIN SPEAKER 1826.

16 J.R.Quinlan *Discovering rules by induction from large collections of examples* in D. Michie (ed) EXPERT SYSTEMS IN THE MICRO-ELECTRONIC AGE (1979) (herafter Quinlan *Discovering rules*); J.R.Quinlan *Inferno: A cautious approach to uncertain inference* COMPUTER JOURNAL (1983) (hereafter Quinlan *Inferno*); J.R.Quinlan *Induction of decision trees* 1 MACHINE LEARNING 81 (1986) (hereafter Quinlan *Induction*); J.R.Quinlan *Induction, knowledge and expert systems* in J.S.Gero and R. Stanton (eds) ARTIFICIAL INTELLIGENCE DEVELOPMENTS AND APPLICATIONS 253 (1988) (hereafter Quinlan *Induction knowledge*).

In order better to explain the process I will examine an example set of data, derived from the work of Hunter and Zeleznikow.[17] The data stems from information gathered in the domain of family law, but similar approaches can be found in many other domains and jurisdictions. The cases were all drawn from real life, and involved a set of divorces in a no-fault jurisdiction.[18] The question for the judge in each of these cases was, after the divorce, what percentage of marital assets should be awarded to the wife and what percentage should go to the husband?

There were a number of features ('attributes') that were considered relevant to the question of what percentage the wife would receive (the 'outcome'). These features included the value of the marital property, the number of children, and whether the wife was working at the time of divorce. Other attributes, assumed to be irrelevant to the outcome but included in the table nonetheless, were the name of the case, the names of the parties, and the name of the judge. The table of (anonymized) cases is shown in figure 1.

case	party_name	judge	value_of_ ppty	no._ children	wife_ working	percentage_ awarded (to wife)
Case 50	White	Green	100,000	2	Yes	55%
Case 51	Green	Red	10,000	3	No	70%
Case 52	Rich	Green	50,000	4	No	65%
Case 53	Smith	Brown	1,000,000	0	Yes	50%
Case 54	Jones	Orange	500,000	3	No	65%
Case 55	Brown	Blue	40,000	0	Yes	50%
Case 56	Smyth	Yellow	10,000	2	Yes	70%

Figure 1. Initial table of divorce data

17 Dan Hunter and John Zeleznikow *An overview of some reasoning formalisms as applied to law* 3 THINK 24 (1994); Zeleznikow and Hunter, BUILDING SYSTEMS supra note 14, at 259-269; John Zeleznikow and Dan Hunter *Reasoning paradigms in legal decision support systems*, 9(6) ARTIFICIAL INTELLIGENCE REVIEW 361-385 (1996).

18 The jurisdiction was the Australian Federal family law system. For a full description see Zeleznikow and Hunter, BUILDING SYSTEMS supra note 14, at 259-269.

Though the table accurately encodes the seven cases in our set of cases, the ID3 algorithm requires this information to be simplified in order for it to induce a set of useable rules. Without simplification of this sort, inductive algorithms can produce excessively narrow and useless rules.

This simplification process has two components. First and most important, for the algorithm to operate effectively, a binary outcome is desirable. Thus, it is necessary to take the multi-valued outcome percentage_awarded and change it to a binary-valued outcome called equal_split, which indicates simply whether the assets were split equally among wife and husband. The values for percentage_awarded which range between 45% and 60% were transposed to becomes 'yes' for the new attribute equal_split. All other ranges were rendered as 'no' for the equal_split attribute.

The second component of the simplification process involves conflating some values of the attributes into simpler values. So for example, the attribute value_of_ppty was changed, to be called asset_rich to indicate whether or not the marriage produced a large asset base. Values of value_of_ppty equaling or above $100,000 were transposed in asset_rich to be 'yes' and all others were rendered as 'no'. The attribute detailing the number of children in the marriage was also changed to a simpler attribute, indicating only the presence or absence of children.

The initial table of figure 1 was therefore simplified to become the table in figure 2.[19]

---

[19] In the version presented in Zeleznikow and Hunter, BUILDING SYSTEMS supra note 14, at 259-269, two attributes—party_name and the judge's name--were removed in the simplification process, on the basis that they were irrelevant to the outcome, and therefore unnecessary. While there are arguments in favor of this, a better approach is to leave the 'noise' attributes in the sample set and allow the ID3 algorithm to see if there is any regularity in these attributes. One can see the merits of this approach if, for example, the ID3 algorithm concluded that one of the

case	party_name	judge	asset_rich	children	wife_working	equal_split
Case 50	White	Green	yes	yes	yes	yes
Case 51	Green	Red	no	yes	no	no
Case 52	Rich	Green	no	yes	no	no
Case 53	Smith	Brown	yes	no	yes	yes
Case 54	Jones	Orange	yes	yes	no	no
Case 55	Brown	Blue	no	no	yes	Yes
Case 56	Smyth	Yellow	no	yes	yes	No

Figure 2. Simplified table of divorce data

There is, of course, a problem with this simplification process. Any change in the data set potentially creates a number of artifacts, since conflating a range of values to binary values inevitably involves choice, and any choice introduces conceptual bias.[20] The conceptual bias here is relatively minor, and I will show that it does not adversely affect the rules generated by the system. However the concern about conceptual bias is real, and in Part IV.A below I discuss this problem, in conjunction with a number of other concerns about the use of data sets in induction systems.

From the simplified data set it is possible to generate a number of decision trees which 'explain' the data.[21] Decision trees are simply a means of showing all of the possible

---

important attributes was whether Judge X was sitting on the case. This might indicate a useful insight, as the judge may have a particular bias in favor of the wife or husband, or decide according to peculiar idiosyncrasies. Equally the party's name might be a relevant factor to determine whether a litigant was vexatious (many cases with the same litigant, all decided against her) or to determine if a particularly rich litigant always wins cases.

