Towards a Structured Analysis of Approximate Problem Solving: a Case Study in Classification

Perry Groot and **Annette ten Teije** and **Frank van Harmelen**
Division of Mathematics and Computer Science,
Faculty of Sciences, Vrije Universiteit,
Amsterdam, The Netherlands

Abstract

The use of approximation as a method for dealing with complex problems is a fundamental research issue in Knowledge Representation. Using approximation in symbolic AI is not straightforward. Since many systems use some form of logic as representation, there is no obvious metric that tells us 'how far' an approximate solution is from the correct solution. This article shows how to do a structured analysis of the use of an approximate entailment method for approximate problem solving, by combining theoretical with empirical analysis. We present a concrete case study of our approach: we use a generic approximate deduction method proposed by Cadoli and Schaerf to construct an approximate version of classification reasoning. We first derive theorems that characterise such approximate classification reasoning. We then present experiments that give further insights in the anytime behaviour of this approximate reasoning method.

Keywords: Approximate Problem Solving, Classification, Anytime Inference.

Introduction

The use of approximation as a method for dealing with complex problems is a fundamental research issue in Knowledge Representation. There are two well known reasons for choosing approximation over exact problem solving. First, exact problem solving may be computationally infeasible. Approximation allows us to reduce the computational complexity of problem solving, and enables anytime reasoning (Dean & Boddy 1988). Second, exact problem solving may not result in any solutions because of inconsistent or incomplete data used. Approximation allows us to make problem solving more robust (Schaerf & Cadoli 1995; ten Teije & van Harmelen 1996).

Using approximation in symbolic AI is not straightforward. Since many systems use some form of logic as representation, there is no obvious metric that tells us 'how far' an approximate solution is from the correct solution.

The literature on approximate and anytime reasoning of the last decade (e.g., (Russell & Zilberstein 1991; Zilberstein & Russell 1996; Zilberstein 1996)) has studied specific *algorithms*: their anytime behaviour, their performance profiles, compositionality of such algorithms, monitoring and control of such algorithms, etc. However, a declarative characterisation of such algorithms is often lacking.

On the other hand, work such as (Schaerf & Cadoli 1995) and (Dalal 1996) have provided declarative formalisms for approximate reasoning, but this work only tackles general logical deduction, and needs to be made more concrete for specific forms of AI problem solving such as planning or classification. Some attempts in this direction have been made by ourselves for diagnosis ((ten Teije & van Harmelen 1996; 1997)) and by Wasserman et al. for belief revision (Chopra, Parikh, & Wassermann 2001).

In this article we try to bridge the gap between these algorithm studies on the one hand and the analytical/declarative characterisations on the other hand. We focus in particular on the approximate deduction method of Cadoli and Schaerf (Schaerf & Cadoli 1995). The method is general and can be applied to any problem that can be formalized in (propositional) logic and uses logical entailment for inferencing. This method has a number of desirable properties for an approximation method, such as reduced computational costs, sound as well as complete approximations, and incremental approximations. However, the method also has a number of limitations. In particular, it is not immediately obvious what the effect is of applying the method on a specific problem domain such as diagnosis or classification. Furthermore, the method uses a parameter S resulting in a whole spectrum of approximations that range from zero to optimal precision. Practical usefulness of the method therefore depends on the choice for S, making this choice a crucial part of the method. Currently, the method has not been evaluated beyond diagnosis in (ten Teije & van Harmelen 1996; 1997) and belief revision (Chopra, Parikh, & Wassermann 2001) by means of a quantitative and qualitative analysis.

This article shows how to do a structured analysis of the use of an approximate entailment method for approximate problem solving. The approach consists of two steps. **Theoretical analysis:** In this step the properties of the approximation method applied on a specific problem domain are analyzed by using the rules of the logic and the properties of the approximation method. This step tries to limit the choices for the parameter S resulting in a smaller search space of useful settings (and changes) for the crucial approximation parameter. **Empirical analysis:** In this step heuristics for the parameter S are determined and experiments are set up that measure the quality of these heuristics. The experiments include choosing problem instances, a quality measure for the heuristics, and a way to compare the measured qualities of the heuristics.

This article presents a concrete case study of the above mentioned approach. First we describe the chosen approximation method and the chosen problem domain. The chosen approximation method is the approximate entailment operator of Cadoli and Schaerf (Schaerf & Cadoli 1995). The chosen problem domain is (taxonomic) classification. Thereafter, the analysis of the approximate entailment operator applied to classification is presented. First we present the theoretical analysis. We give boundaries for the parameter S and we show that for classification the chosen approximation technique is useful from an anytime perspective, but not useful from a robustness perspective. Second we present the empirical analysis. These experiments give further restrictions on S. Two heuristics are defined and applied in several ways. Thereafter conclusions are given of the performed case study and of our general approach towards approximate problem solving. This paper finishes by looking at steps for future work.

