RUTCOR
RESEARCH
REPORT

EXACT AND APPROXIMATE DISCRETE OPTIMIZATION ALGORITHMS FOR FINDING USEFUL DISJUNCTIONS OF CATEGORICAL PREDICATES IN DATA ANALYSIS

Endre Boros[a] Vladimir Meñkov[b]

RRR 18-2003, April 12, 2003

RUTCOR
Rutgers Center for
Operations Research
Rutgers University
640 Bartholomew Road
Piscataway, New Jersey
08854-8003
Telephone: 732-445-3804
Telefax: 732-445-5472
Email: rrr@rutcor.rutgers.edu
http://rutcor.rutgers.edu/∼rrr

[a] RUTCOR, Rutgers University, 640 Bartholomew Road, Piscataway, NJ 08854-8003, USA, boros@rutcor.rutgers.edu

[b] Independent Consultant, Penticton, British Columbia, Canada, vmenkov@cs.indiana.edu

Rutcor Research Report
RRR 18-2003, April 12, 2003

Exact and approximate discrete optimization algorithms for finding useful disjunctions of categorical predicates in data analysis

Endre Boros Vladimir Meñkov

Abstract. We discuss a discrete optimization problem that arises in data analysis from the binarization of categorical attributes. It can be described as the maximization of a function $F(l_1(\mathbf{x}), l_2(\mathbf{x}))$, where $l_1(\mathbf{x})$ and $l_2(\mathbf{x})$ are linear functions of binary variables $\mathbf{x} \in \{0,1\}^n$, and $F : \mathbb{R}^2 \longrightarrow \mathbb{R}$.

Though this problem is NP-hard, in general, an optimal solution $\mathbf{x}^*$ of it can be found, under some mild monotonicity conditions on F, in pseudo-polynomial time. We also present an approximation algorithm which finds an approximate binary solution $\mathbf{x}^\epsilon$, for any given $\epsilon > 0$, such that $F(l_1(\mathbf{x}^*), l_2(\mathbf{x}^*)) - F(l_1(\mathbf{x}^\epsilon), l_2(\mathbf{x}^\epsilon)) < \epsilon$, at the cost of no more than $O(n \log n + 2^{C/\sqrt{\epsilon}} n)$ operations. Though in general C depends on the problem instance, for the problems arising from binarization of categorical variables it depends only on F, and for all functions considered we have $C \leq 1/\sqrt{2}$.

Acknowledgements: This research was partially supported by the National Science Foundation, Grant IIS-0118635, by the Office of Naval Research, Grant N00014-92-J-1375, and by the Rutgers Distributed Laboratory for Digital Libraries, a Strategic Opportunity Project of Rutgers, the State University of New Jersey.

1 Introduction

Feature derivation (or selection) is an important task in data analysis and machine learning. Most practical data sets have many irrelevant attributes, and the eliminations of those, or more precisely, the selection/derivation of the few relevant features is vital to derive efficiently results which are acceptable with high confidence [6]. There is a large number of fairly recent publications addressing feature derivation by various techniques, including optimization methods, as well as statistical approaches (see e.g., [2, 3, 5, 10, 11, 12, 14, 16, 17, 22, 23]). Binarization (or sometime called dichotomization) of the attributes is a standard method to derive simple binary attributes, used by most rule based learning approaches (see e.g., [9, 13, 26, 27]). While the binarization of quantitative or ordinal attributes (e.g., real or integer valued, etc.) is quite well studied (see e.g., [1, 8, 21, 24, 25]), binarization of categorical attributes received much less attention. In fact most statistical methods are applicable with high confidence only if the number of different values of such a categorical attribute is relatively small.

In this paper we consider the problem of deriving a ”most relevant” binary attribute from a categorical one, which has possibly many different, unrelated values.

Let us consider a data set $D = D^+ \cup D^-$, where D^+ is the set of positive examples and D^- is the set of negative examples ($D^+ \cap D^- = \emptyset$ is usually assumed, though this will not be essential for our analysis). Let us further consider a categorical attribute $\mathbf{C}$, which has n different, unrelated values, i.e., which divides the data set D into n non-overlapping non-empty categories C_1, C_2, …, C_n, i.e.

$$D^+ \cup D^- = \bigcup_{i=1}^{n} C_i \quad \text{and} \quad C_i \cap C_j = \emptyset \quad \text{for} \quad 1 \leq i < j \leq n.$$

It is customary to represent $\mathbf{C}$ by the *categorical predicates* $P_i : D \to \{0, 1\}$ defined by $P_i(d) = 1$ if and only if $d \in C_i$, for $i = 1, ..., n$. Most rule based learning algorithms would in fact use only these simple predicates and no other information about $\mathbf{C}$. Though theoretically this is a correct approach, since the set of these n predicates describe perfectly $\mathbf{C}$, in practice only a few binary attributes are selected by the learning algorithm, and since the selection is typically based on some measure of significance, not all predicates P_i may be chosen. In particular, if n is large, each predicate P_i is active only on a small fraction of the training data, and hence the information it carries individually may look insignificant. Consequently, even if $\mathbf{C}$ is a highly relevant attribute, it may not have any effect on the obtained classifier. To address this issue, one may consider all predicates of the form

$$P_{\mathbf{x}} = \bigvee_{i:x_i=1} P_i,$$

where $\mathbf{x} \in \{0, 1\}^n$. However, to consider all 2^n possible disjunctions is not feasible in practice whenever n is large. It is therefore desirable to obtain one or a small number of such disjunctions that are the most useful for distinguishing elements of D^+ from elements of D^- (as much as possible, in the case $D^+ \cap D^- \neq \emptyset$).

To model this problem, let us introduce some notations first. Let $|S|$ denote the cardinality of the set $S \subseteq D$, and for a predicate $P : D \to \{0,1\}$ let $P(S) = \{d \in S \mid P(d) = 1\}$. Let us denote the fractions of the positive and negative examples on which a predicate P holds by

$$X(P) = \frac{|P(D^+)|}{|D^+|} \quad \text{and} \quad Y(P) = \frac{|P(D^-)|}{|D^-|}.$$

Let us further assume that the *distinguishing power* of a predicate P (or its significance, or relevance) is measured by $F(X(P), Y(P))$, where $F : \mathbb{R}^2 \to \mathbb{R}$ is a real valued function in two real variables. With these notations, our main problem can be formulated as

$$\max_{\mathbf{x}\in\{0,1\}^n} F(X(P_\mathbf{x}), Y(P_\mathbf{x})).$$

Let us further introduce the notation $a_i = |P_i(D^+)|/|D^+|$ and $b_i = |P_i(D^-)|/|D^-|$ for $i = 1, ..., n$, and $a_0 = b_0 = 0$. Then we have

$$X(P_\mathbf{x}) = l_1(\mathbf{x}) = a_0 + \sum_{i=1}^{n} a_i x_i \quad \text{and} \quad Y(P_\mathbf{x}) = l_2(\mathbf{x}) = b_0 + \sum_{i=1}^{n} b_i x_i,$$

and thus we can equivalently write our main problem as

$$\max_{\mathbf{x}\in\{0,1\}^n} F(l_1(\mathbf{x}), l_2(\mathbf{x})), \tag{1}$$

or in other words as the maximization of an expression of two linear functions in binary variables. We included a_0 and b_0, with possibly nonzero values, to allow to model more general situations, as well.

2 Main Results

In Section 3 we discuss a few possible choices for $F(X, Y)$, or in other words for measuring the quality (distinguishing power) of a predicate P, and we also discuss the complexity of problem (1) in the corresponding cases. As it will be seen, our problem includes as special cases some easy problems, like the maximization of a linear function in binary variables, which is trivially solvable in $O(n)$ time, or the maximization of the ratio of two linear functions $F(l_1(\mathbf{x}), l_2(\mathbf{x})) = l_1(\mathbf{x})/l_2(\mathbf{x})$ in binary variables where $l_2(\mathbf{x}) > 0$ for all $\mathbf{x} \in \{0,1\}^n$, which was shown to be solvable in $O(n \log n)$ time in [19], and which also has applications in database theory (see e.g., [20]). It also includes as a special case the maximization of the product of two linear functions in binary variables, which was shown to be NP-hard in [18], and therefore problem (1) is also NP-hard, in general. Let us remark that the maximization of the ratio of two linear functions in the more general case when $l_2(\mathbf{x})$ can take both positive and negative values is also NP-hard (see e.g., [7]).

In Section 4 we consider the general discrete optimization problem (1) under some monotonicity condition and provide a solution via dynamic programming. To simplify our statements, we shall assume throughout this paper that $F(X, Y)$ can be computed in $O(1)$ time, for every given $(X, Y) \in \mathbb{R}^2$. This condition is certainly fulfilled by the functions we consider in Section 3.

Theorem 1 *Let us assume that the values of the linear function $l_1(\mathbf{x})$ at binary points are all of the form $p + jq$ for $0 \leq j \leq L$ for some integer L and reals p, q, and that the function $F(X, Y)$ satisfies the following monotonicity condition:*

(M) *For any given fixed value of X, the function $F(X, Y)$ must be a monotonically non-decreasing, or a monotonically non-increasing function of Y. (It may be non-decreasing for some values of X, and non-increasing for others).*

Then, problem (1) can be solved in $O(nL)$ time.

Clearly, the roles of l_1 and l_2 could be interchanged in the above statement. Let us also add that L can be exponentially large in n, in the worst case. However, for the case of finding a "best" predicate as described in the previous section, we have $L \leq |D^+|$, and thus we have the following statement readily implied.

Corollary 1 *Given a data set $D = D^+ \cup D^-$, and a function $F(X, Y)$ satisfying condition (M), then a predicate $P_{\mathbf{x}}$, $\mathbf{x} \in \{0, 1\}^n$ maximizing $F(X(P_{\mathbf{x}}), Y(P_{\mathbf{x}}))$ can be found in $O(n|D^+|)$ time.*

In Section 5 we consider the continuous relaxation of problem (1)

$$\max_{\mathbf{x} \in [0,1]^n} F(l_1(\mathbf{x}), l_2(\mathbf{x})). \tag{2}$$

We show first that results obtained in [18] for the case of $F(X, Y) = XY$ can analogously be extended to the case of an arbitrary function $F(X, Y)$ satisfying condition (M). Based on these results, in Section 6 we provide a polynomial time approximation algorithm for problem (1).

