Int. J. Human-Computer Studies (2000) **53,** 411–436
doi:10.1006/ijhc.2000.0397
Available online at http://www.idealibrary.com on IDEAL®

Concise, intelligible, and approximate profiling of multiple classes

Raúl E. Valdés-Pérez and Francisco Pereira
Computer Science Department, Carnegie Mellon University, Pittsburgh, PA 15213, USA
Vladimir Pericliev
Department of Mathematical Linguistics, Institute of Mathematics and Informatics, bl. 8 Bulgarian Academy of Science, 1113 Sofia, Bulgaria

When a dataset involves multiple classes, there is often a need to express the key contrasting features among these classes in humanly understandable terms, that is, to profile the classes. Commonly, one class is contrasted from the rest by aggregating the latter into a pseudo-class; alternatively, classes are treated separately without coordinating their profiles with those of the other classes. We introduce the *concise all pairs profiling* (CAPP) method for concise, intelligible, and approximate profiling of large classifications. The method compares all classes pairwise and then minimizes the overall number of features needed to guarantee that each pair of classes is contrasted by at least one feature. Then each class profile gets its own minimized list of features, annotated with how these features contrast the class from the others. Significant applications to social and natural science are demonstrated.

© 2000 Academic Press

1. Introduction

A common discovery task in scientific and other professional activities is to express in a concise and humanly understandable way the key contrasting characteristics (or *profiles*) of a classification. When the classes are few, simple methods may be sufficient. When the number of classes is larger, say, 5, 10 or 50, then there is a greater need for simple and approximate profiles, in order to deal with the potentially overwhelming amount of information.

The aim of this article is to describe specialized methods for concise, intelligible, and approximate profiling of large classifications. The key idea is to explicitly compare *all pairs* of classes and then minimize the overall number of features needed to guarantee that each class can be contrasted from every other class by at least one feature. The worst-case asymptotic complexity of the resulting algorithms is poor, but in practice they work well due to the availability of sound problem-reduction methods.

The original motivation was to automate a task of linguistic discovery called componential analysis (Goodenough, 1967; Cherry, Halle & Jakobson, 1953) in which the classes are sharply distinct. However, the methods (called *concise all pairs profiling* or CAPP) become more broadly applicable by generalizing them to deal with overlapping or

1071-5819/00/090411 + 26 $35.00/0 © 2000 Academic Press

internally non-uniform classes. We demonstrate the methods on the original linguistics task and on current problems in psychology, chemistry and biology.

2. Motivation

Let us consider the simple three-way classification in Table 1 in which the table entries indicate how feature values are distributed within each class. We assume that the goal is to convey the key characteristics of class $C1$. A common approach is to aggregate $C2$ and $C3$ into a pseudo-class not-$C1$ and then apply a greedy method that successively finds a next best feature at each step. Thus, a greedy method that used conventional splitting criteria would choose $X3$ as the best initial discriminator, since $X3$ divides the 300 examples into the two sets $\{100\ C1, 50\ \text{not-}C1\}$ and $\{150\ \text{not-}C1\}$, which is a better initial discrimination than is available with $X1$ or $X2$. The method would continue by selecting $X1$ followed by $X2$ (or vice versa), finally arriving at the description $C1 = X3 \wedge X1 \wedge X2$ as shown in Table 1. This is fine if the description will be used as a *classifier* that predicts the class of new examples, because the description is accurate.

However, the description $C1 = X1 \wedge X2$ is more concise and thus is easier for people to understand. How can such a description be found? One way is not to use a greedy method but instead find a guaranteed minimal description, e.g. a guaranteed-shortest decision tree (Murphy & Pazzani, 1994), although this is practical only on small datasets. We propose a method that will find the most concise descriptions on larger, practical datasets. The method compares all classes pairwise and determines the features that can discriminate each pair of classes. Then, these discriminations are assembled into a final description that can discriminate each pair of classes with *at least one* feature.

Consider a second problem, this time with numeric features. Let us say there are N overlapping classes, whose values for one feature F take on a Gaussian distribution with identical standard deviations σ but whose means are spaced by $\sigma/2$. It seems worthwhile to know that the single feature F is enough to distinguish (approximately) each class, e.g. that the F values for class Ci tend to be smaller than for $Ci + 1$, $Ci + 2 \ldots$ but larger than for $Ci - 1$, $Ci - 2$, … (A concrete illustration is the amount of skin

TABLE 1

Two descriptions obtained by (1) *a greedy method based on aggregating classes and* (2) *a comparison of all pairs of classes*†

	Boolean features		
Class	$X1$	$X2$	$X3$
$C1$	100 yes, 0 no	100 yes, 0 no	100 yes, 0 no
$C2$	0 yes, 100 no	100 yes, 0 no	25 years 75 no
$C3$	100 yes, 0 no	0 yes, 100 no	25 years 75 no

† The table entries show the distribution of feature values for each class.
Greedy method ($C1$ vs $\neg C1$): $C1 = X3 \wedge X1 \wedge X2$
Most concise: $C1 = X1 \wedge X2$.

pigmentation in countries (= classes) along a line from say, UK to Sudan.) A discovery method for concise profiling of the classes should be capable of detecting this simple result, in which the simplest available global profile makes use of only a single feature. (Of course, such simple models are infrequently available, but it is important to detect them when possible.) To find this global profile, the method would need to coordinate the selected descriptions for each class, e.g. by minimizing the total features that are used. The methods we propose are able to find such globally simplest profiles.

We conclude from these two examples that there are valuable concise differences among multiple classes that may be missed by aggregating classes into pseudo-classes (example 1) or by failing to coordinate the individual descriptions over all the classes (example 2). Our proposed method is meant to detect these concise differences among classes.

2.1. PROBLEM STATEMENT

Our approach tests all $N(N-1)/2$ pairs of N classes. As shown by the example in Table 1, the advantage of considering all pairs of classes is that easy differences among the classes can be revealed, which would otherwise be obscured by aggregating classes into pseudo-classes. More formally, the problem statement is

> *Given* N classes, at least one example of each class, and symbolic or numeric features that describe the examples, *find* a most concise profile (i.e. a list of features) for each class C, such that any other class is contrasted from C by at least one feature in the profile.

By "most concise profile" is meant the fewest features needed overall to profile all the classes. In general, classes can be overlapping, so we say that classes $C1$ and $C2$ are contrasted by a feature F if the fraction of overlapping feature values is below some maximum amount or ceiling.

One reason for minimizing feature sets is to ensure that the selected features possess a general ability to express the pairwise contrasts in the data, i.e. have the most descriptive power. This is broadly similar to the goal in learning classification rules of pruning or of minimizing feature sets (e.g. Almuallim & Dietterich, 1994) in order to improve generalization accuracy and avoid overfitting. Here, however, the emphasis is on concise approximate description, not inductive prediction.

2.2. EXAMPLE OF A MULTI-CLASS PROFILE

Before developing the algorithm in detail, we show the output of CAPP on a public dataset from the UCI Repository (Blake, Keogh & Merz). The Dermatology dataset involves six classes, 366 examples, and 34 features (all numeric except 1 nominal); the data were contributed in January 1998 by H. Altay Guvenir of Bilkent University.

Table 2 shows one of the simplest profiles (expressed qualitatively for brevity) which involve only five out of the 34 features. For example, the profiles reveal that—in this dataset—**pityriasis rubra pilaris** tends to afflict children; its victims tend to be younger than those of the other dermatology classes. In other words, members of the class **pityriasis rubra pilaris** have values for the *age* feature that are lower than those of the members of any other class. In this case, one feature is enough to contrast the class with

