2006

Computer Models for Legal Prediction

Kevin D. Ashley
University of Pittsburgh School of Law, ashley@pitt.edu

Stephanie Bruninghaus
University of Pittsburgh Graduate Program in Intelligent Systems

ARTICLE

COMPUTER MODELS FOR LEGAL PREDICTION

Kevin D. Ashley
Stefanie Brüninghaus*

ABSTRACT: Computerized algorithms for predicting the outcomes of legal problems can extract and present information from particular databases of cases to guide the legal analysis of new problems. They can have practical value despite the limitations that make reliance on predictions risky for other real-world purposes such as estimating settlement values. An algorithm's ability to generate reasonable legal arguments also is important. In this article, computerized prediction algorithms are compared not only in terms of accuracy, but also in terms of their ability to explain predictions and to integrate predictions and arguments. Our approach, the Issue-Based Prediction algorithm, is a program that tests hypotheses about how issues in a new case will be decided. It attempts to explain away counterexamples inconsistent with a hypothesis, while apprising users of the counterexamples and making explanatory arguments based on them.

CITATION: Kevin D. Ashley and Stefanie Brüninghaus, Computer Models for Legal Prediction, 46 Jurimetrics J. 309–352 (2006).

Today's computerized databases of legal decisions beg an important question: Can software generalize from patterns in and across cases to improve understanding of legal domains, analyze new problem scenarios, predict new case outcomes, and justify those predictions with explanations and arguments? This is not a new question. Since computers first appeared, researchers have attempted to use them to investigate legal domains, analyze problems, and predict outcomes of legal disputes. They have refined and applied not only statistical tools, but also tools developed by researchers in Artificial Intelli-

---

*The authors are, respectively, Professor, University of Pittsburgh School of Law, and Doctoral Candidate, University of Pittsburgh Graduate Program in Intelligent Systems. The research described here has been supported by Grant No. IDM-9987869 from the National Science Foundation.

gence (AI) and Machine Learning (ML) to discover implicit regularities in data and to induce rules. These investigations have shown much promise but have also uncovered pitfalls. Legal scholars have argued that predictions based on ML or data-mining techniques can provide a window into legal domains and reasoning,[1] and they have urged a renewed look at what the applications of these techniques may reveal, including predicting the future behavior of judges and discovering the most predictive types of explanatory legal concepts.[2]

Yet, problems of representing textual cases for purposes of prediction are still major hurdles, and most prediction approaches have not been able to explain predictions in terms of legal reasons that are meaningful to legal practitioners. This article argues that automated prediction should be seen as a technique for bringing information contained in particular sets of cases to bear upon the legal analysis of a problem situation and should be assessed in terms of how well it supports such legal deliberation. The paper compares different approaches to automating prediction in terms of how well they apply the case information to make accurate predictions, as well as how well they explain the applicable case information and justify the prediction. Treating legal deliberation and prediction as analogous in some respects to a process of testing scientific hypotheses,[3] the paper proposes a computerized Issue-Based Prediction (IBP) algorithm to frame and test predictive hypotheses against data.[4] The hypotheses concern the legal outcomes of factual disputes, and the

---

1. Dan Hunter, *Near Knowledge: Inductive Learning Systems in Law*, 5 VA. J.L. & TECH. 9, 71–72 (2000).

2. Frederick Schauer, *Prediction and Particularity*, 78 B.U. L. REV. 773, 774–75 (1998).

3. According to S.L. Hurley, legal deliberation and scientific theorizing are both "responsible to the data to be explained .... In both areas the data in some sense determines the best theory.... [S]ituations that are relevantly similar in respect of data must be treated consistently in theoretical respects, or, more briefly, . . . like cases should be treated alike." S.L. Hurley, *Coherence, Hypothetical Cases, and Precedent*, 10 OXFORD J. LEGAL STUD. 221, 230–34 (1990). The analogy should not be taken too far. As Hurley warns, scientific hypotheses based on causal theories "that account for what has happened in well-designed experiments generate predictions which are then tested against the results of further experiments. By contrast, ethical and legal deliberation ... has a normative role: to give guidance in extending consistently to the case at issue a series of settled ethical or legal judgments about what should be done when the applicable ethical or legal reasons conflict .... Deliberative hypotheses are used to generate not mere predictions of decisions and actions, but decisions and actions themselves . . . they are tested against cases, both actual and hypothetical, in which the right answer about how a conflict of reasons should be resolved is settled." *Id.*

4. IBP's approach is scientific in Llewellyn's sense of the term. "A scientific *approach* to prediction we may have, and we may use it as far as our materials will permit. An exact science *in result* we have not now." KARL N. LLEWELLYN, THE BRAMBLE BUSH 52 (1930) (emphasis added). While taking into account facts and outcome, the IBP approach does not take into account a case's procedural setup, as Llewellyn recommended. The algorithm's method of breaking problems down by issues, comparing cases, and trying to explain away counterexamples is an attempt to systematize comparisons to achieve, where possible, "a perfect working out of comparison and difference ... an *experimentum cruces*." *Id.* The algorithm's focus on explaining away counterexamples to a hypothesis, in effect, systematizes an approach recommended by Radin: "[I]n order to estimate the probable action of a court, it is well to read the books which the court will read and will study, and to refute, if we can, whatever we find in those books that runs

"experimental data" are the decided cases stored in a computerized case database. Given the facts of a new dispute, IBP identifies which legal issues they raise, formulates hypotheses about who should prevail on each issue, tests the hypotheses against past cases in its database, attempts to explain away cases inconsistent with a hypothesis, determines which party is favored for each issue, and combines the analyses into an overall prediction.[5] The paper concludes that IBP does a better job of identifying relevant information in a collection of cases and employing it for predicting and explaining the outcome of particular problems than a plausible set of competitive approaches. IBP's predictions not only are more accurate than those of competing ML approaches, but they also generate argument-like explanations.

Part I summarizes the history of research on legal prediction. Part II presents the IBP approach to legal prediction. Part III offers a comparative analysis of the IBP approach with the major alternative approaches for automated legal prediction: rule learning, case-based algorithms (including nearest neighbor and argument-based algorithms such as IBP), and a Bayesian learning algorithm that implements a kind of statistical prediction. The algorithms are compared in terms of how well they can apply the information contained in the same database of cases to the analysis of particular legal problems. Specifically, the algorithms are compared in terms of the accuracy of their predictions, whether the algorithm can explain predictions in terms meaningful to attorneys, the nature of their generalizations, and the algorithm's ability to deal with inconsistent cases. Part IV considers some ramifications of computerized algorithms for predicting the outcomes of legal problems and for the design of future legal information systems.

I. THE HISTORY OF RESEARCH ON LEGAL PREDICTION

A. Early Work on Computerized Prediction in Law: Nearest Neighbor

In the 1950s through the 1970s, researchers first explored applying mathematical and computational tools to the task of legal prediction. Their

counter to what we wish the law to be,—or generally, to have been." Max Radin, *Case Law and Stare Decisis: Concerning PRÄJUDIZIENRECHT IN AMERIKA*, 33 COLUM. L. REV. 199, 212 (1933).

5. As her Ph.D. dissertation project at the University of Pittsburgh's Graduate Program in Intelligent Systems (ISP), Stefanie Brüninghaus designed and built SMILE+IBP, a program that can reason with textual summaries of case facts and predict their outcomes based on a database of trade secret cases. The IBP program was part of that work. *See* Stefanie Brüninghaus, Issue-Based Prediction in IBP, Chapter 3 of Generating Legal Arguments and Predictions from Case Texts (Dec. 15, 2005) (unpublished draft Ph.D. dissertation, University of Pittsburgh) (on file with author). This dissertation provides the most comprehensive account of the IBP program and the experiments performed with it.

tools ranged from weighting schemes based on judicial votes[6] to the "nearest neighbor" method for assessing case similarity[7] to "boolean lattices" (algebraic structures implementing the operations of conjunction, disjunction, negation, intersection, union, and complement)[8] to a variety of statistical techniques (for example, multiple regression, discriminant analysis, linear programming, and probit analysis).[9] They applied these tools to a variety of legal domains including U.S. Supreme Court right-to-counsel cases,[10] Swiss workmen's compensation cases, and Canadian tax cases.[11]

To apply such prediction tools, the case data must be amenable to computer processing. Legal cases, however, are not recorded in what today would be called a database format. Therefore, one must create a structured scheme of standardized descriptors to represent outcomes and factual circumstances. One then must manually extract the relevant descriptor values from each case text. Depending on the representation scheme, the values for each fact descriptor may be numbers, Boolean (true or false), or values from a predefined set of features.[12] Thus, early prediction researchers developed lists of descriptors for representing relevant general factual features of a class of cases and then manually assembled databases of typically 20 to 100 cases involving a particular issue. For example, Mackaay used Lawlor's database of 64 Canadian capital gains tax cases for which Lawlor had identified 46 fact descriptors relevant to whether the gain was "a mere enhancement of value by realizing a security, or . . . made in an operation of business in carrying out a scheme of profit making."[13] Each case was represented by 46 fact descriptors, which could be true (1) or false (0). The fact descriptors included features like "the present transaction is an isolated one," or "at the time of purchase, private party had an other intention than to resell at a profit."[14]

In making predictions, the project used a nearest-neighbor approach. The idea of the *k*-NEAREST-NEIGHBOR algorithm (*k*NN) is to determine the *k* cases closest to the problem in terms of some similarity measure and assign an outcome according to the majority of those cases. This computational method was first introduced in pattern recognition, but seemed plausible for common law applications, where "like cases are decided alike." The most important element in a nearest-neighbor approach is the similarity measure. Mackaay's

6. Fred Kort, *Predicting Supreme Court Decisions Mathematically: A Quantitative Analysis of the "Right to Counsel" Cases,* 51 AM. POL. SCI. REV. 1, 1–12 (1957).

7. Ejan Mackaay & Pierre Robillard, *Predicting Judicial Decisions: The Nearest Neighbour Rule and Visual Representation of Case Patterns*, 3 DATENVERARBEITUNG IM RECHT 302, 307 (1974).

8. *Id.* at 306.

9. *Id.*

10. Kort, *supra* note 6.

11. Mackaay & Robillard. *supra* note 7, at 303, 306.

12. Often, these schemes can be likened to the organization of a spreadsheet where the descriptors correspond to the columns, and each row corresponds to a case.

13. *Id.* at 311.

14. *Id.* at 327–31.

program uses the simplest measure of how similar two cases are: it counts the features with identical values. Other researchers have applied the nearest-neighbor algorithm to legal prediction tasks using more complex similarity measures that assign weights to different fact descriptors.[15] These measures take into account the correlation between features and outcome or the relative spread of the features' values and assign greater weight to closer neighbors or weigh each attribute differently according to some measure of its relevance.

The pioneer researchers also developed a means for objectively evaluating their prediction techniques. Lawlor employed a "cyclical sampling" approach —now more commonly known as a "leave-one-out" experiment—to test his prediction scheme. Each of the cases in the collection was designated as a test case once and removed from what is often called the training set. The outcome of the test case is then predicted using the training set. So, for every case in the collection, exactly one prediction is made by the system. Using a leave-one-out procedure, Mackaay compared his nearest-neighbor approach to the predictions of an independent lawyer and to two other approaches that computed descriptor weights. He compared the methods in terms of accuracy (i.e., the fraction of the number of cases the method correctly predicted out of the total number of cases). The nearest-neighbor approach and the independent lawyer did best: each got only four of 64 cases wrong, but the sets of cases where mistakes were made did not completely overlap between the different algorithms and the human expert.[16]

Mackaay carefully studied the cases in which any of the predictive methods erred. His program visually displayed the cases so that their apparent distance reflected their similarity. The picture revealed PRO and CON regions of cases with a rough border in between. Significantly, many of the cases that any prediction method had gotten wrong appeared close to the PRO-CON border. Tax law experts related the borderline cases to trends in tax law and to particular contemporary cases in which tax law journals provided in-depth analysis.[17]

Although the nearest-neighbor approach can show where a new case fits in with its most similar neighbors and how close it lies to case outcome boundaries as well as provide some explanation by listing the features shared and not shared among neighbors, it normally does not explain why the shared features account for a result or why unshared features lead to different results because it usually lacks additional information about the features' legal significance. Moreover, similarity in such an approach is often defined in a way that is oddly inappropriate to law. Although it takes feature overlaps into account, it ignores the features' legal meaning and the relationships among the

15. *See* JAMES POPPLE, A PRAGMATIC LEGAL EXPERT SYSTEM 40–41, 75–82, 87–89, 146–51 (1996).

16. MacKaay & Robillard, *supra* note 7, at 313.

17. *Id.* at 310–313.

sets of features that cases share with a problem.[18] Thus, in a nearest-neighbor approach, two quite different cases can be equidistant from a case whose outcome is to be predicted. There is no guarantee that neighboring cases share a core of features from which one might fashion an explanation or argument as to why they should be decided similarly. If they do share such features, it is an empirical coincidence. This is one reason why a good scheme for assigning weights can help: the more crucial features would naturally have higher weights, and neighboring cases would be more likely to share them.

Although many nearest-neighbor approaches incorporate weighting schemes, assigning weights to legal-fact descriptors may often be problematic. A fact descriptor's "weight is highly contextual and depends on individual problem situations."[19] A normally insignificant descriptor may be important in a particular context. Often an attorney may need to argue for a PRO result in a problem case whose nearest neighbors are CON. For purposes of argument, empirically assigned descriptor weights may not help to identify reasons why the problem is exceptional or why the shared descriptors should *not* determine the outcome in these circumstances. Producing such arguments and explanations requires one to understand more of the legal reasons, even the principles and policies underlying the fact descriptors, and how they interact in the problem's context.[20] The *k*-NEAREST-NEIGHBOR algorithm, nevertheless, is a staple in the arsenal of computational tools for reasoning with cases in a variety of domains.