Theorem 2 *Let us consider problem (1), and assume that $F(X, Y)$ satisfies condition (M) as well as the following smoothness condition:*

(S) *The partial second derivatives of $F(X, Y)$ must be uniformly bounded in the sense that there is a constant K such that for any point $(X, Y) = (l_1(\mathbf{x}), l_2(\mathbf{x}))$ for some $\mathbf{x} \in [0, 1]^n$ we have the inequalities*

$$|\partial^2 F(X, Y)/\partial X^2| < K, \quad |\partial^2 F(X, Y)/\partial Y^2| < K, \quad \textit{and} \quad |\partial^2 F(X, Y)/\partial X \partial Y| < K.$$

Then, for every $\epsilon > 0$ we can find in $O(n \log n + 2^{C/\sqrt{\epsilon}} n)$ time a binary vector $\mathbf{x}^\epsilon$ for which $F(l_1(\mathbf{x}), l_2(\mathbf{x})) \leq F(l_1(\mathbf{x}^\epsilon), l_2(\mathbf{x}^\epsilon)) + \epsilon$ holds for all $\mathbf{x} \in \{0,1\}^n$, where $C = (\sum_1^n(|a_i| + |b_i|))\sqrt{K/8}$. Furthermore, $C = \sqrt{(\sum_1^n |a_i|)(\sum_1^n |b_i|)/2}$ in the case of $F(X,Y) = XY$.

By noting that for the best predicate selection we have $\sum_{i=1}^n |a_i| = \sum_{i=1}^n |b_i| = 1$, we obtain the following statement:

Corollary 2 *Given a data set $D = D^+ \cup D^-$, and a function $F(X,Y)$ satisfying conditions (M) and (S), then for every $\epsilon > 0$ a predicate $P_{\mathbf{x}^\epsilon}$ such that $F(X(P_{\mathbf{x}^\epsilon}), Y(P_{\mathbf{x}^\epsilon})) + \epsilon \geq F(X(P_\mathbf{x}), Y(P_\mathbf{x}))$ for all $\mathbf{x} \in \{0,1\}^n$ can be found in $O(n \log n + 2^{C/\sqrt{\epsilon}} n)$ operations, where $C = \sqrt{K/2}$. Furthermore, $C = 1/2$ in the case of $F(X,Y) = XY$.*

Let us add that for all the functions mentioned in Section 3 the value $K = 1$ is sufficient. Furthermore, for one of the naturally arising measures of distinguishing power of predicates, the above approximation results can be further strengthened.

Corollary 3 *Given a data set $D = D^+ \cup D^-$, and a function of the form $F(X,Y) = c + XY$ for some $c > 0$, and let us denote by $\mathbf{x}^*$ an optimal solution of the corresponding problem (1). Then, for every $\epsilon > 0$ we can find in $O(n \log n + 2^{1/2\sqrt{c\epsilon}} n)$ time a binary vector $\mathbf{x}^\epsilon$ satisfying*

$$\frac{F(X(P_{\mathbf{x}^*}), Y(P_{\mathbf{x}^*})) - F(X(P_{\mathbf{x}^\epsilon}), Y(P_{\mathbf{x}^\epsilon}))}{|F(X(P_{\mathbf{x}^*}), Y(P_{\mathbf{x}^*}))|} \leq \epsilon.$$

3 Measuring distinguishing power

How can we measure the distinguishing power of a predicate P, in other words, its ability to distinguish between elements of D^+ and D^-?

A natural idea to consider is the fraction of pairs of positive, negative examples (d, d'), $d \in D^+$ and $d' \in D^-$ which are distinguished by predicate P, i.e., for which $P(d) \neq P(d')$. It is easy to see that the corresponding function F can be written as

$$F(X,Y) = X(1-Y) + (1-X)Y. \tag{3}$$

It is also immediate to see that problem (1) with this choice of F is equivalent with the maximization of the product of two linear functions, in binary variables, since we have $X(1-Y) + (1-X)Y = (2X-1)(\frac{1}{2} - Y) + \frac{1}{2}$. This latter optimization problem is NP-hard, in general, as shown in [20], and therefore (1) is not easier either, in this case.

Let us add that the measure $\zeta_{quad}(P) = F(X(P), Y(P))$ based on (3) is normalized, taking values between 0 and 1, and it appears quite sensible, in the sense that a perfectly distinguishing predicate P — one which is true on all positive examples and false on all negative ones, or vice versa — will have $\zeta_{quad}(P) = 1$, while a trivial predicate P that has the same value on all objects will have $\zeta_{quad}(P) = 0$. In fact ζ_{quad} is used (implicitly) in many set covering formulation based feature selection algorithms (see e.g., [2, 9]). Let us note however that a predicate P that appears completely random — one that is true on half of all records from D^+ and on half of all records from D^- — will have $\zeta_{quad}(P) = F(0.5, 0.5) = 0.5$, which is perhaps not as intuitively appealing.

An alternative measure $\zeta_{lin}(P) = F(X(P), Y(P))$ of distinguishing power, can be based on comparing the probabilities with which P holds on D^+ and D^-. The corresponding function F can be written as

$$F(X, Y) = |X - Y|. \tag{4}$$

This simple measure yields $\zeta_{lin}(P) = 1$ on a perfectly distinguishing predicate P, and $\zeta_{lin}(P) = 0$ on any predicate that returns true with the same probability on D^+ and D^- (i.e., has $X(P) = Y(P)$).

The corresponding optimization problem (1) can be written as the maximization of $\max[l_1(\mathbf{x}) - l_2(\mathbf{x}),\, l_2(\mathbf{x}) - l_1(\mathbf{x})]$ over $\{0, 1\}^n$, and hence it is equivalent with the maximization of a linear function in binary variables, which can trivially be solved in $O(n)$ time. Let us also remark that a very similar measure can be obtained by considering the correlation of the predicate P and the true classification. In fact the measure based on (4) behaves very similarly to such a correlation based measure.

A third possible measure $\zeta_{odds}(P) = F(X(P), Y(P))$ is related to the so called odds ratio, and is based on the function

$$F(X, Y) = c \max\left\{\frac{X + c}{Y + c}, \frac{Y + c}{X + c}\right\} - c, \tag{5}$$

where $c > 0$ is a given constant. When $c \to +\infty$, (5) becomes equivalent to (4) with respect to how it ranks predicates by their distinguishing power. This measure is also normalized, taking values between 0 and 1, and it yields $\zeta_{odds}(P) = 1$ on a perfectly distinguishing predicate P, and $\zeta_{odds}(P) = 0$ on any predicate for which $X(P) = Y(P)$.

Since $X(P) \geq 0$ and $Y(P) \geq 0$, the corresponding optimization problem (1) is equivalent in this case with the maximization of the ratio of two linear functions in binary variables, where the denominator is strictly positive. As we mentioned earlier, this problem was shown to be solvable in $O(n \log n)$ time in [19].

4 A pseudo-polynomial exact algorithm

Let us return now to the general discrete maximization problem (1), and let us show first that we can assume, without any loss of generality, that $l_1(\mathbf{x})$ is a monotonically non-decreasing function in each of its variables.

This is because we can introduce a one-to-one mapping $\mathbf{y} : \{0,1\}^n \longleftrightarrow \{0,1\}^n$ by defining

$$y_i = \begin{cases} x_i & \text{if } a_i \geq 0 \\ 1 - x_i & \text{if } a_i < 0. \end{cases}$$

Defining further $\widehat{l_1}(\mathbf{y}) = \widehat{a}_0 + \widehat{a}_1 y_1 + \cdots \widehat{a}_n y_n = l_1(\mathbf{y}(\mathbf{x}))$ and $\widehat{l_2}(\mathbf{y}) = \widehat{b}_0 + \widehat{b}_1 y_1 + \cdots \widehat{b}_n y_n = l_2(\mathbf{y}(\mathbf{x}))$, we have

$$\widehat{a}_0 = a_0 + \sum_{i:a_i<0} a_i \quad \text{and} \quad \widehat{b}_0 = b_0 + \sum_{i:a_i<0} b_i,$$

and

$$\widehat{a}_i = \begin{cases} a_i \text{ if } a_i \geq 0 \\ -a_i \text{ if } a_i < 0 \end{cases} \quad \text{and} \quad \widehat{b}_i = \begin{cases} b_i \text{ if } a_i \geq 0 \\ -b_i \text{ if } a_i < 0 \end{cases}$$

for $i = 1, ..., n$. Since $\mathbf{y}$ is a bijection on $\{0,1\}^n$, we have

$$\max_{\mathbf{y} \in \{0,1\}^n} F(\widehat{l_1}(\mathbf{y}), \widehat{l_2}(\mathbf{y})) = \max_{\mathbf{x} \in \{0,1\}^n} F(l_1(\mathbf{x}), l_2(\mathbf{x}))$$

and thus, for the problem on the left hand side the condition $\widehat{a}_i \geq 0$ holds, for all $i = 1, ..., n$, as desired.

Since the above transformation can be done in $O(n)$ time, we can assume in the sequel without any loss of generality that $a_i \geq 0$ for all $i = 1, ..., n$.

Proof of Theorem 1: Let $W = \{p + jq \mid j = 0, ..., L\}$, for the reals p and $q \geq 0$, as in the theorem, such that $l_1(\mathbf{x}) \in W$ for all $\mathbf{x} \in \{0,1\}^n$ by our assumption. Let us further define

$$\begin{aligned} z(k,w) &= \min_{\substack{(x_1,\ldots,x_k) \in \{0,1\}^k \\ a_0 + a_1 x_1 + \cdots a_k x_k = w}} b_0 + b_1 x_1 + \cdots b_k x_k, \quad \text{and} && (6) \\ Z(k,w) &= \max_{\substack{(x_1,\ldots,x_k) \in \{0,1\}^k \\ a_0 + a_1 x_1 + \cdots a_k x_k = w}} b_0 + b_1 x_1 + \cdots b_k x_k, && (7) \end{aligned}$$

for all $w \in W$, and let $z(k,w) = +\infty$ and $Z(k,w) = -\infty$ whenever $w \notin W$, for $k = 0, 1, ..., n$. Then, since F is assumed to satisfy condition (M), we have for every $w \in W$ that

$$\max_{\mathbf{x} \in \{0,1\}^n, l_1(\mathbf{x}) = w} F(l_1(\mathbf{x}), l_2(\mathbf{x})) = \max\{F(w, z(n,w)), F(w, Z(n,w))\},$$

and therefore

$$\max_{\mathbf{x} \in \{0,1\}^n} F(l_1(\mathbf{x}), l_2(\mathbf{x})) = \max_{w \in W}\{F(w, z(n,w)), F(w, Z(n,w))\} \tag{8}$$

is implied. Since we assume that $F(X,Y)$ can be computed in $O(1)$ time for a given $(X,Y) \in \mathbb{R}^2$, the right hand side of (8) can be determined in $O(L)$ time. Thus, to prove the theorem, it is enough to show that the quantities $z(n,w)$ and $Z(n,W)$ for $w \in W$ can be computed in $O(nL)$ time.

