A Procedure for Multi-Class Discrimination and some Linguistic Applications

Vladimir Pericliev
Institute of Mathematics & Informatics
Acad. G. Bonchev Str., bl. 8,
1113 Sofia, Bulgaria
peri@math.acad.bg

Raúl E. Valdés-Pérez
Computer Science Department
Carnegie Mellon University
Pittsburgh, PA 15213, USA
valdes@cs.cmu.edu

Abstract

The paper describes a novel computational tool for multiple concept learning. Unlike previous approaches, whose major goal is prediction on unseen instances rather than the legibility of the output, our MPD (Maximally Parsimonious Discrimination) program emphasizes the conciseness and intelligibility of the resultant class descriptions, using three intuitive simplicity criteria to this end. We illustrate MPD with applications in componential analysis (in lexicology and phonology), language typology, and speech pathology.

1 Introduction

A common task of knowledge discovery is multiple concept learning, in which from multiple given classes (i.e. a typology) the *profiles* of these classes are inferred, such that every class is contrasted from every other class by feature values. Ideally, good profiles, besides making good predictions on future instances, should be concise, intelligible, and comprehensive (i.e. yielding all alternatives).

Previous approaches like ID3 (Quinlan, 1983) or C4.5 (Quinlan, 1993), which use variations on greedy search, i.e. localized best-next-step search (typically based on information-gain heuristics), have as their major goal prediction on unseen instances, and therefore do not have as an explicit concern the conciseness, intelligibility, and comprehensiveness of the output. In contrast to virtually all previous approaches to multi-class discrimination, the MPD (Maximally Parsimonious Discrimination) program we describe here aims at the legibility of the resultant class profiles. To do so, it (1) uses a minimal number of features by carrying out a global optimization, rather than heuristic greedy search; (2) produces conjunctive, or nearly conjunctive, profiles for the sake of intelligibility; and (3) gives all alternative solutions. The first goal stems from the familiar requirement that classes be distinguished by jointly necessary and sufficient descriptions. The second accords with the also familiar thesis that conjunctive descriptions are more comprehensible (they are the norm for typological classification (Hempel, 1965), and they are more readily acquired by experimental subjects than disjunctive ones (Bruner et. al., 1956)), and the third expresses the usefulness, for a diversity of reasons, of having all alternatives. Linguists would generally subscribe to all three requirements, hence the need for a computational tool with such focus.[1]

In this paper, we briefly describe the MPD system (details may be found in Valdés-Pérez and Pericliev, 1997; submitted) and focus on some linguistic applications, including componential analysis of kinship terms, distinctive feature analysis in phonology, language typology, and discrimination of aphasic syndromes from coded texts in the CHILDES database. For further interesting application areas of similar algorithms, cf. Daelemans et. al., 1996 and Tanaka, 1996.

2 Overview of the *MPD* program

The Maximally Parsimonious Discrimination program (MPD) is a general computational tool for inferring, given multiple classes (or, a typology), with attendant instances of these classes, the profiles (=descriptions) of these classes such that every class is contrasted from all remaining classes on the basis of feature values. Below is a brief description of the program.

2.1 Expressing contrasts

The MPD program uses *Boolean*, *nominal* and *numeric* features to express contrasts, as follows:

[1] The profiling of multiple types, in actual fact, is a generic task of knowledge discovery, and the program we describe has found substantial applications in areas outside of linguistics, as e.g., in criminology, audiology, and datasets from the UC Irvine repository. However, we shall not discuss these applications here.

- Two classes C1 and C2 are contrasted by a Boolean or nominal feature if the instances of C1 and the instances of C2 do not share a value.
- Two classes C1 and C2 are contrasted by a numeric feature if the *ranges* of the instances of C1 and of C2 do not overlap.[2]

MPD distinguishes two types of contrasts: (1) *absolute contrasts* when all the classes can be cleanly distinguished, and (2) *partial contrasts* when no absolute contrasts are possible between some pairwise classes, but absolute contrasts can nevertheless be achieved by deleting up to N per cent of the instances, where N is specified by the user.

