Quantitative Pharmacophore Models with Inductive Logic Programming

Ashwin Srinivasan[1], David Page[2], Rui Camacho[3], and Ross D. King[4]

[1] IBM India Research Lab, Block 1, Indian Institute of Technology, New Delhi.
[2] Department of Biostatistics, University of Madison, Wisconsin.
[3] LIACC - CIUP, R. Campo Alegre, 4150 Porto, Portugal.
[4] Department of Computer Science, University of Wales, Aberystwyth.

Abstract. Three-dimensional models, or pharmacophores, describing Euclidean constraints on the location on small molecules of functional groups (like hydrophobic groups, hydrogen acceptors and donors, etc.), are often used in drug design to describe the medicinal activity of potential drugs (or 'ligands'). This medicinal activity is produced by interaction of the functional groups on the ligand with a binding site on a target protein. In identifying structure-activity relations of this kind there are three principal issues: (1) It is often difficult to "align" the ligands in order to identify common structural properties that may be responsible for activity; (2) Ligands in solution can adopt different shapes (or 'conformations') arising from torsional rotations about bonds. The 3-D molecular substructure is typically sought on one or more low-energy conformers; and (3) Pharmacophore models must, ideally, predict medicinal activity on some quantitative scale. It has been shown that the logical representation adopted by Inductive Logic Programming (ILP) naturally resolves many of the difficulties associated with the alignment and multi-conformation issues. However, the predictions of models constructed by ILP have hitherto only been nominal, predicting medicinal activity to be present or absent. In this paper, we investigate the construction of two kinds of quantitative pharmacophoric models with ILP: (a) Models that predict the probability that a ligand is "active"; and (b) Models that predict the actual medicinal activity of a ligand. Quantitative predictions are obtained by the utilising the following statistical procedures as background knowledge: logistic regression and naive Bayes, for probability prediction; linear and kernel regression, for activity prediction. The multi-conformation issue and, more generally, the relational representation used by ILP results in some special difficulties in the use of any statistical procedure. We present the principal issues and some solutions. Specifically, using data on the inhibition of the protease Thermolysin, we demonstrate that it is possible for an ILP program to construct good quantitative structure-activity models. We also comment on the relationship of this work to other recent developments in statistical relational learning.

1 Introduction

The primary goal of the pharmaceutical industry is to find, develop and market new drugs for previously untreatable diseases or which have better properties than existing drugs. The development of a new drug is a time-consuming, expensive process—it can take anywhere from 12 to 16 years from the start of a research program to final approval; and the cost can be up to US$700 million. It is clearly of significant humanitarian and commercial interest to investigate techniques and tools that can assist in making the process more efficient.

Most drugs molecules work by binding to "target sites"—commonly proteins—within the body. By interaction with these targets, drugs can modulate their actions. The process of drug development begins with the selecting an appropriate target with which the drug could interact to modulate disease. In some cases, the detailed structure of the target binding site is known (by the use of X-ray crystallographic techniques) or can be guessed (by structure prediction techniques or from the structure of similar molecules); but this situation is still rare and the identification of potential drugs has to proceed without this knowledge.

Chemistry for drug development (see Fig. 1) is concerned with the identification of "ligands": small molecules that are potential drugs. The search for ligands commences with some compounds that are known to interact with the target. These compounds, or "leads", may be identified in a number of ways, for example from large scale empirical testing of available chemicals. They will have some activity, i.e. ability to interact with the target, but may do so only weakly, and may possess other undesirable properties, for example metabolic instability. Once a set of leads have been isolated, their chemical structure can be refined to improve biological activity and reduce side effects. A computational activity that assists this is concerned with constructing models that relate activity to molecular structure. These "structure-activity relationships", or SARs, describe how the structural differences of a set of leads affects their activity, and can be used to suggest new molecules to make which should have enhanced activity. Chemical synthesis and testing of such molecules leads to the discovery of more active molecules until, ultimately, ligands of the desired activity and properties are discovered. These compounds now begin the long process of development including biological trials for efficacy, safety trials, formulation and advanced testing, patent development and finally, application for clinical trials.

The whole process is critically dependent on effective initial chemistry: proposal of a large number of ligands for development and testing, most of which turn out to be useless is clearly to be avoided. In this, good SAR models can play an important role. "Traditional" SAR methods ("1-D" and "2-D" methods in Fig. 1), generate many features for each lead (effectively, a form of domain-specific propositionalisation) and use statistical modelling tools on the resulting, table containing the feature-values for a set of leads and their activity. The features, or *molecular descriptors*, can be properties of the whole molecule, such as partition coefficient, molecular volume, number of rings, etc., and, if the series of leads share a common core, properties of the substructures at the positions of variation The modelling task is to construct a predictive model that relates

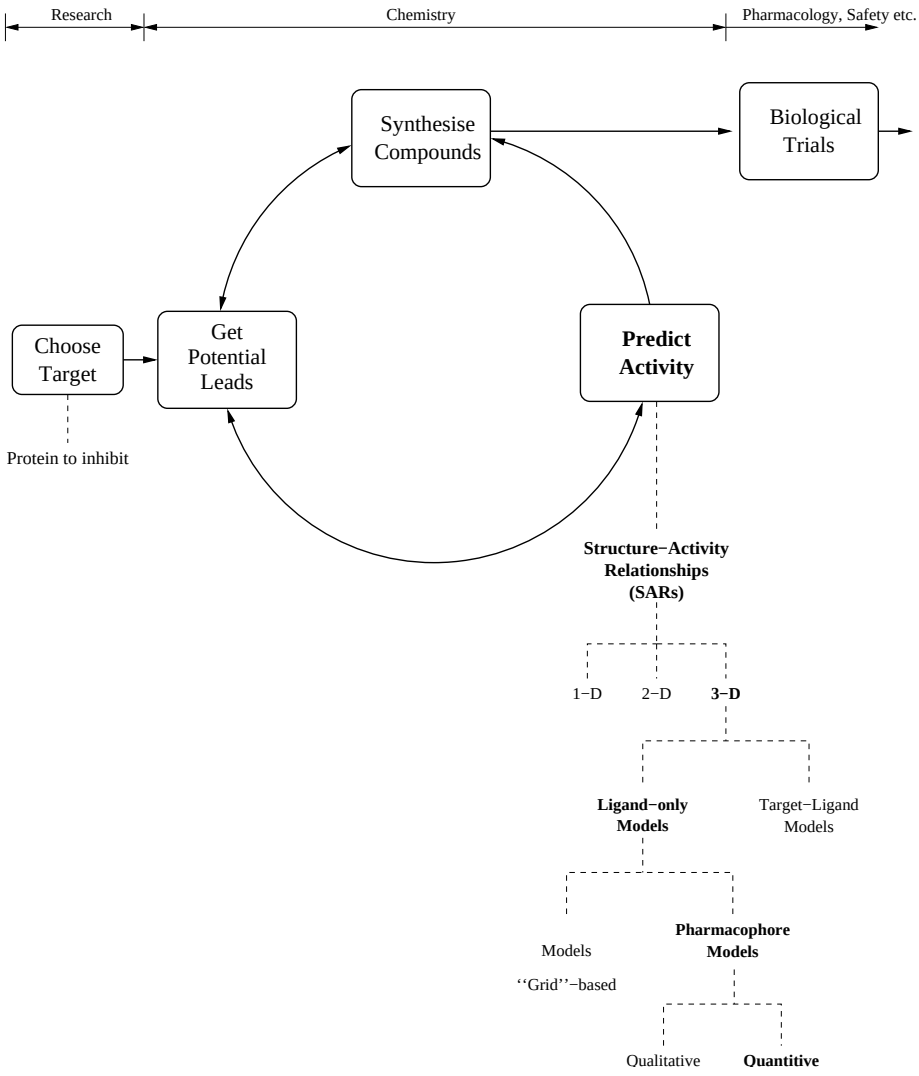

Fig. 1. A simplified view of the chemistry involved in the drug-design process (adapted from a figure kindly provided by Dr. Stuart Green, University of Leeds). The focus of this paper is shown in bold-face.

feature-values to activity. Such models have been widely applied, and there are many examples of successful SAR analyses [12]. There are four principal difficulties with this approach. First, much depends on the ability to identify molecular descriptors that retain all the information necessary for obtaining a good model. Second, constructing reliable models is not easy. The molecular descriptors can number into several millions, and special care must be taken avoid chance correlations. Third, the models are not always easy to use in refining leads. It is easy to calculate the value of a molecular descriptor for a given lead (for example, an electronic partial charge), but quite a different matter to design a molecule that will possess that value. Finally, by dealing with bulk molecular properties, rather than a more explicit representation of the molecules, no account is taken for the three-dimensional shape of the leads.