To this end, let us note that these quantities satisfy the following recursive equations

$$\begin{aligned} Z(k+1,w) &= \max\{Z(k,w),\ Z(k,w-a_{k+1})+b_{k+1}\}, \\ z(k+1,w) &= \min\{z(k,w),\ z(k,w-a_{k+1})+b_{k+1}\}. \end{aligned}$$

Since we have $a_i \geq 0$ for all $i = 1, ..., n$, we can solve these recursions starting with the trivial initial values

$$Z(0,w) = \begin{cases} -\infty, & w \neq a_0 \\ b_0, & w = a_0 \end{cases} \quad \text{and} \quad z(0,w) = \begin{cases} +\infty, & w \neq a_0 \\ b_0, & w = a_0. \end{cases}$$

Thus, we can determine $Z(k,w)$ and $z(k,w)$ for all $0 \leq k \leq n$ and $w \in W$ in $O(nL)$ time, completing our proof. □

5 Solving the continuous problem

As a preparation for presenting an approximate optimization method for the discrete problem (1), we will analyze in this section the continuous optimization problem (2) and its relation to the discrete one. Analogous results has been presented in [18] for the case of $F(X,Y) = XY$. We shall recall here below some of these results, and provide straightforward extensions of some other results for the case of functions $F(X,Y)$ satisfying condition (M).

Let us introduce

$$\Omega(l_1,l_2) = \{(l_1(\mathbf{x}), l_2(\mathbf{x})) \mid \mathbf{x} \in [0,1]^n\} \subseteq \mathbb{R}^2, \tag{9}$$

and let us call a point $(X,Y) \in \mathbb{R}^2$ *feasible* for the continuous problem (2) if $(X,Y) \in \Omega(l_1,l_2)$. Let us also introduce

$$\Delta(l_1,l_2) = \{(l_1(\mathbf{x}), l_2(\mathbf{x})) \mid \mathbf{x} \in \{0,1\}^n\} \subseteq \mathbb{R}^2, \tag{10}$$

and let us call a point $(X,Y) \in \mathbb{R}^2$ *feasible* for the discrete problem (1) if $(X,Y) \in \Delta(l_1,l_2)$.

Clearly, $\Omega(l_1,l_2)$ is a closed, convex subset in $\mathbb{R}^2$ and $\Delta(l_1,l_2) \subseteq \Omega(l_1,l_2)$.

With these notations problem (1) can be reformulated as

$$\max_{(X,Y)\in\Delta(l_1,l_2)} F(X,Y),$$

while its continuous relaxation (2) can be written equivalently as

$$\max_{(X,Y)\in\Omega(l_1,l_2)} F(X,Y).$$

In our analysis of the latter problem, let us first recall from [18] that $\Omega(l_1,l_2)$ is in fact the convex hull of $\Delta(l_1,l_2)$.

Lemma 1 (Proposition 3.1 in [18]) *Let us assume $a_i \geq 0$ for $i = 1, ..., n$, as in the previous section, and let π be a permutation of the indices $1, 2, ..., n$, such that*

$$\frac{b_{\pi(1)}}{a_{\pi(1)}} \geq \frac{b_{\pi(2)}}{a_{\pi(2)}} \geq \cdots \geq \frac{b_{\pi(n)}}{a_{\pi(n)}}, \tag{11}$$

where we assume $\frac{x}{0} = +\infty$ if $x \geq 0$, and $\frac{x}{0} = -\infty$ if $x < 0$. Let us further define binary vectors by

$$\mathbf{q}^j_{\pi(i)} = \begin{cases} 1 & \text{if } i \leq j, \\ 0 & \text{otherwise,} \end{cases} \quad \text{and} \quad \overline{\mathbf{q}}^j_{\pi(i)} = \begin{cases} 1 & \text{if } i > j, \\ 0 & \text{otherwise,} \end{cases} \tag{12}$$

for $i = 1, ..., n$ and $j = 1, ..., n$. Let further $Q_j = (l_1(\mathbf{q}^j), l_2(\mathbf{q}^j)) \in \mathbb{R}^2$ and $\overline{Q}_j = (l_1(\overline{\mathbf{q}}^j), l_2(\overline{\mathbf{q}}^j)) \in \mathbb{R}^2$ for $j = 1, ..., n$, and finally let $\mathcal{Q} = \{Q_1, ..., Q_n, \overline{Q}_1, ..., \overline{Q}_n\}$. Then, we have

$$\Omega(l_1, l_2) = \text{conv}(\mathcal{Q}).$$

Proof. Let us fix an arbitrary value $X^* = l_1(\mathbf{x})$ for some $\mathbf{x} \in [0, 1]^n$, and let us denote by Y^- and Y^+, respectively, the minimum and maximum values of Y such that $(X^*, Y) \in \Omega(l_1, l_2)$. Then, by the definition of $\Omega(l_1, l_2)$ we have

$$\begin{array}{rcll} Y^- & = & \min\ b_1x_1 + b_2x_2 + \cdots + b_nx_n & \\ & & s.t.\ \ a_1x_1 + a_2x_2 + \cdots + a_nx_n & = X^* \\ & & \quad 0 \leq x_i \leq 1,\ \text{for } i = 1, ..., n, & \end{array} \quad \text{and} \quad \begin{array}{rcll} Y^+ & = & \max\ b_1x_1 + b_2x_2 + \cdots + b_nx_n & \\ & & s.t.\ \ a_1x_1 + a_2x_2 + \cdots + a_nx_n & = X^* \\ & & \quad 0 \leq x_i \leq 1,\ \text{for } i = 1, ..., n. & \end{array}$$

These are continuous knapsack problems, and thus by an old result of [15] the optimal solutions have the form

$$Y^- = l_2(\alpha\overline{\mathbf{q}}^{j-1} + (1 - \alpha)\overline{\mathbf{q}}^j) \text{ and } Y^+ = l_2(\alpha'\mathbf{q}^{j'-1} + (1 - \alpha')\mathbf{q}^{j'})$$

for some reals $0 \leq \alpha, \alpha' \leq 1$, and indices j and j', from which the statement readily follows. □

The importance of the above characterization of $\Omega(l_1, l_2)$ is that the function $F(X, Y) = XY$ was shown in [18]to take its optimum on the border of this region. This property can easily be generalized for a much wider family of functions.

Lemma 2 *If a continuous function $F(X, Y)$ satisfies the monotonicity condition (M), then for any convex compact domain $\Omega \subseteq \mathbb{R}^2$, the maximum (and minimum) of $F(X, Y)$ on Ω is attained on the boundary of Ω.*

Proof. The function F being continuous, and the domain Ω being compact, F must attain its maximum at some point $(X^*, Y^*) \in \Omega$:

$$F(X^*, Y^*) = \max_{(X,Y)\in\Omega} F(X, Y) = F^*. \tag{13}$$

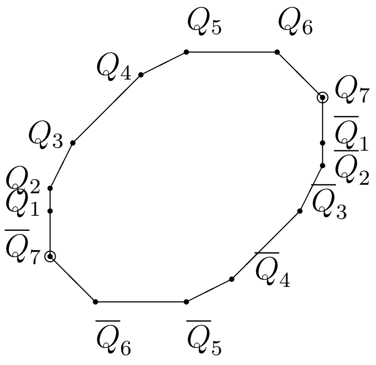


Figure 1: The feasibility domain $\Omega(l_1, l_2)$, when $n = 7$, $a_1 = a_2 = 0$, $b_1 > 0, b_2 > 0$, and $b_3/a_3 \geq b_4/a_4 \geq \cdots \geq b_7/a_7$.

Consider the points (X^*, Y^+) and (X^*, Y^-), where

$$Y^+ = \max_{(X^*,Y)\in\Omega} Y \quad \text{and} \quad Y^- = \min_{(X^*,Y)\in\Omega} Y.$$

They must exist, since Ω is compact. Furthermore, the points (X^*, Y^+) and (X^*, Y^-) must lie on the boundary of Ω (in fact, they belong to the intersection of the vertical line $X = X^*$ with the boundary of Ω). By construction, $Y^- \leq Y^* \leq Y^+$. By the monotonicity condition (M), $F(X^*, \cdot)$ is either monotonically non-increasing or monotonically non-decreasing, which means that either $F(X^*, Y^-) \geq F(X^*, Y^*) = F^*$, or $F(X^*, Y^+) \geq F(X^*, Y^*) = F^*$. But by the definition (13) of F^*, neither $F(X^*, Y^-)$ nor $F(X^*, Y^+)$ can be greater than F^*, implying that either $F(X^*, Y^-) = F^*$, or $F(X^*, Y^+) = F^*$. □

It follows from the above that if the variables are already sorted in accordance with (11), the solution of the continuous optimization problem (2) can be found in $O(n)$ operations, by what we call *border tracing*, i.e., by visiting all breakpoints Q_i and $\overline{Q}_i$ successively, and finding the maximum of $F(X, Y)$ for each of the $2n$ segments of the border of $\Omega(l_1, l_2)$. This complexity is of course based on the assumption that the one dimensional optimization problem $\max_{0\leq\lambda\leq1} F(l_1(\mathbf{a}+\lambda\mathbf{b}), l_2(\mathbf{a}'+\lambda\mathbf{b}'))$ can be solved in $O(1)$ time, a condition fulfilled by all functions considered in this paper. (In practice, fewer than $2n$ nodes and segments may have to be visited. It can be shown, for example that, for $F(X, Y) = XY$ if $F(Q_{i-1}) < F(Q_i)$ and $F(Q_i) > F(Q_{i+1})$, then $F(Q_{i+1}) \geq F(Q_{i+2}) \geq \cdots \geq F(Q_n)$, and hence no segments above Q_{i+1} need to be searched).

The simpler version of border tracing in which only the $2n$ values $F(Q_i)$ and $F(\overline{Q}_i)$, $i = 1, ..., n$ are compared, and no segment maxima are evaluated, will be referred to as *corner hopping.*

Since in the original problem (2) the variables are not sorted in accordance with (11), a sorting procedure, which may involve $O(n \log n)$ operations, may need to be carried out before border tracing (or corner hopping) can start. However, if we need to solve several optimization problems whose sets of coefficient pairs (a_i, b_i) are subsets of the same master set, than only one sorting (sorting of the master set) needs to be done, after which all problems can be solved using that same ordering. This fact may be utilized in the design of algorithms for the approximative solution of the discrete problem.