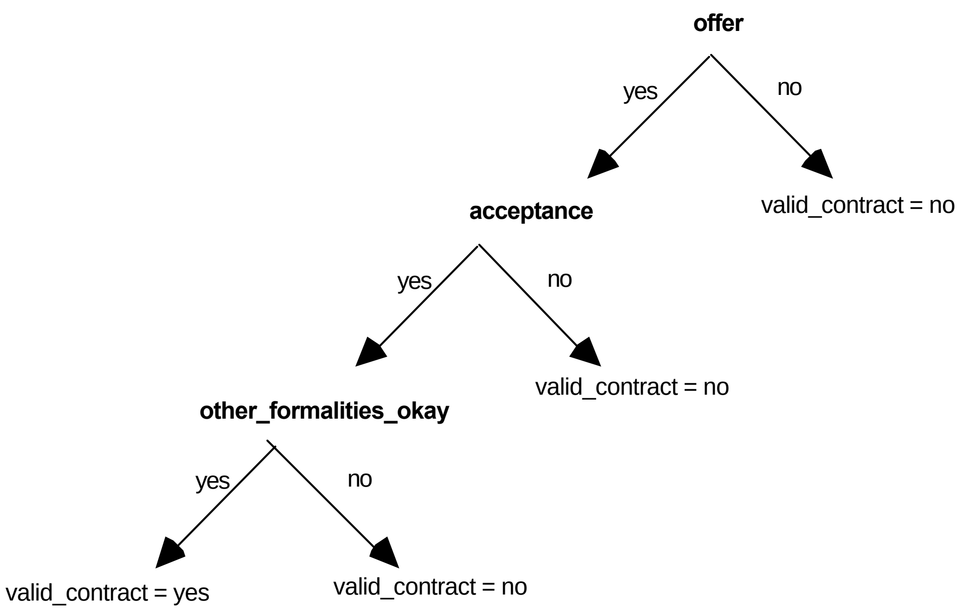
20 Zeleznikow and Hunter, BUILDING SYSTEMS supra note 14, at 265-6.

21 In formal terms, a decision tree is '…a problem representation in which:
1. Each node is connected to a set of possible answers;

alternatives in a set of rules, drawn up as a 'tree' of these rules. The tree begins at the 'root' rule and then splits into a series of 'sub-trees' which deal with subsidiary rules. The nodes show the feature under question and the arcs show the alternative 'answers' to that question. For example, in the domain of contract law we might have the rule:

IF offer AND acceptance AND other_formalities_okay THEN valid_contract = yes ELSE valid_contract = no

This would be drawn in a decision tree as:



2. Each non-leaf node is connected to a test that splits its set of possible answers into subsets corresponding to different test results; and
3. Each path carries a particular test result's subset to another node.'

Zeleznikow and Hunter, BUILDING SYSTEMS supra note 14, at 260.

Furthermore:
4. Each leaf node represents a problem solution.

An identification tree is a decision tree in which each set of possible conclusions is established by a list of samples or examples for that conclusion, Winston, above n 2, 425. For our purposes there is no distinction between the types of tree, and we will hereafter use the term 'decision tree' to mean both.

Figure 3. Example decision tree

Returning to the data in figure 2, it is possible manually to derive a *non-optimal* decision tree. Essentially this involves rendering each case into a rule. Thus, Case 50 can be thought of as a rule:

IF asset_rich AND children AND wife_working THEN equal_split[22]

In doing this exercise manually, we might find that a few cases match-up on various attributes. So, for example, the outcomes of Case 50 and Case 53 (both having an equal_split) can be explained by seeing that both are asset rich marriages and the wife is working. This is a rough and ready type of manual induction, and may lead to interesting results. If we undertake this manual induction process in enough detail, we can generate a decision tree which covers the entire data set. Here is one example of decision tree from the data:

22 For the sake of simplicity I have removed the party_name and judge attributes in this example rule and example decision tree that follows in figure 4.

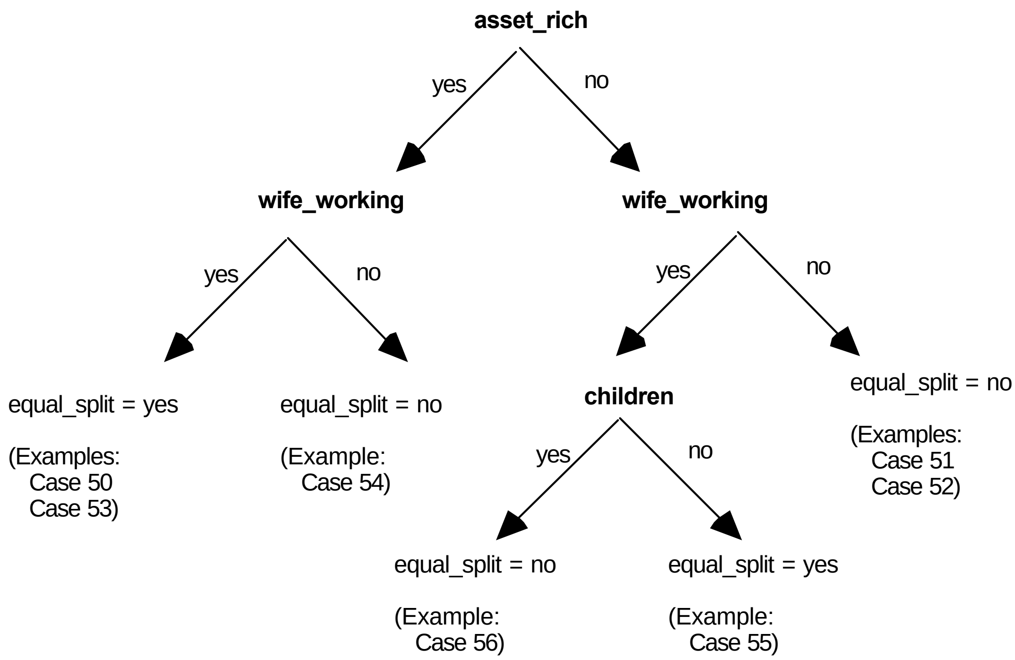


Figure 4. Manually induced decision tree

This manual approach has one major drawback, apart from the fact that it is unreasonably time-consuming and tedious: the manually generated decision tree probably will not be the most efficient. The ID3 algorithm avoids this problem, and generates the most efficient tree possible, on the basis of minimizing disorder in each sub-tree. The disorder formula stems from information theory, and an examination of its function is beyond the scope of this Article.[23] In essence though, the algorithm generates the optimal decision tree by

23 The average disorder formula is:

$$\sum_b \left(\frac{n_b}{n_t}\right) \times \left(\sum_c -\frac{n_{bc}}{n_b} \log_2 \frac{n_{bc}}{n_b}\right)$$

where n_b is the number of samples in branch b, n_t is the total number of samples in all branches and n_{bc} is the total number of samples in branch b of class c.