The program can also invent *derived features*—in the case when no successful (absolute) contrasts are so far achieved—the key idea of which is to express interactions between the given primitive features. Currently we have implemented inventing novel derived features via combining two primitive features (combining three or more primitive features is also possible, but has not so far been done owing to the likelihood of a combinatorial explosion):

- Two Boolean features P and Q are combined into a set of two-place functions, none of which is reducible to a one-place function or to the negation of another two-place function in the set. The resulting set consists of P-and-Q, P-or-Q, P-iff-Q, P-implies-Q, and Q-implies-P.
- Two nominal features M and N are combined into a single two-place nominal function MxN.
- Two numeric features X and Y are combined by forming their product and their quotient.[3]

Both primitive and derived features are treated analogously in deciding whether two classes are contrasted by a feature, since derived features are legitimate Boolean, nominal or numeric features.

It will be observed that contrasts by a nominal or numeric feature may (but will not necessarily) introduce a slight degree of disjunctiveness, which is to a somewhat greater extent the case in contrasts accomplished by derived features.

Missing values do not present much problem, since they can be ignored without any need to estimate a value nor to discard the remaining informative features values of the instance. In the case of nominal features, missing values can be treated as just another legitimate feature value.

2.2 The simplicity criteria

MPD uses three intuitive criteria to guarantee the uncovering of the most parsimonious discrimination among classes:

1. *Minimize overall features.* A set of classes may be demarcated using a number of overall feature sets of different cardinality; this criterion chooses those overall feature sets which have the smallest cardinality (i.e. are the shortest).
2. *Minimize profiles.* Given some overall feature set, one class may be demarcated—using only features from this set—by a number of profiles of different cardinality; this criterion chooses those profiles having the smallest cardinality.
3. *Maximize coordination.* This criterion maximizes the coherence between class profiles in one discrimination model,[4] in the case when alternative profiles remain even after the application of the two previous simplicity criteria.[5]

Due to space limitations, we cannot enter into the implementation details of these global optimization criteria, in fact the most expensive mechanism of MPD. Suffice it to say here that they are implemented in a uniform way (in all three cases by converting a logic formula – either CNF or something more complicated – into a DNF formula), and all can use both sound and unsound (but good) heuristics to deal successfully with the potentially explosive combinatorics inherent in the conversion to DNF.

2.3 An illustration

By way of (a simplified) illustration, let us consider the learning of the Bulgarian translational equivalents of the English verb *feed* on the basis of the case frames of the latter. Assume the following features/values, corresponding to the verbal slots: (1) NP1={hum,beast,phys-obj}, (2) VTR (binary feature denoting whether the verb is transitive or not), (3) NP2 (same values as NP1), (4) PP (binary feature expressing the obligatory presence of a prepositional phrase). An illustrative input to MPD is given in Table 1 (the sentences in the third column of the table are not a part of the input, and are only given for the sake of clarity, though, of course, would normally serve to deriving the instances by parsing).

The output of the program is given in Table 2. MPD needs to find 10 pairwise contrasts between the 5 classes (i.e. *N-choose-2*, calculable by the formula N(N-1)/2), and it has successfully discriminated all

[2] Besides these atomic feature values we may also support (hierarchically) structured values, but this will be of no concern here.

[3] Analogously to the Bacon program's invention of theoretical terms Langley et. al., 1987.

[4] In a "discrimination model" each class is described with a *unique* profile.

[5] By way of an abstract example, denote features by F1...Fn, and let Class 1 have the profiles: (1) F1 F2, (2) F1 F3, and Class 2: (1) F4 F2, (2) F4 F5, (3) F4 F6. Combining freely all alternative profiles with one another, we should get 6 discrimination models. However, in Class 1 we have a choice between [F2 F3] (F1 *must* be used), and in Class 2 between [F2 F5 F6] (F4 must be used); this criterion, quite analogously to the previous two, will minimize this choice, selecting F2 in both cases, and hence yield the unique model Class 1: F1 F2, and Class 2: F4 F2.

