Delaying the Choice of Bias : A Disjunctive Version Space Approach

Michele Sebag*
LMS CNRS-URA 317
Ecole Polytechnique
91128 Palaiseau FRANCE
Michele.Sebag@polytechnique.fr

Abstract

This paper is concerned with alleviating the choice of learning biases via a two-step process:
– The set of all hypotheses that are consistent with the data and cover at least one training example, is given an implicit characterization of polynomial complexity. The only bias governing this induction phase is that of the language of hypotheses.
– Classification of further examples is done via *interpreting* this implicit theory; the interpretation mechanism allows one to relax the consistency requirement and tune the specificity of the theory at no extra induction cost.
Experimental validations demonstrate very good results on both nominal and numerical datasets.

1 INTRODUCTION

In a seminal paper, Mitchell (1980) introduced the term of *bias* to refer to *any basis for choosing one generalization over another, other than strict consistency with the training instances.*

Learning biases proceed from at least two motivations: improve the predictive power of the induced theory (Mitchell 1980) and make induction tractable (Muggleton & De Raedt 1994). Top-down learners are driven by optimality criterions (e.g. quantity of information, Gini criterion, MDL principle) allowing one both to cope with noisy data and to restrict the search to optimal or near optimal regions of the hypothesis space (Quinlan 1993, Botta & Giordana 1993). Divide-and-conquer algorithms also employ learning biases to deal with noise, through specifying the maximal number of inconsistencies acceptable for a hypothesis (Michalski 1983, Ganascia 1993, Muggleton 1995) or the noise model (Norton & Hirsh 1993).

Most learning biases (language biases, built-in heuristics, parameters of control of the learner...) are chosen once and for all before induction. Hence the learner constructs a theory biased according to the expert's knowledge — or guesses — concerning the quality of data and the adequate optimality criterion. But what if expert's guesses are wrong ? A trial and error process then takes place (Srinivasan & Muggleton 1995): the relevance of the biases is estimated from the predictive accuracy of the biased theory (e.g. if many test examples are unclassified, the threshold on the maximal number of inconsistencies is likely too low). However, adjusting the biases by trials and errors is rather expensive: each trial implies performing anew induction from scratch.

This paper is concerned with delaying the choice of search biases until the classification step:
Induction is achieved by a learner inspired from the version spaces framework (Mitchell 1982), termed *DiVS* for *Disjunctive Version Spaces*, which characterizes *all* consistent hypotheses that cover at least one training example. This set of hypotheses is given an implicit characterization of polynomial complexity (remind that an explicit characterization of version spaces is of exponential complexity (Haussler 1988)).
Classification of further examples is done via *interpreting* this implicit theory. The point is that the mechanism of interpretation allows one to control the degrees of consistency and specificity of the theory, *for free.*
Within this two-step scheme, the only bias influencing induction is that of the language of hypotheses. The search biases reflecting the quality of the data can be tuned thereafter (typically, depending on the classification results) at no extra inductive cost.

The paper is organized as follows. Section 2 describes algorithm *DiVS*, and shows how embedding the ver-

*And Equipe I & A, LRI, Université Paris-Sud, 91405 Orsay, FRANCE.

sion space framework into a non standard divide-and-conquer approach allows one to resist noisy data without the need for specific search bias. Section 3 details the interpretation of the implicit theory built by *DiVS*; this interpretation allows one to classify further instances and can be tuned depending on the noise and sparseness of the training data. Experimentations on numerical and qualitative problems are discussed in section 5. Some perspectives for further research are last presented.

2 DISJUNCTIVE VERSION SPACE

This section is primarily interested in getting rid of search biases. As stated by Mitchell (1980), unbiased generalization procedures build the set of all hypotheses complete and consistent with training examples, termed *Version Space* (VS); unfortunately, VS suffers from severe practical and computational limitations. These shortcomings are addressed via hybridizing VS and the divide-and-conquer approach (Michalski 1983).
Only attribute-value languages are considered in this paper. An extension of this approach to first order logic can be found in (Sebag & Rouveirol 1995, 1996).

2.1 STATE OF THE ART

VS characterizes the set of consistent hypotheses from its upper bound G, and the set of complete hypotheses from its lower bound S. When there exists no hypothesis both complete and consistent ($S \not\leq G$), the VS fails and this failure is blamed on either too much specialization of G, or too much generalization of S. Failures occur when dealing with erroneous examples, or learning a target concept that does not fit the hypothesis language (typically, when a disjunctive concept is sought for in a conjunctive concept space).

These shortcomings can be addressed in a more or less deterministic way. For instance, Manago and Blythe (1989) employ ID3 as a pre-processor to determine the conjunctive sub-concepts involved in the target disjunctive concept (the leaves of the tree), and thereafter iteratively use VS to characterize these conjunctive sub-concepts.