6 Approximate solutions for the discrete problem

The vertices Q_i, $\overline{Q}_i$ of $\Omega(l_1, l_2)$ are feasible points not only of the continuous problem (2) but also of the discrete problem (1). Therefore, the $O(n)$ corner hopping process finding the maximum

$$\max\{F(Q_0), F(Q_1), F(Q_2), \dots, F(Q_n), F(\overline{Q}_1), F(\overline{Q}_2), \dots, F(\overline{Q}_n)\}$$

obtains some kind of approximate solution to the discrete problem. The question is, how good is this approximation?

In the easy cases of linear or ratio-based functions $F(X, Y)$, e.g., as in (4) or (5), the maximum of F on $\Omega(l_1, l_2)$ is in fact reached in one of the breakpoints $(X, Y) \in \mathcal{Q}$. Thus, the discrete problem would have the same solution as the continuous one, and consequently corner hopping solves these problems.

For a general F satisfying conditions (M) and (S), the following proposition provides an upper bound for the difference between the solutions of the discrete and continuous problems.

Lemma 3 *Let us assume that F satisfies conditions (M) and (S), let $\mathbf{x}^*$ denote a maximum point of (1), and let $\mathbf{x}'$ be the solution obtained by corner hopping. Then, we have*

$$F(l_1(\mathbf{x}^*), l_2(\mathbf{x}^*)) - F(l_1(\mathbf{x}'), l_2(\mathbf{x}')) \leq \frac{K}{8} \max_{1 \leq i \leq n} (|a_i| + |b_i|)^2$$

Proof. Since $\Delta(l_1, l_2) \subset \Omega(l_1, l_2)$, the maximum value of F for the discrete problem does not exceed that for the continuous problem. Therefore, it is sufficient to show that F on the approximate solution is within $D = \frac{K}{8} \max_{1 \leq i \leq n} (|a_i| + |b_i|)^2$ from the maximum of F over $\Omega(l_1, l_2)$.

To simplify notations, let us further assume that the ordering (11) can be attained by the permutation $\pi = (1, 2, ..., n)$, and us denote by $\mathbf{x}^{**} \in [0, 1]^n$ a point where $F(l_1(\mathbf{x}), l_2(\mathbf{x}))$ attains its continuous maximum.

By Lemma 2, we can assume that the point $(l_1(\mathbf{x}^{**}), l_2(\mathbf{x}^{**}))$ is on the boundary of $\Omega(l_1, l_2)$. That point is either a breakpoint (one of Q_js or $\overline{Q}_j$s), or it lies somewhere on a segment between two consecutive breakpoints, for example between Q_{j-1} and Q_j. In the former case, the approximate solution obtained by corner hopping *is* the true solution for the discrete problem. In the latter case, the vector $\mathbf{x}^{**}$ can be represented as $\mathbf{x}^{**} = \mathbf{q}_{j-1} + \lambda \mathbf{d} = \mathbf{q}_j - (1 - \lambda)\mathbf{d}$, where $\mathbf{q}_{j-1}, \mathbf{q}_j$ are as defined in (12), $\mathbf{d} = [0, \dots, 0, 1, 0, \dots, 0]$ is a the jth unit vector, and where $\lambda \in (0, 1)$.

Thus, by introducing $f(t) = F(l_1(\mathbf{q}_{j-1} + ta_j), l_2(\mathbf{q}_{j-1} + tb_j))$ and integrating over the interval $[0, \lambda]$, we obtain

$$F(l_1(\mathbf{x}^{**}), l_2(\mathbf{x}^{**})) = f(\lambda) = F(Q_{j-1}) + \int_0^\lambda f'(t)\, dt, \tag{14}$$

where $f' \equiv df/dt$. By condition (S) and by our choice of λ, $f(t)$ is a smooth function attaining its maximum at $t = \lambda$, thus we have $f'(\lambda) = 0$. Therefore, by applying the identity $\int_0^\lambda f'(t)\, dt = (tf'(t))|_0^\lambda - \int_0^\lambda f''(t)t\, dt = -\int_0^\lambda f''(t)t\, dt$, we can re-write (14) as

$$F(l_1(\mathbf{x}^{**}), l_2(\mathbf{x}^{**})) = F(Q_{j-1}) - \int_0^\lambda f''(t)\, t\, dt,$$

which implies that

$$F(l_1(\mathbf{x}^{**}), l_2(\mathbf{x}^{**})) - F(Q_{j-1}) = \left| \int_0^\lambda f''(t)\, t\, dt \right| \le \max_{t \in [0,\lambda]} |f''(t)| \, \frac{\lambda^2}{2}.$$

By integrating $f(t)$ on $[\lambda, 1]$ in a similar way, we can also show that

$$F(l_1(\mathbf{x}^{**}), l_2(\mathbf{x}^{**})) - F(Q_j) \le \max_{t \in [\lambda,1]} |f''(t)| \, \frac{(1-\lambda)^2}{2}.$$

By combining the two inequalities (namely, choosing the former when $\lambda < 0.5$, and the latter one when $\lambda \ge 0.5$), and taking into account that for the approximate discrete solution $\mathbf{x}'$ we have $F(l_1(\mathbf{x}'), l_2(\mathbf{x}')) \ge F(Q_i)$ for $i = 1, ..., n$, we obtain

$$\begin{aligned} F(l_1(\mathbf{x}^{**}), l_2(\mathbf{x}^{**})) - F(l_1(\mathbf{x}'), l_2(\mathbf{x}')) &\le \tfrac{1}{8} \max_{\substack{1 \le j \le n \\ t \in [0,1]}} |f''(t)| = \\ = \tfrac{1}{8} \max_{\substack{1 \le j \le n \\ (X,Y) \in [Q_{j-1}, Q_j]}} &\left| \tfrac{\partial^2 F(X,Y)}{\partial X^2} a_j^2 + \tfrac{\partial^2 F(X,Y)}{\partial Y^2} b_j^2 + 2 \tfrac{\partial^2 F(X,Y)}{\partial X \partial Y} a_j b_j \right|. \end{aligned} \tag{15}$$

Since the second partial derivatives of F are bounded by K by condition (S), we obtain

$$F(l_1(\mathbf{x}^{**}), l_2(\mathbf{x}^{**})) - F(l_1(\mathbf{x}'), l_2(\mathbf{x}')) \le \frac{K}{8} \max_{1 \le j \le n} (|a_j| + |b_j|)^2,$$

from which the statement follows. □

The following proposition improves the result of Lemma 3 in the case of $F(X, Y) = XY$.

Lemma 4 *If $F(X, Y) = XY$, then the value of $F(l_1(\mathbf{x}'), l_2(\mathbf{x}'))$ on the approximate solution $\mathbf{x}'$ obtained by corner hopping is within $\frac{1}{4} \max_{1 \le i \le n} |a_i b_i|$ from the optimum value of (1).*

Proof. This can be proven in the same manner as Lemma 3, but substituting

$$\partial^2 F(X,Y)/\partial X^2 = 0, \quad \partial^2 F(X,Y)/\partial Y^2 = 0, \quad \partial^2 F(X,Y)/\partial X \partial Y = 1$$

into inequality (15). □

As Lemmas 3 and 4 show, if all we need is to find an approximate solution of the discrete problem with a large enough precision ϵ, then a sufficiently good solution is obtained by corner hopping, in $O(n \log n)$ time. If the desired precision ϵ is smaller than provided by the above lemmas, then we can use the following strategy (Algorithm A):

```
Algorithm A

Input: Reals $a_i$ and $b_i$, $i = 0, 1, ..., n$, a function $F(X,Y)$ (or an oracle for it), together
    with a constant $K$ as in condition (S), and a constant $\epsilon > 0$.

Step 1: Sort the indices $1, ..., n$ as in (11).

Step 2: Set $D_i = \frac{K}{8}(|a_i| + |b_i|)^2$ (or if $F(X,Y) = XY$ set $D_i = \frac{1}{4}|a_i b_i|$) for $i = 1, ..., n$.

Step 3: Set $L = \{i \mid D_i > \epsilon, i = 1, ..., n\}$.

Step 4: For every subset $I \subseteq L$ repeat the following:

    Step 4.1: Fix $x_i = 1$ for $i \in I$ and fix $x_i = 0$ for $i \in L \setminus I$.
    Step 4.2: Find the best approximate solution for the problem in the rest of
        the variables, $x_j$, $j \notin L$ by corner hopping.
    Step 4.3: Update the best solution, if a better one found in the previous step.

Output: Write out the best solution found.
```

Proof of Theorem 2. We claim that Algorithm A provides a desired approximation within the claimed time.

Since the set of feasible points of the continuous relaxation of each restricted problem with $n - |L|$ variables is a subset of $\Omega(l_1, l_2)$, the bounds in (S) with the same value of K still applies to each of the restricted problems. Therefore, the value of F on the approximate solutions obtained by corner hopping is within ϵ from the optimal solutions of the restricted problems by Lemmas 3 or 4. Since the optimum of 1 will be an optimal solution for one of these restrictions (because we check all possible assignments to the variables in L), the obtained approximation will indeed be within ϵ of the optimum.

Let us next analyze the running time of Algorithm A. Steps 1,2, and 3 are executed only once, and need $O(n \log n)$ time. Steps 4.1-4.3 are repeated $2^{|L|}$ times, each time the corner hopping costs $O(n-|L|)$ time, totaling in $O(n2^{|L|})$. Thus, to finish the proof, we need to estimate $|L|$ from above.

In case of a general function F, only indices with $(K/8)(|a_i|+|b_i|)^2 > \epsilon$, i.e, with $|a_i| + |b_i| > \sqrt{8\epsilon/K}$ are selected into L. Thus,

$$A + B \geq \sum_{i \in L} (|a_i| + |b_i|) > |L|\sqrt{8\epsilon/K},$$

and hence

$$|L| < (A+B)\sqrt{\frac{K}{8\epsilon}}.$$

Similarly, for the case of $F(X,Y) \equiv XY$, the set L includes index i only if $|a_i b_i| > 4\epsilon$. Since $(|a|+|b|)^2 \geq 4|a||b|$, this implies that for each $i \in L$, $|a_i/A| + |b_i/B| > 4\sqrt{\epsilon/AB}$, and hence

$$|L| < \frac{1}{2}\sqrt{\frac{AB}{\epsilon}}.$$

□

Proof of Corollary 3. Given a measure $\zeta(P_{\mathbf{x}}) = c + l_1(\mathbf{x})l_2(\mathbf{x})$, where l_1 and l_2 are as in Section 1, we have $A = \sum_{i=1}^n a_i = 1$ and $B = \sum_{i=1}^n b_i = 1$. Thus for any given $\epsilon' > 0$, Algorithm A requires $O(n \log n + 2^{1/\sqrt{\epsilon'}}n)$ operations to find a disjunction $P_{\mathbf{x}'}$ whose distinguishing power is within ϵ' from that of the best possible disjunction $P_{\mathbf{x}^*}$ for this problem, by Theorem 2.