B. Lessons from Early Prediction Work: Statistical Analysis

Another approach to legal prediction focused on statistical analyses of published judicial decisions.[21] Carried out in law schools, it represented a kind of "systematization of traditional legal research. Instead of reporting the fruits of years of subjective reading of opinions that had crossed one's desk, the legal scholar turned to selecting randomly, coding tirelessly, and then analyzing hundreds of cases."[22] After reading cases from a chosen legal domain,

---

18. The similarity measures in the *a fortiori* models of case-based legal reasoning described in Part I.D.1 (HYPO, CATO and IBP) do take such additional information into account.

19. KEVIN D. ASHLEY, MODELING LEGAL ARGUMENT: REASONING WITH CASES AND HYPOTHETICALS 175 (1990).

20. Part IV, *infra*, presents some approaches to this challenge.

21. Political scientists developed predictive methods to investigate judicial decision making. They expanded lists of fact descriptors beyond "legally relevant" facts to include anything that could predict a judge's decision, such as "the policy attitudes or preferences of the Justices [as evidenced by] the party affiliations of the Presidents who appointed them." Schauer, *supra* note 2, at 784–85. The work tended to show that attitudinal factors were more influential in predicting legal decisions than the legal criteria, a conclusion that has been criticized as an artifact of the biased selections of cases and on other grounds. *Id.* at 785 n.32; *see also* Theodore W. Ruger et al., Essay, *The Supreme Court Forecasting Project: Legal and Political Science Approaches to Predicting Supreme Court Decisionmaking*, 104 COLUM. L. REV. 1150 (2004).

22. Kevin M. Clermont & Theodore Eisenberg, *Litigation Realities*, 88 CORNELL L. REV. 119, 125–6 (2002).

researchers define the legal categories, decide on a comprehensive list of cases, and construct as exhaustive a list of descriptors as possible for representing cases in terms of whatever may contribute to outcomes.[23] The cases are coded manually, and then processed by statistics software. Taking into account the presence and absence of the descriptors, the program computes correlations between descriptors and decisions and the descriptors' predictive weights.

The goal of this program is to create a target function (i.e., a mathematical function that assigns to a new case a classification such as "plaintiff wins"). In logistic regression, for instance, the target function estimates the probability of a party's winning using an exponential function for the features.[24] The exponential functions include feature weights, which are derived from the training data using advanced statistical methods. More significant features will have a higher weight and thus a higher impact on the prediction. However, the interaction between features is very complex and small differences can have huge effects. Although experienced statisticians may have some intuitions about the behavior of these models, one cannot generate legal arguments and explanations from the weights. As a result, the methodology's limitations make statistical analysis of judicial decisions a very risky undertaking. One needs a representative sample of all relevant cases, but "judicial decisions represent only the very tip of the mass of grievances [and] are a skewed sample of that tip of judicial decisions."[25]

Considerations that complicate reliance on predictions include small and skewed sample size, time lags that do not account for changes in the law,[26] and variables not taken into account in the model, including the procedural setting and unarticulated judicial attitudes.[27] Judges may not have disclosed the features that influenced their decision or stated their rationales accurately or

---

23. For example, legal criteria, factual criteria, race and sex of parties, and judges' names have been used as descriptors.

24. Logistic regression requires a certain degree of manual preprocessing. The features have to be statistically independent; the inclusion of two features that impact each other often leads to lower accuracy in assessing the weights of the features. In addition, these models usually perform best if only a carefully selected feature subset is included. Although it may seem counterintuitive, the models tend to do better with less information. While Machine Learning research provides some guidance for this feature selection, it remains largely a matter of trial and error. *See, e.g.*, Ruger et al., *supra* note 21. Along the same lines, fairly large datasets are necessary. There may be too few cases "to draw statistically meaningful conclusions except for very general issues. In some situations, this small numbers problem may prevent identification of trends that would reach statistical significance given a larger sample." William M. Sage, *Judicial Opinion Involving Health Insurance Coverage: Trompe L'Oeil or Window on the World?*, 31 IND. L. REV. 49, 61 (1998).

25. Clermont & Eisenberg, *supra* note 22, at 125–26.

26. Sage, *supra* note 24, at 62.

27. The "attitude of [a] judge toward the granting of motions to dismiss" might make a difference. Marjorie Anne McDiarmid, *Lawyer Decision Making: The Problem of Prediction*, 1992 WIS. L. REV. 1847, 1888 (1992). Or, as Schauer wryly suggests, the fact that the party seeking or defending against an injunction in a West Virginia Supreme Court case is a coal company could be significant. Schauer, *supra* note 2, at 782–83.

completely.[28] Statistical analysis of judicial decisions is subject to a further criticism: the method could not generate legal explanations of its predictions. There is no simple way to interpret the predictions in terms of legal reasoning or argument. One can list the coefficients in a logistic regression, but these do not correspond to any obvious legal argument. At best, one can see where a new case lands in the list of past cases ordered by the probability of their outcomes and compare the new case to its neighbors on the ordered list. As with nearest neighbor, however, the ordering captures cumulative attribute weights across all cases. It may not lend itself to constructing legal arguments as to why one case is stronger than another.

C. Machine Learning in AI & Law

Practitioners of Artificial Intelligence and the Law (AIL) have wrestled with how to achieve both accurate prediction and plausible legal explanations. They have developed techniques for representing legal knowledge in computational models to investigate legal reasoning from a jurisprudential viewpoint and to build legal expert systems and intelligent tutoring systems for teaching law and improving legal information retrieval. One AIL approach has involved rule learning, a Machine Learning (ML) technique that induces predictive rules from decided legal cases. Since induction programs generate rules to make the predictions, the rules can also be used to explain those predictions. The induced rules "are obviously relevant in 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."[29] For instance, an early project applied a well-known ML algorithm (ID3) to derive a decision tree from a database of cases dealing with the division of property upon divorce.[30] A decision-tree learner generates a structured tree of logical tests for determining the target label—here, a case's predicted outcome. The algorithm first selects the attribute best correlated with the outcome of the case. It then creates a decision point with the attribute as test and uses its value to separate the training set into two subsets, each of which will be handed down the respective branch in the tree. This process is repeated recursively, until the branch only contains positive or negative examples, at which point a leaf node with the respective labels is added to the tree. A new example is passed down through that tree until it reaches a leaf node, whose value is used as the prediction. Many different implementations of the basic algorithm exist, which may apply different statistical methods to select the attributes on which to split, to decide what to do if the algorithm cannot find a

28. Sage, *supra* note 24, at 66–67.

29. Hunter, *supra* note 1, at 12.

30. JOHN ZELEZNIKOW & DAN HUNTER, BUILDING INTELLIGENT LEGAL INFORMATION SYSTEMS: REPRESENTATIONS AND REASONING IN LAW, 264–69 (1994).

set of logical tests to discriminate between a set of positive and negative instances, and to simplify the tree.[31]

Zeleznikow and his colleagues applied this technique to a database of cases dealing with whether a transaction involved the deferral of debt, an element in determining whether the transaction is regulated under a Victoria, Australia, Credit Act. A set of 27 cases was represented in terms of five Boolean attributes listed in Figure 1. The learning algorithm induced a tree, shown in Figure 1, with six leaf nodes shown as boxes with a count of the respective training cases.[32] Five of the leaf nodes are "pure" in the sense that they contain only cases in which there was no debt deferral (labeled "no") or cases in which there was debt deferral (labeled "yes"). These pure leaf nodes correspond to a predictive rule. The rule is a path through the tree from the root node to the leaf node with the indicated test results. For instance, *Rule 1* states, If **pay_deferred** = **yes** and **benefit** = **yes** and **duty_pay** = **yes** and **in_money** = **yes** then **debt_deferral.**[33] Rules like this can be used not only to make predictions, but also to explain them. The rule justifies the inductive inference of debt deferral in a new case where the answers to the tests correspond to the pattern captured in Rule 1's preconditions. In four such cases, debt deferral was found.

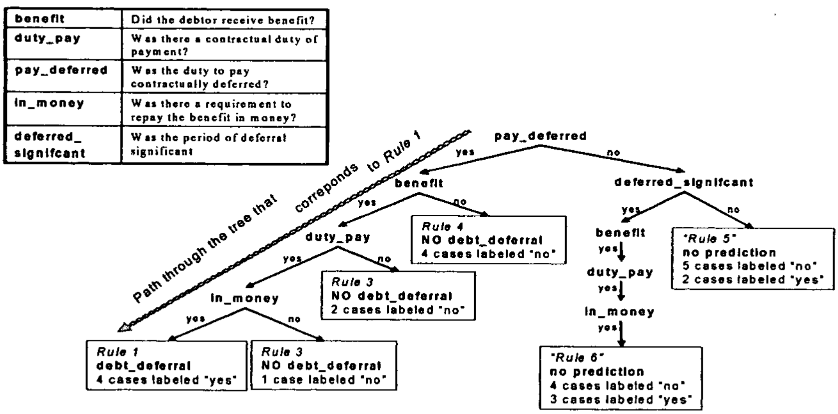

Figure 1: Decision Tree for debt_deferral

31. The best known implementations include ID3 and its successor, C4.5. *See* J. ROSS QUINLAN, C4.5: PROGRAMS FOR MACHINE LEARNING (1993).

32. George Vossos, Incorporating Inductive Case-Based Reasoning Into an Object-Oriented Deductive Legal Knowledge Based System 146, 157 (May 1995) (unpublished Ph.D. dissertation, Latrobe University) (on file with author).

33. *Id.* at 158.

Although such rules are explanatory, they are still inadequate as legal explanations.[34] They do not give reasons for a decision because they do not employ abstract legal concepts, drawn from statutes, legal principles, or underlying policies, normally used to explain a predicted decision. Moreover, as illustrated in the study below, the induced rules may make accurate predictions but not correspond to patterns of reasoning that are familiar to attorneys.

Organizing a database of cases for making predictions can summarize useful information about how that body of cases bears upon a new case. Even a predictive "rule" that fails may convey useful information. In the decision tree of Figure 1, for instance, the leaf node corresponding to *Rule 5* is not pure: it contains three cases that found debt deferral and four others where no debt deferral was found. As the quotation marks around *Rule 5* indicate, this is not a predictive rule. "It is essentially impossible for an induction algorithm to classify two cases which are identical on their attribute-values, but differ on their outcomes."[35] Zeleznikow and Vossos treated such "rules" as indices in the database. In a new case, if **pay_deferred** = **no** and **deferred_significant** = **yes** and **benefit** = **yes** and **duty_pay** = **yes** and **in_money** = **yes**, then there are seven cases in the database similar to the new case. Four of them indicate no deferral of debt and three indicate the contrary. Such a "rule" is not itself the basis for a prediction, but it does point to relevant yet inconsistent cases in the database.[36] Perhaps reading the cases may turn up some basis for deciding that the problem is more relevantly similar to one group or the other in ways not reflected in the cases' descriptors.

D. AIL, Argumentation, and Prediction

The problems of the early prediction work and the later statistical approaches at first led a number of AIL researchers to eschew prediction altogether in favor of automating legal argument. Modeling case-based legal argumentation promised a way around the problems of prediction. Indeed, it led to advances in computationally modeling legal reasoning. As it happened, however, some approaches to modeling case-based legal argumentation can also be applied to making predictions. Recently, researchers in AIL have used prediction results to support the claim that argumentation models capture

34. In law, "[r]egularity of experience is a poor warrant for any legal outcome, and we generally seek to abstract some more general principle. . . . [I]nductive inference alone cannot provide warranted belief in the generalization derived by the inductive process. We need to generate an explanatory or justificatory principle." Dan Hunter, *No Wilderness of Single Instances: Inductive Inference in Law*, 48 J. LEGAL EDUC. 365, 382–83 (1998). As discussed in Parts III and IV, enabling a program to use such justificatory principles or integrate them into their explanations of predictions is still a matter of research.

35. Hunter, *supra* note 1, at 47–48.

36. Vossos, *supra* note 32, at 157–58.

important aspects of legal reasoning.[37] This development has raised the tantalizing prospect of programs that can make predictions, explain them plausibly, and even make arguments attacking the predictions in terms that legal professionals can evaluate.

1. A Fortiori *Models of Legal Argument*

Two recent approaches to computerized prediction and explanation, the CATO program and the Issue-Based Prediction (IBP) algorithm,[38] both assume that cases have been represented in terms of a particular kind of fact descriptor called Factors, first introduced in the HYPO program.[39] Each Factor captures a stereotypical pattern of facts that tends to strengthen or weaken the plaintiff's position concerning a particular kind of legal claim. For trade secret law, the Factors used in the CATO and IBP programs are shown in Table 1. The table shows each Factor's numerical designation, its name, the side it favors, and its meaning.

Table 1: Factors in IBP, CATO, and HYPO

No.	Name	Side	Meaning
F1	Disclosure-in-Negotiations	D	P disclosed its product information in negotiations with D.
F2	Bribe-Employee	P	D paid P's former employee to switch employment to get P's product information.
F3	Employee-Sole-Developer	D	Employee D was sole developer of P's product.
F4	Agreed-Not-to-Disclose	P	D entered into nondisclosure agreement with P.
F5	Agreement-Not-Specific	D	Nondisclosure agreement did not specify info to be treated as confidential.
F6	Security-Measures	P	P adopted security measures.
F7	Brought-Tools	P	P's former employee brought product development information to D.
F8	Competitive-Advantage	P	D's access to P's product information saved D development time or expense.
F10	Secrets-Disclosed-Outsiders	D	P disclosed its product information to outsiders.
F11	Vertical-Knowledge	D	P's info is about customers and suppliers (and thus available independently).
F12	Outsider-Disclosures-Restricted	P	P's disclosures to outsiders were subject to confidentiality restrictions.

