Applying ILP to Diterpene Structure Elucidation from ^{13}C NMR Spectra

Sašo Džeroski[1,2], Steffen Schulze-Kremer[3],
Karsten R. Heidtke[4], Karsten Siems[4] and Dietrich Wettschereck[5]

[1] FORTH-ICS, P.O.Box 1385, 711 10 Heraklion, Crete, Greece
[2] Department of Intelligent Systems, Jožef Stefan Institute
Jamova 39, 1000 Ljubljana, Slovenia
Email: Saso.Dzeroski@ijs.si
[3] Max-Planck Institute for Molecular Genetics
Otto-Warburg-Laboratorium, Department Lehrach
Ihnestrasse 73, 14195 Berlin, Germany
[4] AnalytiCon GmbH
Gustav-Meyer-Allee 25, 13335 Berlin-Wedding, Germany
[5] GMD, FIT.KI, Schloss Birlinghoven, 53745 Sankt Augustin, Germany

Abstract. We present a novel application of ILP to the problem of diterpene structure elucidation from ^{13}C NMR spectra. Diterpenes are organic compounds of low molecular weight that are based on a skeleton of 20 carbon atoms. They are of significant chemical and commercial interest because of their use as lead compounds in the search for new pharmaceutical effectors. The structure elucidation of diterpenes based on ^{13}C NMR spectra is usually done manually by human experts with specialized background knowledge on peak patterns and chemical structures. In the process, each of the 20 skeletal atoms is assigned an atom number that corresponds to its proper place in the skeleton and the diterpene is classified into one of the possible skeleton types. We address the problem of learning classification rules from a database of peak patterns for diterpenes with known structure. Recently, propositional learning was successfully applied to learn classification rules from spectra with assigned atom numbers. As the assignment of atom numbers is a difficult process in itself (and possibly indistinguishable from the classification process), we apply ILP, i.e., relational learning, to the problem of classifying spectra without assigned atom numbers.

1 Introduction

1.1 NMR

Structure elucidation of compounds isolated from plants, fungi, bacteria or other organisms is a common problem in natural product chemistry. There are many useful spectroscopic methods of getting information about chemical structures, mainly nuclear magnetic resonance (NMR) and mass spectroscopy. The interpretation of these spectra normally requires specialists with detailed spectroscopic knowledge and experience in natural products chemistry. NMR-spectroscopy is

the best method for complete structure elucidation (including stereochemistry) of non-crystalline samples. For structure elucidation of secondary natural products (not proteins) only ^{1}H-NMR- and ^{13}C-NMR-spectroscopy, including combined methods such as 2D-NMR-spectroscopy, are important because hydrogen and carbon are the most abundant atoms in natural products. In structure elucidation of peptides and proteins ^{15}N-NMR is sometimes helpful [2].

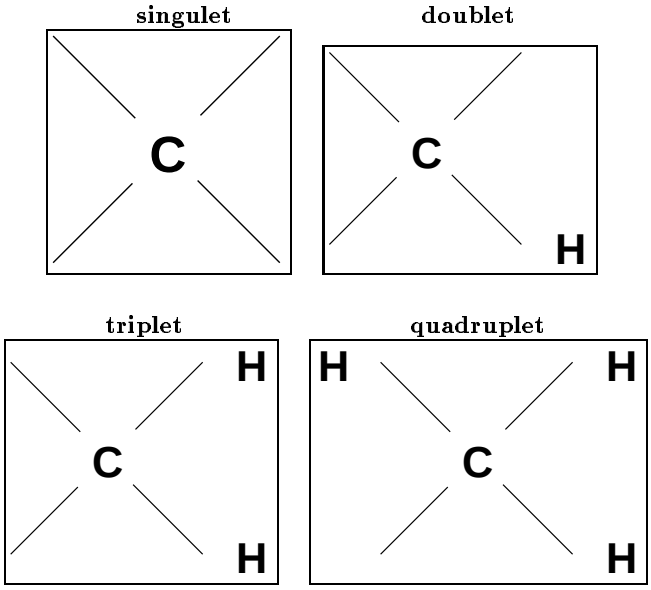

Fig. 1. Multiplicity of Carbon Atoms.

