Machine Learning, 44, 301–324, 2001

Extracting Context-Sensitive Models in Inductive Logic Programming

ASHWIN SRINIVASAN ashwin@comlab.ox.ac.uk
Oxford University Computing Laboratory, Wolfson Building, Parks Road, Oxford OX1 3QD, United Kingdom

Editors: Peter Flach and Sašo Džeroski

Abstract. Given domain-specific background knowledge and data in the form of examples, an Inductive Logic Programming (ILP) system extracts models in the data-analytic sense. We view the model-selection step facing an ILP system as a decision problem, the solution of which requires knowledge of the context in which the model is to be deployed. In this paper, "context" will be defined by the current specification of the prior class distribution and the client's preferences concerning errors of classification. Within this restricted setting, we consider the use of an ILP system in situations where: (a) contexts can change regularly. This can arise for example, from changes to class distributions or misclassification costs; and (b) the data are from observational studies. That is, they may not have been collected with any particular context in mind. Some repercussions of these are: (a) any one model may not be the optimal choice for all contexts; and (b) not all the background information provided may be relevant for all contexts. Using results from the analysis of Receiver Operating Characteristic curves, we investigate a technique that can equip an ILP system to reject those models that cannot possibly be optimal in any context. We present empirical results from using the technique to analyse two datasets concerned with the toxicity of chemicals (in particular, their mutagenic and carcinogenic properties). Clients can, and typically do, approach such datasets with quite different requirements. For example, a synthetic chemist would require models with a low rate of commission errors which could be used to direct efficiently the synthesis of new compounds. A toxicologist on the other hand, would prefer models with a low rate of omission errors. This would enable a more complete identification of toxic chemicals at a calculated cost of misidentification of non-toxic cases as toxic. The approach adopted here attempts to obtain a solution that contains models that are optimal for each such user according to the cost function that he or she wishes to apply. In doing so, it also provides one solution to the problem of how the relevance of background predicates is to be assessed in ILP.

Keywords: ILP, cost-sensitive models, receiver operating characteristic

1. Introduction

In a remarkable article ("Personal Models of Rationality" (Michie, 1990)), that contains nearly equal measures of mathematics, philosophy, and history Donald Michie introduces the extraction of context-sensitive models thus:–

> The use of 'personal' in my title indicates an extremal point on a spectrum the other end of which bears the label 'universal'. To be more general, I should perhaps say 'context-sensitive', rather than 'personal', where context is set by the knowledge, beliefs and purposes held in common by a specialist group.

The term "model" is used here in a sense akin to the data-analytic usage, that is, descriptions of real or potential data. In this paper we are concerned with obtaining such models with an Inductive Logic Programming (ILP) system. Using domain-specific background knowledge, such systems have constructed models for data in software engineering (Bratko & Grobelnik, 1993), stress analysis in engineering (Dolsak & Muggleton, 1992), environmental monitoring (Dzeroski et al., 1994), electronic circuit diagnosis (Feng, 1992), molecular biology and drug design (King et al., 1996; King, Muggleton, & Sternberg, 1992; Muggleton, King, & Sternberg, 1992), and natural language processing (Zelle & Mooney, 1993). Viewed as tools, these ILP systems are special-purpose products developed for the taxonomist interested in accurate classification schemes. It is further understood that the data provided—usually called "examples"—constitute a representative sample of the population of interest. The manufacturer of a general-purpose ILP tool however can expect it to be deployed by a range of clients with access to data that were, or continue to be collected largely independent of analysis concerns. A topical example of this is provided by the empirical construction of models from databases assembled from routine commercial activity (Fayyad et al., 1996).

In such circumstances, model selection is naturally viewed as a decision problem, namely: which model is the optimal choice for the current context? In this paper, we will interpret the terms in this question as follows: a *model* is a description constructed by an ILP system that can be used, in conjunction with the background knowledge, to classify examples;[1] the *current context* will mean the current specification of prior probabilities of classes and misclassification costs; and the *optimal model* is the model that has the least expected misclassification cost of the alternatives considered.[2] This is in line with statistical decision theory (O'Hagan, 1999). With these restrictions, we explore the once-off construction of a solution set from which optimal models can be extracted for all possible contexts. The implication of results obtained elsewhere (Provost & Fawcett, 1997, 1998; Srinivasan, 1999) is that elements of this solution set lie along a particular curve in the Cartesian space used to define the Receiver Operating Characteristic (or ROC), or simply the "operating characteristic", of the ILP system. This curve allows manufacturers of ILP systems, much like their counterparts in the micro-electronics industry, to describe the behaviour of their device across the range of operating conditions expected in the domain. Clients can then pick the operating point—here a model on the curve—that best suits their requirements (see figure 1).

In principle, a simple procedure can be devised to obtain a piecewise-linear approximation to the optimal characteristic curve. Restricting ourselves to binary classifiers for simplicity, the steps would be: (1) Decide on some small number of different ratios of misclassification costs (that is, cost of false positives to false negatives); (2) For each cost ratio, use the examples and background knowledge to obtain the optimal model along with unbiased estimates of its true- and false-positive rates. Each such estimate results in a point in the (Lorentz) diagram; and (3) Obtain the edge of the convex-hull of the points in the diagram (the reason for this is explained in greater detail in Section 3). This procedure is approximate only because a finite number of alternatives are considered in Step 1.

In practice, executing Step 2 is not straightforward. It has been demonstrated (see Quinlan, 1993) that the presence of irrelevant background knowledge—as could be the

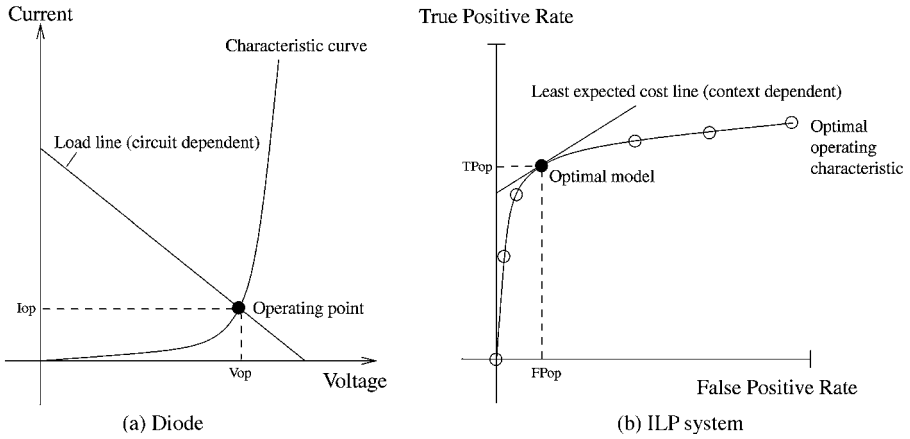


Figure 1. Manufacturer specification of device behaviour. For an electronic device like a diode we are provided with the manufacturer's specification of its (voltage-current) behaviour across the range of legal operating conditions. Its behaviour within any particular circuit can then easily determined by the user by calculating the point of intersection with a "load line"; effectively, the locus of possible operating conditions for that circuit. In an analogous manner, we envisage the manufacturer of the ILP system providing the operating characteristic of the system for the domain (shown here for an ILP system that constructs binary classifiers. Diagrams like these are sometimes called Lorentz diagrams (Hand, 1997)). The current context will identify a point, and hence model, on this curve that is optimal. It can be argued that for problems where actual operating conditions are not known, providing such a curve (or at least approximations to it) is the only responsible course of action for an ILP system manufacturer. To return a single model in such circumstances would be as informative as the diode manufacturer stating that the device provided draws 0.25 mA of current when it has 0.4 V across it.

case here, for not all the background information need be used to construct all models—can result in sub-optimal or manifestly incorrect models. This is due to a combination of the restrictions built-in to practical ILP systems (like the use of non-exhaustive search, resource bounds etc.) and chance-effects. The problem is analogous to the detection of irrelevant features in propositional learning and and the literature contains at least one report (Scott, Niranjan, & Prager, 1998) of a technique for feature-subset selection in variable-cost domains. This technique is adapted here to the problem of selecting subsets of background predicates in ILP, yielding a procedure for "stepwise" inclusion and exclusion of background knowledge. Using this results in a set of first-order models that, while not provably constituting the optimal characteristic curve, nevertheless appear to yield good approximations to that ideal. As it stands, we can only make the following weaker statement about an ILP system equipped with stepwise background selection: any model rejected by the procedure cannot result in a point on the optimal curve for that system.