Approximate entailment

This section gives a short overview of the approximate entailment operators by Cadoli and Schaerf (Schaerf & Cadoli 1995) that allows for a weaker/stronger inference relation. Throughout this section we assume that there is an underlying finite language L used for building all the sentences. In the following we denote with S a subset of L.

Definition 1 (S-3-interpretation) *An S-3-interpretation of L is a truth assignment which maps every letter l of S and its negation $\neg l$ into opposite values. Moreover, it does not map both a letter l of $L \setminus S$ and its negation $\neg l$ into 0.*

Definition 2 (S-1-interpretation) *An S-1-interpretation of L is a truth assignment which maps every letter l of S and its negation $\neg l$ into opposite values. Moreover, it maps every letter l of $L \setminus S$ and its negation $\neg l$ into 0.*

The names given to the interpretations defined above can be explained as follows. For an S-1-interpretation there is *one* possible assignment for letters outside S, namely false for both x and $\neg x$. For an S-3-interpretation there are *three* possible assignments for letters outside S, namely the two classical assignments, plus true for both x and $\neg x$. As a classical interpretation allows *two* possible assignments for letters, such an interpretation is sometimes referred to as a 2-interpretation.

Satisfaction of a formula by an S-1- or S-3-interpretation is defined as follows. The formula is satisfied by an interpretation σ if σ evaluates the formula written in Negated Normal Form (NNF) into true using the standard rules for the connectives.

The notions of S-1- and S-3-entailment are now defined in the same way as classical entailment: A theory T S-1-entails a formula ϕ, denoted by $T \models_1^S \phi$, iff every S-1-interpretation that satisfies T also satisfies ϕ. S-3-entailment is defined analogously and denoted by $T \models_3^S \phi$.

Let $S, S' \subseteq L$ and let $\models_i^S \Rightarrow \models_i^{S'}$ denote $T \models_i^S \phi \Rightarrow T \models_i^{S'} \phi$. The definitions given above then lead to the following result:

Theorem 3 (Approximate Entailment) *Let $S, S' \subseteq L$, such that $S \subseteq S'$, then*

$$\models_3^{\varnothing} \Rightarrow \models_3^S \Rightarrow \models_3^{S'} \Rightarrow \models_2 \Rightarrow \models_1^{S'} \Rightarrow \models_1^S \Rightarrow \models_1^{\varnothing} .$$

This theorem tells us that $\models_3^S$ is a sound but incomplete approximation of the classical entailment $\models_2$, whereas $\not\models_1^S$ is a sound but incomplete approximation of $\not\models_2$ (i.e., $\not\models_1^S \Rightarrow \not\models_1^{S'} \Rightarrow \not\models_2$).

Furthermore, the theorem states that the accuracy of the approximations can be improved by increasing the parameter S until the approximations coincide with the classical entailment.

In the remainder of this paper we only focus on S-3-entailment. The following examples (Schaerf & Cadoli 1995) show that the modus ponens rule does not always hold for S-3-entailment.

Example 4 *Let L be $\{a, b\}$, and T be $\{a, \neg a \vee b\}$. $T \models b$ holds. If $S = \{a\}$ then $T \models_3^S b$. Every S-3-interpretation satisfying T must map a into 1. Then $\neg a$ must be mapped into 0 as $a \in S$. Therefore, as $\neg a \vee b \in T$, b must be mapped into 1.*

Example 5 *Let L be $\{a, b, c\}$, and T be $\{a, \neg a \vee b, \neg b \vee c\}$. $T \models c$ holds, but if $S = \{a\}$, $T \models_3^S c$ does not hold. The mapping that maps the literals a, b and $\neg b$ into 1 and all remaining literals into 0 is an S-3-interpretation that satisfies T, but does not satisfy c.*

Another way to look at S-3-entailment is that it resembles removing all clauses with a letter not in S. Let T be a theory and γ be a formula. Then for a clause c the following relation holds:

$$T \wedge c \models_3^S \gamma \Leftrightarrow T \models_3^S \gamma,$$

if c contains a letter that does not occur both in S or γ.

Classification

This section describes the conceptualization of classification given by (Jansen 2003). In classification the goal is to identify an object as belonging to a certain class. The object is described in terms of a (possibly incomplete) set of observations. Some assumptions have to be made about the representation of the task domain to provide us a vocabulary that can be used to define classification. In (Jansen 2003) six basic ontological types are defined, namely *attribute, object, value, class, feature,* and *observation*. An attribute is a quality which can be associated with a list of possible values. The (finite) set of attributes is denoted with $\mathcal{A}$. It is assumed that each attribute can only have one value at a time. A class will be denoted by the letter c. The set of all classes is denoted by $\mathcal{C}$. A feature is an admissible attribute-value (AV) pair. A feature will be denoted by the letter o. The set of all features is denoted by $\mathcal{O}$. Objects that need to be classified are described by a finite number of AV-pairs. These AV-pairs are called observations. The set of observations for a particular object will be denoted by Obs. The set of attributes occurring in an element e of the domain (e.g., a class, a set of observations) will be denoted by A_e.

