Maximally Parsimonious Discrimination: A Generic Task from Linguistic Discovery

Raúl E. Valdés-Pérez
Computer Science Department
Carnegie Mellon University
Pittsburgh, PA 15213

Vladimir Pericliev
Department of Mathematical Linguistics
Institute of Mathematics and Informatics, bl. 8
Bulgarian Acad. of Sci., 1113 Sofia - Bulgaria

Abstract

Data-driven model building is an important task of scientific discovery that is seeing real success in the development and application of discovery programs. Most efforts have targeted fields of natural science in which the hypothesis spaces are specialized and deal with domains having considerable formal structure. Less work has been directed toward qualitative areas of social science, in which model building also arises. This paper reports the first automation of a modelling task from linguistic anthropology: the analysis of natural-language kinship terminologies in terms of simpler semantic components. Our approach uses three generic simplicity criteria to comprehensively find all the simplest models that are consistent with kinship data. We have reproduced results from the linguistics literature, but have also found simpler models in some cases. The task has strong generic elements: extracts of the code are applied to other data sets to illustrate this potential.

Introduction

Building scientific models from data is an area of scientific reasoning that has been much studied in computational scientific discovery. Most of the model-building discovery programs have addressed theory-driven tasks in which the task hypothesis spaces have a special character: the algorithms for generating models tend to be domain dependent, as well as the heuristics for guiding the search in the space of models. Moreover, most of the work has focused on the natural sciences of biology, chemistry, and physics, although Tetrad, for example, finds application to numeric observational data in social science and medicine.

We have identified a novel model-building task from linguistics that is largely data-driven, in the sense that the problem space dealt with is strongly generic. Moreover, the task resembles traditional inductive concept learning, except that *induction* is not the main task goal; rather, like in most computerized model building, the task is to comprehensively find all the models that are maximally parsimonious, i.e., that can account for the input data while respecting prior knowledge, and that are formally simplest.

We describe the first automation of this linguistics task, carried out by means of a rational analysis of the discovery task's problem space (Valdes-Perez 1995). We present the main algorithms, and discuss some of the findings that we have made in linguistics. Then, we highlight the generic character of the inference task - which we call maximally parsimonious discrimination (MPD) - and run MPD on data sets from the Irvine repository for further illustration. Finally, the procedures and heuristics of MPD are contrasted with those of inductive (multiclass) concept learning: MPD optimizes parsimony in a top-down manner, whereas concept learning typically follows a greedy bottom-up algorithm that has no explicit concern with a sparse, coordinated use of features.

The Linguistics Task

Every known society has a terminology to express kinship (extended family) relations, although not every society uses the same system: different languages classify kin (relatives) together under one linguistic label, or *kinship term*. For example, in English the term *father* denotes a male biological parent, whereas in Seneca (an American Indian language of the Iroquois family), the term *ha?nih* denotes the male parent, but also what English speakers would call *uncle*, also the father's male first cousin, and others (Leech 1974). Thus, different meaning structures underly different kinship terminologies, hence the interest of anthropological linguistics in discovering a concise formal description of these meaning structures possessed by the diverse languages throughout the world.

Kin relations specify the genealogical position of a kin to the speaker, and are expressed in the language-neutral terms of the primary relationships: F=*father*, M=*mother*, B=*brother*, S=*sister*, s=*son*, d=*daughter*, H=*husband*, W=*wife*. These primary relationships are concatenated to express more distant relationships. Some examples are: FB (*father's brother*), MB (*mother's brother*), FSH (*father's sister's husband*), and MSd (*mother's sister's daughter*). According to