Let us also note that if the optimum value $\zeta(P_{\mathbf{x}^*}) < c$, then either $l_(\mathbf{x}^*) < 0$ and $l_2(\mathbf{x}^*) > 0$, or $l_1(\mathbf{x}^*) > 0$ and $l_2(\mathbf{x}^*) < 0$. In either case, by Lemma 3.9 of [18], we have $(l_1(\mathbf{x}^*), l_2(\mathbf{x}^*) \in \mathcal{Q}$, and thus corner hopping in Algorithm A actually finds $\mathbf{x}' = \mathbf{x}^*$.

On the other hand, if $\zeta(P_{\mathbf{x}^*}) \geq c$, then we have

$$\frac{\zeta(P_{\mathbf{x}^*}) - \zeta(P_{\mathbf{x}'})}{\zeta(P_{\mathbf{x}^*})} \leq \frac{\zeta(P_{\mathbf{x}^*}) - \zeta(P_{\mathbf{x}'})}{c} \leq \frac{\epsilon'}{c}$$

by Theorem 2. Therefore, choosing $\epsilon' = c\epsilon$, the statement follows. □

7 Measuring the power of a set of predicates

Before we proceed, in the next Section (Section 8), to discussing the disjunction-selection effectiveness on a specific classification problem with multiple categorical variables, we need to introduce a way to measure usefulness not just of a single predicate, but of a set of predicates $\mathcal{P}$. Having such a measure will help us to evaluate the effectiveness of the predicates, selected with the help of algorithms, introduced in the previous sections.

The eventual goal of creating a set of predicates is to be able to combine these predicates, using conjunction and negation, into rules that can reliably distinguish positive and negative examples. Without going into the details of rule design, it is obvious that if we are to construct, based on a set of predicates $\mathcal{P}$, a rule capable of distinguishing examples $a \in D^+$ and $b \in D^-$, at least one predicate $P \in \mathcal{P}$ must distinguish a and b. Having several rules in $\mathcal{P}$ distinguishing a and b is probably better than having just one such rule, as it provides more options for designing rules.

With this rationale in mind, we can generalize the single-predicate distinguishing power measure $\zeta_{quad}(P)$ from (3) into a the so-called *discounted Hamming distance*. This is a measure of distinguishing power of a set of predicates $\mathcal{P}$, or in other words of how well these predicates distinguish positive examples of D^+ from the negative examples of D^-, and it is defined as

$$d_M(\alpha, \mathcal{P}, D^+, D^-) = \sum_{i=0}^{M-1} a_i(\mathcal{P}, D^+, D^-)\alpha^i + \left(\sum_{i \geq M} a_i(\mathcal{P}, D^+, D^-) \right) \alpha^M. \tag{16}$$

Here M is a fixed positive integer, α is the so-called *discount factor* (typically, chosen as a small positive number, $0 < \alpha < 1$; we use $\alpha = 0.1$ in all experiments presented here), and $a_i(\mathcal{P}, D^+, D^-)$ is the number of pairs $(a, b) \in D^+ \times D^-$ on which exactly i predicates from $\mathcal{P}$ return different values:

$$a_i(\mathcal{P}, D^+, D^-) = |\{(a, b) \in D^+ \times D^- : \delta((\mathcal{P}, a, b) = i\}|, \tag{17}$$

where

$$\delta(\mathcal{P}, a, b) = |\{P \in \mathcal{P} : P(a) \neq P(b)\}|. \tag{18}$$

As one can see from the above definitions, each (a, b) pair which is not distinguished by any predicate from $\mathcal{P}$ (and therefore, cannot be distinguished by any rule constructed from the set $\mathcal{P}$) yields the highest contribution to measure d; contributions from pairs that are distinguished by only one predicate from $\mathcal{P}$ are discounted by factor α; contributions from pairs that are distinguished by exactly two predicates from $\mathcal{P}$ are discounted by α^2, etc. Finally, distinguishing more than M-fold a pair (a, b) does not contribute to this measure.

Thus, the better predicates from $\mathcal{P}$ are capable of distinguishing elements of D^+ from those of D^-, the smaller is $d_M(\alpha, \mathcal{P}, D^+, D^-)$. If $\mathcal{P}$ contains no useful predicates at all, then $d_M(\alpha, \mathcal{P}, D^+, D^-) = |D^+||D^-|$, while if each (a, b) pair is distinguished by $K \leq M$ or more predicates, then $d_M(\alpha, \mathcal{P}, D^+, D^-) \leq |D^+||D^-|\alpha^K$. Although the summation in (16) is labelled as going for $0 \leq i < M$, in reality the pair counts a_i may be non-negative only for $i \leq |\mathcal{P}|$, and hence $M \leq |\mathcal{P}|$ can be always be assumed.

We can use the discounted Hamming distance to select a short list of "most useful" predicates out of a longer predicate list, as follows:

```
Depth-$M$ HDV

Input: The data set $D = (D^+, D^-)$, an integer $M \geq 0$, a real $0 < \alpha < 1$, and a set
    of predicates $\mathcal{P}$.

Step 1: Set $k = 1$.

Step 2: Choose $P_k \in \mathcal{P} \setminus \{P_1, ..., P_{k-1}\}$ for which
    $d(M, \alpha, \{P_1, P_2, \ldots, P_{k-1}, P_k\}, D^+, D^-)$ is the smallest.

Step 3: If $d(M, \alpha, \{P_1, P_2, \ldots, P_k\}, D^+, D^-) > 0$ and $\mathcal{P} \neq \{P_1, P_2, \ldots, P_k\}$, increment
    $k$ and go back to Step 2.

Output: The set $\{P_1, ..., P_k\}$.
```

This algorithm was found to be quite competitive with many other feature selection methods in [10]. In fact this algorithm in case of $M = 1$ coincides with the standard greedy method, used in set covering based feature selection (see e.g., [2, 9]).

8 Some experimental results

To demonstrate the binarization of categorical variables, introduced in the first part of this paper, we needed a data set which has many categorical variables, and several of those with many different values. There are only a few such data sets available in the standard machine learning depositories, among them perhaps the so called *mushroom* data set from the machine learning depository of the University of California at Irvine is the richest in categorical attributes (see [4]).

The *mushroom* data set consists of a set of 8124 records describing various properties of mushroom species. Each record describes one mushroom species, containing values for several of its categorical properties. There were 22 categorical attributes overall, each with its own range of values (4 to 10 values per property). The sum of the sizes of these ranges amounted to 117, corresponding thus to 117 basic predicates. Each species is classified as ”poisonous” or ”edible”, and the objective is to distinguish these two classes.

We divided the entire data set into a training set and a test set in different ways. In one series of experiments (the “20-80 split”), the mushroom set was split randomly into the training set $A_T \cup A_F$ including 1611 records ($\approx 20\%$ of the total), and the test set $B_T \cup B_F$ including the other 6513 records ($\approx 80\%$ of the total).

In the other series of experiments (“cross-validation”), the data set was partitioned randomly into five approximately equal-sized parts numbered S_0 to S_4. Then, for each particular type of computation we would perform $\binom{5}{2} = 10$ runs, one run for every possible pair (i, j), $0 \leq i < j \leq 4$. In each such run, the set $A_{ij} = S_i \cup S_j$, containing $\approx 40\%$ of the records, would be used as the training set, while the remaining set $B_{ij} = S \backslash A_{ij}$, with the other $\approx 60\%$ of the records, would constitute the test set.

In all experiments, one (or none) of the predicate selection methods, described in the first part of this paper, was used. A "master set" of predicates $\mathcal{P}^*$ was produced by including all 117 basic predicates, and additionally some of the best disjunctions obtained with the chosen method for each categorical property. In the tables and graphs that follow, runs are labelled according to how, if at all, the disjunctions were selected in this procedure, as in the following table:

Label	Disjunctions selection
None	No disjunctions (only the basic predicates).
Lin	Best disjunctions selected using the linear formula (4)
Odds	Best disjunctions selected using the ratio formula (5)
Quad2	Best disjunctions selected using the bilinear formula (3), with the exact algorithm (Section 4)
Quad	Best disjunctions selected using the bilinear formula (3), with the approximate algorithm (Section 6) with $\epsilon = 0.01$

For “Lin”, we generated only one best disjunction per categorical property. With “Odds”, “Quad2”, and “Quad”, we selected, for each attribute, several best disjunctions out of those that were encountered during the optimization process. (That is, for “Odds” and “Quad”, the best of the disjunctions corresponding to the corners of the (X, Y) feasibility domain; for “Quad2”, the best of the disjunctions corresponding to $z(n, w)$ and $Z(n, w)$ for all w). In each of the last three methods, the list of “best disjunctions” for each property was truncated to never be longer than the number of values that the property could assume (thus, the total number of added disjunctions could never be greater than the number of basic predicates); finally trivial (one-component) disjunctions were excluded, since they are already included as basic predicates.

In each of the runs the master set $\mathcal{P}^*$ of predicates was narrowed down by depth-M HDV, for some value of M, to a smaller set $\mathcal{P}$ of predicates. In Table 1 we tabulated the obtained set of predicates for some of the methods, using $\approx 20\%$ of the data as training.

For most runs we used $M = 2$ for the HDV procedure. However, in some experiments, specifically indicated below as "infinite depth" runs, we used $M \geq |\mathcal{P}^*|$ to select the cut set.

In fact HDV is a sequential process, and thus $\mathcal{P} = \{P_1, P_2, ..., P_l\}$ is an ordered set, in which predicate P_1 was found to provide the highest level of separation on the training set, and after that P_2 proved to be the most distinguishing one, according to the discounted Hamming measure d_M, etc. In case $M \geq |\mathcal{P}^*|$, we artificially limited $|\mathcal{P}|$ to the top 15 predicates produced by HDV.

