Conditions for Occam's Razor Applicability and Noise Elimination

Dragan Gamberger[1] and Nada Lavrač[2]

[1] Rudjer Bošković Institute, Bijenička 54,10000 Zagreb, Croatia
E-mail: gambi@lelhp1.irb.hr
[2] Jožef Stefan Institute, Jamova 39, 1000 Ljubljana, Slovenia
E-mail: nada.lavrac@ijs.si

Abstract. The Occam's razor principle suggests that among all the correct hypotheses, the simplest hypothesis is the one which best captures the structure of the problem domain and has the highest prediction accuracy when classifying new instances. This principle is implicitly used also for dealing with noise, in order to avoid overfitting a noisy training set by rule truncation or by pruning of decision trees. This work gives a theoretical framework for the applicability of Occam's razor, developed into a procedure for eliminating noise from a training set. The results of empirical evaluation show the usefulness of the presented approach to noise elimination.

1 Introduction

The Occam's razor principle, commonly attributed to William of Occam (early 14th century), states: "Entities should not be multiplied beyond necessity." This principle is generally interpreted as: "Among the theories that are consistent with the observed phenomena, one should select the simplest theory" [9].

Occam's razor is the principle explicitly or implicitly used in many inductive learning systems. This principle suggests that among all the hypotheses that are correct for all (or for most of) the training examples one should select the simplest hypothesis; it can be expected that this hypothesis is most likely to capture the structure inherent in the problem and that its high prediction accuracy can be expected on objects outside the training set [14]. This principle is also used by noise handling and predicate invention algorithms because noise handling and predicate invention have the aim to simplify the generated rules or decision trees in order to avoid overfitting a noisy training set.

Despite the successful use of the Occam's razor principle as the basis for hypothesis construction (e.g., implemented in tree pruning and rule truncation mechanisms), several problems arise in practice. First is the problem of the definition of the most appropriate complexity measure that will be used to identify the simplest hypothesis. The problem is that different measures can select different simplest hypotheses for the same training set. This holds for any Kolmogorov complexity based measure [9], including the MDL (Minimal Description Length) [14, 16], that use approximations of an ideal complexity measure. Second, recent experimental work has undoubtly shown that applications of the Occam's razor may not always lead to the optimal prediction accuracy; a systematic way of improving the results generated by the application of the Occam's razor principle has even been suggested [17]. This is somewhat shocking for the inductive learning community because the principle is the basis of most practical and popular inductive learning systems (e.g., [12]). Additional disorientation is caused by the so-called "conservation law of generalization performance" [15].

Although it is rather clear [13] that real-word learning tasks are different from the set of all theoretically possible learning tasks as defined in [15], there remains the so-called "selective superiority problem" that each algorithm performs best in some but not all domains [1], which can be in a way explained by the bias-variance tradeoff [6]. Our work contributes to the understanding of the abovementioned phenomena by studying for which domains (real or theoretically possible) an inductive learning algorithm based on the Occam's razor principle has a theoretical chance for successful induction.

In this paper, we elaborate the conditions for Occam's razor applicability: for noiseless two class domains we define conditions that guarantee that a hypothesis, that is correct for all the training examples and is the simplest with respect to a selected complexity measure of predefined properties, is correct also for examples outside the training set. So the theorems theoretically solve the problems opened by the "conservation law" by showing that under given conditions effective hypothesis induction is possible. The theorems also theoretically solve the problems concerning the applicability of the Occam's razor principle by giving explicit conditions when the principle can help in induction. In reality, the presented theorems are much less useful because their conditions are so strict that they are rarely fulfilled in practice. For practical applications, the theorems are more interesting for indicating the properties that need to be satisfied for the effective induction using the Occam's razor principle. Moreover, they are the basis for the proposed algorithm for noise elimination.

The paper presents a theoretical framework for the applicability of the Occam's razor principle in Sections 2–7. Section 8 proposes an algorithm for eliminating noise from a training set, and Section 9 provides experimental evidence for the usefulness of the proposed approach to noise elimination.

2 Basic problem definition

We assume a propositional inductive task that starts from a set of training examples E that are part of some example domain D, $E \subset D$. Each of the examples is either positive or negative (a two class problem) and is described by a fixed set of attribute values. Some attribute values can be unknown. The aim of inductive learning systems is to construct a hypothesis (rule, decision tree) with the largest rate of correct class predictions over yet unseen examples in D.