In another line, Norton and Hirsh (1992) use a model of the noise in the data and hypothesize the true data from the observed (possibly corrupted) data. Each observed example gives rise to a set of supposed examples (possibly true), together with their probability. This amounts to "inversing" the noise model. The *Probabilistic Evidence Combination* process (a) maintains in parallel all VSs consistent with (at least) one supposed example originated from any observed example; (b) computes the *a posteriori* probability of a VS by summing the probabilities of the supposed examples it is consistent with; (c) eventually returns the VS with maximum *a posteriori* probability. This approach has been extended to learn disjunctive concepts (Norton & Hirsh 1993) via a divide-and-conquer approach: positive training examples, termed *seeds*, are considered one at a time; examples not belonging to the same conjunctive sub-concept than the current seed can be regarded as noise with respect to this sub-concept; positive examples covered by the current VS are removed from the training set and another seed is considered.

As mentioned by the authors, Probabilistic Evidence Combination asks the question of how noise models and their parameters are acquired. Practically, the number of supposed examples corresponding to an observed example is in L^K if L is the number of modalities of an attribute, and if the noise model allows K attributes to be simultaneously corrupted. Further, the number of VSs maintained in parallel during the early stages of learning is exponential[1]. Last, this scheme is basically limited in what regards numerical attributes (these are currently handled via some preliminary discretization): to directly handle numerical attributes, one should determine the (continuous) distribution of true data from the observed data, and symbolic induction is ill-prepared to learn from such distributions.

2.2 BUILDING ALL CONSISTENT HYPOTHESES

Our approach is guided by two priorities:
− to survive the lack of relevant bias (and the absence of experts);
− to successfully handle numerical attributes without requiring any preliminary discretization (countless works have reported how much the accuracy of "pure" symbolic induction depends on the discretization stage).

We take advantage of the fact that VS will *not* collapse in one particular case[2]: when it considers a single positive example, termed *seed*, and several negative examples, provided that the list of negative examples does not include the seed. This condition is easy to satisfy by discarding either the negative examples having same description than the seed, or the seed itself.

The VS learnt from a seed (the *star* of the seed) is the disjunction of all consistent hypotheses covering this seed. This disjunction is unduly specific in case false negative examples are encountered, but it yet includes

[1] It decreases thereafter, as most VSs collapse and are removed from the list.

[2] To be precise, VS is also ensured not to collapse if either the list of positive examples or the list of negative examples is empty. However, such VSs would be heavily corrupted in case of noisy examples or disjunctive concept.

consistent hypotheses only. Now, if the seed itself is a false positive example, the corresponding star encompasses hazardous hypotheses. This drawback is addressed by symmetrically considering the target concept and its negation: more precisely, every example in turn is taken as seed and generalized against all examples belonging to other concepts than the seed, termed *counter-examples* to the seed[3]. This hopefully allows hazardous hypotheses learnt from false negative examples to counterbalance hazardous hypotheses learnt from false positive examples.

Formally, $DiVS$ builds a theory $\mathcal{H}$ which is the disjunction of stars learnt from seed Ex, noted $H(Ex)$, for Ex ranging over the training set. $H(Ex)$ includes all hypotheses that cover Ex and discriminate counter-examples to Ex; it is given by the conjunction of the set of hypotheses that cover Ex and discriminate Ce, noted $D(Ex, Ce)$ (with D for *discriminate*), for Ce ranging over the counter-examples to Ex:

> $DiVS$ **Algorithm**
> $\mathcal{H} = False$.
>   `For each` Ex `training example`
> $H(Ex) = True$
>   `For each` Ce `counter-example to` Ex
>   `Build` $D(Ex, Ce)$ *(see section 2.3)*
> $H(Ex) = H(Ex) \wedge D(Ex, Ce)$
> $\mathcal{H} = \mathcal{H} \vee H(Ex)$.

$DiVS$ differs from other divide-and-conquer algorithms in two respects. First, most authors (Michalski 1983, Norton and Hirsh 1992, 1993, Muggleton 1995) restrict themselves to learning the target concept, while we both learn the target concept and its negation. As already said, this allows the effects of noisy positive and negative examples to counterbalance each other *without the need for specific bias*, such as lower bounds on the number of examples covered by a hypothesis, or noise models.
Second, we consider *all* training examples instead of pruning the examples covered by previous hypotheses. The problem with pruning is that it induces an additional bias (the eventual theory depends on the choice of seeds) while increasing the complexity of induction (see section 2.5).

Finally $DiVS$ characterizes all hypotheses that cover at least one training example (the seed) and only cover examples with same label as the seed. In contrast with (Sablon 1995), no additional optimality criterion is introduced, viz. we do not require the number of conjuncts involved in the Disjunctive Version Space to be minimal.