To evaluate the efficiency of the proposed procedures, we measured the distinguishing power of $\mathcal{P}$ on the training set itself, and which is perhaps more interesting, on the test set, as well. Since $\mathcal{P}$ is an ordered list, we actually measured the distinguishing power of every prefix of this list. In other words, for the set $\mathcal{P} = \{P_1, P_2, \ldots, P_m\}$ we computed

$$D(k) = d_{m+1}(\alpha, \{P_1, P_2, \ldots, P_k\}, D^+, D^-)$$

No disjunctions	Linear $F = \lvert X - Y \rvert$	Quad2 $F = X + Y - 2XY$ (exact algorithm)	Quad $F = X + Y - 2XY$ (approximation)
$\lvert\mathcal{P}^*\rvert = 117$ $\lvert\mathcal{P}\rvert = 12$	$\lvert\mathcal{P}^*\rvert = 132$ $\lvert\mathcal{P}\rvert = 11$	$\lvert\mathcal{P}^*\rvert = 209$ $\lvert\mathcal{P}\rvert = 11$	$\lvert\mathcal{P}^*\rvert = 154$ $\lvert\mathcal{P}\rvert = 10$
odor = n	odor $\in \{\mathrm{a}, \mathrm{l}, \mathrm{n}\}$	odor $\in \{\mathrm{a}, \mathrm{l}, \mathrm{n}\}$	odor $\in \{\mathrm{a}, \mathrm{l}, \mathrm{n}\}$
bruises = t	spore-print-color $\in \{\mathrm{h}, \mathrm{r}, \mathrm{w}\}$	spore-print-color $\in \{\mathrm{h}, \mathrm{r}, \mathrm{w}\}$	spore-print-color $\in \{\mathrm{h}, \mathrm{r}, \mathrm{w}\}$
population = v	odor = n	odor $\in \{\mathrm{a}, \mathrm{n}\}$	odor = n
stalk-root = b	population $\in \{\mathrm{v}\}$	population = v	population = v
gill-size = b	gill-size = b	gill-size = b	gill-size = b
habitat = d	spore-print-color = r	spore-print-color = r	spore-print-color $\in \{\mathrm{k}, \mathrm{n}, \mathrm{w}\}$
stalk-shape = e	gill-spacing = w	stalk-root $\in \{\mathrm{e}, \mathrm{missing}\}$	stalk-root $\in \{\mathrm{e}, \mathrm{missing}\}$
gill-size = n	gill-color = w	cap-color $\in \{\mathrm{c}, \mathrm{g}, \mathrm{n}, \mathrm{p}, \mathrm{r}\}$	bruises = f
spore-print-color = w	bruises = f	bruises = t	habitat $\in \{\mathrm{d}, \mathrm{p}\}$
gill-color = w	stalk-surface-below-ring = y	spore-print-color = w	stalk-color-below-ring = y
habitat = p	spore-print-color = w	stalk-surface-below-ring = y	
cap-shape = f			

Table 1: The set $\mathcal{P}$ of predicates produced by depth-2 HDV starting with on various master sets $\mathcal{P}^*$, using 20% of the data as training set. The first column corresponds to the basic predicate set (only single-comparison predicates). In other columns, the predicate set $\mathcal{P}^*$ is combined of the same basic predicates, plus a number of the best disjunctions of such basic predicated found using the method indicated, as described in the text.

for each $0 \leq k \leq m$, and presented the graph $D(k)$ as a function of k (where D^+ and D^- denote the set of positive and negative examples in the actual set, used). How fast the graph of $D(k)$ decreases as k increases illustrates how much additional distinguishing power is brought in by additional elements of the predicate list.

Series 1: 20-80 split, $M = 2$: We produced a set of predicates $\mathcal{P}$ on a training set of $\approx 20\%$ of the data set using HDV with depth $M = 2$. Then measured the distinguishing power $D(k)$ of the sets, with depth $M \geq |\mathcal{P}|$, both on the training set and on the test set. As Figure 2 shows, for each disjunction-generation method the curves of $D(k)/D(0)$ for the training set and the test set appear very close together. This means that on this problem the $\approx 20\%$ training set is quite sufficient to generate predicates that are just as effective on the test set as they are on the training set.

Figure 2 also shows that any of the predicate sets that include some disjunctions of basic predicates is superior to the case only including the basic predicates (labels with 'None').

Predicate sets that include disjunctions selected based on the optimization of the bilinear function (3) (labelled 'Quad' and 'Quad2') result in better separation than those using disjunctions selected with the optimization of a linear or odds-based measure. This is not

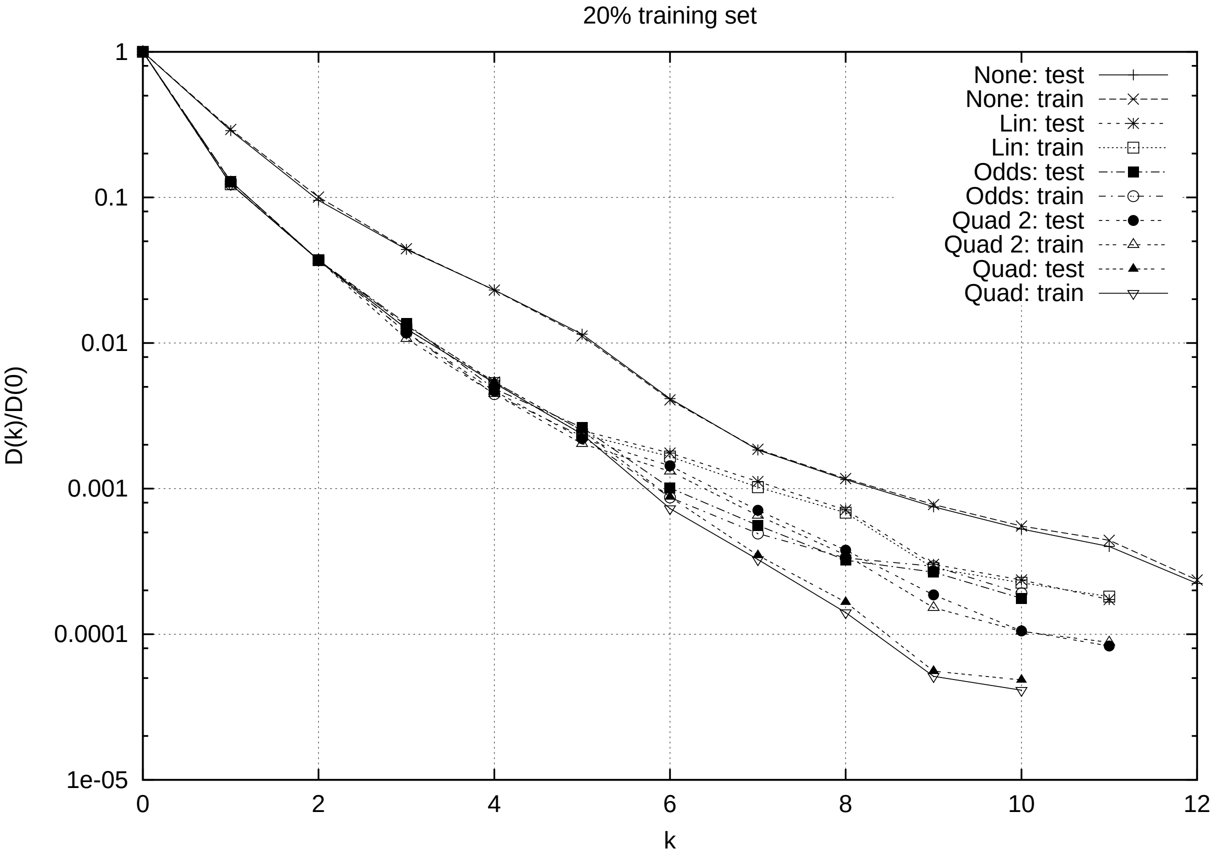


Figure 2: Predicate selection by HDV($M = 2$) using $\approx 20\%$ of the data as training, and the rest as test. Distinguishing power of the obtained predicates measured both on the training and the test sets.

surprising on the training set, since the discounted Hamming distance measure (16) is directly connected to the single-predicate power measure (3). However, the same behavior on the test sets is somewhat surprising, since it cannot be connected directly to the way those predicates were selected – since for the selection process we did not use the test records.

The difference between the predicate sets based on exact optimization of (3) (labelled 'Quad2') and on approximate optimization (labelled 'Quad') is most likely due not to the best selected disjunction (it is the same in both cases, since the ϵ in the approximate algorithm is small enough to guarantee the exact maximum), but to the composition of the list of "next-best" disjunctions, which is different for these two methods.

Series 2: Cross-validation (40-60 splits), $M = 2$: In these experiments, the data set was divided into a $\approx 40\%$ training set and $\approx 60\%$ test set in $\binom{5}{2} = 10$ different ways, as we described above. For each version, $\mathcal{P}$ was generated with HDV($M = 2$), and then its distinguishing power was measured on the test set using the discounted Hamming distance measure with $M = |\mathcal{P}| + 1$. The results are averaged over the 10 runs, and displayed in Figure 3.