1.	*great-grandfather*	FFF, FMF, MFF, MMF	2.	*great-grandmother*	FFM, FMM, MFM, MMM
3.	*great-uncle*	MMB, MFB, FMB, FFB	4.	*great-aunt*	MMS, MFS, FMS, FFS
5.	*grandfather*	FF, MF	6.	*grandmother*	FM, MM
7.	*uncle*	MB, FB, FFs, FMs, MFs, MMs, FSH, FSH, MSH, MSH, FFdH, FMdH, MMdH, MFdH			
8.	*aunt*	MS, FS, FFd, FMd, MFd, MMd, FBW, FBW, MBW, MBW, FFsW, FMsW, MMsW, MFsW			
9.	*father*	F	10.	*mother*	M
11.	*son*	s	12.	*daughter*	d
13.	*brother*	B, Fs, Ms	14.	*sister*	S, Fd, Md
15.	*cousin*	MSd, MBd, FSd, FBd, MSs, MBs, FSs, FBs, MMSdd, MMSsd, MMBdd, MMBsd, MFSsd, MFBdd, etc.			
16.	*nephew*	Bs, Ss, Fss, Mss, Fds, Mds, WSs, WBs, HSs, HBs, WMds, WFss, WFds, WMss, HMds, HFss, HFds, HMss			
17.	*niece*	Bd, Sd, Fsd, Msd, Fdd, Mdd, WSd, WBd, HSd, HBd, WMdd, WFsd, WFdd, WMsd, HMdd, HFsd, HFdd, etc.			
18.	*grandson*	ss, ds	19.	*granddaughter*	dd, sd
20.	*great-grandson*	sss, sds, dss, dds	21.	*great-granddaughter*	ddd, ssd, sdd, dsd
22.	*husband*	H	23.	*wife*	W
24.	*father-in-law*	WF, HF, WMH, HMH	25.	*mother-in-law*	WM, HM, WFW, HFW
26.	*son-in-law*	dH, WdH, HdH	27.	*daughter-in-law*	sW, WsW, HsW
28.	*brother-in-law*	HB, WB, WMs, WFs, HMs, HFs, WMHs, WFWs, HMHs, HFWs, SH, FdH, MdH, MHdH, FWdH			
29.	*sister-in-law*	HS, WS, WFd, WMd, HFd, HMd, WMHd, WFWd, HMHd, HFWd, BW, FsW, MsW, MHsW, FWsW			
30.	*step-father*	MH	31.	*step-mother*	FW
32.	*step-son*	Hs, Ws	33.	*step-daughter*	Hd, Wd
34.	*step-brother*	MHs, FWs	35.	*step-sister*	MHd, FWd

Table 1: Data on English Kinship Terminology

linguistics convention, *ego* is the relative from whom the relationship is traced (i.e., the speaker), and *alter* is the person to whom the kinship term refers. Kinship terms normally cover a set of several kin, e.g., the English kinship term *cousin* covers the kin-types MSd MBd FSd FBd MMSdd MMSsd and many others. The set of all kinship terms in a language forms the kinship vocabulary of this language. In the science of linguistics, this vocabulary constitutes a *semantic (or lexical) field*.

The task of the linguist is to determine the relevant semantic features that can contrast the meaning of any of the kinship terms within the semantic field from any other kinship term. The disjunctive listing of kinship examples (e.g., consider the many ways in English to be a *cousin*) gets translated into a conjunctive rule that covers all the input examples. In general, the task may involve inventing new features in order to better handle the problems posed by a given language, in addition to finding the conjunctive rules, which together are called a <u>componential model</u> in linguistics. For over two decades componential analysis has been a valuable tool in (anthropological) linguistics research (Goodenough 1967; Leech 1974).

The criteria for the quality of this type of semantic modelling are its consistency, parsimony, and comprehensiveness. That is, all kinship terms (heretofore, <u>kinterms</u> for short) should be mutually discriminated (if possible), few features should be used and each kinterm description should be as succinct as possible, and the full set of alternative simplest models should be considered.

Method

Table 1 lists extensive data on English kinship first compiled by (Lounsbury 1964). Our KINSHIP program works directly on such input, but to illustrate our approach, we will analyze a simplified system of only five kinterms: *son-in-law, daughter-in-law, sister, sister-in-law, and father.*

kinterm	consanguinality	generation	sex
son-in-law		-1	m
daughter-in-law		-1	f
sister	consanguineal	0	
sister-in-law	affinal	0	
father		1	

Table 2: A Componential Model of Five Kinship Terms

A componential analysis might conclude that the simplest model of the semantics of these kinterms is as shown in Table 2. Only the three features *consanguinality* (blood relations are called *consanguineal*, and relations by marriage are *affinal*), *generation*, and *sex*[1] are needed. For example, a sufficient model for *sister-in-law* is *affinal* and *generation=0* meaning that the kinship relation arises by marriage of the speaker (*ego*) and is considered to be of the same generation. Note that it is not necessary to specify that *sister-in-law* is female, because the stated two-feature description is enough to contrast *sister-in-law* from the remaining kinterms.

The right way to interpret a consistent componential model like that in Table 2 is that no other kinterm possesses the conjunction of feature values that describe that kinterm. Thus, no other kinterm in the simplified problem being considered has *generation=1*, which is the entire description for *father*.

