Using inductive logic programming to discover knowledge hidden in chemical data

Bryant, CH, Adam, AE, Taylor, DR and Rowe, RC

http://dx.doi.org/10.1016/S0169-7439(97)00023-3

Using Inductive Logic Programming to Discover Knowledge Hidden in Chemical Data

C.H.Bryant*, A.E.Adam (Computation Department)
D.R.Taylor (Chemistry Department)

University of Manchester Institute of Science and Technology,
PO Box 88, Manchester, M60 1QD, United Kingdom.

R.C.Rowe

Zeneca Pharmaceuticals, Alderley Park, Macclesfield, Cheshire,
SK10 2NA, United Kingdom.

Abstract

This paper demonstrates how general purpose tools from the field of Inductive Logic Programming (ILP) can be applied to analytical chemistry. As far as these authors are aware, this is the first published work to describe the application of the ILP tool Golem to separation science.

An outline of the theory of ILP is given, together with a description of Golem and previous applications of ILP. The advantages of ILP over classical machine

* *Correspondence to:* C.H.Bryant, School of Computing and Mathematics, The University of Huddersfield, HD1 3DH, United Kingdom.

induction techniques, such as the Top-Down-Induction-of-Decision-Tree family, are explained.

A case-study is then presented in which Golem is used to induce rules which predict, with a high accuracy (82%), whether each of a series of attempted separations succeed or fail. The separation data was obtained from published work on the attempted separation of a series of 3-substituted phthalide enantiomer pairs on (R)-N-(3,5-dinitrobenzoyl)-phenylglycine.

1 Introduction

Inductive Logic Programming (ILP) is an active area of research in computer science which has given rise to a number of general purpose tools that can be applied to chemistry. ILP has been defined [1] as the intersection between machine induction [2] and logic programming [3]. (For an introduction to machine induction and its application to analytical chemistry see [4].)

The most widely used language in logic programming is Prolog (**Pro**gramming in **log**ic) [5]. Most ILP systems use a subset of Prolog as the representational formalism for both hypotheses and observations. In doing so ILP overcomes two of the main limitations of classical machine learning techniques such as the Top-Down-Induction-of-Decision-Tree family [6]:–

1. The use of what is essentially a propositional logic which is a limited knowledge representation formalism.
2. The difficulty in using substantial background knowledge in the learning process.

The first limitation is important because it prevents classical machine learning techniques from being used for those domains which cannot be represented using propositional logic. The greater representative power provided by Prolog allows ILP to induce rules which express relationships that cannot be represented by propositional logic. For example, rules can be induced that reason not only about properties of observations but also about the relationship between those observations, where an observation corresponds to a leaf in decision tree. This is illustrated

during the case-study (see Section 2.2.4).

The second limitation is also important because "... one of the well-established findings of artificial intelligence is that the use of domain knowledge is essential for achieving intelligent behaviour. Logic offers an elegant formalism to represent knowledge and hence incorporate it in the induction task" [7]. ILP offers the opportunity to use both specialist knowledge on particular problems in chemistry and general chemical knowledge during induction. General knowledge refers to knowledge which is common-place amongst chemists. An example of its usefulness during induction is discussed in Section 2.3.2.

The main areas in which ILP has been applied are scientific discovery, knowledge acquisition and programming assistants. Applications of ILP to scientific discovery and knowledge acquisition include drug design, protein folding, diterpene structure elucidation from ^{13}C NMR spectra [12],diagnosis of faults in the power supply of satellites and rheumatology diagnosis. ILP has also been applied to finite element mesh design. Papers that review ILP applications include [7], [13] and [14].

The application of ILP to the domains of drug design and protein folding was deemed a success in that it resulted in new knowledge which was subsequently published in refereed journals of these domains and which was meaningful to scientists working in these domains [7]. Both of these applications provided an insight into the stereochemistry involved in each case [13] and demonstrated that ILP is able to make use of a substantial amount of general chemical knowledge.

The remainder of this section gives an outline of the theory of ILP and describes Golem, the most widely field tested ILP tool. As far as these authors are aware, Golem has not been applied to separation science previously.

1.1 Theoretical Outline of ILP

An ILP algorithm is given an initial background theory T and some evidence (examples) $E = E^+ \cup E^-$. Positive evidence E^+ is true[1] and negative evidence E^- is false. The algorithm then has to induce a hypothesis H that together with T

[1] in the intended domain. (For the sake of conciseness, this proviso will be assumed when referring to true and falsehood in the remainder of this paper and thus it will not be repeated.)

explains the examples E. In the general case, T, E and H can be any set of clauses. Usually however, the theory, evidence and hypothesis have to satisfy certain syntactic restrictions, which are referred to as the *bias* B [7] [8]. An inappropriate bias can prevent the learner from finding the intended hypotheses. [8]

In ILP the search space is typically structured by means of the dual notions of generalisation and specialisation. [7] believe that generalisation corresponds to induction, specialisation corresponds to deduction and that therefore induction should be viewed as the inverse of deduction. They use the definitions listed below where G and S are hypotheses, r is an inference rule and R is a set of inference rules. (The symbol $\models$ represents logical entailment and the symbol $\square$ represents false [9]. A conjunction of clauses is a set of clauses logically ANDed together.)

- A hypothesis G is more general than a hypothesis S if and only if $G \models S$. S is also said to be more specific than G.
- A deductive inference rule $r \in R$ maps a conjunction of clauses G onto a conjunction of clauses S such that $G \models S$; r is called a specialisation rule.
- An inductive inference rule $r \in R$ maps a conjunction of clauses S onto a conjunction of clauses G such that $G \models S$; r is called a generalisation rule.

