Linear Programming Computational Procedures for Ordinal Regression

V. SRINIVASAN

Stanford University, Stanford, California

ABSTRACT The ordinal regression problem is an extension to the standard multiple regression problem in terms of assuming only ordinal properties for the dependent variable (rank order of preferred brands in a product class, academic ranks for students in a class, etc.) while retaining the interval scale assumption for independent (or predictor) variables. The linear programming formulation for obtaining the regression weights for ordinal regression, developed in an earlier paper, is outlined and computational improvements and alternatives which utilize the special structure of this linear program are developed and compared for their computational efficiency and storage requirements. A procedure which solves the dual of the original linear programming formulation by the dual simplex method with upper bounded variables, in addition to utilizing the special structure of the constraint matrix from the point of view of storage and computation, performs the best in terms of both computational efficiency and storage requirements. Using this special procedure, problems with 100 observations and 4 independent variables take less than ½ minute, on an average, on the IBM 360/67. Results also show that the linear programming solution procedure for ordinal regression is valid—the correlation coefficient between "true" and predicted values for the dependent variable was greater than .9 for most of the problems tested.

KEY WORDS AND PHRASES ordinal multiple regression, nonmetric methods, linear programming, computer code

CR CATEGORIES 4.40, 5.41, 5.5

1. *Introduction*

The well-known multiple regression problem [5, 7] consists of estimating regression weights $\{w_1, w_2, \cdots, w_t\}$ for t predictors (independent variables, attributes) from n observations ($t < n$). For the jth observation, if we denote by α_j the value for the dependent variable and by $y_{j1}, y_{j2}, \cdots, y_{jt}$ the values for the t predictors, the multiple regression model can be stated as:

$$\alpha_j = w_0 + w_1 y_{j1} + w_2 y_{j2} + \cdots + w_t y_{jt} + e_j, \tag{1}$$

where e_j denotes the error term for the jth observation. The weights are usually estimated by either MSSE (Minimizing Sum of Squared Errors) [5, 7] or MSAE (Minimizing Sum of Absolute Errors) procedures [3, p. 334].

One of the crucial assumptions involved in such estimation procedures is the requirement that *both* the dependent and predictor variables be measured on interval scales. However, guaranteeing interval scale measurement is very difficult for dependent variables such as "preference" for different brands in a product class [8, 11], "attitude" toward (or "satisfactions" between) alternate mass transit configurations [10], (academic) ranks for students in a class [2], etc. Such measurement problems are considerably alleviated by *ordinal multiple regression*, which requires only the

Author's address: Graduate School of Business, Stanford University, Stanford, CA 94305

predictor variables to be "intervally" scaled but permits the dependent variable to be "ordinally" scaled. By a logic similar to that in [9], it can be shown that if n is sufficiently large compared to t, the inequalities implied by the ordinal data are sufficient to provide (approximately) ratio scaled estimates for the weights.

A procedure suggested by Carroll [1] makes it possible to extend the conventional MSSE regression procedure to the situation where the dependent variable is known only on an ordinal scale His iterative algorithm involves performing the MSSE regression on successive monotone transformations [6] of the dependent variable so as to improve the fit. However, the linear programming procedure provided by Srinivasan and Shocker [14] for estimating ordinal regression weights possesses some important advantages over the former procedure. First, the weights can be constrained as to sign or left unconstrained, as desired. Second, the procedure can be shown to always yield estimates of weights which globally minimize a "poorness of fit" measure, although it is not clear whether such a result will always hold for Carroll's procedure. One disadvantage, however, of both these procedures in contrast to MSSE regression is the lack of statistical tests of significance for the estimated weights. It is to be hoped that the pragmatic validity of these approaches will be sufficient to encourage eventual development of such tests.

The present paper utilizes the special structure of the linear programming method [14] mentioned earlier to develop computationally efficient procedures for ordinal regression. The basic linear programming framework is briefly reviewed in Section 2. Section 3 develops the several computational improvements and alternatives which are compared in Section 4 for their relative computational efficiency. Section 4 also provides some simulation results which demonstrate that the ordinal regression model together with the linear programming estimation procedure is valid in terms of its predictive power, defined in terms of the correlation coefficient between predicted and "true" values for the dependent variable.