Suppose that domain D is ideal in the sense that it contains no errors (noise) and no contradictions. In such a case, a target theory T that needs to be discovered is correct for all the examples in D (i.e., it is true for all the positive and false for all the negative examples in D). Every inductive learner uses some form of knowledge representation which defines the hypothesis space. In this work, the target hypothesis H_T denotes the simplest hypothesis from the hypothesis space that is correct for all the examples in D. It represents the target theory in the given hypothesis space. A necessary (but not a sufficient) condition for a hypothesis H to be the target hypothesis H_T is that it must be correct for all the examples in E. But as this condition may be true for many hypotheses, the inductive learning problem is to select the most appropriate one.

3 Complexity of the least complex hypothesis

Suppose that a hypothesis complexity measure $c(H)$ is defined and let $H_t = H_t(E)$ be a hypothesis that is correct for all the training examples in E. For every training set E it is theoretically possible to find many different hypotheses H_t and to determine their complexity $c(H_t)$. By selecting the hypothesis of a minimal complexity ($arg\ min_t\ c(H_t)$), the complexity of the least complex hypothesis from the hypotheses space that is correct for all the examples in E can be determined. In this work it is called the least complex correct hypothesis, and its complexity, denoted by $g(E) = min_t\ c(H_t)$, is the so-called CLCH value (Complexity of the Least Complex correct Hypothesis). If E is an empty set or a set consisting of examples of only one class then $g(E) = 0$ by definition. From the definition for the target hypothesis H_T it follows that $g(D) = c(H_T)$.

Definition 1. A complexity measure c is *reasonable* if for two hypotheses, H_1 and H_2, where H_2 is obtained by conjunctively or disjunctively adding conditions to H_1, the following relation holds: $c(H_1) \leq c(H_2)$.[3]

The paper assumes that the used complexity measure $c(H)$ is reasonable.

Lemma 2. *Let $c(H)$ be a complexity measure and $g(E) = min_t\ c(H_t)$.*

$$\textit{If } E_1 \subset E_2 \textit{ then } g(E_1) \leq g(E_2).$$

Proof: Every hypothesis that is correct for all the examples in E_2 is correct also for all the examples in E_1. So the correct hypothesis of a minimal complexity for E_2 is also correct for all the examples in E_1, and $g(E_2)$ can not be smaller than $g(E_1)$. □

4 Complexity curve

Definition 3. A training set E is called *saturated* if $g(E) = c(H_T)$. Otherwise it is called *non-saturated.*

[3] All complexity measures used in practical inductive systems are reasonable. But, for example, if $c(H)$ is a reasonable complexity measure then the complexity measure defined as $-c(H)$ or as $1/c(H)$ is not reasonable.

Intuitively, the notion of a saturated training set E means that E has (more than) enough training examples for inducing a correct target hypothesis H_T. A consequence of Definition 3 and Lemma 2 is that, under conditions of an unchanged language bias and complexity measure, any subset of a non-saturated training set is also non-saturated, and any superset of a saturated training set remains saturated. The fact $g(D) = c(H_T)$ implies that any non-saturated training set E can be transformed into a saturated set by adding training examples to E.

Suppose that the complexity $c(H_T)$ is finite. Under this condition, the ideal curve of $g(E)$ values for one of the many possible sequences of training sets with the following property:

$$\emptyset \subset E_1 \subset E_2 \subset ... \subset E_n ... \subset E_S \subset ... \subset D$$

is presented in Figure 1. In the rest of the paper, this curve is called a *complexity curve*. In reality, this function is insreasing in a stepwise manner, where the step magnitude depends on the sensitivity of the used complexity measure.

Fig. 1. An ideal curve of CLCH (Complexity of the Least Complex correct Hypothesis) values for a sequence of training domains E with the following property: $\emptyset \subset E_1 \subset E_2 \subset ... \subset E_n \subset ... \subset E_S \subset ... \subset D$.

As a consequence of Lemma 2, the curve is monotonically increasing, and $g(E) \leq c(H_T)$ for every E. The curve must have a saturation point (S in Figure 1) since the smallest subset of D, denoted by E_S, exists which completely determines the target hypothesis and for which it is true that $g(E_S) = c(H_T)$. The training sets E, $E_S \subset E$, with CLCH values in the saturation part of the complexity curve (at the right of point S) are saturated training sets, whereas the training sets E, $E \subset E_S$, at the left of point S are non-saturated.