This paper is organised as follows. Section 2 contains a short specification for a class of ILP systems that identify context-sensitive models within a decision-theoretic setting. Operating characteristics for systems that construct classifiers is described in Section 3. With some restrictions on the form of cost-functions involved, the results concerning the operating characteristics suggest an idealised implementation of the specification provided in Section 2. This implementation is in Section 4. Section 5 describes the technique for

stepwise background selection that results in an approximation to this ideal implementation. Section 6 demonstrates the use of the technique in two important practical problems concerned with the toxicity of chemicals. The problems are instances where it is routine to extract models for use in very different contexts. The results are evidence of how a practical ILP system combined with background selection can be used to identify a set of models that are near-optimal. In doing so, it provides a solution, by construction, to the problem of assessing the relevance of background predicates. Section 7 concludes the paper.

2. Decision-theoretic ILP

The prevalent use of ILP systems has been to construct models for discriminating between two sets of examples (traditionally termed "positive" and "negative", although the positive examples may actually contain instances of more than two classes). The constraints describing the "normal semantics" for ILP (Muggleton & Raedt, 1994) have been adopted as the basis for systems in this class. These are reproduced in figure 2.

These constraints do not provide an accurate picture of practical ILP systems. First, all such systems allow the construction of models that violate *Posterior Satisfiability*. This arises by acknowledging that some members of E^- may be "noisy". Practical ILP systems thus allow clauses inconsistent with the examples to be included in the model. Second, systems usually attempt to identify a model that maximises (or minimises) some measure like expected accuracy (or error). This is not required by the constraints in figure 2. Third, the use of a variety of syntactic and semantic constraints has been found crucially important

Input. Given the following:

1. B: background knowledge consisting of a finite set of clauses;
2. E: a finite set of examples $E^+ \cup E^-$ where:
   - *Positive Examples.* E^+ is a finite, non-empty set of definite clauses;
   - *Negative Examples.* E^- is a finite set of Horn clauses;
   - *Prior Necessity.* $B \not\models E^+$; and
   - *Prior Satisfiability.* $B \cup E^- \not\models \square$

Output. Find:

H: a set of clauses $\{D_1, D_2, \ldots\}$ and:

- *Posterior Sufficiency.* $B \cup H \models E^+$;
- *Posterior Satisfiability.* $B \cup H \cup E^- \not\models \square$

Figure 2. Logical constraints constituting the so-called "normal semantics" of what is required from an ILP system. See Nienhuys-Cheng and de Wolf (1997) for a precise definition of the logic programming terms used here. Briefly, "definite clauses" are rules that are of the form $h \leftarrow b_1, b_2, \ldots$, where $\leftarrow$ should be read as "if". In addition to these kinds of rules "Horn clauses" allow rules of the form $\square \leftarrow b_1, b_2, \ldots$ where $\square$ denotes "false" or a contradiction. In the above $\models$ should be read as "implies." It is normal practice for the D_i to be definite clauses. It is usual to call H a hypothesis, although we will prefer the term "model". This is meant in the sense of being "a model for describing the data" and should not be confused with its logical meaning.

Input. Given the following:

1. B: background knowledge consisting of a finite set of clauses;
2. E: a finite set of examples;
3. I: a finite set of semantic constraints on models;
4. $\mathcal{L}$: a finite set of syntactic restrictions on models;
5. $\mathcal{C}$: a set of acceptable cost functions

Output. Find:

H: a set of models $\{H_1, H_2, \ldots\}$ where:

- Each $H_i \in H$ satisfies the restrictions in $\mathcal{L}$ and is a model for E
- For each $H_i \in H$, $B \cup H_i \cup I \not\models \square$
- For any cost function C in $\mathcal{C}$ there is an $H_i \in H$ such that for the set $\mathcal{H}$ of models considered for E, $C(H_i|B, E) = \min_{H_j \in \mathcal{H}} C(H_j|B, E)$

Figure 3. Constraints describing an ILP system for extracting optimal models from data. We do not elaborate on the nature of restrictions constituting $\mathcal{L}$, save that they are typically in the form of language restrictions. The notion of what constitutes "a model for E" is also deliberately imprecise. For the purposes of this paper, H_i is a model of E if $B \cup H_i$ assigns a class label to each element of E. The class labels allowed are those that appear in E, with the possible addition of a distinguished label "?" to denote "unknown". $C(H|B, E)$ is to be read as "the cost of H given B and E." As it is phrased, $\mathcal{H}$ need not be all possible models for E, but only those considered. The model is therefore only optimal in a restricted sense.

to make model-construction tractable. Again, this is not reflected in figure 2. We adopt the position that ILP systems for data analysis are more correctly viewed as implementations of a specification based on decision-theoretic principles (O'Hagan, 1999). Figure 3 shows such a prescription. There is little that is truly novel in this figure. The use of syntactic and semantic restrictions is not new and has been anticipated in several ways within the literature: we refer the reader to Muggleton and Raedt (1994) for an overview. To our knowledge, at least one ILP implementation (described in Srinivasan and Camacho (1999)) allows the minimisation of a user-defined cost function. However, the notion in figure 3 of providing a set of cost functions does not appear to have been suggested before. This defines a setting within which to address the problem of varying contexts described in the previous section (with each context mapping to an element of $\mathcal{C}$).

Implementations that restrict $\mathcal{C}$ and H to singleton sets result in special cases. These include: (a) the system described in Srinivasan and Camacho (1999), where no restrictions are placed on the cost function; (b) systems that construct models to minimise expected error. These use a particular cost function that assigns the same cost for all misclassification errors; and (c) systems that construct models for the 2-class case by assigning an infinite cost to misclassifying one of the classes. This results in an implementation that conforms to a variant of the normal semantics in figure 2.

This paper is concerned with cases where $\mathcal{C}$ and H are not confined to singleton sets. In particular, the goal is to investigate the construction of a solution set H from which optimal models can be extracted for *any* context. If the elements of $\mathcal{C}$ are known to be of a particular form namely, linear functions of misclassification errors, then results obtained

from a re-appraisal of the (receiver) operating characteristic of system performance can be used to guide the construction of such a solution set.

3. Operating characteristics for classifier systems

Empirical studies comparing systems that construct classifiers typically examine unbiased estimates of predictive accuracy of the classifiers. This corresponds to case (b) in the penultimate paragraph of the preceding section. The assumption is that the model with the highest accuracy is the optimal choice (in the sense of minimising expected costs) for the problem. A qualification on optimality is needed here, as choice is usually restricted to the models available (rather than all possible models) and the estimates are typically subject to sampling errors. Comparisons based on predictive accuracy overlook two important practical concerns, namely (a) class distributions cannot be specified precisely. Distribution of classes in the observed data (the "training" set) are thus rarely matched exactly on new data (the "test" set); and (b) that the costs of different types of errors may be unequal. Using techniques developed in signal detection, Provost and Fawcett (1997) describe an elegant method that takes these considerations into account. Their presentation is restricted to the analysis of systems that construct binary classifiers and is summarised here (taken in part from Srinivasan, King, and Bristol (1999)):

1. Let the two classes be denoted $+$ and $-$ respectively. Let $\pi(+)$ and $\pi(-) = 1 - \pi(+)$ be the prior probabilities of the classes. Suppose a system constructs a binary classifier for which we have unbiased estimates for the following: *TP*, the proportion of instances observed to be $+$ which are classified as such; and *FP*, the proportion of instances observed to be $-$ which are classified as $+$. Using the notation in Breiman et al. (1984), let the costs of false positives and false negatives be $C(+\mid -)$ and $C(-\mid +)$ respectively (that is, the cost of classifying an instance as $+$ when it is really a $-$, and vice versa).
2. The expected misclassification cost of the classifier is then given by $\pi(+) \cdot (1 - TP) \cdot C(-\mid +) + \pi(-) \cdot FP \cdot C(+\mid -)$. For brevity, "expected misclassification cost" will henceforth be simply called "cost". It is easy to see that two classifiers have the same cost if:

$$\frac{TP_2 - TP_1}{FP_2 - FP_1} = \frac{\pi(-) \cdot C(+\mid -)}{\pi(+) \cdot C(-\mid +)} = m$$

   Further, denoting $1 - TP$ as *FN*, the false negative rate, the cost can be rewritten as a linear function of the misclassification rates *FN* and *FP* thus: $\pi(+) \cdot C(-\mid +) \cdot FN + \pi(-) \cdot C(+\mid -) \cdot FP$.
3. The operating characteristic of the system is a point in the two-dimensional Cartesian space defined by *FP* on the X axis and *TP* on the Y axis. This is sometimes called a Lorentz diagram (Hand, 1997). Operating characteristics can also be constructed for a system that constructs a model with continuous output values (for example, a class probability tree). Such a model yields a set of binary classifiers each obtained by employing a different threshold on the output value (for example, instances are classified as $+$ if the probability of class $+$ is at least 0.1, 0.2, 0.3, ...). Each classifier results in a

point in the Lorentz diagram and the operating characteristic of the system is thus a curve (defined by the set of points). A set of points also describes the behaviour of a system as one or more critical parameters are varied. These are those parameters, changes to whose values result in significant changes to the model produced (for example, the minimum acceptable accuracy of rules). The resulting models correspond to a points in a Lorentz diagram.