2.3 LANGUAGE OF HYPOTHESES

The elementary step of the $DiVS$approach consists of characterizing the set of hypotheses $D(Ex, Ce)$ that cover the current seed Ex and discriminate a counter-example Ce to the seed.

We restrict ourselves to hypotheses expressed as conjunction of selectors $[att \in V]$, where V denotes a subset of the domain of attribute *att* (Michalski 1983). We furthermore require V to be an interval if *att* is linear (valued in $\mathbb{R}$ or $\mathbb{N}$), and a single value if *att* is nominal (valued in a finite set). These restrictions meet most real-world problems; besides, they ensure that $H(Ex, Ce)$ will differ from the simple negation of Ce. More details on this, among which the handling of tree-structured attributes, can be found in (Sebag 1994).

Assumption. In the remainder, we assume that any two examples belonging to different target concepts can be discriminated in the hypothesis space.

Table 1: a seed and a counter-example

	Shape	*Size*	*Color*	*Thickness*	*class*
Ex	*circle*	*3*	*blue*	*7.25*	*A*
Ce	*triangle*	*12*	*blue*	*3.1*	*B*

Of evidence, a selector can discriminate Ex and Ce only if it is based on an attribute that takes different values for Ex and Ce: examples in Table 1 cannot be differentiated from their color.
Attribute $Size$ does take different values for Ex and Ce. As interval $[0, 12)$ is the maximal range of size including the size of Ex and excluding the size of Ce, selector $[Size \in [0, 12)]$ (for short, $[Size < 12]$) is the maximally general selector built on $Size$ that covers Ex and discriminates Ce; following (Michalski 1983), such a selector is termed *maximally discriminant*.
Similarly, the maximally discriminant selectors respectively built on the linear attribute $Thickness$ and on the nominal attribute *Shape* in our hypothesis language are $[Thickness > 3.1]$ and $[Shape = circle]$.

Clearly, a conjunctive hypothesis covers Ex and discriminates Ce iff it is less general than the disjunction of maximally discriminant selectors:

$$[Shape = circle] \ \vee \ [Size < 12] \ \vee \ [Thickness > 3.1]$$

More generally, let $Sel_k(Ex, Ce)$ be the maximally general selector based on attribute att_k that discriminates Ex from Ce, if it exists, and $false$ otherwise. Then, a hypothesis h belongs to $D(Ex, Ce)$ iff it is less general than $Sel_k(Ex, Ce)$ for some k, i.e. iff it is

[3] The counter-examples to a positive example are the negative examples; and the counter-examples to a negative example are the positive examples. More generally, when the data represent several mutually exclusive concepts, the counter-examples to an example of one concept are the examples of all other concepts.

less general than the disjunction of $Sel_k(Ex, Ce)$ over k (noted $\vee_k Sel_k(Ex, Ce) \prec h$):

$$(\, h \in D(Ex, Ce)\,) \iff (\bigvee_k Sel_k(Ex, Ce) \prec h) \quad (1)$$

By definition, the set $H(Ex)$ of consistent hypotheses covering Ex is given by the conjunction of $D(Ex, Ce)$ for Ce ranging over the counter-examples to Ex:

$$H(Ex) = \bigwedge_{Ce\ counter-example\ to\ Ex} D(Ex, Ce) \quad (2)$$

2.4 STANDARD CLASSIFICATION

One can derive from relations (2) and (1) whether a given instance E is covered by a hypothesis in $H(Ex)$, (noted *E belongs to $H(Ex)$*): E belongs to $H(Ex)$ iff E belongs to every $D(Ex, Ce)$, for Ce ranging over the counter-examples to Ex ; and E belongs to $D(Ex, Ce)$, iff it satisfies[4] at least one selector discriminating Ex from Ce.

But knowing whether E belongs to any $H(Ex)$ gives enough means to classify E, via a nearest neighbor-like process. Let E be defined as "neighbor" to Ex if E belongs to $H(Ex)$; the class of E can thereafter be determined by a majority vote among its neighbors in the training set.

```
Neighbor (E, Ex) :            (E belongs to H(Ex))
   For each Ce counter-example to Ex
      if NOT Belongs(E, D(Ex,Ce))
         return false
   return true
```

```
Belongs(E, D(Ex,Ce)) :
   For each attribute att_k
      if E satisfies Sel_k(Ex,Ce)
         return true
   return false
```

Simply put, *DiVS* rather constructs an oracle than an explicit theory. This oracle achieves the classification of further examples; it is made of theory $\mathcal{H}$ (stored as the list of $D(Ex_i, Ex_j)$, for Ex_i in the training set and Ex_j counter-example to Ex_i), and this theory is interpreted according to relations (1) and (2). More precisely, the actual classifier constructed by *DiVS* consists of $\mathcal{H}$ and of the standard nearest neighbor algorithm, calling the above *Neighbor* boolean function.