The saturation property has some interesting characteristics:

- The CLCH value $g(E)$ is defined with respect to the given hypothesis space. Changes in the language bias can change the hypothesis space and the set of hypotheses over which the CLCH value for E is computed. So the changes in the language bias can change the $g(E)$ values and the conditions for the saturation property of the training set. If by changes in the learning bias a non-saturated training set is converted into a saturated set, this is a reliable sign that a more appropriate bias was found.
- The CLCH value is defined with respect to a selected complexity measure. Although the properties of the complexity curve do not depend on the properties of the complexity measure (except that it must be reasonable), this does not mean that the properties of the complexity measure have no influence on the practical curve form for the given training set E. Namely, the same training set can be saturated for one complexity measure and non-saturated for another. The choice of an appropriate complexity measure is in this sense similar to the choice of an appropriate language bias and depends exclusively on the characteristics of domain D. In most cases, a consequence of the use of an inappropriate complexity measure and/or an inappropriate hypothesis space is that a training set E should contain a larger number of examples in order to be saturated.
- For an example domain D, many different sequences of training sets with the property $\emptyset \subset E_1 \subset E_2 \subset ... \subset E_n \subset ... \subset D$ can be constructed, resulting in different complexity curve forms. If the same E_n occurs in different set sequences, then, given a fixed language bias and complexity measure, it can not be saturated according to one complexity curve and non-saturated according to another. This follows from the fact that the saturation property is defined by the condition $g(E) = c(H_T)$ which does not depend on the properties of other elements in the sequence. All the complexity curves are still monotonically increasing because of the property that any subset of a non-saturated training set is non-saturated and that any superset of a saturated training set is saturated.

The most important implication of the properties of the saturated training sets is that the conditions for successful Occam's razor based induction can be formulated as theorems. On the other hand, the complexity curve has also interesting properties when noisy examples are included in the training set. On this basis, a novel noise handling approach can be defined (see Section 8).

5 Occam's razor theorems

Definition 4. Two hypotheses are *substantially different* if they predict a different class for at least one example in D.

Theorem 5. *A saturated training set E is a necessary condition for the applicability of Occam's razor.*

Proof: Suppose that E is a non-saturated training set, therefore $g(E) < c(H_T)$. The target hypothesis H_T is defined as the simplest hypothesis in the hypothesis space that is correct for all the examples in D, therefore $g(D) = c(H_T)$. Since $g(E) < g(D)$, a hypothesis H_t, which is correct for all the training examples in E and has the minimal complexity, has a smaller complexity than the complexity of the least complex hypothesis that is correct for all the examples in D. This means that H_t can not be correct for all the examples in D and that it is substantially different from H_T with a worse prediction accuracy in D. In the case of a non-saturated training set E, the application of Occam's razor will therefore *always* result in the selection of a wrong target hypothesis. The conclusion is that a saturated training set is a necessary condition for the applicability of Occam's razor. □

It must be noted that the consequences of Theorem 5 do not depend on the properties of the selected complexity measure. Non-optimal prediction results obtained by Occam's razor, reported in [17], can be interpreted as the consequence of non-saturated training sets used in the experiments. This follows from the fact that improved prediction accuracy was obtained by hypotheses modifications on the domain parts in which training sets did not have any training examples [17].

Definition 6. A complexity measure is *ideal* if it is so sensitive that no two substantially different hypotheses H_1 and H_2 in the hypothesis space have the same complexity, and that it can not hold that $c(H_1) = c(H_2) = g(E)$.

Theorem 7. *An ideal complexity measure is a necessary condition for the applicability of Occam's razor.*

Proof: Suppose that a non-ideal complexity measure is used and that there are two substantially different correct hypotheses H_1 and H_2 for which $c(H_1) = c(H_2) = g(E)$, where $g(E)$ is the complexity of the least complex hypothesis correct for all examples in E. In this case, the correct target hypothesis $H_T = H_1$ or $H_T = H_2$ can be selected only by chance. Consequently, an ideal complexity measure is a necessary condition for the applicability of Occam's razor. □

The sufficient condition for Occam's razor applicanbility can now be defined.

