The utility of background knowledge in learning medical diagnostic rules.

Article *in* Applied Artificial Intelligence · July 1993

DOI: 10.1080/08839519308949989 · Source: DBLP

Nada Lavrac
Jožef Stefan Institute
409 PUBLICATIONS **10,230** CITATIONS

Sašo Džeroski
Jožef Stefan Institute
549 PUBLICATIONS **14,635** CITATIONS

□ THE UTILITY OF BACKGROUND KNOWLEDGE IN LEARNING MEDICAL DIAGNOSTIC RULES

NADA LAVRAČ, SAŠO DŽEROSKI, VLADIMIR PIRNAT, and VILJEM KRIŽMAN
Jožef Stefan Institute, Jamova 39, 61111 Ljubljana, Slovenia

Inductive learning algorithms have frequently been applied to the problem of learning medical diagnostic rules. Most learning algorithms use an attribute-value language to describe training examples and induced rules. Consequently, the background knowledge that can be used in the learning process is of a very restricted form. To overcome these limitations, the inductive learning system LINUS incorporates attribute-value learners into a more powerful logic programming framework in which background knowledge can be used effectively. This paper describes the application of LINUS to the problem of learning rules for early diagnosis of rheumatic diseases. In addition to the attribute-value descriptions of patient data, LINUS was given background knowledge provided by a medical specialist. Medical evaluation of the rules induced by LINUS using the CN2 attribute-value learner and measurements of their performance in terms of classification accuracy and information content show that the use of background knowledge substantially improves the quality of induced rules.

INTRODUCTION

Inductive learning algorithms have frequently been applied to the problem of learning medical diagnostic rules. Most learning algorithms use an attribute-value language to describe training examples and induced concept descriptions. Consequently, the background knowledge that can be used in the learning process is of a very restricted form. This implies that many learning tasks cannot be solved by attribute-value learning algorithms such as the members of the AQ (Michalski et al., 1986) and TDIDT (Top Down Induction of Decision Trees; Quinlan, 1986) families of inductive learners.

To overcome these limitations, the inductive learning system LINUS (Lavrač et al., 1991a; Lavrač and Džeroski, 1992, 1993) incorporates various attribute-value learners into a more powerful logic programming framework in which background knowledge can be used effectively. On the one hand, LINUS can be used as an attribute-value learner enhanced with the effective use of background knowledge. On the other hand, it can also be used to induce relational descriptions. As such, LINUS belongs to the family of inductive logic programming (ILP) systems (Muggleton, 1992), which induce concept descriptions in the form of logic programs.

This paper describes the application of LINUS to the problem of learning rules for early diagnosis of rheumatic diseases. Correct diagnosis in an early stage of a

Applied Artificial Intelligence: 7:273–293, 1993

0883-9514/93 $10.00 + .00

rheumatic disease is a hard problem. Having passed all the investigations, many patients cannot be diagnosed reliably after their first visit to a specialist. The reason for this is that symptoms, clinical manifestations, and laboratory and radiological findings for various rheumatic diseases are similar and not specific. Diagnosis can also be incorrect because of the subjective interpretation of anamnestic, clinical, laboratory, and radiological data (Pirnat et al., 1989). This diagnostic domain was used in earlier experiments (Pirnat et al., 1989; Karalič and Pirnat, 1990) with the inductive learning system ASSISTANT (Cestnik et al., 1987), a member of the TDIDT family.

In the application described in this paper, LINUS used the CN2 (Clark and Boswell, 1991) attribute-value learner to learn diagnostic rules from the anamnestic data of patients with rheumatic diseases. In this real-life medical problem, the use of CN2 was appropriate because it can deal with incomplete (missing) and erroneous (noisy) data. In addition to the attribute-value descriptions of patient data, LINUS was given background knowledge provided by a medical specialist in the form of typical co-occurences of symptoms.

This study shows how the noise-handling mechanisms of CN2 and the ability of LINUS to use background knowledge affect the performance (i.e., classification accuracy and information content) and the complexity of the induced diagnostic rules. In addition, a medical evaluation of the rules shows that the use of background knowledge in LINUS improves the quality of rules. The performance of CN2 is also compared to the performance of trees induced with LINUS using ASSISTANT (Lavrač et al., 1991b). Like the algorithms of the TDIDT family, the CN2 algorithm has the ability to cope with noisy data but it induces descriptions in the form of if-then rules. It turned out that CN2 is better suited for the use of the particular medical background knowledge in the induction of diagnostic rules.

In the next section, the LINUS inductive learning system is described. Following sections present the diagnostic problem, give the background knowledge to be used in the induced diagnostic rules, and give the results of the experiments and the medical evaluation of induced rules. The paper concludes with a discussion and directions for further work.

THE INDUCTIVE LEARNING SYSTEM LINUS

Inductive learning technology can be used to construct expert knowledge bases more effectively than traditional dialogue-based techniques for knowledge acquisition. Recent developments in inductive learning are concerned with systems that induce concept descriptions in restricted first-order logic (Muggleton, 1991; Quinlan, 1990). The system LINUS induces concept descriptions in the deductive hierarchical database (DHDB) formalism, a form of logic programs restricted to typed nonrecursive program clauses (Lloyd, 1987). As such, LINUS is one of the inductive logic programming systems (Muggleton, 1992).

The Knowledge Representation Formalism

In the application described in this paper, LINUS was used as an attribute-value learner enhanced with the use of background knowledge to learn descriptions of individual diagnostic classes. We will describe the knowledge representation formalism of LINUS for this case only.