^{1}H-NMR- and ^{13}C-NMR-spectroscopy are quite different: in a ^{13}C-NMR-spectrum every carbon atom occurs as a separate signal in most cases, while in ^{1}H-NMR-spectra many signals overlap and are therefore difficult to interpret [17]. ^{1}H-NMR- spectra are logically decomposable and the specialist could get direct information about the structure (including stereochemistry) from the resonance frequency and shape of the signals, provided a sufficiently high resolution of resonance signals can be experimentally achieved. In ordinary ^{13}C-NMR-spectra only the resonance frequency is observed and no signal splitting (decomposition) is possible. However, the absolute value of the resonance frequency provides more information about chemical structure and (important for prediction and database maintenance purposes) is not as sensitive to varying experimental conditions (such as the magnetic strength of the NMR-spectrometer or the type of solvent) as ^{1}H-NMR-spectroscopy is.

Additional measurements can be used to determine the number of hydrogens directly connected to a particular carbon atom. This number is the so-called multiplicity of the signal: `s` stands for singulet, which means there is no proton (i.e., hydrogen) connected to the carbon; `d` stands for a doublet with one proton connected to the carbon; `t` stands for a triplet with two protons and `q` for a quartet with three protons bound to the carbon atom. Figure 1 shows each of the four situations in the order listed above. Because of the simpler nature of ^{13}C-NMR-data as compared to ^{1}H-NMR-data, the former are easier to handle and therefore remain the preferred basis for automatic structure elucidation [9].

1.2 Diterpenes

Diterpenes are one of a few fundamental classes of natural products with about 5000 members known [13]. The skeleton of every diterpene contains 20 carbon atoms. Sometimes there are additional groups linked to the diterpene skeleton by an oxygen atom with the possible effect of increasing the carbon atom count to more than 20 per diterpene. About 200 different diterpene skeletons are known so far, but some of them are only represented by one member compound. Most of the diterpenes belong to one of 20 common skeleton types.

The problem of structure elucidation of diterpenes requires knowledge about biosynthesis, the way in which biological organisms synthesize natural products. Not every combination of carbons, protons and oxygens that is feasible from a chemical point of view actually occurs in nature, as biosynthesis in biological organisms is limited to a characteristic subset of chemical reactions that constrain the structure space. Structure elucidation of diterpenes from ^{13}C-NMR-Spectra can be separated into three main stages: 1) identification of residues (ester and/or glycosides), 2) identification of the diterpene skeleton, and 3) arrangement of the residues on the skeleton.

This work deals with the second stage, the identification of the skeleton. A skeleton is a unique connection of carbon atoms each with a specific atom number and, normalized to a pure skeleton molecule without residues, a certain multiplicity (s, d, t or q). Figure 2 shows the ^{1}H-NMR- and ^{13}C-NMR-Spectra of a diterpene belonging to the skeleton type Labdan. The structure of the diterpene is also depicted, giving the arrangement and numbering of the 20 skeletal carbon atoms.

The task we address is to identify the skeleton (type) of diterpenoid compounds, given their ^{13}C-NMR-Spectra that include the multiplicities and the frequencies of the skeleton atoms. This task is usually done manually by human experts with specialized background knowledge on peak patterns and chemical structures. In the process, each of the 20 skeletal atoms is assigned an atom number that corresponds to its proper place in the skeleton and the diterpene is classified into one of the possible skeleton types. We address the problem of learning classification rules from a database of peak patterns for diterpenes with known structure. Recently, propositional methods were successfully applied to classifying spectra with assigned atom numbers [7]. As the assignment of atom