Leads, being small molecules are flexible and can adopt different shapes (conformations) by torsional rotations about bonds within the molecule. Molecules rapidly convert from one conformation to another, so that no single conformation can be isolated and tested for a given activity. Hence in general one does not know *a priori* which conformation of a molecule is an active conformation for a given form of activity. Computational chemists usually employ one of two approaches to this problem. In the first, the shape and electrostatic interactions are dealt with via a calculated interaction with a "probe" atom or group at points on a three-dimensional grid which surrounds the molecules. Statistical methods are then used to identify those parts of the molecule which are responsible for activity. The analysis can be displayed graphically to aid in the design of new molecules. The major disadvantage with this approach is that, in order to compare values at the calculated points between molecules, the conformation of the molecules, and a common coordinate frame, or alignment, must be chosen in advance of the analysis. This is equivalent to deciding on the manner in which the molecules interact with the target. If the molecules contain a large common structural element this alignment may be straightforward, but this is often not the case.

The second approach uses a representation of biological activity called a pharmacophore. This is an abstraction of the molecular structure to the, usually, small number of key features which contribute the majority of the activity, together with their geometric arrangement represented by pairwise distances (see Fig. 2, from [8]). An advantage of the pharmacophore representation is that it expresses biological activity in a language that is familiar to chemists within the pharmaceutical industry. These representations are also readily convertible into search queries of compound databases, which is an effective means of identifying additional active molecules [9].

That Inductive Logic Programming (ILP) is particularly well-suited to represent and discover SARs has been argued persuasively elsewhere [8, 20]. The most important reasons given are: (i) the comprehensibility of the models constructed allow an easy translation of the models into new chemical structures; (ii) ILP programs do not need to align the leads to a common spatial reference frame; and (iii) the first-order representation used by ILP naturally deals with

Fig. 2. A schematic diagram of how a "pharmacophore" definition (right) is related to a lead-target interaction (from [8]). The pharmacophore contains the key functional interactions and the geometric relationships between them expressed as distances d1, d2, and d3. For example, the functional groups may be a hydrogen donor and two hydrogen acceptors; d1 might be 4.5 Angstroms, d2 might be 5.0 Angstroms, and d3 might be 3.75 Angstroms.

difficulties arising from the fact that the leads may assume several shapes (or conformations), not all of which may be responsible for the biological activity. Non-relational learning algorithms—those that need a feature vector representation of data—require modification to handle point iii. Furthermore, given even just two shapes for each molecule, these approaches either require worst-case exponential time (in the number of molecules) to perform the alignment mentioned in point ii, or must at times work with an incorrect alignment—one in which the parts of the molecules responsible for interaction are not aligned with each other. Regarding point i, comprehensibility, the most complex SARs found by ILP using only the structures of leads and their activities are pharmacophore models in the form of rules like the following:

Molecule M is "active" if:
M has a hydrogen donor at position P1 and
M has a hydrogen acceptor at position P2 and
M has a hydrogen acceptor at position P3 and
the distance between P1 and P2 is 4 ± 1 Å and
the distance between P1 and P3 is 3 ± 1 Å and
the distance between P2 and P3 is 5 ± 1 Å.

Here, positions P1, P2, and P3 are points in 3-dimensional space and the rule represents a "3-point pharmacophore". The rule, evidently simple to understand, nevertheless highlights an important shortcoming of SARs constructed by ILP: they are classificatory in nature and unable to quantify their predictions of ac-

tivity[1]. In this paper, we investigate identification by ILP of two kinds of quantitative SARs, namely, class-probability models of the form:

> The probability of molecule M being "active" is P if:
> M has a hydrogen donor at position P1 and
> M has a hydrogen acceptor at position P2 and
> M has a hydrogen acceptor at position P3 and
> the distance between P1 and P2 is X1 and
> the distance between P1 and P3 is X2 and
> the distance between P2 and P3 is X3 and
> P is the probability of being "active" given X1, X2, X3.

and regression models of the form:

> The affinity of molecule M is A if:
> M has a hydrogen donor at position P1 and
> M has a hydrogen acceptor at position P2 and
> M has a hydrogen acceptor at position P3 and
> the distance between P1 and P2 is X1 and
> the distance between P1 and P3 is X2 and
> the distance between P2 and P3 is X3 and
> A is the expected affinity given X1, X2, X3.

We will call these "quantitative pharmacophore models": specifically, the rules above are 3-point quantitative pharmacophore models. The reason for interest in the regression model is evident: when accurate quantitative information of lead-activity is available, SARs that relate lead-structure to activity are most useful. Even when we only have categoric information about the lead-activity (for example, leads are "active" or "inactive"), class-probability models can still be extremely useful for the reasons below:

- Inherent uncertainties in the domain may make categoric classification difficult. For example, data may be laboratory measurements obtained from an imprecise assay;
- Applications involving decision-making often require probability estimates for use in cost/benefit calculations. For example, a synthesis of a particular molecule may proceed only if there was a very high probability of it being biologically active;
- It may be important to rank alternatives with a given class value. For example, laboratory constraints may require synthesis of molecules to proceed in small batches. An ordering on molecules predicted to be active is then needed.

[1] These classificatory rules have been used to construct boolean features, which have then be used by regression techniques to build quantitative models for activity [21, 37]. We are concerned here with constructing quantitative models solely with the use of ILP.

In this paper, both the geometric constraints and the "numeric" constraint in the rules above (that is, computing P or A given X1, X2 and X3) are to be obtained during hypothesis construction by an ILP program. It is our intention to investigate the use of statistical procedures as background knowledge for obtaining the numeric constraint. The following questions—*novel for both ILP and statistical methods*—arise for rules above:

Model Identification. Molecule M may have several hydrogen donors and acceptors, at different positions, resulting in different sets of values for X1, X2 and X3[2]. The novelty for ILP is that the domain dictates only one of these sets is relevant for identifying the the appropriate numeric constraint. Which one?

Prediction. Each set of values for X1, X2 and X3 results in a prediction for P or A. Which of these values should be returned[3]? This question has not arisen previously in either ILP or statistical methods.

These two tasks are unlike those confronted by a statistician during normal discourse. There it is common, due to random variation, to obtain *different* affinity values for repeated measurements of the *same* values for X1, X2 and X3. Machine learning researchers will recognise the questions here as consequences of a multiple-instance representation [6]. Novel here, however, is the real-valued prediction task in a multiple-instance setting. While this combination has been addressed before, in two papers [32, 1], as discussed in the next paragraph an adequate solution to model identification and prediction in this setting has not been found. And the combination of real-valued prediction in a multiple-instance setting has not been addressed explicitly before in ILP, although the paper by Ray and Page [32] pointed to its importance.

The approaches proposed in the two papers already mentioned performed well on *model identification* from *synthetic data*. Note that model identification is easy to assess on synthetic data, where the data generator can record the model it used to generate the real-valued response for each data point. Model identification is difficult or impossible to assess on real-world data, where one sees only the response value and not the correct model, or set of bindings for variables. Hence one must rely on *prediction* to assess performance on most real-world tasks. The approaches in the two papers have not been demonstrated to perform well on real-world tasks, such as predicting quantitative drug activity values.

The present paper presents procedures to address both model identification and prediction, and it tests these procedures on real-world data for quantitative drug activity prediction. For illustrative purposes, we will use linear regression as

[2] The logic programmer will recognise this as different substitution-sets arising from a non-determinate definition for the donor and acceptor predicates.

[3] We could consider using some representative set of values for distances (for example, the average) for both model identification and prediction. However, what this representative should be may not be apparent and in any case, will require a different rule to the ones shown.

a background predicate to perform quantitative prediction, although the procedures proposed are not restricted to that particular modelling technique (as will be demonstrated in Section 6.1). The paper is organised as follows. In Section 2 we present a simple example to illustrate the principal issues. Section 3 introduces relevant terminology from the literature on multiple-instance learning. "Solutions" to the problems that arise during model identification and prediction are in Sections 4 and 5 respectively. Application of the procedures developed to quantitative structure-activity relations is in Section 6. Section 7 concludes the paper. The paper is accompanied by two appendices. Appendix A describes the statistical procedures used in the empirical study. Appendix B relates the work in Section 6.1 to other, more general, work on incorporating probabilities in ILP and relational learning generally.

2 An Example

Consider learning the following simple 1-point quantitative pharmacophore model using an ILP system:

> The affinity of molecule M is Y if:
> M has a hydrogen donor at position X and
> Y = m X + c.