37. *See, e.g.*, Vincent Aleven, *Using Background Knowledge in Case-Based Legal Reasoning: A Computational Model and an Intelligent Learning Environment*, 150 ARTIFICIAL INTELLIGENCE 183, 183–237 (2003).

38. Aleven, *supra* note 37, at 183; Stefanie Brüninghaus & Kevin Ashley, *Predicting Outcomes of Case-Based Legal Arguments, Proceedings of the Ninth International Conference on Artificial Intelligence and Law* 233–42 (2003).

39. *See* ASHLEY, *supra* note 19.

No.	Name	Side	Meaning
F13	Noncompetition-Agreement	P	P and D entered into a noncompetition agreement.
F14	Restricted-Materials-Used	P	D used materials subject to confidentiality restrictions.
F15	Unique-Product	P	P was only manufacturer of product.
F16	Info-Reverse-Engineerable	D	P's product information could be learned by reverse engineering.
F17	Info-Independently-Generated	D	D developed its product by independent research.
F18	Identical-Products	P	D's product was identical to P's.
F19	No-Security-Measures	D	P did not adopt any security measures.
F20	Info-Known-to-Competitors	D	P's information was known to competitors.
F21	Knew-Info-Confidential	P	D knew that P's information was confidential.
F22	Invasive-Techniques	P	D used invasive techniques to gain access to P's information.
F23	Waiver-of-Confidentiality	D	P entered into an agreement waiving confidentiality.
F24	Info-Obtainable-Elsewhere	D	P's info could be obtained from publicly available sources.
F25	Info-Reverse-Engineered	D	D discovered P's information through reverse engineering.
F26	Deception	P	D obtained P's information through deception.
F27	Disclosure-In-Public-Forum	D	P disclosed its information in a public forum.

Factors are a kind of expert knowledge about the claim. Statutes, case opinions, law review articles, Restatements and other scholarly works often identify fact patterns that are legally relevant, affect the strength of a claim, and apply to more than a few cases. For a fact pattern to be deemed a Factor, a judge in at least one opinion must have indicated that the case was decided as it was because of, or in spite of, the presence of such facts. A case is represented in terms of those Factors whose fact patterns appear to be present as a matter of direct inference from the text. A Factor does not apply if either the fact pattern is known to be absent, or there is insufficient information reported from which to infer that the fact pattern is present.[40]

Since Factors represent a claim's relevant factual strengths and weaknesses, they support an *a fortiori* model of a precedent's persuasive force. If a problem has all the precedent's Factors relevant to a conclusion, including at

40. As noted, Factors represent factual patterns that tend to strengthen one side's claim. Ordinarily, the absence of a Factor favoring one side is not counted as a strength of the opposing side. When courts specifically refer to the absence of a pattern associated with a Factor as a strength for the opposing side, however, a new Factor will be created to represent the absence of the pattern. Another exception is in distinguishing a cited case. *See infra* note 43.

least one factor favoring the conclusion, but no additional factors favoring the opposite conclusion, it follows that the conclusion is favored as strongly in the problem as in the case. One can make a reasonable argument by analogy that the problem should be decided *a fortiori* as the case was. The analogy comprises the set of shared Factors relevant to the conclusion. If the problem has some additional Factors favoring the conclusion, or the case has some additional Factors favoring the opposing side, then the conclusion is favored even more strongly in the problem than in the case. If, on the other hand, the case has some Factors favoring the conclusion not shared in the problem, or the problem has some Factors militating against the conclusion and not shared in the case, then the problem's conclusion does not follow *a fortiori* from the case. The case may still be cited for the conclusion in the problem, but those unshared Factors are relevant differences between the problem and the case and can be used in an argument distinguishing the problem from the case.

The HYPO program embodied such an *a fortiori* model computationally.[41] It defined in terms of subset relations over Factors the conditions in which cases could be cited in support of a decision, and it generated arguments by analogy for and against a proposed conclusion.[42] The model assumes that a primary argument role of cases is conflict resolution; a case decision resolves strengths and weaknesses and can show how to resolve similar conflicts in future cases. HYPO used its *a fortiori* model not to make predictions but to generate actual arguments by analogy that attorneys can read and evaluate as legal arguments.[43] HYPO records the court's holding, the Dimensions (i.e.,

---

41. HYPO's *a fortiori* model bears some resemblance to a technique employed in the early prediction work. Kort and Lawlor introduced "the idea of polarization of variables and of case ranking. A binary variable is correctly polarized if its '1' state is more favourable to a PRO decision than its '0' state. Given the assumption of independence of variables, one can define a partial order amongst cases: a case A is ranked above case B if the fact pattern of A can be obtained from that of B by changing at least one '0' state into '1'. If a case is decided PRO, it may then be inferred that all higher ranked cases should also, and inversely if it is decided CON, all lower ranked cases should be so decided." Mackaay & Robillard, *supra* note 7, at 306 n.11. Lawlor referred to this phenomenon as convex consistency, which "is applied only to pairs of cases in which the facts present in one case are a sub-set of the facts present in the other case." Reed Lawlor, *Axioms of Fact Polarization and Fact Ranking—Their Role in Stare Decisis*, 14 VILL. L. REV. 703, 719 (1969); Franklin Fisher, *On the Existence and Linearity of Perfect Predictors in "Content Analysis,"* 2 MODERN USES OF LOGIC IN LAW 1 (1960). He designed techniques for assigning weights to polarized fact descriptors for purposes of predicting case outcomes mathematically. He also noted that polarized fact descriptors were useful in modeling argument phenomena like distinguishing. Lawlor, *supra*, at 716–17. Lawlor's was a model of prediction, however, not of argumentation. He did not implement a computer program that could generate arguments automatically.

42. ASHLEY, *supra* note 19; Kevin Ashley, *An AI Model of Case-Based Legal Argument from a Jurisprudential Viewpoint*, 10 ARTIFICIAL INTELLIGENCE AND LAW 163, 206 (2002).

43. There are a number of differences between Lawlor's polarized descriptors and Factors. Lawlor's prediction approach assumed that a case was represented by a list of binary features that were either present as strengths for the plaintiff or absent and thus strengths for the defendant. By contrast, in HYPO each Factor was represented as a Dimension, a knowledge representation device Ashley invented to capture such information as the prerequisites for applying a Factor to a case, the range of values a Factor might have in a case, and the direction in the range that favored

knowledge structures representing Factors) consistent with the holding, and those that are not. HYPO uses such information not for prediction or for adjusting attribute weights but to generate arguments showing how courts previously have resolved conflicting factual strengths and weaknesses.

2. *Making Predictions with CATO's Argumentation Criteria*

To improve the arguments that a program like HYPO could generate, Aleven invented a hierarchical representation of the meaning of Factors. The Factor Hierarchy relates each Factor to one or more issues or elements in trade secret law and thereby provides more abstract reasons why the Factor matters. The Factor Hierarchy excerpts of Figure 2 illustrate how Factor F12, Outsider-Disclosures-Restricted (P), is related to the issue of whether the plaintiff's information is a trade secret (F101).[44] The presence of Factor F12 shows that the plaintiff took efforts to maintain the secrecy of the information (F102, F123) and that the information was apparently not known or available outside of the plaintiff's business (F105, F106). In addition, F12 also provides evidence against the conclusion that the information was legitimately obtained (F120). The support relations can be positive or negative (not shown) and strong or weak. Strength is associated not with a weight, but with whether a conclusion may be blocked for certain purposes of argument in the presence of opposing evidence.

CATO made more sophisticated arguments than HYPO could. CATO could do more reasoning about the legal significance of similarities and distinctions. It could also organize the cases by the issues in the Factor Hierarchy and generate multi-case arguments for plaintiff and defendant. In addition, for any case that was distinguishable because it had extra Factors

---

plaintiff. HYPO's Dimensions were not simply binary; some had numerical and other ranges to capture magnitudes that could be relevant to the comparative strength of a Dimension in a particular case (e.g., fewer disclosures to outsiders were better for plaintiff's trade secret misappropriation claim). Cases in HYPO were not treated as being represented with values for all the features. Rather, HYPO represented a case primarily as the set of the proplaintiff and prodefendant Dimensions that were known to apply to it (i.e., whose prerequisites were satisfied in the case.) HYPO did not generally treat the absence of a proplaintiff Dimension as a positive strength for the defendant. Instead, it took the argument context into account. For instance, in distinguishing a problem from a case, HYPO (and CATO) would call attention to the absence in the problem of a Dimension in the case that favored the case's outcome, making the cited case stronger for that side than the problem. In the context of distinguishing an opponent's case, this is a reasonable argument move. For another example, as a way of showing how an argument could be strengthened, HYPO could hypothesize that a missing (i.e., near-miss) Dimension applied and demonstrate the new arguments that would ensue as a result. Another difference concerns weighting. For prediction, Lawlor treated it as axiomatic that "fact descriptors can be ranked relative to each other in accordance with their relative strengths with reference to a particular issue. A fact of higher rank has greater strength or weight than a fact of lower rank." Lawlor, *supra* note 41, at 706. In designing HYPO, Ashley eschewed assigning quantitative weights to Dimensions. ASHLEY, *supra* note 19.

44. Aleven, *supra* note 37, at 192. In the Factor Hierarchy, base-level Factors have two-digit identifiers (see Table 1); issues and reasons (i.e., abstract Factors) have three-digit identifiers.

supporting a conclusion or that lacked some of the Factors opposing it, CATO could emphasize or downplay the significance of the distinction in terms of the issues and reasons why they mattered. Interestingly, in the Factor Hierarchy, a Factor might have more than one interpretation; it could be related to more than one issue or reason for its significance. This meant that in downplaying or emphasizing distinctions, the program had to make choices of how best to characterize a distinguishing Factor. Aleven's algorithm decided which reasons to pursue and how abstractly to characterize them in terms of context-sensitive criteria in order to undercut Factors in the same case, to draw an abstract contrast, or to emphasize corroborating support.[45]

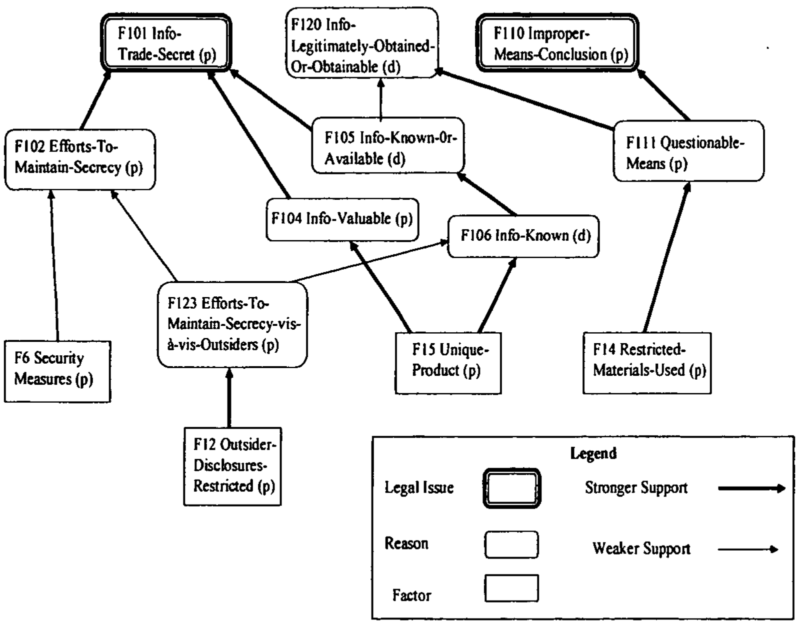

Figure 2: Factor Hierarchy Excerpts

45. *Id.* at 205–07.

The following example shows how CATO analyzes a trade secret law case, *K & G Oil Tool & Service Co. v. G & G Fishing Tool Service.*[46] The facts of the *K & G* case can be summarized in terms of seven Factors, five favoring plaintiff (F6 Security-Measures (P), F15 Unique-Product (P), F21 Knew-Info-Confidential (P), F18 Identical-Products (P), F14 Restricted-Materials-Used (P)) and two favoring defendant (F16 Info-Reverse-Engineerable (D), F25 Info-Reverse-Engineered (D)).[47]

In analyzing the *K & G* case, CATO (as HYPO before it) could retrieve all of the cases in the database that shared any Factors with the problem and sort them in the order of the inclusiveness of the sets of Factors each case shared with *K & G*. The program could then select the citable cases (cases with at least one Factor favoring the winner) that also had the maximal subsets of *K & G*'s Factors and for which no cases won by the opponent had more inclusive subsets. Having selected these so called best untrumped cases (BUCs), the programs made arguments analogizing them to the problem in terms of the shared Factors (i.e., relevant similarities) as well as responses distinguishing them from the problem in terms of unshared Factors (i.e., distinctions). Figure 3 shows arguments CATO generates with *Technicon Data Systems Corp. v. Curtis 1000*,[48] one of those best untrumped cases a plaintiff can cite, but one that a defendant can distinguish from *K & G*.

---

46. 314 S.W.2d 782 (Tex. 1958).