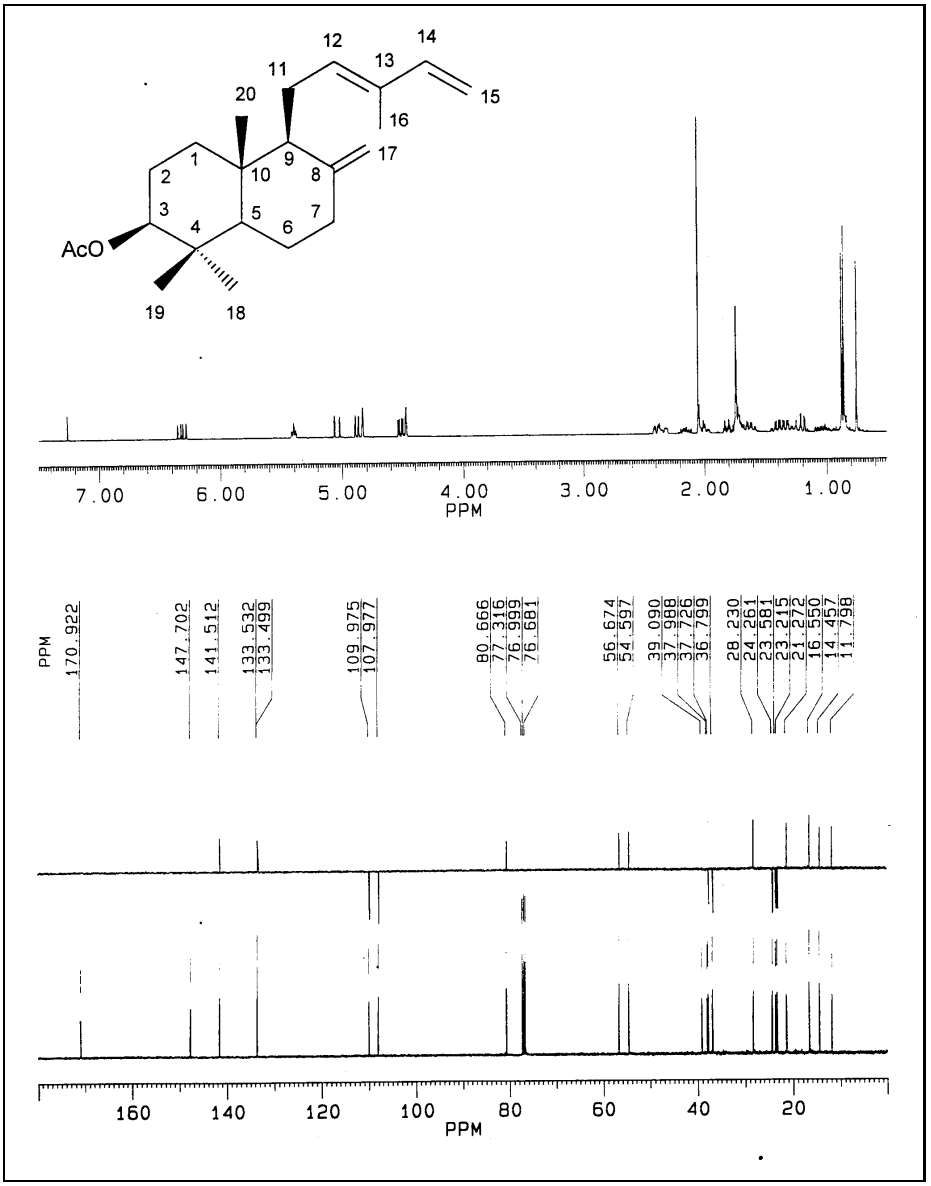

Fig. 2. 1H-NMR and 13C-NMR Spectra of a diterpene belonging to the skeleton type Labdan, with acetyl as a residue. The additional measurements (above the 13C-NMR Spectrum) show 25 signals (s=5, d=5, t=7, q=5). At positions 76.681, 76.999, and 77.316 is the signal of the solvent.

Table 1. Facts describing the 13-C-NMR spectrum of a particular molecule that belongs to skeleton type Labdan (52).

```
nmr(v1,52,v1_1,1,t,39.00).
nmr(v1,52,v1_2,2,t,19.30).
nmr(v1,52,v1_3,3,t,42.20).
nmr(v1,52,v1_4,4,s,33.30).
nmr(v1,52,v1_5,5,d,55.60).
nmr(v1,52,v1_6,6,t,24.30).
nmr(v1,52,v1_7,7,t,38.30).
nmr(v1,52,v1_8,8,s,148.60).
nmr(v1,52,v1_9,9,d,57.20).
nmr(v1,52,v1_10,10,s,39.70).
nmr(v1,52,v1_11,11,t,17.60).
nmr(v1,52,v1_12,12,t,41.40).
nmr(v1,52,v1_13,13,s,73.50).
nmr(v1,52,v1_14,14,d,145.10).
nmr(v1,52,v1_15,15,t,111.40).
nmr(v1,52,v1_16,16,q,27.90).
nmr(v1,52,v1_17,17,t,106.30).
nmr(v1,52,v1_18,18,q,33.60).
nmr(v1,52,v1_19,19,q,21.60).
nmr(v1,52,v1_20,20,q,14.40).
```

numbers is a difficult process in itself (and possibly indistinguishable from the classification process), we apply ILP, i.e., relational learning, to the problem of classifying spectra without assigned atom numbers.

The rest of the paper is organized as follows: Section 2 describes the database and the pre-processing of the data, then summarizes recent work on applying propositional learning in this domain that uses data with assigned atom numbers. When the atom numbers are not available, but only multiplicities and frequencies, we have an ILP problem. Section 3 describes several formulations of this problem and gives the results of applying the FOIL ILP system to these formulations. Section 4 gives a comparison to C4.5, nearest neighbor classification and backpropagation networks applied to different propositionalizations of the problem. Section 5 presents the results of applying RIBL, a relational instance-based learner to the ILP problem formulations. Finally, Section 6 concludes with a discussion and an outline of directions for further work.