Hence the notions of generalisation and specialisation are incorporated into a search using inductive and deductive inference rules. Generalisation prunes the search space as follows. When $T \wedge H \wedge E^- \models \square$ then all generalisations H' of H will also be inconsistent with $T \wedge E^-$. Such generalisations can therefore be pruned from the search: they would only result in an over-generalisation. Specialisation prunes the search space as follows. When $T \wedge H \not\models E^+$ then none of the specialisations H' of H will imply the evidence. Such specialisations can therefore be pruned from the search: they would only result in an over-specialisation.

A very large number of tools for ILP have been described in the literature. A review of these is outside the scope of this paper. For a discussion of the characteristics of ILP tools and an overview of a representative sample of these see [7]. Section 1.2 describes Golem, one of the most widely field tested ILP tools.

1.2 Golem

Golem (version 1.0α) [10] is an ILP tool which is available in the public domain and which was developed at the Turing Institute, Glasgow. Golem starts from initial background theory and evidence (examples) and repeatedly generalises its hypothesis by applying an inductive inference rule, taking care that the hypothesis does not imply negative examples. Golem is a non-incremental ILP system, that is, it is given the initial background theory and evidence prior to induction. It is non-interactive in the sense that it does not pose questions to the User about how the background theory and evidence should be interpreted. Golem generates rules of the following form using Plotkin's framework of relative least general generalisation:-

target_predicate:- background_predicate_1, background_predicate_2,
... background_predicate_n.

The body (antecedent) of the rule is a conjunction of n background predicates, where $n \geq 1$. The syntax follows that of Edinburgh Prolog. Golem does not have numerical operators (such as $<$ and $>$) built-in and so it cannot generate rules that include these. Golem needs the following three files as input.

Foreground This contains the positive examples, that is instantiations of the target predicate that are true.

Negatives This contains the negative examples, that is instantiations of the target predicate that are false.

Background This contains the background knowledge, that is instantiations of the background predicates that are true, and the parameter settings.

Two aspects of the bias of Golem are relevant to this paper.

The Syntactic Bias of Golem All the positives, negatives and background knowledge that is input to Golem must be represented using Prolog facts only.

The Semantic Bias of Golem Golem can only be used to generate rules that are *ij*-determinate. Roughly speaking, a rule is "... determinate if all of its literals are determinate; and a literal is determinate if each of its variables that

does not appear in preceding literals has only one possible binding given the bindings of its variables that appear in preceding literals" [7]. Section 2.2.2 includes an example of an indeterminate rule. Each new variable in a body literal of a determinate rule can be linked to a variable in the head through a path of predicates. i specifies the maximum number of predicates that are allowed in any such path. j is the maximum number of terms[2] in a literal that can determine the binding of a variable in that literal. The User may specify the values of i and j; by default Golem assigns the value 2 to both [11].

2 A Case-Study

This section describes a case-study in which Golem was used to develop rules which predict whether each one of a series of phthalide enantiomer pairs can be separated by a particular chiral stationary phase.

2.1 Enantioseparations

The separation of enantiomers by High Performance Liquid Chromatography (HPLC) using chiral stationary phases (CSPs) is based on the formation of transient diastereomeric complexes between the enantiomers of the solute and a chiral selector that is an integral part of the stationary phase. The difference in stability between these complexes leads to a difference in retention time: the enantiomer that forms the less stable complex will be eluted first. If the difference in stability is too small no separation is observed. Such enantioseparations are important in many scientific disciplines, including stereoselective synthesis, mechanistic and catalytic studies, agrochemistry, medicine and pharmacology. (See [15] for a review of enantioseparations.)

Since enantioseparations are performed in many disciplines and since there is a choice of over 80 commercially available CSPs, guidelines are needed on the choice of materials for enantioseparations by HPLC. A computer system which

[2]Term is intended to have it first order predicate logic meaning here. It corresponds to an argument in Prolog terminology.

could guide analysts in the choice of materials for enantioseparations by HPLC would be beneficial because there are currently few guidelines on how to choose the materials and they are difficult to access: the papers describing them are spread across a wider range of scientific journals than analysts can be reasonably expected to survey. Analysts need guidance because it may not be possible to attempt to separate an enantiomer on a series of CSPs on a 'trial and error' basis for the following reasons [16]:-

- CSPs are expensive. A small laboratory may not be able to afford to stock a wide range of CSPs.
- CSPs have a limited shelf-life. Laboratories will only want to purchase a CSP when they believe that they have a use for it.
- The process of attempting separations on a series of CSPs may take too long: the optimisation of an enantioseparation on a *single* CSP often takes at least a day.

CHIRBASE [17] [18] [19] is a conventional database which makes data on enantioseparations accessible but it is expensive. Furthermore it does not tell an analyst how to use such data, that is guide an analyst in the selection of materials for a particular enantioseparation. CHIRULE is a computer system that was designed to provide such guidance. CHIRULE was developed by Stauffer and is described in PhD thesis [20]. It uses similarity searching on molecular properties to retrieve a list of enantiomer pairs that are chemically similar to a given enantiomer pair, together with columns that have been reported in the literature to have successfully separated them. However in his thesis Stauffer does not report testing CHIRULE to see which CSPs it would recommend when it was given enantiomer pairs which have been reported in the literature as having been separated on Pirkle-type CSPs. Pirkle-type CSPs are so named because their invention is credited to W.H.Pirkle's group at the University of Illinois. They are also referred to as the 'brush' or 'multiple interaction' type. They are chiral selectors of moderate molecular weight covalently bonded to silica.

The case-study was performed as part of a project concerned with the development of an expert system for enantioseparations by HPLC. An expert system is a

computer program that represents and reasons with knowledge of some specialist subject with a view to solving problems or giving advice[21]. The characteristics of expert systems are described in [22] together with previous expert systems for chromatography. As far as the authors of this work are aware this is the first project to have taken a validated first step towards a computer system that gives guidance on the selection of materials for enantioseparations on Pirkle-type CSPs.