Training examples, obtained from a database of patients' records, are represented as ground facts. For example, the training example

patient(male, thorax, inflammatory_spine_disease).

represents a record of a male patient who has pain in the thorax and whose diagnosis belongs to the class of inflammatory spine diseases. When learning the description of the class inflammatory spine disease, this fact is labeled ⊕ and is treated as a positive example. Facts for other diagnostic classes are the *negative examples* for learning the description of this class.

A hypothesis, that is, an induced description of a diagnosis, consists of a set of DHDB rules of the form

$$Class = Diagnosis \quad \textbf{if} \quad L_1, \ldots, L_m.$$

where *Diagnosis* is a diagnostic class and L_i are *conditions*. Borrowing the logic programming terminology (Lloyd, 1987; Ullman, 1988), the conjunction of conditions L_i is called the *body* of a rule. Similar to attribute-value if-then rules, conditions in DHDB rules can have the form $X = a$, where X is an attribute name and a is a constant of the appropriate type. In the case of real-valued attributes, conditions can have the form $X < a$ and/or $X > a$, where a is a real-valued constant. These are both illustrated in the following if-then rule.

Class = extraarticular_rheumatism **if**
  Duration_of_rheumatic_diseases < 1.5,
  Number_of_painful_joints < 0.5,
  Other_pain = other,
  Other_manifestations = no.

Unlike attribute-value if-then rules, conditions that result from the applications of background knowledge can appear in the body of DHDB rules. In our experiments, background knowledge of a functional nature was used—for example, functions of the form $Z = f(X,Y)$ where variables X and Y are attributes describing the training examples and Z is a "new" variable whose value is computed from the values of X and Y in the training examples. For notational convenience, we will write $f(X,Y,Z)$. In this case, in addition to the conditions $Z=a$, $Z>a$, and $Z<a$, the condition $f(X,Y,Z)$ must be added to the body of the rule. This is illustrated by the following rule:

Class = inflammatory_spine_diseases **if**
Sex = male,
Duration_of_rheumatic_diseases > 9,
Other_manifestations = no,
grouping4(Joint_pain, Spinal_pain, Value),
Value = 'no_pain & spondylitic'.

The value of the third argument of function *grouping4*, that is value *'no_pain & spondylitic'*, denotes a characteristic combinations of values of attributes *Joint_pain* and *Spinal_pain;* it represents a meaningful co-occurrence of the symptoms *no_pain* in joints and *spondylitic* pain in the spine.

Learning in LINUS

The main idea in LINUS is to incorporate different attribute-value learning algorithms into the logic programming environment. LINUS incorporates three attribute-value learners: ASSISTANT, a member of the TDIDT family, and two members of the AQ family, NEWGEM (Mozetič, 1985) and CN2. The incorporation into the DHDB environment is provided by a special interface consisting of over 2000 lines of PROLOG code. This DHDB interface transforms the positive examples (given ground facts) and negative examples (possibly generated by the DHDB interface) from the DHDB form into attribute-value tuples. The most important feature of this interface is that, by taking into account the types of arguments of the examples, applications of background knowledge predicates and functions are considered as possible new attributes for learning by an attribute-value learner. Existing attribute-value learners can then be used to induce decision trees or rules, which are in turn transformed into the DHDB rule form by the DHDB interface.

Compared to attribute-value learning, the LINUS approach has a number of advantages. It allows relational descriptions, use of compound terms, compact description of concepts, use of background knowledge in concept descriptions, and inclusion of existing successful learning programs into the logic programming environment. In LINUS, attribute-value learners are used that embody years of research work, that are known to perform well, and that were tested and evaluated on a number of real-life domains. To their advantageous features (e.g., mechanisms for handling noisy data in ASSISTANT and CN2) the ability to learn logical definitions of relations in a more expressive representational formalism is added. Thus, LINUS is also a member of the family of inductive logic programming systems.

The CN2 Algorithm

The domain of early diagnosis of rheumatic diseases is characterized by noisy and missing data. It was therefore appropriate to use CN2 as an attribute-value learner incorporated into LINUS.

The rule induction system CN2 combines the ability to cope with noisy data of the algorithms of the TDIDT family with the if-then rule form and the flexible search strategy of the AQ family. It has retained the covering approach and the beam search mechanism from AQ but has removed its dependence on specific examples during the search. Furthermore, in order to deal with noisy data, it has extended the search space to rules that do not perform perfectly on the training data. Initially, CN2 used an entropy-based function as a search heuristic to induce ordered rules (Clark and Niblett, 1989). It has recently been extended with the ability to induce unordered rules and to use Bayesian accuracy estimates as search heuristics (Clark and Boswell, 1991; Džeroski et al., 1992).

When learning if-then rules, objects of a given class are labeled ⊕ and treated as positive examples $\mathcal{E}^+$, and all other objects are treated as negative examples $\mathcal{E}^-$ of the selected class. Given the current training set of examples (initially set to the entire training set) and the current hypothesis (initially set to the empty set of rules), the outer loop of the CN2 algorithm for inducing unordered rules (Clark and Boswell, 1991) implements the "covering" algorithm. It constructs a hypothesis in three main steps:

- Construct a rule.
- Add the rule to the current hypothesis.
- Remove from the current training set the positive examples covered by the rule.

This loom is repeated until all the positive examples are covered.

Let $\mathcal{H}$ denote the current hypothesis and $\mathcal{E}_c$ denote the current set of training examples. Algorithm 1 is an outline of the CN2 covering algorithm, which is essentially the same as the AQ covering algorithm. The best body is chosen according to the search heuristic, which measures the expected classification accuracy of the rule, estimated by the Laplace probability estimate (Clark and Boswell, 1991).