4. A specification of π and C defines a family of lines with slope m (as defined in item 2 above) in the Lorentz diagram. All classifiers on a given line have the same cost, and the lines are called *iso-performance* lines. Lines with a higher Y intercept represent classifiers with lower cost (follows from the cost formula in item 2). Imprecise specifications of π and C will give rise to a range of possible m values.
5. Minimum cost classifiers lie on the edge of the convex hull of the set of points in item 3 above. For a given value of $m = m_1$, potentially optimal classifiers occur at points in the Lorentz diagram where the slope of the hull-edge is m_1 or at the intersection of edges whose slopes are less than and greater than m_1 respectively (the proof of this is in Provost and Fawcet (1998)). If operating under a range of m values (say $[m_1, m_2]$), then potentially optimal classifiers will lie on a segment of the hull-edge (see figure 4). Of course, statements about "optimality" only refers to the set of classifiers considered and not all possible classifiers that may exist.

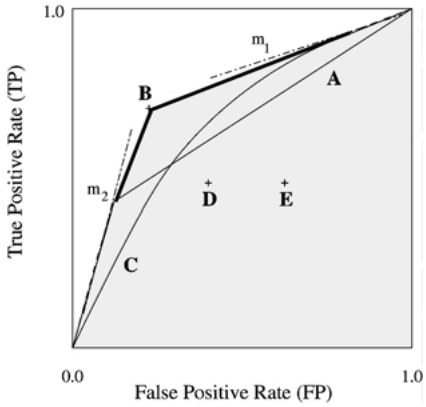


Figure 4. Operating characteristics of systems. Here A and C are systems that result in models with continuous-valued outputs. Their operating characteristics are curves obtained from binary classifiers derived by thresholds on their output values. B, D, and E are systems that construct binary classifiers. Their operating characteristics are points. The edge of the convex hull is the piecewise-linear curve separating the shaded area from the unshaded one. Potentially optimal classifiers lie on this edge and are found by comparing the slope of a linear segment comprising the edge, against the value m determined by the current specification of priors and costs. Thus for $m = m_1$, the only classifier that is potentially optimal is one derived from system C. Imprecise specification of priors and costs will result in a range of values and optimal models then lie on a segment of the hull. Thus if $m \in [m_1, m_2]$ then potentially optimal classifiers lie along the thickened line segment (classifiers from A, B, and C are thus candidates). Classifiers from D and E can never be optimal for any value of m. A system that results in optimal classifiers for any value of m would have a "step" characteristic joining the points $(0, 0)$, $(0, 1)$ and $(1, 1)$.

At this stage, the reader may be concerned with the computational value of this result. Convex hulls of n points in the 2−dimensional (XY) plane can be obtained in $O(n \log n)$ time. For dimensions $d > 3$, this is $O(n^d)$. Would it not be more efficient, therefore, to simply compute the value of the cost function for each of the points, and select the ones with least cost? The power of the approach rests on the general result that the edge of the hull contains the optimal classifiers for *any* choice of priors and costs. Thus, the hull computation can be seen as a once-off effort, that helps eliminate classifiers that could not possibly be optimal under any circumstance. The following additional properties are of interest:

(a) *Estimation.* Obtaining the operating characteristic for a binary classifier requires unbiased estimates of the true- and false-positive rates of the classifier. This does *not* require representative samples.
(b) *Comparison.* A system can only be said to be better than another if its operating characteristic is uniformly dominant. This is a special case of comparing systems based on the area under their characteristic curves. This has been shown to be equivalent to using the Wilcoxon test for testing the null hypothesis that the performance of the systems are equivalent (Hand, 1997).
(c) *Other performance criteria.* The properties concerning optimality are shown in Provost and Fawcett (2000) to extend to performance criteria other than expected cost. An example is the "Neyman-Pearson" criterion in which a user stipulates the maximum false-positive rate that can be tolerated. In the Lorentz diagram this is a vertical line (see figure 5). If the line does not pass through a hull vertex, it would appear that the best classifier would the one at the vertex just to the left of the vertical line. This is clearly

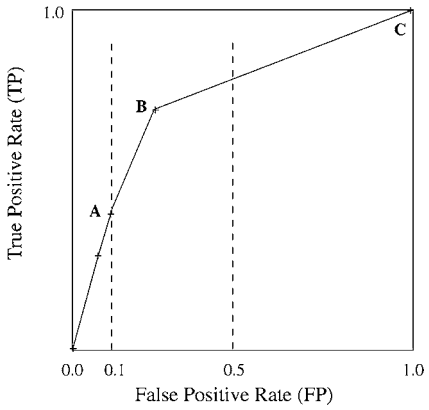


Figure 5. Model selection under the Neyman-Pearson criterion. In this the user specifies the maximum false positive rate that is acceptable. Two such values are shown here (0.1 and 0.5). Classifiers that best comply to these requirements are located at the point of intersection of the dashed lines with hull-edge. For a maximum false positive rate of 0.1, this is clearly a classifier from system **A**. A maximum rate of 0.5 intersects the edge between classifiers from systems **B** and **C**. A classifier that achieves a false positive rate of 0.5 on average is obtained by a probabilistic choice between **B** and **C**.

sub-optimal. However, it is possible to achieve optimal performance (on average) using a randomised decision rule (see next).

(d) *Randomised model selection.* Every point along a line joining a pair of adjacent vertices of the hull represents a realisable classifier. This is evident for the end-points of the line. Let $T_i = (fp_i, tp_i)$ be the co-ordinates of a point along the line joining vertices $T_1 = (fp_1, tp_1)$ and $T_2 = (fp_2, tp_2)$. Then it is easy to show that the expected false and true positive rates from a procedure that randomly selects model T_1 with probability $p = (fp_i - fp_2)/(fp_1 - fp_2)$ and model T_2 with probability $q = 1 - p$ are fp_i and tp_i (this point has been noted in different ways by Provost and Fawcett (2000) and Scott, Niranjan, and Prager (1998)).

Provost and Fawcett (1997) leave open the question of whether the results extend to systems that construct models to discriminate between more than 2 classes. For cost functions that are linear in the misclassification rates, the result concerning the location of optimal classifiers does extend to arbitrary number of classes. This follows directly from results about convex sets that form the basis of linear programming algorithms. Details of this generalisation are available in Srinivasan (1999).

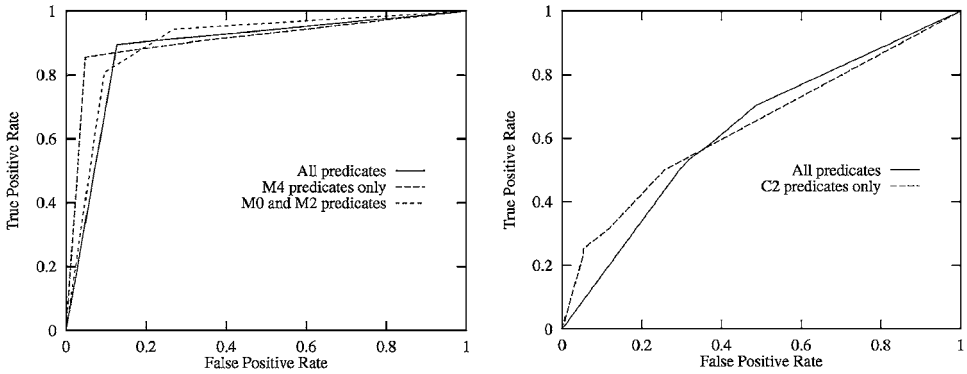
4. Constructing the optimal characteristic for an ILP system

The result that for certain forms of the cost function, optimal classifiers lie on the edge of the convex hull of operating characteristics is of relevance for the developer of an ILP system. As a thought exercise, consider all models that can be constructed by the ILP system. These would arise, for example, by varying critical parameters, incorporating different background predicates, etc. Each model is represented by a point in the Lorentz diagram and the edge of the hull of these points contains the models that minimise any linear function of the misclassification rates. The usual definition of the expected misclassification cost of a model is such a function, namely: $\sum_{i,j,i\neq j} \pi(j)C(i \mid j)P(i \mid j)$ where $\pi(j)$ is the prior probability of a class j; $C(i \mid j)$ is the cost of misclassifying an instance of class j as class i; and $P(i \mid j)$ is the rate at which the model misclassifies an instance of class j as being class i. With the terminology adopted in this paper, a context specifies particular numeric values to the $\pi(j)$ and $C(i \mid j)$. The optimal model for the resulting cost function can be directly identified from the points comprising the hull-edge. In this sense, the piecewise-linear curve comprising the hull-edge is the "optimal characteristic curve" of figure 1, and the points on this curve correspond to models comprising the set H of figure 3.