2 The data and propositional learning results

2.1 The database

AnalytiCon GmbH maintains a database of diterpenoid compounds. This is done using the ISIS (Integrated Scientific Information System), a product of MDL Information Systems, Inc., on IBM PCs under MS Windows. The relational

database contains information on 1503 diterpenes with known structure, stored in three relations - `atom`, `bond`, and `nmr`.

The first relation specifies to which element an atom in a given compound belongs. The second relation specifies which atoms are bound and in what way in a given compound. The `nmr` relation stores the measured ^{13}C-NMR-Spectra. For each of the 20 carbon atoms in the diterpene skeleton, it contains the atom number, its multiplicity and frequency. For each compound, the skeleton type which it represents is also specified within the database. Table 1 gives an excerpt from the `nmr` relation describing molecule `v1`. The relation schema is `nmr(MoleculeID,SkeletonType,AtomID,AtomNumber,Multiplicity,Frequency)`.

2.2 Pre-processing

Every substituent or residue connected to a carbon atom exerts a characteristic shift on the resonance frequency signal of the carbon atom and sometimes also changes its multiplicity. Simple rules based on expert knowledge can be used to take this effect into account.

Table 2. Rules for generating reduced multiplicities from observed multiplicities.

```
IF ObservedM = s AND Frequency in [ 64.5, 95 ] THEN ReducedM = d
IF ObservedM = s AND Frequency in [ 96, 114 ]  THEN ReducedM = t
IF ObservedM = s AND Frequency in [ 115, 165 ] THEN ReducedM = d
IF ObservedM = s AND Frequency in [ 165, 188 ] THEN ReducedM = q
IF ObservedM = s AND Frequency in [ 188, inf ] THEN ReducedM = t
IF ObservedM = d AND Frequency in [ 64.5, 95 ] THEN ReducedM = t
IF ObservedM = d AND Frequency in [ 105, 180 ] THEN ReducedM = t
IF ObservedM = d AND Frequency in [ 96, 104 ]  THEN ReducedM = q
IF ObservedM = d AND Frequency in [ 180, inf ] THEN ReducedM = q
IF ObservedM = t AND Frequency in [ 59, 90 ]   THEN ReducedM = q
IF ObservedM = t AND Frequency in [ 90, inf]   THEN ReducedM = q
```

They transform the raw, measured, NMR-multiplicities into the so-called reduced multiplicities, which carry more information about the skeleton types, as shown below. These rules are given in Table 2. They leave the measured frequencies unchanged. The reduced multiplicities that correspond to those in Table 1 are given in Table 3. Note that the multiplicities of atoms 8, 13, 14, 15, and 17 are changed.

2.3 Propositional learning: experiments and results

Given the above data, Džeroski et al. [7] formulate several propositional learning problems. Twenty-three different skeleton types are represented in the whole set of 1503 compounds: there are thus 23 possible class values. The attributes are

Table 3. Pre-processed facts describing the 13-C-NMR spectrum of molecule `v1`.

```
red(v1,52,v1_1,1,t,39.00).
red(v1,52,v1_2,2,t,19.30).
red(v1,52,v1_3,3,t,42.20).
red(v1,52,v1_4,4,s,33.30).
red(v1,52,v1_5,5,d,55.60).
red(v1,52,v1_6,6,t,24.30).
red(v1,52,v1_7,7,t,38.30).
red(v1,52,v1_8,8,d,148.60).
red(v1,52,v1_9,9,d,57.20).
red(v1,52,v1_10,10,s,39.70).
red(v1,52,v1_11,11,t,17.60).
red(v1,52,v1_12,12,t,41.40).
red(v1,52,v1_13,13,d,73.50).
red(v1,52,v1_14,14,t,145.10).
red(v1,52,v1_15,15,q,111.40).
red(v1,52,v1_16,16,q,27.90).
red(v1,52,v1_17,17,q,106.30).
red(v1,52,v1_18,18,q,33.60).
red(v1,52,v1_19,19,q,21.60).
red(v1,52,v1_20,20,q,14.40).
```

the multiplicities and frequencies of each of the 20 skeleton atoms. For instance, attribute `A1M` is the multiplicity of atom number 1, and `A7F` is the frequency of atom number 7. There are thus 40 attributes, twenty discrete (multiplicities) and 20 continuous (frequencies). At a suggestion from the domain experts, a version of the problem is also considered where only the multiplicities are used as attributes. Four series of experiments are performed, with the following sets of attributes: observed multiplicities and frequencies, reduced multiplicities and frequencies, observed multiplicities only and reduced multiplicities only.