47. Plaintiff K & G developed a tool to remove metal debris from oil and gas wells. The internal construction of the device was not generally known in the business. *F15 Unique-Product (P)*. There was testimony that the design could be determined by thoroughly examining the tool, even without disassembling it. *F16 Info-Reverse-Engineerable (D)*. K & G had entered into a licensing agreement with defendant G & G by which the defendant would sublease the K & G device to other companies, in exchange for 25 percent of the rentals charged. According to the agreement, the defendant was prohibited from disassembling plaintiff's device. *F6 Security-Measures (P)*. It was mutually understood that the purpose of this agreement was to guard against anyone determining the internal construction of the tool. *F21 Knew-Info-Confidential (P)*. However, G & G did take apart the device in order to examine its internal construction. *F25 Info-Reverse-Engineered (D)*. *F14 Restricted-Materials-Used (P)*. The defendant then used information that it gained from the examination to construct its own device, which was substantially the same as plaintiff's. *F18 Identical-Products (P)*. The plaintiff had not disclosed information about the internal construction of its device to others outside its business. *Id.* at 782–88; *see also* G & G Fishing Tool Service v. K & G Oil Tool & Service Co., 305 S.W.2d 637, 638–41 (Tex. Civ. App. 1957).

48. 224 U.S.P.Q. 286 (Del. Ch. 1984).

→ **Analogize:** Where plaintiff took measures to keep its information secret [F6], defendant used materials that contained confidentiality restrictions [F14], and defendant knew that plaintiff's information was confidential [F21], even though plaintiff's information could be discovered by reverse engineering plaintiff's product [F16] and defendant reverse engineered plaintiff's product [F25], plaintiff should win a claim of trade secrets misappropriation, as in Technicon Data Systems Corp. v. Curtis 1000, Inc., 224 U.S.P.Q. 286 (Del Ch. 1984).

← **Distinguish:** *Technicon* is distinguishable. It is stronger for plaintiff than is the current problem. In *Technicon,* plaintiff imposed confidentiality restrictions in connection with its disclosures to outsiders [F12]. This was not so in *K & G.*

→ **Downplay:** In *Technicon,* plaintiff's disclosures to outsiders were subject to confidentiality restrictions [F12]. This was not so in *K & G.* However, this is not a significant distinction. First, in *K & G,* defendant used materials that were subject to confidentiality restrictions [F14] and plaintiff was the only manufacturer making the product [F15]. Therefore, in both cases, defendant may have acquired plaintiff's information through improper means (F110, F111, F120). Second, in *K & G,* plaintiff adopted security measures [F6]. In both cases, therefore, plaintiff took efforts to maintain secrecy of its information (F102, F123).

Figure 3: Sample CATO Arguments About *K & G* Case

CATO supports two argument moves HYPO did not: downplaying or emphasizing distinctions. In the example, plaintiff gets the last word, since the distinction can be downplayed. In downplaying the significance of F12 Outsider-Disclosures-Restricted (P), which *Technicon* has but *K & G* does not, the program uses the information in the Factor Hierarchy, shown in Figure 2, to draw a more abstract analogy between the cases in terms of issues and reasons they share despite the difference (i.e., F102, F110, F111, F120, F123).

Once Factors were incorporated into hierarchical representations that informed the model and made more sophisticated arguments, it was a natural step to see whether the hierarchically represented information about Factor meanings could be used for prediction. As factual patterns are associated with a claim's strengths or weaknesses, Factors are assumed to be relevant to the likely outcome of problems. If additional knowledge about the reasons why Factors matter, including use in downplaying and emphasizing distinctions, could be shown to improve the predictions, it would support the conclusion that CATO's model of argumentation was reasonable.

The argument-based prediction methods assume that if the most relevant cases for making an argument all favor the same side, that side has a very strong argument, while the opponent cannot make a stronger, or at least equally strong, argument. This strategy was implemented in the following decision rule: retrieve the relevant cases according to some relevance criterion. If there are relevant cases, and all had the same outcome, predict that side will win; otherwise abstain. Aleven defined seven criteria for selecting the most relevant cases in terms of HYPO's and CATO's argumentation criteria and tested their predictive value.[49] Each relevance criterion employed an increasing amount of CATO's legal argument knowledge. The HYPO-BUC criterion based a prediction on the best untrumped cases. Another criterion, referred to as the CATO-NoSignDist criterion, refined HYPO-BUC; it based its predictions on the best untrumped cases among those cases that had no significant distinctions. A distinction was "significant" if and only if it could be emphasized but not downplayed, a criterion unique to CATO's legal argument model.

CATO's focus on significant distinctions do not make much difference in predicting the outcome of *K & G*. The database contains five best untrumped cases to cite for the plaintiff in *K & G*, and none for the defendant. The HYPO-BUC criterion bases its prediction for plaintiff on all five cases. The CATO-NoSignDist criterion bases its prediction for plaintiff on the same five cases. On some occasions, however, the CATO-NoSignDist criterion does filter out cases that are significantly distinguishable from the problem, leading to different predictions from those made by the HYPO-BUC criterion. Part III.A.1 reports some results illustrating the differences.

II. ISSUE-BASED PREDICTION

While prediction based on case comparisons is not tantamount to predicting outcomes of real-world legal disputes, it is a way of extracting from a database of cases information relevant to analyzing a problem and of communicating it intelligibly in the form of a prediction accompanied by explanations. In this sense, prediction can support legal deliberation about a case even if the prediction is subject to practical limitations. Even when the cases at the end of a decision tree branch are inconsistent, the induced "rule" acts as an index to conflicting cases in the database relevant to analyzing a problem. The example of CATO suggests that programs can do more than simply point to relevant cases; they can reason intelligently with inconsistent data, make predictions, and explain them in ways natural for attorneys.

This can be analogized to scientific hypothesis testing of inconsistent data. The IBP algorithm frames and tests predictive hypotheses concerning the outcomes of factual disputes against the "experimental data" comprising the decided cases in a computerized case database. The IBP algorithm, described in pseudo-code in Figure 4, frames hypotheses predicting which side is favored and tests them against cases in the database. Given a new dispute represented in terms of Factors, IBP identifies relevant legal issues based on its Domain Model, determines which party is favored for each issue, and combines the analyses into an overall prediction. For each issue, IBP formulates a hypothesis predicting which side is favored for the issue, tests the

49. Aleven, *supra* note 37, at 214.

hypothesis against the past cases, and, if the cases are inconsistent, attempts to explain away the exceptions to the hypothesis. The record of this hypothesis-testing process serves as a legally natural explanation of its predictions and a useful summary of how the cases in the database bear upon the legal analysis of the problem.

Input: Current fact situation (*cfs*)

A. Identify issues raised by *cfs* Factors
B. For each issue raised, determine the side favored for that issue:
    1. if all issue-related Factors favor the same side, then return that side,
    2. else retrieve issue-related cases in which all issue-related Factors apply
        a. if there are issue-related cases, then carry out Theory-Testing: form hypothesis that same side *s* will win that won majority of cases
            i. if all issue-related cases favor side *s*, then return side *s*,
            ii. else try to explain away exceptions with outcomes contrary to hypothesis
                (a) if all exceptions can be explained away, then return side *s* favored by hypothesis
                (b) else, return "abstain"
        b. if no issue-related cases are found, then call Broaden-Query
            i. if query can be broadened, then call Theory-Testing for each set of retrieved cases
            ii. else return "abstain"
C. Combine prediction for each issue

Output: Predicted outcome for *cfs* and explanation

Figure 4: IBP Algorithm

IBP's Domain Model for trade-secret law is used for identifying relevant issues in step A and for combining the predictions in step C. The Domain Model is an interpretation of the trade-secret domain, including the logical relationships among the issues as well as the factual categories that affect the outcomes of issues in the cases.[50] As shown in Figure 5 it breaks a trade-secret claim (Trade-Secret-Misappropriation) into two major elements, both of which must be satisfied: "Is the information a trade secret?" (Info-Trade-Secret) and

50. The logical relationships among the issues in IBP's Domain Model are drawn from two sources: (1) "'Trade secret' means information, . . . that: (i) derives independent economic value, . . . from not being generally known to, and not being readily ascertainable by proper means . . . and (ii) is the subject of efforts that are reasonable under the circumstances to maintain its secrecy." UNIFORM TRADE SECRETS ACT § 1(4) (1985); (2) "One . . . is liable [for trade secret misappropriation if] (a) he discovered the secret by improper means, or (b) his disclosure or use constitutes a breach of confidence . . . . RESTATEMENT (FIRST) OF TORTS § 757 (1939).

"Was the information misappropriated?" (Info-Misappropriated). It further subdivides each element into issues. Info-Trade-Secret comprises two conjoined issues: "Was the information of value?" (Info-Valuable) and "Did the plaintiff maintain secrecy of the information?" (Maintain-Secrecy). Info-Misappropriated includes three issues "Was there a confidential relationship between plaintiff and defendant?" (Confidential-Relationship), "Did the defendant use the plaintiff's information?" (Info-Used), and "Did the defendant employ improper means to access the plaintiff's information?" (Improper-Means). Although the Domain Model captures rules relating the issues, it does not employ logical rules of conjunction and disjunction to relate Factors and issues; instead, it merely specifies which Factors are relevant to each issue.[51] Using the Domain Model, IBP can identify which issues the situation raises given a problem situation's Factors.

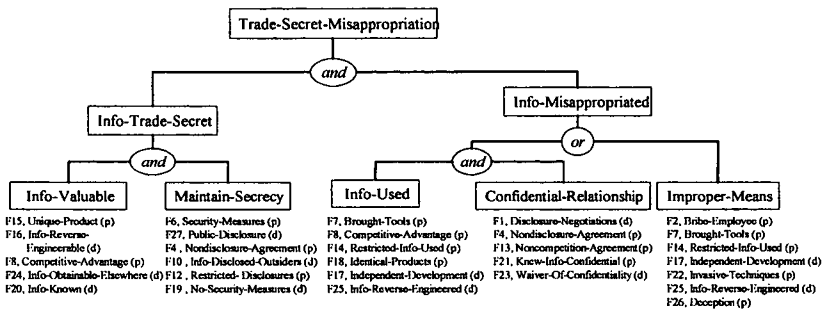

Figure 5: IBP's Domain Model

IBP's hypothesis formation and testing begins once the issues have been identified (Figure 4, Step B). Given an issue, if all of the problem's Factors related to that issue favor the same side, IBP concludes that side will win the issue. If the issue-related Factors conflict, providing evidence for both sides, IBP relies on the precedents in its case base to resolve this conflict by calling its Theory-Testing function. It retrieves the cases in the database that share all of the issue-related Factors with the problem (i.e., the issue-related cases). If

---

51. In relating subissues to relevant Factors, and through them to cases in the database indexed by these Factors, IBP's Domain Model is similar to that of CABARET. *See* Edwina Rissland & David Skalak, *CABARET: Statutory Interpretation in a Hybrid Architecture*, 34 INT. J. OF MAN-MACHINE STUDIES 839 (1991). Unlike CABARET, however, IBP employs this representation for the purpose of predicting outcomes.

all these cases were won by the same side, IBP assumes that this side is favored for the issue in the problem. Sometimes, cases favoring both sides may be returned, with a majority favoring one side or the other. This suggests that the hypothesis is the majority side is favored and that the other cases are exceptions that can be explained away to save the hypothesis. IBP implements this reasoning by determining which side won the majority of the issue-related cases. That side is hypothesized to be the favored one for the issue in the current fact situation. IBP then examines each exception in the minority set and attempts to explain it away.

IBP explains away such a counterexample by determining whether it can be distinguished in a special way from the problem and from the cases in the majority set (the positive instances of the hypothesis). In explaining away exceptions, IBP looks for alternative explanations for their outcome to save its hypothesis. It tries to find strong Factors, unrelated to the issue IBP is working on, to which the exception's outcome can be attributed. Specifically, it determines whether the exception has certain Factors called Knock-Out (KO-Factors) that are not related to the issue and that are neither present in the problem nor in the positive instances of the hypothesis. A Factor is defined as a Knock-Out if the probability that a side wins when the Factor applies is at least 80% greater than the baseline probability of the side's winning.[52]

IBP employs KO-Factors only indirectly by attempting to explain away past cases that appear to be exceptions to a predictive hypothesis. If IBP finds a KO-Factor to which the outcome of an exception can be attributed, it does not treat the exception as a reason to abandon its hypothesis. If IBP cannot distinguish all exceptions to its hypothesis in this way, it abstains for that issue.

IBP also defines a category of Weak Factors for which the probability of the favored side's winning, given that one knows the Factor applies is less than 20% over the baseline probability of the side's winning. While Weak Factors provide evidence on disputed issues, the courts in the cases in our database appear not to have treated them as sufficient on their own to raise an issue. Weak Factors are more relevant in the context of other Factors. If the only issue-related Factor in a case is weak, IBP regards the issue as not having been raised and does not pose a hypothesis to test concerning that issue. For instance, if a case has Factor F10 Info-Disclosed-Outsiders (D) and no other

52. The former probability is calculated as the number of cases where the Factor applies and the side won divided by the number of cases in the collection where the Factor applies. The baseline probability is calculated as the number of cases where the side won divided by the number of cases in the collection. There is also a semantic requirement; a KO-Factor must represent behavior that is paradigmatically proscribed or encouraged under trade secret law. IBP's list of KO-Factors includes: F8 Competitive-Advantage (P) (i.e., defendant saved development time and money by using plaintiff's information), F17 Info-Independently-Generated (D), F19 No-Security-Measures (D), F20 Info-Known-to-Competitors (D), F26 Deception (P), F27 Disclosure-In-Public-Forum (D).

Factors related to the issue of Security-Measures, IBP ignores that issue for purposes of formulating predictive hypotheses.[53]

If no cases in the database share all of the issue-related Factors in the problem, IBP tries to broaden the query. It relaxes its constraints (Figure 4, Step B.2.b) to see if a more general but still pertinent hypothesis can be tested. If two or more issue-related Factors favor a side, IBP removes each one in turn from the query. Using the new queries, it tries to find cases that share the less inclusive set of Factors, carrying out the same hypothesis testing as above. If a case retrieved by one of the new queries favors that side, one may conclude the problem is even stronger for that side given the dropped Factors. If the query cannot be broadened, or if it can be broadened for both sides, IBP abstains on the issue.