2. *A Review of the Linear Programming Procedure for Ordinal Regression*

2.1 PRELIMINARIES. Without loss of generality, let us relabel, if necessary, the observations $\{1,2,\cdots,n\}$ to be in the same order as that of the dependent variable. If the dependent variable is expressed on an ordinal scale, this means that the observations are to be rearranged, if necessary, so that $j=1$ corresponds to the observation with the largest value for the dependent variable, $j=2$ the second largest, $\cdots$, $j=n$ the observation with the smallest value for the dependent variable. We assume that ties, if any, are broken randomly so as to yield a strict rank order.[1] The ordering of the observations $\{1,2,\cdots,n\}$ implies the set of $N = n(n-1)/2$ paired comparison dominance judgments:

$$\Omega = \{(1,2),(1,3),\cdots,(1,n),(2,3),\cdots,(2,n),\cdots,(n-1,n)\}, \tag{2}$$

where for each pair (j, k), observation j has its value for the dependent variable greater than or equal to that for k. Given a set of estimates $\{w_1,w_2,\cdots,w_t\}$, we can determine

$$s_j = w_1y_{j1} + w_2y_{j2} + \cdots + w_ty_{jt} \quad \text{for} \quad j = 1,2,\cdots,n, \tag{3}$$

the predicted values for the dependent variable. The above equation together with the paired comparison judgments (2) imply that the estimated weights should be such that violations (if any) of the inequalities (4) below should be "as minimal as possible":

$$s_j \geq s_k \quad \text{for } (j,k) \in \Omega. \tag{4}$$

[1] The procedures discussed in this paper can be modified to handle ties explicitly by a method similar to that discussed in [13, Sec 3(f)], but this would involve a significant increase in the computational effort

It is assumed that the estimated weights $\{w_p\}$ are required to be nonnegative in all their components, i.e.

$$w_p \geq 0 \quad \text{for } p = 1,2,\cdots,t. \tag{5}$$

This, however, involves no loss of generality since an ordinal regression problem which does not have this requirement can be reformulated as a problem satisfying (5), but with at most one additional predictor variable.[2]

2.2. PROCEDURE LP1. The linear programming formulation for estimating the regression weights along with its intuitive justification is given below. For greater details the reader should refer to its original development [14]. The reader is assumed to be fully familiar with linear programming terminology [3, 4, 12].

$$\text{Minimize} \quad \sum_{(j,k)\in\Omega} z_{jk} = B \tag{6}$$

subject to

$$\sum_{p=1}^{t} (y_{jp} - y_{kp})w_p + z_{jk} \geq 0 \quad \text{for } (j, k) \in \Omega, \tag{7}$$

$$\sum_{p=1}^{t} \left[\sum_{(j,k)\in\Omega} (y_{jp} - y_{kp}) \right] w_p = 1, \tag{8}$$

$$w_p \geq 0 \quad \text{for } p = 1,2,\cdots,t, \tag{9}$$

$$z_{jk} \geq 0 \quad \text{for } (j, k) \in \Omega. \tag{10}$$

In the linear program (6)–(10), the coefficients $\{y_{jp}\}$ are the known values for the predictor variables. As explained earlier, the set Ω is defined by the rank order of the observations based on the dependent variable. The regression weights $\{w_p\}$ are the "decision variables" in the linear program. The variables z_{jk} may be interpreted as a measure of the *poorness of fit* for pair (j, k) associated with the "solution" $\{w_p\}$. To see this, we rewrite eqs. (7) and (10) as

$$z_{jk} \geq \max\{0, -\sum_{p=1}^{t} w_p(y_{jp} - y_{kp})\}. \tag{11}$$

Since the objective (6) minimizes the sum of z_{jk}, the inequalities (11) will be "tight" at the optimum so that from (3) and (11) we get:

$$z_{jk} = \max\{0, -(s_j - s_k)\}. \tag{12}$$

If $s_j \geq s_k$, then from (12) we obtain $z_{jk} = 0$, so that in accord with (4) there is no poorness of fit for this pair. However, if the solution $\{w_p\}$ leads to $s_j < s_k$, this violates (4) and the quantity $-(s_j - s_k)$ may be defined as the poorness of fit for the pair (j, k). Consequently objective (6) minimizes the total poorness of fit, i e. summed over all pairs in Ω. Equation (8) is a normalization constraint to take into account the fact that the weights $\{w_p\}$ are determined only to a scalar multiple. It avoids the trivial solution $w_p = 0$ for all p. (Essentially it requires the constraints (4) to hold *at least* in the aggregate, i.e. when summed over all $(j, k) \in \Omega$. As shown in [14, p. 480] this imposes no "real" restriction on the solution procedure.) The reasons for using this form of normalization rather than others such as $w_1 = 1$, $\sum_{p=1}^{t} w_p = 1$, etc., are

[2] If some weight w_r were to be constrained nonpositive, we may replace r by its "antiattribute" l, i e define $y_{jl} = -y_{jr}$, and constrain w_l to be nonnegative If some or all of the weights are to be left unconstrained as to sign, this can be accomplished by appropriately defining an additional predictor variable and requiring all the $(t + 1)$ weights to be nonnegative [17, p 77]

discussed in [14]. The procedure of directly optimizing the linear program (6)–(10) will be referred to as procedure LP1.