TABLE 2
Qualitative profiles of six dermatology classes

Class	Profile (features are in bold)
Pityriasis rubra Pilaris	**Age** (range = [7,22], $\mu = 10.25$, $\sigma = 3.62$, $N = 20$) less than for all the other classes
Pityriasis rosea	**Band-like-infiltrate** = 0.0, $N = 49$ less than for lichen plaus **Koebner-phenomenon**(range = [0,3], $\mu = 1.18$, $\sigma = 0.80$, $N = 49$) more than for pityriasis rubra pilaris, cronic dermatitis, and seboreic dermatis **Thinning-of-the-suprapapillary-epidermis** = 0.0, $N = 49$ less than for psoriasis
Cronic dermatitis	**Fibrosis-of-the-papillary-dermis** (range = [1,3], $\mu = 2.29$, $\sigma = 0.72$, $N = 52$) more than for all the other classes
Lichen plaus	**Band-like-infiltrate**(range = [2,3], $\mu = 2.72$, $\sigma = 0.45$, $N = 72$) more than for all the other classes
Psoriasis	**Thinning-of-the-suprapapillary-epidermis**(range = [0,3], $\mu = 2.05$, $\sigma = 0.75$, $N = 112$) more than for all the other classes
Seboreic dermatitis	**Band-like-infiltrate**(range = [0,2], $\mu = 0.03$, $\sigma = 0.25$, $N = 61$) less than for lichen plaus **Age**(range = [10,70], $\mu = 35.47$, $\sigma = 13.47$, $N = 60$) more than for pityriasis rubra pilaris **Koebner-phenomenon**(range = [0,2], $\mu = 0.03$, $\sigma = 0.25$, $N = 61$) less than for pityriasis rosea **Fibrosis-of-the-papillary-dermis** = 0.0, $N = 61$ less than for chronic dermatitis **Thinning-of-the-suprapapillary-epidermis**(range = [0,1], $\mu = 0.02$, $\sigma = 0.13$, $N = 61$) less than for psoriasis

all other classes: the same holds for the three classes **cronic dermatitis**, **lichen plaus**, and **psoriasis**. At the other extreme, one feature is used for each pairwise contrast in the profile for **seborreic dermatitis** (a scaling rash that sometimes itches; known as dandruff when it occurs on the scalp). This and other experiments with these data suggest that **seborreic dermatitis** does not have a small number of sharply characteristic features.

There exist alternative simplest profiles because the features turn out to be quite discriminating, at least in the sense that many pairs of classes can be contrasted by several different features. Although not shown here, the contrasts are quite sharp, i.e. the overlaps among the feature values used in the profiles is small (maximum of 21%). The information about the feature values' range mean μ, standard deviation σ and number of data points N is shown for convenience; these statistics have no direct role in determining the profiles. Finally, these class profiles should not be viewed as logic propositions (e.g.

neither conjunctions nor disjunctions) but as annotations that include at least one feature for each pair of classes.

2.3. DIFFERENCE BETWEEN PROFILES AND RULES

In the limiting case when all classes possess necessary and sufficient conjunctive conditions, CAPP's profiles will be the most concise conjunctive descriptions for each class. In this case, the features that appear in a profile (e.g. featureA = value1 for symbolic features) will be the smallest set that can characterize members of the class, and no members of other classes will possess those feature values. Hence, in this limiting case, profiles can be re-interpreted as conjunctive rules.

However, we think that profiles should not be equated to (predictive) rules for these reasons:

1. When the classes overlap (i.e. necessary and sufficient conjunctive conditions are not available) and when classes *Ci* and *Cj* are contrastable for reasons other than for classes *Ci* and *Ck*, then rules can be poor predictors but the profiles can be adequate descriptions (an example is given shortly).
2. The profiles are never treated as rules in the CAPP procedure. That is, propositions of the form and inferential direction $P1 \wedge \cdots \wedge Pi \Rightarrow$ Class k are *never* generated nor tested against the data.
3. Numeric features appear in profiles thus: *Ci* tends to have smaller (or larger) values of feature *F* than *Cj* and *Ck*, with an overlap of (say) 21%. All of the dermatology class profiles from Table 2 are of this form. No rule immediately follows from such profiles, and such information does not immediately follow from rules.

To illustrate point 1 from the previous list, let us consider an extreme case involving N classes. Consider a class $C1$ that has the Boolean value True 50% of the time for all features. Assume that any other class *Cj* never has the value True for feature *Fj*, but is 50% True for the other features. (For concreteness: $C1$ could be the general population, $C2$ the Girl Scouts, $C3$ is a club for above-average height, $C4$ a club for above-average wealth, etc., and the features are sex, tall/short, rich/poor, etc.) Then the only available profile for $C1$ is that it is 50% True for the features *Fj*, $j = 2, \ldots, N$. (i.e. more boys than the Girl Scouts, shorter people, more indigents, etc.) Thus, $C1$'*s* profile contrasts $C1$ from any other class by at least one feature. Given the highly overlapping classes, this profile seems adequate to convey the salient contrasting characteristics of $C1$.

Suppose that we now turn this profile into the corresponding rule: $F2 \wedge F3 \wedge \cdots FN \Rightarrow C1$. How good would this rule be? If the features are uncorrelated, then the probability that any single example of $C1$ will satisfy the rule is $(1/2)^{N-1}$, which goes quickly to zero as the N classes become numerous. Thus, profiles can be poor rules, but even so they are not worthless, because of their roles as simple, understandable, and approximate descriptions of the salient contrasting features.

In the appendix, profiles and $C4.5$ rules (Quinlan, 1993) are compared in detail on a 10-class numeric dataset taken from images of protein expression in cells. The comparison reveals significant differences between profiles and rules in their representational forms, and also in their goals: rules emphasize finding coherent subclasses that

enable reliable prediction, whereas profiles emphasize finding approximate descriptions of all class members.

3. Detailed description of the CAPP method

The next subsections first define the notion of partial contrast, and then present the *concise all pairs profiling* procedure interleaved with an illustration of its operation.

3.1. ABSOLUTE AND PARTIAL CONTRASTS

We say that two classes $C1$ and $C2$ are *absolutely* contrasted by a numeric feature F if their ranges do not overlap, e.g. if the smallest feature value of any member of $C1$ is greater than the largest value of any member of $C2$. We do not consider cases where the values for $C2$ lie within a "hole" of $C1$'s values, partly because of the added complexity and partly because we seek simple statements like *for feature F, C2's values tend to be greater than C1's values*. Two classes are absolutely contrasted by a Boolean or nominal feature when the classes have no shared values.

Our implementation of *partial* contrasts treats numeric and symbolic features analogously by testing whether two classes can be contrasted *absolutely* after removing up to some percentage of the overlapping values. The overlap value is normalized to between 0 and 1, so that removing all values from a class corresponds to an overlap of 1. Thus, the overlap between a class and itself is 1.

Let us consider partial contrasts that use Boolean features. Say that the eight examples of a class $C1$ are 7 True and 1 False, and the 10 examples of $C2$ are 3 True and 7 False. Their (smallest) overlap is obtained by removing the 1 False value for $C1$ and the 3 True values for $C2$, giving an overlap measure of $1/8 + 3/10 = 0.425$ or 42%. Nominal features are handled similarly.

Partial contrasts for *numeric* features are handled analogously: Given two sorted lists of numbers, what overlap needs to be removed so that the smallest number in one list is larger than the largest number in the second list? (Both directions should be tried.) When A and B are of unequal lengths, the cost of removing an entry from either list is the reciprocal of the list's length, which is the number of data points. As an example of numerical overlaps, two otherwise identical Gaussian distributions that are shifted by half their standard deviation overlap by roughly 80%.

Thus, the overlaps among numeric and symbolic feature values are handled analogously, without coercing one data type into another. In both cases, missing or not-applicable feature values can be ignored. In our applications so far, the intuitive notion of overlap has been understandable to non-specialist users of the method, which is important.†

The following description of the algorithms will assume a fixed, maximum-allowable overlap between feature values. However, the program can find the smallest overlap

†The overlap measure is similar to a simple estimate of the expected misclassification cost (Breiman, Friedman, Olshen & Stone, 1984, pp. 94–99), which the cited authors found inadequate for their task of choosing the best among alternative tree splits. Here, however, two overlap values are never compared, because overlaps are only tested for being within the threshold. In any case, a measure such as the GINI index (Breiman et al., 1984) could replace our overlap measure without changing the overall approach.

ceiling that still contrasts all class pairs by doing a binary search between the extremes of 0 and 80% (we consider anything above 80% as excessive).‡

3.2. ALGORITHM

A class profile should guarantee that every other class is contrasted by at least one feature. The CAPP method minimizes the total number of features needed to profile all the classes. Thus, it does not follow the most common approaches to feature selection (analyzed in Blum & Langley, 1997) which rely on a heuristic search that adds (*forward selection*) or removes (*backward elimination*) features, or some hybrid of these heuristic searches. The CAPP method does not carry out a heuristic search: it guarantees finding a minimum feature set.

The first step is straight-forward: for each class and feature, collect the feature values of the class examples. Numeric features give a sorted list, and symbolic features give a multiset (a set with counts for each member of the set). These feature values are then stored with the corresponding class. So, the information that is crucial when building decision trees or rules—knowing that an *example* has at the same time the value A for one feature and the value B for another feature—is discarded here.