Here "Y = m X + c" is a statistical model in which m and c are parameters to be estimated and Suppose data available are as follows: (1) affinity values of 5 different molecules; and (2) records of hydrogen donor locations on each molecule (a molecule can have more than one donor). The data are tabulated in Fig. 3. Logically speaking, the rule above states that the affinity of the molecule is a linear function of one of the hydrogen donor positions (without specifying which one: the logician will recognise this as a consequence of the variable X being existentially quantified within the rule body).

Since for each N, it is not apparent which of the X values are to be paired with the corresponding Y, a combinatorial problem arises[4]. For this simple problem, the combinatorics are manageable: there are only $3 \times 1 \times 2 \times 1 \times 2 = 12$ different tables containing exactly 1 entry for each of the 5 values of X. Two such tables are shown in Fig 4.

Parameter estimates can now be obtained using each such table. The estimates returned are those that result in the best fitting model. For the data in Fig 4, the best fitting model Y = 2.572 X - 33.155 is obtained with the table in Fig 5. The resulting rule is therefore:

> The affinity of molecule M is Y if:
> M has a hydrogen donor at position X and
> Y = 2.572 X - 33.155.

[4] Some theoretical results known for multiple-instance learning can be found in [7].

Mol (M)	Aff. (Y)	Donor (X)
		25
1	50	33
		29
2	100	50
3	150	73
		75
4	200	90
5	250	120
		110

Fig. 3. "Training" data for the problem of predicting affinity Y using hydrogen donor location X. For a given value of N, it is known that the value of Y depends on one of X values—but it is not known which one.

Mol (N)	Aff. (Y)	Donor (X)
1	50	25
2	100	50
3	150	73
4	200	90
5	250	120

Mol (N)	Aff. (Y)	Donor (X)
1	50	33
2	100	50
3	150	75
4	200	90
5	250	110

Fig. 4. Example tables obtained by combinatorial enumeration of the data in Fig. 3. The table on the left yields the model Y = 2.161 X - 4.739, with sample correlation coefficient 0.997. The one on the right yields Y = 2.565 X - 33.677, with sample correlation coefficient 0.998.

Section 4 describes procedures that equips an ILP system to reach such a model (or at least, an approximation to it) given the data in Fig 3.

Now consider using this rule on the data shown in Fig. 6. It is evident that simply executing the rule above will yield two predictions: Y = 188.037 (using X = 86) and Y = 203.469 (using X = 92). Which of these predictions should be returned is the problem posed by (Q2). In Section 5, we examine some solutions to this problem.

3 Terminology

It is convenient at this point to introduce some terminology from the machine learning literature on multiple instance learning. The following statement of the multi-instance learning task is largely from [32]. Data consists of a set of n bags ($1 \leq n < \infty$). The i^{th} bag consists of m_i instances ($1 \leq m_i < \infty$) and a label y_i (which may be nominal or real-valued). Instance j of bag i is described by a d-dimensional vector of values ($1 \leq d < \infty$). The task is to construct a model

Mol (N)	Aff. (Y)	Donor (X)
1	50	33
2	100	50
3	150	73
4	200	90
5	250	110

Fig. 5. Table that results in the best fitting linear model relating Y and X values. The best fitting model is Y = 2.572 X - 33.155, with sample correlation coefficient 0.999.

Mol (N)	Aff. (Y)	Donor (X)
6	?	86
		92

Fig. 6. "Test" data. The task is to predict the affinity of molecule 6 given its hydrogen donor locations.

predicting the y values. For each bag i, it can be assumed that one of the m_i instances is sufficient for predicting y_i.

It should be evident that the standard statistical task of predicting a dependent variable, given values for independent variables is a special case of the multi-instance task as presented here (with one instance per bag). The formulation here is slightly different to the original description of multi-instance modelling in [6], which only deals with binary classification problems (that is, the y_i take one of two values: "positive" or "negative"). There, bags are treated asymetrically. If i is a positive bag, it is assumed that at least one of the m_i instances is positive (that is, a single instance is sufficient for predicting y_i). If i is a negative bag, it is assumed that *all* the m_i instances are negative (that is, all instances are necessary for predicting y_i). We have found this distinction to be unnecessary as the same effect is achieved by using m_i different "negative" bags, each with a single instance.

In Fig. 3 therefore, there are 5 bags (one for each molecule). Bag 1, with y value 50, has 3 instances; Bag 2, with y value 100, has 1 instance and so on.

4 Model Identification

Model identification requires the following:

1. All values of relevant variables required by the statistical procedure are to be collected (in the example introduced earlier, this corresponds to all the values of N, Y and X in Fig. 3).
2. A search for the best statistical model is then to be conducted using subsets of values collected in Step 1 (each such subset results in a table of the type in Fig. 4).

An ILP system capable of "lazy evaluation" (as described in [36]) is sufficiently powerful to allow this form of model identification. For a predicate specified as being lazily evaluated, certain arguments are annotated as being functionally dependent on others. We can loosely term the former "output" and the latter "input" arguments. The actual values of the output arguments to be used during normal execution of the literal are computed as follows:

(a) The ILP system collects all possible values of the input arguments.
(b) The ILP system assumes that the background knowledge contains a definition specifying the computation of the output arguments as a function of the values collected above. This definition is then used to compute the values of output arguments.

Without entering into implementation details, it suffices to state here that (a) is sufficient to achieve the requirement of Step 1 above; and the definition in (b) can be an implementation of the search procedure required in Step 2.

This still leaves open the actual search procedure to be used. An exhaustive strategy—as was used in the simple example in Section 2—although optimal, may not be tractable. An efficient, but approximate, procedure modelled on the Expectation-Maximization (EM) algorithm was proposed by Ray and Page [32] specifically for use in multi-instance linear regression. The same procedure can, in fact, be adapted for use with any modelling procedure that returns some estimate of the error in fitting a model to (a table of) data. The modified procedure is shown in Fig. 7; and a single execution of the outer loop (Step 4) for the example in Section 2 is in Fig. 8.

Provided model construction in Step 14 terminates, it is easy to show that the entire procedure terminates (the procedure contains two loops, bound by finite constants R and T). The run-time complexity of the procedure primarily depends on the complexity of the model construction in Step 14 and the computation of "closest" instances in Step 11 (the error calculation in Step 12 can in fact be done in conjunction with Step 11). Statistical model construction procedures are usually $O(n)$ algorithms (and rarely worse than quadratic). For many reasonable error measures, Steps 11–18 are usually $O(d \cdot m \cdot n)$ where $m = max_i(m_i)$. There is no guarantee that the model returned will be optimal.

In summary, we propose that an ILP system capable of lazy evaluation, using the procedure in Fig. 7 can address the two basic requirements of model identification presented at the outset of this section. The reader will note that the procedure returns the model *and* the instances used to obtain these estimates: the reason for this will become apparent in the following section. Thus, in our example, rules returned will be of the form:

> The affinity of molecule M is Y if:
> M has a hydrogen donor at position X and
> Y = m X + c using instances I with error E.

In the sections that follow, we may, on occassions omit references to I, or E, or both.

Input: integers R and T ($1 \leq R, T < \infty$), and n bags ($1 \leq n < \infty$) where bag i contains a set of d-dimensional vectors ($1 \leq d < \infty$).
Output: A model M, the instances I used to construct the model, and its error E

01. $m_{best} \leftarrow$ a randomly constructed model
02. $i_{best} \leftarrow \emptyset$
03. $e_{best} \leftarrow \infty$
04. **for** $r = 1 \ldots R$
05. $\quad m_0 \leftarrow$ a randomly constructed model
06. $\quad i_0 \leftarrow \emptyset$
07. $\quad e_0 \leftarrow \infty$
08. $\quad t \leftarrow 1$
09. $\quad done \leftarrow false$
10. $\quad$ **while** (not done **and** $t \leq T$)
11. $\quad\quad i_t \leftarrow$ the "closest" instances in the n bags, given m_{t-1}
12. $\quad\quad e_t \leftarrow$ the error of m_{t-1} on i_t
13. $\quad\quad$ **if** ($e_t < e_{t-1}$)
14. $\quad\quad\quad m_t \leftarrow$ the model constructed with instances i_t
15. $\quad\quad\quad t \leftarrow t + 1$
16. $\quad\quad$ **else**
17. $\quad\quad\quad done \leftarrow true$
18. $\quad\quad$ **endif**
19. $\quad$ **endwhile**
20. $\quad$ **if** ($e_{t-1} < e_{best}$)
21. $\quad\quad m_{best} \leftarrow m_{t-1}$
22. $\quad\quad i_{best} \leftarrow i_{t-1}$
23. $\quad\quad e_{best} \leftarrow$ the error of m_{best} on i_{best}
24. $\quad$ **endif**
25. **endfor**
26. **return** $m_{best}, i_{best}, e_{best}$

Fig. 7. An EM-like procedure for model identification with multi-instance data. The actual model construction procedure (Step 14) is deliberately left unspecified, as is the meaning of "closest" in Step 11. It is normally acceptable for numerical prediction to adopt some quadratic loss function (for example, squared difference between predicted and actual value) to measure error. The same measure is then used to determine closeness.