The approach taken by the authors to developing an expert system for enantioseparations has been to induce rules from data downloaded from a relational database of enantioseparations. This database is described in [23].

2.2 Experimental

This section describes the data on enantioseparations used in the case-study, how that data was represented in the input to Golem and the approach taken to enabling Golem to make generalisations about the distance of structural features from the asymmetric centre.

2.2.1 The Data Set

The data used in the case-study was taken from a study performed by Pirkle and Sowin [24] that investigated mechanisms by which enantiomers are separated on CSPs. The paper on the study lists both successful and failed attempts to separate a series of 3-substituted phthalides. Table 1 includes the structure of phthalide and indicates the location of substituents at the 3 position. The table gives details of the attempted separation on (R)-N-(3,5-dinitrobenzoyl)-phenylglycine of all the 3-substituted phthalides listed in the paper. The first column of the table assigns a label to each one to allow them to be easily identified later in this paper. The last column indicates the degree of separation that was achieved.

The data shown in Table 1 was represented using the predicates separates_on(E, C) and <structural_feature>(E, D) (see Section 2.2.2). The following convention was adopted for the bindings of the E variable where <enantiomer_pair_id> is a unique database key and label is the label used in Table 1. (The significance of the key is of no concern to this paper.)

no_<enantiomer_pair_id>_<label>

Figures 3, 4, 5 and 6 show all the data that was input to Golem. All the default settings of Golem were used except that i and j were both set to ten. The literals for the <structural_feature>(E, D) predicate were stored in the background file (see Figures 5 and 6) that was input to Golem. The literals for the predicate separates_on(E, C) were split between the foreground and negatives files that were input to Golem (see Figure 3). Those separates_on literals representing successful separations were stored in the foreground file and those representing failed separations in the negatives file. A separation was considered successful if the separation factor (selectivity) was greater than or equal to 1.04, otherwise it was considered a failure.

2.2.2 Selecting Predicates for Golem

This section describes how predicates were chosen that both meet the constraints imposed by the bias of Golem and represent relationships between data on enantioseparations and structural features of enantiomers.

The aim of the case-study was to develop rules[3] using Golem which correctly predict whether each of the series of 3-substituted phthalides will separate on (R)-N-(3,5-dinitrobenzoyl)-phenylglycine. To represent such rules in Prolog it is necessary to use a predicate that maps enantiomer pairs to CSP chiral selectors. Hence the predicate separates_on(E, C), where E = enantiomer pair and C = CSP chiral selector, was used as the target predicate for Golem.

Names of attributes in the database (see Section 2.1) represent structural features and the values of these attributes represent the distances of the occurrences of the structural features from the asymmetric centre in terms of the number of connecting bonds. [4] gives full details of this representation. There are four possible forms of predicate that can represent such data given the syntactic bias of Golem, that is the restriction to Prolog facts:–

[3]The authors realise that an expert system for enantioseparations by HPLC would need to provide the Users of such a system with more information than just which CSP chiral selector to use. However in this work, the first step in the development of such a system, the recommendations were limited to CSP chiral selectors so that the experiments with ILP would remain tractable.

1. **has_feature(F, E, D)** F = structural feature (including the occurrence) and D = distance from the asymmetric centre.
2. **has_feature(F, O, E, D)** Here F = structural feature and O = occurrence of a structural feature. This form requires that the names of the structural feature attributes are split into their constituent feature and occurrence parts.
3. **<structural_feature>(E, D)** This form requires a predicate for each of the structural feature attributes. An example of such a predicate is bg6_3(E, D).
4. **<structural_feature>(O, E, D)**

The third form of predicate was used with Golem because Golem's semantic bias precludes the use of the first, second or fourth. To understand why consider the form of the rules that Golem would generate if the target predicate was specified as separates_on(E, C) and the only background predicate was has_feature(F, E, D):–

$$\text{separates\_on(E, C) :– has\_feature(F, E, D) } \ldots$$

When E is bound in the separates_on literal the rule is indeterminate because the has_feature literals are indeterminate: more than one has_feature literal will be needed to represent an enantiomer pair. The preclusion of the has_feature predicate illustrates the restrictive nature of the semantic bias of Golem.

2.2.3 Enabling Golem to Generalise Distances

Providing Golem with just those predicates selected in the previous section is not sufficient to enable it to make useful generalisations because without additional background predicates Golem is not able to make generalisations about the distance at which chiral recognition can take place. Golem cannot induce rules of the following form because it does not have numerical operators built in.

$$\text{separates\_on(E, C) :- bg6\_3(E, D), D} < n.$$

where n is an integer representing a specific distance.

Without additional background predicates Golem will only induce rules that reason about the presence of structural features at particular distances or at no particular distance. These authors believe that for a machine induction tool to be

of use for enantioseparations it must be able to generalise distance values in a more flexible manner than this.

The most obvious way to enable Golem to generalise distances is to use a background predicate such as greater_than(D1, D2). This involves specifying all the possible Prolog facts for this predicate for all the distance values input to Golem,[4] as shown in Figure 4. Such predicates can be used to represent the same generalisations that can be represented by the corresponding numeric operators.

Since Golem may or may not use particular background predicates during induction the provision of such a predicate allows Golem to reason about the presence of structural features at particular distances, *at a range of distances* or at no particular distance.

2.2.4 Inducing Inter-Feature Relationships

The expressive power of Prolog can represent relationships between the structural feature predicates that cannot be expressed using the language of classical induction techniques (see Section 1). Consider the following rule which states that feature_y must be further from the asymmetric centre than feature_x.

```
separates_on(E, csp):– feature_x(E, Distance_for_x),
                       feature_y(E, Distance_for_y),
                       greater_than(Distance_for_y, Distance_for_x).
```

An ILP tool will be capable of inducing a rule such as that shown above providing that the rule complies with the declarative bias of the tool. This is the case for Golem.