The main algorithm consists of three stages A–C, each having several substeps. The presentation is interleaved with an indented illustration on a real, but abstracted, example. The overall procedure is admittedly somewhat complicated. The main idea of each stage is described in the first paragraph of each of the three subsections: **A. minimize overall features**; **B. minimize individual profiles**; and **C. maximize coordination**. The algorithmic details can be skipped on first reading.

A. *Minimize overall features.* This first stage finds a minimized feature set $\mathcal{F}$ that can guarantee contrasting all pairs of classes. This feature set will be used to build the profiles.

A1. For each pair of classes, form a disjunction of all features that can contrast the pair subject to the overlap ceiling.

For example, suppose that six classes ($C1$, $C2$, … , $C6$) can be contrasted by a 20% overlap ceiling, and that the class instances are described by the 21 features (A, B, … , U). Then the first two classes $C1$ and $C2$ might be contrasted by any of the features `A, B, C, D, E, F` but not by `G, H`… For example, the feature `A` can contrast $C1$ and $C2$ while respecting the 20% overlap ceiling in the values of that feature. This yields a disjunction (or `A B C D E F`) *which means* that feature `A` can contrast $C1$ and $C2$ or `B` can contrast the pair, etc. Similarly, classes $C1$ and $C3$ might be contrasted by any of the features `B, C, D, E, F, G, H` and so on.

A2. Above we noted how the pairs $C1$, $C2$, and $C1$, $C3$ can be constructed. We need also to contrast all class pairs, i.e. $C1$, $C4$ and all the others. Each pair yields a disjunction, and all pairs need to be contrasted, so all the $N(N-1)/2$ disjunctions are conjoined

‡A binary search will first try 40% overlap, then if all class pairs are contrasted by that overlap ceiling, a value of 20% is tried and so on, until the smallest ceiling that still enables contrasting all pairs is reached. A separate issue is that Jerome Pesenti pointed out that it is best to confirm that less than 50% of both class's values overlap, otherwise on tiny datasets or in highly degenerate cases it can be unclear which class tends to have smaller values. For example, the overlap between (1 5) and (2 3 4) is 50% (delete either the 1 or the 5), but no clear tendency is present.

(here $N = 6$ so there are 15 disjunctions). The result is a *conjunctive normal form* (CNF) formula§ which expresses the full range of possible feature sets that can contrast all class pairs. If a pair cannot be contrasted by any feature, then this fact is reported and the disjunction for that case is skipped, otherwise the formula would collapse into unsatisfiability.

With the 15 class pairs to be contrasted, we obtain the following CNF formula which contains 15 disjunctions:

```
(and (or A B C D E F)
     (or B C D E F G H)
     (or B C D E F I J K L M N O P)
     (or B D E F H Q R)
     (or B C D E F)
     (or A G H)
     (or J K L M N O P)
     (or C H Q R S T U)
     (or A)
     (or G H J K L M N O P)
     (or G Q R S)
     (or G U)
     (or H J K L M N O P Q R T)
     (or J K L M N O P)
     (or H Q R T U))
```

Thus, the first line `(or A B C D E F)` says that classes `1` and `2` can only be contrasted by one (or more) of these six features. The ninth line `(or A)` says that one of the class pairs can be contrasted *only* by the `A` feature and so on.

A3. Apply known problem-reduction methods, summarized in Section 3.3 below, to reduce the size of the CNF formula (size reduction is crucial because a later manipulation of this formula is exponential in its size). For example, there are three disjunctions that contain the `A` feature:

```
(or A B C D E F)
(or A G H)
(or A)
```

The last of these *requires* `A` to be chosen, so the first two disjunctions become superfluous, because they are subsumed by the third `(or A)` and hence are discarded. Similarly, the last of the three disjunctions

```
(or B C D E F G H)
(or B C D E F I J K L M N O P)
(or B C D E F)
```

subsumes the first two, i.e. satisfying the last disjunction will automatically satisfy the first two. Hence the latter are also discarded. Other term subsumptions also lead to deletions.

§According to Pankhurst (1983), this idea was first described by Kautz (1968) in the context of fault testing and diagnosis of digital circuits, but only for Boolean features (logic values).

Applying all the term-subsumption simplifications to the above CNF formula yields

```
(and (or B D E F H Q R)
     (or B C D E F)
     (or A)
     (or G Q R S)
     (or G U)
     (or J K L M N O P)
     (or H Q R T U))
```

This formula states constraints on the acceptable choices of features; these constraints are unchanged by the simplification.

A second problem-reduction method is available, which checks not for subsumption relations between *disjunctions* but between *features*. The basic idea is that *if* a feature X can satisfy a disjunction whenever a feature Y can, *then* Y can be deleted everywhere.

For example, each of the example features B, C, D, E could be eliminated because each is subsumed by the feature F; wherever each appears, F also appears. Similarly, the E feature subsumes B, C, D, F and could eliminate all four of these features. A different choice of features to eliminate, based on a different choice of subsuming feature, can lead to a different result, but any such choice will preserve a guarantee of finding at least one minimal feature set.

Doing three feature-subsumption reductions based on the choices of subsuming features F, P and R, followed by one term-subsumption reduction, yields

```
(and (or F) (or A) (or G R) (or G U) (or P) (or R U))
```

Thus, any choice of feature set that would, for example, contain the C feature has been eliminated. If *all* equally concise feature sets are needed, then this second reduction cannot be used.

This second reduction method changes the constraint inherent in the formula, because not all possible feature sets remain represented, but at least one smallest feature set does remain.

A4. Convert the simplified CNF formula to a *disjunctive normal form* (DNF) formula; this changes the form but not the logical meaning of the formula. Each disjunction of the resulting DNF formula expresses an alternative feature set. Choose the smallest feature set and call it $\mathscr{F}$.

Converting the last CNF formula gives this equivalent DNF formula:

```
(or (and A F G P R) (and A F G P U) (and A F P R U))
```

and we can arbitrarily set $\mathscr{F}$ to the last term's features {A F P R U}.

CNF → DNF conversion is asymptotically worst-case intractable, because the NP-complete MINIMUM COVER problem (Garey & Johnson, 1979, p. 222) can be reduced to it. However, the dozens of practical cases we have attempted are all easily handled by the problem-reduction methods, which simplify the CNF greatly before its conversion, as we see in this example

B. *Minimize individual profiles.* We are given a minimal feature set $\mathscr{F}$ as the output of stage A. Now we need to determine what features (or subset of $\mathscr{F}$) will actually be used to

profile each individual class *Ci*. For each *Ci*, we find its possible minimal profiles *using only* the features in $\mathscr{F}$, not the full original set. The following steps are done once for each class *Ci*.

B1. For every other class *Cj* ($i \neq j$), form a disjunction of all the features in $\mathscr{F}$ that can contrast *Ci* and *Cj* subject to the overlap ceiling. A class *Ci* will then obtain $N-1$ disjunctions, where N is the number of classes, and each disjunction will be small since only the features in $\mathscr{F}$ can be used.

Taking *Ci* as class 1, we compare it against the remaining five classes with respect to the minimized feature set {A F P R U}, giving these five disjunctions:

```
(or A F) (or F) (or F P) (or R) (or F)
```

B2. Continue with steps A.2–A.4, except that $N-1$ disjunctions are involved rather than the $N(N-1)/2$ disjunctions from the all-pairs case in stage A.

B3. Each disjunction of the resulting DNF formula determines an alternative profile for *Ci*. We keep only the shortest disjunctions.

The previous disjunctions are conjoined to yield

```
(and (or A F) (or F) (or F P) (or R) (or F))
```

whose conversion to DNF gives `(or (and F R))`. Thus, the simplest profile for class 1 given $\mathscr{F}$ uses the two features F and R. Similarly, the CNF formula for class 2 is

```
(and (or A F) (or A) (or P) (or F R U) (or A))
```

which simplifies to just `(and (or P) (or F R U) (or A))`, whose subsequent conversion to DNF gives

```
(or (and A F P)
    (and A P U)
    (and A P R))
```

Thus, there are three equally concise profiles for class 2, each involving three features. If any term in this DNF contained more than three features, it would be discarded as non-simplest.

C. *Maximize coordination.* As just seen for class 2, the output of stage B can be several alternative, equally-concise feature sets for each of the classes. One might just choose arbitrarily among these alternatives. We prefer to coordinate these choices, using the idea that if class *Ci* can make use of either of the features A or B and likewise for class *Cj*, then it is better for both to use A or both to use B, rather than have *Ci* use A and *Cj* use B. This coordination is accomplished by converting a complex Boolean formula to DNF, analogously to the previous criteria. This stage C is less important to understand than stages A and B.