2.5 COMPLEXITY

Under the standard assumption that attribute domains are explored with a bounded cost (Hirsh 1992), the complexity of induction in *DiVS* is $\mathcal{O}(N^2 \times P)$, where N denotes the number of training examples and P the number of attributes. This complexity increases up to $\mathcal{O}(N^3 \times P)$ if training examples covered by the current hypotheses are pruned. Classification has same complexity.

This approach resembles that of Hirsh (1992), in the sense it polynomially characterizes the extension of the Version Space. One important difference lies in the fact that *DiVS* can deal with disjunctive concepts and noisy examples; a more thorough discussion will be found in (Sebag 1994).

[4] E satisfies the selector $[att_k \in V]$ iff $att_k(E) \in V$.

3 TUNING THE CLASSIFICATION

This section describes how to adjust the above classification process, in order to account for the rate of noise and the sparseness of the training set.

3.1 TUNING THE CONSISTENCY

Real-world datasets always include false examples; as noted by Clark and Niblett (1987), this implies that the set of strictly consistent hypotheses is *both large and not of the highest predictive power*. Hence, most learners are nowadays concerned with finding hypotheses "consistent enough", i.e. admitting a bounded number of inconsistencies within the training examples, rather than consistent.

Let ε be a positive integer, and let $H_\varepsilon(Ex)$ denote the set of hypotheses that cover Ex and cover at most ε counter-examples to Ex. A given instance E is said neighbor to Ex with inconsistency ε, or for short, *ε-neighbor* to Ex, if it belongs to $H_\varepsilon(Ex)$.

The following simple counting procedure returns *true* if E is ε-neighbor to Ex:

```
ε-Neighbor(E, Ex, ε) :
   NI = 0
   For each Ce counter-example to Ex
      if (NOT Belongs(E, D(Ex,Ce)))
         NI = NI + 1
         If (NI > ε)
            return false
   return true
```

The oracle described in section 3 can thus be modified to accommodate an arbitrary level of noise, via replacing function *Neighbor* by the relaxed *ε-Neighbor* function. Note this implies not extra computational cost.

Let us consider the classifier defined by the nearest neighbor process relying on the above ε-neighbor function. Parameter ε clearly defines a bias on this classifier; and the judicious value for ε depends on the quality of the training set.

But the advantage of our approach is that the degree of inconsistency ε needs not be known at the time of induction, since $\mathcal{H}$ *does not depend* on ε.

In case the expert does not know precisely the rate of noise in his/her data (which is rather frequent), ε will still be adjusted by a trial and error process. But induction is here done once and for all, whereas, whenever the theory produced by induction depends on ε, each trial requires to perform induction anew.

3.2 TUNING THE GENERALITY

By construction, a star $H(Ex)$ includes hypotheses maximally general among the consistent hypotheses covering Ex (the G set). The problem is that $H(Ex)$ may turn out to be *too* general, especially when training data are sparse. Concretely, this shows up as most further instances belong to most stars, hence are considered neighbors to most training examples; via the nearest neighbor process, such instances are all classified in the majority class.

We must thus be able to somehow restrict the generality of hypotheses in $H(Ex)$. This is done via specializing $D(Ex, Ce)$, more precisely the function *Belongs(E,D(Ex,Ce))*: The idea consists in requiring E to satisfy several selectors in $D(Ex, Ce)$ (instead of only one), in order to belong to $D(Ex, Ce)$.
More formally, let M be a positive integer ($M \geq 1$), and consider the $M - of - P$ concept built from selectors $Sel_k(Ex, Ce)$, noted $D_M(Ex, Ce)$: E belongs to $D_M(Ex, Ce)$ if it satisfies at most M selectors $Sel_k(Ex, Ce)$. Of evidence, $D_1(Ex, Ce)$ is $D(Ex, Ce)$, and $D_M(Ex, Ce)$ becomes more and more specific as M increases. The following function computes whether E belongs to $D_M(Ex, Ce)$:

```
M-Belongs(E, D(Ex,Ce), M) :
   NS = 0
   For each attribute att_k
      if E satisfies Sel_k(Ex,Ce))
         NS = NS + 1
         If (NS ≥ M)
            return true
   return false
```

Let now $H_M(Ex)$ be defined as the conjunction of $D_M(Ex, Ce)$, for Ce ranging over the counter-examples to Ex:

$$H_M(Ex) = \bigwedge_{Ce\ counter-example\ to\ Ex} D_M(Ex, Ce)$$

Similarly, $H_1(Ex)$ is $H(Ex)$ and $H_M(Ex)$ becomes more and more specific as M increases.