Three different machine learning approaches were used: backpropagation networks [19, 18], nearest neighbor classification [4, 20], and decision tree induction [14, 16]. Table 4 gives a summary of the accuracies on unseen cases for the three approaches and the four different problems as estimated by ten-fold cross-validation.

Table 4. Classification accuracy on unseen cases when learning from classified 13C-NMR spectra with assigned atom numbers.

Problem/Approach	Backpropagation networks	Nearest neighbor	C4.5
Observed	89.6%	96.7%	96.4%
Reduced	95.5%	99.3%	97.6%
Observed - No Frequencies	83.9%	94.4%	93.1%
Reduced - No Frequencies	97.1%	98.8%	98.4%

While the accuracies achieved are very high and the problem may be considered solved at first sight, one should bear in mind that the propositional formulation specified above crucially depends on the atom number information. The assignment of atom numbers is a very difficult and important part of the classification process, and classification rules that do not rely on atom number assignments need to be derived for practical purposes. Without atom number information, there is no obvious propositional representation of the problem and more careful representation engineering (feature construction) or the use of more powerful techniques, such as ILP seems necessary. In the remainder of the paper, we describe experiments that apply each of these approaches separately, as well as their combination.

Table 5. The classes, their chemical names and number of instances in the database.

Code	Chemical name	Number of instances
c2	Trachyloban	9
c3	Kauran	353
c4	Beyeran	72
c5	Atisiran	33
c15	Ericacan	2
c18	Gibban	13
c22	Pimaran	155
c28	6,7-seco-Kauran	9
c33	Erythoxilan	9
c36	Spongian	10
c47	Cassan	12
c52	Labdan	448
c54	Clerodan	356
c71	Portulan	5
c79	5,10-seco-Clerodan	4
c80	8,9-seco-Labdan	6

3 Applying FOIL

As we are dealing with a learning problem where 23 different classes exist, there are 23 target relations `classC(MoleculeID)`, where `C` ranges over the possible classes. For example, the target relation `class52(MoleculeID)` corresponds to the skeleton type Labdan, most common among the 1503 diterpenes in the database (448 instances). The correspondence between the chemical names and the codes in the data is given in Table 5 together with the number of instances of each class. Only classes with more than one instance are listed.

3.1 Using multiplicities and frequencies

Given that much better results were obtained using the reduced multiplicities in earlier experiments, we decided to use only these (and not the observed multi-

plicities) for our experiments with ILP. After one takes the atom number information away, the background relation `red` can be simplified to a three argument relation `red(MoleculeID,Multiplicity,Frequency)`. A particular fact of this relation states that the 13C-NMR spectrum of a given molecule contains a peak of a given multiplicity and frequency. For example, the fact `red(v1,t,39.00)` states that the 13C-NMR spectrum of molecule `v1` has a peak at frequency 39.00 ppm with multiplicity `t` (a triplet).

The background relation `red` is nondeterminate, as there are 20 tuples for each molecule. This prevents the use of ILP systems like GOLEM [12] or DINUS [10]. While PROGOL [11] would be applicable, preliminary experiments showed it has prohibitive time complexity if longer rules/clauses are needed. Therefore, we used FOIL [15], in particular FOIL6.4. Except for a variable depth limit of one, all other settings were left in their default state.

We first used FOIL to induce rules on the entire data set, and then performed ten-fold cross validations on the same partitions of training examples as used for the propositional case by Džeroski et al. [7]. For each experiment, FOIL was run 23 times, i.e., once for each target relation. The rules from the 23 runs were then taken together to produce a rule set. Before classification, the rules from the rule set were checked against the training set to record the number of examples of each class that they cover, as well as to note the majority class in the training set. This information is then used when classifying new examples, e.g., when testing the accuracy of the rule set.

For classification, we implemented a procedure analogous to that of CN2 [3]: when an example is classified, all clauses that cover it are taken into account. The distributions of examples of each class are summed for all rules that match and the majority class is assigned to the example. If no rule matches, the majority class (from the training set) is assigned.

Given the background relation `red(MoleculeID,Multiplicity,Frequency)`, FOIL induced 90 rules from the entire data set, comprising 607 literals in their bodies. At first sight, the rules are quite specific, covering relatively few examples. Also, many examples are left uncovered. Thus, even on the training set, these rules only achieve 51.6% accuracy. The ten-fold cross-validation yields 46.5% accuracy on unseen cases as the average over the ten runs. It seems that the background knowledge is sufficient to distinguish among the different classes (the rules typically cover examples of one class only), but FOIL induces too specific rules. The most general rule `class52(A) :- red(A,d,B), B=<73.7, red(A,C,D), D>38.5, D=<44.6, B>73.2` covers 43 examples of class 52.