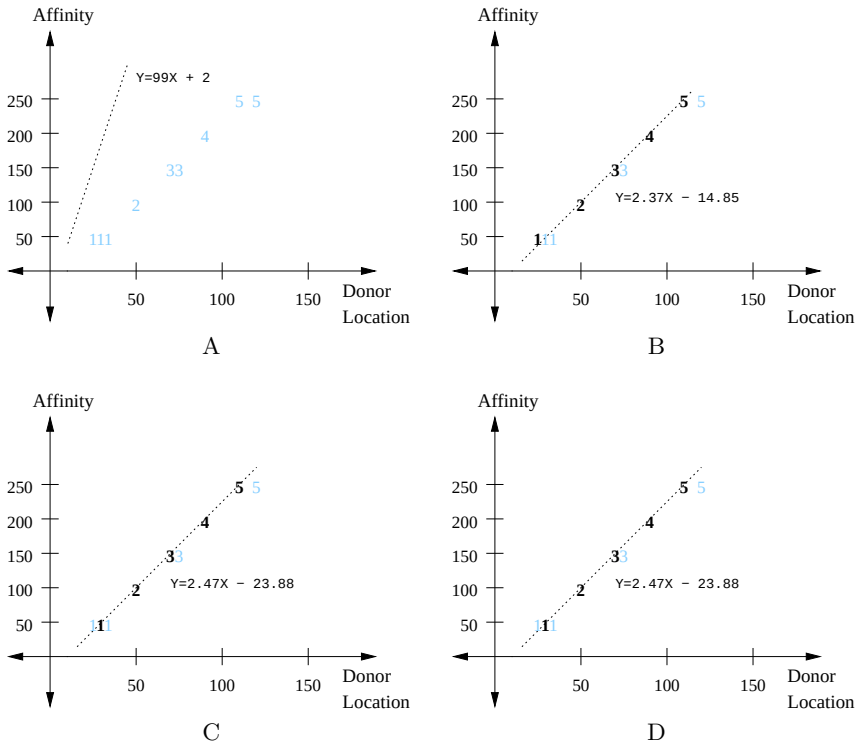

Fig. 8. A single iteration of the outer loop of the procedure in Fig. 7 with the data in Fig. 3. Points are labelled by the molecule number (given by the "Mol" column in Fig. 3) which constitute the bags. Thus, the three points "1 1 1" in the Graph A correspond to the 3 donor-affinity pairs constituting molecule 1. Initially a random model is constructed (here $Y = 99X + 2$ in Graph A) and conventionally assigned an infinite prediction error. The "closest" instance from each bag is selected—these are shown highlighted—and the prediction error using these instances determined. If this has a a lower prediction error, then a new model is constructed using the instances selected (Graph B). The process is repeated until there is no change in the errors of models constructed (Graphs C and D). The model returned in the iteration shown is $Y = 2.47X - 23.88$. A better fitting model exists ($Y = 2.57X - 33.15$) and may be found on a different iteration.

5 Prediction

In Section 1, it was illustrated how multiple data instances can result in several predictions. The reason for this is that the example rules shown there all act as "instance-level" predictors. However, what is usually needed is a "bag-level" predictor (we want a single prediction for the molecule's affinity, for example). Two solutions suggest themselves:

1. *Domain-independent prediction.* For example, arbitrarily select a single instance and return the prediction based on that instance; or
2. *Domain-dependent prediction.* For example, compute the bag-level prediction using some meaningful function of the set of instance-level predictions (this could be, for example, some representative value like the median).

We will consider each of the solutions in turn. For clarity, we will assume in each case that the procedure in Fig. 7 has returned the following rule (this is the one with lowest mean-squared error):

The (instance-level) affinity of molecule M is Y if:
M has a hydrogen donor at position X and
Y = 2.572 X - 33.155 using instances I.

5.1 Domain-independent prediction

Implementing the domain-independent solution simply requires an additional constraint be included in the rule(s) obtained after the model identification step to convert it to a bag-level predictor. For example:

The (bag-level) affinity of molecule M is Y if:
M has a hydrogen donor at position X and
Y = 2.572 X - 33.155 using instances I and
X is the best instance given I.

where the condition "X is the best instance given I" is true for just one of the different possible values of X. But how is this selection to be achieved? First-order regression implemented in [18] and [36] both pick the first value of X (which is tantamount to random selection). This solution, which is easy to implement, can lead to counter-intuitive results. Consider, for example, using the rule above to predict Y-values for the training data in Fig. 3. Recall that the sample correlation of the model was 0.999. The bag-level predictions using the "choose first instance" option are shown in Fig. 9. The correlation coefficient now is 0.997, and we are in the unusual position of being unable to reproduce our performance on data that was used to construct the model in the first place.

The difference arises, clearly, because we cannot guarantee that the "choose first instance" rule will select the same instances as were used to estimate the

Mol (N)	Actual Affinity	Predicted Affinity	Instance Used (X)
1	50	31.145	25
2	100	95.445	50
3	150	154.601	73
4	200	198.325	90
5	250	249.765	110

Fig. 9. Actual and predicted values of affinity using the first instance in each bag in Fig. 3. Predictions are done using the equation Y = 2.572 X - 33.155.

parameters. A more informed choice is possible with the actual instances I used for parameter estimation. The "best" instance could then be selected as the first one that is within some pre-specified distance (tolerance) to an instance in I (in a manner reminiscent of "lazy" or instance-based learners [23]). With a sufficiently small tolerance (0, for example) we can ensure performance on the training data at least will be reproducible. With a sufficiently large tolerance (∞, for example), the constraint's behaviour degenerates to that of choosing the first instance. Superficially, using such a constraint appears related to the approach in [22] where the probability of a class label for new data point depends on the proximity of that data point to training instances. For new data, instances selected will be those that within some tolerance of the instances used for parameter estimation. It remains an open question as to what should be done if none, or several, of the new instances satisfy this constraint.

5.2 Domain-dependent prediction

Implementing this solution requires some domain-specific knowledge. Suppose the domain dictated that of all instance-level predictions, the highest one was most appropriate. One way to obtain this is to use something along the lines of the following:

The (bag-level) affinity of molecule M is Y if:
    The (instance-level) affinities of molecule M is the set Ys and
    Y is the highest value in Ys.

The (instance-level) affinities of molecule M is the set Ys if:
    Ys is the set of values Yi s.t.
        the (instance-level) affinity of molecule M is Yi.

where, as before:
The (instance-level) affinity of molecule M is Y if:
    M has a hydrogen donor at positions X and
    Y = 2.572 X - 33.155.

In the experimental work described in the next section, we will be using this form of domain-dependent prediction.

6 Quantitative Pharmacophore Models for Thermolysin Inhibition

Our test-bed for the construction of quantitative pharmacophore is the inhibition of Thermolysin. Thermolysin is a zinc-containing protease that consists of two spherical domains separated by a deep cleft that constitutes the active site. Zinc-containing proteases like Thermolysin play an important role in physiological processes like digestion and blood pressure regulation. Data on a number of inhibitors of Thermolysin are readily available in the literature: we use the molecules studied by [21]. The data consist of crystal structures and corresponding activity values (pKi = -logKi) of 31 inhibitors.

All experiments use the ILP system Aleph [35] (specifically, Aleph Version 4). The experiments were performed on machine equipped with two 512 Mhz Pentium III processors with 128 megabytes of random access memory. We follow the work of [8] in providing the ILP system with background knowledge of the following:

- *Compound-specific knowledge.* This is in the form of the atom and bond structure of each compound, as well as its 3-dimensional conformation (for each of the three lowest energy conformers identified by [21]). This information is represented by first-order atomic formulae: we refer the reader to [8] for examples of this encoding.
- *General chemical and geometric knowledge.* Generic chemical knowledge provided is in the form of a library of elementary chemical concepts (for example definitions of hydrogen donors, hydrogen acceptors, hydrophobic groups, esters, ethers, etc.). We have provided the same concepts as those in [21] (the authors there have used 39 such concepts). The only geometric knowledge needed for this task is a procedure for calculating the Euclidean distance between two points.