Proposition : *Let h be a maximally general hypothesis in $H_M(Ex)$; then h does not belong to $H_{M+1}(Ex)$.*
Proof. Let h be a maximally general hypothesis in $H_M(Ex)$, and assume h belongs to $H_{M+1}(Ex)$. For the sake of brevity, the proof is given in the case of a boolean language. Let us write h as $L \wedge h'$, where L is a literal that does not appear in h'.
Let $\mathcal{E}$ be the set of counter-examples Ce to Ex such that L appears in $H(Ex, Ce)$.
The fact that h belongs to $H_M(Ex)$ implies that h belongs to $D_M(Ex, Ce)$ for any counter-example Ce.
Now consider h': h' belongs to $D_M(Ex, Ce)$ for any Ce not in $\mathcal{E}$. But the fact that h belongs to $D_{M+1}(Ex, Ce)$ implies that h' belongs to $D_M(Ex, Ce)$ for any Ce in $\mathcal{E}$. Finally, h' belongs to $D_M(Ex, Ce)$ for every counter-example Ce; hence h' belongs to $H_M(Ex)$, which contradicts the fact that h is maximally general in $H_M(Ex)$. □

A given instance E is said neighbor to Ex with specificity M, or for short, *M-neighbor* to Ex, if it belongs to $H_M(Ex)$. This can be computed by calling *M-Belongs* instead of *Belongs* in function *Neighbor*.

3.3 TUNING BOTH

In order to tune both the consistency and the generality of the classification, we finally define a (ε, M)-neighbor function:

```
(ε,M)-Neighbor(E, Ex, ε, M) :
   NI = 0
   For each Ce counter-example to Ex
      if (NOT M-Belongs(E, D(Ex,Ce), M))
         NI = NI + 1
         If (NI > ε)
            return false
   return true
```

Consider the classifier given by the nearest neighbor process based on the above neighbor function, with tunable inconsistency ε and specificity M, and let us examine the combined effects of parameter M and ε.

Parameter M controls the "size" of star $H_M(Ex)$: the bigger M, the smaller (i.e. the more specific) the stars, the less neighbors an instance E has in the training set. Therefore, the classifier makes less and less mistakes as M increases — but, unfortunately, it also classifies less and less instances: E is unclassified if it has no neighbors. A tradeoff between the rates of misclassified and unclassified examples must thus be sought, via tuning the degree of specificity M. This is a general concern; again, the point is that our approach allows one to set the degree of specificity of the classifier *after* induction instead of *before*.

In opposition, the bigger ε, the bigger (i.e. the more general) the stars, since they are allowed to cover more counter-examples. The effects of M and ε thus counterbalance each other.

4 EXPERIMENTAL VALIDATION

This section reports experimental validation of our learning scheme on several datasets at the UC Irvine Repository (Murphy and Aha 1995).

4.1 EXPERIMENTAL SETTINGS

For each dataset, $\mathcal{H}$ was built as described in section 2, and we vary the consistency ε and specificity M of the classification. Parameter ε is here used as an upper bound on the *percentage* of inconsistencies, rather than on the *number* of inconsistencies, in order to ease the interpretation of the results; ε ranges from 0 to 25%. Parameter M varies in [1..20].

We detail the experiments conducted on two problems: a nominal problem originated from biology and a numerical problem designed by Breiman et al. (1984). Results obtained on other problems are more briefly discussed.

4.2 THE PROMOTER GENE SEQUENCE

Examples are composed of sequences of nucleotides. They are described by 59 attributes valued in $\{A, C, G, T\}$. The associated class gives the promoter activity of the example (boolean). The 106 examples are equally distributed among the two classes. See (Towell et al. 1990) for more details.

Table 2: Errors on the promoter problem estimated by the leave-one-out procedure

Reference Results	Total Errors
KBANN	4
BP	8
k-NN	13
ID3	19

DiVS Results

ε	M	Misc.	Unc.	Total Errors
0	5	3	1	4
	6	4	1	5
	7	2	1	3
	8	1	1	**2**
	9	2	2	4
5 %	5	16	12	28
	6	4	6	10
	7	8	0	8
	8	1	1	**2**
	9	3	3	6

Table 2 first shows some reference results. $KBANN$ is an hybrid learner, where an incomplete human expertise has been encoded into a neural net (NN), and refined using and adapting the NN machinery (Towell et al. 1990); BP denotes the standard backpropagation, k-NN stands for a k-nearest neighbor classifier (with $k = 3$); and $ID3$ needs no presentation here. For various values of ε and M, *DiVS* failures are decomposed into misclassified (Misc) and unclassified (Unc) examples. The number of unclassified examples typically increases when M is either too low (because instances are covered by many hypotheses and tie conflicts are observed), or too high (because many instances are covered by none hypothesis).