Classes	Instances	Illustrations
1.otglezdam	1. NP1=hum VTR NP2=beast ¬PP	1.He feeds pigs
	2. NP1=hum VTR NP2=beast¬PP	2.Jane feeds cattle
2.xranja	1. NP1=hum VTR NP2=hum¬PP	1.Nurses feed invalids
	2. NP1=beast VTR NP2=beast ¬PP	2.Wild animals feed their cubs regularly
3.xranja-se	1. NP1=beast ¬VTR PP	1.Horses feed on grass
	2. NP1=beast ¬VTR PP	2.Cows feed on hay
4.zaxranvam	1. NP1=hum VTR NP2=phys-obj PP	1.Farmers feed corn to fowls
	2. NP1=hum VTR NP2=phys-obj PP	2.This family feeds meat to their dog
5.podavam	1. NP1=phys-obj VTR NP2=phys-obj PP	1.The production line feeds cloth in the machine
	2. NP1=phys-obj VTR NP2=phys-obj PP	2.The trace feeds paper to the printer
	3. NP1=hum VTR NP2=phys-obj PP	3.Jim feeds coal to a furnace

Table 1: Classes and Instances

Classes	Profiles
1.otglezdam	¬PP NP1xNP2=([hum beast])
2.xranja	¬PP NP1xNP2=([hum hum] ∨ [beast beast])
3.xranja-se	NP1=beast PP
4.zaxranvam	NP1=hum PP
5.podavam	66.6% NP1=phys-obj PP

Table 2: Classes and their Profiles

classes. This is done by the overall feature set {NP1, PP, NP1xNP2}, whose first two features are primitive, and the third is a derived nominal feature. Not all classes are absolutely discriminated: Class 4 (*zaxranvam*) and Class 5 (*podavam*) are only partially contrasted by the feature NP1. Thus, Class 5 is 66.6% NP1=phys-obj since we need to retract 1/3 of its instances (particularly, sentence (3) from Table 1 whose NP1=hum) in order to get a clean contrast by that feature. Class 1 (*otglezdam*) and Class 2 (*xranja*) use in their profiles the derived nominal feature NP1xNP2; they actually contrast because all instances of Class 1 have the value 'hum' for NP1 and the value 'beast' for NP2, and hence the "derived value" [hum beast], whereas *neither* of the instances of Class 2 has an identical derived value (indeed, referring to Table 1, the first instance of Class 2 has NP1xNP2=[hum hum] and the second instance NP1xNP2=[beast beast]). The resulting profiles in Table 2 is the *simplest* in the sense that there are no more concise overall feature sets that discriminate the classes, and the profiles—using only features from the overall feature set—are the shortest.

3 Componential analysis

3.1 In lexicology

One of the tasks we addressed with MPD is semantic componential analysis, which has well-known linguistic implications, e.g., for (machine) translation (for a familiar early reference, cf. Nida, 1971). More specifically, we were concerned with the componential analysis of kinship terminologies, a common area of study within this trend. KINSHIP is a specialized computer program, having as input the *kinterms* (=classes) of a language, and their attendant *kintypes* (=instances).[6] It computes the feature values of the kintypes, and then feeds the result to the MPD component to make the discrimination between the kinterms of the language. Currently, KINSHIP uses about 30 features, of all types: binary (e.g., male={+/-}), nominal (e.g., lineal={lineal, co-lineal, ablineal}), and numeric (e.g., generation={1,2,..,n}).

In the long history of this area of study, practitioners of the art have come up with explicit requirements as regards the adequacy of analysis: (1) *Parsimony*, including both overall features and kinterm descriptions (=profiles). (2) *Conjunctiveness* of kinterm descriptions. (3) *Comprehensiveness* in displaying all alternative componential models.