For a simple discussion of the use of the disorder formula in the ID3 algorithm see Winston ARTIFICIAL INTELLIGENCE supra note 12, at 429-431. For detailed discussion of ID3 and its reducing the amount of disorder at each node, thereby producing the simplest, least-branching decision tree that explains all the data. The optimal tree for the above case set, and the one generated by the ID3 algorithm, is shown in figure 5. The cases covered by each rule or sub-tree are indicated at the leaf nodes.

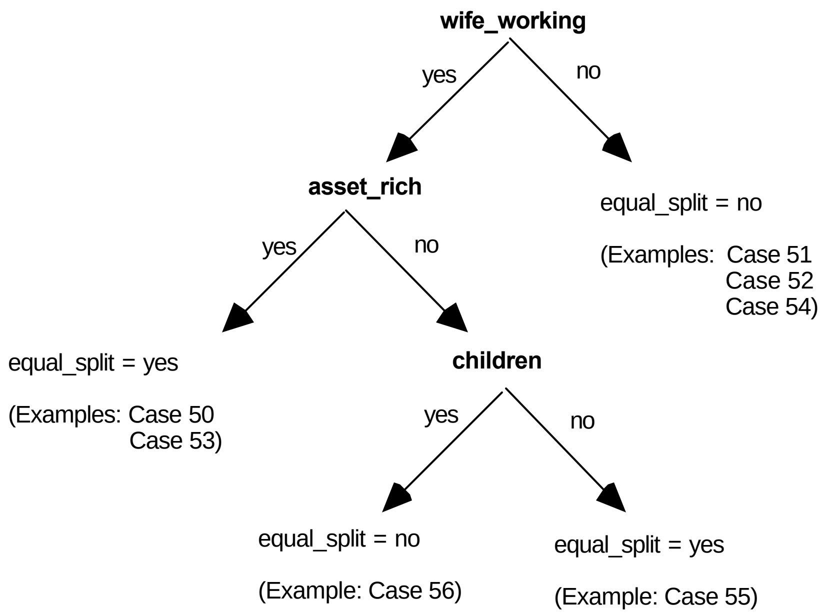


Figure 5. ID3-generated decision tree

We can generate a set of rules from this decision tree. We use the simple process of tracing each path in the tree from root node to leaf node, noting the intermediate arc-node pairs as antecedents and leaf arc-node pairs as consequents.[24] The rules derived from the above decision tree are:[25]

---

use of information theory formulæ see Quinlan *Discovering rules* supra note 16, Quinlan *Induction* supra note 16.

24 Winston ARTIFICIAL INTELLIGENCE supra note 12, at 432.

25 Note that the order of the generation of rules will depend on whether one follows a depth-first or breadth-first search strategy, and whether the search occurs left to right, or right to left. The

Rule 1. IF NOT wife_working THEN equal_split = no

Rule 2. IF wife_working AND asset_rich THEN equal_split = yes

Rule 3. IF wife_working AND NOT asset_rich AND children THEN equal_split = no

Rule 4. IF wife_working AND NOT asset_rich AND NOT children THEN equal_split = yes

Inductive derivation of these rules offers a number of advantages. It is much easier for us to comprehend the legal rules than it is to try to understand the original set of cases. This is particularly true where the case set is very large; but even in small data sets, such as the example above, the rules convey more information than the bare case information. This may be used as a basis for understanding the legal domain, or for criticizing the decisions of judges. Take as an example, Rule 2 above. We might question whether it is appropriate for judges to divide the marital property equally where the wife is working and the marriage is rich in assets, irrespective of the existence of children. The rule exposes potential injustice more clearly than a number of disconnected cases.

Another useful feature of these inductively derived rules is that they can immediately be used in a rule-based expert system. These expert systems have been developed in law in many legal areas,[26] but their commercialization and application have been limited, in part, by the cost of generating the rules to use in the system. Rules in these systems are usually derived from expert knowledge, which must be extracted and encoded in a very labor-

---

rules in this example are derived by a breadth-first, left-to-right search. This means that the entire left branch is first generated, then the next sub-branch to the right, and so forth. However, though the order of generation will change, the rules themselves are always the same. Since we are dealing with declarative rather than procedural rules the order of presentation of the rules in the rule base makes no difference to the inferencing process.

26 Zeleznikow and Hunter, BUILDING SYSTEMS supra note 14.

intensive, time-consuming, and expensive process.[27] Using inductive learning algorithms has the potential to avoid the 'knowledge acquisition bottleneck.'

Now that I have examined the basic mechanism of inductive learning using the ID3 algorithm, I turn to a number of implementations of this approach in law.

III. Implementations in law.

> Nothing is repeated, and everything is unparalleled.
>
> Edmond and Jules de Goncourt[28]

Compared with the other artificial intelligence paradigms,[29] induction systems are relatively uncommon in law. The most important applications are those of Tyree[30] and Karpf,[31] which I discuss in the followings sections.[32] It is important to note that the

---

27 Zeleznikow and Hunter, BUILDING SYSTEMS supra note 14, at 161-163.

28 Edmond and Jules de Goncourt, JOURNAL, April 15, 1867.

29 The three major paradigms that have been successfully used in law are rule-based reasoning approaches, case-based reasoning approaches, and neural networks. For a review of legal rule-based systems see Zeleznikow and Hunter, BUILDING SYSTEMS supra note 14. For legal case-based reasoning see Zeleznikow and Hunter, BUILDING SYSTEMS supra note 14, at 140-158; Kevin D. Ashley *Case-based reasoning and its implications for legal expert systems* 1 ARTIFICIAL INTELLIGENCE AND LAW 113 (1992); Kevin D. Ashley MODELLING LEGAL ARGUMENT: REASONING WITH CASES AND HYPOTHETICALS. (1990). For legal neural networks see Hunter *Minds*, supra note 3.