Let us now consider how to arrive at the componential model in Table 2 starting from the raw data in Table 1 on the five kinterms and their individual instances. The first step is to compute the feature values of all the instances. Currently our KINSHIP program

[1] This feature is more properly called *sex-of-alter*, because in some languages the sex of *ego* (the speaker) helps determine the right term to express a kinship relation between *ego* and *alter*; the American Indian language Seneca is such a language.

computes 20 features, all of specific relevance to genealogy and kinship, some taken from the anthropological and linguistics literature.

The second step is to express the feature values for each kinterm as a function of the feature values of the instances. The rules for this function are as follows. A kinterm is assigned

- a single feature value if the feature is Boolean and all the instances possess that value, otherwise N/A.
- a minimal set of feature values if the feature is nominal and the value of every instance appears in the set, otherwise N/A.
- a range if the feature is numeric and all the instances' values lie within the range (which is chosen to be the smallest possible).

The latter two rules, involving a set of nominal values and a numeric range, allow a degree of disjunctiveness in formulating the contrast between two kinterms.

Parsimony. At this point the instance feature values have been transferred to their parent kinterms. The next step is to set up a logical formula that expresses the constraint of pairwise contrast: for every pair of kinterms, form a disjunction of all the features that serve to cleanly contrast the pair. In the case of a Boolean feature, a pair of kinterms is contrasted if their Boolean values are unequal; for numeric and nominal features, they are contrasted if their ranges and sets (respectively) do not overlap.[2] Then, the conjunction of all these pairwise disjunctions expresses (as a conjunctive normal form - CNF) the possible feature sets that can cleanly contrast all the kinterms. If a pair cannot be contrasted, then this is reported and the contribution of that kinterm pair to the CNF is skipped, otherwise the formula would collapse into unsatisfiability.

The previous system of five kinterms gives rise to the following CNF formula, which possesses 10 (*5 kinterms, choose 2*) conjuncts:

```
(and (or sex)
     (or sex distance generation consanguinality)
     (or sex generation)
     (or distance generation lineality ancestor
         consanguinality generation-of-last-link
         structural-equivalence)
     (or distance generation consanguinality)
     (or generation)
     (or sex distance generation lineality
         ancestor generation-of-last-link
         consanguinality structural-equivalence)
     (or distance consanguinality)
     (or sex generation lineality
         ancestor generation-of-last-link)
     (or sex distance generation lineality ancestor
         consanguinality structural-equivalence))
```

For example, the first conjunct expresses the single contrast (sex) between *son-in-law* and *daughter-in-law*; the second conjunct expresses four possible contrasts between *son-in-law* and *sister*, and so on. The subsequent conversion of this CNF to disjunctive normal form (DNF) yields simply:

```
(or (and distance generation sex)
    (and consanguinality generation sex))
```

which, of course, expresses the same meaning as the CNF formula, but in a different (DNF) form. The disjuncts of this DNF represent all the possible subsets of features that can explain the data. In this case, there are two possible subsets, both of which are equally simple (contain fewest features). We will choose the second subset to continue our example (both would be followed up by the program). This step illustrates the **first** of our three simplicity criteria: select a minimal feature set. All three simplicity criteria that we will introduce are applied by manipulating logical formulas.

Given the set of available features (following our example, *consanguinality*, *generation*, and *sex*), the next step is to express the possible descriptions for each individual kinterm. For a given kinterm, we compare its feature values with those of every other kinterm, and form a CNF that expresses the possible contrasts, much like before. Table 3 shows these CNF formulas for each individual kinterm.

Table 3 also shows the DNF equivalents of the CNF formulas; the corresponding formulas are exactly equivalent in logical meaning, but the DNF is in the form needed to apply our **second** simplicity criterion: choose the smallest expression for each kinterm, i.e., having the fewest features. For example, *father* possesses two alternative expressions, one of which (*generation*) is simpler. Hence, the first alternative for *father* is discarded. This criterion is reminiscent of preferring the most general conjunctive rule, except that here it is unclear that a shorter rule is *more general*, because the feature sets may not even intersect.