As seen, these requirements fit nicely with most of the capabilities of MPD. This is not accidental, since, historically, we started our investigations by automating the important discovery task of componential analysis, and then, realizing the generic nature of the discrimination subtask, isolated this part of the program, which was later extended with the mechanisms for derived features and partial contrasts.

Some of the results of KINSHIP are worth summarizing. The program has so far been applied to more than 20 languages of different language families. In some cases, the datasets were partial (only consanguineal, or blood) kin systems, but in others they were complete systems comprising 40-50 classes with several hundreds of instances. The program has re-discovered some classical analyses (of the Amerindian language Seneca by Lounsbury), has successfully analyzed previously unanalyzed languages (e.g., Bulgarian), and has improved on previous analyses of English. For English, the most parsimonious model has been found, and the *only* one giving conjunctive class profiles for all kinterms, which sounds impressive considering the massive efforts concentrated on analyzing the English kinship

[6] Examples of English kinterms are *father*, *uncle*, and of their respective kintypes are: Fa (father); FaBr (father's brother) MoBr (mother's brother) FaFaSo (father's father's son) and a dozen of others.

system.[7]

Most importantly, MPD has shown that the huge number of potential componential (=discrimination) models—a menace to the very foundations of the approach, which has made some linguists propose alternative analytic tools— are in fact reduced to (nearly) unique analyses by our 3 simplicity criteria. Our 3rd criterion, ensuring the coordination between equally simple alternative profiles, and with no precedence in the linguistic literature, proved essential in the pruning of solutions (details of KINSHIP are reported in Pericliev and Valdés-Pérez, 1997; Pericliev and Valdés-Pérez, forthcoming).

3.2 In phonology

Componential analysis in phonology amounts to finding the distinctive features of a phonemic system, differentiating any phoneme from all the rest. The adequacy requirements are the same as in the above subsection, and indeed they have been borrowed in lexicology (and morphology for that matter) from phonological work which chronologically preceded the former. We applied MPD to the Russian phonemic system, the data coming from a paper by Cherry et. al., 1953, who also explicitly state as one of their goals the finding of minimal phoneme descriptions.

The data consisted of 42 Russian phonemes, i.e. the transfer of feature values from instances (=allophones) to their respective classes (=phonemes) has been previously performed. The phonemes were described in terms of the following 11 binary features: (1) vocalic, (2) consonantal, (3) compact, (4) diffuse, (5) grave, (6) nasal, (7) continuant, (8) voiced, (9) sharp, (10) strident, (11) stressed. MPD confirmed that the 11 primitive overall features are indeed needed, but it found 11 simpler phoneme profiles than those proposed in this classic article (cf. Table 3). Thus, the average phoneme profile turns out to comprise 6.14, rather than 6.5, components as suggested by Cherry et. al.

The capability of MPD to treat not just binary, but also non-binary (nominal) features, it should be noted, makes it applicable to datasets of a newer trend in phonology which are not limited to using binary features, and instead exploit multivalued symbolic features as legitimate phonological building blocks.

4 Language typology

We have used MPD for discovery of linguistic typologies, where the classes to be contrasted are individual languages or groups of languages (language families).