C1. For every class *Ci* that has more than one minimal feature set as the output of stage B, we remove the *common core* of features that appear in all of these alternatives, and form a DNF formula that expresses the remaining choices of features for profiling *Ci*.

For brevity, consider a simple two-class problem involving *Ci* and *Cj*. Suppose class *Ci* has these two alternatives as the result of stage B:

```
(or (and A B) (and B C))
```

Since feature B appears in both and will therefore necessarily be chosen, the alternatives can be expressed as `(or (and A) (and C))`. Similarly, suppose class *Cj* has these two choices:

```
(or (and A C D E F) (and B C D E F))
```

After removing their common parts, the choices are `(or (and A) (and B))`.

C2. Conjoin all the DNF formulas from the last step, discarding any empty formulas.

Conjoining the two previous DNF formulas gives this nested expression

```
(and (or (and A) (and B))
     (or (and A) (and C))
```

C3. Convert the resulting nested logic formula to DNF and select the shortest disjuncts. Each of these corresponds to an alternative feature set that can be used to complete the individual class profiles. Let $\mathscr{G}$ designate one of these feature sets.

The conversion to DNF yields `(or (and A) (and B C))`, whose shorter feature set is simply $\mathscr{G} = \{A\}$.

C4. For every class *Ci*, remove any candidate feature set if it contains a feature that is absent from $\mathscr{G}$ and is not a member of the common core defined in step C1.

Ci's two choices were `(or (and A) (and C))`, so only the first is picked. *Cj*'s two choices were `(or (and A C D E F) (and B C D E F))` and again the first is picked.

C5. Finally, take the cross-product of all the remaining choices and form the individual profiles for every class; report all alternative complete profiles unless the user requests only one.

In the current example, the two classes remained with only one choice, so their individual profiles are unique. In the general case, if some remain with multiple choices, then the cross-product of the alternatives is carried out if the user wants all equally concise profiles, otherwise just one can be picked arbitrarily.

Finally, we arrive at a minimized list of features for each class, whose union is a minimized overall feature set. The last step is to annotate the list with how the classes are contrasted (qualitatively or quantitatively at the user's option), and with some statistics, as was shown in Table 2.

Summarizing, the CAPP procedure consists of three main stages: minimize overall features, minimize individual profiles and maximize coordination. All stages operate by formulating, simplifying, and transforming Boolean formulas.

3.3. PROBLEM-REDUCTION METHODS AND POSSIBLE HEURISTICS

The most expensive step in the whole approach is converting a possibly large CNF formula to a DNF formula at the **A. Minimize Overall Features** stage. Not only is the worst-case computational complexity of this step problematic (Garey & Johnson, 1979, p. 222), but the size of the CNF grows quadratically ($N(N-1)/2$) with the number of classes and also grows with the number of features.

However, a *term-subsumption* problem-reduction method usually simplifies greatly the size of the CNF formula. For example a formula `(and (or a) (or a b c))` can be correctly reduced to just `(and (or a))`. Thus, given two terms (disjunctions) $D1$ and $D2$, if $D1$'s disjuncts are a subset of $D2$'s disjuncts, then delete $D2$. Another problem-reduction method (*feature-subsumption*) can be applied if the user only wants a single profile rather than all equally concise profiles. In the CNF formula, suppose that whenever a feature $F1$ appears, the feature $F2$ also appears; this means that $F2$ can contrast at least all classes that $F1$ can. Hence, $F1$ can be deleted everywhere from the CNF without losing the possibility of finding *some* simplest profile.

On random CNF formulas, these two problem-reduction methods may not reduce the formula size at all. However, our experience has been that they work very well on real datasets because (1) two otherwise similar classes may differ in only a few features, so that this small set can subsume more numerous differences between other class pairs and (2) there are patterns in the relations between features, so that often one feature dominates another in their discriminative ability. This facet of real datasets is absent from random data.

If problem reduction cannot simplify the CNF formula enough to enable the problematic CNF $\rightarrow$ DNF conversion, a well-known greedy set-covering heuristic (Chvatal, 1979; Almuallim & Dietterich, 1994) is available: find the feature F that appears most often among the remaining sets to be covered and add F to the set of chosen features. In this case, a set corresponds to a disjunction in the CNF formula; the greedy set-covering heuristic dictates finding the feature that appears in (or satisfies) the most remaining disjunctions. The greedy heuristic can be used successively until the conversion to DNF becomes feasible, or it can be used by itself to build the feature set, dispensing entirely with any CNF-to-DNF conversion. In this case, non-minimal feature sets will typically be obtained.

4. Applications

The original goal of this work to automate from scratch a discovery task from the recent history of work at the intersection of anthropology and linguistics: the *componential analysis* of kinship semantics (Goodenough, 1967; Leech, 1974). Thus, the next section presents this task and our results in some detail. (For companion articles in linguistics that target this and other applications to linguistic discovery, refer to Pericliev & Valdes-Parez, 1998*a*,*b*). Current applications to psychology and chemistry are then discussed more briefly.

4.1. ANALYSIS OF KINSHIP TERMINOLOGIES IN LINGUISTICS

Every known society has a terminology to express kinship (extended family) relations, although not every society uses the same system: a language may classify kin (relatives) together under one linguistic label, or *kinship term*, that would make little sense to a speaker of a different language family. For example, in English the term *father* solely denotes a male biological parent, whereas in Seneca (a North 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). The world views expressed as kinship systems differ widely, hence the interest of anthropological linguistics in discovering a concise formal description of the kinship systems that underlie the thousands of the world's natural languages.

Kin relations specify the genealogical position of a kin to the speaker, and can be 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*). Kinship terms normally cover a set of several kin, e.g. the English kinship term *cousin* covers the examples MSd MBd FSd FBd MMSdd MMSsd and many others. In linguistics science, the set of all kinship terms in a language 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 profile that covers all the input examples. Here, conjunctivity is central, as was argued by Lounsbury 1965, p. 1074:

> We feel that we have failed if we cannot achieve conjunctive definitions for every terminological class in the system. Were we to compromise on this point and admit disjunctive definitions (class sums, alternative criteria for membership) as on a par with conjunctive definitions (class products, uniform criteria for membership), there would be no motivation for analysis in the first place, for definitions of kin classes by summing of discrete members [···] are disjunctive definitions par excellence.

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 kinship profiles, which together are called a *componential model* in linguistics. For over three decades, componential analysis has been a valuable tool in anthropological linguistics research (Goodenough, 1967; Leech, 1974).

The task is formulated as a profiling problem by equating kinship terms with classes, kinship attributes (e.g. sex of kin) with features, and the different ways to be a kin (e.g. a mother's brother's daughter = cousin) with class examples.

The criteria for the quality of this type of semantic modeling are its consistency, parsimony, and comprehensiveness. That is, all kinship terms should be mutually contrasted (if possible), few features should be used, each kinship term should be described as succinctly as possible, and the full set of alternative simplest models should be considered.

Our KINSHIP program contains the CAPP program in the sense that it starts with raw data on the kinship examples, computes features on these examples, and then applies CAPP to the result.

Most of the features have been taken from the linguistic and anthropological literature on kinship, although we have defined some features on our own which to our knowledge lack precedent in those literatures.