The most striking point is that, whereas *DiVS* has been supplied with no additional information, it *can* outperform *KBANN*, which benefits both from available human expertise and from the power of NN. Experimenting with *DiVS* gives precise hints as to the quality of the data: these data likely include very few noise (the classifier behaves better for $\varepsilon = 0$). Also, the redundancy of the description is rather high: no matter what the value of ε is, the optimal value of M is 8. A naive interpretation would be: "it needs at least 8 attributes to make a real difference between two examples".

4.3 THE WAVEFORM PROBLEM

Examples are built by linear combinations of fixed waveforms (Breiman et al. 1984). An example is described by 21 real-valued attributes; examples are equally distributed among three classes.

The waveform problem presents three difficulties from the standpoint of ML algorithms: classes are overlapping and data involve numerical noise[5]; last, the separation of the classes is basically an additive law.

Table 3: Error rates on the waveform pb averaged on 10 independent selections of 300-example training sets, and measured on a fixed 5 000 test set.

Reference Results	% Errors
BP	17.1 ± 1
LD	20.4 ± 1
SIA	24.3 ± 0.7
k-NN	27 ± 1.7
ID3	28 ± 1.8

DiVS Results

ε	M	% Errors
5 %	8	18.5 ± 0.4
	9	19.2 ± 0.2
	10	20.1 ± 0.2
10 %	8	**17.9 ± 0.3**
	9	18.7 ± 0.3
	10	19.3 ± 0.1
15 %	8	18.6 ± 0.1
	9	18 ± 0.3
	10	18.6 ± 0.2

[5] If the three classes are equi-represented, the misclassification rate is lower bounded by 14%.

Table 4: Predictive accuracy observed on other datasets estimated using a ten-fold cross-validation[6].

Dataset	P	N	Majority	best so far	best of DIVS	ε	M
breast-wis	10	699	65.5	C4.5 : 95.4 ± 0.7	95.6 ± 0.6	5%	1
hepatitis	19	155	79.2	C4.5 : 80.0 ± 3.7	84 ± 1	10%	3
iris	4	150	23.3	LD : 98	97.2 ± 1	15%	2
labor-neg	16	53	65.3	C4.5 : 85.7 ± 3.5	93.3 ± 3	15%	1
tic-tac-toe	9	958	65.4	C4.5 : 85.6 ± 1.1	91.4 ± 0.6	10%	2
vote	16	435	61.4	NN : 95.3	93.7 ± 1	0	1
vote-free	15	435	61.4	IDC : 89.1 ± 1.8	89.8 ± 1	5%	2

Table 3 shows results obtained by standard backpropagation (BP), linear discriminant (LD), an AQ-like algorithm optimized by genetic algorithm (SIA) (Venturini 1993), a k-nearest neighbor classifier (k-NN) (with k=1) and $ID3$.

All these results are reported from (Gascuel et al. 1995). Again, the most striking point is that *DiVS can* outperform linear discriminant and k-NN on purely numerical data, in spite of its poor "numerical skills": remind it is only allowed to compare a value with values encountered in the training set.

The variation of the predictive accuracy of DiVS demonstrates that the "true" percentage of noise is near to 15%, and the redundancy of the description is rather high. Note the standard deviation of the accuracy is very low.

4.4 OTHER PROBLEMS

A few other problems have been considered so far. For each of these problems, Table 4 gives the number of attributes (P), the number of examples in the dataset (N), the percentage of examples in the majority class ($Majority$), the best result found in the literature, referencing either $C4.5$ (Quinlan 1993), the decision table approach (IDT) (Kohavi 1995), the decision committees approach (IDC) (Nok & Gascuel 1995), or neural nets (NN) and linear discriminant (LD) (Holte 1993). The best results obtained by *DiVS* are indicated together with the corresponding values of ε and M.

Of course, this comparison is biased since it only reports the best predictive accuracy obtained with *DiVS*. On-going experiments investigate the use of two test sets: the first one allows *DiVS* to internally determine the optimal values of ε and M, and the second one allows one to estimate the actual predictive accuracy of *DiVS*, according to the values determined on the first test set. Preliminary results indicate that optimal values of ε and M are quite steady: actual predictive accuracy is very close to the optimal one indicated above.

5 DISCUSSION AND PERSPECTIVES

After discussing the main strengths and weaknesses of *DiVS*, we present some avenues for further research.

5.1 STRENGTHS

As claimed in the introduction, the presented framework frees induction from all search biases, in the sense that it constructs a theory $\mathcal{H}$ that only depends on the training examples and on the hypothesis language.

In particular, induction does not require any *a priori* knowledge about the quality of the data. This property is much desirable on many real-world problems: most experts are far from knowing beforehand the rate of noise, the noise model or the representativity of their data.
The classification based on $\mathcal{H}$ can nevertheless be tuned according to two parameters, ε and M, that respectively control the degree of consistency and specificity of the hypotheses used for classification. As a matter of fact, the predictive accuracy of *DiVS*, viewed as a function of ε and M, gives precise indications as to the rate of noise and representativity of the data (see Tables 2 and 3). *DiVS* could therefore be used to *a posteriori* determine the quality of the data.