2.3 Results and Discussion

Figure 1 shows the rules that were generated by Golem and gives their English translation. The rules have a high accuracy (82%) for the training set. Neither

[4] If the number of distance values was large then the number of Prolog facts needed would require that either another way of enabling Golem to generalise distances was sought or that the Prolog facts were automatically generated by some method. However in this project there were only ten distance values (0, 1, 2, ... 9).

covers any of the five failed separations (negatives). Together they cover 13 of the 17 successful separations (positives).

$$Accuracy = \frac{13+5}{17+5} \times 100 = 82\%$$

Rule for 1-naphthyl Derivatives.

```
separates_on(A,dnbpg) :- c_2(A,B),
                         bg6_3(A,C),
                         greater_than(3,C),
                         greater_than(B,3).
```

Enantiomeric 3-substituted phthalides will be separated on (R)-N-(3,5-dinitrobenzoyl)-phenylglycine if they have:–

1. at least three six-membered aromatic rings and the third closest of these to the asymmetric centre is less than three bonds away.
2. at least two one-membered alkyl chains and the second closest of these to the asymmetric centre is more than three bonds away.

Rule for Anthryl Derivatives.

```
separates_on(A,dnbpg) :- tri1(A,1).
```

Enantiomeric 3-substituted phthalides will be separated on (R)-N-(3,5-dinitrobenzoyl)-phenylglycine if they have at least one tricyclic ring system and the closest such system to the asymmetric centre is one bond away.

Figure 1: Rules Induced by Golem for Phthalide Data.

The second rule includes the structural feature `tri1` (which refers to the closest occurrence of a tricyclic ring system to the asymmetric centre) and covers c and d, the only phthalides that have this feature. This is reflected in both the table of separation data taken from the literature (only c and d have an anthryl group in Table 1) and the background file input to Golem (only c and d have a literal involving `tri1` in Figures 5 and 6).

Phthalides e-v all have a naphthyl substituent. Notice that of the 15 phthalides with a naphthyl group attached at the 1 position, 14 (e-r) are positives and only one (s) is a negative. s is the only phthalide with a methyl group attached at the two position of the naphthyl. There are three phthalides with a naphthyl at the 2 position; two of these (u and v) are negatives and the third (t) is a positive but only just since $\alpha = 1.04$. If t is ignored then a reasonable rule for classifying phthalides e-v is as follows.

If there is a naphthyl substituent attached to the phthalide at the 1 position
and the naphthyl does not itself have a substituent at the 2 position
then a successful separation can be achieved.

Such a rule reflects some of the findings of the analysts who performed the separations. The "... 1-naphthylphthalides 1e-s (excepting 1s) resolve substantially better than the ... 2-naphthyl (1t-v) ... substituted phthalides. ... The dimethylnaphthyl group of 1s, due to steric interaction between the $2'$-methyl of the naphthyl ring and the peri-hydrogen of the phthalide benzo ring, cannot achieve the conformation necessary for effective chiral recognition ...". [24]

The first rule induced by Golem covers 11 of the 14 phthalides with a naphthyl group attached at the 1 position[5] and excludes both phthalide s and the phthalides with a 2-naphthyl substituent.

The rule makes the distinction between the 1 and the 2 position by reasoning about the distance at which the third closest six-membered aromatic ring occurs; this distance is equal to two for the 1-naphthyl but is equal to three for the 2-naphthyl. The rule states that the distance must be less than three and so excludes the three phthalides with a 2-naphthyl substituent. Thus the part of the rule that reasons about the distance of the `bg6_3` structural feature has a chemical justification.

The rule excludes phthalide s by stating that the second closest one membered alkyl chain must be more than three bonds away from the asymmetric centre. Figure 2 shows that a one membered alkyl chain at the 2-position on the naphthyl would be three bonds away. Hence the part of the rule that reasons about the

[5]Section 2.3.1 discusses why Golem did not induce a rule that covers phthalides e, f and g.

distance of the `c_2` structural feature from the asymmetric centre has a chemical justification too.

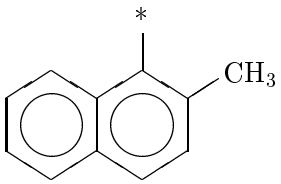


Figure 2: A Naphthyl Substituent Connected to the Asymmetric Centre at the 1 Position and to a One-Membered Alkyl Chain at the Two Position.

Despite the limited representation of molecules Golem induced rules that:–

- predict with a high accuracy (82%) which of the enantiomers shown in Table 1 separate on (R)-N-(3,5-dinitrobenzoyl)-phenylglycine.
- are chemically justified in that they reflect some of the findings of the analysts who performed the separations.[6]

Thus the case-study proved that Golem is able to induce rules for enantioseparations by making generalisations about the distance values. However the case-study demonstrated two shortcomings of Golem which are discussed next.

2.3.1 Enabling Golem to Include Negations of Literals

The discussion of Golem so far has not considered whether rules could be induced which specify that particular structural features are *not* present in an enantiomer. The experiment involving the phthalide data does provide an example of why it may be desirable for an ILP tool to be able to induce such rules. Recall that neither of the rules that were induced covered phthalides e, f and g and that these phthalides had 1-naphthyl substituents which did not have any substituents themselves.

[6]The authors recognise that the knowledge discovered by Golem is not original; the purpose of the case study was to demonstrate how ILP can be used to discover knowledge hidden in analytical chemistry data. Experiments involving ILP which attempt to discover new knowledge from a larger data set of enantioseparations will form the subject of future work.