The descriptions for three of the kinterms are thus determined; the only remaining source of variability concerns the two terms *son-in-law* and *sister* in Table 3. Both of these terms possess two equally simple alternative descriptions, which implies a potential for $2 \times 2 = 4$ alternative full models. The problem of choosing among these alternatives motivates our **third** simplicity criterion, which seeks to impose a global coherence on the choices remaining after the first two criteria have been applied. We note that the two alternatives for *son-in-law* both involve the feature *sex*, so this feature must appear; the remaining choices for this kinterm are captured by the formula `(or (and generation) (and consanguinality))`. Similarly, the two choices for *sister* both involve *consanguinality*, so the remaining alternatives are captured by the formula `(or (and sex) (and generation))`. Conjoining these two choices and converting the result to DNF yields

```
(or (and generation) (and consanguinality sex))
```

from which the simplest is just *generation*. We now go back to the choice points for *son-in-law* and *sister*

[2] Of course, a *statistical* contrast could be introduced by allowing a certain fraction of overlap.

kinterm	CNF formula	DNF equivalent

son-in-law	`(and (or sex) (or consanguinality generation sex)`
	`(or generation sex)`
	`(or consanguinality generation))`
		`(or (and generation sex)`
		`(and consanguinality sex))`

daughter-in-law	`(and (or sex) (or consanguinality generation)`
	`(or generation)`
	`(or consanguinality generation sex))`
		`(or (and generation sex))`

sister	`(and (or consanguinality generation sex)`
	`(or consanguinality generation)`
	`(or consanguinality) (or generation sex))`
		`(or (and consanguinality sex)`
		`(and consanguinality generation))`

sister-in-law	`(and (or generation sex) (or generation)`
	`(or consanguinality)`
	`(or consanguinality generation sex))`
		`(or (and consanguinality generation))`

father	`(and (or consanguinality generation)`
	`(or consanguinality generation sex)`
	`(or generation sex))`
	`(or consanguinality generation sex))`
		`(or (and consanguinality sex)`
		`(and generation))`

Table 3: CNF Formulas and DNF Equivalents for Individual Kinship Terms

and choose the description [*generation, sex*] for *son-in-law* and [*consanguinality, generation*] for *sister*. Thus, the four potential descriptions of the two kinterms are reduced to a single one by minimizing the variability in these choices. This is accomplished by setting up not a CNF expression as in the first two simplicity criteria, but rather a *conjunction of DNF's*, and this nested expression is converted to a simple DNF to carry out our third and final simplicity criterion. The final result is the simplest componential model shown in Table 2; specific feature values are included for each kinterm.

We can now summarize the effects of our three simplicity criteria. The first serves to minimize the set S of features that are used. Given a set S, the second criterion serves to minimize the kinterm descriptions by using the fewest features from S. There may still remain alternatives, so the third criterion strives for a global coherence among these alternatives. Even after applying these criteria, there is no guarantee that the result is a unique most-parsimonious model. Domain-dependent criteria could be introduced, or all the alternative simplest models can simply be reported to the user.

Lack of space precludes a more formal treatment of the algorithm and its computational complexity; this is left for future publications.

Linguistics Results

We have applied MPD (in the form of the specialized KINSHIP program) to substantially complete data sets on English (35 kinterms) and Bulgarian (40 terms), and to smaller data sets from the linguistics literature such as Hindi (30 terms), Polish (27 terms), Seneca (17 terms - consanguineal only), and Swedish (23 terms). In every case, the simplest componential models contain between 4 and 7 features out of the 20 that are currently available in the program; of course the specific features vary across languages. For some languages, a unique simplest model exists (e.g., English) whereas for others, there are multiple models of equal simplicity. This should not be hastily interpreted as an inherent characteristic of a language, because the possibility of a unique simplest model depends on the available features and on their specific definitions (there is some variance of feature definitions in linguistic practice).

A companion paper (Pericliev & Valdes-Perez 1997) for a linguistics audience describes our linguistics contributions, which we summarize here. KINSHIP has reproduced Lounsbury's classic analysis of Seneca (Leech 1974), successfully analyzed a previously unanalyzed language (Bulgarian), and improved on previous analyses of English. The program discovered both the most parsimonious model of English, and the *only one* respecting the requirement that all kinterms have conjunctive definitions (half a dozen known analyses have either tacitly ignored or artificially circumscribed this issue). Also, our task formalization has handled aspects that were previously only implicit (e.g., our third simplicity criterion is based on an intuition that was never explicitly formulated in the literature). Finally, we have shown mistakes in analyses performed by human analysts, further illustrating the need for computerized tools in data-driven model building.

We have begun making contact with linguistic anthropologists with the intent of analyzing the many languages, whether living or extinct, whose kinship systems have been collected but never successfully analyzed. We believe that this carries a potential for a boom in knowledge of the interrelationships between the world's thousands of languages.