Additional background knowledge required for the construction of class-probability and regression models will be described below.

6.1 Class-probability models

It is our goal in this section to demonstrate that an ILP system capable of using the procedure in Fig. 7 and statistical procedures for conditional probability estimation can construct class-probability models for structure-activity prediction. Recall that these models were of the form:

The probability of molecule M being active is P if:
M has a hydrogen donor at position P1 and
M has a hydrogen acceptor at position P2 and
M has a hydrogen acceptor at position P3 and
the distance between P1 and P2 is X1 and
the distance between P1 and P3 is X2 and

the distance between P2 and P3 is X3 and
P is the probability of being active given X1, X2, X3

Candidate statistical procedures for computing P considered are logistic regression and "naive" Bayes (see Section A.1). The reasons for selecting these techniques are:

- They are simple;
- There is abundant support for their use in class probability estimation, both in the statistical and machine learning literature (see for example, [29]). In addition, it can be shown that the logistic function is the appropriate Bayesian choice under some fairly general conditions (see [17]); and
- They can be seen as special cases of more general work on combining Bayesian networks and ILP (see Appendix B).

Data Of the 31 Thermolysin inhibitors available, we have designated the top 15 inhibitors to be be "active" and the remaining 16 inhibitors to be "inactive". In order to build class-probability models, the actual examples are of the form "probability of molecule m being active is p", where p is one of 1.0 or 0.0 (depending on whether m is active or inactive).

Additional background knowledge In addition to the background information described earlier we also include the following:

- *Constraints on legitimate models.* Legitimate models are required to contain functional groups, pairwise distances between the groups and a numeric constraint that predicts the probability of being active.
- *Model identification.* This definition implements the procedure described in Fig. 7. The actual model construction within this procedure is done by one of logistic regression or naive Bayes.

Method Our method is straightforward. Using background definitions of logistic regression or naive Bayes:

1. Construct the "best" class-probability model for the largest pharmacophore possible for the inhibitors.
2. Estimate the performance of the model obtained in the previous step.

The following details are relevant:

(a) Looking for the largest pharmacophore model for a set of active molecules is a characteristic of this kind of domain (see [8] for more details). Models can be seen as containing two sorts of constraints: first, those in the "prefix", consisting of the functional groups and their pairwise distances (the pharmacophore *per se*); and second, the numeric constraint that predicts probabilities using the distances between these functional groups. In Step

(1) we restrict models to be a single clauses. This is similar to [8] and offers the most comprehensible models for activity). Further, the search for this model is done in 2 steps:(i) the largest prefixes common to all inhibitors are found; and (ii) each prefix is extended to include the numeric constraint. For the Thermolysin data, the largest pharmacophore models contain 4-point pharmacophores (that is, there is no single clause 5-point pharmacophore model that can be used to explain the activity of all of the 31 inhibitors). The search procedure in Fig. 7 searches for the "best" numeric constraint by minimising a loss-function. Here we will use a standard quadratic loss function (sum of squared differences between actual and predicted probabilities) summed over all training instances.

(b) In Step (2) performance estimates will be estimated using a leave-one-out procedure. That is, each inhibitor is, in turn, categorised as a "test" instance. A class-probability model is constructed using the remaining instances and then used to predict the probability of the test instance being active. The non-determinate nature of the domain will result in a set of predictions for the probability. We have elected to select the highest of these as the final prediction (as described in Section 5.2). This is a reasonable choice for the domain considered, as we are interested in the best chance of a molecule being able to inhibit Thermolysin (if the set of predictions is empty, then the test instance is not included in estimating the performance). The performance is summarised by an ROC curve, obtained by changing the threshold probability above which the test instance is to be classified as being active (for example, if this threshold is 0.3, and the model predicts the probability of the test instance being active as being 0.2, then the test instance is classified as being inactive).

Results Figure 10 shows ROC curves generated by changing threshold probabilities in the manner described above. The curves are, in fact, the convex hull of the points denoting classificatory models obtained from the corresponding class-probability models (it has been shown elsewhere [31] that the convex hull contains the optimal classifiers under some very general decision-theoretic conditions).

The curves clearly show how we can obtain higher positive prediction rates at the expense of an increased false-positive rate: obtaining a set of models with such properties is straightforward once we have class-probability models. For false-positive rates above 0.2, the ROC curve obtained with naive Bayes dominates that with the logistic regression. That is, as long as a false alarm rate of at least 20% is tolerable, the use of naive Bayes as a background predicate results in better predictive performance. Otherwise, the use of logistic regression yields a better model. The position is more clear-cut on the comprehensibility front: the use of logistic regression results in a simple equation for predicting probability. In contrast, the when using the naive Bayes procedure, probabilities are kernel-density estimates obtained from the training instances. This makes the corresponding rule returned by the ILP program significantly harder to un-

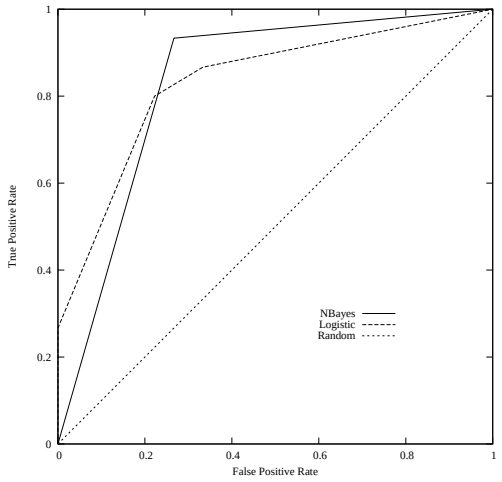

Fig. 10. ROC curves for Thermolysin Inhibition. "NBayes" is the curve obtained using a naive Bayesian procedure for probability estimation. "Logistic" is the corresponding curve using logistic regression as background knowledge. "Random" is the performance of a classifier that assigns class-values using a coin toss.

derstand. Example rules obtained with logistic regression and naive Bayes are shown in Figures 11 and 12.

6.2 Regression models

The structure-activity relations constructed so far do not include predictions of actual binding affinities. This is usually harder and in this section we demonstrate that an ILP system capable of using the procedure in Fig. 7 and appropriate statistical procedures for conditional mean estimation can construct a regression model that predicts binding affinities. Recall that an example is a rule of the form:

The affinity of molecule M is A if:
M has a hydrogen donor at position P1 and
M has a hydrogen acceptor at position P2 and

The probability of molecule M being "active" is P if:

M has a hydrogen donor at position P1 and
M has a hydrogen donor at position P2 and
M has a hydrogen donor at position P3 and
M has a hydrogen donor at position P4 and
the distance between P1 and P2 is X1 and
the distance between P1 and P3 is X2 and
the distance between P1 and P4 is X3 and
the distance between P2 and P3 is X4 and
the distance between P2 and P4 is X5 and
the distance between P3 and P4 is X6 and

$$\text{P} = \frac{1}{1+e^{-87.5+16.1X1-6.4X2+0.1X3-36.3X4+43.8X5-80.1X6}}$$

Fig. 11. An example of a class-probability model obtained with the use of logistic regression. The regression procedure computes the numbers in the equation for P.

M has a hydrogen acceptor at position P3 and
the distance between P1 and P2 is X1 and
the distance between P1 and P3 is X2 and
the distance between P2 and P3 is X3 and
A is the expected affinity given X1, X2, X3

Once again our choice of statistical procedures—linear and kernel regression (see Section A.2)—is primarly based on simplicity and prevalance of use.

Data Data consists of the 31 Thermolysin inhibitors along with their activity (pKi) values. Examples are thus of the form "the affinity of molecule m is y" where y is some floating-point number.

Additional background knowledge In addition to the background information described earlier we also include the following:

- *Constraints on legitimate models.* Legitimate models are required to contain functional groups, pairwise distances between the groups and a numeric constraint that predicts the affinity.
- *Model identification.* This definition implements the procedure described in Fig. 7. The actual model construction within this procedure is done by one of linear or kernel regression.