Classes	1	2	3	4	5	6	7	8	9	10	11
k	−	+	+		+		−	−	−		
k	−	+	+		+		−	−	+		
g	−	+	+		+		−	+	−		
g		+	+		+		−	+	+		
x	−	+	+		+		+				
c	−	+	+		−		−				
š	−	+	+		−		+	−			
ž	−	+	+		−		+	+			
t	−	+	−		−	−	−	−	−	−	
t	−	+	−		−	−	−	−	+	−	
d	−	+	−		−		−	+	−		
d		+	−		−		−	+	+		
s	−	+	−		−	−	+	−	−		
s	−	+	−		−	−	+	−	+		
z	−	+	−		−		+	+	−		
z	−	+	−		−		+	+	+		
ŝ	−	+	−		−	−	−			+	
n	−	+	−		−	+			−		
n	−	+	−		−	+			+		
p	−	+	−		+	−	−	−	−		
p	−	+	−		+	−	−	−	+		
b	−	+	−		+		−	+	−		
b		+	−		+		−	+	+		
f	−	+	−		+		+	−	−		
f	−	+	−		+		+	−	+		
v	−	+	−		+		+	+	−		
v	−	+	−		+		+	+	+		
m	−	+	−		+	+			−		
m	−	+	−		+	+			+		
‘u	+	−	−	+	+						+
u	+	−	−	+	+						−
‘o	+	−	−	−	+						
‘e	+	−	−	−	−						
‘i	+	−	−	+	−						+
i	+	−	−	+	−						−
‘a	+	−	+								+
a	+	−	+								−
r	+	+					−		−		
r	+	+					−		+		
l	+	+					+		−		
l	+	+					+		+		
j	−	−									

Table 3: Russian phonemes and their profiles

In one application, MPD was run on the dataset from the seminal paper by Greenberg (1966) on word order universals. This corpus has previously been used to uncover linguistic *universals*, or similarities; we now show its feasibility for the second fundamental typological task of expressing the *differences* between languages. The data consist of a sample of 30 languages with a wide genetic and areal coverage. The 30 classes to be differentiated are described in terms of 15 features, 4 of which are nominal, and the remaining 11 binary. Running MPD on this dataset showed that from 435 (30-Choose-2) pairwise discriminations to be made, just 12 turned out to be impossible, viz. the pairs:

(berber,zapotec), (berber,welsh)
(berber,hebrew), (fulani,swahili)
(greek,serbian), (greek,maya)
(hebrew,zapotec), (japanese,turkish)
(japanese,kannada), (kannada,turkish)
(malay,yoruba), (maya,serbian)

The contrasts (uniquely) were made with a minimal set of 8 features: {SubjVerbObj-order, Adj < N, Genitive < N, Demonstrative < N, Numeral < N, Aux < V, Adv < Adj, affixation}.

In the processed dataset, for a number of languages there were missing values, esp. for features

[7] We also found errors in analyses performed by linguists, which is understandable for a computationally complex task like this.

(12) through (14). The linguistic reasons for this were two-fold: (i) lack of reliable information; or (ii) non-applicability of the feature for a specific language (e.g., many languages lack particles for expressing yes-no questions, i.e. feature (12)). The above results reflect our default treatment of missing values as making no contribution to the contrast of language pairs. Following the other alternative path, and allowing 'missing' as a distinct value, will result in the successful discrimination of most language pairs. Greek and Serbian would remain indiscriminable, which is no surprise given their areal and genetic affinity.

5 Speech production in aphasics

This application concerns the discrimination of different forms of aphasia on the basis of their language behaviour.[8]

We addressed the profiling of aphasic patients, using the CAP dataset from the CHILDES database (MacWhinney, 1995), containing (among others) 22 English subjects; 5 are control and the others suffer from anomia (3 patients), Broca's disorder (6), Wernicke's disorder (5), and nonfluents (3). The patients are grouped into classes according to their fit to a prototype used by neurologists and speech pathologists. The patients' records—verbal responses to pictorial stimuli—are transcribed in the CHILDES database and are coded with linguistic errors from an available set that pertains to phonology, morphology, syntax and semantics.

As a first step in our study, we attempted to profile the classes using just the errors as they were coded in the transcripts, which consisted of a set of 26 binary features, based on the occurrence or non-occurrence of an error (feature) in the transcript of each patient. We ran MPD with primitive features and absolute contrasts and found that from a total of 10 pairwise contrasts to be made between 5 classes, 7 were impossible, and only 3 possible. We then used derived features and absolute contrasts, but still one pair (Broca's and Wernicke's patients) remained uncontrasted. We obtained 80 simplest models with 5 features (two primitive and three derived) discriminating the four remaining classes.