Obtaining the optimal operating characteristic would require an implementation that executed the following steps (for simplicity, binary classification is assumed):

Procedure OC:

1. Obtain all background knowledge B;
2. Vary critical parameters in all possible ways to obtain models $T_1, T_2, \ldots$ with background knowledge B;
3. For each T_i obtain unbiased estimates (FP_i, TP_i); and
4. Find the operating characteristic by obtaining edge of the convex hull of the (FP_i, TP_i).



Mutagenesis problem (King et al., 1996) Carcinogenesis problem (Srinivasam & King, 1997)

Figure 6. Operating characteristics of an ILP system with and without all background predicates. For each problem predicates are organised into groups. For mutagenesis these groups are: atom, bond structure with related arithmetic operations (M0); expert-defined structural alerts (M1); expert-identified bulk properties with related arithmetic operations (M2); planar ring structures (M3); and three-dimensional structure of molecules (M4). For carcinogenesis the groups are: atom, bond structure with related arithmetic operations (C0); expert-defined toxicity alerts (C1); results from genotoxicity tests (C2); and planar ring structures (C3). The characteristics are obtained as follows: (1) models of increasing generality are obtained using the ILP system P-Progol; (2) the corresponding points in the Lorentz diagram are determined from a 10-fold cross-validation estimate of the true- and false-positive rates of each model; and (3) the operating characteristic is the edge of the convex hull of these points. Ideally, the dominant characteristic should be obtained with all background predicates as the ILP system has access to all the information available. It is evident that this is not the case here, illustrating that the search technique employed finds sub-optimal solutions due to the presence of predicates that are irrelevant in some contexts. Removal of their definitions allows the search to find better models for those contexts.

For such an implementation to be both viable and correct requires: (a) a small number of values that can be assigned to the critical parameters; and (b) for a given assignment of values for the parameters, the model returned is the best one possible. These requirements are difficult to meet in practice. Parameters may be real-valued; search methods may be incomplete and, as mentioned earlier, may return sub-optimal models due to the presence of irrelevant background predicates. Empirical evidence of this is shown in figure 6.

It is evident from these graphs that the operating characteristic of a system with access to all background predicates is not uniformly dominant. This suggests the following modification to the implementation steps earlier:

Procedure OC′:

1. Obtain all the background knowledge B;
2. With each subset B_i of B, vary critical parameters in all possible ways to obtain models $T_{B_i,1}, T_{B_i,2}, \ldots$ with background knowledge B_i;
3. For each $T_{B_i,j}$, obtain unbiased estimates $(FP_{B_i,j}, TP_{B_i,j})$; and
4. Find the operating characteristic by obtaining the edge of the convex hull of the $(FP_{B_i,j}, TP_{B_i,j})$.

Assuming some finite set of values for critical parameters, we can take the curve so obtained as providing the best approximation to the optimal characteristic.

The number of models constructed by Procedure OC′ is clearly exponential in the number of predicates in the background knowledge. In Scott, Niranjan, and Prager (1998), the authors confront an analogous problem when dealing with feature-based learning methods. Their solution takes the form of an incremental hull-construction procedure that uses stepwise variable selection procedures developed for model-construction in statistics (described in Section 5). The resulting hull-edge—defining the operating characteristic—identifies a set of models each obtained using different feature subsets. While all models rejected by the procedure are provably sub-optimal, it cannot be shown that the answer returned is a subset of the set of optimal models. Nevertheless, the operating characteristic has been found empirically to dominate those obtained from normal usage, suggesting the result to be a reasonable approximation to the optimal solution set. The method also appears to be reasonably efficient in practice, although in the worst-case it remains exponential in the number of features. It is straightforward to adapt this technique to ILP, yielding procedures for the stepwise selection of background predicates. This adaptation is described in the following section.

5. Stepwise background selection

Stepwise procedures for variable-subset selection are employed in conjunction with statistical model-fitting methods like regression. The need for such procedures arises when regressor variables are not clearly defined by an underlying theory. Instead, a large number of potential regressors are available and we are interested in selecting some subset of these to appear in the model. The techniques are now routinely provided in most modern statistical packages (see for example, the description in Norusis (1994)). The main steps are summarised in figure 7.

It is normal to start the stepwise procedure in figure 7 with $I = \emptyset$. Although it is not guaranteed to find the best subset, and in the worst case, the number of subsets examined can be exponential in $|V|$ the method has been found to work well in practice. Restricted variants of the method are also popular: *forward selection* starts with $I = \emptyset$ and dispenses with the exclusion steps (Steps 6–7 in figure 7); *backward elimination* starts with $I = V$ and dispenses with the inclusion steps (Steps 4–5 in figure 7). While both variations examine no more than $O(|V|^2)$ subsets, empirical studies suggest that backward elimination usually yields better models (Stuart, Ord, & Arnold, 1999).

This stepwise variable selection procedure has been adapted in Scott, Niranjan, and Prager (1998) to obtain the operating characteristic of any feature-based system that constructs classifiers. The resulting algorithm ("Parcel") returns a set of feature-subsets from which the operating characteristic is obtained as follows: (a) the system constructs one model for each feature subset; (b) each model is represented by a point in the Lorentz diagram. This point may be estimated from sample data; and (c) the operating characteristic is the piecewise-linear curve joining these points. The feature-subsets are obtained by an incremental convex-hull construction process, the main steps of which are summarised in figure 8.

$svs(V, I, F_{in}, F_{out})$: Given a set of potential regressor variables V; an initial subset of variables $I \subseteq V$; and minimum values of the F statistic that a variable must achieve to enter (F_{in}) or remain (F_{out}) in the regression equation, returns a subset $S \subseteq V$ identified by a stepwise variable selection procedure.

1. $i = 0$
2. $S_i = I$, $V_i = V \setminus I$
3. Increment i
4. Let v_{in} be the single best variable in V_{i-1} that can be included (that is, on inclusion, gives the greatest increase in the coefficient of determination)
5. If $f(v_{in}|S_{i-1}) \geq F_{in}$ then $S = S_{i-1} \cup \{v_{in}\}$; otherwise $S = S_{i-1}$
6. Let v_{out} be the single best variable in S that can be excluded (that is, on exclusion, gives the greatest increase in the coefficient of determination)
7. If $f(v_{out}|S \setminus \{v_{out}\}) \leq F_{out}$ then $S_i = S \setminus \{v_{out}\}$; otherwise $S_i = S$
8. If $S_i = S_{i-1}$ then return S_i; otherwise continue
9. $V_i = V \setminus S_i$
10. Go to Step 3

Figure 7. A stepwise variable selection procedure for multiple linear regression. The coefficient of determination (often denoted by R^2) denotes the proportion of total variation in the dependent variable that is explained by the fitted model. Given a model formed with the set of variables X, it is possible to compute the observed change in R^2 due to the addition of some variable v. The probability that the true value of this change is 0 can be obtained from a use of the F statistic (Walpolt & Myers, 1978). The function $f(v \mid X)$ returns the value of the F distribution under the null hypothesis that there is no change in R^2 by adding variable v to those in X. The thresholds F_{in} and F_{out} thus specify acceptable probability levels for the inclusion (and exclusion) of variables. It is evident that $F_{in} > F_{out}$ in order to avoid the same variable from repeatedly being included and excluded. A correct implementation of $svs(\ldots)$ also requires sample data and the appropriate regression function to be provided as parameters. We have ignored these here for simplicity.

The procedure in figure 8 has the nice property that on each iteration, the hull constructed contains all previous ones. As with stepwise regression, in the worst case the number of subsets examined is exponential in $|F|$. While restricted procedures analogous to forward selection or backward elimination are possible, they do not alter the worst-case bound.[3] While this does not appear to have caused difficulty in practice (see Scott, Niranjan, & Prager, 1998), our adaptation of the method to stepwise selection of background predicates will introduce two small extensions:

– We will allow predicates to be considered as groups. A predicate group is simply a set of predicates: this is similar to "hierarchical linear regression modelling" where a set of variables can be examined as a single "block". This feature is available in statistical packages like SPSS (Norusis, 1994). This serves two purposes. First, it allows a reduction in the total number of subsets to be examined (although groups can contain only a single predicate, in which case no savings are to be made). Second, for the problem considered, predicates can fall into natural groups (for example, a predicate may necessarily require the presence of others). Examples of such groups appeared in figure 6; and

$sfs(F, I)$: Given a a set of features F; and an initial set of feature-subsets $I = \{f_1, \ldots\}$ where each $f_i \subseteq F$, returns an set of feature-subsets $S = \{s_1, \ldots\}$ where each $s_i \subseteq V$ is used by a feature-based system to construct a model that defines the final operating characteristic.