Denoting by B^* the optimal value of the objective (6), an *index of fit* C^* for the ordinal regression problem is defined as the following transformation of B^*:

$$C^* = B^*/(1+B^*). \tag{13}$$

Since C is a strictly monotone transformation of B, the estimates $\{w_p\}$ which minimize B also minimize C, and conversely. Furthermore, since B^* is a nonnegative, C^* is bounded by zero and one, and provides a direct analogue to the "stress" measure in multidimensional scaling [6] and to $1 - R^2$ (R^2=coefficient of determination) in multiple regression [5, 7]. Consequently, the index C^* can be compared across different problems.

3. *Improved Computational Procedures for Ordinal Regression*

Direct optimization of the linear program (6)–(10) (i.e. computational procedure LP1) becomes computationally unwieldy when the number of observations, n, is large. For a problem with n observations and t predictors, set Ω consists of $N = n(n-1)/2$ pairs so that LP1 has $(N+1)$ constraints[3] and $(N+t)$ variables. The computational improvements LP2 through LP4 below also solve the linear program (6)–(10) and hence will obtain the same parameter estimates $\{w_p\}$. They are, however, likely to prove computationally more efficient since they utilize the special structure of the linear program (6)–(10) in devising special purpose procedures. (The analogy is to the stepping-stone [3] or u-v method [4] of solving transportation problems rather than direct optimization using the simplex method.)

3.1. PROCEDURE LP2. The solution to the linear program (6)–(10) can be considerably facilitated by considering its *dual linear program* [e.g. 4, pp. 124–127]. For each pair $(j, k) \in \Omega$, let us denote by u_{jk} the dual variable associated with the corresponding constraint (7). Let μ be the dual variable associated with (8). Then the dual to (6)–(10) is obtained as:

Maximize μ (14)

subject to

$$\sum_{(j,k)\in\Omega} (y_{jp} - y_{kp})u_{jk} + \left[\sum_{(j,k)\in\Omega} (y_{jp} - y_{kp})\right]\mu \le 0 \quad \text{for } p = 1,2,\cdots,t, \tag{15}$$

$$u_{jk} \le 1 \quad \text{for } (j,k) \in \Omega, \tag{16}$$

$$u_{jk} \ge 0 \quad \text{for } (j,k) \in \Omega \tag{17}$$

Constraints (15) correspond to the variables $\{w_p\}$. Similarly, the upper bound constraints (16) correspond to the variables z_{jk}. Introducing slack variables S_p ($S_p \ge 0$) for $p = 1,2,\cdots,t$ (note the distinction between s (eq. (3)) and S), constraints (15) can be rewritten in "equality form" as

$$\sum_{(j,k)\in\Omega} (y_{jp} - y_{kp})u_{jk} + \left[\sum_{(j,k)\in\Omega} (y_{jp} - y_{kp})\right]\mu + S_p = 0 \quad \text{for } p = 1,2,\cdots,t. \tag{18}$$

The variable μ is unconstrained as to sign, since constraint (8) is an equality. However, since the objective (6) is nonnegative (by definition of z_{jk} – see (12)) and hence bounded from below, it follows from linear programming duality theory that the optimum value of the objective (14), i.e. μ is also nonnegative. Thus the procedure LP2 is to solve the upper bounded linear program:

Maximize μ (19)

[3] Throughout this paper, when we refer to constraints we exclude the nonnegativity constraints (of the form (9)–(10)) and upper bound constraints on the variables

subject to

$$\sum_{(j,k)\in\Omega} (y_{jp} - y_{kp})u_{jk} + \left[\sum_{(j,k)\in\Omega} (y_{jp} - y_{kp})\right]\mu + S_p = 0 \quad \text{for } p = 1,2,\cdots,t, \tag{20}$$

$$u_{jk} \leq 1 \quad \text{for } (j,k) \in \Omega, \tag{21}$$

$$u_{jk},\ \mu,\ S_p \geq 0 \ \text{ for } (j,k) \in \Omega, \text{ and } p = 1,2,\cdots,t. \tag{22}$$

The tableau for the problem (19)-(22) is shown in Figure 1. The problem (19)-(22) has t constraints and $(N+t+1)$ variables (with the N variables u_{jk} bounded from above by unity). Thus LP2 has less storage requirements and can be expected to be computationally more efficient than LP1. As stated earlier, constraints (15) and (and hence (20)) correspond to the variables $\{w_p\}$ in LP1. Thus the estimates for the regression weights $\{w_p\}$ are obtained as the optimal dual variables corresponding to the upper bounded linear program (19)-(22).