The following example shows IBP's approach to prediction and how it differs from CATO's. IBP's output for the *K & G* case of Figure 2 is shown in Figure 6. Using the Domain Model, IBP identifies four relevant issues: Maintain-Secrecy, Confidential-Relationship, Info-Valuable, and Info-Used. IBP predicts plaintiff is favored for all issues and, thus, that plaintiff will win the case. For Maintain-Secrecy and Confidential-Relationship, IBP finds no conflicts among the issue-related Factors and concludes that plaintiff is favored.

For the issue Info-Valuable, however, IBP finds conflicting Factors and carries out hypothesis testing. It retrieves the cases that share these issue-related Factors: seven cases won by plaintiff, one by defendant. Since more cases favor plaintiff, IBP hypothesizes that plaintiff is favored on that issue. The hypothesis is that where F16 Info-Reverse-Engineerable (D) and F15 Unique-Product (P) apply in a trade secret misappropriation case, plaintiff is favored for that issue Info-Valuable.

53. The other Weak Factors include F1 Disclosure-in-Negotiations (D) and F16 Info-Reverse-Engineerable (D).

```
Prediction for KG, which was won by PLAINTIFF
   Factors favoring plaintiff: (F21 F18 F15 F14 F6)
   Factors favoring defendant: (F25 F16)
   Issue raised in this case is MAINTAIN-SECRECY
        Relevant factors in case: F6(P)
        The issue-related factors all favor the outcome PLAINTIFF.
   Issue raised in this case is CONFIDENTIAL-RELATIONSHIP
        Relevant factors in case: F21(P)
        The issue-related factors all favor the outcome PLAINTIFF.
   Issue raised in this case is INFO-VALUABLE
           Relevant factors in case: F16(D) F15(P)
        Theory testing has no clear outcome, try to explain away exceptions.
        Cases won by plaintiff:
           AMERICAN-CAN (F4 F6 F15 F16 F18)
           HENRY-HOPE (F4 F6 F15 F16)
           ILG-INDUSTRIES (F7 F10 F12 F15 F16 F21)
           KAMIN (F1 F10 F16 F18 F15)
           KUBIK (F7 F15 F16 F18 F21)
           MASON (F15 F16 F6 F21 F1)
           TELEVATION (F6 F10 F12 F15 F16 F18 F21)
        Cases won by defendant:
           NATIONAL-REJECTORS (F7 F10 F15 F16 F18 F19 F27)
        Trying to explain away the exceptions favoring DEFENDANT
           NATIONAL-REJECTORS can be explained away because of unshared
           ko-factor (s) (F27 F19).
   Issue raised in this case is INFO-USED
           Relevant factors in case: F25(D) F18(P) F14(P)
        Theory testing did not retrieve any cases, broadening the query.
        For INFO-USED, the query can be broadened for PLAINTIFF.
        Each pro-P Factor (F14 F18) is dropped for new theory testing.
           Theory testing with Factors (F14 F25) gets the following cases:
              TECHNICON PLAINTIFF F6 F10 F12 F14 F16 F21 F25)
           In this broadened query, PLAINTIFF is favored.
              Theory testing with Factors (F18 F25) gets the following cases:
           MINERAL-DEPOSITS PLAINTIFF F1 F16 F18 F25)
           In this broadened query, PLAINTIFF is favored.
        By a-fortiori argument, the PLAINTIFF is favored for INFO-USED.
        Therefore, PLAINTIFF is favored.
   Outcome of the issue-based analysis:
     For issue CONFIDENTIAL-RELATIONSHIP, PLAINTIFF is favored.
     For issue INFO-VALUABLE, PLAINTIFF is favored.
     For issue INFO-USED, PLAINTIFF is favored.
     For issue MAINTAIN-SECRECY, PLAINTIFF is favored.
   => Predicted outcome for KG is PLAINTIFF, which is correct.
```

Figure 6: IBP's Output for the *K & G* Case

Seven cases are consistent with the hypothesis, but *National Rejectors* is an exception.[54] IBP succeeds at explaining away this exception; it finds two Knock-Out Factors in *National Rejectors* that do not apply to the *K & G* problem: F19 No-Security-Measures (D) and F27 Disclosure-In-Public-Forum (D). In *National Rejectors*, F19 captures the fact that the plaintiff had not taken security measures. This Factor does not apply to the *K & G* case, or to any of the other retrieved cases. It is also not related to the issue Info-Valuable, which focuses on the value of the information to competitors. It is reasonable to assume that in *National Rejectors*, the defendant won because of a lack of security measures and not because of facts related to the issue Info-Valuable. Moreover, *National Rejectors* has Factor F27, which allows a similar inference. IBP thus successfully explains away *National Rejectors* and concludes that plaintiff is favored for the issue Info-Valuable.

For the issue Info-Used, IBP again finds conflicting Factors. Two of these Factors favor plaintiff, and one favors defendant, but IBP cannot retrieve any cases that have all three issue-related Factors. It succeeds in broadening the query for plaintiff by dropping each of the pro-plaintiff Factors in turn. Each time the query is broadened, the argument becomes weaker for the plaintiff, but IBP still retrieves only pro-plaintiff cases. Consequently, IBP concludes that for this issue plaintiff is favored *a fortiori*, which presents an even stronger pro-plaintiff scenario.

IBP combines its analysis of the issues using the Domain Model. Since plaintiff is favored on all four issues, it predicts plaintiff will win the *K & G* case. IBP's explanation of its prediction is a summary of its process of identifying issues, forming hypotheses and testing them for each issue, and combining issue analyses into an overall prediction. It represents an interpretation of the legal significance of the problem's facts given its Domain Model and the cases in the database.

IBP's explanation is not intended to embody the same analysis as the court that decided a case. Among other things, the court uses a different knowledge base of cases, and courts' case analyses and rationales are not represented in IBP's case representation or in the Case Database. Nevertheless, it is instructive to compare IBP's analysis of a case like *K & G* with the court's and with CATO's. In its opinion in the *K & G* case, the Texas Supreme Court wrote:

> The Court of Civil Appeals was of the opinion that [plaintiffs'] magnetic fishing tool was not subject to protection as a 'trade secret,' primarily because . . . witnesses . . . testified that they could reconstruct the [plaintiff's] tool without first disassembling it. It may be that they could, but this hardly reaches the controlling point or issue in the case. According to the jury's findings, which were not attacked, G & G did not learn how to make the . . . tool . . . by observing it in an assembled or unbroken condition but learned of its internal proportions, qualities and mechanisms by taking it apart despite

54. Nat'l Rejectors, Inc. v. Trieman, 409 S.W.2d 1 (Mo. 1966).

> an agreement that it would not do so. We have here the violation of a confidence and the breach of a contract. Under such circumstances, the injured party is entitled to full relief, both legal and equitable . . . .[55]

Both IBP and the court address the issues of whether there was a breach of a confidential relationship, and whether information is of sufficient value to be a trade secret where it is subject to being reverse engineered. They address the issues somewhat differently, however. The court treats them more as a conflict of issues: where there is a violation of a confidence even though the information might have been reverse engineered. In this respect, its argument is like CATO's argument in Figure 3. By contrast, IBP treats the issues separately, and predicts plaintiff would win both issues. Its analysis regarding Info-Valuable finds cases where the product's uniqueness (Factor F15 Unique-Product (P)) overcomes the effect of the fact that the information is reverse engineerable. The interpretations are different, but not wildly so, and all seem reasonable.

In this example, IBP's prediction for *K & G* agrees with CATO's predictions using both the HYPO-BUC and CATO criteria of Part I.D.2, and all of these predictions happen to be correct. To get a better sense of how these predictive methods compare, Part III analyzes them over a wider range of cases.

III. COMPARATIVE EVALUATION OF IBP AND OTHER METHODS

Prediction can be seen as a way of extracting from a database of cases analytical information relevant for deliberating about a problem and of communicating that information intelligibly and succinctly in the form of the prediction and accompanying explanation. Testing predictive legal hypotheses against data and attempting to explain away counterexamples, as in scientific hypothesis testing and in IBP, is an effective way to implement prediction. By virtue of its explanations, it is potentially a more useful way to implement intelligent retrieval from legal databases than the plausible alternatives.

This hypothesis was tested by comparing various computerized techniques for making predictions from labeled examples in a database of cases. Each technique was run in the same experimental setup on the same collection of cases, with CATO's database of 184 trade secret misappropriation cases represented in terms of Factors. Plaintiffs won 108, and defendants won 76 of these cases. This database was assembled at two different times, both prior to the invention of IBP: 148 cases were collected for the original CATO program and 36 additional cases had been collected for a different purpose. For this experiment, the cases were converted from their Factor representation into

55. K & G Oil Tool & Service Co. v. G & G Fishing Tool Service, 314 S.W.2d 782, 787 (Tex. 1958).

ordered lists of binary features indicating which factors applied, together with the outcome, plaintiff or defendant, encoded as the class label.

The experiment involved three types of algorithms with complementary strengths. Although each algorithm was run on the same set of cases, they make different uses of that information, as discussed in Table 2.

Table 2: Prediction Algorithms Compared

Type	Algorithm	Description
1. Rule learning Algorithms	C4.5	Induces decision trees
	Ripper	Induces rules with a covering algorithm
	RL	Induces rules and combines available evidence
2. Case-Based algorithms	kNN/IB1	Classic k Nearest Neighbor
	HYPO-BUC	Decision rule based on best untrumped cases
	CATO-NoSignDist	Decision rule based on best untrumped cases without significant distinctions
	IBP	Issue-based hypotheses tested against cases
	IBP-Cases	Hypotheses tested against cases without issues derived from Domain Model
	IBP-Model	Issue-based analysis without hypotheses tested against cases
3. Statistical learning Algorithms	Logistic Regression	Statistical methods using logistic regression
	Naïve Bayes	Uses statistical weighting
Other	Baseline	Predict majority class (plaintiff)

A. The Prediction Algorithms Compared

The first type of algorithms used in this experiment include learned rules or classifiers that can be expressed as rules. C4.5 is a decision-tree learning algorithm[56] and a successor to the ID3 algorithm discussed above in Part I.C. Most notably, C4.5 can deal with impure leaf nodes and includes functions for pruning or simplifying the induced tree by removing unnecessary nodes and branches, which often leads to improved performance. For comparability, the decision trees were automatically converted to rules. Ripper, the second rule learning program,[57] was designed to generate efficient rules that are more

56. *See* QUINLAN, *supra* note 31.

57. William Cohen, *Text Categorization and Relational Learning*, PROCEEDINGS OF THE TWELFTH INTERNATIONAL CONFERENCE ON MACHINE LEARNING 124–32 (1995).

reminiscent of "rules of thumb." It incrementally adds rules that cover the training data in a patchwork-like manner and also generalizes from impure subsets. While C4.5 tends to overfit the data by inducing overly specific trees, Ripper's rules tend to be very simple and intuitive.[58] The third algorithm is the data-mining program RL.[59] It follows a different strategy and works in two steps.[60] Some of the rules RL used were similar to those learned by Ripper. The majority of the rules, however, captured which side is favored by the Factors that are highly correlated with the case outcome.[61] Finally, it learned some rules that seem intuitively sensible from a legal viewpoint.[62]

---

58. When run on the full data set, Ripper learned five rules. *Id.* There are four prodefendant rules about when the plaintiff certainly will lose and a proplaintiff default rule. The numbers in parentheses indicate, respectively, the number of times the rule fired and the number of those firings which led to erroneous predictions for the training set:

1. If F20 Info-Known-to-Competitors (D) applies then predict DEFENDANT (31/1).
2. If F6 Security-Measures (P) does NOT apply and F19 No-Security-Measures (D) applies then predict DEFENDANT (18/2).
3. If F27 Disclosure-In-Public-Forum (D) applies and F8 Competitive-Advantage (P) does NOT apply then predict DEFENDANT (7/1).
4. If F24 Info-Obtainable-Elsewhere (D) applies, then predict DEFENDANT (6/2).
5. Otherwise, predict PLAINTIFF (101/15).

Ripper's pro-plaintiff default rule is reflected in the CATO database, in which there are more cases won by plaintiff than by defendant. The baseline algorithm used for comparing the prediction algorithms relies on the same information and predicts the majority class no matter what the facts of the new problem. In the CATO data set, this means predicting that plaintiff wins.

59. Foster Provost et al., *Rule-Space Search for Knowledge-Based Discovery*, http://pages.stern.nyu.edu/~fprovost/Papers/rule-search.pdf (last visited May 11, 2006).

60. First, it derives rules from the labeled examples in the training set. Given a training example, RL identifies candidate rules that may explain the outcome of the case. It collects the rules that are most predictive for the entire training set and keeps track of their success rates. In contrast to Ripper, RL uses a covering algorithm with replacements. If Ripper finds a rule that correctly predicts a set of cases, it considers these cases covered and removes them from the training set. On the other hand, RL returns these cases in the training set. Ripper's strategy works well when there is only one explanation for the outcome of a case. In more complex domains, however, where more than one aspect of a case may contribute to its outcome, RL may be more effective. Second, when a new and previously unseen case is classified, RL considers all applicable rules, and if necessary, assigns weights to them. RL has a number of techniques for identifying candidate rules and assigning weights to conflicting evidence in the classification phase.

61. For instance, "If F15 Unique-Product (P) applies then outcome = plaintiff" and "If F6 Security-Measures (P) applies, then outcome = plaintiff." It also learned some complementary rules, such as "If F6, Security-Measures (P) does not apply, then outcome = defendant."

62. "If F4 Agreed-Not-to-Disclose (P) does not apply and F21 Knew-Info-Confidential (P) does not apply then outcome = defendant." In other words, the rule says, where defendant did not enter into a nondisclosure agreement, and where the defendant was not on notice that the information was secret, the defendant will win.