1. $i = 0$
2. $S_i = I$
3. Increment i
4. Let $F_i = \{f | f \subseteq F$ and for all $f' \subseteq F$ s.t. $|f \Delta f'| = 1 \Rightarrow f' \in S_{i-1}\}$. That is F_i is the set of feature-subsets obtained by all possible additions or deletions of a single feature from elements of S_{i-1}
5. Let C_i be the convex hull of points corresponding to models obtained with each feature-subset in $S_{i-1} \cup F_i$,
6. Let S_i be the elements of $S_{i-1} \cup F_i$ that result in points on the edge of C_i
7. If $S_i = S_{i-1}$ then return S_i; otherwise continue
8. Go to Step 3

Figure 8. A stepwise feature selection procedure. This forms the basis of the Parcel algorithm. In the implementation described in Scott, Niranjan, and Prager (1998), the algorithm is invoked with $I = \emptyset$. This corresponds to the operating characteristic of a random guesser. The procedure then examines feature-subsets that contain just one feature. Models are constructed with each such subset and those subsets that result in points on the edge of the convex hull are retained. The procedure then iterates by adding and deleting features in the stepwise manner adopted by the corresponding procedure for linear regression. At each iteration, only those subsets comprising the edge of the convex hull are retained and the procedure halts when there is no change in the hull (in the actual implementation, Parcel halts when there is no "significant" change in the hull). A correct implementation of *sfs*(...) also requires sample data and the appropriate classification function to be provided as parameters. We have ignored these here for simplicity.

– We will allow an upper bound on the number of iterations of hull-construction. This will ensure that the number of subsets examined is polynomial in the number of predicate-groups.

With these minor changes, stepwise background selection is very similar to the procedure for stepwise feature selection: the main steps are in figure 9. We note the following features of the procedure:

1. In Step 6, the convex hull obtained on an iteration contains the hulls obtained on all previous iterations. This follows trivially by construction.
2. In Step 7, models constructed with predicate groups not in S_i are necessarily sub-optimal. This follows directly from the fact that the corresponding points in the Lorentz diagram are not on the edge of the convex-hull.
3. Models constructed with predicate groups in S_i are not necessarily optimal. This follows from the fact that with a set of background predicates, a practical ILP system may be unable to extract the best model.
4. In the worst case, the number of subsets examined is $O(|B|^k)$. Assume the procedure is invoked $I = \emptyset$ and that there at most n variations possible for the parameters. In

$sbs(B, I, k)$: Given a a set of predicate groups B; an initial set of predicate groups $I = \{b_1, \ldots\}$ where each $b_i \subseteq B$; and an upper-bound k on the number of iterations, returns an set of predicate groups $S = \{s_1, \ldots\}$ where each $s_i \subseteq B$ is used by an ILP system to construct a model that defines the final operating characteristic.

1. $i = 0$
2. $S_i = I$
3. Increment i
4. If $i > k$ then return S_{i-1}; otherwise continue
5. Let $B_i = \{b | b \subseteq B$ and for all $b' \subseteq B$ s.t. $|b \Delta b'| = 1 \Rightarrow b' \in S_{i-1}\}$. That is B_i is the the set of predicate groups obtained by all possible additions or deletions of a single group from elements of S_{i-1}
6. Let C_i be the convex hull of points corresponding to models obtained by varying critical parameters systematically with each predicate group in $S_{i-1} \cup B_i$,
7. Let S_i be the elements of $S_{i-1} \cup B_i$ that result in points on edge of C_i
8. If $S_i = S_{i-1}$ then return S_i; otherwise continue
9. Go to Step 3

Figure 9. A stepwise background selection procedure for an ILP system inspired by the Parcel algorithm. A "predicate group" refers to a set of predicate definitions. A correct implementation of $sbs(\ldots)$ also requires sample data and the ILP system to be provided as parameters. We have ignored these here for simplicity.

the worst case, after the first iteration each element of B with each possible parameter variation lies on the hull-edge. Each of these is now in $|B|$ possible ways to give sets of 2 predicate-groups each. Again, in the worst case, we can assume that all possible choices lie on the hull-edge. Thus, on the ith iteration, the number of subsets examined is $n \times {}^{|B|}C_i$, and the total number of subsets is $\sum_{i=1}^{k} n \times {}^{|B|}C_i$. This is $O(|B|^k)$. The same bound is reached if the procedure was invoked with $I = B$, or if greedy variants akin to forward selection or backward elimination were used.

These remarks suggest that it is possible to obtain a reasonably efficient technique for approximating the optimal characteristic curve of an ILP system. We investigate its performance on some realistic problems in the next section.

6. Empirical evaluation

The stepwise background selection procedure of the previous section is evaluated empirically on two real-world biochemical datasets. The data describe chemical compounds and illustrate well the two main issues raised in the paper. First, the data are of interest to clients with very different goals. The toxicologist seeks to identify all chemicals that may be potentially hazardous, and therefore seeks models that make as few errors of

omission as possible. The synthetic chemist, on the other hand, seeks models with a low rate of commission errors which could be used to direct efficiently the synthesis of new compounds. Second, the data describe bulk and structural properties of chemicals and are largely agnostic to the requirements of either type of client. With each dataset we compare the operating characteristic obtained by using stepwise background selection against that obtained by using the "all-subsets" procedure (OC′) described in Section 4. Recall that the curve obtained with the latter is taken as the best approximation to the optimal characteristic of an ILP system.

6.1. Materials

6.1.1. Domains. The following is adapted from Srinivasan (2000).

Mutagenesis. Models are to be constructed for discriminating amongst nitroaromatic chemicals with varying mutagenic activity.[4] The data pertain to chemicals examined by Debnath et al. (1991), with the view of constructing a linear model to predict levels of biological activity using molecular properties as regressor variables. These properties were extended in King et al. (1996) and Srinivasan et al. (1996) to include general structural information (atoms, bond connectivity, etc.: described in the following section). The original chemical study listed the mutagenic activity of 230 compounds. These were taken to be composed of two disparate groups of 188 and 42. Each compound has an associated mutagenicity value obtained from a procedure known as the Ames test. We concentrate only on the subgroup of 188 compounds, as these are sufficient for the purposes of the study here (The 42 compounds were found to be outliers for a particular analysis technique, namely linear regression. The purpose here is to demonstrate stepwise background selection, and not curve fitting). The classification of the compounds is binary: those with positive log mutagenicity are labelled 'active" and those which have zero or negative log mutagenicity are labelled "inactive". Of the 188 chemicals, 125 (67%) are active.

Carcinogenesis. The data pertain to chemicals undergoing rodent carcinogenicity tests conducted within the U.S. National Toxicology Program (NTP). These tests are conducted on a very wide range of chemicals and the data describing these chemicals have been used to construct models to discriminate true carcinogens from a pool of potential carcinogens (this is harder than discriminating the former from non-carcinogens (Benigni (1998)). Outcomes from rodent tests with approximately 300 chemicals are available. As with the mutagenesis problem, the data on bulk molecular descriptors have been augmented with structural information (see the following section) and have previously formed the basis for a popular "challenge" problem for symbolic machine learning techniques (see Srinivasan, King, & Bristol (1997) and Srinivasan et al. (1997)). Compounds are again classified in one of two classes: "true carcinogens" and "others". The dataset refers to 337 chemicals of which 182 are classified as true carcinogens (54%).

6.1.2. Background knowledge and predicate groups. Data constituting the background knowledge for mutagenesis is contained in the definition of 29 predicates. These naturally fall into the following groups:

M0. Molecular description at the atomic level. This includes the atom and bond structure, the partial charges on atoms, and arithmetic constraints (equalities and inequalities). There are 5 predicates in this group;
M1. Structural properties identified by experts as being related to mutagenic activity. These are: the presence of three or more benzene rings, and membership in a class of compounds called acenthrylenes. There are 2 predicates in this group;
M2. Chemical properties identified by experts as being related to mutagenic activity, along with arithmetic constraints (equalities and inequalities) The chemical properties are: the energy level of the lowest unoccupied molecular orbital ("LUMO") in the compound, an artificial property related to this energy level (see Debnath et al., 1991), and the hydrophobicity of the compound. There are 6 predicates in this group;
M3. Generic planar groups. These include generic structures like benzene rings, methyl groups, *etc.*, and predicates to determine connectivity amongst such groups. There are 14 predicates in this group; and
M4. Three-dimensional structure. These include the positions of individual atoms, and constraints on distances between atom-pairs. There are 2 predicates in this group.