3.2 PROCEDURE LP3. Procedure LP3, while essentially the same as LP2, utilizes the structure of the coefficients in LP2 to its advantage. Suppose we solve LP2 by the *revised simplex method*. Let us denote by $\{w_p^i\}$ the dual variables corresponding to the linear program LP2 at the ith iteration. Consider any of the first N columns in Figure 1 and let (j,k) be the pair associated with this column. Since the objective function has all its elements equal to zero corresponding to the first N columns, it follows that the *reduced price* (the "$c_j - z_j$" in familiar linear programming terminology) is given by:

$$-\sum_{p=1}^{t} w_p^i (y_{jp} - y_{kp}) = -\left(\sum_{p=1}^{t} w_p^i y_{jp}\right) + \left(\sum_{p=1}^{t} w_p^i y_{kp}\right) = -s_j^i + s_k^i. \tag{23}$$

Consequently, in procedure LP3 none of the first N columns of Figure 1 are explicitly stored. At the ith iteration of the revised simplex method, the values s_j^i are computed from eq. (3), using the current dual variables $\{w_p^i\}$. The reduced price corresponding to the column for pair (j,k) is then simply obtained as $s_k^i - s_j^i$. Thus the main advantage of LP3 over LP2 is the reduction in storage corresponding to the first N columns and the improved efficiency in "pricing out" these N columns. The remaining details for procedure LP3 are essentially the same as the revised simplex method for the upper bounded linear programs and need not be repeated here.

Two versions of LP3 were used in choosing the column for basis entry at the beginning of every iteration:

(a) Version LP3(a)—most positive indicator rule: Choose the column vector with the most positive reduced price[4] to be the pivot column for the next iteration. (Since the linear program (19)-(22) is a maximization problem, column vectors with positive reduced prices qualify as candidates for a pivot.) If all columns have nonpositive reduced prices, the current solution is optimal.

(b) Version LP3(b)—most positive for an observation rule: This rule consists of steps (i)-(iv) below for an iteration. For the first iteration, initialize $l' = 1$.[5]

(i) Set $l = l'$. Go to (ii).

(ii) Among the pairs $\{(l,k),\ k = l+1,\ l+2,\cdots,n\}$, if any of them has a positive reduced price go to (iii). Otherwise go to (iv).

(iii) From the pairs considered in (ii) above, identify the pivot column to correspond

[4] As mentioned earlier, the reduced price for column u_{jk} is $s_k^i - s_j^i$ This is true if u_{jk} is nonbasic at its lower bound If, however, the nonbasic vector u_{jk} is at its upper bound, the reduced price would be $s_j^i - s_k^i$

[5] The initial basis for LP3 ((a) and (b)) consists of the column vector corresponding to μ and *any* $(t-1)$ of the slack vectors (the slack vector S_q not in the basis is chosen such that the corresponding $\sum_{(j,k)\in\Omega} (y_{jq} - y_{kq})$ is positive) Since the right-hand side vector in Figure 1 has all its components zero, the basic solution also has all its components equal to zero and hence is primal feasible It can also be shown that the vector corresponding to μ never leaves the basis in subsequent iterations.