A domain theory, denoted by DT, is a conjunction of class descriptions. Class descriptions can be represented in a number of ways. Within this article it is assumed that classes are represented by necessary conditions. For example

$$c \rightarrow (a_1 \vee \ldots \vee a_n) \wedge \ldots \wedge (b_1 \vee \ldots \vee b_m).^1$$

The class name is represented by the proposition c and implies its features (i.e., AV-pairs). Features are here represented as atomic propositions with an index for ease of representation. a_1 represents the feature where a designates the attribute and the index $_1$ a certain value. For example

$$\begin{aligned} blackbird \quad \rightarrow \quad & ((plumage\ =\ black) \vee \\ & (plumage\ =\ brown)) \wedge \\ & (bill\ =\ yellow). \end{aligned}$$

Furthermore, we assume that whenever $c \rightarrow a_1 \vee \ldots \vee a_n$ is part of the domain theory, each $a_1, \ldots, a_n$ has the same type (i.e., they contain the same attribute).

It is assumed that an attribute can only have one value at a time. In case multi-valued attributes are transformed into a number of binary attributes additional rules need to be added to the domain theory to enforce this assumption. This can be done by adding rules for each feature of the following form:

$$a_1 \rightarrow (\neg a_2 \wedge \ldots \wedge \neg a_n).$$

Under this representation the candidate solutions S with respect to some classification criteria can be formulated using logical entailment. The classification criteria we consider are weak, strong, and explanative classification. In weak classification a class c is a solution when it is consistent with the domain theory DT and the observations Obs. In strong classification a class c is a solution when the domain theory together with c explains all observations. That is, we want candidate solutions to actually possess the properties that have been observed. In explanative classification a class c is a solution if the class is explained by the observations. That is, a class is a candidate solution if all its properties are observed. Note that in strong classification a candidate solution may have *more* properties while in explanative classification a candidate solution may have *less* properties than the ones actually observed. Using the above representation these classification criteria can be formalized as follows:

Definition 6 *Let DT be a domain theory containing class descriptions with necessary conditions in which disjunctions are not allowed. Then the classification criteria can be formalized as follows:*

Weak classification:

$$S_W = \{c \mid DT \cup \{c\} \cup Obs \not\models \bot\}.$$

Strong classification:

$$S_S = \{c \mid DT \cup \{c\} \models Obs\} \cap S_W.$$

Explanative classification:

$$S_E = \{c \mid \{o \mid DT \cup \{c\} \models o\} \subseteq Obs\} \cap S_W.$$

These definitions can also be found in the literature (e.g., (Jansen 2003)). Usually the definitions for strong- and explanative classification are simplified by omitting the intersection with S_W as this has no influence on the formalizations of the criteria when using propositional logic and the classical entailment operator. As we will substitute an approximate entailment operator $\models_3^S$ for the classical entailment operator $\models$, *which also occurs in* S_W, we keep the intersection with S_W part of our formalization. Of the formalizations given in Definition 6, only the formalization of weak classification can also be used in case of class descriptions with disjunctions.

Related work

Classification has been studied before by other researchers, for example in the context of Description Logics (DLs). As (Schaerf & Cadoli 1995) studies the approximation of DLs using S-1- and S-3-entailment the question rises what the differences are between our approach and theirs.

Although the same term 'classification' is used in our work as well as within the context of DLs the meaning, however, is quite different. In DLs classification means placing a concept expression in the proper place in a taxonomic hierarchy of concepts. This is done by checking known subsumption relations between concepts without referring to specific instances. In our approach classification means identifying to which class a specific instance belongs. In DLs this is called *instance checking*.

Another difference between our work and the work done in DLs is the use of three different classification criteria (weak, strong, and explanative). Classification criteria are not discussed in the context of DLs. An instance simply belongs to a concept when it satisfies all the properties of the concept. Any property satisfied by the instance which is not mentioned in the concept has no influence on the outcome. Note that this is the same as explanative classification used in our work.

Finally we like to point out that our work focuses on a specific problem task, namely classification, whereas (Schaerf & Cadoli 1995) uses their approximation techniques in a more general setting, i.e., approximate satisfiability within a certain logic. The general nature of (Schaerf & Cadoli 1995) makes it unclear if their proposed approximation technique is applicable to practical problem solving. Our work fills some of this gap. Furthermore, the approximation technique of (Schaerf & Cadoli 1995) uses a parameter S which is important for the effectiveness of the method. Our work addresses this problem and proposes a structured analysis for choosing this parameter.

Approximating classification using S-3-entailment

This section gives the results of a theoretical analysis of the influence of the approximate S-3-entailment relation of Cadoli and Schaerf on the various classification forms. This

[1] Note that $c \rightarrow a \wedge b$ is equivalent to $(c \rightarrow a) \wedge (c \rightarrow b)$ and equivalent to $\neg c \vee (a \wedge b)$. We will use these equivalent representations without further notice.

analysis is the first step in our approach to restrict the crucial parameter S by exploiting the formalization of our specific problem domain (classification). The results of this analysis will be used in the empirical analysis for concrete guidelines for setting boundaries for the parameter S.