Theorem 8. *If $E \subset D$ is a saturated training set, and the hypothesis complexity measure c is ideal, then the simplest hypothesis that is correct for all the examples in E is correct also for all the examples in D.*

Proof: For a saturated training set E it is true that $g(E) = c(H_T)$. In the case of an ideal complexity measure no two substantially different hypotheses can have the same complexity value. This means that the simplest hypothesis for E is the same as, or non-substantially different from, the target hypothesis for D. The non-substantial difference means that the two hypotheses have the same class predictions over the whole domain D. Consequently, a saturated training set and an ideal complexity measure are a sufficient condition for the applicability of Occam's razor. □

The importance of Theorem 8 is that it theoretically proves that successful induction is possible. But the practical usefulness of the theorem is very restricted. The main problem is that the saturation property of the training set E can not be tested by verifying the condition $g(E) = c(H_T)$ because $c(H_T)$ could be determined only from D. Although a practical algorithm for the saturation test is suggested in the following section, this test (or any other test based only on the training set) regardless of its precision, can never guarantee that E is indeed saturated. Moreover, no practical application of the Occam's razor principle can guarantee that the induced hypothesis is correct for the whole D or that no other better hypothesis exists.

An additional problem regarding Theorems 7 and 8 is that it is difficult or even impossible to guarantee that a complexity measure is ideal for the given hypothesis space. So the second condition of Theorem 8 can not be fulfilled in practice either. A non-ideal complexity measure can also lead to an additional unreliability of the practical algorithm for the saturation test.

The conclusion is that in practice applying Occam's razor can never guarantee the optimality of the chosen hypothesis. But applications of very different practical complexity measures that are all known to be non-ideal have undoubtly shown that the selection of an appropriate complexity measure is not a critical problem and that effective induction is possible also using non-ideal complexity measures. Results presented in Section 9 show that even using a complexity measure of a rather low sensitivity (a large number of substantially different hypotheses having the same complexity) can enable induction with a high prediction accuracy. Theorem 7 shows that theoretically only a distinction between ideal and non-ideal complexity measures is important. In practice, when the user can choose among different non-ideal measures, the preference should be given to measures that are more sensitive.

6 Practical saturation test

The form of the complexity curve presented in Figure 1 suggests that the saturation property of E may be detected by checking whether in its subset or superset neighbourhood the CLCH values do not change. As in real learning situations only subsets of E can be formed, the practically implementable saturation test can be stated as follows:

"If for all the possible subsets E_x that can be formed by the exclusion of one (or few) examples from E it is true that $g(E_x) = g(E)$, then E is saturated".

The problem of this test is that it can not guarantee that E is saturated. Namely, in practice, the complexity curve does not always have such an ideal form as presented in Figure 1; non-saturated training sets E may also have small areas of subsets of constant CLCH values. This can especially occur in the case of unrepresentable training sets, e.g., when all the examples in E are from a small part of D. Generally, by increasing the number of E_x subsets in the test, the reliability of the test can be increased but the answer that a training set is saturated can never be completely reliable. An additional problem, arising due to a non-ideal complexity measure, is that the condition $g(E_x) = g(E)$ can be satisfied although in the ideal case it should not be.

Algorithm 1 tests whether a training set E, given as the input to the algorithm, is saturated or not. In addition, the output of the algorithm is the so-called *critical subset* A, containing examples whose elimination may lead to a saturated training set.

Algorithm 1 $SaturationTest(E)$

Input: E (training set)
Parameter: r (saturation test level)
Output: $(true, \emptyset)$ or $(false, A)$, where A is the critical subset of E

```
flag f ← true, critical set A ← ∅
determine g(E)
for a = 1,...,r do
    while a different subset A, A ⊂ E, |A| = a can be selected do
        form E_x = E \ A
        determine g(E_x)
        if g(E_x) < g(E) then flag f = false and exit the for loop
    end while
end for
if flag f = false then output (false, A)
else output (true, ∅)
```

The practical number of E_x subsets is variable and depends on domain properties, the number of training examples, and the expected result reliability. In practice, the number of E_x subsets is limited by the number r of examples which may be excluded from E in order to obtain E_x subsets. In Algorithm 1, the range of r is set to $[1, 3]$.

7 Complexity curve for noisy training sets

The presented definitions of the CLCH value $g(E)$ and the complexity curve enable a novel systematic approach to noise handling by example elimination. This is currently the most interesting practical implication of the theory presented in the paper.