Primal Variables / Dual Variables	u_{12}	u_{13}	. .	u_{1n}	u_{23}	u_{24}	. . .	u_{2n}		$u_{n-1,n}$	μ	S_1	S_2	. . .	S_t	Right hand side
w_1	y_{11} $-y_{21}$	y_{11} $-y_{31}$	. . .	y_{11} $-y_{n1}$	y_{21} $-y_{31}$	y_{21} $-y_{41}$	. .	y_{21} $-y_{n1}$	. . .	$y_{n-1,1}$ $-y_{n1}$	$\sum_{(j,k)\in\Omega} (y_{j1}-y_{k1})$	1	0	. . .	0	0
w_2	y_{12} $-y_{22}$	y_{12} $-y_{32}$	. . .	y_{12} $-y_{n2}$	y_{22} $-y_{32}$	y_{22} $-y_{42}$	. .	y_{22} $-y_{n2}$	. .	$y_{n-1,2}$ $-y_{n2}$	$\sum_{(j,k)\in\Omega} (y_{j2}-y_{k2})$	0	1	. . .	0	0
. .	.	. .	. . .	. .	. .	.	. . .	. .	. .	.	. .	. .	. .	. . .	. .	. .
w_t	y_{1t} $-y_{2t}$	y_{1t} $-y_{3t}$	. . .	y_{1t} $-y_{nt}$	y_{2t} $-y_{3t}$	y_{2t} $-y_{4t}$	. . .	y_{2t} $-y_{nt}$	. . .	$y_{n-1,t}$ $-y_{nt}$	$\sum_{(j,k)\in\Omega} (y_{jt}-y_{kt})$	0	0	. .	1	0
Upper Bounds	1	1	. . .	1	1	1	. .	1	. . .	1 .	M	M	M	. . .	M	
Objective Function (maximize)	0	0	. .	0	0	0	. . .	0	. . .	0	1	0	0	. . .	0	

NOTE M is a very large positive number (e.g., 10^{30}).

FIG 1 Tableau for upper bounded linear program

to the pair with the most positive reduced price. Define $l' = l+1$ for the next iteration.

(iv) Update $l = l+1$ (if l becomes equal to n, redefine l as 1). If $l \neq l'$ go to (ii). Otherwise, choose the vector with the most positive reduced price among the slacks $S_1, S_2, \cdots, S_t$ and define $l'=1$. If, however, all the slacks $S_1, S_2, \cdots, S_t$ have nonpositive reduced prices, the current solution is optimal.

While rule (a) examines all the N pairs as potential candidates for basis entry, rule (b) examines, on an average, only a small proportion of these pairs (empirically this proportion tends to become smaller for larger values of n). Thus rule (b) would involve less search time per pivot. However, the reduced price under (a) can be expected to be more positive than under (b) and consequently (a) would result in a smaller number of iterations. Section 4 compares these two rules for their overall computational time.

Rules (a) and (b) correspond to the "matrix most negative rule" and the "row most negative rule," respectively, for pivot choice in transportation problems [15] (also a special structured linear program). Section 4 concludes that for "small" values of n ($n \leq 20$), rule (a) is computationally more efficient, while for "larger" n ($n \geq 30$ (say)), rule (b) is more efficient. These results are analogous to those found for transportation problems. Rules (a) and (b) do not exhaust all possibilities for choosing the column for basis entry Rules similar to the "first encountered negative rule," "lot minimum rule," etc., for transportation problems can be examined. However, as is the case for transportation problems [15], these additional rules are not likely to perform as well as rule (b) for large n, ($n \geq 40$ (say)). For small n such as $n=30$, average computational time for LP3 is less than 1 second on the IBM 360/67 and seems hardly worth improving upon.

3.3 Procedure LP4. Procedure LP4 solves the linear program (19)-(22) by the *dual simplex method with bounded variables* [16]. Suppose we wish to pivot on the lth row during the ith iteration of the dual simplex method. Then, for each column we need to know (a) its reduced price and (b) its updated coefficient on the lth row of Figure 1 (obtained as $D_l{}^i A$ where $D_l{}^i$ denotes the lth row of the inverse of the basis matrix at the ith iteration and A is the vector of coefficients in Figure 1 for the column under consideration). Consider any of the first N columns and let (j,k) be the pair associated with this column As in LP3, the reduced price for this column is obtained simply as $s_k{}^i - s_j{}^i$ (with the $s_j{}^i$ computed from (3) using the current duals $\{w_p{}^i\}$). The column vector A for the (j,k)-th pair has components $\{y_{j1}-y_{k1}, y_{j2}-y_{k2}, \cdots, y_{jt}-y_{kt}\}$. Let $D_l{}^i$ be represented as $(d_1{}^i, d_2{}^i, \cdots, d_t{}^i)$ so that the updated coefficient for the lth row in the column for pair (j,k) is:

$$\sum_{p=1}^{t} d_p{}^i(y_{jp} - y_{kp}) = \sum_{p=1}^{t} d_p{}^i y_{jp} - \sum_{p=1}^{t} d_p{}^i y_{kp} = r_j{}^i - r_k{}^i.$$

Thus for the ith iteration of the dual simplex method, the updated coefficients along row l for the first N columns can be efficiently obtained by (a) determining $(d_1{}^i, d_2{}^i, \cdots, d_t{}^i)$, the lth row of the basis inverse at the ith iteration, (b) computing $r_j{}^i = \sum_{p=1}^{t} d_p{}^i y_{jp}$ for $j = 1,2,\cdots,n$, and (c) finding the differences $r_j{}^i - r_k{}^i$ for the pairs $(j,k) \in \Omega$. Consequently, the first N columns of the matrix of Figure 1 need not be explicitly stored.