We assume that class descriptions are given by necessary conditions. The approximations are obtained by selecting $\models_3^S$ instead of the usual entailment and additionally choosing an appropriate set of propositional letters S. More precisely, we define

$$\begin{array}{rcl} S_{W3}^S & = & \{c \mid DT \cup \{c\} \cup Obs \not\models_3^S \bot\}, \\ S_{E3}^S & = & \{c \mid \{o \mid DT \cup \{c\} \models_3^S o\} \subseteq Obs\} \cap S_{W3}^S, \\ S_{S3}^S & = & \{c \mid DT \cup \{c\} \models_3^S Obs\} \cap S_{W3}^S, \end{array}$$

for some set S of propositional letters.

These definitions can be used to approximate the classical classification forms from above (too many solutions) or below (too few solutions):[2]

Lemma 7 *Let $S, S' \subseteq L$, such that $S \subseteq S'$. Then*

$$\begin{array}{ccccccccc} S_W & \subseteq & S_{W3}^{S'} & \subseteq & S_{W3}^S & \subseteq & S_{W3}^{\varnothing} & = & \mathcal{C}, \\ S_E & \subseteq & S_{E3}^{S'} & \subseteq & S_{E3}^S & \subseteq & S_{E3}^{\varnothing} & = & \mathcal{C}, \\ \varnothing & = & S_{S3}^{\varnothing} & \subseteq & S_{S3}^S & \subseteq & S_{S3}^{S'} & \subseteq & S_S. \end{array}$$

The effect of using S-3-entailment in the definitions of classical classification can be described in simpler terms. Before doing so, let $A_{incorrect}$ be a set of attributes and $O_{incorrect}$ be the set of all features with an attribute in $A_{incorrect}$. Weakening weak classification by allowing the values of attributes in $A_{incorrect}$ to be inconsistent can then be formalized as

$$S_W^{incorrect} = \{c \mid DT \cup \{c\} \cup (Obs \setminus O_{incorrect}) \not\models \bot\}.$$

Approximate weak classification (using S-3-entailment) can be described as follows:

Theorem 8 $S_{W3}^{\mathcal{C} \cup (\mathcal{O} \setminus O_{incorrect})} = S_W^{incorrect}$.

Our analysis of S_{W3}^S also showed that excluding a class c from the parameter S results in c always being a solution, i.e., $c \notin S \Rightarrow c \in S_{W3}^S$. A reasonable choice for S should therefore include the set of all classes $\mathcal{C}$ to obtain a reasonable approximation of classical weak classification. Hence, the above theorem characterizes approximate weak classification for all reasonable choices of S.

Theorem 8 gives us an interpretation of S_{W3}^S. It states that S_{W3}^S is the same as weak classification that allows certain attributes to be inconsistent. Furthermore, Theorem 8 gives us a method for computing $S_W^{incorrect}$ as the algorithm given by Cadoli and Schaerf (Schaerf & Cadoli 1995) for computing S-3-entailment can be used.

Approximate explanative classification can be described as follows:

Theorem 9 *Let $D \subseteq \mathcal{C}$, and $O \subseteq \mathcal{O}$. If the domain theory DT has only class descriptions without disjunctions then*

$$S_{E3}^{(\mathcal{C} \setminus D) \cup O} = S_E \cup D.$$

[2]Proofs are omitted but can be obtained from (Groot 2003).

Hence, this theorem states that the usefulness of approximating explanative classification by using S-3-entailment is limited. There is no interpretation of S_{E3}^S possible in terms of explanative classification in which missing or inconsistent attributes are allowed as there is a one to one correspondence between classes not in S and classes in S_{E3}^S. More precisely, any sequence of sets $S_1 \subseteq S_2 \subseteq \cdots \subseteq S_n$ used as parameter in S_{E3}^S to approximate explanative classification with increasing accuracy as S_i increases, leads to the sequence of solution sets $S_E \cup D_1 \supseteq S_E \cup D_2 \supseteq \cdots \supseteq S_E \cup D_n$, where $D_i = \mathcal{C} \setminus S_i$, approximating S_E in an obvious but uninformative way.

Approximate strong classification can be described as follows:

Theorem 10 *Let $D \subseteq \mathcal{C}$ and $O \subseteq \mathcal{O}$. If the domain theory DT has only class descriptions without disjunctions then*

$$S_{S3}^{D \cup O} = S_S \cap D.$$

Note that this theorem states that the usefulness of approximating strong classification by using S-3-entailment is limited. There is no interpretation of S_{S3}^S possible in terms of strong classification in which missing or inconsistent attributes are allowed as there is a one to one correspondence between classes not in S and classes not in S_{S3}^S. Strong classification with S-3-entailment can only be used to restrict strong classification to a subset of the class hierarchy.