30 Alan L.Tyree EXPERT SYSTEMS IN LAW (1989) (hereafter Tyree EXPERT SYSTEMS)

31 J. Karpf *Inductive modelling in law—Example based expert systems in administrative law* in PROCEEDINGS OF THE THIRD INTERNATIONAL CONFERENCE ON ARTIFICIAL INTELLIGENCE AND LAW 297 (1991, Oxford: ACM Press) (hereafter Karpf *Inductive modelling*).

32 There have been a number of other uses of inductive algorithms in law, but for various reasons they are not considered here. The introductory work of Hunter and Zeleznikow is reviewed in

dearth of examples of inductive learning systems in legal domains is not indicative of any fundamental problem with these systems. It just seems that they have not captured the attention of many researchers.

A. Tyree

Professor Alan Tyree was one of the first to examine the use of inductive algorithms in legal domains. In his early book he presented an approach to using cases in law which relied on induction and specifically Quinlan's ID3 algorithm. [33]

For this system, Tyree used a legal domain that is heavily dependent on cases, the English law of trover. 'Trover' is the law relating to property rights in lost and found personal property, and specifically deals with the circumstances where a finder of lost chattels may enforce property rights against another. Essentially, the law of trover deals with situations where someone finds a very valuable chattel on another's real property, and the true owner of the chattel is never found. So, for example, if I find a valuable necklace while working in my employer's building, and the owner never comes forward to claim it, the law of trover will determine which of me and my employer gets to keep the necklace.

Part II, see Zeleznikow and Hunter, BUILDING SYSTEMS supra note 14, at 259-269. A.S. Pannu *Using genetic algorithms to inductively reason with cases in the legal domain* in PROCEEDINGS OF THE FIFTH INTERNATIONAL CONFERENCE ON ARTIFICIAL INTELLIGENCE AND LAW 175 (1995, College Park: ACM Press) uses a completely different paradigm--genetic algorithms--to induce information in legal domains. The use of genetic algorithms involve different considerations to the ones discussed here, so this approach is not relevant. Edwina L. Rissland and M. Tibor Friedman *Detecting change in legal concepts* in PROCEEDINGS OF THE FIFTH INTERNATIONAL CONFERENCE ON ARTIFICIAL INTELLIGENCE AND LAW 127 (1995, College Park: ACM Press) (Hereafter Rissland and Friedman *Change*) use inductive learning algorithms, but their focus is upon its use to determine the 'drift' in a legal concept over time. Their approach, though important, is also not relevant to the discussion here.

33 Tyree EXPERT SYSTEMS supra note 30, at 161-175.

Trover is an unusual legal domain, in that it is entirely case derived, there being no statutory guidelines at all. Furthermore, there are a very small number of appellate-level cases which determine the entire domain. There are in fact only eight English cases for the entire domain. Tyree analyzed which types of domains he thought were suitable for inductive algorithms, and chose ones which shared trover's characteristics: a case-based domain, which has a small number of appellate-level decisions.

Tyree therefore fed the eight trover cases[34] into an inductive algorithm similar to ID3, and produced the following decision tree:[35]

---

34 The cases were, in alphabetical order and identified by a letter, A: *Armory v Delamire* (1721) 1 Strange 505, B: *Bridges v Hawkesworth* (1851) 21 LJQB 75, C: *Elwes v Briggs Gas Co* (1886) 33 ChD 562, D: *Hannah v Peel* [1945] 1 KB 509, E: *London v Yorkwin* [1963] 1 WLR 982, F: *Moffatt v Kazana* [1969] 2 QB 152, G: *South Staffordshire Water Co v Sharman* [1896] 2 QB 44, H: *Yorkwin v Appleyard* [1963] 1 WLR 982.

Each of these cases were indexed on 9 attributes. The complete list of attributes was as follows:

A1: The finder is the occupier of the premises where the chattel was found.
A2: The chattel was attached.
A3: The other claimant is the owner of the p remises where the chattel was found.
A4: The other party is the true owner of the chattel or is claiming through the rights of the true owner.
A5: The finder handed over the chattel to the other claimant after the finding.
A6: One of the parties is relying on the terms of an agreement made with the other which purports to give her the right to the chattel.
A7: The finder is a servant of the other party.
A8: The chattel was hidden or was in a position so as to be difficult to find.
A9: There was an attempt made to find the true owner of the chattel or alternatively the chattel was clearly abandoned.
A10: One of the parties knew of the existence of the chattel prior to the finding.

The complete array of attribute-value pairs for the eight cases is:

	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10
A:	0	0	0	0	1	0	0	0	0	1
B:	0	0	1	0	1	0	0	0	1	0
C:	1	1	1	0	0	1	0	1	1	0
D:	0	0	1	0	0	0	0	1	1	0
E:	1	1	1	0	1	1	0	1	1	0
F:	1	0	0	1	0	0	0	1	1	1
G:	0	1	1	0	0	0	1	1	1	0
H:	0	1	0	0	0	0	1	1	1	0