3.2 Using multiplicities only

According to the domain expert (Karsten Siems), the multiplicities should suffice for correct classification, at least when atom numbers are available [7]. If we remove the `Frequency` argument from the relation `red`, all the information left about a particular molecule is captured by the number of atoms which have multiplicity `s`, `d`, `t`, and `q`, respectively. We store this information in the relation

`prop(MoleculeID,SAtoms,DAtoms,TAtoms,QAtoms)`. For our molecule `v1`, we have the fact `prop(v1,2,4,8,6)`.

Given the entire data set, the 23 target relations and the background relation `prop`, FOIL induced 17 rules with 52 literals in the bodies. Same settings were applied in FOIL as for the experiments with `red`. The rules induced in this case are much more general than the ones obtained with `red`. The most general rule `class52(A) :- prop(A,B,C,D,E), E>5, C>3, D>7, B>1` covers 429 examples (of which 355 of class 52, 72 of class 54). Many rules cover examples of several classes. It seems that the background knowledge is insufficient to completely distinguish among the different classes and FOIL is forced to induce overly general rules. This, however, has positive effect on accuracy as compared to the experiments with `red`. Using `prop`, FOIL achieves 69.0% accuracy on the entire data sets, and 70.1% accuracy on unseen cases (ten-fold cross-validation).

At this point, note that we have in fact applied FOIL to a propositional problem. Namely, from the nondeterminate representation with the `red` relation, we have constructed a four-feature representation of each molecule. We postpone the comparison to propositional learners to the next section.

3.3 Combining engineered features and a relational representation

Having introduced the four new features with the `prop` relation, which seem to capture some general properties of the molecules, we repeated the experiment with the nondeterminate relational representation. This time FOIL was given both the relation `red` and the relation `prop`. The same settings of FOIL were applied as in the previous two subsections.

Given the entire data set, the 23 target relations and the background relations `red` and `prop`, FOIL induced 68 rules with 401 literals in the bodies. The rules were more general than rules induced using `red` only, but were more specific than rules induced using `prop` only. In most cases, each rule covers examples of one class only. The most general rule `class52(A) :- prop(A,B,C,D,E), E>5, C>3, D>7, B>1, red(A,d,F), F>54.8, F=<72.3` covers 227 examples of the correct class. The induced rules achieve 83.2% accuracy on the entire data set and 78.3% accuracy on unseen cases (ten-fold cross-validation). Combining the engineered features with the relational representation thus has a positive effect.

4 Comparing FOIL to propositional approaches

4.1 C4.5 using multiplicities only

As mentioned above, the relation `prop` defines a propositional version of our classification problem. We therefore applied a propositional learner, C4.5 [16], to this problem. The same experimental setting (tree induced on whole data set first, then ten-fold cross-validation) and the default settings of C4.5 were used.

The induced tree achieves 80.4% accuracy on the entire data set. The leaves of the tree typically contain examples of several different classes, confirming

the suspicion that the four features do not suffice for completely correct classification. The classification accuracy on unseen cases as measured by ten-fold cross-validation (same folds as above) is 78.5%, which is almost the same as the accuracy achieved by FOIL using both `red` and `prop`.

4.2 Nearest neighbor using multiplicities only

We also applied nearest neighbor classification [4, 20] to the propositional problem defined by the relation `prop`. The same experimental setting (cross-validation folds) was used. Training on the entire dataset gives 100% accuracy.

Cross-validation on the number of neighbors used in classification was tried out, as well as two different forms of feature weighting, but the basic nearest neighbor method performed best. The classification accuracy on unseen cases (average over the ten folds) is 79.0%, which is almost the same as the accuracy of C4.5 and the accuracy achieved by FOIL using both `red` and `prop`.

4.3 Backpropagation networks

Various network architectures were explored to see how in comparison backpropagation networks would perform at the classification of skeletons based on unassigned 13C NMR data. This was done independently of the above experimental setup, but using the same partitions for cross-validation.