Let us define a noisy example as an example in D for which the target theory T and, consequently, the target hypothesis H_T are not correct. There can be different reasons for the occurrence of noisy examples in the training set E. Such examples make the inductive learning task much more complex because the basic assumption used in previous sections that H_T must be correct for all the training examples does not hold. The problem of noise handling has been approached in different ways: noise-handling mechanisms can be incorporated in search heuristics, in stopping criteria or in post-processing (e.g., [12]). Systems employing such procedures are called *noise-tolerant* systems since they try to avoid overfitting the possibly noisy training set.

Let us suppose that e_n is a noisy example for which the target hypothesis is not correct. If e_n is in contradiction with one or more training examples in E then it is easy to detect noise by detecting contradictions, i.e., examples that differ only in their class value. By eliminating a pair of examples that causes the contradiction, noise can be eliminated from the training set. This is a simple and straightforward noise elimination procedure, but consequently some non-noisy examples can be eliminated as well. But more frequently, e_n is not in contradiction with any of the training examples in E. Suppose that e_n is such a noisy example and that in the training set E there are no contradictions even when it includes noise. This condition is necessary because no hypothesis can be correct for all the training examples if it includes contradictions, and consequently the value $g(E)$ for such E can not be determined.

Theorem 9. *Assume an ideal complexity measure. If E is a saturated subset of D, e_n is a noisy example, and $E_n = E \cup \{e_n\}$, then $g(E) < g(E_n)$.*

Proof: Lemma 2 implies that $g(E) \leq g(E_n)$. The least complex hypothesis correct for all the examples in E is the target hypothesis (since E is a saturated training set) and for E_n the least complex hypothesis is not the target hypothesis (since the target hypothesis is not correct for e_n). As the used complexity measure is ideal, the complexity of two substantially different hypotheses must be different, hence $g(E) < g(E_n)$. □

The ideal curve of $g(E)$ values for a sequence of training sets with the property $\emptyset \subset E_1 \subset E_2 \subset ... \subset E_{n-1} \subset E_n \subset ...$ where E_{n-1} is a saturated non-noisy training set and E_n is obtained from E_{n-1} by adding of one (or more) noisy examples, is presented in Figure 2. The increase of the $g(E)$ value after the inclusion of noisy examples is a consequence of Theorem 9.

8 Noise handling by noise elimination

Suppose that E_n is obtained from a noiseless and saturated set E_{n-1} by adding of exactly one noisy example e_n. When applying the saturation test (Algorithm 1) on the set E_n, the result is that E_n is detected as non-saturated because there exists $E_x = E_{n-1} = E_n \setminus \{e_n\}$ such that $g(E_x) < g(E_n)$. This means that after adding a noisy example e_n the saturated training set is no longer saturated. This also means that Algorithm 1 can be used as a procedure to detect the presence of noise in the training set: it enables to identify a noisy example e_n as the difference between the starting training set E_n and its subset E_x which results in a lower $g(E_x)$ value, i.e., $\{e_n\} = E_n \setminus E_x$.

To generalize, more than one noisy training example can be added to a saturated training set, thus the approach can be modified by extending the range of

Fig. 2. An ideal curve of CLCH (Complexity of the Least Complex correct Hypothesis) values for a sequence of training domains E, where $E_n = E_{n-1} \cup \{e_n\}$, and e_n is a noisy example.

E_x subsets. Consequently, by again reversing the argument, Algorithm 1 can be used iteratively so that in each iteration one or a few detected noisy examples are excluded from the training set, and such a reduced training set is the input for the next iteration of noise elimination. This approach is general but it can not guarantee that the detection of noisy examples will be successful in all the situations. Namely, it is theoretically possible that noisy examples occur in such a combination that the elimination of any single noisy example does not enable to decrease the CLCH value. But even for very noisy practical applications this is actually not a serious problem, especially if the saturation test level used in the saturation test equals 2 or 3.

The iterative noise elimination algorithm that results in the reduced training set with eliminated noisy examples is presented in Algorithm 2.

Algorithm 2 $NoiseElimination(E)$

Input: E (training set)
Output: E (reduced training set), A (critical example set)
$A \leftarrow \emptyset$
repeat
apply $SaturationTest(E) = (f,\, A')$
if $(f,\, A') = (true,\, \emptyset)$ **then** exit the repeat loop
else $E \leftarrow E \setminus A'$, $A \leftarrow A \cup A'$
end repeat
output $(E,\, A)$

An advantage of this algorithm is the explicit detection of potentially noisy examples (set A) while by noise tolerant systems they can be determined only indirectly from the properties of the selected hypothesis. These examples can be shown to the user, who can decide whether the examples are actually due to noise and can be eliminated, or whether they represent exceptions that need to be covered by the induced hypothesis.