Method Our method is similar to that used for the construction of class-probability models. That is, using background definitions of linear or kernel regression:

1. Construct the "best" regression model for the largest pharmacophore possible for the inhibitors.

The probability of molecule M being "active" is P if:
M has a hydrogen donor at position P1 and
M has a hydrogen donor at position P2 and
M has a negatively charged atom at position P3 and
M has a negatively charged atom at position P4 and
the distance between P1 and P2 is X1 and
the distance between P1 and P3 is X2 and
the distance between P1 and P4 is X3 and
the distance between P2 and P3 is X4 and
the distance between P2 and P4 is X5 and
the distance between P3 and P4 is X6 and
P is the kernel density estimate using instances I

Where I is:

P	X1	X2	X3	X4	X5	X6
0.0	5.6	2.2	4.3	4.5	2.2	4.1
0.0	5.3	2.2	4.8	5.7	2.2	5.3
...	...	...	...	...	...	...
...	...	...	...	...	...	...
...	...	...	...	...	...	...

Fig. 12. An example of a class-probability model obtained with the use of naive Bayes. The entries in the tabulation are the training instances returned by the search procedure in Fig. 7.

2. Estimate the performance of the model obtained in the previous step.

The following details are relevant:

(a) Models will again contain two sorts of constraints: first, those in the "prefix", consisting of the functional groups and their pairwise distances (the pharmacophore *per se*); and second, the numeric constraint that predicts affinity using the distances between these functional groups. As before, we restrict models to be a single clause and the search is done in 2 steps:(i) the largest prefixes common to all inhibitors are found; and (ii) each prefix is extended to include the numeric constraint. We use the standard quadratic loss function (sum of squared differences between actual and predicted affinities) summed over all training instances.
(b) For kernel regression, we use the Epanechnikov quadratic kernel [14] whose window size is determined by the k^{th}-nearest neighbour (with $k = 3$). These choices are arbitrary, although the Epanechnikov kernel has some optimality properties that are described in [13].
(c) In Step (2) performance estimates will be estimated using a leave-one-out procedure. That is, each inhibitor is, in turn, categorised as a "test" instance. A regression model is constructed using the remaining instances and then used to predict the affinity of the test instance. The non-determinate nature of the domain will result in a set of predictions for the affinity. As with class-probability models, we have elected to select the highest of these as

the final prediction (and as before, if the set of predictions is empty, then the test instance is not included in estimating the performance). Spearman's rank correlation coefficient (r_{CV}) between actual and predicted values will be taken to be representative of the performance of the model.

Results Figure 13 shows the correlation coefficients obtained by the leave-one-out procedure described. Also tabulated are the best results obtained with the standard QSAR 3-D technique CoMFA (comparative molecular field analysis) and a two-stage approach by King and colleagues ([21]).

Method	r_{CV}
Linear	0.34
Kernel	0.68
CoMFA	0.78
King	0.86

Fig. 13. Leave-one-out estimates of the rank correlation between actual and predicted values of affinity. "Linear" and "Kernel" stand for linear and kernel regression respectively. "CoMFA" represents the best QSAR model obtained with comparative molecular field analysis. "King" represents the approach described in [21].

The models obtained with linear and kernel regression are not as good as those with CoMFA or King. This is not surprising, given the models constructed here—single clauses with a regression constraint—are very simple (in contrast, King uses a kind of voting with multiple pharmacophore models). Nevertheless, the performance using linear regression is particularly poor, suggesting that the assumption of a linear model may be inappropriate. There are also good biological reasons to believe that this may be the case: assuming an ideal binding geometry, affinity values should decrease as distances deviate (on either side) from the ideal distances between functional groups. Quadratic regression should yield a better model under these circumstances: this is confirmed by an improved r_{CV} value of 0.55. This is, of course, still well short of the mark achieved by King: some portion of the blame appears to lie in the selection rule used to obtain a final prediction (that is, the highest value of all predictions obtained). Better correlation values are obtainable, but the selection rule is not evident *a priori*.

Example rules obtained with linear and kernel regression are shown in Figures 14 and 15.

7 Concluding Remarks

The process of developing a new drug is long, laborious and expensive. A large part of the time and effort goes into the testing and assesment of compounds that

The affinity of molecule M is A if:
M has a hydrogen donor at position P1 and
M has a hydrogen acceptor at position P2 and
M has a hydrogen donor at position P3 and
M has a negatively charged atom at position P4 and
the distance between P1 and P2 is X1 and
the distance between P1 and P3 is X2 and
the distance between P1 and P4 is X3 and
the distance between P2 and P3 is X4 and
the distance between P2 and P4 is X5 and
the distance between P3 and P4 is X6 and
A = 1.06 + 0.55 X1 - 0.63 X2 + 0.29 X3 - 0.26 X4 + 0.06 X5 + 0.91 X6.

Fig. 14. An example of a regression model obtained with the use of linear regression. The regression procedure computes the numbers in the equation for A.

ultimately prove unsuitable as medications. Given a biological target to modulate, the role of chemistry is to identify the initial set of chemical compounds that can be taken forward for development. In this, structure-activity relationships (SARs) can play an important role in ensuring that a large proportion of the compounds proposed are also effective. In the past, the relational representation used by ILP has been repeatedly shown to be particularly well-suited to the task of constructing good SARs, with one caveat: the representation of activity has been a simple categorisation (usually "active" or "inactive"). In this paper, we have sought to extend ILP-constructed SARs to true quantitative models by utilising some standard statistical procedures as background knowledge. In doing so, we have had to confront some specific issues that arise during model construction and prediction when using statistical procedures within a first-order logic setting. To address these, we have proposed an ILP system that: (a) identifies statistical models in the SAR by employing the procedure in Section 4; and (b) uses the resulting SAR within domain-independent or domain-specific procedures to ensure deterministic prediction (as described in Section 5).

The obvious limitation of the procedures proposed here is the lack of provable properties, in particular, about the optimality of models constructed. In the absence of such properties, we have attempted to demonstrate the practical utility of the procedures using using data on the inhibition of Thermolysin. To the best of our knowledge, the results represent the first examples of a constructing truly quantitative 3-dimensional SARs with ILP.

The literature on attempting to incorporate statistical models within ILP hypotheses is relatively sparse. Both [18] and [36] are concerned with the use of linear regression by an ILP program (as a built-in definition in [18] and as a background predicate in [36]). In the former, the multiple instance problem is avoided by restricting background predicates that introduce the independent variables to be strictly deterministic (that is, functional). The latter simply ig-

The affinity of molecule M is A if:

M has a hydrogen donor at position P1 and
M has a hydrogen donor at position P2 and
M has a negatively charged atom at position P3 and
M has a negatively charged atom at position P4 and
the distance between P1 and P2 is X1 and
the distance between P1 and P3 is X2 and
the distance between P1 and P4 is X3 and
the distance between P2 and P3 is X4 and
the distance between P2 and P4 is X5 and
the distance between P3 and P4 is X6 and
A is the kernel regression estimate using instances I

Where I is:

A	X1	X2	X3	X4	X5	X6
3.3	3.7	2.2	5.5	2.9	2.3	4.2
7.5	6.6	6.8	4.7	12.0	10.0	4.6
...	...	...	...	...	...	...
...	...	...	...	...	...	...
...	...	...	...	...	...	...

Fig. 15. An example of a regression model obtained with the use of kernel regression. The entries in the tabulation are the training instances returned by the search procedure in Fig. 7.

nores the multi-instance problem and treats all instances as independent data points. Adopting the same approach, Slattery and Craven [4] examine the use of a built-in naive Bayesian classifier. To this extent, the work here appears to the first to confront squarely the problems of using statistical procedures with non-determinate background knowledge. More generally, there is an increasing awareness within machine learning of the importance of learning statistical models from relational data. In Appendix B, we place this work within the context of broader efforts in the emerging field of statistical relational learning.

The techniques presented are not confined to the particular statistical procedures used, a particular ILP system, or to the construction of SARs. This suggests three interesting ways in which the work here could be extended. First, the use of other conditional estimation procedures may yield better SAR models than those obtained here. Second, the same techniques could be used to extend the capabilities of other quantitative ILP approaches like those that construct first-order regression trees. Third, and most interestingly, the same procedures should provide any ILP system with the tools necessary to construct complex theories that combine first-order and 'propositional' aspects. An example of this are the theories described in [34]. A.D. Shapiro's work on *structured induction* requires expert assistance to decompose hierarchically a complex induction task into sub-problems that can solved inductively (in Shapiro's case, using a tree learner). While the approach was shown on difficult chess endgames to yield novel and comprehensible theories, the need to provide a complete decomposi-

tion of the task has remained a principal difficulty with the technique. It is of interest to see the extent to which an ILP system can construct such theories automatically (equipped, for example, with a tree learner as background knowledge and appropriate domain-specific constraints).

Acknowledgements

Much of this work was done when the first author was at the Computing Laboratory, Oxford. Thanks are also due to Steve Moyle for several interesting discussions on the techniques described here and his generous computational support; to Nathalie Marchand-Geneste for the data on the Thermolysin inhibitors; and to Ravi Kothari for suggesting the use of kernel regression.