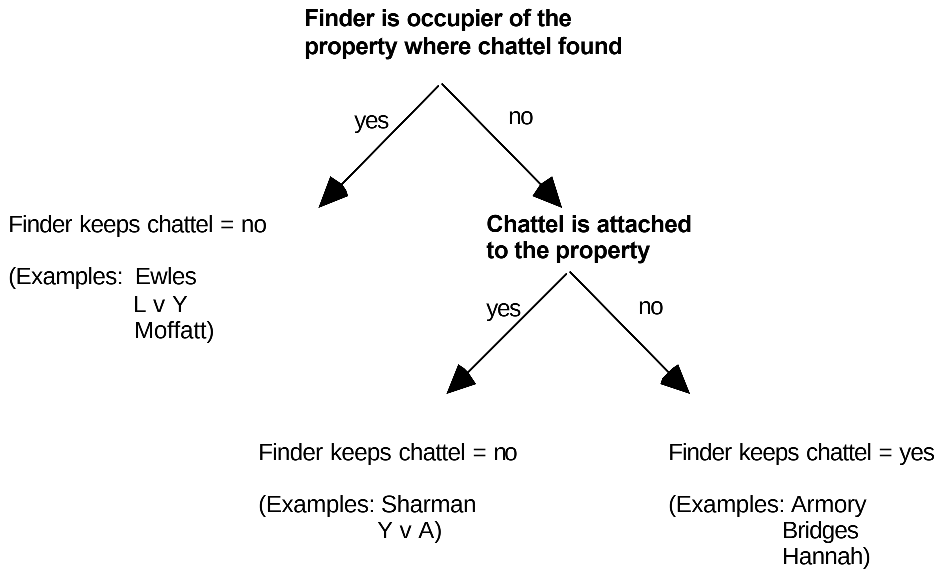


Figure 6. Decision tree for trover cases

Tyree's work is seminal because he shows that artificial intelligence-based inductive algorithms can usefully be applied to law. He also indicates the importance of legal inductive inference in his discussion of why AI rule-based systems have problems in legal domains:

> The difficulty of formulating rules which capture case law may reflect the fact that case law reasoning is closer to inductive than to deductive reasoning. Although reasoning with case law may have some deductive components, the essence of it

---

See Tyree EXPERT SYSTEMS supra note 30, at 161-175

35 Tyree EXPERT SYSTEMS supra note 30, at 161-165. The tree is given after a discussion of Minimal Spanning Trees, a type of inductive algorithm related to ID3. Tyree in fact seems to have induced this tree manually. He does not mention until later in his work that automatic induction is possible and instead presents the example as one of manual induction of rules from cases. He states later, Tyree EXPERT SYSTEMS supra note 30, at 170, that the ID3 algorithm produced a decision tree with only five nodes, but unfortunately does not provide the decision tree which it generated.

> would appear to be to generalize from a number of instances rather than the application of logical rules.[36]

However, Tyree's use of the domain of trover causes some concerns. A basic problem with his approach is the use of domains which have a small number of appellate-level cases. In Part IV is will develop the idea that inductive inference generally -- and inductive algorithms specifically -- require a large number of fairly common cases to work effectively. Small numbers of idiosyncratic appellate cases can lead to inductively derived rules that are inaccurate, and not representative of the legal domain being modeled. I will develop this argument in more detail in Part IV, but it is important to note here that Tyree's approach relies on domains which share the unusual features of the law of trover.

This is not the case with the other main implementation which has been attempted in law, that of Karpf.

B. Karpf

Jorgen Karpf reported on two implementations built using what he characterized as inductive approaches, which use algorithms similar to the ID3 algorithm.[37] The first system classified whether or not an employee was a 'wage-earner' for the purposes of employment law. In Karpf's civil law-based domain, wage-earners are entitled to various

---

36 Tyree EXPERT SYSTEMS supra note 30, at 133.

37 Karpf *Inductive modelling* reported on four systems, two of which appear to rely mostly on neural networks. The other two systems mentioned seem to integrate neural networks with induction, though it is hard to see how each approach is used or integrated with the other: 'The system integrates a neural network with minimal entropy coding using the ID3-algorithm…' Karpf *Inductive modelling* 298. In any event, Karpf is clearly using inductive approaches to

allowances such as holiday pay. The equivalent in common law jurisdictions might be the distinction between casual staff and permanent employees. The second system was built to assist a decision-making tribunal to assess whether a person's disease was work-related. In both systems a large number of cases were used for the training set.

At this point it is clear that Karpf parted company with Tyree. Karpf noted that Tyree suggested[38] that induction works best where there are only a small number of cases and that developers in law 'should take a small number of [representative] cases…' [39] in order to generate a coherent and well-pruned decision tree. Unlike Tyree, Karpf used a great many cases that he gathered from real administrative decisions. These quasi-legal decisions are very similar to lower court determinations, since they involve relatively determinate questions about the nature of employment or whether a disease is created by employment. Karpf's approach has the benefit of using large numbers of these commonplace cases rather than Tyree's small selection of leading or hypothetical cases. The results of this experiment indicate that Karpf's approach has a better legal theoretical grounding than Tyree's, and in Part IV I discuss a taxonomy of cases which can assist the developers of legal induction systems.

It is difficult however to make further assessments of Karpf's systems. Unfortunately the paper provides little detail about the technical detail of the systems implemented. It must be said that this was not the paper's purpose since he sought primarily to assess the feasibility of computerized induction in law. It does nevertheless make any assessment of

---

derive legal rule systems, and hence his approach—and especially the nature of the cases he used—is of significance here.

38 Tyree EXPERT SYSTEMS supra note 30, at 134, 170.

39 Tyree EXPERT SYSTEMS supra note 30, at 134.

this approach virtually impossible. However on the work that is reported it appears that induction can be effectively used on a large training set of commonplace cases to generate useful rules that can then be used in a rule-based legal expert system.[40] This is an encouraging development, and one that will lead onto my discussion in Part IV of the appropriate use of cases in induction systems.

C. Review of implementations

It is interesting to note the divergent approaches in the two main implementations to date. One uses leading and hypothetical cases to induce rules, while the other uses a large number of commonplace cases. This is a major difference, and in the next section I outline a theory of which type of approach is more justifiable within law.

It is also interesting to note that there are such a small number of implementations in law. Inductive inference has been successfully used in a number of areas with a great deal of success.[41] Further, law looks to be an ideal domain for inductive learning algorithms, given its dependence on the derivation of rules from cases. The small number of legally based implementations can, it seems, be attributed to a lack of understanding of inductive algorithms rather than any misfit between the domain and the algorithm. I explore this issue further in Part IV below.

---

40 Karpf *Inductive modelling.*

41 These areas include example-based planning, medical diagnostics, and engineering modeling; see Winston ARTIFICIAL INTELLIGENCE supra note 12.

IV. Legal theory and inductive algorithms

I know by my own pot how the others boil.