Summarizing the theoretical analysis, S-3-entailment applied to classification is of limited use. Only weak classification with S-3-entailment can be interpreted in terms of inconsistent attributes. Nevertheless, with S-3-entailment one obtains three approximate classification methods that can be used to incrementally increase or decrease the number of solutions. However, the quality of these approximating methods depend on the chosen order of S. The following section looks at this issue in more detail.

Empirical analysis

The analysis of the previous section results in a number of restrictions for reasonable choices for S. In the case of strong classification and explanative classification, the previous section showed that adding observations to S had no influence on the outcome. Hence, for both forms of classification, only the order in which the classes are added to S is important. In the second step of our approach we use these restrictions for developing concrete guidelines for S in the context of strong classification and explanative classification in the following two subsections respectively. (Additional experiments omitted from this article were performed to corroborate the theoretical results. These can be obtained from (Groot 2003).)

Strong classification

With approximate strong classification the set of solutions is approximated from below (sound but incomplete), i.e., by adding more classes to S more strong solutions are obtained. A reasonable choice for S therefore seems to be to prefer a class c over a class d when class c is more likely to be a strong solution. As for strong classification $A_{Obs} \subseteq A_c$

must hold (i.e., all attributes in Obs must also occur in c and their values must match), this seems to be the case when (1) the number of attributes in the class description of c is higher than the number of attributes in the class description of d, and/or (2) the number of possible values that can be assigned to attributes of c is less than the number of possible values that can be assigned to attributes of d. (More precisely, the second heuristic is computed by taking the product of the number of possible values per attribute.)

These two heuristics lead to four possible orders. S_1: apply only the first heuristic, S_2: apply only the second heuristic, S_3: apply the first followed by the second heuristic (in case two classes have the same number of attributes), and S_4: apply the second heuristic followed by the first heuristic.

The goal is to compute the average approximating behaviour of the various orders of S on a set of observations. As the number of strong solutions will be different for different object descriptions, each result on one specific object description needs to be normalized before the average can be computed. In the following the quality of an approximating algorithm will be mapped on the interval [0,1] by dividing the obtained value through the maximum value that can be obtained, i.e., the number of classical solutions.

For the experiment of strong classification some conditions were set for the theory while the rest of the theory was created randomly. The conditions consisted of 100 classes and 10 attributes. The maximum allowed number of values for an attribute was set at 5 and the class descriptions contained between 5 and 10 attributes. Furthermore, 30 random object descriptions were created containing one, two, or three observations. (This theory was chosen so that the object descriptions did not result in too few strong solutions.) The results of the experiment are shown in Figure 1.

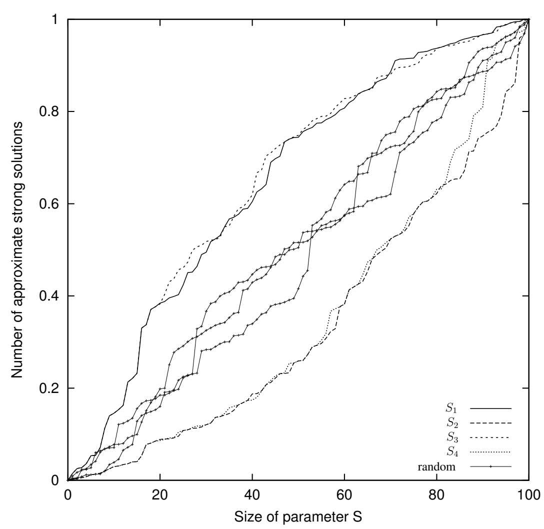

Figure 1: Results of various orders used in approximate strong classification.

Figure 1 shows the results of the orders $S_1, \ldots, S_4$ as well as three random orders. This means that the order of the classes were chosen at random before the experiment, and then this random order was used on all 30 object descriptions. These random orders are used for comparison with the heuristic orders. More random orders were tested in practise, which resulted in similar behaviour as shown in Figure 1, but they are left out for readability.

The results show that the order that selects classes with the highest number of attributes, i.e., S_1, performs much better than a random order for S. Figure 1 also shows that S_1 can be further improved by also considering the number of possible values, i.e., S_3. The orders that first select the classes with the lowest number of possible values, i.e., S_2 and S_4 perform much worse than a random order of S. This may seem surprising at first, but not when considering the results of order S_1. Classes with the lowest number of possible values are often classes with less attributes (as one less attribute means one less number in the product used to compute the second heuristic), hence orders S_2 and S_4 tend to take the opposite order of S than S_1.

Explanative classification

With approximate explanative classification the set of solutions is approximated from above (complete but unsound), i.e., by adding more classes to S more incorrect solutions are discarded. A reasonable choice for S therefore seems to be to prefer a class c over a class d when class c is less likely to be an explanative solution. As for explanative classification $A_c \subseteq A_{Obs}$ must hold, this seems to be the case when (1) the number of attributes in the class description of c is higher than the number of attributes in the class description of d, and/or (2) the number of possible values that can be assigned to attributes of c is higher than the number of possible values that can be assigned to attributes of d.