TABLE 3
Consanguineal kinship terms in Seneca

1.	ha?nih	F FB FMSs FFBs FMBs FFSs FFFBss
2.	no?yẽh	M MS MMSd MFBd MMBd MFSd MMMSdd
3.	hakhno?sẽh	MB MMSs MFBs MMBs MFSs MMMSds
4.	ake:hak	FS FMSd FFBd FMBd FFSd FFFBsd
5.	hahtsi?	Be MSse FBse MMSdse FFBsse MFBdse FMSsse MMBdse
6.	he?kẽ:?	By MSsy FBsy MMSdsy FFBssy MFBdsy FMSssy MMBdsy
7.	ahtsi?	Se MSde FBde MMSdde FFBsde MFBdde FMSsde MMBdde
8.	khe?kẽ:?	Sy MSdy FBdy MMSddy FFBsdy MFBddy FMSsdy MMBddy
9.	akyã:?sc:?	MBs FSs MMSss FFBds MFBss FMSds MMBss MBd FSd
10a.	he:awak (*for male speaker*)	ms mBs mMSss mFBss mMBss mFSss mMMSdss
10b.	he:awak (*for female speaker*)	fs fSs fMSds fFBds fMBds fFSds fMMSdds
11a.	khe:awak (*for male speaker*)	md mBd mMSsd mFBsd mMBsd mFSsd mMMSdsd
11b.	khe:awak (*for female speaker*)	fd fSd fMSdd fFBdd fMBdd fFSdd fMMSddd
12.	heyẽ:wõ:tẽ?	mSs mMSds mFBds mMBds mFSds mMMSdds
13.	hehsõ?nch	fBs fMSss fFBss fMBss fFSss fMMSdss
14.	kheyẽ:wõ:tẽ?	mSd mMSdd mFBdd mMBdd mFSdd mMMSddd
15.	khehsǒ?neh	fBd fMSsd fFBsd fMBsd fFSsd fMMSdsd

Table 3, taken directly from Leech (1974), shows kinship examples for various consanguineal (blood) relations in Seneca.¶ The componential analysis of Seneca kinship in Lounsbury (1964) was a prominent contribution (Leech, 1974) to the study of kinship as well as the the methodology of componential analysis.

The Seneca kinship examples are represented in a notation that leads from the speaker to the one spoken about (kin). Thus, the third example for the first kinship term **ha?nih** is FMSs, which means "speaker's father's mother's sister's son". In some cases, the relative age of the kin is a relevant property of the example: an "e" at the end of a term signifies that the kin is older than the speaker and a "y" means that the kin is younger. Also, in some cases the sex of the speaker matters, so that lower-case "m" (male) or "f" (female) begins the example description.

Table 4 shows KINSHIP's conclusions, which are the same as those of Lounsbury (1964) as retold by Leech (1974). (The English annotations, e.g., "my father" for **ha?nih** in the first entry, are not translations, but rather English renditions of the simplest example of each term.) For example, the kinship term **ha?nih** is described as being a male of the previous generation of and standing in a "parallel" relation to, the speaker ("parallel" is an attribute that was invented by kinship researchers): no other kinship term has these three feature values. Overall, five features are minimally sufficient to profile all the

¶Our MMMSdd term in line 2 of the table is incorrectly listed as MMSdd by Leech.

TABLE 4
Profiles of Seneca kinship terms

1.	ha?nih 'my father'	generation = 1	parallel	male	
2.	no? yẽh 'my mother'	generation = 1	parallel	female	
3.	hakhno?sẽh 'my uncle'	generation = 1	¬parallel	male	
4.	ake:hak 'my aunt'	generation = 1	¬parallel	female	
5.	hahtsi? 'my elder brother'	generation = 0	parallel	male	senior
6.	he?kẽ? 'my younger brother'	generation = 0	parallel	male	¬senior
7.	ahtsi? 'my elder sister'	generation = 0	parallel	female	senior
8.	khe?kẽ:?, 'my younger sister'	generation = 0	parallel	female	¬senior
9.	akyã:? se:? 'my cousin'	generation = 0	¬parallel		
10.	he:awak 'my son'	generation = −1	parallel	male	
11.	khe:awak 'my daughter'	generation = −1	parallel	female	
12.	heyẽ:wõ:tẽ? 'my nephew'	generation = −1	¬parallel	male	male-speaker
13.	hehsõ?neh 'my nephew'	generation = −1	¬parallel	male	female-speaker
14.	kheyẽ:wõtẽ? 'my niece'	generation = −1	¬parallel	female	male-speaker
15.	khehsõ?neh 'my niece'	generation = −1	¬parallel	male	female-speaker

kinship terms. In this case, since all features are Boolean or nominal and all contrasts are absolute, CAPP's profiles are identical to the desired conjunctive descriptions which are equivalent to classification rules.

The re-discovery of these profiles (i.e. componential-analytic model) validates the KINSHIP and CAPP programs in the sense that they reproduce a prominent scientific contribution starting from the same empirical data, which is a common yardstick in research on scientific discovery.††

Elsewhere (Pericliev & Valdés-Pérez, 1998*a*), we have shown that the kinship system of Yankee English admits simpler models than have been published. At the same time, we reported the first such analysis of Bulgarian. Both the English and Bulgarian data sets were substantially more complete and challenging than the Seneca data set shown here. KINSHIP has also been used to analyze a dozen or so kinship systems of other languages.

The task of profiling kinship relations involves dozens of kinship terms or classes, so that the number of pairs is quite large (e.g. *35 choose 2* = 595 in our English kinship data). However, the term-subsumption method of Section 3.3 works wonderfully on kinship because in many languages, many kinship terms will differ only by their value for **sex** (consider the English brother and sister, aunt and uncle, etc.). Hence, the disjunction `(or sex)` will appear in the CNF and thus enable deleting all other disjunctions containing the sex feature.

††One of Lounsbury's contributions was his invention of a distinctly new feature called *parallel*, a description of which is found in Leech (1974), which he carried out in a data-driven way (personal communication). Our KINSHIP program has no such capability for inventing new *primitive* features from the data, hence does not measure up to Lounsbury's achievement. Parenthetically, we have been able to invent a new feature *fractional-generation* that combines the expressiveness of the *generation* and *senior* features in a natural and general way, and thus enables even simpler profiles for the kinship classes.

4.2. OTHER APPLICATIONS

We describe two current collaborative applications of CAPP to chemistry and psychology. Unlike the original anthropological linguistics problem, these applications involve strictly numeric features. Also, these applications are instances of a general class of scientific application to which CAPP is very suited: relating classes based on structure (brain lesions, chemical elements) to behavioral features, or classes based on behavior to structural features. In all cases, the goal is to understand better, in the absence of an accurate theory that links the two, how structure relates to behavior.

4.2.1. Psychology

A significant application to psychology involved profiling children with different types of brain lesion with respect to their behavioral characteristics (MacWhinney *et al.*, 2000). There were six structural classes (five types of brain lesion and one control group), 170 examples (children, of whom 150 were in the control group), and 18 numeric behavioral features that measured how well the children did in verbal laboratory tests. CAPP found that three features were enough to profile all the classes at an overlap ceiling of 40%. The following qualitative, English-rendered excerpt consists of two profiles: one for the control group and another for the group of left-side focal lesions resulting from cerebral infarct:

> The control group (150 children) is better at visual naming than the minimal damage, hydrocephalus, left periventricular hemorrhage, and left cerebral infarct groups, and better at storing, elaborating, and following oral directions than the minimal damage, left periventricular hemorrhage, and right lesion groups.
>
> The left cerebral infarct group (7 children) is worse at storing, elaborating, and following oral directions than the right lesion and control groups, is better at word repetition than the hydrocephalus group, but worse at word repetition than the minimal damage and left periventricular hemorrhage groups.

The full CAPP profiles for brain lesions are reported elsewhere (MacWhinney *et al.*, 2000).

4.2.2. Chemistry

Our goal was to profile a set of eight metal catalysts in terms of their comparative ability to carry out types of chemical reactions, using data on 168 reactions and their energies (activation energy barriers) published earlier (Hu *et al.*, 1998). Thus, the eight classes correspond to metal catalysts (iron, copper, nickel, palladium, platinum, rhodium, iridium and ruthenium), an example corresponds to a specific chemical reaction and the several dozen numeric features (which we defined ourselves) are mainly of the form *energy of a reaction of a type X*, where the definition of a type refers to the bonds broken and formed as a reaction converts the reactants to the products. If a given reaction (out of the 168 in the dataset) is not of a type X, then the feature *energy of a reaction of a type X* has the value *not applicable*.

An example of a discovered profile at an overlap ceiling of 43% (Zeigarnik, Valdes-Perez & Pesenti, 2000), expressed qualitatively and in English for convenience, is the following for iron (Fe):

> Fe is worse than Cu, Ni, Pd, Pt, Rh, and Ir in its ability to carry out reactions of the type M-x-C–H-x-M, and worse than Cu, Pd, Pt, Rh, Ru and Ir in its ability to carry out reactions of the type M-x-C–O-x-M.

M-x-C–H-x-M is our notation for a reaction type that involves *breaking* a metal–carbon (M-x-C) and a metal–hydrogen (H-x-M) bond, and *forming* a carbon–hydrogen (C–H) bond. Unlike the applications to linguistics and psychology, in this case our collaborator felt that global minimization of features was less important, partly because one pair of metals was not sharply distinguished, hence the overlap ceiling would be set too high and the other metals would get profiles that were too crude, i.e. gave too much weight to conciseness over sharpness of contrast. Hence, the feature set for each of the metal classes was minimized separately. That is, stage A of Section 3.2 was skipped, and the individual minimizations in stage B made use of the entire original feature set, rather than the globally minimized feature set.