Let k be the total number of classes in the problem. Let n_c^+ be the number of covered positive examples of the class *ClassValue* in the current training set $\mathcal{E}_c$ and n_c be the total number of covered positive and negative examples. The Laplace

Algorithm 1 (CN2—the covering algorithm)

```
Given: Training examples $\mathcal{E}^+$ and $\mathcal{E}^-$ for the selected ClassValue.
Initialize the hypothesis $\mathcal{H} := \emptyset$.
Initialize the training set $\mathcal{E}_c := \mathcal{E}^+ \cup \mathcal{E}^-$.
repeat
  Call the BeamSearchAlgorithm($\mathcal{E}_c$, ClassValue) to find the BestBody of a rule.
  if BestBody $\neq$ not_found
  then $\mathcal{H} := \mathcal{H} \cup \{Class = ClassValue$ if $BestBody\}$,
       i.e., add the best rule to $\mathcal{H}$.
       Remove from $\mathcal{E}_c$ the positive examples covered by BestBody.
until BestBody = not_found.
Output: Hypothesis $\mathcal{H}$.
```

classification accuracy estimate *A*(*Rule*) estimates the probability that an example covered by a *Rule* is positive:

$$A(Rule) = p(\oplus \mid Rule) = \frac{n_c^+ + 1}{n_c + k}$$

Algorithm 2 (CN2—the beam search algorithm)

```
Given: Training examples $\mathcal{E}_c$ for the selected ClassValue.
Initialize Beam: = {true}.
Initialize NewBeam: = $\emptyset$.
Initialize BestBody: = not_found.
while Beam $\neq \emptyset$ do
  for each Body in Beam do
      for each possible specialization Spec of Body do
          if Spec is better than BestBody and
             Spec is statistically significant
          then BestBody: = Spec.
               Add Spec to NewBeam.
          if Size of NewBeam > BeamSize
          then remove worst body from NewBeam.
      endfor
  endfor
  Beam: = NewBeam.
endwhile
Output: BestBody.
```

In its previous version, instead of the Laplace estimate, the CN2 algorithm was using an entropy-based search heuristic (Clark and Niblett, 1989).

The beam search for the best body of a rule in Algorithm 2 proceeds in a top-down fashion. The search starts with the most general body (*true*), which covers all the training examples. At each step, a set of candidates (*Beam*) for the best body, as well as the best body found so far, are kept. To specialize a body, a condition of the form $X = a$ is added to it as a conjunct. All the possible specializations of the candidates from *Beam* are considered. The best body found so far is accordingly updated and the *BeamSize* most promising candidates among the newly generated specializations are chosen.

The best body found so far is, in addition, tested for its statistical significance. This is to ensure that it represents a genuine regularity in the training examples and not a regularity due to chance. In this way, the undesirable bias of the entropy and apparent accuracy metrics used to estimate the quality of a clause is, at least partly, avoided. For a detailed treatment of the heuristics in CN2 and their role in handling noisy data, we refer the reader to Clark and Boswell (1991) and Džeroski et al. (1992).

The LINUS Learning Algorithm

The outermost level of the LINUS learning algorithm consists of the following steps:

- Establish the training sets of positive and negative facts.
- Using background knowledge, transform facts in the DHDB form into attribute-value tuples.
- Induce a concept description by an attribute-value learner.
- Transform the induced if-then rules into the form of DHDB rules.

In the *first step*, the sets of positive and negative facts are established. The generation of negative facts takes into account the types of arguments and the closed-world assumption. There are three different options for negative examples generation. However, since in our application separate rules are learned for the individual diagnostic classes, negative facts are given explicitly. Negative facts are examples of all the patients not belonging to the given diagnostic class.

In the *second step* of the algorithm, the positive and negative facts are transformed into an attribute-value form. The algorithm first checks which are the possible applications of the background knowledge predicates and functions (called utility predicates and functions) and then generates attribute-value tuples by assigning values to the enlarged set of attributes. Value *true* or *false* is assigned to each application of a utility predicate on the argument values of the relation to be learned. Similar computation is performed for functions, except that values of the output argument of a function are computed, instead of only assigning values *true* and false.

Let us illustrate the application of a utility function with an example. Suppose a patient is described by the relation *patient* having as arguments values of the attributes sex and location of pain and the patient's diagnosis. Suppose that a set of training examples includes the following fact:

patient(male, thorax, inflammatory_spine_disease).

As will be shown later, a male patient having pain in the thorax has one of the characteristic co-occurrences of symptoms. Such a characteristic grouping of symptoms can be defined as specialist background knowledge in the form of a utility function, which can be considered in inducing descriptions of the individual diagnostic classes. This background knowledge can be defined as follows:

```
% grouping3( sex/input, location/input, grouping3_value/output)
grouping3( male, thorax, male_thorax ) :- !.
grouping3( male, heels, male_heels ) :- !.
grouping3( _, _, irrelevant ).
```

This utility function states two typical combinations of values of two attributes, sex and location of pain. All the other combinations of values are irrelevant. Note that, for practical reasons, the definition is encoded in the PROLOG syntax, using the cut (!) operator.

In this step of the algorithm, tuples of attribute values are generated. An additional attribute appears in the tuples, which stands for the value of the utility function. The form of tuples (the set of attributes) is the following:

$$\langle sex, location, grouping3\_value, diagnosis \rangle$$

For the given positive example *patient(male, thorax, inflammatory_spine_disease)*, the following tuple is generated:

$$\langle male, thorax, male\_thorax, inflammatory\_spine\_disease \rangle$$

In this way, the set of attributes to be considered for learning is enlarged, including one additional attribute for the characteristic combination of symptoms, and the appropriate value is assigned to the new attribute. A similar computation is performed for all positive and negative examples.