References

1. R.A. Amar, D.R. Dooly, S.A. Goldman, and Q. Zhang. Multiple-Instance Learning of Real-Valued Data. In C.E. Brodley and A.P. Danyluk, editors, *Proceedings of the Eighteenth International Conference on Machine Learning (ICML 2001)*, pages 3–10, San Francisco, 2001. Morgan Kaufmann Publishers.
2. W. Buntine. Operations for learning with graphical models. *Journal of Artificial Intelligence Research*, 2:159–225, 1994.
3. V. S. Costa, D. Page, M. Qazi, and J. Cussens. CLP(BN): Constraint Logic Programming for Probabilistic Knowledge. In *Proceedings of the Nineteenth International Conference on Uncertainty in AI (UAI-2003)*, San Francisco, CA, 2003. Morgan Kaufmann. (to appear).
4. M. Craven and S. Slattery. Relational Learning with Statistical Predicate Invention: Better Models for Hypertext. *Machine Learning*, 43(1-2):97–119, 2001.
5. J. Cussens. Stochastic logic programs: Sampling, inference and applications. In *Proceedings of the Sixteenth Annual Conference on Uncertainty in Artificial Intelligence (UAI-2000)*, pages 115–122, San Francisco, CA, 2000. Morgan Kaufmann.
6. T. Dietterich, R. Lathrop, and T. Lorano-Perez. Solving the multiple-instance problem with axis-parallel rectanges. *Artificial Intelligence*, 89:31–71, 1997.
7. D.R. Dooly, S.A. Goldman, and S.S. Kwek. Real-Valued Multiple-Instance Learning with Queries. In N. Abe, R. Khardon, and T. Zeugmann, editors, *Proceedings of the Twelfth Conference on Algorithmic Learning Theory (ALT 2001)*, volume 2225 of *LNAI*, pages 167–180, Berlin, 2001. Springer-Verlag.
8. P. Finn, S. Muggleton, D. Page, and A. Srinivasan. Pharmacophore Discovery using the Inductive Logic Programming system Progol. *Machine Learning*, 30:241–270, 1998.
9. P.W. Finn. Computer-based screening of compound databases for the identification of novel leads. *Drug Design Today*, 1(9):363–370, 1996.
10. N. Friedman, L. Getoor, D. Koller, and A. Pfeffer. Learning probabilistic relational models. In *Proceedings of the 16th International Joint Conference on Artificial Intelligence*. Stockholm, Sweden, 1999.
11. L. Getoor, N. Friedman, D. Koller, and A. Pfeffer. Learning probabilistic relational models. In S. Dzeroski and N. Lavrac, editors, *Relational Data Mining*, pages 307–335. Springer, Berlin, 2001.

12. C. Hansch and A. Leo. *Exploring QSAR: Fundamentals and Applications in Chemistry and Biology*. American Chemical Society, Washington, DC, 1995.
13. W. Hardle. *Applied Nonparametric Regression*. Cambridge University Press, Cambridge, England, 1994.
14. T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer-Verlag, Berlin, 2001.
15. D. Heckerman, C.Meek, and D. Koller. Probabilistic models for relational data. Technical Report MSR-TR-2004-30, Microsoft Research, 2004.
16. G.H. John and P. Langley. Estimating continuous distributions in Bayesian classifiers. In *Proceedings of the Eleventh International Conference on Uncertainty in AI*, pages 338–345, San Francisco, CA, 1995. Morgan Kaufmann.
17. M. I. Jordan. Why the logistic function? A tutorial discussion on probabilities and neural networks. Technical Report 9503, Computational Cognitive Science, MIT, Cambridge, MA, 1995.
18. A. Karalic and I. Bratko. First-Order Regression. *Machine Learning Journal*, 26:147–176, 1997. Special Issue on ILP.
19. K. Kersting and L. De Raedt. Bayesian Logic Programs. Technical Report 151, Department of Computer Science, University of Freiburg, Germany, 2001.
20. R.D. King, A.Srinivasan, and M.J.E. Sternberg. Relating chemical activity to structure: an examination of ILP successes. *New Gen. Comput.*, 13(3,4):411–433, 1995.
21. N. Marchand-Geneste, K.A. Watson, B. Alsberg, and R.D. King. A new approach to pharmacophore mapping and QSAR analysis using Inductive Logic Programming. Application to Thermolysin inhibitors and Glycogen Phosphorylase B inhibitors. *Journal of Medicinal Chemistry*, 45:399–409, 2002. (with corrections in Vol 46, pg. 653).
22. O. Maron. *Learning from Ambiguity*. PhD thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA, 1998.
23. T. M. Mitchell. *Machine Learning*. Mc-Graw-Hill, New York, 1997.
24. S. Muggleton. Stochastic logic programs. In L. DeRaedt, editor, *Advances in Inductive Logic Programming*, pages 254–264. IOS Press, 1996.
25. S. Muggleton. Semantics and derivation for stochastic logic programs. In *UAI-2000 Workshop on Fusion of Domain Knowledge with Data for Decision Support*, 2000.
26. S.H. Muggleton. Learning structure and parameters of stochastic logic programs. In *Proceedings of the Twelfth International Conference on Inductive Logic Programming (ILP02)*, pages 198–206, Berlin, 2002. Springer.
27. R. Ng and V.S. Subrahmanian. Probabilistic logic programming. *Information and Computation*, 101(2):150–201, 1992.
28. L. Ngo and P. Haddawy. Probabilistic logic programming and bayesian networks. In *Algorithms, Concurrency and Knowledge*, pages 286–300. Springer, 1995.
29. C. Perlich, F. Provost, and J. Simonoff. Tree Induction vs. Logistic Regression: A Learning-curve Analysis. *Journal of Machine Learning Research*, 4:211–255, 2003.
30. D. Poole. Probabilistic Horn abduction and Bayesian networks. *Artificial Intelligence*, 64(1):81–129, 1993.
31. F. Provost and T. Fawcett. Robust classification for imprecise environments. *Machine Learning*, 42:203–231, 2001.
32. S. Ray and C.D. Page. Multiple Instance Regression. In *Proceedings of the Eighteenth International Conference on Machine Learning*, San Mateo, CA, 2001. Morgan Kaufmann.

33. T. Sato. A statistical learning method for logic programs with distributional semantics. In L. Sterling, editor, *Proceedings of the Twelth International conference on logic programming*, pages 715–729, Cambridge, Massachusetts, 1995. MIT Press.
34. A.D. Shapiro. *Structured Induction in Expert Systems*. Addison-Wesley, Wokingham, 1987.
35. A. Srinivasan. The Aleph Manual. Available at http://www.comlab.ox.ac.uk/oucl/research/areas/machlearn/Aleph/, 1999.
36. A. Srinivasan and R.C. Camacho. Numerical reasoning with an ILP program capable of lazy evaluation and customised search. *Journal of Logic Programming*, 40(2,3):185–214, 1999.
37. A. Srinivasan and R.D. King. Feature construction with Inductive Logic Programming: a study of quantitative predictions of biological activity aided by structural attributes. *Data Mining and Knowledge Discovery*, 3(1):37–57, 1999.

A Statistical Procedures for Conditional Estimation

A.1 Procedures for estimating conditional probabilities

We are concerned here with the binary classification problem in which each of n training data points are labelled by a random variable ω which takes values from a discrete set $\{+,-\}$ (in the structure-activity case, denoting "active" and "inactive"). Assume the data are in the form of a d-dimensional random vector $\mathbf{X} = [X_1, X_2, \ldots, X_d]^T$ where the $X_i \in \Re$. Given a particular vector $\mathbf{x} = [x_1, x_2, \ldots, x_d]^T$ we wish to obtain an estimate of the conditional probabilities of $\omega = +$ (the corresponding probability for $\omega = -$ follows automatically). From Bayes rule, the relevant posterior is given as:

$$P(\omega = +|\mathbf{x}) = \frac{P(\mathbf{x}|\omega = +)P(\omega = +)}{P(\mathbf{x})}$$

Two well-known simple cases follow from specific assumptions about the data:

1. The assumption that the class-conditional densities $P(\mathbf{x}|\cdot)$ are from the exponential class (of which the Gaussian is a member) results in (see [17]):

   $$P(\omega = +|\mathbf{x}) = \frac{1}{1 + e^{-\xi}}$$

   where ξ is a linear equation of the X_i. The logistic regression procedure uses sample data to estimate probabilities under this assumption.
2. The assumption that the X_i are conditionally independent of each other given the value of ω, results in:

   $$P(\omega = +|\mathbf{x}) \propto \prod_1^d P(x_i|\omega = +)P(\omega = +)$$

   The naive Bayesian procedure uses sample data to estimate probabilities under this assumption. The individual class-conditional densities can be estimated using one-dimensional kernel density estimates ([16]).