5. Discussion

5.1. EVALUATION

The goal of knowledge discovery is to improve human understanding of some domain. It is difficult to devise a quantifiable performance metric for this aim. As in other model-building tasks in scientific discovery that involve parsimony as a preference criterion (e.g. Langley, Simon, Bradshaw & Zytkow, 1987; Valdés-Pérez, 1994*a*,*b*; Valdés-Pérez & Zytkow, 1996), quality can be judged by the extent to which a method optimizes a metric that is conventional *or* arguably desirable for the task. Our profiles resemble models (e.g. in the kinship problem, the original practitioners certainly viewed their task as one of model building) and simplicity is conciseness of description.

Another way to legitimize work in knowledge discovery is to adduce creditable evidence that the procedures and/or conclusions are significant from the viewpoint of the application (e.g. Saitta & Nerri, 1998). Here, our evidence (see Section 4.1) consists of publications in the linguistics literature that report methods, re-discoveries, simpler models for known data and models for never-analyzed data (Pericliev & Valdés–Pérez, 1997; Pericliev & Valdés-Pérez, 1998). There are also two published CAPP-generated profiles in chemistry and psychology. Thus, CAPP's profiles have been judged interesting and understandable enough to merit publication by social and natural scientists.

Following Valdés–Pérez (1999), one can also evaluate CAPP by asking how "the design of the program, or the circumstances of its application, heighten the chances that its use will lead to knowledge that is novel, interesting, plausible and intelligible". CAPP's models tend to be (1) novel because the program comprehensively explores a combinatorial space that is dense with possibilities that are easy to overlook otherwise; (2) interesting because they are (maximally) concise; (3) plausible because feature minimization tends to counteract the curse of dimensionality; one can also employ permutation tests of significance (Good, 1994) when the data are sparse, as in the cited psychology data; and (4) intelligible because, by design, the program seeks short, unified profiles of each class and prefers rough approximations to the finer distinctions that are available only by adding layers of subclasses.

5.2. RELATED WORK

The PFOIL-CNF program (Mooney, 1995) was stated to be a quite natural representation for "nearly conjunctive" concepts, which makes it close to CAPP's profiles, and thus a specific comparison is warranted. PFOIL-CNF, which grows disjunctions using an information-gain heuristic, compared well in terms of classification accuracy with its predecessor (Quinlan, 1990) using several datasets from the UCI Repository (Blake *et al.*).

Although the goal of CAPP is to learn profiles, which are approximate descriptions and not classification rules, we will compare their respective outputs on the classic SOYBEAN data set (Michalski & Chilausky 1980), keeping in mind that their outputs differ in kind, so that a direct comparison, much less a quantitative one, is not easy. PFOIL-CNF is best suited for discrete-valued variables, thus the naturally numeric features in SOYBEAN were treated as nominal features, so we do likewise to ease the comparison.

Table 5 shows the PFOIL-CNF output reported in Mooney (1995) on the class Frog Eye Leaf Spot.‡‡ The second description in the table is CAPP's profile when absolute contrasts are required and only the single class Frog Eye Leaf Spot needs to be profiled. Interestingly, 10 of 14 classes can be contrasted cleanly with just the three features shown, all of which take on a single value for the target class. The third entry in the table is the case when the target class is required to contrast with *all* the other classes; an overlap ceiling of 31% is needed to accomplish this. In this case, three features are also found, one of which `(leafspot-size)` was seen in the previous profile. Arguably, the profiles are able to deliver more understandable descriptions of the target class, partly due to their conciseness and partly due to their ability to approximate the entire class, and not subclasses (disjunctions) within the target class.

Other related work is the CN2 induction algorithm (Clark & Niblett, 1989) which extends the basic AQ family of learning algorithms introduced by Michalski, Mozetic, Hong and Laurac (1986) in order to better handle inconsistent data. Again, CN2 learns classification rules whereas CAPP finds profiles, but we can compare their underlying algorithms. CN2 is a bottom-up, heuristic algorithm which uses information gain and likelihood ratios to guide the construction of ordered rules that are similar to decision lists (Rivest, 1987). CN2's treatment of multiclass datasets follows the usual practice: when describing one class, the examples from all the other $N - 1$ classes are aggregated and treated as negative examples.

The OPUS algorithm (Webb, 1995) for efficient and admissible unordered search was applied to machine learning and demonstrated on several UCI datasets. CAPP shares with OPUS the spirit of trying to find optimal solutions within a search space, which in the case of OPUS (applied to machine learning) consisted of pure conjunctive rules that maximize a Laplace accuracy estimate. However, OPUS does not seem to deal with numeric features. Also, its application to multi-class data treated each class separately, rather than globally, hence it is unclear to us whether it can find globally minimal feature sets.

‡‡It is fair to point out that this example was chosen by the author to illustrate specific difficulties with repetitive disjuncts.

TABLE 5
Comparative descriptions of Frog Eye Leaf Spot (SOYBEAN)

output of CNF learner [25]
fruit-spots = colored ∨ leafspot-size = > −1/8 ∧
external-decay = firm-and-dry ∨ leaf-shared = absent ∧
external-decay = firm-and-dry ∨ temp = norm ∨ stem-cankers = above-sec-nde ∧
fruit-pods = diseased ∨ seed-tmt = fungicide ∨ hail = no ∨ area-damaged = scattered ∧
stem-cankers = above-sec-nde ∨ plant-growth = norm ∧
stem-cankers = above-sec-nde ∨ seed = norm ∧
stem-cankers = above-sec-nde ∨ date = 8 ∨ date = 9 ∨ date = 10 ∨ hail = no ∧
stem-cankers = above-sec-nde ∨ germination = 80–89% ∨ date = 9 ∨ area-damaged = low-areas ∨ precip = norm ∧
stem-cankers = above-sec-ndc ∨ plant-stand = normal ∨ crop-hist = same-last-sev-yrs

CAPP profile (contrasts absolutely with 10 classes; the four uncontrasted classes are: alternarialeaf-spot,
phyllosticta-leaf-spot, brown-spot, and phytophthora-rot)

int-discolor (observed values: none [#cases = 91])
leafspot-size (observed values: > −1/8 [#cases = 91])
mold-growth (observed values: absent [#cases = 91])

CAPP profile (maximum 31% overlap; contrasts with all other 14 classes)

fruit-pods (observed values: diseased [#cases = 64], norm [#cases = 27])
leafspot-size (observed values: > −1/8 [#cases = 91])
roots (observed values: norm [#cases = 91])

The R-MINI program also explicitly tries to minimize rule lengths using well-developed heuristic methods from minimization of switching circuits (Hong, 1997), but this program is limited to categorical features and in the context of multiclass problems, it does not share CAPP's goal of comparing all pairs of classes rather than each class against the rest.

The methods underlying CAPP share the same spirit as in Holte (1993) who showed that very simple 1-level decision trees perform well on many of the common datasets used in machine learning research. The link is that CAPP relies on a single feature to contrast any pair of classes, and assembles these single features into a multiple-feature profile to describe a class. However, even if Holte's conclusions were not true, such simple methods seem necessary if one wants a global picture of numerous overlapping and/or noisy classes, because some accuracy often needs to be sacrificed to gain simplicity. This theme has been explored in the context of trading off decision tree complexity to gain simplicity by means of pruning (e.g. Iba, Wogulis & Langley, 1988; Bohanec & Bratko, 1994 and many others).

Exact and heuristic algorithms for finding minimal sets of Boolean features are described in Almuallin & Dietterich (1994); these correspond well to CAPP's stage A goal of finding minimal feature sets. The basic ideas are similar, except that they address concepts expressed as arbitrary Boolean formulas, whereas CAPP expresses concepts as profiles (feature lists), develops the ideas in the context of heterogeneous features

(both numeric and symbolic), and uses the minimal feature sets in stages B and C to find minimal individualized profiles for each class.

6. Conclusion

Classification problems can involve different goals. One goal is to induce an accurate automatic classifier of future examples. The second goal is to re-represent the significant relations in the data in a manner that is optimized more for human understanding and reporting and less for accurate prediction.