We found this profiling unsatisfactory from a domain point of view for several reasons[9] which led us to re-examining the transcripts (amounting roughly to 80 pages of written text) and adding manually some new features that could eventually result in more intelligible profiling. These included:

Classes	Profiles
Control Subjects	average errors=[0, 1.3]
Anomic Subjects	average errors=[1.7, 4.6] prolixity=[7, 7.5] fluency
Broca's Subjects	¬fluency 87% ¬semi-intelligible
Wernicke's Subjects	prolixity=[12, 30.1] fluency
Nonfluent Subjects	¬fluency semi-intelligible

Table 4: Profiles of Aphasic Patients with Absolute Features and Partial Contrasts

(1) *Prolixity.* This feature is intended to simulate an aspect of the Grice's maxim of manner, viz. "Avoid unnecessary prolixity". We try to model it by computing the average number of words pronounced per individual pictorial stimulus, so each patient is assigned a number (at present, each word-like speech segment is taken into account). Wernicke's patients seem most prolix, in general.

(2) *Truthfulness.* This feature attempts to simulate Grices' Maxim of Quality: "Be truthful. Do not say that for which you lack adequate evidence". Wernicke's patients are most persistent in violating this maxim by fabricating things not seen in the pictorial stimuli. All other patients seem to conform to the maxim, except the nonfluents whose speech is difficult to characterize either way (so this feature is considered irrelevant for contrasting).

(3) *Fluency.* By this we mean general fluency, normal intonation contour, absence of many and long pauses, etc. The Broca's and non-fluent patients have negative value for this feature, in contrast to all others.

(4) *Average number of errors.* This is the second numerical feature, besides prolixity. It counts the average number of errors per individual stimulus (picture). Included are all coder's markings in the patient's text, some explicitly marked as errors, others being pauses, retracings, etc.

Re-running MPD with absolute primitive features on the new data, now having more than 30 features, resulted in 9 successful demarcations out of 10. Two sets of primitive features were used to this end: {average errors, fluency, prolixity} and {average errors, fluency, truthfulness}. The Broca's patients and the nonfluent ones, which still resisted discrimination, could be successfully handled with nine alternative derived Boolean features, formed from different combinations of the coded errors (a handful of which are also plausible). We also ran MPD with primitive features and partial contrasts (cf. Table 4). Retracting one of the six Broca's subjects allows all

[8] We are grateful to Prof. Brian MacWhinney from the Psychology Dpt. of CMU for helpful discussions on this application of MPD.

[9] First, one pair remained uncontrasted. Second, only 3 pairwise contrasts were made with absolute primitive features, which are as a rule most intuitively acceptable as regards the comprehensibility of the demarcations (in this specific case they correspond to "standard" errors, priorly and independently identified from the task under consideration). And, third, some of the derived features necessary for the profiling lacked the necessary plausibility for domain scientists.

classes to be completely discriminated.

These results may be considered satisfactory from the point of view of aphasiology. First of all, now all disorders are successfully discriminated, most cleanly, and this is done with the primitive features, which, furthermore, make good sense to domain specialists: control subjects are singled out by the least number of mistakes they make, Wernicke's patients are contrasted from anomic ones by their greater prolixity, anomics contrast Broca's and nonfluent patients by their fluent speech, etc.

6 *MPD* in the context of diverse application types

A learning program can profitably be viewed along two dimensions: (1) according to whether the output of the program is addressed to a human or serves as input to another program; and (2) according to whether the program is used for prediction of future instances or not. This yields four alternatives:

type (i) (+human/-prediction),
type (ii) (+human/+prediction),
type (iii) (-human/+prediction), and
type (iv) (-human/-prediction).

We may now summarize MPD's mechanisms in the context of the diverse application types. These observations will clear up some of the discussion in the previous sections, and may also serve as guidelines in further specific applications of the program.