The rule for 1-naphthyl derivatives (see Figure 1) excluded e, f and g by specifying that the second closest one-membered alkyl chain must be more than three bonds away. If instead the rule specified that this feature must not be three bonds away then its coverage would be extended to include e, f and g but would otherwise remain the same. The amended rule and the original rule for the anthryl derivatives together cover 16 of the 17 positives and exclude the five negatives. The positive which is not covered is t, the phthalide which was only just separated. This suggests that if ILP tools were able to induce rules which specify that features are not present then they may be able to induce better rules than would otherwise be the case.

Unfortunately it is not easy to enable Golem to induce such rules because it can only input Prolog facts: Golem must be supplied with Prolog facts representing the negation of every possible instantiation of each predicate that is not true for any given enantiomer. Consider how many literals would be needed. Given that 115 structural feature predicates and ten distance[7] values are represented in the database, there are $(115 \times 10 = 1150)$ possible instantiations of these predicates for any one enantiomer pair. Typically only eight of these are needed to represent an enantiomer pair. Thus approximately $((115 \times 10) - 8 = 1142)$ literals would be needed for each enantiomer pair to enable Golem to induce rules with negations of literals. Obviously having to use such a large number of literals to represent enantiomers is not desirable.

2.3.2 Why Golem Could be Used to Generalise Only Distances

Sections 2.2.3 and 2.2.4 described how Golem can be used to induce rules for enantioseparations by making generalisations about the distances from the asymmetric centre of occurrences of structural features. Although Golem was successfully used to induce rules for the phthalide data, only part of the potential of ILP for enantioseparations was tested.

ILP should allow substantial background knowledge to be used during induction (see Section 1). However the bias of Golem places restrictions on the chemical

[7]Excluding the predicate for the number of asymmetric centres which has nine possible values; this fact is ignored during this approximation.

knowledge which Golem can use, given the aim of using the data stored in the database on the structural features of enantiomer pairs. Recall that this data must be represented as instantiations of the structural feature predicates (see Section 2.2.2) in the background file input to Golem. Since Golem cannot input Prolog rules, it cannot use background rules that would enable it to make generalisations about the chemical features themselves, as opposed to the distances at which those features occur.

Although this proved sufficient for the phthalide data set, these authors believe that in some cases it may not matter whether a chemical feature is the closest, second closest or third closest occurrence of that feature, as long as the feature is present at a particular distance or within a range of distance values. If a machine induction tool is to be able to induce rules from the database that reflect this then it must be capable of generalising the data on both the occurrences and the distances.

Chemists often reason in terms of structural features that are more general than those that are represented in the database. For example reasoning about aromatic rings, as opposed to aromatic rings of a particular size, is omnipresent in chemistry. Enabling a machine induction tool to generalise the data in the database on the structural features would give it the potential to generate more concise rule-sets and to make discoveries that would not be possible otherwise.

Recently Progol [25], the successor to Golem, became available in the public domain. Progol is able to input background knowledge expressed as Prolog rules and, therefore, should be capable of generalising data on features and their occurrences. Experiments involving Progol will form the subject of future work.

3 Conclusion

The case-study shows that Golem can induce rules that

- predict with a high accuracy (82%) whether each of a series of attempted enantioseparations succeed or fail.

- are chemically justified in that they reflect some of the findings of the analysts who performed the separations.

As far as these authors are aware this is the first published work to describe the application of Golem to separation science.

In the opinion of the authors chemometricians (and other scientists) should consider ILP as a potentially superior alternative to classical machine induction techniques, such as the Top-Down-Induction-of-Decision-Tree family, whenever at least one of the following is true:-

- The rules to be generated may need to reason about relationships *between* the observations, where an observation corresponds to a leaf in a decision tree.
- The knowledge discovery task requires that a substantial amount of either specialist knowledge on a particular problem or general chemical knowledge be used during induction.

The case-study was performed as part of a project concerned with developing rules for expert systems for chromatography [22] using machine induction techniques. A comparison of the application of ILP and DataMariner (a commercially available tool which generates essentially propositional rules) [4] to the domain of enantioseparations will form the subject of future published work.

4 Acknowledgements

The funding was provided by the EPSRC, under the remit of the Total Technology programme, and by Zeneca Pharmaceuticals. C.H.Bryant should like to thank Dr A.Srinivasan, from the Oxford University Computing Laboratory, for answering some of his questions on Golem.

References

[1] S. Muggleton, Inductive Logic Programming. *New Generation Computing*, 8 (1991) 295-318.

[2] S. Kocabas, A Review of Learning. *Knowledge Engineering Review*, 6 (1991) 195-222.

[3] J.W.Lloyd, *Foundations of Logic Programming*, Springer-Verlag, Berlin, 1984.

[4] C.H. Bryant, and A.E.Adam and D.R. Taylor and R.C. Rowe, Towards an Expert System for Enantioseparations: Induction of Rules Using Machine Learning, *Chemometrics and Intelligent Laboratory Systems*, 34 (1996) 21-40.

[5] W.F. Clocksin and C.S. Mellish, *Programming in Prolog*, 1^{st} Ed, Springer-Verlag, 1981.

[6] J.R. Quinlan, Induction of Decision Trees. *Machine Learning*, 1 (1986) 81-106.

[7] S. Muggleton, and L. De Raedt, Inductive Logic Programming: Theory and Methods. *Journal of Logic Programming*, 19, No.20 (1994) 629-679.

[8] N. Lavrac and L. De Raedt, Inductive Logic Programming: A Survey of European Research. *AI Communications*, 8, No.1 (1995) 3-19.

[9] C.J. Hogger, *Essentials of Logic Programming*, 1^{st} Ed, Oxford University Press, 1990.