These two heuristics lead to four possible orders. E_1: apply only the first heuristic, E_2: apply only the second heuristic, E_3: apply the first followed by the second heuristic (in case two classes have the same number of attributes), and E_4: apply the second heuristic followed by the first heuristic.

The goal of the experiment is to compute the average approximating behaviour of the various orders. As the set of classic solutions is approximated from above (incorrect solutions are discarded when S increases) normalization of the result of each object description is more complicated than the case of strong classification. To obtain an increasing quality function on the interval $[0, 1]$ as in Figure 1 we apply for a theory with n classes the normalization $(n - v(i, o, s))/(n - v(n, o, s))$ where $v(i, o, s)$ is the size of the solution set at iteration i for some object description o and some chosen order s. Note that $v(n, o, s)$ is equal to the number of classical explanative solutions.

The theory used for the experiment consisted of 100 classes and 10 attributes. The maximum number of allowed values per attribute was set at 7 and the class descriptions contained between 1 and 3 attributes. Furthermore, 30 random object descriptions were created consisting of eight, nine, and ten attributes. The results of the experiment are shown in Figure 2.

In this experiment, all heuristic orders produce a better approximating behaviour when compared with the random

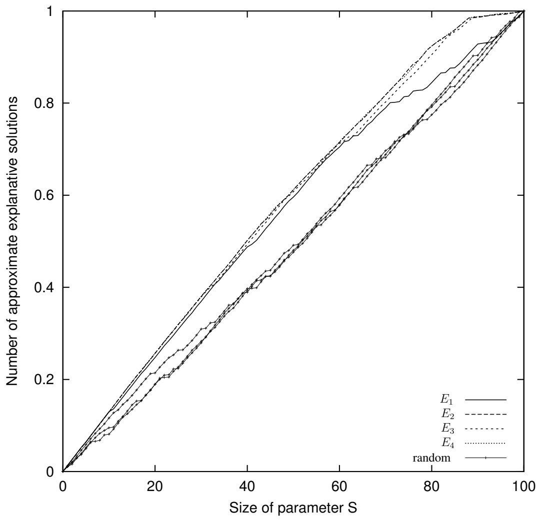

Figure 2: Results of various orders used in approximate explanative classification.

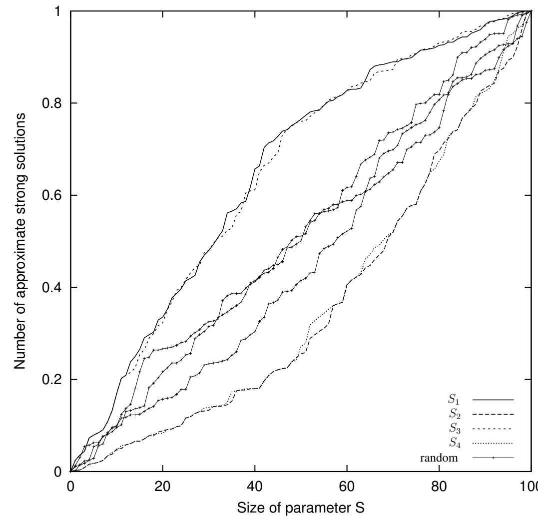

Figure 3: Results of various orders used in repeated experiment of approximate strong classification.

orders. However there is little to choose between the various heuristic orders. Order E_1, which only considers the number of attributes, performs worst, but it can be improved by also considering the number of values (E_3). The best orders are E_2 and E_4. Note that classes with many possible values for its attributes are often also classes with many attributes. Hence, the second heuristic tends to prefer classes also preferred by the first heuristic. Therefore, all orders based on the two heuristics result in very similar approximating behaviour.

Note that similar experiments on a different theory resulted in similar behaviour as in Figure 2. However, the difference between the various orders $E_1, \ldots, E_4$ was even less, probably caused by a lower number of explanative solutions.

Experiments continued

The previous experiments were only performed on one theory, which is too few to reach a general conclusion. Although similar behaviour may be expected, more theories should be used in the experiments.

One question that rises from the previous experiments is if the experiments are robust (or repeatable), i.e., will the results of two experiments be similar when both experiments are run using similar conditions. Therefore, another theory was created using the same conditions (10 attributes with upto 5 possible values and 100 classes with 5 upto 10 possible attributes) on which the experiments from the previous section were repeated.

In this section we focus on the experiments with strong classification. The results of the repeated experiment are shown in Figure 3.

Note that the results of Figure 3 are very similar to the results of Figure 1. The heuristics S_1 and S_3 perform better than the random heuristics and heuristics S_2 and S_4 perform worse than the random heuristics. This indicates that the experiment is robust. There is only a slight difference between the heuristics S_1 and S_3. In Figure 1 the heuristic S_3 performed slightly better than the heuristic S_1, but in Figure 3 it is the other way around, heuristic S_1 performs slightly better than heuristic S_3. As the heuristic S_3 uses more knowledge about the domain to construct the order than heuristic S_1 one might expect it to have a better approximating behaviour. However, these experiments do not confirm this.