The background knowledge for carcinogenesis is contained in the definition of 41 predicates. These naturally fall into the following groups:

C0. Molecular description at the atomic level. This is similar to M0 above and is comprised of 5 predicates;
C1. Toxicity properties identified by experts as being related to carcinogenic activity, and arithmetic constraints. These are an interpretation of the descriptions in Ashby and Tennaut (1991), and are contained within the definitions of 5 predicates;
C2. Short-term assays for genetic risks. These include the *Salmonella* assay, in-vivo tests for the induction of micro-nuclei in rat and mouse bone marrow *etc.* The test results are simply "positive" or "negative" depending on the response and are encoded by a single predicate definition; and
C3. Generic planar groups. These are similar to M3 above, extended to 30 predicate definitions.

All experiments will involve subsets of predicate groups only. Thus, for mutagenesis, there are 31 non-trivial subsets of predicate-groups. For carcinogenesis, this number is 15.

6.1.3. Algorithms and machines. All experiments use the ILP system P-Progol (Version 2.7.5).[5] The experiments were performed on machine equipped with a 266 Mhz Pentium II processor and 128 megabytes of random access memory.

6.2. Method

We adopt the following method:

For each problem (Mutagenesis and Carcinogenesis):

1. Obtain the operating characteristic by examining all possible subsets in the manner described in Section 4 (Procedure OC′). Step 3 in that procedure requires unbiased estimates of the true positive rate and false positive rate of each model constructed. This is obtained using a 10-fold cross-validation design (Weiss & Kulikowski, 1991), as is an estimate of the time taken to construct each model;
2. Select subsets using the stepwise background selection procedure described in Section 5 (figure 9). An operating characteristic is then constructed using the models obtained with these subsets. The estimates of true and false positive rates for any model are obtained using a 10-fold cross-validation design, as is an estimate of the time taken to construct each model (using the same "data splits" as Step 1); and
3. The results from Step 2 are compared against those from Step 1 (see details below).

The following details are relevant:

(a) With a given subset, procedures invoked in Steps 1, 2 construct models by systematic variation of the critical parameters of the ILP system. Experiments on the carcinogenesis data reported in Srinivasan and King (1997) suggest that the most sensitive parameter for P-Progol is one that stipulates the minimum acceptable accuracy of rules on data available for model construction (P-Progol calculates this as the ratio $t_p/(t_p + f_p)$ where t_p, f_p are the number of true- and false-positive classifications made by the rule on the data). Here, this will be the only critical parameter considered. Initial experimentation suggested that it was adequate to consider changes to values in steps of 10%, and for both domains, minimum rule accuracies below 70% produced trivial models. Therefore, with each subset, we will consider 4 settings for minimum rule accuracy: 100%, 90%, 80%, and 70%. Each such setting results in a model, whose true- and false-positive rates are estimated from a 10-fold cross-validation;
(b) The models corresponding to the operating characteristic in Step 1 will be taken as near-optimal. In Step 3 we will examine how many of these are identified by the characteristic constructed in Step 2. To construct this characteristic, stepwise background selection requires a bound k on the number of iterations. There is no prescriptive value for this, and we will examine the change in the number of near-optimal models identified as k increases;
(c) For each value of k above, we will test for significant differences in the areas under the characteristics obtained in Steps 1 and 3. A significance test described in Scott, Niranjan, and Prager (1998) allows the calculation of the z value. Normally, a value of $z > 1.96$ indicates a significant difference (at $P = 0.05$). However the reader should be aware of two issues. First, the test in Scott, Niranjan, and Prager (1998) is biased: only large differences are adjudged significant. Second, repeated use of any statistical test will result in an erroneous result simply due to chance-effects. The standard correction (sometimes called the "Bonferroni correction": see Salzberg (1997)) also requires differences to be large before they can be adjudged significant. For example, applying the correction to comparisons here requires a value of $z > 2.57$ to indicate a significant difference at $P = 0.05$. We will take this as the critical value of z; and
(d) The time taken by each method is simply be the sum of the times taken to construct all the models examined.

Method	Mutagenesis				Carcinogenesis			
	Found	Seen	Time	Area	Found	Seen	Time	Area
OC′	6	31	5265	0.945	3	15	15277	0.659
sbs $k = 0$	0	0	0	$0.500^{\dagger}$	0	0	0	$0.500^{\dagger}$
$k = 1$	3	5	543	0.918	2	4	2663	0.641
$k = 2$	4	15	2041	0.940	2	7	4413	0.643
$k = 3$	4	22	3239	0.942	2	9	6083	0.653
$k = 4$	5	24	3784	0.944	3	11	9606	0.659
$k = 5$	5	27	4704	0.944	3	13	12945	0.659

Figure 10. Comparative performance of an "all-subsets" approach (Procedure OC′ in Section 4) and stepwise background selection (Procedure *sbs* in figure 9) with an upper-bound k on the number of iterations. For each problem, we tabulate the following: "found" refers to the number of near-optimal models identified by the operating characteristic obtained; "seen" refers to the number of models constructed; "time" is the estimate of time (in seconds) to construct the operating characteristic (obtained from 10-fold cross-validation); and "area" is the area under the operating characteristic. A † marking an area entry denotes a significant difference to the area under the characteristic obtained with OC′. In all cases, we do not include the two trivial models. One classifies all instances as "negative" and the other classifies all as "positive". These correspond to points (0, 0) and (1, 1) in the Lorentz diagram.

6.3. Results and discussion

Figure 10 tabulates the performance of stepwise background selection with increasing values of k (the number of iterations allowed). The principal details in figure 10 are these: (1) For both problems, with sufficiently large k, stepwise background selection identifies all (carcinogenesis) or nearly all (mutagenesis) the models found by the exhaustive search employed by Procedure OC′; and (2) With stepwise selection, larger values of k result in characteristic curves that approximate more closely the one from OC′. However, for both problems, there are diminishing returns from progressive increases in k: the models examined and the time taken increase out of proportion with the gains made in area under the characteristic curve.

These results suggest that it is possible to obtain a good approximation to the "optimal" characteristic using the stepwise background selection procedure with low values of k. This is illustrated graphically in figure 11. Of course in actual use, the value of k would have to be pre-specified or established by some stopping criterion. The Parcel algorithm (figure 8) uses as stopping criterion the z value from the test for significant changes in area. Adopting this approach would cause stepwise background selection to halt at $k = 2$. It is also instructive to examine the parameter values and predicate-groups used to obtain each vertex of the operating characteristics shown in figure 11. These are tabulated in figure 12.

We make the following additional observations about the experiments:

– The ILP system performs better on the mutagenesis problem. This is apparent both from figures 10 and 11, which show a large difference in the areas under the best operating characteristic in each case. In the mutagenesis problem, this area is close to the maximum attainable (1.0). This suggests that the data available are largely adequate for the ILP

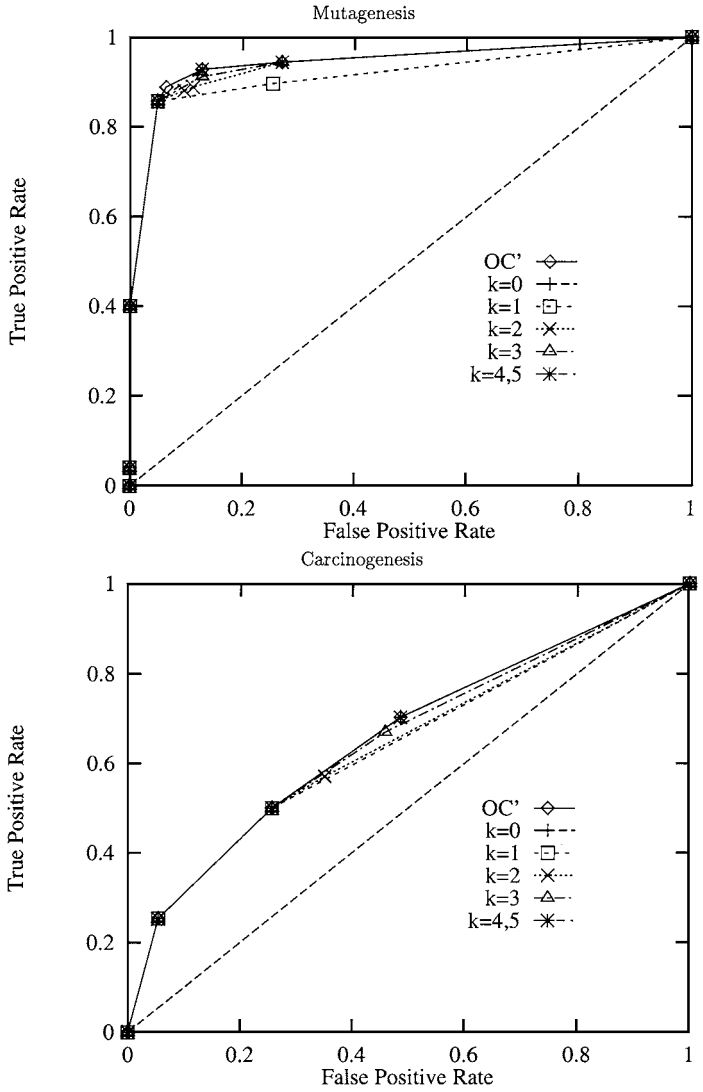


Figure 11. Convergence of stepwise background selection to the result of performing an exhaustive search (Procedure OC′). The characteristic for $k = 0$ is the line joining (0, 0) (the model that classifies all instances as "negative") and (1, 1) (the model that classifies all instances as "positive"). For each curve, vertices other than (0, 0) and (1, 1) correspond to models obtained with some value assigned to the parameter being varied (minimum rule accuracy) and some subset of predicate groups.

system to construct models across the full range of costs possible. Better performance curves for carcinogenesis may be obtained with changes to one or both of the data and the ILP system used.