[10] S. Muggleton, and C. Feng, Efficient Induction of Logic Programs. *in S. Arikawa and S. Goto and S. Ohsuga and T. Yokomori (Ed.s) Proc. 1^{st} Conference on Algorithmic Learning Theory, Japanese Society for Artificial Intelligence, Tokyo,* 1990, 368-381.

[11] Golem (version 1.0α) On-line Manual (1990).

[12] S. Dzeroski, S. Schulze-Kremer, K.R. Heidtke, K. Siems and D. Wettschereck, Applying ILP to Diterpene Structure Elucidation from ^{13}C

NMR Spectra. *Presented at a workshop in Bari, Italy on* 2^{nd} *July 1996 entitled 'Data Mining with Inductive Logic Programming' associated with the* 13^{th} *International Conference on Machine Learning.*

[13] I. Bratko and R. King Applications of Inductive Logic Programming. *SIGART Bulletin* 5, No.1 (1994) 43-49.

[14] I. Bratko, and S. Muggleton, Applications of Inductive Logic Programming, *Communications of the ACM* 38, No.11 (1995) 65-70.

[15] D.R. Taylor, and K. Maher, Chiral Separations by High-Performance Liquid Chromatography. *Journal of Chromatographic Science*, 30 (1992) 67-85.

[16] Personal Communication with Dr D.R.Taylor, of the Chemistry Department at the University of Manchester Institute of Science and Technology, UK, 1993.

[17] C. Roussel and P. Piras, CHIRBASE: A Molecular Database for Storage and Retrieval of Chromatographic Chiral Separations. *Pure & Applied Chemistry*, 65 (1993) 235-244.

[18] B. Koppenhoefer, A. Nothdurft, J. Pierrot-Sanders, P. Piras, C. Popescu, C. Roussel, M. Stiebler, and U. Trettin, CHIRBASE, a Graphical Molecular Database on the Separation of Enantiomers by Liquid-, Supercritical Fluid-, and Gas Chromatography. *Chirality*, 5 (1993) 213-219.

[19] B. Koppenhoefer, R. Graf, H. Holzschuh, A. Nothdurft, U. Trettin, P. Piras, and C. Roussel, CHIRBASE, a Molecular Database for the Separation of Enantiomers by Chromatography. *Journal of Chromatography*, 666 (1994) 557-563.

[20] S.T. Stauffer. *Expert System Shells in Chemistry: CHIRULE, a Chiral Chromatographic Column Selection System using Similarity Searching and Personal Construct Theory.* PhD Thesis. Virginia Polytech Ins. State Univ. USA. (1993)

[21] P. Jackson, *Introduction to Expert Systems.* 2^{nd} Ed., Addison-Wesley, 1990.

[22] C.H. Bryant, A.E. Adam, D.R. Taylor and R.C. Rowe, A Review of Expert Systems for Chromatography. *Analytica Chimica Acta*, 297 (1994) 317-347.

[23] C.H. Bryant, *Data Mining for Chemistry: the Application of Three Machine Induction Tools to a Database of Enantioseparations*. Ph.D. Thesis. University of Manchester Institute of Science and Technology, UK. (1996).

[24] W.H. Pirkle, and T.J. Sowin, Direct Liquid Chromatographic Separation of Phthalide Enantiomers. *Journal of Chromatography*, 387 (1987) 313-321.

[25] S. Muggleton, Inverse Entailment and Progol. *New Generation Computing*, 13, No.3-4 (1995) 245-286.

A Phthalide Data

The data on separations in this appendix was taken from a study performed by Pirkle and Sowin [24].

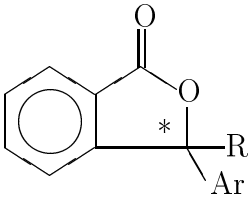
Phthalide			Separation Factor (α) (Selectivity)
Lable			
	Ar	R	
a	Phenyl	CH_2CH_3	1.00
b	4-Methoxyphenyl	CH_3	1.00
c	9-Anthryl	H	1.05
d	10-Methoxy-9-anthryl	H	1.05
e	1-naphthyl	H	1.09
f	1-naphthyl	CH_3	1.57
g	1-naphthyl	CF_3	1.15
h	1-[6,7-$(CH_3)_2$]-naphthyl	CH_3	2.03
i	1-[6,7-$(CH_3)_2$]-naphthyl	$CH(CH_3)_2$	2.89
j	1-[6,7-$(CH_3)_2$]-naphthyl	cyclohexyl	3.72
k	1-[3,7-$(CH_3)_2$]-naphthyl	CH_3	2.17
l	1-[3,7-$(CH_3)_2$]-naphthyl	phenyl	3.05
m	1-[4,7-$(CH_3)_2$]-naphthyl	H	1.15
n	1-[4,7-$(CH_3)_2$]-naphthyl	CH_3	2.06
o	1-[4,7-$(CH_3)_2$]-naphthyl	$(CH_2)_3CH_3$	2.39
p	1-[4,7-$(CH_3)_2$]-naphthyl	$(CH_2)_7CH_3$	2.55
q	1-[4,7-$(CH_3)_2$]-naphthyl	$C{\equiv}C\text{-}(CH_2)_5CH_3$	2.42
r	1-[4,7-$(CH_3)_2$]-naphthyl	$(CH_2)_9CHCH_2$	2.75
s	1-[2,3-$(CH_3)_2$]-naphthyl	CH_3	1.00
t	2-naphthyl	H	1.04
u	2-naphthyl	CH_3	1.03
v	2-[6,7-$(CH_3)_2$]-naphthyl	$CH(CH_3)_2$	1.00

Table 1: Enantioseparations of Phthalides on (R)-N-(3,5-dinitrobenzoyl)-phenylglycine.