The second type of algorithms in this experiment comprised case-based learning algorithms. There were two different kinds: an algorithm (IB1) that computes similarity using a numeric function,[63] and several algorithms that were developed from HYPO's *a fortiori* model.[64]

The third type of algorithms can be characterized as statistical learning methods. These algorithms use advanced statistical models to compute a prediction. In this experiment, we included logistic regression, the statistical method described in Part I.B. We employed a data mining package to perform logistic regression[65] and ran it as is without performing any feature selection. As mentioned above, the performance of this algorithm can be improved through feature selection, which may require considerable expertise to get the maximum benefit. We used a standard automated method, and, given our focus on fully-automated processes and our preference for methods that can generate legally intuitive explanations (which logistic regression cannot), we did not further pursue this trial-and-error process of selecting features in search of more accurate predictions.[66]

We also included Naïve Bayes, another statistical ML algorithm that classifies a new instance by assigning the most probable target value given the attributes that describe the instance.[67] Specifically, it calculates the probability that a side wins given the case's Factors, P(*outcome*|*factors*), for each of "plaintiff wins" and "defendant wins," and then picks the more likely winner. However, this target function cannot be calculated directly from the available data. Therefore, the algorithm relies on small pieces of evidence, easily observable probabilities, and combines them using Bayes Rule to calculate the above values. The Naïve Bayes algorithm makes a simplifying assumption of independence among the descriptors given the target value.[68]

---

63. IB1 is a case-based learning algorithm that implements a nearest neighbor approach similar to that described in Part I.A. The algorithm uses a similarity function to identify the most relevant cases for a problem, its nearest neighbors, based on which a prediction is made. Similarity in IB1 is defined as the Euclidean (or straight-line) distance between the examples, a different distance metric than that used in MacKaay's *k*NN approach discussed in Part I.A.

64. The HYPO-related case-based prediction algorithms do not compute numerical similarity; they rely on symbolic reasoning and arguments comparing cases. The HYPO-BUC and CATO-NoSignDist algorithms have been described above in Part I.D.2. Part II describes the IBP prediction algorithm. Also included are two variations of the IBP algorithm, IBP-Cases and IBP-Model, discussed below.

65. *See generally*, IAN H. WITTEN & FRANK EIBE, DATA MINING: PRACTICAL MACHINE LEARNING TOOLS AND TECHNIQUES (2d ed. 2005).

66. Using a different statistical package, another student used a best fit logistic regression to identify eight Factors that most strongly correlated with case outcomes in the original CATO database of 148 cases and then applied it to predict the outcomes of the later 36 cases, obtaining an accuracy of 86%. C. Allen Black, Statistical Predictions of Case Outcomes Using the CATO Dataset (2003) (unpublished student paper, University of Pittsburgh School of Law) (on file with author).

67. TOM MITCHELL, MACHINE LEARNING 177 (1997).

68. Two events are independent if the occurrence of one event does not have an impact on the other's probability. In this context, independence means that the probability of observing a particular list of Factors is equal to the *product* of the probabilities that each individual Factor is

1. *Experimental Design*

The 184 cases in the CATO/IBP database on which each algorithm was run were drawn from a variety of state and federal courts. The opinions vary from trial court through the highest appellate courts. For each case, the database includes the opinion on substantive issues of the highest level court to consider the case. The cases span a period of decades. Most date from the 1970s through 1990s; some are earlier. The dates of the cases, the jurisdiction, and the court level were not taken into account by any prediction method in these experiments.

The experiments were run in a leave-one-out cross-validation. This procedure is widely used in ML experiments and guarantees that the training and test sets never overlap. For each algorithm the test case was removed from the database, a procedure that was repeated 184 times (once for each case). Its outcome was hidden, and only its Factors were visible. The classifiers were trained on a training set of the remaining 183 cases and then used to predict the outcome of the test case. The predicted outcome was compared to the previously hidden real outcome of the test case to determine if the prediction was correct, a mistake, or an abstention. The result was recorded. The test case was then returned to the database.

A number of the algorithms offer a variety of parameters for experimentation; we ran all of the algorithms with their default parameter settings except for RL, for which we selected settings likely to work best.

2. *Result*

Each algorithm's accuracy is reported in Table 3; the results are visualized in Figure 7. Accuracy is defined as the number of correct predictions divided by the number of cases included in the experiment, which comprises the sum of the correct predictions, incorrect predictions, and abstentions. IBP predicted the outcome of 169 cases correctly, made 14 errors, and abstained once for an accuracy of 91.8%. RL was the runner-up to IBP with an accuracy of 88%, followed by Naïve Bayes with 86.4% accuracy. Of the other rule learning programs, C4.5 did better, coming in fourth with an accuracy of 85%; Ripper achieved 83%.

---

observed. Using this assumption, probabilities for individual Factors were computed simply by counting cases in the database. Although this assumption often is not satisfied (and therefore is called "naïve"), in many applications it does not have a large effect on the result. We did not try another computerized prediction algorithm that has been used for legal applications, a connectionist or neural network. For instance, the Split-Up program predicted marital property divisions in divorce cases. Explaining predictions is difficult for a connectionist system because the information on which the prediction is based is distributed among weights. Interestingly, Split-Up hierarchically organizes multiple specialized networks by issues from which explanations can be generated. *See* John Zeleznikow et al., *Project Report: Split-Up – A Legal Expert System which Determines Property Division Upon Divorce*, 3 ARTIFICIAL INTELLIGENCE AND LAW 267 (1995).

Table 3: Correct Predictions, Abstentions, Errors, Accuracy, and Significance for Each Prediction Algorithm

Algorithm	Correct	Abstain	Errors	Accuracy	Significance Probability[69]
IBP	169	1	14	0.918	---
RL	162	0	22	0.880	0.08
Naïve Bayes	159	0	25	0.864	0.03
IBP-Cases	144	30	10	0.783	0.00
CATO-NoSignDist	143	22	19	0.777	0.00
C4.5	156	0	28	0.848	0.01
Logistic Regression	154	0	30	0.837	n/a
Ripper	152	0	32	0.826	0.00
Nearest Neighbor	151	0	33	0.821	0.00
HYPO-BUC	125	50	9	0.679	0.00
IBP-Model	132	38	14	0.717	0.00
Baseline	106	0	78	0.576	0.00

69. The significance probability (p-value) indicates whether the differences between IBP's performance and that of each other algorithm in the experiment are statistically significant. It is the probability of a difference as large or larger than the observed difference assuming that the true difference is zero. Generally, $p < 0.05$ is considered indicative of a true difference. We used McNemar's test. *See* Thomas Dietrich, *Statistical Tests for Comparing Supervised Classification Learning Algorithms*, Oregon State University Technical Report 13 (1996). Essentially, one goes through every one of the incorrectly predicted cases and scores which algorithm does better. Since some of the algorithms make abstentions (e.g., IBP and the case-based argumentation algorithms), we adapted the test by counting a correct prediction as "better" than an abstention and an abstention as "better" than an error. The difference between the accuracy of IBP and RL, the runner-up, was not significant ($p = 0.08$). The other differences were significant. The software we used did not allow for a comparison between IBP and Logistic Regression.

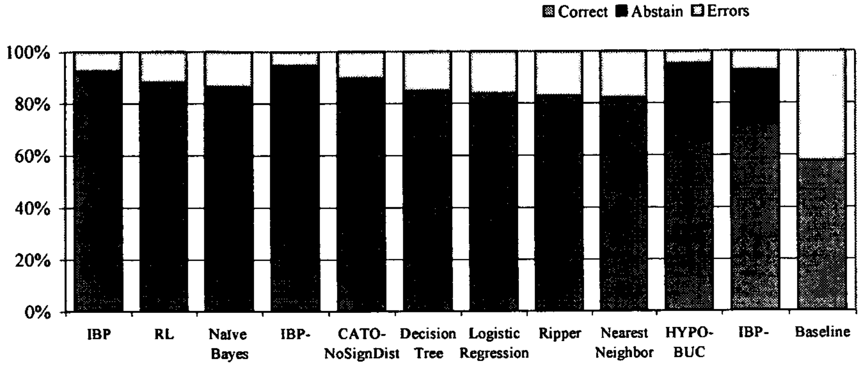

Figure 7: Comparison of Prediction Algorithm Accuracy

B. Discussion

IBP's hypothesis-testing approach has two advantages over the alternatives. Not only does IBP most accurately predict how cases will be decided based on past cases in its database, but it also generates an explanation of its predictions in terms an attorney can assess. In analyzing the results, it is helpful first to consider the sources of IBP's successful predictions and the reasons for its failures.

1. *Why IBP Works*

Since IBP was developed with CATO's Case Database, we first ruled out that IBP had been optimized with respect to these 148 cases. As noted, 36 cases had been collected for a different purpose and were not added to the collection until after IBP was completed; IBP's predictions for these 36 cases were no less accurate than for the initial 148.

To investigate the sources of IBP's successful predictions and related questions, we created two *ablated* versions of IBP called IBP-Cases and IBP-Model. An ablated version of a program is one in which particular components or knowledge sources have been "turned off." The ability to ablate a computational model's features or knowledge allows one to investigate systematically the sources of its power.

The first ablated version, IBP-Cases, made predictions using only cases. It had no knowledge about issues; that is, it had no access to the information about issues in IBP's Domain Model shown in Figure 5. Instead, IBP-Cases, in effect, implemented only the ultimate "issue," which is whether plaintiff should win the claim of trade secret misappropriation. IBP-Cases used the same database of cases as IBP. Since it dealt with only the ultimate "issue," IBP-Cases would first consider all of the problem's Factors together. If they conflict in the Theory-Testing function, it would seek cases that share all of the Factors and hypothesize that the side with the majority of such cases wins. The explaining away of counter-examples had to be modified because there was only one issue; IBP did not require that the KO-Factors be related to a different issue. If IBP-Cases failed to find any cases with all of the problem's Factors, it would broaden the query by systematically dropping Factors until it found cases with which to formulate and test a weakened prediction hypothesis. A minor technical modification was necessary to enable it to broaden queries.

By contrast, IBP-Model made predictions using only the issues and Factors in its Domain Model. It could not access cases. For each issue in a new problem, IBP-Model simply tested whether all of the issue-related Factors favored the same side. If so, IBP-Model inferred that side was favored for the issue; otherwise, it abstained. Since IBP-Model did not reason with precedents, it could not carry out Theory-Testing, explain away exceptions, or broaden queries.

IBP performed significantly better (91.8% accurate) than IBP-Cases (78.3% accurate). IBP abstained less frequently than IBP-Cases at the cost of four additional errors. IBP predicted cases correctly considerably more often than IBP-Cases (169 vs. 144), offsetting the additional errors.

Since IBP and IBP-Cases both employ the same cases and case representation, we conclude that knowledge of issues as represented in IBP's Domain Model accounts for IBP's more accurate predictions.[70] The intermediate legal issues contribute to IBP's predictive accuracy. As the *K & G* example in Figure 6 illustrates, IBP's knowledge about legal issues focuses Theory-Testing on Factors related to an issue rather than on all of the Factors in the problem. IBP-Cases' queries are more frequently unproductive because they fail to focus adequately on particular issues and on the conflicting Factors relevant to those issues. For two of the issues in *K & G*, IBP's issue-based analysis retrieves and analyzes cases before predicting an outcome.

Intuitively, it makes sense that including legal issues improves predictive accuracy. Our Factor representation of cases does not include the judges' reasoning. It reflects neither judges' opinions about which issues were most crucial nor their rationales in resolving any conflicts. Judges who decide trade secret misappropriation cases, however, are aware of and refer to the legal issues identified in the Restatement and Uniform Trade Secrets Act provisions, the same issues reflected in IBP's Domain Model. In formulating hypotheses about which party will win, IBP uses as its conceptual framework the "right" background knowledge about legal issues. When IBP uses its Domain Model to construct rationales relating Factors to issues, and ultimately, to outcomes, it employs issues that judges also employ. Its rationales therefore are likely to be legally reasonable. Without this conceptual framework, the hypotheses are not

70. Kevin Ashley & Stefanie Brüninghaus, *A Predictive Role for Intermediate Legal Concepts*, PROCEEDINGS 16TH ANNUAL CONFERENCE ON LEGAL KNOWLEDGE AND INFORMATION SYSTEMS 153–62 (2003).

as successful for prediction, are too conceptually unfocused, and fail to correspond to the Factors' significance and meaning. For legal and technical reasons, it is very difficult computationally to represent judges' rationales in legal opinions, as in the GREBE program.[71] IBP's ability to generate accurate predictions and reasonable explanations even without representing judicial rationales is therefore significant.

While the Domain Model makes IBP's predictions more accurate, it alone is not sufficient. IBP-Model took only the legal issues and corresponding Factors into account; it could not use cases to make predictions. As shown in Table 3 and Figure 7, it did not perform as well as IBP, achieving an accuracy of only 71.7% and abstaining often. In all 38 cases where IBP-Model abstained, the issue-related Factors favored both sides. Without access to cases, however, IBP-Model could not resolve these conflicts. Significantly, for all 38 cases where IBP-Model abstains, IBP's predictions were correct. Framing predictive hypotheses around issues and testing them against past cases result in greater accuracy.