Componential analysis falls under type (i): a componential model is addressed to a linguist/anthropologist, and there is no prediction of unseen instances, since *all* instances (e.g., kintypes in kinship analysis) are as a rule available at the outset.[10]

The aphasics discrimination task can be classed as type (ii): the discrimination model aims to make sense to a speech pathologist, but it should also have good predictive power in assigning future patients to the proper class of disorder.

Learning translational equivalents from verbal case frames belongs to type (iii) since the output of the learner will normally be fed to other subroutines and this output model should make good predictions as to word selection in the target language, encountering future sentences in the source language.

We did not discuss here a case of type (iv), so we just mention an example. Given a grammar G, the learner should find "look-aheads", specifying which of the rules of G should be fired first.[11] In this task, the output of the learner can be automatically incorporated as an additional rule in G (an hence be of no direct human use), and it should make no predictions since it applies to the *specific* G, and not to any other grammar.

For tasks of types (i) and (ii), a typical scenario of using MPD would be:

> Using all 3 simplicity criteria, and finding all alternative models, follow the feature/contrast hierarchy: primitive features & absolute contrasts > derived & absolute > primitive & partial > derived & partial

which reflects the desiderata of conciseness, comprehensiveness, and intelligibility (as far as the latter is concerned, the primitive features (normally user-supplied) are preferable to the computer-invented, possibly disjunctive, derived features).

However, in some specific tasks, another hierarchy seems preferable, which the user is free to follow. E.g., in kinship under type (i), the inability of MPD to completely discriminate the kinterms may very well be due to *noise* in the instances, a situation by no means infrequent, esp. in data for "exotic" languages. In a type (ii) task, an analogous situation may hold (e.g., a patient may be erroneously classed under some impairment), all this leading to trying first the primitive & partial heuristic. There may be other reasons to change the order of heuristics in the hierarchy as well.

We see no clear difference between types (i)-(ii) tasks, placing the emphasis in (ii) on the human addressee subtask rather than on prediction subtask, because it is not unreasonable to suppose that a concise and intelligible model has good chances of reasonably high predictive power.[12]

We have less experience in applying MPD on tasks of types (iii) and (iv) and would therefore refrain from suggesting typical scenarios for these types. We offer instead some observations on the role of MPD's mechanisms in the context of such tasks, showing at some places their different meaning/implication in comparison with the previous two tasks:

(1) Parsimony, conceived as a minimality of class profiles, is essential in that it generally contributes to reducing the cost of assigning an incoming instance to a class. (In contrast to tasks of types (i)-(ii), the Maximize-Coordination criterion has no clear meaning here, and the Minimize-Features may well be

[10] We note that componential analysis in phonology can alternatively be viewed of type (iii) if its ultimate goal is speech recognition.

[11] A trivial example is G, having rules: (i) s1→np, vp, ['.'] ; (ii) s2→vp, ['!'] ; (iii) s3→aux, np, v, ['?'], where the classes are the LHS, the instances are the RHS, and the profiling should decide which of the 3 rules to use having as input say *Come here!*.

[12] By way of a (non-linguistic) illustration, we have turned the MPD profiles into classification rules and have carried out an initial experiment on the LED-24 dataset from the UC Irvine repository. MPD classified 1000 unseen instances at 73 per cent, using *five* features, which compares well with a *seven* features classifier reported in the literature, as well as with other citations in the repository entry.

sacrificed in order to get shorter profiles).[13]

(2) Conjunctiveness is of less importance here than in tasks of type (i)-(ii), but a better legibility of profiles is in any case preferable. The derived features mechanism can be essential in achieving intuitive contrasts, as in verbal case frame learning, where the interaction between features nicely fits the task of learning "slot dependencies" (Li and Abe, 1996).

(3) All alternative profiles of equal simplicity are not always a necessity as in tasks of type (i)-(ii), but are most essential in many tasks where there are *different* costs of finding the feature values of unseen instances (e.g., computing a syntactic feature, generally, would be much less expensive than computing say a pragmatic one).