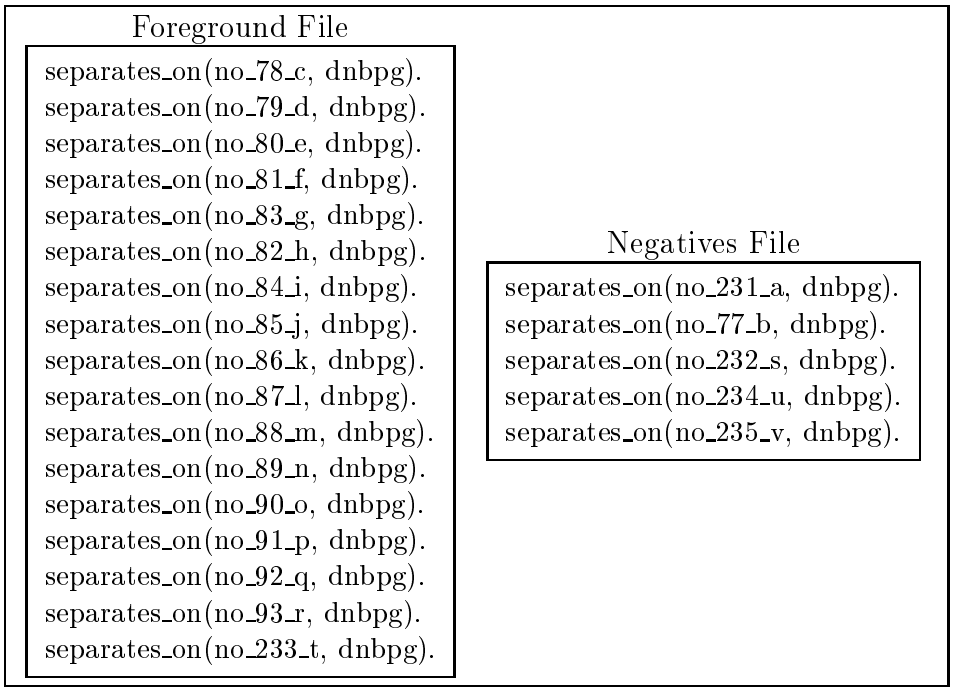


Figure 3: Foreground and Negatives Files Input to Golem.

```
!- set(i, 10).    !- set(j, 10).
greater_than(1, 0). greater_than(2, 0). greater_than(3, 0). greater_than(4, 0).
greater_than(5, 0). greater_than(6, 0). greater_than(7, 0). greater_than(8, 0).
greater_than(9, 0). greater_than(2, 1). greater_than(3, 1). greater_than(4, 1).
greater_than(5, 1). greater_than(6, 1). greater_than(7, 1). greater_than(8, 1).
greater_than(9, 1). greater_than(3, 2). greater_than(4, 2). greater_than(5, 2).
greater_than(6, 2). greater_than(7, 2). greater_than(8, 2). greater_than(9, 2).
greater_than(4, 3). greater_than(5, 3). greater_than(6, 3). greater_than(7, 3).
greater_than(8, 3). greater_than(9, 3). greater_than(5, 4). greater_than(6, 4).
greater_than(7, 4). greater_than(8, 4). greater_than(9, 4). greater_than(6, 5).
greater_than(7, 5). greater_than(8, 5). greater_than(9, 5). greater_than(7, 6).
greater_than(8, 6). greater_than(9, 6). greater_than(8, 7). greater_than(9, 7).
greater_than(9, 8).
```

Figure 4: Part of the Background File Input to Golem. The *i,j* settings and the greater_than(D1, D2) Prolog Facts.