A standard backpropagation network [19, 18] with 960 input neurons, no hidden units and 23 output units was trained with the same input data as for the ILP experiments. We divided the 960 input neurons into four equally large sets of 240 nodes, one each for singulets, doublets, triplets and quadruplets. The best representation was to have for each multiplicity 24 frequency intervals (0 - 10 ppm, 11 - 20 ppm, ..., 231 - 240 ppm) with 10 nodes each and to feed the value of 1 for each frequency in the spectrum into the next unoccupied input neuron of the appropriate interval. All remaining input nodes are filled with zeros. Thus, the artificial neural net sees a discretized distribution of multiplicity signals.

During cross-validation, accuracy on the training set (average for the 10 runs) reached 97.9%, while accuracy on unseen cases reached 79.9%. Other network variations, e.g., feeding the actual frequency value into the appropriate neuron instead of 1 or using a 4 x 15 architecture that receives the sorted frequencies for each multilicity as input did not produce better results.

5 Applying RIBL

RIBL (relational instance-based learning) [8] generalizes the nearest neighbor method to a relational representation. RIBL first constructs cases by putting together all facts that relate to (in this case) a single molecule. Training cases are stored for further reference. When a new molecule has to be classified, RIBL calculates the similarities between it and each of the training cases, then assigns it the class of the nearest training case.

The similarity measure used in RIBL is a generalization of similarity measures used in propositional instance-based learners. In fact, given a propositional problem, RIBL becomes the classical nearest neighbor method and has the same performance as the latter. This is a very desirable property for a relational learner.

We used the same formulations and experimental methodology as for FOIL (Section 3). As it stores all training cases, RIBL achieves 100% accuracy when the entire data set is given for both training and testing. For estimating accuracy on unseen cases, ten-fold cross validations on the same partitions as above were perfomed.

Table 6. Accuracies of different approaches when classifying NMR spectra without assigned atom numbers.

Problem/System	FOIL	C4.5	RIBL
red	46.5%	NA	86.5%
prop	70.1%	78.5%	79.0%
red+prop	78.3%	NA	91.2%

Table 6 gives the accuracies on unseen cases achieved FOIL, C4.5 and RIBL on the three different formulations of the problem. Given only the `red` relation, RIBL achieved 86.5% classification accuracy (average over the ten partitions) on unseen cases. This is an increase of 40% over the accuracy achieved by FOIL. Note that propositional approaches (C4.5) are not applicable to this formulation of the problem.

Given only the `prop` relation, RIBL behaves identically to the nearest neighbor method, thus yielding 78.5% accuracy on unseen cases, a performance equivalent to that of C4.5. When provided with both the `red` and the `prop` relations, RIBL achieves 91.2% accuracy on unseen cases. Using the engineered features improves RIBL's performance by roughly 5%, pushing further the best result achieved at classifying diterpen NMR spectra without assigned atom numbers. Again, propositional approaches (C4.5) are not applicable to this formulation of the problem.

6 Discussion

For practical purposes, ^{13}C-NMR spectra of diterpenes without assigned atom numbers have to be classified. This is a problem that is not straigthforwardly transformable to propositional form and calls for either representation engineering or the use of inductive logic programming. We explored both approaches separately and in combination. Adding the engineered features to the natural relational representation improved the performance of both relational learning systems used.

Using the engineered features only, propositional approaches (in particular C4.5, nearest neighbor and neural networks) achieve around 79% accuracy on unseen cases. This is roughly 20% less than the accuracies achieved when classifying ^{13}C-NMR spectra of diterpenes with assigned atom numbers.

Using FOIL on the natural relational representation yields unsatisfactory results. Combining the relational representation with the engineered features greatly improves FOIL's performance. However, given the engineered features only FOIL performs much worse than C4.5, so that the best performance of FOIL (combining the relational representation and the engineered features) is comparable to that of C4.5. The reason for FOIL's poor performance is that the rules induced are overly specific as indicated by their coverage and confirmed by expert comments. However, the rules found by FOIL are quite short, indicating that the problem lies in the search heuristic: it directs the search to short rules with small coverage, despite the fact that longer rules with higher coverage exist.

From the above it is clear that a desirable property of relational learning systems is the following: given a propositional problem, a relational learning system should perform comparably to propositional systems. RIBL, which extends the nearest neighbor approach to a relational framework has this property. Given the engineered features only, it achieves 79% accuracy on unseen cases. Given the relational representation only, RIBL performs much better than FOIL (86% vs 46% accuracy on unseen cases). Finally, combining the relational representation and the engineered features it achieves 91% accuracy on unseen cases, 11% better than the best propositional result of 80% (backpropagation networks with 960 features).