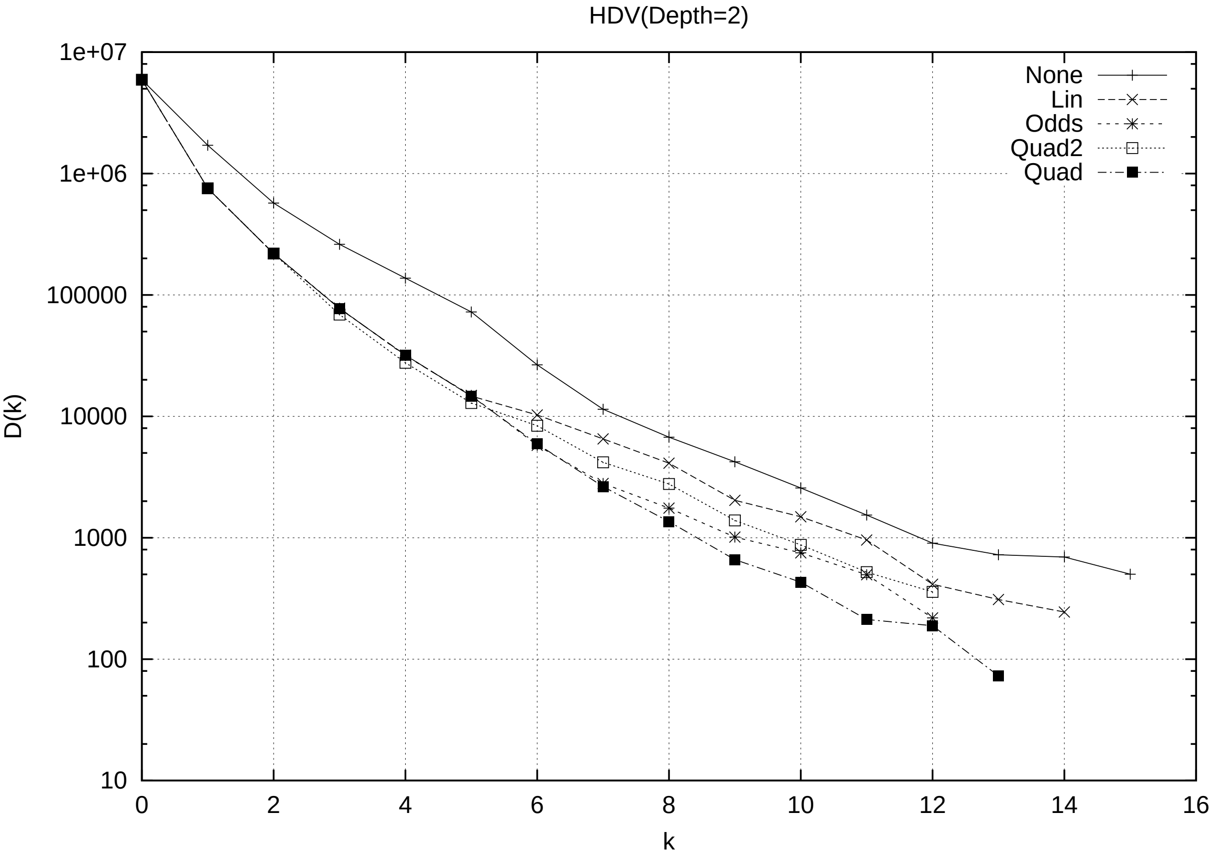


Figure 3: Cross-validation (Series 2). Averaged D(k) for ten different 40-60 splits, predicate selection by HDV($M = 2$).

Series 3: Cross-validation (40-60 splits), $M > |\mathcal{P}|$: These series of experiments was identical to Series 2, but $M > |\mathcal{P}|$ is used in the HDV predicate selection algorithm. Results over the runs are averaged, and displayed in Figure 4.

The above results appear consistent between each other in that any predicate set including well-selected disjunctions is superior to one consisting only the basic predicates. Furthermore, among the predicate generation approaches, “Lin” seems the weakest, and ”Quad2” seems to be the best. Perhaps the advantage of ”Quad2” to the others comes from the fact that it not only finds a best predicate, but also generates several ”good” ones, corresponding to the vertices of $\Omega(l_1, l_2)$, and thus providing a richer set for the HDV selection procedure.

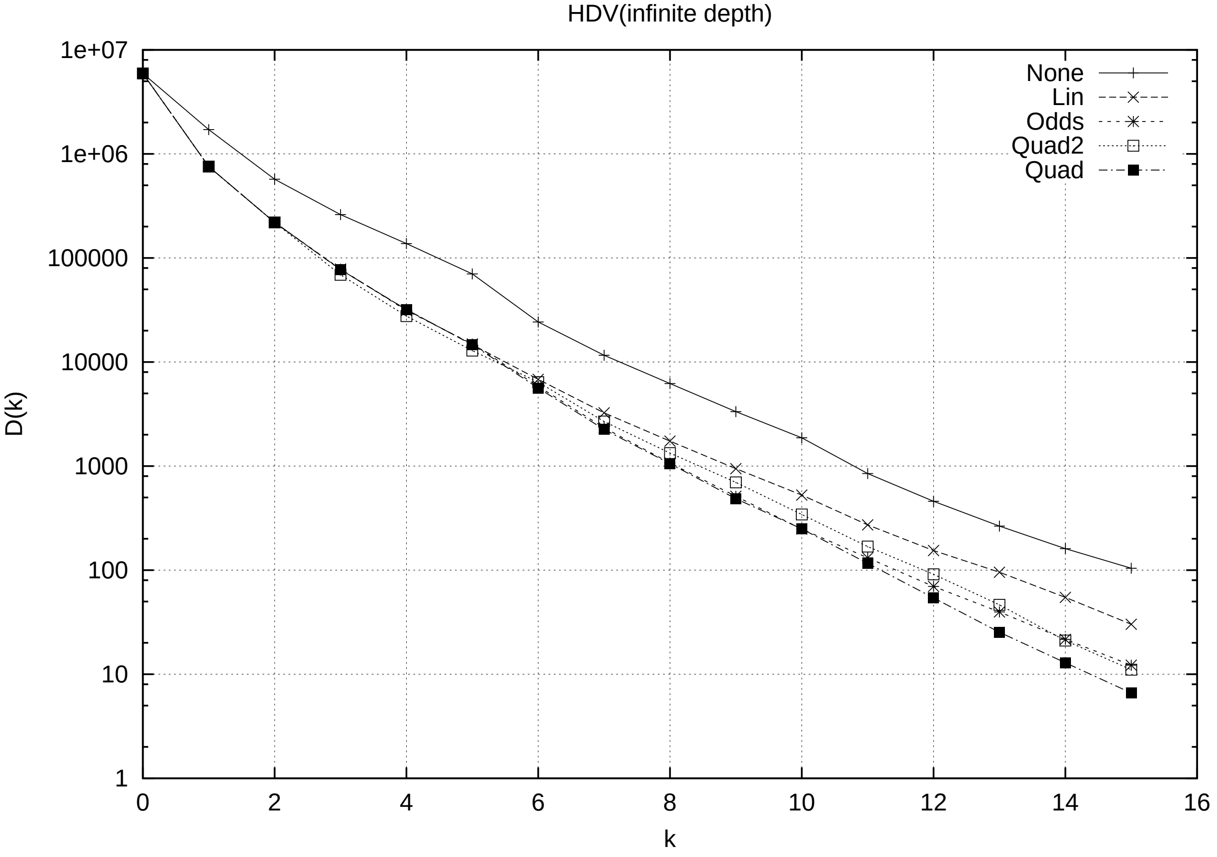


Figure 4: Cross-validation (Series 3). Averaged D(k) for ten different 40-60 splits, cut set generation by HDV($M = \infty$).

One important question left is, how much do results differ when different training sets are selected? Figure 5 shows this for two different methods of generating the master cut set ("None" and "Quad"), and cut set selection with HDV($M = 2$) in Series 2 experiments. While variation between different cut sets is noticeable, it is much smaller than the difference between the $D(k)$ values with and without disjunctions.

What causes the variation between the power of cut sets obtained on different training sets? Our hypothesis is that it was not so much the difference in the disjunction sets on different training sets (the difference between lists was typically under 10% of disjunctions), but the fact that the HDV algorithm used for selecting the cut set from the master cut set was optimizing for a somewhat different measure than the one we would eventually measure (i.e., depth-2 discounted Hamming distance vs. infinite-depth discounted Hamming distance). As a result, HDV($M = 2$) on different training sets may select cut sets that are quite similar in its own (depth-2) terms, but are more different in terms of the eventual measure. A good way to test the hypothesis is too look at the results obtained in Series 3 experiments, where infinite-depth discounted Hamming distance was used both in cut set selection and in measuring the power of resulting lists on the training set. These results are presented in the graphs on Figure 6, and show that the variability of $D(k)$ among different training sets in this series is much smaller than in Series 2; this appears to agree well with our hypothesis.

References

[1] A. Agresti. *Analysis of ordinal data.* Wiley, New York, 1984.

[2] H. Almuallim and T. Dietterich. Efficient algorithms for identifying relevant features. In *Proceedings of the Ninth Canadian Conference on Artificial Intelligence*, pages 38–45, Vancouver, BC, 1992. Morgan Kaufmann.

[3] D.A. Bell and H. Wang. A formalism for relevance and its application in feature subset selection. *Machine Learning*, 41:175–195, 2000.

[4] C.L. Blake and C.J. Merz. UCI repository of machine learning databases, 1998.

[5] A. Blum and P. Langley. Selection of relevant features and examples in machine learning. *Artificial Intelligence*, 67:245–285, 1997.

[6] A. Blumer, A. Ehrenfeucht, D. Haussler, and M. K. Warmuth. Occam's razor. *Information Processing Letters*, 24:377–380, 1987.

[7] E. Boros and P. L. Hammer. Pseudo-boolean optimization. *Discrete Applied Mathematics*, 123:155–225, 2002.

[8] E. Boros, P.L. Hammer, T. Ibaraki, and A. Kogan. Logical analysis of numerical data. *Mathematical Programming*, 79:163–190, August 1997.

[9] E. Boros, P.L. Hammer, T. Ibaraki, A. Kogan, E. Mayoraz, and I. Muchnik. An implementation of logical analysis of data. *IEEE Transactions on Knowledge and Data Engineering*, 12(2):292–306, May 2000.

[10] E. Boros, T. Horiyama, T. Ibaraki, K. Makino, and M. Yagiura. Finding essential attributes from binary data. *Annals of Mathematics and Artificial Intelligence*, 2003. To appear.

[11] P.S. Bradley, O.L. Mangasarian, and W.N. Street. Feature selection via mathematical programming. *INFORMS Journal on Computing*, 10:209–217, 1998.

[12] W. Brauer and M Scherf. Feature selection by means of a feature weighting approach. Technical Report FKI-221-97, Institute für Informatik, Technische Universität München, 1997.

[13] L. Breiman, J.H. Friedman, R.A. Olshen, and C.J. Stone. *Classification and Regression Trees*. Wadsworth International Group, 1984.

[14] R. Caruana and D. Freitag. Greedy attribute selection. In *Machine Learning: Proceedings of the Eleventh International Conference*, pages 28–36, Rutgers University, New Brunswick, NJ, 1994.

[15] G.B. Dantzig. Discrete variable extremum problems. *Operations Research*, 5:266–277, 1957.

[16] R. Kohavi G. John and K. Pfleger. Irrelevant features and the subset selection problem. In *Machine Learning: Proceedings of the Eleventh International Conference*, pages 121–129. Morgan Kaufmann, 1994.

[17] M.A. Hall and L.A. Smith. Practical feature subset selection for machine learning. In *Proceedings of the 21st Australasian Computer Science Conference*, pages 181–191. Springer Verlag, 1998.

[18] P. Hammer, P. Hansen, P. Pardalos, and D. Rader. Maximizing the product of two linear functions in 0-1 variables. RUTCOR Research Report 2-1997, Rutgers University, 640 Bartholomew Road, Piscataway, NJ 08854-8003, USA, February 1997.

[19] P.L. Hammer and S. Rudeanu. *Boolean Methods in Operations Research and Related Areas*. Springer-Verlag, Berlin, Heidelberg, New York, 1968.