A.2 Procedures for estimating conditional means

We are concerned here with the regression problem in which each of n training data points are labelled by a random variable $Y \in \Re$. Assume the data are in the form of a d-dimensional random vector $\mathbf{X} = [X_1, X_2, \ldots, X_d]^T$ where the $X_i \in \Re$. We seek a *regression* function $f(\mathbf{X})$ for predicting Y given a particular vector $\mathbf{x} = [x_1, x_2, \ldots, x_d]^T$. It can be shown [14] that for a quadratic error loss function $(Y - f(\mathbf{X}))^2$ the function f that minimises the expected prediction error is:

$$f(\mathbf{x}) = \mathrm{E}(Y|\mathbf{x})$$

Two well-known simple cases follow from specific assumptions about f:

1. The assumption that f is a linear function of its arguments results in:

$$\mathrm{E}(Y|\mathbf{x}) = \alpha + \sum_1^d \beta_i x_i$$

   The linear regression procedure uses sample data to estimate α and the β_i under this assumption.
2. The assumption that f can be approximated by a weighted average of training instances in the neighbourhood of $\mathbf{x}$ results in:

$$\mathrm{E}(Y|\mathbf{x}) = \frac{\sum_1^n K(\mathbf{x}, \mathbf{x}_i) y_i}{\sum_1^n K(\mathbf{x}, \mathbf{x}_i)}$$

   where K is a kernel density function. The kernel regression procedure uses sample data to obtain this weighted average, given a particular choice of K.

B Relationship to Statistical Relational Learning

Statistical relational learning (SRL) is an active area of research within machine learning. While the acronym appears to have been first used by Jensen and Getoor in 2000 for a AAAI workshop that greatly fostered the growth of this research area, work on the topic goes back at least to 1994. The class-probability SAR models constructed here is one example of the construction of explicit probabilistic modes from relational data. SRL attempts to investigate relational and probabilistic learning in a unified manner. This appendix briefly reviews the history of SRL and shows how the present work fits within it.[5]

The earliest work in the SRL setting that we are aware of is by Buntine, who introduced the idea of *plates* into graphical models [2]. Plates provide a means of extending graphical models to relational data. Nevertheless, the learning component of that work learned only the parameters, not the structure of a

[5] A more comprehensive collection of papers can be found at www.cs.wisc.edu/ page/838.html, though it is by no means exhaustive.

model. Friedman, Getoor, Koller and Pfeffer were the first to learn the structure of plate-like models, by learning the structure of probabilistic relational models, or PRMs [10]. Recently Heckerman, Meek and Koller have shown how PRMs and other related models can be expressed using plates, and they have provided extensions to Buntine's original plates [15].

Muggleton's stochastic logic programs [24] or SLPs approached statistical relational learning from a different direction. Whereas Buntine scaled probabilistic (graphical) models to handle relational data, SLPs extended relational learning techniques to include explicit probabilities. This differed from earlier work by Poole [30], Ng and Subrahmanian [27], Sato [33], Ngo and Haddawy [28] and others on probabilistic logic programs because it emphasized the importance of learning. Learning algorithms for stochastic logic programs were further developed and significantly extended by Cussens and Muggleton [5, 25, 26].

Many of these ideas can be seen as special cases of more recent, general approaches to include Bayesian networks in logic programs [3, 19]. The remainder of this appendix shows in more detail how the use of logistic regression or naive Bayes as background predicates are special cases of the CLP(BN) proposal in [3]. CLP(BN) refers to an extension of the language of logic programs to allow the inclusion of Bayesian constraints on variables in a clause. Programs written in CLP(BN) are thus constraint logic programs (CLPs) and require a specialised solver for answering queries. The answer can now include marginal probability distributions on variables. Conceptually, a clause in CLP(BN) can be thought of as being composed of a logical portion, augmented by a Bayesian network (see Fig. 16). As with ordinary logic programs, answering a query may call on several clauses, resulting in increasingly larger Bayesian networks as variables from different clauses are unified during the course of a proof (we refer the reader to [3] for the precise syntax and semantics of CLP(BN) programs).

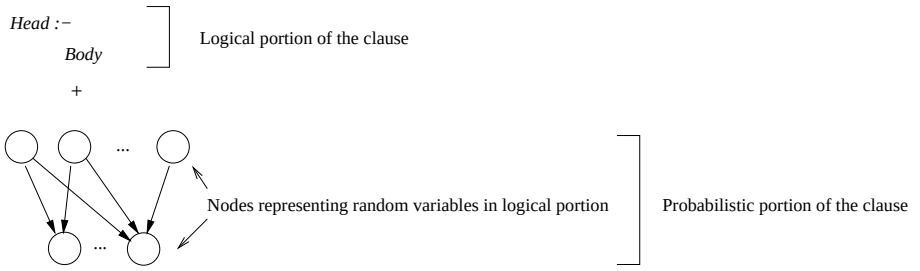

Fig. 16. A pictorial representation of a CLP(BN) clause. The logical portion of the clause is a normal logic program (with the *:-* to be read as "if"; *Head* being a literal and *Body* being a conjunction of literals. The probabilistic portion is a Bayesian network that encodes the conditional probability distribution amongst terms in the logical portion.

A comparison of the rule in Fig. 11 and that in Fig. 16 readily shows the representation adopted in this paper to be closely related to the CLP(BN) language. In fact, the naive Bayes and logistic regression procedures simply encode specific Bayesian network topologies that reflect the underlying assumptions described in Section A.1 (see Fig. 17). When using the logistic function to compute class probability, the rule in Fig. 11 is thus conceptually identical to the CLP(BN) clause in Fig. 18.

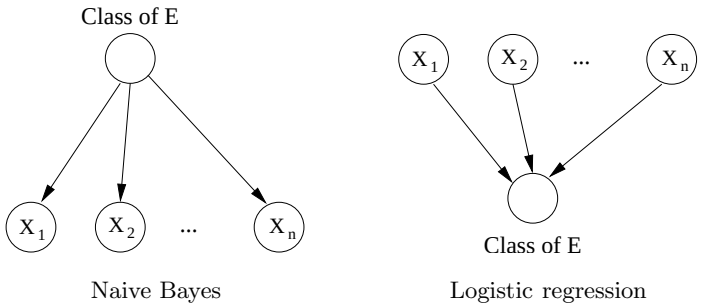

Fig. 17. Conditional dependencies assumed by the naive Bayes and logistic regression procedures. Here $X_1, X_2, \ldots X_n$ are existentially quantified variables. For simplicity, we have omitted showing details of the (conditional) probability distributions associated with each node.

Given this correspondence, an immediate question that arises is this: why not simply use CLP(BN) as the representation language? For the construction of general probabilistic models in ILP, the approach here is clearly inadequate. However, it still has some merits when constructing class-probability rules:

1. No extensions are required to the language of logic programs;
2. The problem of parameter estimation with non-deterministic background predicates is addressed directly. This is side-stepped in CLP(BN) by the use of aggregation functions (an approach it inherits from its intellectual predecessor, probabilistic relational models [11]). Essentially, this means replacing sets of values by some single representative (for example, the average). For many problem domains (including the applications to drug design in Section 6), it may not be sensible to use such functions;
3. Statistical procedures for parameter estimation are usually extremely efficient and, under some circumstances, return optimal estimates; and
4. Any procedure for probability estimation can be used relatively easily by encoding it as a background predicate.

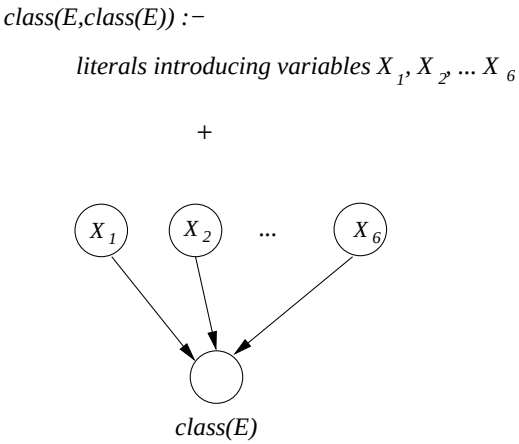

Fig. 18. CLP(BN) representation of the rule in Fig. 11, using the logistic function to compute the class probability distribution for *class(E)*.