The important point to emphasize here is that MPD generally leaves these mechanisms as program parameters to be set by the user, and thus, by changing its inductive bias, it may be tailored to the specific needs that arise within the 4 types of tasks.

7 Conclusion

The basic contributions of this paper are: (1) to introduce a novel flexible multi-class learning program, MPD, that emphasizes the conciseness and intelligibility of the class descriptions; (2) to show some uses of MPD in diverse linguistic fields, at the same time indicating some prospective modes of using the program in the different application types; and (3) to describe substantial results that employed the program.

A basic limitation of MPD is of course its inability to handle *inherently disjunctive* concepts, and there are indeed various tasks of this sort. Also, despite its efficient implementation, the user may sometimes be forced to sacrifice conciseness (e.g., choose two primitive features instead of just one derived that can validly replace them) in order to evade combinatorial problems. Nevertheless in our experience with linguistic (and not only linguistic) tasks MPD has proved a successful tool for solving significant practical problems. As far as our ongoing research is concerned, we basically are focussing on finding novel application areas.

Acknowledgments. This work was supported by a grant #IRI-9421656 from the (USA) National Science Foundation and by the NSF Division of International Programs.

References

C. Cherry, M. Halle, and R, Jakobson. 1953. Toward the logical description of languages in their phonemic aspects. *Language* 29:34-47.

W. Daelemans, P. Berck, and S. Gillis. 1996. Unsupervised discovery of phonological categories through supervised learning of morphological rules. *COLING96*, Copenhagen, pages 95-100.

J. Bruner, J. Goodnow, and G. Austin. 1956. *A Study of Thinking.* John Wiley, New York.

J. Greenberg. 1966. Some universals of grammar with particular reference to the order of meaningful elements. In J. Greenberg, ed. *Universals of Language*, MIT Press, Cambridge, Mass.

C. Hempel. 1965. *Aspects of Scientific Explanation.* The Free Press, New York.

P. Langley, H. Simon, G. Bradshaw, and J, Zytkow. 1987. *Scientific Discovery: Computational Explorations of the Creative Process.* The MIT Press, Cambridge, Mass.

Hang Li and Naoki Abe. 1996. Learning dependencies between case frame slots. *COLING96*, Copenhagen, pages 10-15.

B. MacWhinney. 1995. *The CHILDES Project: Tools for Analyzing Talk.* Lawrence Erlbaum, N.J.

E. Nida. 1971. Semantic components in translation theory. In G. Perren and J. Trim (eds.) *Applications of Linguistics*, pages 341-348. Cambridge University Press, Cambridge, England.

V. Pericliev and R. E. Valdés-Pérez. 1997. A discovery system for componential analysis of kinship terminologies. In B. Caron (ed.) *16th International Congress of Linguists*, Paris, July 1997, Elsevier.

V. Pericliev and R. E. Valdés-Pérez. forthcoming. Automatic componential analysis of kinship semantics with a proposed structural solution to the problem of multiple models. *Anthropological Linguistics.*

J. R. Quinlan. 1986. Induction of decision trees. *Machine Learning*, 1:81-106.

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

H.Tanaka. 1996. Decision tree learning algorithm with structured attributes: Application to verbal case frame acquisition. *COLING96*, Copenhagen, pages 943-948.

R. E. Valdés-Pérez and V. Pericliev. 1997. Maximally parsimonious discrimination: a task from linguistic discovery. *AAAI97*, Providence, RI, pages 515-520.

R. E. Valdés-Pérez and V. Pericliev. 1998. Concise, intelligible, and approximate profiling of numerous classes. Submitted for publication.

[13] E.g., instead of the profile [xranja-se: NP1=beast PP] in Table 2, one may choose the valid shorter profile [xranja-se: ¬VTR], even though that would increase the number of overall features used.