Furthermore, as demonstrated on the waveform problem (Table 3), *DiVS*can successfully handle numerical data without *a priori* discretization; this is a significant advantage compared to the *Probabilistic Evidence Combination* process (Norton & Hirsh 1993), IDT (Kohavi 1995) or IDC (Nok & Gascuel 1995). Another advantage lies in the polynomial complexity of *DiVS* (quadratic with respect to the number of examples, linear with respect to the number of attributes), whereas both IDT and IDC are basically exponential in the number of attributes (in $\mathcal{O}(P^k)$ if P is the number of attributes and k the maximal size of monomials involved in decision tables or committees).

Last, this approach gives outstanding experimental results. These results are clearly over-estimated in Table 4, that reports the results of *DiVS* at its best; however, Table 2 witnesses that DiVS at its best can outperform the state of the art. In our opinion, the quality of these results is mainly explained by the redundancy of the constructed theory. This explanation is accredited by the comparison of $DiVS$ and SIA (Table 3): both learners explore the same search space, and the only difference is that SIA finds the "optimal" rules covering the examples (various optimality functions have been considered), whilst $DiVS$ finds "all" rules. Redundancy has been widely acknowledged as a factor of robustness and reliability (see (Nok & Gascuel 1995) among others). Clearly, the theory $\mathcal{H}$ constructed by $DiVS$ is maximally redundant, in the sense that it includes "all" consistent hypotheses provided they cover at least one training example.

5.2 WEAKNESSES

The main weakness of our approach lies in the poor comprehensibility of $\mathcal{H}$. This is partly due to its form (disjunction of conjunctions of disjunctions); but if $\mathcal{H}$ was classically expressed as a disjunction of conjunctions, it would be no easier to read, since its size is basically exponential in the number of attributes (Haussler 1988).

At first sight, $DiVS$ may just seem another unintelligible learner to be compared with, for instance, neural nets.

However, two important characteristics distinguish $DiVS$ from neural nets. First, $DiVS$ can be extended to first order logic (Sebag & Rouveirol 1995, 1996). Second, in spite of the unintelligibility of $\mathcal{H}$, the classification process is definitely *not* a black box: For any instance E, an intelligible and concise excerpt of $\mathcal{H}$ that "explains" the classification of E can be determined[7]. More details on this second-order intelligibility can be found in (Sebag 1995).

5.3 PERSPECTIVES

As mentioned earlier on, on-going experiments are concerned with an internal determination of optimal values of the biases (parameters) ε and M.

Another perspective of this approach consists in using the experimental information given by the $DiVS$ theory. Some training examples Ex_i are such that $H(Ex_i)$ exactly covers one training example, Ex_i itself; such training examples clearly are isolated, and can be considered corrupted. This information can be used to clean a dataset before learning.

A third perspective consists in extracting from $\mathcal{H}$ an intelligible theory having same predictive accuracy. The challenge is twofold: is there an actual correlation between redundancy and predictive accuracy ? If the answer is yes, this seemingly implies that the trade-off between intelligibility and accuracy is factual (redundancy hinders intelligibility for simple reasons of the size of the theory). If the answer is no, and optimally accurate short theories exist, this asks a further question (see also (Srinivasan & Muggleton 1995)): what could be the relevant optimality function, allowing one to sort the accurate hypotheses ?

Last but not least, this approach will be examined from the standpoint of cognitive modeling: $DiVS$ is able to construct a theory from data with unknown reliability, to tune this theory depending on a posteriori guesses concerning the reliability of these data, and to use the tuned theory to classify further instances. Furthermore, and still more typical of human beings: despite it cannot articulate the theory it has constructed, it can nevertheless provide articulate justifications for its verdicts.

[7]By construction, the classification of a given instance E derives from its neighbors Ex_i in the training set. For any Ex_i neighbor to E, let us consider the most specific hypothesis covering E and Ex_i; it is straightforward to show that this hypothesis is consistent (up to the fixed degree of inconsistency ε), it covers E and Ex_i and votes for classifying E in the class of Ex_i.

Acknowledgments

Many thanks to Marc Schoenauer from CMAP, Ecole Polytechnique, and to Antoine Cornuejols, Nicolas Graner and Karine Causse, from Equipe I& A, LRI, Orsay, for their help regarding the argumentation or the readability of this paper. Thanks also to the anonymous referees, who pointed out missing references.

References

Botta, M. and A. Giordana. Smart+ : A multistrategy learning tool. In *Proceedings of IJCAI-93*, pages 937–943. Morgan Kaufmann, 1993.