Another question that rises from the previous experiments is if the results also hold for theories constructed using different conditions. Another two theories, which we will refer to as `domain-2` and `domain-3` were therefore created based on the theory of the previous experiment, but with slightly different conditions. The theory `domain-2` consisted of 10 attributes with 1 upto 5 possible values and 100 classes, but the class descriptions contained only 5 or 6 attributes (instead of 5 upto 10 attributes). The theory `domain-3` consisted of 100 classes with 5 upto 10 attributes and 10 attributes, but the attributes were only allowed upto 2 possible values (instead of 5). The results of approximate strong classification with these theories are shown in Figures 4 and 5.

Note that the results of Figure 4 are very similar to the results of Figure 1 or Figure 3. There is somewhat more difference between the heuristics S_2 and S_4, but they still perform worse than the random orders. However, Figure 5 is somewhat different than the other results of approximate strong classification. There is now much less difference between the four heuristic orders and the three random orders. Nevertheless, heuristic S_3 seems to outperform the other orders. The heuristic S_1 which performed well in the other experiments only performs well until the parameter S contains about 60 elements. Thereafter, heuristic S_1 drops below the other heuristics. Hence, all experiments indicate that heuristic S_3 is the best choice for an order in which elements are added to the parameter S.

The experimental results also show that the quality of the

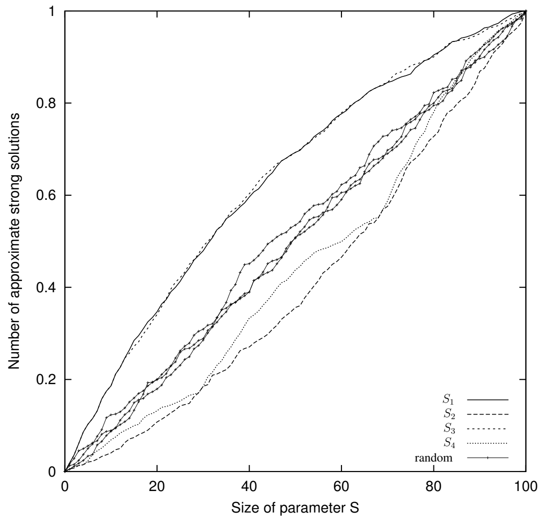

Figure 4: Results of various orders used in approximate strong classification using theory `domain-3`.

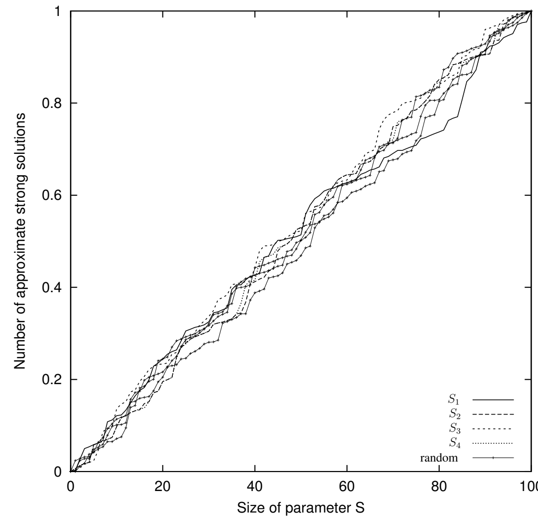

Figure 5: Results of various orders used in approximate strong classification using theory `domain-2`.

heuristic order S_3, i.e., the difference in quality between the order S_3 and other orders, depends on the theory used. The results shown in Figures 4 and 5 indicate that the quality of the approximation depends on certain parameters of the theory used. Some parameter will have more influence on the quality of the approximation then other parameters. Figures 4 and 5 suggest that in this case study the variation in the number of attributes in a class description has more influence on the quality of the approximation than the number of possible values for an attribute. However, more experiments should be run before any conclusion can be drawn.

Finally note that the specific results obtained in these empirical experiments are not the main purpose of this article, but rather the approach followed to make the approximation technique of Cadoli and Schaerf applicable to a specific problem domain. First, a theoretical analysis was performed for limiting the choices of the crucial parameter S in the approximation method of Cadoli and Schaerf. Second, an empirical analysis was performed for evaluating specific choices for S.

Conclusions

This study began with the premise that complex problem solving can benefit from approximation techniques. In particular, approximation can be used to reduce computational costs of problem solving or to make problem solving more robust against incorrect/incomplete data.

A general technique for approximating logical inference problems one can use is replacing the standard entailment operator by the approximate entailment operators developed in (Schaerf & Cadoli 1995). Although this technique is general and has some desirable properties, little is known about the method when applied to specific problems. This article analyzed the applicability of the method of Cadoli and Schaerf to classification.