The problem of the suggested noise detection and elimination procedure is that Algorithm 1 (used as a saturation test) can not distinguish whether the training set is indeed a saturated set including few noisy examples, or whether it is a non-noisy but non-saturated set. In both cases, the algorithm detects a non-saturated training set and tries to find, by example eliminations, its saturated subset. This practically means that examples will be eliminated also when the starting set is noiseless but not-saturated. It is hard to predict how many falsely noisy examples will be eliminated in such a case. The good property, however, is that even in such an unfavourable situation, after iterative elimination of quasi noisy examples the result will be a reduced saturated training set that correctly captures some subconcept of the target theory.

Compared with other noise handling procedures used in inductive learning, the described noise elimination approach has a worse time complexity. It is the consequence of its iterative nature where in every iteration only one or a few noisy examples can be detected and eliminated. And each iteration is time consuming because it includes many CLCH value computations for E_x subsets where the number of different E_x subsets must be large in order to guarantee the algorithm's noise sensitivity. For practical purposes, a substantially quicker heuristic algorithm for noise elimination has been developed [5] based on the above ideas. On the other hand, the main advantage of the algorithm is that noise handling is separated from the hypothesis construction process. Because of that, hypothesis construction is not influenced by noisy examples and therefore the employed learning techniques do not need to incorporate noise handling.

9 Experimental results with noise elimination

The presented saturation test and noise elimination algorithms are implemented in the ILLM (Inductive Learning by Logic Minimization) system [3], a propositional rule learner characterized by the systematic usage of the Occam's razor principle. The system constructs rules in the mixed DNF-CNF form from a set of (automatically or user defined) literals. The used hypothesis complexity measure in ILLM is defined as the number of different literals used to construct a rule. This complexity measure turns out to be well suited for the realization of the described algorithms because it enables that the CLCH value computations can be performed by relatively fast and simple covering algorithms over the set of all example pairs built of one positive and one negative training example. Although this property significantly improves the time complexity of the CLCH value computations, in ILLM a heuristic covering algorithm and a heuristic modification of the noise elimination algorithm had to be implemented in order to obtain a

Number of introduced class errors	LINUS-ILLM Reduced Total number of of eliminated examples	Correctly eliminated ex.	Acc.	LINUS-ILLM Original Acc.
0	3.6 (5,4,2,3,4)	0.0 (0,0,0,0,0)	99.8	99.5
10	13.3 (15,14,12,13,14)	10.0 (10,10,10,10,10)	99.8	97.7
20	24.0 (25,24,22,25,24)	20.0 (20,20,20,20,20)	99.8	95.1
30	36.2 (38,38,32,33,35)	30.0 (30,30,30,30,30)	99.8	93.6
40	41.4 (58,44,43,19,43)	35.0 (40,40,40,16,39)	98.4	92.2
50	39.2 (31,40,46,39,40)	33.0 (26,36,33,33,37)	95.9	90.2

Table 1. Averages and numbers of eliminated examples in 5 domains with 1000 training examples in datasets with 0–50 class errors, together with the average classification accuracy in the reduced datasets and the original datasets prior to noise elimination.

reasonable overall execution speed. See [3] for the details of the ILLM algorithm, whereas the descriptions of the heuristic CLCH value computation and heuristic noise elimination algorithms can be found in [4, 5].

9.1 KRK chess endgame

The results of the ILLM noise elimination algorithm are presented for the well-known inductive logic programing problem of learning illegal positions in a King-Rook-King chess endgame. Five domains with 1000 training examples were selected. A selected number of intentionally introduced class errors was introduced in the training dataset, resulting in datasets with 0, 10, 20, 30, 40 and 50 incorrect class assignments. Test sets consisted of 4000 examples. For other details of the experiment see [4], including the description of the LINUS transformation procedure [8] for solving this relational learning problem. LINUS using ILLM was used for hypothesis formation.