The *third step* of the algorithm is the induction of a concept description that depends on the choice of the learning algorithm. Training examples in the form of tuples are transformed into the appropriate input form for learning by the ASSISTANT, NEWGEM, or CN2 algorithm; learning by the selected attribute-value learner is invoked; and the obtained concept description (in the form of a decision tree or if-then rules) is transcribed into the form of if-then rules. In our application the CN2 algorithm was used.

In the *fourth step,* the induced if-then rules are transformed into the form of DHDB rules. Before this transformation is performed, a special *postprocessor* checks whether the if-then rules can be made more compact. This is used in LINUS with ASSISTANT only on the if-then rules derived from decision trees. In noisy domains, postprocessing eliminates irrelevant conditions from the body of a rule. A condition is *irrelevant* if it can be removed from the rule without decreasing its classification accuracy on the training set.

DIAGNOSTIC PROBLEM AND EXPERIMENTAL DATA

Data about 462 patients were collected at the University Medical Center in Ljubljana. There are over 200 different rheumatic diseases, which can be grouped into 3, 6, 8, or 12 diagnostic classes. As suggested by a specialist, eight diagnostic classes were considered. Table 1 shows the names of the diagnostic classes and the numbers of patients belonging to each class.

The experiments were performed on anamnestic data, without taking into account data about patients' clinical manifestations and laboratory and radiological findings. The 16 anamnestic attributes are as follows: sex, age, family anamnesis, duration of present symptoms, duration of rheumatic diseases, joint pain (arthrotic, arthritic), number of painful joints, number of swollen joints, spinal pain (spondylotic, spondylitic), other pain (headache, pain in muscles, thorax, abdomen, heels), duration of morning stiffness, skin manifestations, mucosal manifestations, eye manifestations, other manifestations, and therapy.

Of 462 patients' records only 8 were incomplete; 12 attribute values were missing (for attributes sex and age). This was not problematic because LINUS using CN2 can handle missing data.

TABLE 1. The Eight Diagnostic Classes and the Corresponding Numbers of Patients

Class	Name	Number of patients
A1	Degenerative spine diseases	158
A2	Degenerative joint diseases	128
B1	Inflammatory spine diseases	16
B234	Other inflammatory diseases	29
C	Extraarticular rheumatism	21
D	Crystal-induced synovitis	24
E	Nonspecific rheumatic manifestations	32
F	Nonrheumatic diseases	54

MEDICAL BACKGROUND KNOWLEDGE

The available patient data were augmented with additional diagnostic knowledge. A specialist for rheumatic diseases has provided his knowledge about the typical co-occurrences of symptoms. Six typical groupings of symptoms were suggested by the specialist.

1. The first grouping relates the joint pain and the duration of morning stiffness. The characteristic combinations are given in the following table; all other combinations are insignificant or irrelevant:

Joint pain	Morning stiffness
No pain	≤ 1 hour
Arthrotic	≤ 1 hour
Arthritic	> 1 hour

2. The second grouping relates the spinal pain and the duration of morning stiffness. The following are the characteristic combinations:

Spinal pain	Morning stiffness
No pain	≤ 1 hour
Spondylotic	≤ 1 hour
Spondylitic	> 1 hour

3. The third grouping relates sex and other pain. Indicative is the pain in the thorax or in the heels for male patients; all other combinations are nonspecific:

Sex	Other pain
Male	Thorax
Male	Heels

4. The fourth grouping relates joint pain and spinal pain. The following are the characteristic co-occurrences:

Joint pain	Spinal pain
No pain	Spondylotic
Arthrotic	No pain
No pain	Spondylitic
Arthritic	Spondylitic
Arthritic	No pain
No pain	No pain

5. The fifth grouping relates joint pain, spinal pain, and the number of painful joints:

Joint pain	Spinal pain	Painful joints
No pain	Spondylotic	0
Arthrotic	No pain	1 ≤ joints ≤ 30
No pain	Spondylitic	0
Arthritic	Spondylitic	1 ≤ joints ≤ 5
Arthritic	No pain	1 ≤ joints ≤ 30
No pain	No pain	0

6. The sixth grouping relates the number of swollen joints and the number of painful joints:

Swollen joints	Painful joints
0	0
0	1 ≤ joints ≤ 30
1 ≤ joints ≤ 10	0 ≤ joints ≤ 30

This background knowledge is encoded in the form of utility functions, introducing specific function values for each characteristic combination of symptoms. All the other combinations (except the ones explicitly specified in the preceding tables) have the same function value *irrelevant*. The characteristic combinations of attribute values are given names that are mnemonic and understandable. Their names are "artificial" (not used by specialists), but they represent meaningful co-occurrences of symptoms that have their role in expert diagnosis. An example utility function implementing the third grouping of symptoms was given earlier.

The choice of functions instead of utility predicates is based on the same argument. It is important for a specialist to know the exact value of the function (i.e., the exact combination of symptoms) and not only to know that some combination of values of individual attributes has occurred (which would be the case if this knowledge were encoded in the form of utility predicates having only values *true* and *false*).

EXPERIMENTS AND RESULTS

In order to evaluate the effects of background knowledge and noise-handling mechanisms on the classification accuracy and the complexity of the induced rules, two groups of experiments were performed: one group on the whole set of patient data and the other on 10 different partitions of the data set into training and testing examples. In each group, the experiments were further designed along another two dimensions. In one half of the experiments the background knowledge described in the preceding section was used in the learning process. Each of the groupings contributed an additional attribute, giving a total of six new attributes. The set of attributes used for learning with CN2 thus consisted of 16 initial and 6 new attributes. In the other half of the experiments no background knowledge was used, and the set of attributes consisted of the 16 initial attributes only. Along the other dimension, the significance test noise-handling mechanism in CN2 (with a significance level of 99%) was used in one half of the experiments and not in the other half.