– Both problems have clearly benefitted from the grouping of predicates. This has allowed a significant reduction in the overall time-complexity of the methods compared. In

Method	Mutagenesis		Carcinogenesis	
	Minimum Rule Acc.	Predicates Used	Minimum Rule Acc.	Predicates Used
OC'	100%	M1 *	90%	C2 *
	100%	M3 *	70%	C2 *
	100%	M4 *	70%	C0,C1,C2,C3 *
	100%	M2,M3,M4 *		
	70%	M0,M1,M3,M4 *		
	70%	M0,M2 *		
$sbs\ k = 1$	100%	M1 *	90%	C2 *
	100%	M3 *	70%	C2 *
	100%	M4 *		
	70%	M2		
$k = 2$	100%	M1 *	90%	C2 *
	100%	M3 *	70%	C2 *
	100%	M4 *	70%	C1,C2
	90%	M1,M4		
	70%	M1,M2		
	70%	M0,M2 *		
$k = 3$	100%	M1 *	90%	C2 *
	100%	M3 *	70%	C2 *
	100%	M4 *	70%	C0,C1,C2
	70%	M0,M1,M4		
	70%	M0,M2 *		
$k = 4, 5$	100%	M1 *	90%	C2 *
	100%	M3 *	70%	C2 *
	100%	M4 *	70%	C0,C1,C2,C3 *
	70%	M0,M1,M3,M4 *		
	70%	M0,M2 *		

Figure 12. Further details on vertices of the operating characteristics. For a given method, one row tabulates a value for the principal parameter (minimum rule accuracy) and a subset of predicate groups. The ILP system uses this information to construct a model. The true- and false-positive rates of the model are estimated, and the resulting point forms a vertex in the corresponding operating characteristic. For a given method, the rows from top to bottom result in models with increasing true- and false-positive rates (that is, points that progressively move from (0, 0) to (1, 1) in the characteristic). A "*" alongside a row indicates that the resulting model is near-optimal, in the sense that it is also identified by exhaustive search. Details for the trivial models corresponding to (0, 0) and (1, 1) are not tabulated.

general, the reader may be concerned on the viability of stepwise background selection when such groups may not be apparent. Efficiency can be achieved at the expense of exactness. If, for example, we allow the characteristic obtained on an iteration to be approximated by one that contained no more than a fixed number of vertices, then the number of models constructed is $O(|B|)$. We do not explore this further here, but accept that such compromises may be inevitable for some problems.

– In general, the manual approach used here for the identification of the critical parameter (minimum rule accuracy) and its values (100%, 90%, 80%, and 70%) is unsatisfactory. Normal design practice makes it the manufacturer's responsibility to identify the parameters that critically govern particular aspects of system behaviour. For an ILP system with this information, it would be preferable to precede the stepwise procedure by some form of automatic sensitivity analysis.
– The tabulation in figure 12 allows us to extract models for different cost requirements. Here we present two such models, obtained from the entry under "Mutagenesis" for *sbs*, $k = 2$. The first model in figure 13 is fairly specific and avoids errors of commission at the

A "synthetic-chemist friendly" model:
A nitroaromatic has positive log mutagenicity if it has:
a pair of connected benzene rings; or
at least one non-aromatic 6-membered carbon ring.

A "toxicologist friendly" model:
A nitroaromatic has positive log mutagenicity if it has:
a LUMO value of at most -1.387; or
a carbonyl oxygen with partial charge of -0.391; or
a carbonyl oxygen with partial charge of at most -0.418; or
a pair of aliphatic carbons connected by a single bond.

Figure 13. Example models comprising the operating characteristic obtained with stepwise background selection ($k = 2$). The models correspond to the second and last rows against sbs, $k = 2$ in the "Mutagenesis" column of figure 12.The first has an estimated false-positive rate of 0% and a true-positive rate of 40%. This corresponds to the point (0, 0.4) in the Lorentz diagram in figure 11. The corresponding figures for the second model are 27% and 94% respectively.

expense of errors of omission. This is likely to be of interest to a synthetic chemist. The second model in figure 13 is fairly general and avoids errors of omission at the expense of errors of commission. Such a model is more likely to interest the toxicologist. The models are presented here for illustrative reasons only. In actual practice, a classifier best suited to a client's cost constraints would be selected either directly from those constituting the operating characteristic, or by using the randomised model selection procedure described in Section 3.

Finally, we note some wider implications for ILP systems:

- The operating characteristic provides a complete description of system-performance. Such a specification is necessary when evaluating an ILP system's behaviour in and across problems where error-costs and class-distributions can vary.
- The operating characteristic provides, by construction, one solution to the question of how the relevance of background predicates is to be assessed. Thus, predicates that do not appear in any model comprising the operating characteristic are clearly irrelevant. Of the rest, relevance is decided based on the current specification of error-costs and class-distributions. The same principle can also be used to guide the "invention" of a new background predicate. As a heuristic, any such predicate can be taken as useful if its inclusion alters the current operating characteristic (or, as a stricter criterion, causes a significant increase in the area under the operating characteristic).

7. Concluding remarks

This paper has been concerned with the use of an ILP system to extract classifiers under the following conditions: (a) the costs of mis-classification and distribution of classes may vary in ways that are difficult to anticipate; and (b) the data have been collected in a

manner that is largely independent of analysis concerns. Such scenarios are increasingly presenting themselves as programs are called upon to extract models from large, pre-compiled databases. Under these conditions, any one classifier is unlikely to yield the best performance in all circumstances. Results originally developed in signal-detection theory system state—with some restrictions on the nature of the cost function—that cost-optimal classifiers for such a situation correspond to points on a particular curve (its "operating characteristic") in a special Cartesian space. The main contribution of this paper has been to show how methods developed in statistics and feature-based learning can be adapted to allow an approximate identification of this curve (and the corresponding models) within ILP.

The notion of the operating characteristic for an ILP system has an engineering analog. Performance tests—measurements of the behaviour of a product under the environmental conditions expected—are a routine part of engineering product design (Cain, 1969). It is also normal for manufacturers to provide "data sheets" that contain graphs of typical performance characteristics obtained from such testing. These are an invaluable guide to the user, as it allows him or her to anticipate the product's behaviour under the conditions of interest. The approach taken here allows us to emulate this kind of engineering practice by including a data sheet that contains the ILP system's operating characteristic.

In closing, we return to the paper by Michie cited at the outset of this paper. There, he emphasises the value of an approach that yields "... personal theories, executable in the head, and structured to the agent's viewpoint and goals" (Michie, 1990, page 395). Michie's solution is a much more elegant one than that presented here. He provides a technique of constructing a full-scale probabalistic (Bayesian) propositional classifier. The combination of decision-theory and ILP advocated here can be seen as one step towards this goal.

Acknowledgments

The author is supported by a Nuffield Trust Research Fellowship at Green College, Oxford, and also by the European Union project IST-1999-11495:SolEuNet. This paper is an expanded version of the abstract presented in the "late-breaking papers" session of the Ninth International Workshop on Inductive Logic Programming (ILP'99). As always, Donald Michie has provided the author with invaluable advice. Thanks are also due to: Simukai Utete for bringing to the author's attention the work of M. Niranjan and his co-workers; David Page and James Cussens for help with statistical and algorithmic issues; Steve Moyle for the generous use of his personal computer; Foster Provost and Tom Fawcett for discussions on ROC curves and making available their computer program ROCCH (www.croftj.net/~fawcett/ ROCCH/); and the reviewers of this paper for their useful suggestions. This paper is dedicated to the memory of the author's grandmothers, who passed away as the last paragraphs here were being written. Their affection and support are greatly missed.