The Generic Task

The original motivation for this work was to provide a first automation of componential analysis, specifically of kinship terminologies, which has been an important discovery task in linguistics and anthropology. Subsequently, it became clear to us that most of the computations, as well as the task itself, were rather generic. The linguistics-dependent code consisted mainly of the calculation of feature vectors from the raw data on kinterm members, so we separated these procedures from the generic part. Our name for the linguistics program is KINSHIP, whereas we are calling the generic part MPD - *maximally parsimonious discrimination.* Further discussion will now use the generic term *class* where we earlier used the linguistic *kinterm.*

MPD can be applied to data on class instances whose feature vectors have been determined independently. The first step then becomes transferring the feature values of the class instances into the feature values of their class. The remaining steps are the same as in the kinship case: apply the three parsimony criteria to find all the simplest models consistent with the data.

To further develop MPD as a generic tool for multi-class discrimination, we have begun adding capabilities for inventing derived features and for allowing statistical rather than absolute contrasts (e.g., numeric ranges can overlap to a specified degree).

Currently, derived features are invented using two heuristics. The first heuristic is inspired by the Bacon program (Langley *et al.* 1987): for every pair of numeric features, create two derived features that consist of the product and the quotient of the feature pair. The second heuristic is: for every pair of Boolean features, invent various logical combinations which currently consist of *and, nand, iff, or, xor,* and *implies.* These derived features are useful in applying MPD to data sets from the Irvine repository below.

Manipulation of logical expressions

The manipulation of logical expressions is carried out by Lisp functions that convert CNF or more general expressions into DNF. This CNF grows quadratically with the number of the classes, so its conversion to DNF is potentially the most expensive step in the whole process. Hence, the code incorporates two heuristics for simplifying a CNF expression before converting it to DNF. The more important heuristic checks for subsumption relations between two terms of a CNF: when the literals in a term are a subset of the literals in another term, the latter term can be dropped. For example, the expression `(and (or A B C) (or B) (or C D))` yields `(and (or B) (or C D))`, since `(or A B C)` is subsumed by the singleton `(or B)`. In practice, this heuristic is very often applicable, and drastically simplifies the subsequent exponential DNF conversion, rendering it feasible on data sets having dozens of classes. For example, on the English data of Table 1, whose CNF consists of 595 conjuncts, the heuristic speeds up DNF conversion from 41 seconds to 0.1 seconds on a standard workstation. On Bulgarian kinship having 778 conjuncts, the speedup is from 716 seconds to 0.2, and similarly for other languages.

Class	Description
1	milk
2	feathers
3	¬milk, ¬feathers, backbone, ¬fins, ¬aquatic-and-breathes
4	¬milk, fins
5	¬milk, ¬feathers, *backbone, ¬fins,* aquatic-and-breathes
6	¬backbone
7	¬backbone

Table 4: Class Contrasts in Zoo Data Set

It is important to understand the *meaning* of these logical expressions: they are not standard propositions, but rather imperatives. For example, `(or A B)` means that the feature `A` or the feature `B` *must be used.* Thus, it would be incorrect to simplify these logical expressions by, for example, reducing `(or A B)` to just `(or A)` if the proposition `B` $\Rightarrow$ `A` is true given the meanings of `A` and `B`. As a second example, under such imperative semantics, the expression `(or positive negative zero)` would not be a tautology, because it asserts that *one of those three features must be used.*

Application to other data sets

We have applied the MPD algorithm to two data sets (Zoo and audiologyS) from the Irvine Repository of machine learning databases. These data sets were selected because they contained multiple classes and had some relation to science, which is where we expect that the task of maximally parsimonious discrimination will find most scope.

Zoo. This is an artificial data set containing 7 classes, 101 instances, and 16 features, all Boolean except for the number of legs (numeric). To give an idea of its contents: the first instances (alphabetically) for each class are aardvark, chicken, pitviper, bass, frog, flea, and clam.

MPD with no feature invention reports that classes 3,5 and 6,7 cannot be cleanly contrasted. MPD with feature invention over the Boolean variables finds that classes 6,7 still cannot be contrasted cleanly. Table 4 shows the maximally parsimonious model, accepting that classes 6,7 remain uncontrasted. The italicized features disappear from the description of class 5 when the program augments the feature set with the derived (and underlined) feature *aquatic-and-breathes.*

AudiologyS. This data set was originally collected by Professor Jergen at Baylor College of Medicine. It consists of 24 classes, 200 instances, and 69 features, all Boolean or symbolic ordinal. MPD without feature invention reports that 17 of the 276 pairwise classes cannot be contrasted cleanly. The simplest models (too large to be shown here) contain 11 features, and around 2.75 features are used to describe each class. MPD with feature invention does not succeed in reducing significantly the number of uncontrasted classes.