The 91% accuracy achieved by RIBL is in the range of the accuracies with which experts classify diterpenes into skeleton types given ^{13}C NMR spectra only. That number can actually only be estimated since it is expensive to have an expert carry out a statistically significant number of structure predictions without using other additional information that often becomes available from heterogeneous sources (such as literature, and ^{1}H NMR spectra). This basically means that ^{13}C NMR is not completely sufficient for classifying diterpenes and that great improvements of classification accuracy are not to be expected.

The main direction for further work thus is to provide classification accuracy at the level already achieved in conjunction with satisfactory explanation. RIBL can offer only the neirest neighbor used to classify a given instance as an explanation of that classification, but no general knowledge can be offered for inspection by domain experts. Newer versions of RIBL (based on Aha's IB3 [1]), which store only a fraction of all train cases, may offer a small number of prototypes as explanations. Alternatively, one might apply mFOIL [6, 10] and use the m-estimate to guide the search towards more general rules (larger m in the estimate of the accuracy prefers rules that cover more examples). ICL [5] is also an interesting candidate ILP system to apply to this problem.

Acknowledgements

Sašo Džeroski is an ERCIM (European Research Consortium for Informatics and Mathematics) fellow at ICS-FORTH. This work started during his visit to GMD (German National Research Center for Information Technology) supported by the same ERCIM fellowship. This work is also supported in part by the ILP2 project (ESPRIT IV LTR Project 20237 Inductive Logic Programming 2).

References

1. Aha, D., Kibler, D., and Albert, M. Instance-based learning algorithms. *Machine Learning*, 6: 37–66, 1991.
2. Abraham, R.J., Loftus, P. *Proton and Carbon 13 NMR Spectroscopy, An Integrated Approach.* Heyden, London, 1978.
3. Clark, P. and Boswell, R. Rule induction with CN2: Some recent improvements. In *Proc. Fifth European Working Session on Learning*, pages 151–163. Springer, Berlin, 1991.
4. Cover, T.M., and Hart, P.E. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13: 21–27, 1968.
5. De Raedt, L., and Van Laer, V. Inductive constraint logic. In *Proc. Sixth International Workshop on Algorithmic Learning Theory*, pages 80–94. Springer, Berlin, 1995.
6. Džeroski, S. Handling imperfect data in inductive logic programming. In *Proc. Fourth Scandinavian Conference on Artificial Intelligence*, pages 111–125. IOS Press, Amsterdam, 1993.
7. Džeroski, S., Schulze-Kremer, S., Heidtke, K., Siems, K., Wettschereck, D. Diterpene structure elucidation from ^{13}C NMR spectra with machine learning. In *Proc. ECAI'96 Workshop on Intelligent Data Analysis in Medicine and Pharmacology*, 1996.
8. Emde, W., Wettschereck, D. Relational instance-based learning. In *Proc. Thirteenth International Conference on Machine Learning*, pages 122–130. Morgan Kaufmann, San Mateo, CA, 1996.
9. Gray, N. A. B. *Progress in NMR-spectroscopy, Vol. 15*, pp. 201–248, 1982.
10. Lavrač, N., Džeroski, S. *Inductive Logic Programming: Techniques and Applications.* Ellis Horwood, Chichester, 1994.
11. Muggleton, S. Inverse entailment and PROGOL. *New Generation Computing*, 13: 245–286, 1995.
12. Muggleton, S., and Feng, C. Efficient induction of logic programs. In *Proc. First Conference on Algorithmic Learning Theory*, pages 368–381. Ohmsha, Tokyo, 1990.
13. *Natural products on CD-ROM.* Chapman and Hall, London, 1995.
14. Quinlan, J.R. Induction of decision trees. *Machine Learning* 1(1): 81–106, 1986.
15. Quinlan, J.R. Learning logical definitions from relations. *Machine Learning*, 5(3): 239–266, 1990.
16. Quinlan, J.R. *C4.5: Programs for Machine Learning.* Morgan Kaufmann, San Mateo, CA, 1993.
17. Schulze-Kremer, S. *Molecular Bioinformatics - Algorithms and Applications.* de Gruyter, Berlin, 1995.
18. *Stuttgart Neural Network Simulator.* Computer code available from the University of Stuttgart, Germany, via anonymous ftp `ftp://ftp.informatik.uni-stuttgart.de/pub/SNNS`, 1995.
19. Tveter, D. R. *Fast-Backpropagation.* Computer code available from the author. Address: 5228 N Nashville Ave, Chicago, Illinois, 60656, `drt@chinet.chi.il.us`, 1995.
20. Wettschereck, D. A study of distance-based machine learning algorithms. PhD Thesis, Department of Computer Science, Oregon State University, Corvallis, OR, 1994.

This article was processed using the LaTeX macro package with LLNCS style