The main results of the theoretical analysis are the formulas obtained that describe the effect of replacing the entailment operator by the S-3-entailment operator for the three classification forms. It was proven that, using S-3-entailment, approximate weak classification behaves identical to weak classification that allows certain observations to be inconsistent. Furthermore, approximate strong classification behaves identical to strong classification restricted to a subset of all classes. The solutions to approximate explanative classification are identical to classical explanative classification to which a set of classes is added.

A theoretical analysis does not provide all necessary information about a method. Although a theoretical analysis helps in setting boundaries for the parameter S, it does not provide clear guidelines for choosing which letters should be added to the parameter S and in which order. The empirical analysis was performed to fill in some of these gaps. Two experiments were performed, one in the context of strong classification and one in the context of explanative classification, for validating heuristic orders of S for a given theory.

Although an empirical analysis is useful to accompany a theoretical analysis, some drawbacks of this approach were found. Although some of these may be obvious, one should be aware of these drawbacks. First, results of an empirical analysis may depend on the particular instance used for analyzing a method. For example, the empirical results shown may depend on the theory used for classification. Determining the structure of such a theory and representing it as a set of parameters may be hard if not impossible. Second, even if a domain can be characterized as a set of parameters, there may be too many parameters to explore using an empirical analysis. Third, if not all instances of a problem can be used in an empirical analysis, then the results of this analysis cannot be used as facts. An empirical analysis can be used to corroborate a statement or theory and provide insight in a problem. However, an empirical analysis cannot be used to prove a property of a method unless all instances are explored.

Future work

The approach towards a structured analysis of approximate problem solving taken in this article contains two forms of analyses. The theoretical analysis for obtaining properties and setting boundaries for the parameter S, and the empirical analysis for constructing concrete guidelines for the parameter S. Both of these analyses allow room for further research.

The theoretical analysis gives a detailed analysis of S-3-entailment used in classification. However, the analysis can be extended in two ways. First, the analysis is restricted to classes defined by necessary conditions. Other representations should be included in future research. Second, all obtained results are for S-3-entailment. No results are obtained for S-1-entailment. Both theoretical analyses have to be accompanied with an empirical analysis.

In the empirical analysis it was clear which (heuristic) order performed better when two orders were compared with each other. In both experiments (i.e., strong classification and explanative classification), the quality of an order was always higher, lower, or the same when compared with the quality of another order. When the quality of an order is sometimes higher and sometimes lower than the quality of another order, it becomes unclear which order should be preferred. A general framework should be created for the comparison of such performance profiles. This framework should answer the question which performance profile is preferred with respect to a number of preferred properties.

Besides extending the theoretical and empirical analysis, other variations of the performed case study can be explored. These are other problem domains, another logic, or other approximation techniques.

References

Chopra, S.; Parikh, R.; and Wassermann, R. 2001. Approximate belief revision. *Logic Journal of the IGPL* 9(6):755–768.

Dalal, M. 1996. Anytime families of tractable propositional reasoners. In *International Symposium on Artifical Intelligence and Mathematics AI/MATH-96*, 42–45. Extended version submitted to Annals of Mathematics and Artifical Intelligence.

Dean, T., and Boddy, M. 1988. An analysis of time-dependent planning problems. In *Proceedings of the 7th National Conference on Artificial Intelligence*, volume 1, 49–54. San Mateo: Morgan Kaufmann.

Groot, P. 2003. *A Theoretical and Empirical Analysis of Approximation in Symbolic Problem Solving*. Ph.D. Dissertation, SIKS, Vrije Universiteit, Amsterdam.

Jansen, M. G. 2003. *Formal explorations of knowledge intensive tasks*. Ph.D. Dissertation, University of Amsteram.

Russell, S. J., and Zilberstein, S. 1991. Composing real-time systems. In *Proceedings of the 12th International Joint Conference on Artificial Intelligence, Sydney, Australia*, volume 1, 212–217. San Mateo: Morgan Kaufmann.

Schaerf, M., and Cadoli, M. 1995. Tractable reasoning via approximation. *Artificial Intelligence* 74:249–310.

ten Teije, A., and van Harmelen, F. 1996. Computing approximate diagnoses by using approximate entailment. In Aiello, G., and Doyle, J., eds., *Proceedings of the Fifth International Conference on Principles of Knowledge Representation and Reasoning (KR-96)*. Boston, Massachusetts: Morgan Kaufman.

ten Teije, A., and van Harmelen, F. 1997. Exploiting domain knowledge for approximate diagnosis. In Pollack, M., ed., *Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence (IJCAI-97)*, volume 1, 454–459. Nagoya, Japan: Morgan Kaufmann.

Zilberstein, S., and Russell, S. J. 1996. Optimal composition of real-time systems. *Artificial Intelligence* 82(1-2):181–213.

Zilberstein, S. 1996. Using anytime algorithms in intelligent systems. *Artificial Intelligence Magazine* fall:73–83.