Linguistic Phonology. We have recently applied the MPD algorithm to data in phonology (the study of language sounds) from a classic paper that reported a componential analysis of Russian phonemes (Cherry, Halle, & Jakobson 1953). The data set consists of 42 classes (phonemes) and 11 binary features (voiced, etc.); the class individuals are not available from the paper. KINSHIP confirmed that all 11 features are needed, but our second and third parsimony criteria led to simpler solutions than were reported.

Comparison to inductive classification

The difference between the task addressed by this paper and the traditional machine learning task of inductive concept learning can be illustrated with the following analogy. Suppose one has a database of paintings by various masters, each of whom has a consistent style (if their styles have varied, then one forms classes such as "early Picasso"), together with features calculated on their paintings such that each painting is converted to a feature vector. One reasoning task is: given a newly discovered painting, determine who painted it (classify the painting). This is inductive classification of multiclass concepts, which can be approached in several ways (Dietterich & Bakiri 1995).

A second reasoning task is: given the same database, express the *differences* among these painters in the most parsimonious ways possible. The goal here is to *understand* in simple ways how the style of these painters differ; the goal is not prediction of heretofore unseen paintings. Typically, one would like to find *all* equally-parsimonious solutions. The contribution of this paper corresponds to this second task, not to the task of classifying new unseen data.

One could use the rules implicit in a classifier (e.g., decision tree) to express these differences, but these algorithms are developed to predict unseen instances, not to contrast among seen concept classes. Such algorithms have no explicit concern with minimizing the use of discriminating features nor minimizing the description of individual classes. There is no correspondence with our three simplicity criteria.

There are two main approaches to learning multiclass decision trees (Dietterich & Bakiri 1995). The first approach is to induce a separate classifier for each class. The AQ11 program (Dietterich *et al.* 1982, section XIV/D4) followed this approach: to learn a given class, all the data lying in that class are positive instances, and all data external to the class are classified as negative instances. This scheme sacrifices overall coordination among the classes (e.g., minimizing the features that can be used), but this seems rarely to matter for the task of inductive classification.

The second approach to learning multiclass decision trees is to induce a single classifier where each class is viewed as a separate value of a single concept. Here also, there is no guarantee of using a minimal set of features. Moreover, the description of individual classes may contain irrelevant features: even when conjunctive concepts are feasible, multiclass decision trees may possess several paths from the root to a concept class. An example from the linguistics data is illustrative: the English kinterm *cousin* does not have gender, but most English kinterms do depend on the sex of the kin. Hence, one expects that *sex* will be an early branch point, and that paths to *cousin* will make reference to this irrelevant feature.

Summarizing, one expects that by explicitly optimizing (top-down) for parsimonious discrimination among multiple classes, as is done here, the solutions will turn out differently. Bottom-up decision-tree learners follow greedy approaches that do not seek a global optimum.

A final characteristic of our task is that one typically seeks to comprehensively find *all* equally parsimonious solutions. This is not a goal that arises in traditional concept learning. By itself, this aspect suffices to distinguish our task from the standard approaches to inductive classification.

Acknowledgments. This work was supported in part by a grant #IRI-9421656 from the (USA) National Science Foundation and by the Center for Light Microscope Imaging and Biotechnology at Carnegie Mellon.

References

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

Dietterich, T. G., and Bakiri, C. 1995. Solving multiclass learning problems via error-correcting output codes. *Journal of Artificial Intelligence Research* 2:263–286.

Dietterich, T.; London, B.; Clarkson, K.; and Dromey, G. 1982. Learning and inductive inference. In Cohen, P. R., and Feigenbaum, E. A., eds., *The Handbook of Artificial Intelligence*, volume 3. William Kaufmann.

Goodenough, W. H. 1967. Componential analysis. *Science* 156:1203–1209.

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

Leech, G. N. 1974. *Semantics.* Harmondsworth, UK: Pelican.

Lounsbury, F. 1964. The structural analysis of kinship semantics. In Lunt, H., ed., *Proceedings of the 9th International Congress of Linguists.* The Hague: Mouton and Co. 1073–1090.

Pericliev, V., and Valdes-Perez, R. E. 1997. A discovery system for componential analysis of kinship terminologies. In *Proceedings of the 16th International Congress of Linguists.* Forthcoming.

Valdes-Perez, R. E. 1995. Some recent human/computer discoveries in science and what accounts for them. *AI Magazine* 16(3):37–44.