This article has introduced methods for uncovering the salient contrasting features in a large classification, i.e. five or so classes up to a hundred or two (the largest problem we have tackled). The CAPP approach finds concise contrasting profiles that guarantee that each class is contrasted from every other class by at least one feature. The main novelty is that all classes are compared pairwise, which can detect easy contrasts that would be obscured by aggregating competing classes into pseudo-classes. The pairwise contrasts are then used to find a globally minimal feature set. The theoretical computational complexity of the approach is problematic, but good problem-reduction methods are available that can handle all practical problems we have tried. Any recalcitrant cases can be handled with a greedy set-covering heuristic.

We have applied the CAPP methods collaboratively to significant problems in anthropological linguistics, psychology and chemistry, and have shown that the methods can generate knowledge that is deemed significant and publishable in those fields.

This work was supported by Grants #IIS-9988084 and #IIS-9819340 from the (USA) National Science Foundation, by the (USA) NSF Division of International Programs, and by contract #I-813 with the Bulgarian Ministry of Education and Science. Thanks to the anonymous reviewers for their suggestions which clarified greatly the presentation.

References

ALMUALLIM, H. & DIETTERICH, T. G. (1994). Learning Boolean concepts in the presence of many irrelevant features. *Artificial Intelligence*, **63,** 279–305.

BLAKE, C., KEOGH, E. & MERZ, C. UCI repository of machine learning databases [www.ics.uci.edu/~mlearn/mlrepository.html]. Department of Information and Computer Science, University of California, Irvine, CA.

BLUM, A. & LANGLEY, P. (1997). Selection of relevant features and examples in machine learning. *Artificial Intelligence*, **97,** 245–271.

BOHANEC, M. & BRATKO, I. (1994) Trading accuracy for simplicity in decision trees. *Machine Learning*, **15,** 223–250.

BOLAND, M. & MURPHY, R. (1999). After sequencing: quantitative analysis of protein localization. *IEEE Eng. Med. Biol. Mag.* **18,** 115–119.

BREIMAN, L., FRIEDMAN, J., OLSHEN, R., & STONE, C. (1984). *Classification and Regression Trees.* New York: Chapman & Hall.

CHERRY, C., HALLE, M. & JAKOBSON, R. (1953). Toward the logical description of languages in their phonemic aspect. *Language*, **29,** 34–47.

CHVATAL, V. (1979). A greedy heuristic for the set covering problem. *Mathematics of Operations Research* **4,** 233–235.

CLARK, P. & NIBLETT, T. (1989). The cn2 induction algorithm. *Machine Learning*, **3,** 261–283.

GAREY, M. R. & JOHNSON, D. S. (1979). *Computers and Intractability: A Guide to the Theory of NP-Completeness.* San Francisco: W. H. Freeman.

GOOD, P. (1994). *Permutation Tests*. New York: Springer-Verlag.
GOODENOUGH, W. H. (1967). Componential analysis. *Science*, **156,** 1203–1209.
HEI, M., CHEN, H., YI, J., LIN, Y., LIN, Y., WEI, G., & LIAO, D. (1998). CO_2-reforming of methane on transition metal surfaces. *Surface Science*, **417,** 82–96.
HOLTE, R. C. (1993). Very simple classification rules perform well on most commonly used datasets. *Machine Learning*, **3,** 63–91.
HONG, S. J. (1997). R-MINI: An iterative approach for generating minimal rules from examples. *IEEE Transactions on Knowledge and Data Engineering*, **9,** 709–717.
IBA, W., WOGULIS, J. & LANGLEY, P. (1988). Trading off simplicity and coverage in incremental concept learning. *Proceedings of the 5th International Conference on Machine Learning* pp. 73–79, Los Altos, CA: Morgan Kaufmann.
KAUTZ, W. (1968). Fault testing and diagnosis in combinatorial digital circuits. *IEEE Transactions on Computing*, **C17,** 352–366.
LANGLEY, P., SIMON, H., BRADSHAW, G., & ZYTKOW, J. (1987). *Scientific Discovery*: *Computational Explorations of the Creative Processes*. Cambridge, MA: MIT Press.
LEECH, G. N. (1974). *Semantics*. Harmondsworth, UK: Pelican.
LOUNSBURY, F. (1964). The structural analysis of kinship semantics. In H. LUNT, Ed. *Proceedings of the 9th International Congress of Linguistics*, pp. 1073–1090. The Hague: Mouton and Co.
LOUNSBURY, F. (1965). Another view of the Trobrian kinship categories. *American Anthropologist*, **67,** 142–185. Special publication on Formal Semantic Analysis.
MACWHINNEY, B., FELDMAN, H., SACCO, K. & VALDÉZ PÉREZ, R. E. (2000) Online measures of basic language skills in children with early focal brain lesion. *Brain and Language*.
MICHALSKI, R. & CHILAUSKY, S. (1980). Learning by being told and learning from examples: an experimental comparison of the two methods of knowledge acquisition in the context of developing an expert system for soybean disease diagnosis. *Journal of Policy Analysis and Information Systems*, **4,** 126–161.
MICHALSKI, R., MOZETIC, I., HONG, J. & LAVRAC, N. (1986). The multi-purpose incremental learning system AQ15 and its testing application to three medical domains. *Proceedings of National Conference on Artificial Intelligence*, pp. 1041–1045, Morgan Kaufmann.
MOONEY, R. J. (1995). Encouraging experimental results on learning CNF. *Machine Learning*, **19,** 79–92.
MURPHY, P. & PAZZANI, M. (1994). Exploring the decision forest: An empirical investigation of OCCAM's razor in decision tree induction. *Journal of Artificial Intelligence Research*, **1,** 257–275.
MURPHY, R. & BOLAND, M. (1999). Pattern analysis meets cell biology. *Microstructures and Microanalysis*, *Suppl.* 2: *Proceedings*, **5,** 510–511.
PANKHURST, R. (1983). An improved algorithm for finding diagnostic taxonomic descriptions. *Mathematical Biosciences*, **65,** 209–218.
PERICLIEV, V. & VALDÉS-PÉREZ, R. E. (1997). A discovery system for componential analysis of kinship terminologies. *Proceedings of the 16th International Congress of Linguists*.
PERICLIEV, V. & VALDÉS-PÉREZ, R. E. (1998*a*). Automatic componential analysis of kinship semantics with a proposed structural solution to the problem of multiple models. *Anthropological Linguistics*, **40,** 272–317.
PERICLIEV, V. & VALDÉS-PÉREZ, R. E. (1998*b*). A procedure for multi-class discrimination and some linguistic applications. *Proceedings of Coling-ACL*: *17th International Conference on Computational Linguistics and 36th Annual Meeting of the Association for Computational Linguistics*, pp. 1034–1040.
QUINLAN, J. (1990). Learning logical definitions from relations. *Machine Learning*, **5,** 239–266.
QUINLAN, J. (1993). *C4.5*: *Programs for Machine Learning*. LOS Altos, CA: Morgan Kaufman.
RIVEST, R. (1987). Learning decision lists. *Machine Learning*, **2,** 229–246.
SAITTA, L. & NERI, F. (1998). Learning in the "Real World". *Machine Learning*, **30,** 133–163.
VALDÉS–PÉREZ, R. E. (1994*a*). Algebraic reasoning about reactions: Discovery of conserved properties in particle physics. *Machine Learning*, **17,** 47–68.
VALDÉS–PÉREZ, R. E. (1994*b*). Conjecturing hidden entities via simplicity and conservation laws: Machine discovery in chemistry. *Artificial Intelligence*, **65,** 247–280.

VALDÉS–PÉREZ, R. E. (1999). Principles of human computer collaboration for knowledge discovery in science. *Artificial Intelligence*, **107,** 335–346.
VALDÉS–PÉREZ, R. E. & ZYTKOW, J. M. (1996). Systematic generation of constituent models of particle families. *Physical Review E*, **54,** 2102–2110.
WEBB, G. I. (1995). OPUS: an efficient admissible algorithm for unordered search. *Journal of Artificial Intelligence Research*, **3,** 431–465.
ZEIGARNIK, A. V., VALDÉS–PÉREZ, R. E. & PESENTI, J. (2000). Comparative properties of transition metal catalysts inferred from activation energies of elementary steps of catalytic reactions. *Journal of Physical Chemistry*, **104,** 997–1008.