2. *Analysis of IBP's Errors*

We next analyzed IBP's errors and found that a number of cases in the collection are very hard or even impossible to predict correctly. Based on the expected error distribution, we identified those cases that most or all of the prediction algorithms get wrong.[72] More than half of IBP's errors are on these anomalous cases. There are four main reasons why a case could be anomalous: (1) a case's Factor representation omitted a feature that the court deemed important; (2) although the Factor representation captured relevant facts, it failed to capture important details; (3) there were interpretive errors in manually assigning Factors by case enterers; or (4) the court's resolution of conflicting Factors was unique across the database of cases.[73]

First, in some cases, the Factor representation of the case omitted a feature that the court deemed important. In *Burten v. Milton Bradley Co.*,[74] the plaintiff submitted a proposal for a new board game to the defendant toy manufacturer and signed a form in which it acknowledged that there was no confidential relationship between the parties.[75] After ostensibly rejecting the idea, the defendant came out a year later with a very similar game.[76] The court held for the plaintiff even though he had signed a waiver of confidentiality, a fact represented in the CATO version of the case with Factor F23, Waiver-of-

71. L. KARL BRANTING, REASONING WITH RULES AND PRECEDENTS – A COMPUTATIONAL MODEL OF LEGAL ANALYSIS 111–34 (1999).

72. In the ML literature, such cases are often called noisy and removed from the collection to increase accuracy. For legal reasoning, however, these cases are binding precedents that cannot be deleted.

73. Brüninghaus & Ashley, *supra* note 38, at 240–41.

74. 763 F.2d 461 (1st Cir. 1985).

75. *Id.* at 462.

76. *Id.*

Confidentiality (D).[77] The court decided that the signed agreement was ambiguous, and that the jury correctly determined that it did not apply, a consideration for which CATO has no corresponding Factor.[78] As a result, all algorithms in the experiment predicted *Burten* incorrectly.[79]

Second, in some cases, the Factor representation failed to capture important details. For instance, Factor F16 Info-Reverse-Engineerable (D) applies in CATO's representation for the prodefendant case of *Speciner v. Reynolds Metals Company*,[80] but IBP predicts that plaintiff will win based on other factors. Factor F16 applies when the plaintiff's information can be learned by reverse engineering. The court gave more significance to the fact that the product could be reverse engineered *with only minimal effort*, holding that marketing the product effectively disclosed the information.[81] All algorithms predicted *Speciner* incorrectly. A dimensional representation of Factors[82] could account for how little time or effort it takes to reverse engineer the product, emphasizing a Factor's significance in a particular case.

Third, some errors result from faulty interpretive decisions by case enterers in manually assigning Factors to a case. For instance, to represent the measures that plaintiff took in maintaining the security of its information, CATO offers two choices: Factor F6 Security-Measures (P), which represents that plaintiff took such measures, and Factor F19 No-Security-Measures (D), which represents that plaintiff did not. In *Junkunc v. S.J. Advanced Technology and Manufacturing Corp.*,[83] the plaintiff took several isolated measures to maintain secrecy of its information. Since the court explicitly listed these measures, the case enterer applied Factor F6.[84] IBP predicted plaintiff would win, but the court held for defendant, apparently placing more significance on the several security holes than on the existing but ineffective security measures.[85] On the other hand, CATO's representation of *Allen Manufacturing Company v. Loika*,[86] employs F19 because the plaintiff had taken only minimal measures to protect the information and had left many gaping security holes. The court, however, apparently decided that even minimal measures were sufficient.[87] Both representational choices contributed to IBP's erroneous predictions for each case.

Fourth, in some cases, the court's resolution of conflicting Factors was unique across the database of cases, indicating that more cases of that type are

77. *Id.* at 466.
78. *Id.* at 467.
79. *Id.* at 466.
80. 177 F.Supp. 291 (S.D.N.Y. 1959), *aff'd*, 279 F.2d 337 (2d Cir. 1960).
81. *Id.* at 296.
82. *See* ASHLEY, *supra* note 19, at 107.
83. 498 N.E.2d 1179 (Ill. App. Ct. 1986).
84. *Id.* at 1183.
85. No algorithm included in the evaluation predicted *Junkunc* correctly. Using HYPO's dimensional approach, one could represent the set of measures plaintiff took, including "none."
86. 144 A.2d 306 (Conn. 1958)
87. *Id.* at 310.

needed or that the court in a particular case regarded a certain fact as extraordinarily significant. For instance, IBP and the other prediction algorithms all mistakenly predicted that plaintiff would win in *Wexler v. Greenberg*.[88] The court held that the defendant, a former employee, was justified in disclosing plaintiff's process because, while working for the plaintiff, the defendant was the sole developer of the information.[89] The court's protecting an employee from a postemployment restraint on competition is not unusual in trade secret law, but this case happened to be the only such example in the CATO database, contributing to IBP's erroneous prediction. In two other cases where IBP made errors, the CATO database contained similar cases, but the courts' resolutions of the two cases were unique. In *Franke v. Wiltschek*,[90] the court held for plaintiff despite weaknesses on the issue of whether the information was really a trade secret, appearing to regard the defendant's breach of a confidential relationship as of paramount importance.[91] Similarly, in *Goldberg v. Medtronic*,[92] the plaintiff won even though the information had been generally known and had been disclosed to third parties, apparently because the defendant used information it had obtained in breach of a confidential relationship with the plaintiff.[93]

Sometimes, the court itself may be in error, which may account for the uniqueness of the resolution in CATO's database. A counterexample that is an anomalous case because of a court's error could cause the program to abstain when it need not do so. Of course, the question arises how one can tell if the court is in error. Uniqueness of a decision among a large number of indistinguishable cases is one criterion. For instance, the unique case may reflect a social change in value preferences that require a different resolution. There are some "objective" criteria for determining that a court was mistaken. Subsequent court opinions may call the decision into question or even overrule it. The decision will be "yellow-flagged" or "red-flagged" in the full-text legal databases. For instance, the *Goldberg* decision was subsequently criticized and has been assigned a yellow-flag in the Westlaw database, indicating that the court in a later decision declined to follow the decision in *Goldberg*.

As noted, IBP attempts to explain away counterexamples by identifying KO-Factors. However, the presence of warning flags or other types of cautionary indicators could help to "explain away" a counterexample and avoid unnecessary abstentions. Another approach involves downplaying or emphasizing distinctions using CATO's Factor Hierarchy. Still another

88. 160 A.2d 430 (Penn. 1960).
89. *Id.* at 437.
90. 209 F.2d 493 (2d Cir. 1953).
91. *Id.* at 494–95.
92. 686 F.2d 1219 (7th Cir. 1982).
93. *Id.* at 1226–28.

involves reasoning about the case in terms of preferences among the underlying normative values at stake in cases of this type.[94]

Probably, no computer model can achieve 100% accuracy for a large database of real cases realistically represented. The representations always leave something out, and even if not, real cases are sometimes remarkably balanced, the precedents conflict, or the precedents are wrongly decided.

3. *IBP v. the Next Four: RL, Naïve Bayes, C4.5, and Ripper*

In terms of accuracy, RL is a close second to IBP, and the difference between them is not significant. Its design seems well-suited to learning from cases like those in the CATO database that are: (1) used to resolve conflicting evidence; and (2) more complex in that more than one aspect of a case may contribute to its outcome. IBP uses issues from its Domain Model to represent these multiple case aspects. RL does not employ issues, but it employs other techniques. As noted, it identifies and retains alternative candidate rules that may explain a training case's outcome and keeps track of their success rates. In the classification stage, it applies all applicable rules, and, if necessary, assigns them weights to resolve conflicting evidence. RL's use of a covering algorithm with replacements also helps. RL does not remove training cases that have been covered by a learned rule as other algorithms do. Instead, these cases stay in the training set where they may lead to learning other rules focusing on different aspects of the case.

The third-ranking algorithm, Naïve Bayes, achieves high accuracy, confirming previous findings.[95] Interestingly, it also achieves this accuracy without using domain knowledge such as that represented in IBP's Domain Model. Theoretical and empirical studies across a variety of learning domains have shown that Naïve Bayes is useful where sample size is small and that it is an optimal learner for conjunctive and disjunctive concepts, even though these violate the independence assumption.[96] As Ripper's induced rules suggest, one

---

94. *See* Trevor Bench-Capon & Giovanni Sartor, *Theory Based Explanation of Case Law Domains*, PROCEEDINGS OF THE 8TH INTERNATIONAL CONFERENCE ON ARTIFICIAL INTELLIGENCE AND LAW 12–21 (2001); Trevor Bench-Capon & Giovanni Sartor, *A Model of Legal Reasoning with Cases Incorporating Theories and Values*, 150 ARTIFICIAL INTELLIGENCE 97, 97–143 (2003); Allison Chorley and Trevor Bench-Capon, *AGATHA: Automated Construction of Case Law Theories Through Heuristic Search*, PROCEEDINGS OF THE TENTH INTERNATIONAL CONFERENCE ON ARTIFICIAL INTELLIGENCE AND LAW 45–54 (2005). In the latest work modeling legal argumentation as theory construction, the AGATHA program generated predictions comparable to IBP's in accuracy but for a reduced case base. Although the program makes use of value preferences, it is not yet clear how such generated theories inform useful explanations or square with statutory texts, similar to the Uniform Trade Secret Act or the authoritative Restatement, provisions of which IBP's Domain Model interprets.

95. *See* Aleven, *supra* note 37.

96. *See* Pedro Domingos & Michael Pazzani, *Beyond Independence: Conditions for the Optimality of the Simple Bayesian Classifier*, PROCEEDINGS, THIRTEENTH INTERNATIONAL CONFERENCE ON MACHINE LEARNING 105–112 (1996); Pedro Domingos & Michael Pazzani, *On the Optimality of the Simple Bayesian Classifier under Zero-One Loss*, 29 MACHINE LEARNING 103, 103–30 (1997).

can treat the concepts "plaintiff wins" or "defendant wins" as conjunctive and disjunctive concepts involving KO-Factors. Indeed, Naïve Bayes' computations of Factor probabilities are similar to those used in the definition of KO-Factors. Upon inspecting the final weight vector Naïve Bayes generates, we found generally that IBP's KO-Factors had been assigned high weights, and its weak Factors had been assigned low weights. Thus, Naïve Bayes appears to be doing a good job of what Ripper and C4.5 are attempting.

Having analyzed the sources of IBP's accuracy, we may venture another explanation of why its predictions are more accurate than the next four best-performing programs: RL, Naïve Bayes, C.4.5, and Ripper. These four programs share an "eager" approach to learning as opposed to IBP's "lazy" approach. In the ML literature, the difference between lazy and eager learning is that a lazy learner may "defer the decision of how to generalize beyond the training data until each new query instance is encountered."[97] An eager method "generalizes beyond the training data before observing the new query."[98] Thus, "a lazy learner has the option of (implicitly) representing the target function" (i.e., the function, to be learned from the training instances, that assigns a classification such as "plaintiff wins" to a new case) "by a combination of many local approximations, whereas an eager learner must commit at training time to a single global approximation."[99] By delaying the representation of the target function until it can be composed of local approximations based on a particular problem's facts, the expectation is that the learned function will be better tailored to the problem's context.

RL, C4.5, and Ripper are eager in that they induce their rules before seeing the facts of a particular problem to which the rules will ultimately be applied. Naïve Bayes is eager in that the probabilities used are computed in advance of knowing which particular Factors are present in a problem.[100] IBP's generalizations are the issue-based hypotheses it generates for a particular problem and tests against the cases returned from the database. IBP is lazy in that it delays generalizing from the database until the facts of the problem are known.

The ML literature's distinction between eager and lazy learning is reflected in models of common law reasoning. Edward Levi's model of legal reasoning with case examples, for instance, involves lazy learning; the rule for deciding cases may change as the rule is applied to a specific problem.[101] Similarly, Schauer notes that common law and US constitutional law are

97. MITCHELL, *supra* note 67, at 244.
98. *Id.* at 245.
99. *Id.*
100. *Id.* at 182.
101. EDWARD LEVI, AN INTRODUCTION TO LEGAL REASONING 3, 4 (1949).

domains in which judges may change a rule as they apply it in the context of a specific set of facts.[102]

IBP's greater predictive accuracy over RL, Naïve Bayes, C4.5, and Ripper may be due to the increased context sensitivity its lazy learning approach affords. However, lazy learning has to be applied carefully. The experiments included several lazy learning approaches, HYPO-BUC, CATO-NoSignDist, and *k*NN (IB1), which did not outperform the eager learners.

Although an eager method, since its rules are learned before the classification phase, RL apparently is designed to achieve much of what lazy learning offers. Its strategy of cover and replacement of cases, its retention of alternative rules for a training case, and its delayed, classification-phase assignment of weights to alternative rules to resolve conflicting evidence, make it a "lazy" implementation of an eager rule-learning approach.[103] IBP's approach to lazy learning is unique in that its domain model supports context-sensitive reasoning. The Domain Model's issues focus hypothesis formulation and testing on conflicting Factors, which helps ensure that the lazily formed hypotheses are conceptually focused. We conjecture that this combination is the key to explaining IBP's greater accuracy; it appears to achieve an effective balance of lazy learning and providing rules similar to those induced by eager learners.

4. *Comparing Algorithms' Explanations*

Beside accuracy, the algorithms should be compared in terms of whether and how they explain their predictions. Intuitive legal explanations may draw analogies between a problem and past cases and give reasons why the similarities and differences justify treating the problem similarly or differently in terms of applicable legal rules and their underlying purposes.

Despite its relatively accurate predictions, Naïve Bayes cannot generate intuitive legal explanations. Its outputs and weights are of limited use for making real-world legal arguments. The information that allows it to generalize from past cases and the probabilities it computes do not lend themselves readily to fashioning a rule or a qualitative argument intelligible to attorneys. An argument like, "Your honor, my client should win because our AI system has calculated that in this case he has a probability of 0.67 of winning," does not relate to legal standards of argument.