Learning from the Entire Training Set

In the first group of experiments, the data about all 462 patients were used. Four experiments were performed in which the use or nonuse of background knowledge and the use or nonuse of the significance test were varied. The classification accuracy and the relative information score (both calculated on the training set), as well as the complexity of the induced rule sets, were measured. Table 2 gives the results of these experiments.

The classification accuracy was measured as the percentage of examples correctly classified by the rule set. The complexity was measured by the number of rules in the set, as well as the total number of conditions appearing in all rules in the set. Finally, the information content, that is, relative information score (Kononenko and Bratko, 1991), was measured, which takes into account the difficulty of the classification problem. To this end, a suitably modified implementation of CN2 (Džeroski et al., 1992) was used.

To scale the evaluation of a classifier to the difficulty of the problem, the relative information score takes into account the prior probability of diagnoses (diseases). That is, a correct classification into a more probable diagnosis provides less information than a correct classification into a rare diagnosis, which is represented by only a few training examples (Kononenko and Bratko, 1991). For example, in domains where one of the diagnostic classes is highly likely, it is easy to achieve high classification accuracy. The completely uninformed classifier that assigns the most common diagnosis to all patients would in that case have undeservedly high classification accuracy.

The results in Table 2 show that the use of background knowledge increased the accuracy of the induced rules on the training set. More important, it also increased

TABLE 2. Classification Accuracy, Relative Information Score (Both Measured on the Training Set Itself), and Complexity of Rules Derived from all 462 Examples

Background knowledge	Significance test	Accuracy (%)	Relative inf. score (%)	Number of rules	Number of conditions
No	No	62.8	49	96	302
No	Yes	51.7	22	30	102
Yes	No	72.9	59	96	301
Yes	Yes	52.4	30	38	120

the information content of the rules. The total number of conditions appearing in all rules increased with the use of background knowledge when the significance test was used and remained unchanged when the significance test was not used.

The use of the significance test greatly decreased the size of the rule sets, both with and without using background knowledge. However, it also caused a decrease in the classification accuracy on the training set. This is natural, since the main function of this noise-handling mechanism is to prevent rules from overfitting the training set. Although this decreases the classification accuracy on the training set, it usually increases the classification accuracy on unseen cases (see Table 6).

All groupings appear in the induced rules. In the rules induced with no significance test, the most common groupings are *grouping5* with 13 and *grouping4* with 12 occurrences; *grouping2* occurs five times, *grouping1* and *grouping6* four times each, and *grouping3* twice. When the significance test is used, the number of occurrences decreases, as the number of rules (conditions) decreases drastically. In this case, *grouping1* occurs twice, *grouping2* three times, *grouping3* and *grouping6* once each, *grouping5* seven times, and *grouping4* nine times. The most common groupings (4 and 5) combine spinal pain, the most informative attribute, with other relevant features.

To summarize, all utility functions from the background knowledge appeared in the induced rules. The use of background knowledge improved both the classification accuracy and the relative information score of the induced rules at the cost of a slight increase of rule complexity. In the following section, a medical evaluation of the effects of background knowledge on the induced rules is presented.

Medical Evaluation of Diagnostic Rules

The induced diagnostic rules were shown to a specialist for rheumatic diseases, who found most of the rules meaningful and understandable. The specialist evaluated the entire set of induced rules according to the following procedure. For each of the conditions in a rule, one point was given to the rule if the condition was in favor of the diagnosis made by the rule, minus one point if the condition was against the diagnosis and zero points if the condition was irrelevant to the diagnosis. The mark of a rule was established as the sum of the points for all conditions in the rule.

The actual marks ranged from minus one to three. Intuitively, mark 3 was given to rules that are very characteristic for a disease and could even be published in a medical book. Mark 2 was given to good, correct rules. Mark 1 was given to rules that are not wrong but are not too characteristic for the diagnosis. Mark 0 was given to rules that are possible according to the specialist's knowledge; however, they reflect a coincidential combination of features rather than a characteristic one. Mark –1 was given to misleading rules, which actually indicate that the diagnosis is not likely.

Two sets of rules were evaluated, the ones induced with and without the use of background knowledge, both induced using the significance test. Table 3 shows some rules that were evaluated as best by the specialist and their marks. They were

TABLE 3. Rules for Early Diagnosis of Rheumatic Diseases Induced with the Use of Background Knowledge