Notes

1. We will therefore use the terms "model" and "classifier" interchangeably.
2. The restriction to models considered, rather than all possible models is simply for reasons of tractability. A formula for calculating expected costs appears in a later section.

3. A principal difference is that stepwise techniques for regression are only required to identify a single subset of variables. The operating characteristic can require several subsets of features—in pathological cases, all possible subsets of features may be needed.
4. Mutagenic chemicals mutate DNA sequences. Nitroaromatics occur in automobile exhaust fumes and are also common intermediates in the synthesis of many thousands of industrial compounds. Highly mutagenic nitroaromatics have been found to be carcinogenic and it is of considerable interest to the chemical and pharmaceutical industry to determine which molecular features result in compounds having mutagenic activity.
5. This is available at: www.comlab.ox.ac.uk/oucl/research/areas/machlearn/PProgol/pprogol.pl

References

Ashby, J., & Tennant, R. W. (1991). Definitive relationships among chemical structure, carcinogenicity and mutagenicity for 301 chemicals tested by the U.S. NTP. *Mutation Research, 257*, 229–306.

Benigni, R. (1998). (Q)SAR prediction of chemical carcinogenicity and the biological side of the structure activity relationship. In *Proceedings of The Eighth International Workshop on QSARs in the Environmental Sciences*, 1998. Baltimore.

Bratko, I., & Grobelnik, M. (1993). Inductive learning applied to program construction and verification. In *Third International Workshop on Inductive Logic Programming* (pp. 279–292). Available as Technical Report IJS-DP-6707, J. Stefan Inst., Ljubljana, Slovenia.

Breiman, L., Friedman, J. H., Olshen, R. A., & Stone, C. J. (1984). *Classification and regression trees.* Belmont: Wadsworth.

Cain, W. D. (1969). *Engineering product design*. London: London Business Books.

Debnath, A. K., Lopez de Compadre, R. L., Debnath, G., Schusterman, A. J., & Hansch, C. (1991). Structure-activity relationship of mutagenic aromatic and heteroaromatic nitro compounds. Correlation with molecular orbital energies and hydrophobicity. *Journal of Medicinal Chemistry*, *34:2*, 786–797.

Dolsak, B., & Muggleton, S. (1992). The application of inductive logic programming to finite element mesh design. In S. Muggleton (Ed.), *Inductive logic programming*. London: Academic Press.

Dzeroski, S., Dehaspe, L., Ruck, B., & Walley, W. (1994). Classification of river water quality data using machine learning. In *Proceedings of the Fifth International Conference on the Development and Application of Computer Techniques Environmental Studies*. Southampton: Computational Mechanics Publications.

Fayyad, U. M., Piatetsky-Shapiro, G., Smyth, P., & Uthurusamy, R. (Eds.). (1996). *Advances in knowledge discovery and data mining*. Menlo Park, CA: AAAI press (co-published by MIT Press).

Feng, C. (1992). Inducing temporal fault dignostic rules from a qualitative model. In S. Muggleton (Ed.). *Inductive logic programming*. London: Academic Press.

Hand, D. J. (1997). *Construction and assessment of classification rules*. Chichester: Wiley.

King, R. D., Muggleton, S. H., Srinivasan, A., & Sternberg, M. J. E. (1996). Structure-activity relationships derived by machine learning: The use of atoms and their bond connectivities to predict mutagenicity by inductive logic programming. In *Proc. of the National Academy of Sciences*, *93*, 438–442.

King, R. D., Muggleton, S. H., & Sternberg, M. J. E. (1992). Drug design by machine learning: The use of inductive logic programming to model the structure-activity relationships of trimethoprim analogues binding to dihydrofolate reductase. In *Proc. of the National Academy of Sciences*, *89:23*, 11322–11326.

Michie, D. (1990). Personal models of rationality. *Journal of Statistical Planning and Inference*, *25*, 381–399.

Muggleton, S., King, R., & Sternberg, M. (1992). Predicting protein secondary structure using inductive logic programming. *Protein Engineering*, 5, 647–657.

Muggleton, S., & De Raedt, L. (1994). Inductive logic programming: Theory and methods. *Journal of Logic Programming*, *19:20*, 629–679.

Nienhuys-Cheng, S., & de Wolf, R. (1997). *Foundations of inductive logic programming*. Berlin: Springer.

Norusis, M. J. (1994). *SPSS: Base system user guide. release 6.0*. 444 N Michigan Ave, Chicago, Illinois 60611: SPSS Inc.

O'Hagan, A. (1999). *Kendall's advanced theory of statistics* (Vol. 2B). London: Arnold.

Provost, F., & Fawcett, T. (1997). Analysis and visualization of classifier performance: Comparison under imprecise class and cost distributions. In *Proceedings of the Third International Conference on Knowledge Discovery and Data Mining (KDD-97)* (pp. 43–48). Menlo Park, CA: AAAI Press.

Provost, F., & Fawcett, T. (1998). Robust classification systems for imprecise environments. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence (AAAI-98)*. Menlo Park, CA: AAAI Press.

Provost, F., & Fawcett, T. (2000). Robust classification for imprecise environments. *Machine Learning*, (to appear). Version available at http://www.stern.nyu.edu/~fprovost.

Quinlan, J. R. (1993). FOIL: a midterm report. In *European Conference on Machine Learning*. (Vol. 667, LNAI, pp. 3–20). Berlin: Springer-Verlag.

Salzberg, S. L. (1997). On comparing classifiers: Pitfalls to avoid and a recommended approach. *Data Mining and Knowledge Discovery, 1*, 317–327.

Scott, M. J. J., Niranjan, M., & Prager, R. W. (1998). Parcel: Feature subset selection in variable cost domains. Technical Report CUED/F-INFENG/TR.323, Cambridge University Engineering Department, Cambridge, UK. Version available at http://svr-www.eng.cam.ac.uk/reports/people/niranjan.html.

Srinivasan, A. (1999). Note on the location of optimal classifiers in n-dimensional ROC space. Technical Report PRG-TR-2-99, Oxford University Computing Laboratory, Oxford.

Srinivasan, A. (2000). A study of two probabilistic methods for searching large spaces with ILP. (Submitted).

Srinivasan, A., & Camacho, R. C. (1999). Numerical reasoning with an ILP program capable of lazy evaluation and customised search. *Journal of Logic Programming, 40:2, 3*, 185–214.

Srinivasan, A., & King, R. D. (1997). Carcinogenesis predictions using ILP. In N. Lavrac, & S. Dzeroski (Ed.) In *Proceedings of the Seventh International Workshop on Inductive Logic Programming (ILP97) LNAI* (Vol. 1297). Berlin: Springer. A version also in Intelligent Data Analysis in Medicine, Kluwer.

Srinivasan, A., King, R. D., & Bristol, D. W. (1999). An assessment of ILP-assisted models for toxicity and the PTE-3 experiment. In S. Dzeroski, & P. A. Flach (Ed.). In *Proceedings of the Ninth International Workshop on Inductive Logic Programming (ILP99) LNAI* (Vol. 1634). Berlin: Springer.

Srinivasan, A., King, R. D., & Bristol., D .W. (1999). An assessment of submissions made to the predictive toxicology evaluation challenge. In *Proceedings of the Sixteenth International Conference on Artificial Intelligence (IJCAI-99)*. Los Angeles, CA: Morgan Kaufmann.

Srinivasan, A., King, R. D., Muggleton, S. H., & Sternberg, M. J. E. (1997) The predictive toxicology evaluation challenge. In *Proceedings of the Fifteenth International Conference on Artificial Intelligence (IJCAI-97)*. Los Angeles, CA: Morgan Kaufmann.

Srinivasan, A., Muggleton, S. H., King, R. D., & Sternberg, M. J. E. (1999) Theories for mutagenicity: A study of first-order and feature based induction. *Artificial Intelligence, 85*, 277–299.

Stuart, A., Ord, K., & Arnold, S. (1999). *Kendall's advanced theory of statistics* (Vol. 2A). London: Arnold.

Walpole, R. E., & Myers, R. H. (1999). *Probability and statistics for engineers and scientists* (2nd Ed.). New York: Collier Macmillan.

Weiss, S. M., & Kulikowski, C. A. (1991). *Computer systems that learn*. San Mateo, CA: Morgan Kaufmann.

Zelle, J., & Mooney, R. (1993). Learning semantic grammars with constructive inductive logic programming. In *Proceedings of the Eleventh National Conference on Artificial Intelligence* (pp. 817–822). Menlo Park, CA: AAAI Press.

Received May 26, 2000
Revised August 29, 2000
Accepted December 20, 2000