Paper accepted for Publication by Editor, Professor D. Sleeman

Appendix

This section directly compares profiles against the rules that are extracted from an induced decision tree using the C4.5rules program. The dataset was provided by Robert Murphy, who is a cell biologist at Carnegie Mellon. The data consist of 10 classes, 862 examples roughly evenly distributed among the classes and 85 numeric features having no missing values. The classes represent types of protein localization in cells and the features are different measures of protein expression in cells using image analysis (Boland & Murphy 1999; Murphy & Boland, 1999). The aim of this project is to develop a typology or classification of the spatial patterns of protein presence within cells (since a protein may be present in some parts of cells but not others) and to develop an automated way to assign a new protein to one of these classes, in order to gain clues as to what the protein does. Prof Murphy developed satisfactory classifiers based on neural nets, but is further interested in articulating the class differences in a manner understandable to biologists.

Thus, the advantages of the dataset are that the data are real, it is desirable to describe the classes, and the classes significantly overlap. That is, necessary and sufficient conjunctive conditions for class membership in terms of the available features are not available, except for one of the classes.

We selected three classes to illustrate our main points. The first class is DAP, which contains 87 examples. C4.5 rules extract a single rule:

```
Rule 1:
        DNAimage:distance⇐0
:class DAP
```

Every example in DAP is classified by this rule and no other examples satisfy its premise. The corresponding profile is

```
DAP
     DNAimage:distance = 0.0
          much less than for all other classes
```

(For brevity, all pairwise contrasts will be shown qualitatively, and only the maximum overlap between the target class and the others will be quantified; here the overlap is zero.) Unsurprisingly, the rule and profile are largely identical, since in this simplest case, the data suggest that there exists a necessary and sufficient condition for membership in

DAP.

We consider a second class ERDAK, for which C4.5rules find the following three rules that cover respectively 56, 17 and 7 out of the total of 86 examples in ERDAK, and no examples from the other classes.

```
Rule 1:
     object:number > 19
     DNAimage:distance > 0
     convex_hull:fraction_of_overlap > 0.3069
     Z_4 × 0 > 0.69013
     Z_8 × 2 > 0.066581
     Z_9 × 9 <= 0.0014325
     sum_of_squares <= 535.88
 -> class ERDAK

Rule 2
     object:EulerNumber <= 8
     object_size:ratio > 995
     DNAimage:overlap <= 0.74566
     Z_12 × 12 <= 0.00051321
 -> class ERDAK

Rule 3:
     object:number > 19
     convex_hull:fraction_of_overlap > 0.54333
     Z_2 × 0 > 0.47842
     edges:direction_maxmin_ratio <= 1.43
 -> class ERDAK
```

Since the first rule covers the most examples in the target class, we could select it as the best C4.5rules approximation to a profile. (An alternative is to try to combine rules to obtain more coverage, which presents its own sets of complications. Also, we would not even know how to compare one profile against multiple rules.) The corresponding profile (with a maximum pairwise overlap of 0.2) is

```
ERDAK
     DNA/image : overlap(0.10 to 0.67, mean = 0.43, sd = 0.11)
          much more than for GPP130 MC151 PHAL
          much less than for DAP NUCLE
     edges : direction_maxmin_ratio(1.1 to 1.9, mean = 1.3, sd = 0.14)
          much less than for GIANTIN GPP130 NUCLE PHAL TUBUL
     object: eulernumber( − 61 to 124, mean = − 4.9, sd = 31.5)
          much less than for TFR
     object_size : ratio(1240.5 to 20961, mean = 8745.0, sd = 3712.4)
          much more than for DAP GIANTIN GPP130 H4B4 NUCLE
```

Now the best rule and the best profile diverge more than in the first example, since the rule has appreciably more features among its premises. The reason is that the extracted

rule emphasizes predictive accuracy, hence it identifies a sub-class of ERDAK which, in this case, is perfectly accurate on all the examples. The profile uses fewer features, but it is also less precise, since it emphasizes finding an approximate description that takes all class examples into account, rather than finding coherent subclasses within the target class.

One difference between rules and profiles that becomes noticeable is that a profile explicitly states which classes are contrasted by which features. In our view, making the inter-class contrasts explicit is important for gaining a concise understanding of a moderate-to-large classification that may have highly overlapping classes. Of course, rule descriptions could be annotated with this information by comparing the target class against each of the other classes, but only as an afterthought, rather than as a designed approach to the comparison of all class pairs. The difference between afterthought and purposeful design will be clearer after our third example.

Let us consider a third and final class TFR, where the ensuing rules apply, respectively, to 22, 21, 7, 6, 3 and 8 out of the original 91 examples, but also mistakenly classify some negative examples into the target class TFR.

```
Rule 41:
     object:EulerNumber > 135
     DNAimage:overlap > 0.082192
     Z_1 × 1 > 0.00078213
     Z_2 × 0 > 0.47842
     Z_4 × 0 ⩽ 0.69013
     info_measure_corr_2 ⩽ 0.95648
 -> class TFR

Rule 54:
     object:EulerNumber > 120
     Z_4 × 0 > 0.69013
     Z_12 × 6 ⩽ 0.050937
     angular_second_moment ⩽ 0.0021489
 -> class TFR

Rule 71:
     convex_hull:fraction_of_overlap > 0.3049
     Z_4 × 0 > 0.69013
     Z_9 × 3 > 0.023537
     Z_9 × 9 > 0.0014325
     sum_of_squares > 146.818
     edges:direction_maxmin_ratio ⩽ 1.3609
 -> class TRF

Rule 46:
     object:EulerNumber⇐120
     object_size:average⇐25.5714
     object_size:ratio > 973
     object_distance:variance⇐965.905
 -> class TFR
```

```
Rule 53:
     angular_second_moment ⩽ 0.0021489
     edges:area_fraction > 0.81231
 -> class TFR

Rule 25:
     DNAimage:overlap > 0.065544
     Z_2 × 0 ⩽ 0.47842
     correlation ⩽ 0.61695
     edges:area_fraction > 0.49792
 -> class TFR
```

This class is clearly more complicated than the first two, since the rules misclassify some examples and they capture only small subclasses: at best, slightly under 25% of the class examples. It is unclear which rule is best, since the rule with the highest coverage also has the most premises. The profile for TFR (with a maximum pairwise overlap of 0.42) is

```
TFR
     DNA image: distance (1.4 to 92.0, mean = 29.2, sd = 15.5)
         much more than for DAP
         more than for NUCLE
     DNA image: overlap (0.07 to 0.66, mean = 0.30, sd = 0.13)
         more than for GPP130 MC151 PHAL
         much less than for DAP NUCLE
     object: number (3 to 1425, mean = 219.9, sd = 230.5)
         much more than for DAP GIANTIN GPP130 NUCLE
         more than for ERDAK H4B4
     object_size: average (9.0 to 2790.3, mean = 97.1, sd = 317.7)
         much less than for DAP ERDAK NUCLE PHAL TUBUL
         less than for GIANTIN GPP130
```

Four features suffice to convey the gross differences between TFR and the rest. Thus, TFR's examples tend to have intermediate values of the second feature, as shown by simultaneously tending to be "more than" and "less than" the feature values of some other classes.

Conclusion: Our profiling methods minimize the number of features needed to ensure that all pairs of classes are contrasted explicitly by at least one feature. In principle, the coverage is 100% (all examples of every class are taken into account); what varies is a single reported parameter that expresses the largest of the minimum (MAXIMIN) overlaps between the feature values from every pair of classes. The goal is to provide an approximate description of all class examples, rather than to look for coherent subclasses.

It is possible to turn a rule extracted by C4.5rules into something that resembles a profile by explicitly comparing the target class with every other class along each of the rule's features, annotating the rule with the results of these pairwise comparisons and finally discarding the rule's premises. That is, if a rule contains a premise $Z\_2 \times 10 \leqslant 0.47842$, then the "profilized" rule would instead say that $Z\_2 \times 0$ contrasts the target class from classes $X1$, $X2$, and $X5$.

To turn rules into good profiles, several decisions would need to be made:

- Exactly what is meant by contrasting two classes with one feature.
- How to guarantee that each of the other classes is contrasted with the target class.
- How to define and ensure minimality of the individual profile, as well as of the joint set of all class profiles. The above examples have treated rules in isolation, but in some applications (e.g. the kinship terminologists described elsewhere) it is desirable to minimize the overall features that are used, thus some inter-class coordination in the selection of rules would be needed.
- How to trade off the rule's coverage within the class against the rate of misclassified examples.

Most of these issues are "off the track" for rules, whose raison d'etre is to identify reliably predictive subclasses hidden within larger, possibly overlapping classes. Rules tend to trade off coverage to gain accuracy, whereas profiles are designed to emphasize coverage over precision.