French Proverb

There are a number of salient features that arise out of the preceding discussion. The following two sections outline the basic lessons which should be borne in mind when applying inductive learning algorithms to legal domains. Part IV.A explains some problems with the simplification process explained in Part II, and how it may create artifacts in the induction process. Part IV.B discusses the nature of the legal cases used to derive rules, and concludes that appellate-level cases are generally inappropriate in inductive algorithms.

A. The simplification process

In Part II.B above, I explained how a set of data encoding legal cases must typically be simplified in a number of ways. This included turning multi-value outcomes into binary outcomes, and reducing the number of values for certain attributes. The reason for this was to reduce the number of potential rules that the system induces. That is, the intention is to 'prune' very bushy decision trees, thereby reducing the number of similar rules, and deriving the most useful applicable legal rules. [42]

---

[42] In the legal environment there have been a couple of experiments in using simplification to prune the decision trees. Andrew Stranieri compared the results of neural networks, induction algorithms and expert heuristic approaches in family law cases, see John Zeleznikow, Andrew Stranieri, and Bryn Lewis. Using induction in legal expert systems in RESEARCH AND DEVELOPMENT IN EXPERT SYSTEMS, PROCEEDINGS OF EXPERT SYSTEMS 101-114 (Cambridge 1995); John Zeleznikow, Andrew Stranieri, and Mark Gawler *Split-Up: A legal expert system*

However, this approach has some dangers. Any conflation of information from real life into a small number of categories inevitably involves a human choice, and this manipulation may invalidate the conclusions drawn from the data. To take the family law example given in Part II.B, recall that the initial table included specific data as to the percentage awarded to both the wife and the husband, whereas the simplified table assumed a 40% to 60% split was an equal_split. This is somewhat arbitrary, and may introduce human-derived errors.

Further, there is the problem that induction algorithms have with conflicting cases. This is not mentioned in the example, but case-conflict is an issue that is resolved by simplification. In essence the problem is this: rule induction systems assume that the set of cases are consistent, and therefore all cases can be classified in a decision tree. They cannot readily handle inconsistent cases such as we find in law. For example, think of two legal cases which have identical values on all attributes but for unknown reasons (judicial bias, change in law over time, etc) have different outcomes.

It is essentially impossible for an induction algorithm to classify two cases which are identical on their attribute-values, but differ on their outcomes.[43] Since this occurs often in

---

which determines property division upon divorce 3 ARTIFICIAL INTELLIGENCE AND LAW 267 - 275 (1996) (Hereafter Zeleznikow, Stranieri, Gawler *Split-Up*). The ID3 algorithm produced 62 rules from the 105 case training set. This can be contrasted to the thirteen rules elicited from the domain expert. The large number of rules resulting from the use of the ID3 algorithm was not entirely unexpected. The assumption underlying the ID3 algorithm is that classes are defined by their attributes. As the ID3 algorithm continues to divide the input data set, the number of classes represented at each division is only reduced occasionally. This results in a very large decision tree and a large number of rules. This experiment showed the difficulty of using non-simplified data in rule induction systems.

43 A.S. Pannu *Using genetic algorithms to inductively reason with cases in the legal domain* in PROCEEDINGS OF THE FIFTH INTERNATIONAL CONFERENCE ON ARTIFICIAL INTELLIGENCE AND LAW 175, at 177 (1995, College Park: ACM Press) describes this problem as one of 'noise resulting from contradictory data' though it would be better to class it not as noise but rather (1) a flaw in classification due to incomplete information (either lacking chronological attributes or

law we must be wary of how we handle conflict in induction. There are a number of ways that this type of conflict is typically resolved. The first way is to simplify the data set by removing one case, on the basis that it is 'poorly decided'. However, this is a facile answer to a difficult problem and one that ignores at least one relevant case—the one discarded. Another way of resolving the problem is to introduce a new attribute under which the conflicting case might be classified. Unfortunately, there is no guarantee that this attribute will be a useful classification mechanism. For example, it is possible to introduce an attribute for 'date', which will mean that the unusual, conflicting case can be weeded out by a rule which asks whether the case was decided the date of the conflicting case. This will, of course, resolve the conflict, but using this device is hardly satisfactory: clearly, the fact that a case was decided on a particular date can hardly be the most relevant reason for distinguishing it from an otherwise identical, but conflicting, decision.

For the reasons explained above, simplification of the data set, though necessary, is somewhat problematic. This is not to say that simplifying data will inevitably lead to artifacts in the rules derived, but any simplification process must be justified and carefully monitored.

---

more obviously a new attribute resolving the conflict) or more troubling, (2) the problem may simply be that law is indeterminate as legal skeptical theories argue, see as a small sample Roberto M. Unger KNOWLEDGE AND POLITICS (1975) (arguing the basic tenet of Critical Legal Studies of the centrality of political choice in law); Stanley Fish *Working on the chain gang: Interpretation in law and literature* 60 TEXAS LAW REVIEW 551 (1982) (radical indeterminacy thesis arguing against Dworkin's chain novel view of legal interpretation) (Hereafter Fish *Working*); Stanley Fish *Wrong again* 62 TEXAS LAW REVIEW 299 (1983) (radical indeterminacy thesis as against Dworkin's criticisms) (Hereafter Fish *Wrong*); Duncan Kennedy *Freedom and constraint in adjudication: A critical phenomenology* 36 JOURNAL OF LEGAL EDUCATION 518 (1986) (judging as personal choice); James D.A. Boyle *The anatomy of a torts class* 34 AMERICAN UNIVERSITY LAW REVIEW 1003 (1985) (ability to show students the open texture of all legal reasoning). Pannu is however able to show that genetic algorithms may be capable of dealing with conflicting cases which have contradictory outcomes for identical attribute vectors. This is not so with the ID3 algorithm, nor indeed with any induction algorithm based on information theory.

With this issue out of the way, the next concern is with the nature of the cases used in inductive learning algorithms.

B. The nature of the cases

Inductive learning algorithms are similar to a number of other types of artificial intelligence techniques – specifically case-based reasoning and neural networks – in that they all rely on the use of cases to derive useful results. Given the reliance that these techniques have on the cases they use, we might expect to see a great deal written about the nature of cases, how to categorize them, and the effect that an inappropriate use of cases will have upon the reasoning process. This is not so. Therefore this Part IV.B is given over to an analysis of the different types of cases which these systems use. My fundamental argument will be that there is a separation between landmark, leading and commonplace cases, and that this distinction is relevant to the use of case-based reasoning applied to legal domains.