The rule learning algorithms, RL, C4.5 and Ripper, do generate rules that explain their predictions. As we have seen, however, these rules do not necessarily correspond to the kinds of explanations that are acceptable for a lawyer. RL's rules come closest. They usually correspond to reasonable

---

102. Frederick Schauer, *Is the Common Law Law?*, 77 CAL. L. REV. 455, 464, 470 n.41 (1989) (reviewing MELVIN A. EISENBERG, THE NATURE OF THE COMMON LAW (1988); *see also* Kevin Ashley & Edwina Rissland, *Law, Learning and Representation*, 150 ARTIFICIAL INTELLIGENCE 17, 19 (2003).

103. RL is one answer to Tom Mitchell's query, "Can we create eager methods that use multiple local approximations to achieve the same effects as local methods?" MITCHELL, *supra* note 67, at 245.

intuitions about trade secret law. By contrast, Ripper's rules tend to summarize conditions when the minority class will apply—that is, when it is most likely that plaintiff will lose. Although this is a reasonable strategy for an ML program, it does not correspond to the kinds of legal rules attorneys normally invoke in justifying assertions. These learning algorithms are designed to relate features to outcomes, not to generate an explanatory model of a domain. As a result, their rules do not reflect knowledge of trade secret law issues nor reasons why Factors are relevant to the issues.

While a nearest-neighbor algorithm identifies examples relevant for making fairly accurate predictions, it cannot explain why the examples are relevant except in terms of its similarity measure. As discussed in Part I.A, this measure combines relevant similarities or differences in a manner that does not relate to familiar patterns of analogical legal argument. The case- and argument-based predictors (HYPO-BUC and CATO-NoSignDist) do a better job. They explain predictions in terms of arguments that draw analogies between a problem and the most similar cases in terms familiar to attorneys (see Figure 3). Enabling programs like these to justify assertions in terms of legal rules and their underlying purposes is a matter of on-going research.

IBP's output, shown in Figure 6, explains predictions in terms of formulating and testing hypotheses about issues. IBP's explanations deal in terms that are intuitively accessible by attorneys. For each issue, the program formulates a hypothesis in terms of possibly conflicting Factors, finds cases that support or contradict the hypothesis, attempts to explain away the counterexamples in terms of KO-Factors that account for their outcome, and differentiates them from the positive instances. As we have seen, a number of the KO-Factors corresponds to the kinds of rules the induction programs generate. There is a subtle but important difference, however. IBP uses KO-Factors, which are defined in terms of predictive strength and semantics, to explain away counterexamples to a predictive hypothesis. By contrast, the inductive learners, like Ripper, apply a default if they can not find highly predictive Factors (the KO-Factors). This approach is more alien to traditional legal reasoning.

There may be good reasons to combine IBP's approach to prediction and explanation with CATO's approach to argumentation. IBP achieves higher accuracy than the CATO algorithm (91.3% vs. 77.7%), abstains less frequently, and makes fewer errors. IBP breaks up cases into issues and focuses on conflicts of issue-related Factors. CATO implements a complementary strategy that considers a case in a more gestalt-like way. Its relevance criteria consolidate evidence from across issues. Although the result is a lower accuracy, sometimes CATO retrieves cases that can make normatively reasonable arguments for a position even though it is predicted to lose.[104] This is useful if one represents the weaker side and needs to make a reasonable argument in support of an apparently weaker position. In the future, we will

104. Ashley & Brüninghaus, *supra* note 70, at 160.

study if IBP's focus on issues can help CATO select the strongest arguments for the weaker side.[105]

IV. RAMIFICATIONS FOR THE FUTURE

From a practical viewpoint, if the goal of prediction research in law is to generate predictions one "can bank on," then there is a long way to go. As noted, relying on the predictions of any of the algorithms as configured above for purposes of determining the actual odds of winning or losing a lawsuit is subject to significant risks such as the incomplete and biased selection of cases. To evaluate these risks, one would need to undertake a kind of experiment we have not: field tests that assess the quality of predictions on a larger set of diverse, real-world problems.

From a different "practical" viewpoint, however, prediction algorithms may be seen as a means for achieving a "systematization of traditional legal research."[106] From the limited set of cases that happen to be in an electronic database, they automatically extract reasons helpful in making decisions about a problem's outcome. The experiments can be seen as comparing algorithmic methods for extracting such reasons. In this light, IBP fares well. It shows how an on-line legal case database can help attorneys in an automated way to test hypotheses about resolving problems against the data. Of course, this use of predictions is limited, too. IBP's predictions and explanations are limited by the variables represented and by the need to read the cases to ensure the user agrees with the interpretations. In addition, the explanations do not refer to statutory rules or underlying policies. All of the cases and problems, moreover, "fit the model." Hunter distinguishes among legal domains that depend on landmark, leading, or commonplace cases. The last are more appropriate for computerized prediction because they tend to reflect the law rather than reconstruct it.[107] CATO's trade secret cases tend toward the commonplace. They do not address other claims, the procedural setting, or such issues as preemption under federal law.[108]

---

105. Interestingly, the experiment reveals that the CATO prediction algorithm performs better than HYPO-BUC. CATO's additional knowledge about the significance of Factor distinctions enables it to select the more relevant Best Untrumped Cases on which to base its predictions. CATO achieves an accuracy of 77.7% compared to HYPO-BUC's 67.9%. While the CATO algorithm abstains less frequently, it makes nearly twice as many errors as HYPO-BUC. They illustrate different tradeoffs between accuracy and coverage (i.e., mistakes and abstentions).

106. Clermont & Eisenberg, *supra* note 22, at 125.

107. Hunter, *supra* note 1, at 54–63.

108. These issues may interact with issues IBP's model does address. For instance, if a defendant copied information fixed in a tangible medium of expression and covered by the subject matter of copyright, a trade secret claim may be preempted under § 301 of the Copyright Act 17 U.S.C.S. § 301 (2004). A trade secret claim that involved an extra element of breach of confidence would not be preempted, but it could be if it involved only improper means. Although IBP's model does not address preemption, some of the same Factors (e.g., regarding confidential relationship) would be relevant to the preemption analysis. One would need to modify the model to add preemption.

More generally, our focus on prediction for deliberation—that is, on analyzing problems given the information in a case database—provides some insights about legal reasoning. Legal prediction research facilitates identifying variables to explain the outcomes of legal cases. Schauer, for instance, emphasizes the possibility of using prediction research systematically to find the right "explanatory variables" or "tendencies" or "factors" for a legal domain.[109] Underlying his concern is the antagonism between predictability of decision-making versus particularism or context-sensitivity. If they exist, the explanatory variables should enable a reasoner to account for a problem's particular circumstances in a way that balances context-sensitivity and predictability. Schauer asks whether conceptual categories that have value in predicting outcomes relate to legal doctrine or to other domain features; alternately, he asks whether there are no categories with predictive value, as legal decision-making is too "particularist" and sensitive to the specific context of the case to be amenable to prediction.[110] "It might also be the case," he suggests, "that such explanatory variables filled what would otherwise be gaps or indeterminacies in the law."[111]

Factors appear to be an appropriate explanatory variable for filling "what would otherwise be gaps or indeterminacies in the law." As used in IBP, CATO, and HYPO, Factors occupy an intermediate level of abstraction between raw facts of cases and legal conclusions pertaining to a claim's issues and elements. They capture stereotypical fact patterns that strengthen or weaken a side's argument on a claim. In assembling the list of Factors, we have drawn on observations of judges and legal scholars who summarize the kinds of fact situations that appear to have mattered in deciding cases involving a particular type of claim. The experiments show that Factors have utility

109. Schauer, *supra* note 2. Discovering explanatory variables (i.e., good predictive categories) automatically in the databases of case texts is still far in the future, requiring much greater facility for natural language understanding than programs currently possess. Given preexisting lists of predictive features, ML programs can select the most predictive of them. Conceivably, a program could also automatically tune a model like IBP's based on cases it encounters, changing logical connections or learning to ignore certain segments that are not used or not determinative. Recent work of Chorley and Bench-Capon, *supra* note 94, does something like this in generating a theory or model of the domain based on value preferences evidenced in the cases.

110. Schauer, *supra* note 2, at 786.

111. *Id.* at 788. Specifically, Schauer says:

> Even in the application of "the best interests of the child" or the other phrases that commonly go under the heading of "standards" rather than "rules," it might turn out upon serious empirical investigation that there were explanatory variables that would enable people to predict the outcome of future cases. Some of these might reflect background but nonlegal norms, such as "give custody to the parent with the higher income" or "give custody to the mother." Others might in fact reflect unstated legal variables. And still others might simply track convergences in human beliefs and behavior, not necessarily easily captured by a norm in a narrow sense. But not only might [the research] program reveal that the explanatory variables of judicial decisions—the descriptive rules—would depart from the variables announced as explaining those decisions, but it might also be the case that such explanatory variables filled what would otherwise be gaps or indeterminacies in the law.

Id.

in predicting outcomes of legal cases and provide a basic terminology for explaining those predictions.[112] The comparative experiment illustrates that the explanatory variables have a range of uses in prediction. Using strong predictors in rules for directly predicting outcomes, as in C4.5 and Ripper (and Naïve-Bayes, in effect), may not be their best use. In IBP, KO-Factors are best used to explain away counterexamples to a hypothesis, a use that supports both prediction and argumentation. Similarly, even though IBP's Weak Factors are too weak to affect predictions empirically, they still have a role to play in arguments and explanations concerning a prediction. An empirically weak predictor may still be the basis for a normatively reasonable counter-argument.

Our research on IBP will continue in two directions. First, we have coupled IBP with a program called SMILE that learns to identify Factors in *textual* descriptions of problem scenarios. SMILE uses marked-up sentences from the squibs of the cases in CATO's database as labeled examples. Each squib contains a textual description of the facts of a case. For each Factor, we manually annotated the sentence(s) in the squib from which one can directly infer that the Factor applies to the case. We have experimented with methods for representing the texts of the sentences to make them more effective as training instances. For instance, we have substituted the names of parties and products with their roles in the trade secret case: plaintiff, defendant, plaintiff's information, etc. We have also used an information extraction program that carried out shallow parsing to identify patterns that capture information related to "who did what" and "what was done to whom." To our knowledge, the combined program, SMILE+IBP, is the first computer program that can generate predictions and explanations of legal cases input as *texts*.[113]

Second, we plan to examine how IBP can formulate other kinds of predictive hypotheses. IBP supports testing only hypotheses resolving conflicts of issue-related Factors of the type where plaintiff's trade secret misappropriation claim is strong on certain Factors, and plaintiff should win the issue related to those Factors even though it is weak on certain other Factors. It would be useful to test other types of hypotheses involving conflicting issues, statutory elements, or normative interests such as:

112. Interestingly, Ruger et al., *supra* note 21, at 1194, claim that the six general characteristics used to predict outcomes of Supreme Court cases in the 2002 Term play the same role in filling Schauer's gap. These characteristics, however, could not be used to provide a legal explanation of the prediction.

113. *See* Brüninghaus, *supra* note 5; Stefanie Brüninghaus & Kevin Ashley, *Generating Legal Arguments and Predictions from Case Texts*, PROCEEDINGS OF THE TENTH INTERNATIONAL CONFERENCE ON ARTIFICIAL INTELLIGENCE AND LAW 65–74 (2005).

- Where plaintiff's case is strong on one issue, plaintiff should win even though it is weak on another issue (e.g., strong on Info-Misappropriated issue but weak on Info-Trade-Secret issue; strong on the Improper-Means issue but weak on the Maintain-Secrecy issue).
- Where defendant's legitimate interest is strong (e.g., in protecting technical employees' freedom to use skills and change employment), defendant should win even though the plaintiff trade secret owner's interest is also strong (e.g., plaintiff's property interest in protecting its return on investment from theft and diminution or its interest in protecting expectations of confidentiality).
- Where one side's claim is strong regarding certain elements of a statute (e.g., the UTSA), the plaintiff should win even though it is weak on another element.
- Where a trade secret claim is strong under one model (e.g., based on the UTSA), but weaker under another model (e.g., based on state criminal law concerning trade secrets).

It is also a goal to test hypotheses that involve more detailed information about why a Factor applies.[114]

While enabling some of these predictive hypotheses is fairly straightforward, others require improvements in the case representation, Domain Model, or in the algorithm for testing hypotheses. Testing more general hypotheses, for instance, will require including in the Domain Model the underlying normative interests or changing the logical connections. More specific hypotheses will require representing more detailed information about case facts and their connections to Factors, as represented in HYPO's Dimensions.[115] Accounting for temporal ordering of events in cases will require an augmented representation. We are also interested in ordering the cases in the database temporally and determining the effects on predictions when the case dates are taken into account, perhaps leading to the identification of legal trends or changes over time.

This article has presented a "design space" of dimensions to consider in building and assessing algorithms for predicting outcomes of new cases based on a database of precedents. Prediction algorithms can be seen as a means for extracting and presenting information from particular databases of cases to guide analysis of new problems. Accordingly, legal prediction has practical value despite the limitations that make reliance on predictions risky for other

---

114. For example, a hypothesis may involve the sufficiency of particular combinations of security measures or the effect of temporal orderings: Is there a difference between a plaintiff's making a public disclosure before or after the defendant breaches a confidence?

115. See ASHLEY, *supra* note 19. RL has the ability to learn rules involving multi-valued attributes and attribute hierarchies, and can differentiate "known to be absent" from "not mentioned as present." We hope to discover whether this gives RL an edge in dealing with detailed Dimensional representations of Factors.

real-world purposes. Prediction algorithms should be compared not only in terms of accuracy, but also in terms of their ability to explain predictions in terms of reasonable legal arguments. Our IBP program tests hypotheses about how issues in a new case will be decided, attempts to explain away counterexamples inconsistent with a hypothesis, but also apprises users of counterexamples and makes explanatory arguments based on them.