The results are given in Table 1. The obvious expectation that the training sets of 1000 training examples without intentionally introduced noise are saturated, was disproved by the experiments. The noise elimination algorithm resulted in the elimination of 2-5 'noisy' training examples per experiment. The accuracy of rules constructed from reduced training sets was 99.8% . The result seems surprising since the KRK domain is known to be noiseless. However, it is also known that in this domain there is a subconcept (white king blocks the check given by the white rook) with a small number of examples that describe this situation. Examples for this subconcept (although correct) are detected, because of their small number, by the described noise elimination algorithm as potentially noisy examples and are therefore eliminated from the training sets. Generated rules from the reduced training sets did not include the critical subconcept and the prediction accuracy resulted in the accuracy of 99.8%.

When a small amount of intentional classification noise was added to the training examples (between 10 and 30 examples per experiment), the noise elimination procedure successfully detected all the actual noise and additionally excluded examples representing the critical subconcept described above. The result is that the induced rules from the reduced training sets have the same accuracy of 99.8% as those obtained from the noiseless domain. With more than 3% of inserted noise (more than 30 corrupted examples), the noise elimination could not correctly detect all the noisy examples. It also sometimes detected as 'noisy' the examples that are neither noisy nor examples of the critical subconcept. The achieved accuracies are in this situation lower than those obtained for the noiseless domain, but they are still very high.

The right column in Table 1 (accuracy of LINUS-ILLM without noise elimination) is also very interesting. Let us first observe the results in the first row of Table 1. One would expect that in the case when it is in advance known that a domain is noiseless (a very rare practical situation) it might be better not to use the suggested noise elimination algorithm. In this case rule induction is done from the training set which includes a small number of examples of the critical subconcept. But in contrast with the expectations, the prediction accuracy of 99.5% is lower than the one obtained by eliminating examples for the critical subconcept. The reason is that the training sets including the examples of the critical subconcept are non-saturated (as detected also by the noise elimination algorithm) and learning from such domains can sometimes lead to unexpected results which mostly depend on the characteristics of the used complexity measure. On the other hand, exclusion of the examples of the critical subconcept has the drawback that some correct examples have been eliminated but it has enabled that the reduced training sets can be saturated and that all other subconcepts can be induced effectively. The advantage of the suggested noise elimination algorithm is even more obvious if training sets include noise which can be seen from other rows in the last column in Table 1.

9.2 Early diagnosis of rheumatic diseases

Another experimental domain is a real medical problem of early diagnosis of rheumatic diseases which is known to be a difficult machine learning problem due both to its nature and to the imperfection in the dataset. The domain includes 8 diagnostic classes. In order to enable noise elimination, each training set was transformed to 8 different two-class subproblems. Noise elimination, described in detail in [5], was performed on each two-class subproblem where an example was eliminated from the original training set if it was detected as noisy in any of the subproblems. In this way, the reduced training set was generated. The performance of the noise elimination algorithm was tested so that the well known CN2 algorithm [2] was used on both the original and reduced training sets. Experiments were performed on 10 different partitions of data into 70% training and 30% testing examples. For each partition, in total 3 experiments were done: CN2 with the significance test for noise handling applied to the original training set, CN2 without significance test applied to the original training set, and CN2

Partition	Accuracy			Relative information score		
	CN2-ST Original	CN2-NoST Reduced	CN2-NoST Original	CN2-ST Original	CN2-NoST Reduced	CN2-NoST Original
1	47.5	45.3	38.1	17.0	26.0	21.0
2	45.3	44.6	44.6	20.0	28.0	23.0
3	51.1	47.5	45.3	17.0	24.0	19.0
4	44.6	38.8	43.9	17.0	20.0	24.0
5	46.0	41.7	40.3	21.0	25.0	22.0
6	49.6	50.4	48.2	15.0	24.0	26.0
7	44.6	46.8	42.4	21.0	31.0	27.0
8	41.0	43.2	38.8	21.0	25.0	19.0
9	43.9	48.2	45.3	16.0	29.0	23.0
10	39.6	43.2	41.7	23.0	25.0	23.0
Average	45.3	45.0	42.9	18.8	25.7	22.7

Table 2. Accuracy and relative information score results on 10 partitions of the rheumatology training set.

without significance test applied to the reduced training set. Results for the classification accuracy and relative information score are given in Table 2 [5].