1. A taxonomy of cases

Kolodner defines a case as 'a contextualised piece of knowledge representing an experience that teaches a lesson fundamental to achieving the goals of the reasoner'.[44] There are aspects to this definition which should not be contentious, though they are often overlooked. For example, the contextualised nature of each case is a vital and important

---

44 Janet Kolodner CASE BASED REASONING. (1993) (Hereafter Kolodner CASE BASED REASONING).

feature of any case. The case, centered as it is within its own interpretive community,[45] means that classification of cases is difficult, generalization without understanding of the context is dangerous, and retrieval must be based not only on pattern or factor matching but also on context-matching. Let us explore these features in more detail.

The implementations of inductive learning, examined above, rely on classification using information theory. In generalizing from cases, these approaches assume that the experience being modeled has the same context. If the contexts of the past and current cases differ then the expressions, the factors, and the outcomes of the cases will necessarily mean different things. This problem is most worrying when combining cases from different jurisdictions, different court hierarchies, or even different judges. The concepts, terms and attributes may stand for very different concepts, though they are actually the same expression. This is not to say that classification is impossible, just that it must be recognized that each case assumes a context and that all cases within a classification group must carry similar contexts.

Kolodner suggests that a case necessarily '…teaches a lesson fundamental to…the reasoner'.[46] Certainly some cases do fit this description. Most notably within law, those decisions from appellate courts which form the basis of later decisions and provide guidance to lower courts do provide a fundamental lesson, or normative structure for subsequent reasoning. These cases are, on one view, formal, binding and inviolate

---

45 Dworkin posits the notion of an interpretive community, which determines the scope of interpretation, Ronald Dworkin LAW'S EMPIRE (1986). This is challenged in part by Stanley Fish, see Fish *Working*, supra note 42, and Fish *Wrong*, supra note 42.

46 Kolodner CASE BASED REASONING, supra note 44.

prescriptions for future decision-making[47] or less emphatically, beacons from which inferior or merely subsequent courts navigate their way through new fact situations. The usual name for such cases is 'landmark' or 'leading' cases, both terms indicating the importance of these cases in guiding future decisions.

Most decisions in any jurisdiction are, however, not leading or landmark cases. They are commonplace cases; occurring in lower courts and dealing with relatively minor matters such as vehicle accidents, small civil actions, petty crime, divorce, and the like. These cases are rarely, if ever, reported upon by court reporting services, nor are they often made the subject of learned comment or analysis. More importantly, each case does not have the same consequences as the leading cases. A commonplace case will fall within the socially accepted interpretation of this type of matter. For example, in contract law, rarely do cases discuss the formation of the contract in terms other than offer and acceptance, given that this is a settled question not worthy of further analysis. Equally, in torts, the categories of tort are relatively fixed and the determinations at lower court level apparently determinate. This is not to say that the law is static at this lower court level. One may, for example, analyze a number of commonplace cases to see new trends emerging, but each individual case contributes only a tiny part in any change of the law, if indeed it contributes at all.

Leading or landmark cases are therefore of a fundamentally different character to commonplace cases.[48] Leading cases will individually have a profound effect on the subsequent disposition of all cases in that domain, whereas commonplace cases will only

---

47 Roscoe Pound *Mechanical jurisprudence* 8 COLUMBIA LAW REVIEW 605 (1908).

48 It is worth noting that leading and landmark cases may go unrecognized at the time of decision, but become central later. In most cases however, they will be recognized immediately

have a cumulative effect, and that effect will only be apparent over time. There is also a fundamental distinction between leading and landmark cases. Though there is no generally accepted distinction between the two terms, landmark cases are really those where the law undergoes a complete sea-change. For example, cases like *Marbury v Madison* or *Brown v Board of Education* are clearly landmark cases because the entire nature of subsequent law changes at these points. Leading cases, antithetically, can then be seen as those which change a number of smaller aspects within the case regime. Leading cases will often restate definitively what the law has been, and make subtle changes to the way we view the law. However, unlike landmark cases, leading cases do not make the previous decisions irrelevant.

This division of cases into landmark, leading and commonplace is of direct relevance to inductive learning algorithms. Since rule induction systems use cases as their primary source material we must consider what types of cases researchers might appropriately use in inductive reasoners. In order to decide this, it is necessary to understand the process of inductive learning algorithms in more detail.

Rule induction from decision trees, typified by the ID3 algorithm, classifies cases into minimally-disordered clusters based on attributes and their values, and then derives rules from this classification. The initial classification process assumes that all cases carry the same weight. The addition of a single new case to the case set of a rule induction system cannot drastically alter the knowledge of the system. It will simply be classified along with all others in the set. It may change some of the classification groupings or clusters, but it will not remove old cases which will still exert a powerful influence over the classification

---

as the most important cases in the domain. Whether the case is immediately recognized or only structure derived by the algorithm. However, we know that landmark cases defy simple classification along existing lines. Landmark cases completely recreate the necessary and sufficient conditions for each outcome. This means that, in essence, new attributes must be created for the entire set to account for the change in the law or else the landmark case will conflict completely with previously decided cases. Furthermore, all old, irrelevant cases must be removed from our knowledge base, since they are no longer a good model of the legal domain under consideration.

To understand why this is so, let us consider an inductive algorithm faced with a landmark case which totally changes the law. The decision tree which has been inductively derived from earlier cases will be completely worthless. First, the decision tree will incorrectly predict the outcome of the landmark case. More important however, the previous attributes will no longer be relevant for predicting the outcome of subsequent cases. The old attributes will be either wrong, or irrelevant to the new legal regime introduced by the landmark case. Until the landmark case, the new considerations would not have been mentioned in previous decisions and so would not be included as part of the attributes in the old set. Essentially all the previous cases, together with their attributes, must be discarded and only the landmark case and its attributes encoded. All old cases then become irrelevant following the introduction of a landmark case.[49]

This means that induction algorithms will be inappropriate where landmark cases have occurred, are likely to occur, or even possibly may occur. Since virtually any legal domain

---

becomes leading/landmark over time does not affect the analysis discussed in this section.