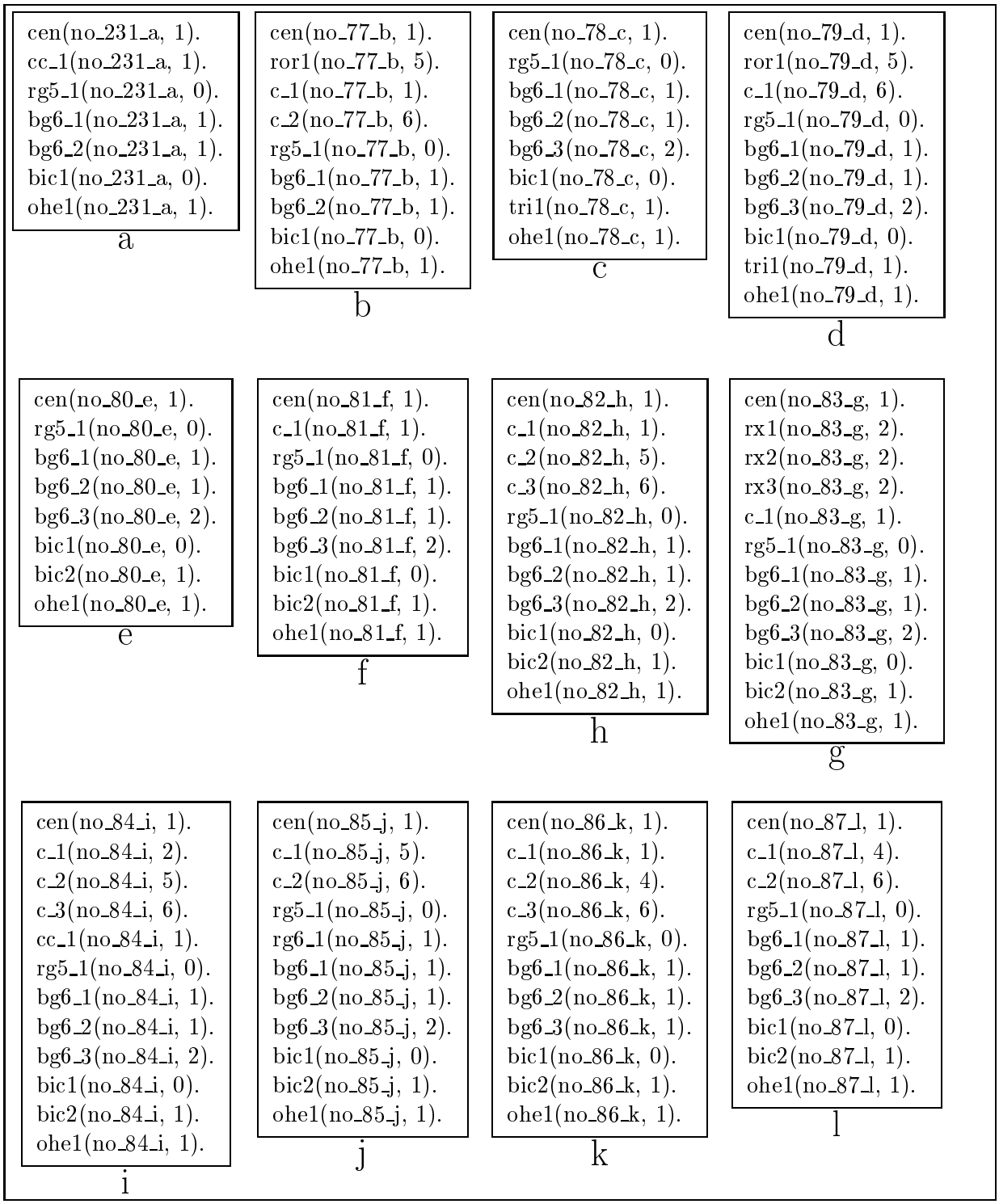


Figure 5: Part of the Background File Input to Golem. The <structural_feature>(E, D) Prolog Facts for the phthalides *a–l*.

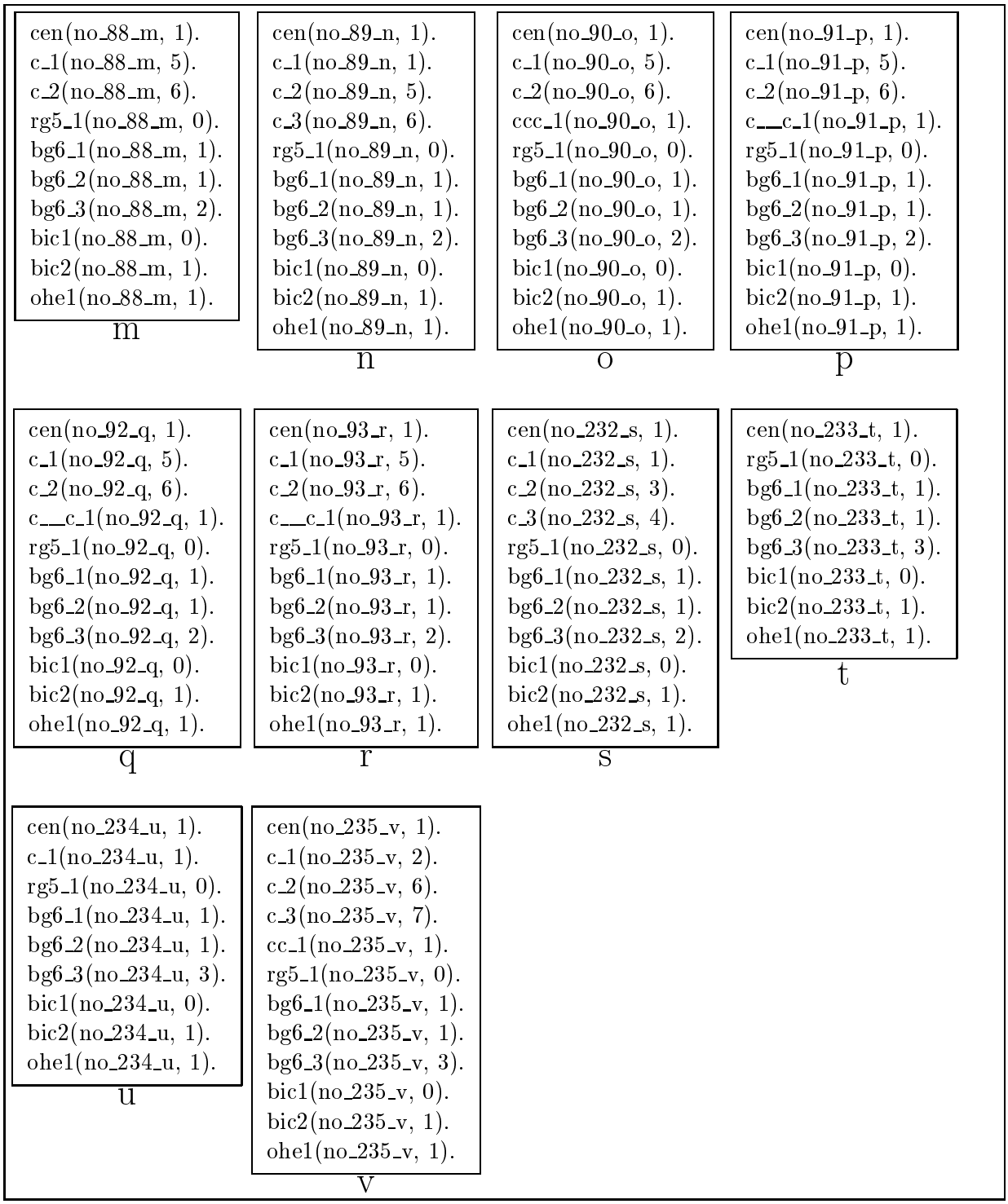


Figure 6: Part of the Background File Input to Golem. The <structural_feature>(E, D) Prolog Facts for phthalides *m–v*