Class = degenerative_spine_disesease **if** *Duration_of_present_symptoms > 6.5_months,* *Duration_of_rheumatic_diseases < 5.5_years,* *Number_of_painful_joints > 16,* *grouping2(Spinal_pain, Duration_of_morning_stiffness, Value),* *Value = 'spondylotic & dms =< 1_hour'.*	Mark:2
Class = degenerative_spine_diseases **if** *Age < 66.5,* *Other_pain = no,* *Skin_manifestations = no,* *grouping5(Joint_pain, Spinal_pain, Number_of_painful_joints, Value),* *Value = 'no_pain & spondylotic & pj = 0'.*	Mark: 2
Class = degenerative_joint_diseases **if** *Age > 46.5,* *Duration_of_present_symptoms < 30_months,* *Number_of_painful joints < 19,* *Duration_of_morning_stiffness < 0.75_hours,* *grouping3(Sex, Other_pain, Value1),* *Value1 = irrelevant,* *grouping4(Joint_pain, Spinal_pain, Value2),* *Value2 = 'arthrotic & no_pain'.*	Mark: 3
Class = inflammatory_spine_diseases **if** *Sex = male,* *Duration_of_rheumatic_diseases > 9_years,* *Other_manifestations = no,* *grouping4(Joint_pain, Spinal_pain, Value),* *Value = 'no_pain & spondylitic'.*	Mark: 2
Class = other_inflammatory_diseases **if** *Age < 78.5,* *Number_of_painful_joints > 18,* *Number_of_swollen_joints > 2.5,* *grouping1(Joint_pain, Duration_of_morning_stiffness, Value),* *Value = 'arthritic & dms > 1_hour'.*	Mark: 3
Class = extraarticular_rheumatism **if** *Duration_of_rheumatic_diseases < 1.5_year,* *Number_of_painful_joints < 0.5,* *Othre_pain = other,* *Other_manifestations = no.*	Mark: 0
Class = crystal_induced_synovitis **if** *Sex = male,* *Age > 24,* *Number_of_painful_joints < 14.5,* *Other_pain = no,* *Duration_of_morning_stiffness < 0.25_hours,* *Eye_manifestations = no,* *grouping4(Joint_pain, Spinal_pain, Value),* *Value = 'arthritic & no_pain'.*	Mark: 2

TABLE 3. *Continued*

Class = crystal_induced_synovitis **if** *Sex = male,* *Age > 46.5,* *Number_of_painful_joints > 3.5,* *Skin_manifestations = psoriasis.*	Mark: 1
Class = nonspecific_rheumatic_diseases **if** *Age > 29.5, Age < 33.5,* *Number_of_painful_joints > 10.5,* *Spinal_pain = no_pain.*	Mark: 0
Class = nonrheumatic_diseases **if** *Age < 53.5,* *Duration_of_present_symptoms > 2_months,* *Number_of_swollen_joints < 0.5,* *Other_pain = no,* *Eye_manifestations = no,* *grouping5(Joint_pain, Spinal_pain, Number_of_painful_joints, Value),* *Value = 'no_pain & no_pain & pj = 0'.*	Mark: 3

induced by LINUS using CN2 with the use of background knowledge and the significance test. All rules for the diagnosis *E* (nonspecific rheumatic manifestations) were given mark 0. This is due to the fact that the only characteristic of this class is that all patients with rheumatic diseases who cannot be diagnosed otherwise are assigned this diagnosis.

Tables 4 and 5 summarize the medical expert evaluation of the induced rules. Out of 30 rules induced without background knowledge, no rules were given mark 3, two rules were given mark 2, fifteen mark 1, ten mark 0, and three mark –1 (misleading rules). The average rule mark is 0.53 and the three misleading rules cover 34 examples. On the other hand, out of 38 rules induced with background knowledge, three rules were given mark 3, nine mark 2, fourteen mark 1, eleven mark 0, and only one rule was misleading. The average rule mark in this case is 1.05 and the misleading rule covers only five examples. All these figures indicate that

TABLE 4. Medical Expert Evaluation of Rules Without Background Knowledge

	Number of rules with mark					Total no. of	
Class	3	2	1	0	–1	rules	Average mark
A1			4	1	2	7	0.29
A2			3	2	1	6	0.33
B1		1	2			3	1.33
B2			3	1		4	0.75
C			1	2		3	0.33
D		1	2			3	1.33
E				3		3	0.00
F				1		1	0.00
Overall	0	2	15	10	3	30	0.53

TABLE 5. Medical Expert Evaluation of Rules with Background Knowledge

	Number of rules with mark					Total no. of	
Class	3	2	1	0	−1	rules	Average mark
A1		3	2	2		7	1.14
A2	1	1	3	1	1	7	1.00
B1		1	2			3	1.33
B2	1	2	4			7	1.57
C				3		3	0.00
D		1	2			3	1.33
E				4		4	0.00
F	1	1	1	1		4	1.50
Overall	3	9	14	11	1	38	1.05

the background knowledge substantially improves the induced rules from the medical expert point of view.

Performance on Unseen Examples

Since the ultimate test of the quality of induced rules is their performance on unseen examples, the second group of experiments was performed on 10 different partitions of the data set into training and testing examples. Thus, corresponding to each of the experiments on the entire training set, four series of 10 experiments each were performed, where 70% of the entire data set was used for learning and 30% for testing. A different partition into training and testing examples was used in each of the experiments and the same partitions were used for all four series.

The experiments were aimed at investigating the effects of background knowledge and noise-handling mechanisms on the performance of induced rules on unseen cases. The performance of the rules was measured in terms of the classification accuracy and the relative information score. A summary of the results of the experiments is given in Table 6. The classification accuracy and the relative information score for all experiments are given in Table 7 and Table 8, respectively.

As expected, the average classification accuracy is much lower than the classification accuracy on the training data in Table 2. The relative information score

TABLE 6. Average Classification Accuracy, Relative Information Score (Measured on Testing Data), and Complexity of Rules Induced by LINUS Using CN2

Background knowledge	Significance test	Accuracy (%)	Relative inf. score (%)	Number of rules	Number of conditions
No	No	42.9	23	72	222
No	Yes	45.3	19	30	76
Yes	No	43.9	24	74	226
Yes	Yes	48.6	26	24	88

TABLE 7. Classification Accuracy of Rules, Derived by LINUS Using CN2 with and without Background Knowledge (BK), on the Testing Set for Each of the Ten Partitions of the 462 Examples

	No significance test		Significance level 99%	
Partition	Without BK (%)	With BK (%)	Without BK (%)	With BK (%)
1	38.1	45.3	47.5	48.9
2	44.6	44.6	45.3	51.8
3	45.3	42.4	51.1	48.9
4	43.9	40.3	44.6	48.2
5	40.3	43.2	46.0	46.8
6	48.2	48.2	49.6	48.2
7	42.4	44.6	44.6	48.9
8	38.8	43.2	41.0	48.2
9	45.3	41.0	43.9	48.2
10	41.7	46.0	39.6	48.2
Average	42.9	43.9	45.3	48.6

also drops substantially. Finally, the rule set size is also reduced, as the rules have to explain a smaller number of examples (70% of the entire data set).