In order to compare the results of the proposed noise elimination algorithm to the CN2 significance test noise-handling mechanism, columns CN2-ST-Original and CN2-NoST-Reduced in Table 2 need to be be compared. This is actually the most interesting comparison since, in terms of the classification accuracy, CN2 with the significance test (CN2-ST) is known to perform well on noisy data. On the other hand, in order to observe the effect of noise elimination, the results in columns CN2-NoST-Original and CN2-NoST-Reduced need to be compared.

The effect of noise elimination (CN2-NoST-Reduced) is comparable to CN2 (CN2-ST-Original) in terms of the classification accuracy, but a significant improvement was achieved in terms of the relative information score [7].

10 Summary

This work discusses the conditions for the Occam's razor applicability. To do so, the notion of the complexity of the least complex hypothesis (CLCH value) that is correct for all training examples is introduced. On this basis, the complexity curve that reflects a dependency between a sequence of training sets and the corresponding CLCH values is presented. Conditions for the existence of the saturation part of the complexity curve are given. It is shown that a training set whose CLCH value is in the saturation part of the complexity curve, and the ideal sensitivity of the used complexity measure are the necessary and sufficient conditions for the application of the Occam's razor principle in hypothesis formation, guaranteeing the optimal prediction accuracy of the induced hypothesis. Based on the properties of the complexity curve for intentionally introduced noisy examples, the conditions for effective noise elimination are elaborated and an iterative noise elimination procedure is proposed. The results of empirical evaluation show the usefulness of the presented approach to noise elimination.

Acknowledgements

This research was financially supported by the Croatian Ministry of Science, the Slovenian Ministry of Science and Technology, and the ESPRIT Project 20237 Inductive Logic Programming 2. The authors are grateful to an anonymous reviewer for many valuable comments on the submitted version of the work, and to Sašo Džeroski for contributing to the evaluation of the noise elimination algorithm on the rheumatology dataset, collected by the specialists of the University Medical Center in Ljubljana. Our thanks goes also to Vladimir Pirnat, Aram Karalič and Igor Kononenko who prepared this data in a form appropriate for the experiments.

References

1. Brodley, M.: Recursive automatic bias selection for classifier construction. Machine Learning **20** (1995) 63–94
2. Clark, P., Niblett, T.: The CN2 Induction Algorithm. Machine Learning **3** (1989) 261–284
3. Gamberger, D.: A minimization approach to propositional inductive learning. In Proc. of the 8th European Conference on Machine Learning (1995) 151–160
4. Gamberger, D., Lavrač, N.: Noise detection and elimination applied to noise handling in a KRK chess endgame. In Proc. of the 5th International Workshop on Inductive Logic Programming (1996) 59–75
5. Gamberger, D., Lavrač, N., Džeroski, S.: Noise Elimination in Inductive Concept Learning: A case study in medical diagnosis. In Proc. of the 7th International Workshop on Algorithmic Learning Theory (1996) 199–212.
6. Kohavi, R., Wolpert, D.H.: Bias Plus Variance Decomposition for Zero-One Loss Functions. In Proc. of the 13th International Conference on Machine Learning (1996) 275–283
7. Kononenko, I., Bratko, I.: Information-based evaluation criterion for classifier performance. Machine Learning **6** (1991) 67–80
8. Lavrač, N., Džeroski, S.: Inductive Logic Programming: Techniques and Applications. Ellis Horwood (1994)
9. Li, M., Vitányi, P.: An Introduction to Kolmogorov Complexity and its Applications. Springer (1993)
10. Quinlan, J.R.: Induction of Decision Trees. Machine Learning **1** (1986) 81–106
11. Quinlan, J.R.: Learning Logical Definitions from Relations. Machine Learning **5** (1990) 239–266
12. Quinlan, J.R.: C4.5: Programs for Machine Learning. Morgan Kaufmann (1992)
13. Rao, R., Gordon, D., Spears, W.: For every generalization action, is there really an equal or opposite reaction? Analysis of conservation law. In Proc. of the 12th International Conference on Machine Learning (1995) 471–479

14. Rissanen, J.: Modeling by the shortest data description. Automatica **14** (1978) 465–471
15. Schaffer, C.: A conservation law for generalization performance. In Proc. of the 11th International Conference on Machine Learning (1994) 259–265
16. Stahl, I.: Compression Measures in ILP. In L. De Raedt (ed.): Advances in Inductive Logic Programming IOS Press (1996) 295–307
17. Webb, G.I.: Further Experimental Evidence against the Utility of Occam's razor. Journal of Artificial Intelligence Research **4** (1996) 397–417