49 For an attempt to use induction in this manner, see Rissland and Friedman *Change*, supra note 32.

has this potential we must remain suspicious of induction as a general model of legal reasoning.

The same problem also occurs with leading cases, but to a lesser extent. Leading cases tend not to rewrite the entire attribute set, but may vary it significantly. It is common for leading cases to generate conflict on the old attribute set, since leading cases often introduce a new concept which is used for interpreting the domain. Further, since the case would not have reached appellate level without being open to multiple interpretations, it is likely that any decision made will not be on the basis of a simple, existing legal interpretation. Therefore in domains based on leading cases we will see the same issue: the likely conflict between cases based upon the old attribute set, and the need to incorporate a new attribute or attributes to 'explain' the new decision.

With commonplace cases however we are less likely to be presented with this problem. Commonplace cases decided at lower court levels are generally not decided on fine or technical points of law, and instead are disposed of on simple legal grounds which have much more to do with the interpretation of facts than of the law.[50] Since these cases can be classified simply on a relatively static set of attributes they are extremely suitable for induction systems that rely on attribute stability. Without attribute stability each new case would force a complete recoding of all cases using new attributes, determined at the time the new case is decided. This is plainly not feasible for artificial intelligence models of induction.

---

50 Jerome Frank called the law school's emphasis on appeal decisions, 'appellate court-itus', Jerome Frank LAW AND THE MODERN MIND (1963). He suggested that the idea that most law was done in these upper courts as the 'upper court myth.' This is borne out by the heavy emphasis placed by socio-legal theorists on prosecution, enforcement and lower court decisions, see for example Roger Cotterell THE SOCIOLOGY OF LAW (2nd ed, 1992).

The problem expressed above relates to the use of an induction system without a deep understanding of the relative weight of each case, particularly when dealing with landmark cases. Unless the creator of the induction system is aware of the relative importance of each case, the output derived by the induction algorithm will be flawed. Hence, the cases used by rule induction systems should be the commonplace ones, which have a cumulative effect on the law and which reflect the law, rather than reconstruct it.

2. Applying the taxonomy

It is instructive to take this lesson and examine the implementations made to date. As indicated in Part III.B, the implementations undertaken by Karpf relied on a large number of lower-level cases. This is consistent with the argument presented above, and the outcome of those experiments show that commonplace cases work well with inductive learning systems.

In contrast to this, Tyree used leading cases from the law of trover. As indicated above, this is problematic. Trover is an inappropriate domain, for two reasons. All of the cases in the trover domain were high-level appellate decisions, which came from the English High Court or above. This is contrary to my argument that leading cases were dangerous and inappropriate in induction systems. Secondly, there were only eight cases in the training set, which tends to undermine the strength of the inductive inference drawn.[51]

---

51 For a discussion of this point in induction generally, see Hunter *No Wilderness*, supra note 2, and Golding LEGAL REASONING, supra note 4, 43-44.

Further to this, Tyree suggested that hypothetical cases may be used to build inductively generated expert systems.[52] He even argued that real cases should be avoided:

> It is not generally suitable to use actual cases as examples in order to build a legal expert system using inductive methods. This is not only because of the lack of suitable examples, but also because of the desire to base the system on a small number of representative cases.[53]

He gave no reasons for either assertion. As to the first assertion—that suitable examples are lacking—it would seem that virtually any domain with a set of discrete cases which are either leading or preferably commonplace would be suitable as 'examples' from which to derive rules using induction.

As to the second point, it is unclear why we need base the system on a small number of representative or even hypothetical cases. Possibly Tyree is referring to the tendency for inductive methods to generate inordinately bushy decision trees which lead to largely useless rules.[54] There are two answers to this concern. First, one can apply the simplification process explained above.[55] Alternatively, inductive methods are capable of generating small trees provided all cases fall within a clear classification structure. In commonplace cases, like those found in lower courts, the cases are often very clearly of a simple classification. If they are not, then we must ask why this might be so. For lower court cases are decided on generally clear law and if they fail to fall into clear groupings or clusters then something unusual must be going on. Inductive methods could then be used

52 Tyree EXPERT SYSTEMS supra note 30, at 168.

53 Tyree EXPERT SYSTEMS supra note 30, at 170.

54 See Zeleznikow, Stranieri, Gawler *Split-Up*, supra note 42.

55 Supra Parts II.B and IV.A.

in much the same way as statistical methods are used by socio-legal research to uncover the actual process of decision-making in given domains.[56]

Hence, contrary to Tyree's argument, it would be better to use real commonplace cases, rather than hypothetical leading cases. Using imaginary cases carries with it enormous dangers in generating spurious decision trees which may accurately represent one user's view of a domain, but which does not accurately reflect the rules underlying the domain.

V. The future of inductive learning in law

> Learning, n. The kind of ignorance distinguishing the studious.
>
> Ambrose Bierce, *The Devil's Dictionary*

Induction is an interesting model of legal reasoning, since it provides a method of modeling legal principles and rules, and adjusting these principles and rules over time as the law changes. I have suggested that induction has potential in modeling law, but that the artificial intelligence implementations to date are too speculative and under-developed to assess whether induction may yet become a generally useful technique in modeling legal reasoning. I have argued however that induction should be further investigated, since it has the potential to be an extremely useful mechanism for understanding legal domains.

---

56 See Hunter, *No Wilderness*, supra note 2, and Hunter *Minds*, supra note 3.

Induction is also interesting because it has the potential to bridge the gap between explicitly symbolic reasoning approaches, like deductive inference and analogical reasoning, and statistical approaches like neural networks. Induction relies implicitly on statistical reasoning as the basis for the strength of any inductive inference. However, it generates symbolic rules that can be manipulated symbolically by lawyers. This has great appeal for lawyers who are used to reasoning with symbolic information and who often distrust statistics. This alone indicates that inductive learning is an approach that should be investigated further.