The use of background knowledge improved the classification accuracy and the relative information scores achieved. According to the statistical *t*-test for dependent samples, the difference in accuracy is not statistically significant when the CN2 significance test is not used. However, it is very significant (at the *t*-test 99% level) when the significance test is used in CN2. The relative information score difference is statistically significant in both cases (at the *t*-test 95% and 99.99% levels, respectively, with and without the significance test in CN2).

The significance test noise-handling mechanism in CN2 greatly reduces the complexity of the induced rules. It also improves the classification accuracy sig-

TABLE 8. Relative Information Scores of Rules, Derived by LINUS Using CN2 with and without Background Knowledge (BK), on the Testing Set for Each of the Ten Partitions of the 462 Examples

	No significance test		Significance level 99%	
Partition	Without BK (%)	With BK (%)	Without BK (%)	With BK (%)
1	21	23	17	26
2	23	24	20	28
3	19	22	17	24
4	24	21	17	22
5	22	25	21	26
6	26	24	15	24
7	27	30	21	27
8	19	24	21	26
9	23	23	16	25
10	23	27	23	31
Average	23	24	19	26

nificantly. When background knowledge is used, it significantly improves both the classification accuracy (t-test significance level 99.99%) and the relative information score (t-test significance level 95%).

All groupings appear in the rule sets induced with use of background knowledge. In the rules induced with no CN2 significance test, the most common groupings are *grouping4* with 96 and *grouping5* with 80 occurrences in the 10 rule sets; *grouping6* occurs 38 times, *grouping2* occurs 31 times, *grouping1* occurs 23 times, and *grouping3* occurs 12 times. When the CN2 significance test is used, the number of occurrences decreases, as the number of rules (conditions) decreases drastically. In this case, *grouping1* occurs 10 times, *grouping2* occurs 7 times, *grouping3* occurs 5 times, *grouping4* occurs 73 times, *grouping5* occurs 71 times, and *grouping6* occurs 24 times.

Experiments were also conducted with LINUS using ASSISTANT on the same partitions of the data set as used by LINUS using CN2. ASSISTANT induces decision trees, where noise is handled by either tree prepruning or postpruning (Cestnik et al., 1987). A summary of the results of these experiments is given in Table 9.

The background knowledge did not significantly influence the performance of the trees induced by LINUS using ASSISTANT. This is probably due to the fact that the background knowledge is in a form more suitable for rule induction; that is, the groupings typically distinguish between one diagnosis and all the others. In decision trees, on the other hand, features that distinguish best among all possible diagnoses are favored.

The noise-handling mechanisms of prepruning and postpruning in ASSISTANT increased the accuracy of the induced trees. The increases are statistically significant (at the 98% level according to the t-test for dependent samples) in the case of prepruning and postpruning, when no background knowledge is used, and in the case of postpruning when background knowledge is used. It is interesting to note that prepruning increased both the classification accuracy and the relative information score, while postpruning increased the accuracy at the expense of decreasing the relative information score.

TABLE 9. Average Classification Accuracy and Relative Information Score (Measured on Testing Data) of Trees Generated by LINUS Using ASSISTANT

Background knowledge	Pruning	Accuracy (%)	Relative information score (%)
No	No	41.80	28.68
No	Pre	44.42	31.03
No	Post	48.92	25.31
Yes	No	42.34	28.90
Yes	Pre	43.17	30.32
Yes	Post	48.99	25.94

Finally, let us mention that the accuracies and relative information scores achieved by LINUS using CN2 with background knowledge and with LINUS using ASSISTANT (both with and without background knowledge) are almost the same. However, the rules induced by CN2 proved to be easier to understand by the medical expert than the decision trees induced by ASSISTANT. In that context, the background knowledge substantially improved the quality of the rules induced by CN2 as evaluated by the medical expert (see Tables 4 and 5).

DISCUSSION

LINUS, an inductive learning system, was used to learn diagnostic rules from anamnestic data of patients with rheumatic diseases. In addition to the available patient data, described by values of a fixed set of attributes, LINUS was given domain-specific (background) knowledge, which specified some of the characteristic co-occurrences of symptoms.

Medical evaluation of rules induced by LINUS using CN2 shows that the use of background knowledge substantially improves the quality of the induced rules from a medical point of view. However, even when background knowledge is used, the average rule is *not wrong, but not too characteristic* (average mark 1.05). The analysis by a specialist for rheumatic diseases indicated several reasons:

- Anamnestic data are by nature very noisy because they are, in fact, patients' own descriptions of the disease, only interpreted by a specialist for rheumatic diseases. Interpretation of these data is subjective and therefore extremely unreliable. In many cases the data even contradicts the expert's background knowledge.
- The grouping of about 200 different diagnoses into only eight diagnostic classes is problematic. For degenerative diseases (classes A1 and A2) many examples are available. Nearly 74% of all the data set consists of patients' records for these two diagnostic classes, together with the class of nonrheumatic diseases (see Table 1). Furthermore, some diagnostic classes are relatively nonhomogenous, having few common characteristics. Consequently, the specific background knowledge, which is usually used in differential diagnosis, cannot be used effectively when dealing with such large groups of diagnoses.
- Some of the patients had more than one diagnosis, but only one diagnosis was included in the example set.
- Data were collected by different medical doctors without achieving their collective consensus. This could be why the data sometimes contradict the expert's background knowledge.