Breiman, L., J.H. Friedman, R.A. Olshen, and C.J. Stone. *Classification and Regression by tree.* Wadsworth, Belmont California, 1984.

Clark, P. and T. Niblett. Induction in noisy domains. In I. Bratko and N. Lavrac, editors, *Proc. of European WorkShop on Learning*, pages 11–30. Sigma Press, 1987.

Ganascia, J.-G. TDIS: An algebraic formalization. In *Proceedings of IJCAI-93*, pages 1008–1013. Morgan Kaufmann, 1993.

Gascuel, O. and P. Gallinari et al. Méthodes symboliques-numériques de discrimination. *5 emes Journées Nationales PRC-IA*, pages 29–76, 1995.

Haussler, D. Quantifying inductive bias : AI learning algorithms and Valiant's learning framework. *Artifi-*

cial Intelligence, 36:177–221, 1988.

Hirsh, H. Polynomial-time learning with version spaces. In *Proceedings of National Conference on Artificial Intelligence*, 1992.

Holte, R.C. Very simple classification rules perform well on most commonly used datasets. *Machine Learning*, 11:63–90, 1993.

Kohavi, R. The power of decision tables. In N. Lavrac and S. Wrobel, editors, *Proceedings of ECML-95, European Conference on Machine Learning*, pages 174–189. Springer-Verlag, 1995.

Manago, M. and E. Blythe. Learning disjunctive concepts. In K. Morik, editor, *Knowledge Representation and Organization in ML*, pages 211–225. Springer-Verlag, 1989.

Michalski, R.S. A theory and methodology of inductive learning. In R.S Michalski, J.G. Carbonell, and T.M. Mitchell, editors, *Machine Learning : an artificial intelligence approach*, volume 1. Morgan Kaufmann, 1983.

Mitchell, T.M. Generalization as search. *Artificial Intelligence*, 18:203–226, 1982.

Mitchell, T.M. The need for bias in learning generalizations, 1980, reprinted In *Readings in Machine Learning*, pages 184–191. Morgan Kaufmann, 1991.

Muggleton, S. Inverse entailment and PROGOL. *New Gen. Comput.*, 13:245–286, 1995.

Muggleton, S. and L. De Raedt. Inductive logic programming: Theory and methods. *Journal of Logic Programming*, 19:629–679, 1994.

Murphy, P.M. and D.W. Aha. *UCI Repository of machine learning databases.* http://www.ics.uci.edu/mlearn/MLRepository.html, 1995.

Nok, R. and O. Gascuel. On learning decision committees. In A. Prieditis and S. Russell, editors, *Proceedings of ICML-95, International Conference on Machine Learning*, pages 413–420. Morgan Kaufmann, 1995.

Norton, S.W. and H. Hirsh. Classifier learning from noisy data as probabilistic evidence combination. In *AAAI92: Proceedings of the 10^{th} National Conference on AI*, pages 141–146, AAAI Press, 1992.

Norton, S.W. and H. Hirsh. Learning DNF Via Probabilistic Evidence Combination. In P. Utgoff, editor, *Proceedings of ICML-93, International Conference on Machine Learning*, pages 220–227 Morgan Kaufmann, 1993.

Quinlan, J.R. *C4.5: Programs for Machine Learning.* Morgan Kaufmann, 1993.

Sablon, G. *Iterative Version Space.* PhD thesis, Katholieke Universiteit Leuven, Belgium, 1995.

Sebag, M. Using constraints to building version spaces. In L. De Raedt and F. Bergadano, editors, *Proceedings of ECML-94, European Conference on Machine Learning.* Springer-Verlag, 1994.

Sebag, M. 2^{nd} order understandability of disjunctive version spaces. In *Workshop on Machine Learning and Comprehensibility, IJCAI-95.* Report LRI, Université Paris-Sud, 1995.

Sebag, M. and C. Rouveirol. Constraint inductive logic programming. In L. de Raedt, editor, *Advances in ILP.* IOS Press, 1996.

Sebag, M., C. Rouveirol and J.F. Puget. ILP + Stochastic Bias = Polynomial Approximate Learning. In J. Wnek and R.S. Michaslki, editors, *Proc. of Multi-Strategy Learning.* AAAI Press, 1996, to appear.

Srinivasan, A. and S. Muggleton. Comparing the use of background knowledge by two ILP systems. In L. de Raedt, editor, *Proceedings of ILP-95.* Katholieke Universiteit Leuven, 1995.

Towell, G. J. Shavlik, and M. Noordewier. Refinement of approximate domain theories by knowledge-based artificial neural networks. In *Proceedings AAAI-90*, pages 861–866, 1990.

Venturini, G. SIA : A supervised inductive algorithm with genetic search for learning attribute-based concepts. In Bradzil P., editor, *Proceedings of European Conference on Machine Learning*, pages 280–296. Springer Verlag, 1993.