[20] P. Hansen, M. Poggi de Aragão, and C.C. Ribeiro. Boolean query optimization and the 0-1 hyperbolic sum problem. *Annals of Mathematics and Artificial Intelligence*, 1:97–109, 1990.

[21] D.R. Johnson and J.C. Creech. Ordinal measures in multiple indicator models: A simulation study of categorization error. *American Sociological Review*, 48:398–407, 1983.

[22] K. Kira and L. Rendell. The feature selection problem: Traditional methods and a new algorithm. In *Proceedings of the Tenth National Conference on Artificial Intelligence*, pages 129–134, Menlo Park, 1992. AAAI Press/The MIT Press.

[23] D. Koller and M. Sahami. Toward optimal feature selection. In *ICML-96: Proceedings of the Thirtenth International Conference on Machine Learning*, pages 284–292. Morgan Kaufmann, 1997.

[24] J.S. Long. *Regression models for categorical and limited dependent variables*. Sage, Thousand Oaks, CA, 1997.

[25] R.X. MacCallum, S. Zhang, K.J. Preacher, and D.D. Rucker. On the practice of dichotomization of quantitative variables. *Psychological Methods*, 7:19–40, 2002.

[26] J.R. Quinlan. Induction of decision trees. *Machine Learning*, 1:81–106, 1986.

[27] J.R. Quinlan. *C4.5: Programs for Machine Learning.* Morgan Kaufmann, 1992.

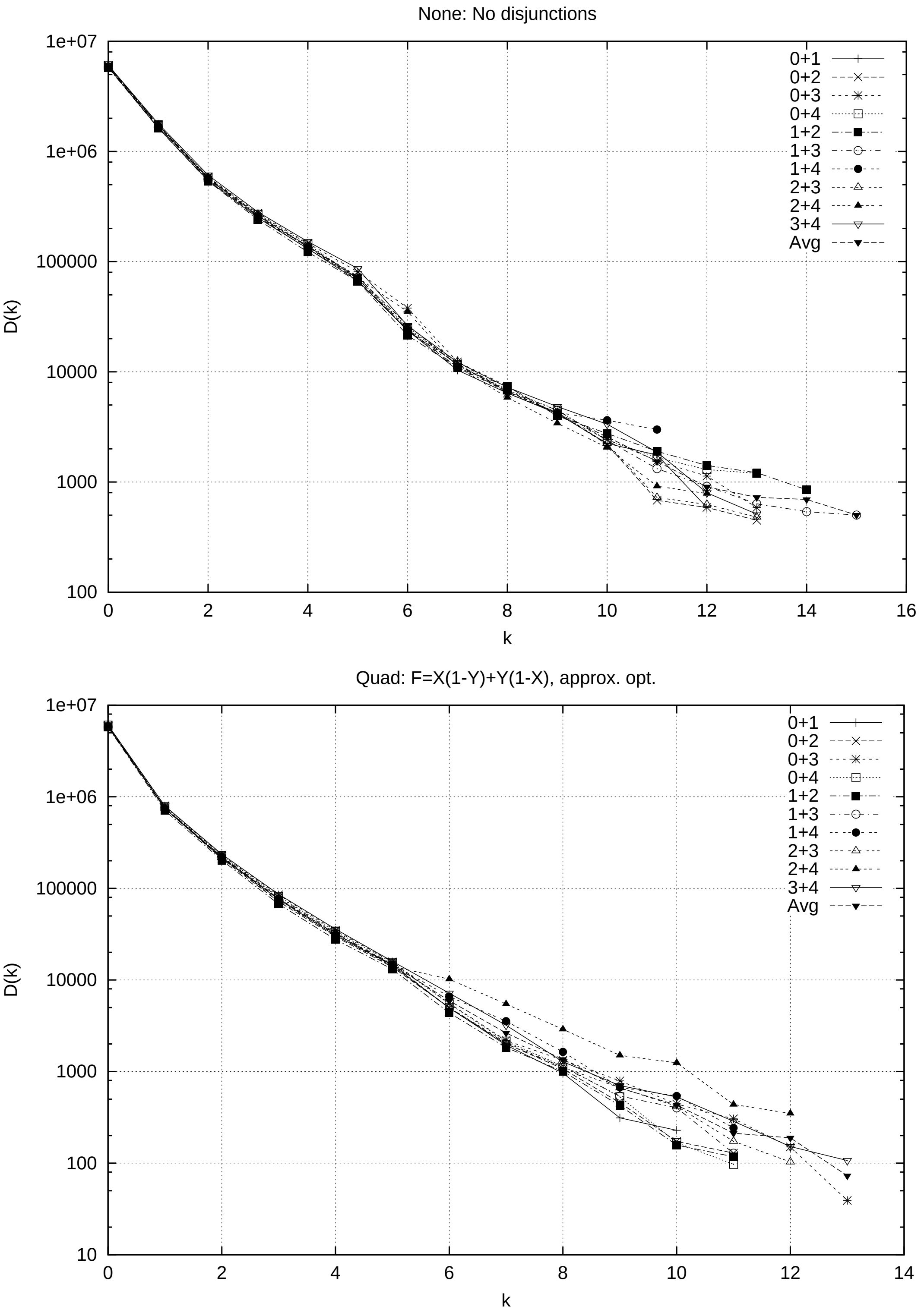
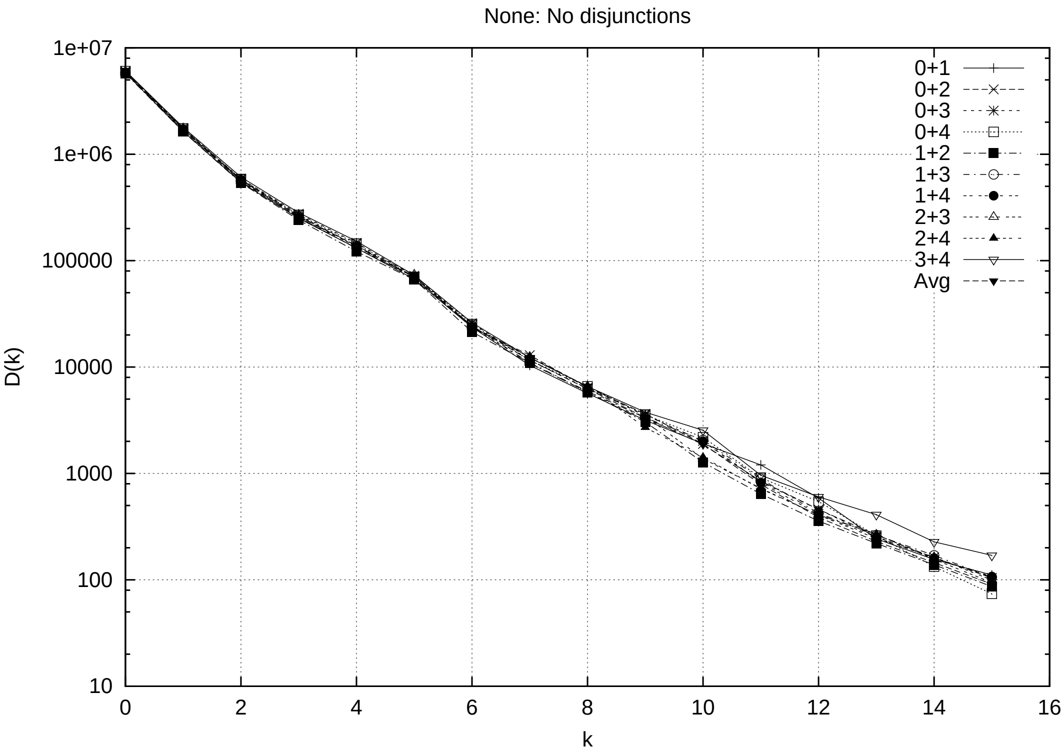


Figure 5: Cross-validation (Series 2). Ten different 40-60 splits, cut set generation by HDV($M = 2$). Each graph compares the distinguishing power of the cut sets generated on ten different training sets.



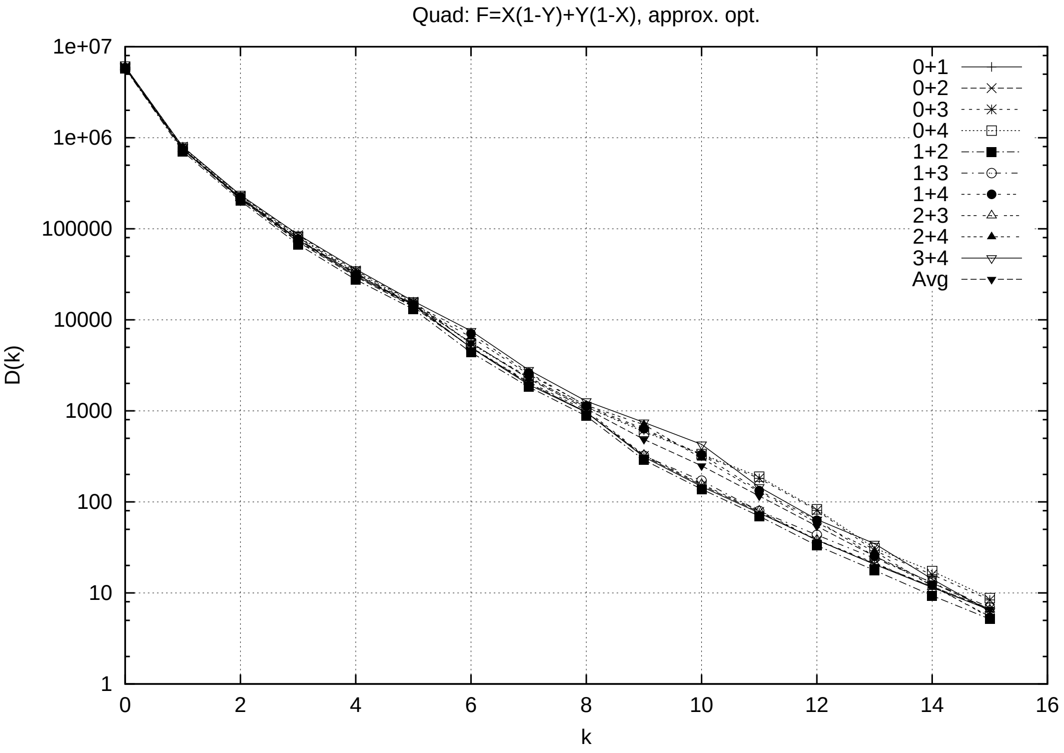


Figure 6: Cross-validation (Series 3). Ten different 40-60 splits, cut set generation by HDV($M = \infty$). Each graph compares the distinguishing power of the cut sets generated on ten different training sets.