This analysis shows the problems present in the data set. Nevertheless, useful information can still be extracted from such data. The relative information score of classifications of training examples (without using the significance test) was 49%

(accuracy 62.8%) when no background knowledge was used and 59% (accuracy 72.9%) when background knowledge was used (Table 2). The relative information score of a classifier that always returns the prior probability distribution of diagnostic classes is zero (accuracy 34%). The information score of the induced rules thus indicates that useful information is contained in the data set.

The main goal of the study was to analyze how the use of background knowledge and the noise-handling mechanism of LINUS using CN2 affect the performance and the complexity of induced diagnostic rules. The use of noise-handling mechanisms improved the classification accuracy of induced rules. We have also shown that the use of additional background knowledge can improve the classification accuracy and the relative information score of induced rules. We expect that still better results could be achieved by restricting the problem to differential diagnosis in a smaller subset of diagnoses where specific (background) knowledge could have a more substantial role. Furthermore, the selection of typical patients only is also expected to contribute to better results.

ACKNOWLEDGMENTS

This research was financially supported by the Slovenian Ministry of Science and Technology. It is a part of the ESPRIT III Basic Research Project No. 6020 Inductive Logic Programming. The authors are grateful to the specialists of the University Medical Center in Ljubljana who helped collect the data. Aram Karalič prepared the data in a form appropriate for the experiments, and Dunja Mladenič provided the environment for transforming the data into the form accepted by LINUS. Thanks to Ivan Bratko and Bojan Cestnik for their comments on the experiments with LINUS using ASSISTANT published in the Proceedings of the 1991 Scandinavian AI Conference, Roskilde (IOS Press, Amsterdam) and to Igor Kononenko for his comments on this paper.

REFERENCES

Cestnik, B., Kononenko, I., and Bratko, I. 1987. ASSISTANT 86: A knowledge elicitation tool for sophisticated users. In *Progress in Machine Learning*, eds. I. Bratko and N. Lavrač, pp. 31–45. Wilmslow: Sigma Press.

Clark, P., and Boswell, R. 1991. Rule induction with CN2: Some recent improvements. *Proc. Fifth European Working Session on Learning*, pp. 151–163. Berlin: Springer.

Clark, P., and Niblett, T. 1989. The CN2 induction algorithm. *Machine Learning* 3(4):261–283.

Džeroski, S., Cestnik, B., and Petrovski, I. 1992. The use of Bayesian probability estimates in rule induction. *Proc. First Slovenian Electrical Engineering and Computer Science Conference*, pp. 155–158, September 28–30, Portorož, Slovenia.

Karalič, A., and Pirnat, V. 1990. Machine learning in rheumatology. *Sistemica* 1(2):113–123.

Kononenko, I., and Bratko, I. 1991. Information-based evaluation criterion for classifier's performance. *Machine Learning* 6(1):67–80.

Lavrač, N., and Džeroski, S., 1992. Inductive learning of relations from noisy examples. In *Inductive Logic Programming*, ed. S. Muggleton, pp. 495–514. London: Academic Press.

Lavrač, N., and Džeroski, S., 1993. Weakening the language bias in LINUS. *J. Exp. Theor. Artif. Intell.* 5(2). In press.

Lavrač, N., and Džeroski, S., and Grobelnik, M. 1991a. Learning nonrecursive definitions of relations with LINUS. *Proc. Fifth European Working Session on Learning,* pp. 265–281. Berlin: Springer.

Lavrač, N., Džeroski, S., Pirnat, V., and Križman, V. 1991b. Learning rules for early diagnosis of rheumatic diseases. *Proc. Third Scandinavian Conference on Artificial Intelligence,* pp. 138–149. Amsterdam: IOS Press.

Lloyd, J. 1987. *Foundations of Logic Programming.* 2d ed. Berlin: Springer.

Michalski, R., Mozetič I., Hong, J., and Lavrač, N. 1986. The multi-purpose incremental learning system AQ15 and its testing application on three medical domains. *Proc. Fifth National Conference on Artificial Intelligence,* pp. 1041–1045. San Mateo, CA: Morgan Kaufmann.

Mozetič, I., 1985. NEWGEM: Program for learning from examples, technical documentation and user's guide. Reports of Intelligent Systems Group UIUCDCS-F-85-949, Department of Computer Science, University of Illinois, Urbana Champaign.

Muggleton, S. 1991. Inductive logic programming. *New Generation Computing* 8(4):295–318.

Muggleton, S., ed. 1992. *Inductive Logic Programming.* London: Academic Press.

Pirnat, V., Kononenko, I., Janc, T., and Bratko, I. 1989. Medical analysis of automatically induced diagnostic rules. *Proc. Second European Conference on Artificial Intelligence in Medicine,* pp. 24–36. Berlin: Springer.

Quinlan, J. 1986. Induction of decision trees. Machine Learning 1(1):81–106.

Quinlan, J. 1990. Learning logical definitions from relations. *Machine Learning* 5(3):239–266.

Ullman, J. 1988. *Principles of Database and Knowledge Base Systems.* 2 vols. Rockville, MD: Computer Science Press.

Revised version received November 16, 1992