There are two potential advantages for procedure LP4 over LP3. First, LP4 maintains dual feasibility at every iteration. Thus the current dual variables represent a feasible set of regression weights (i.e. satisfy (5). In fact, they also satisfy (8) since the column vector corresponding to μ is a basis vector at every iteration[6]). Consequently,

[6] The initial dual feasible basic solution is constructed as follows Let l be the index corresponding to which $\sum_{(j,k)\in\Omega}(y_{jp} - y_{kp})$ attains its maximum over $p = 1,2, \cdots, t$ (This maximum is clearly positive, otherwise from (8) it follows that there is no feasible solution $\{w_p\}$ Although infeasibility is a theoretical

for a very large problem ($n \simeq 500$, say), even if we stop the procedure before attaining global optimality (because of excessively long computational time), we will still obtain feasible and hopefully "good" estimates for the regression weights. Second, in method LP3 we need a vector U (say) to indicate whether columns $1,2,\cdots,N$ in Figure 1 are nonbasic at their upper bound or not. If $n=200$ (say), this vector is of size 19,900. In method LP4, however, this information is implicitly known from the relative magnitudes of s_j and s_k. Assuming nondegeneracy,[7] since the solution is dual feasible, it follows that for pair (j,k), (i) if $s_j < s_k$, u_{jk} is nonbasic at its upper bound 1, (ii) if $s_j = s_k$, (j,k) is in the current basis, and (iii) if $s_j > s_k$, (j,k) is nonbasic at its lower bound 0. Consequently, no vector such as U is needed. This reduces storage requirements as well as computational effort. However, despite these two advantages it is hard to say whether LP4 would be computationally more efficient than LP3 since the number of iterations in the primal and dual methods could be quite different. The overall computational efficiencies of these methods are now compared in Section 4.

4. *Computational Comparison of the Procedures*

Table 1 reports the storage requirements of the procedures for several problem sizes (n,t). It may be noted that the storage requirements for computing with double precision (displayed in parentheses) are generally less than double the storage required with single precision because of some integer arrays used in these procedures. From Table 1 it is clear that the procedure LP1 is impractical for present-day computers for any large size problems (say, $n \geq 50$, $t \geq 5$). Likewise LP2 becomes unwieldy for $n \geq 100$, $t \geq 10$. Procedures LP3 and LP4, however, require only a modest amount of storage, although LP4 is clearly superior to LP3. Such differences can be important for "time-shared" computing, where on-line storage is at a premium compared to batch processing.

Several 4-attribute problems ($t=4$) for different values of n were randomly generated to compare the computational efficiency of the procedures. For each value of n, 20 problems were randomly generated and tested (for $n=150$ and 200, because of excessive computation time, only 5 replications (problems) were used).

The steps involved in generating and testing *each problem* are outlined in the Monte-Carlo (simulation) procedure below:

(i) A (4×4) variance-covariance matrix was randomly generated by first defining a (4×4) matrix $G = \{g_{ij}\}$ with its components drawn as (independent) random numbers uniformly distributed between $-.5$ and $+.5$. The matrix product $V = G^T \cdot G$ (where G^T denotes the transpose of G) defines a valid variance-covariance matrix for the variables (y_1,y_2,y_3,y_4) (say). (See [5, eq. (4-68)]). The variables (y_1,y_2,y_3,y_4) were standardized to have means zero and variances 1 by setting $\hat{V}$, the variance covariance matrix of the standardized variables, equal to the correlation matrix corresponding to V.

(ii) $(n + 30)$ observations were drawn independently from the 4-variate normal distribution with means zero and variance-covariance matrix $\hat{V}$. (For each observation this can be accomplished by sequentially drawing 4 univariate normal random numbers; see [7, p. 213] for details.) The last 30 observations constitute the "hold-out" sample to test the predictive validity of the ordinal regression procedure.

(iii) A set of four independent random numbers uniformly distributed between $-.5$ and $+.5$ were drawn to represent the "true" weights $\{w_1', w_2', w_3', w_4'\}$.

---

possibility, it never occurs in practice (see [14], p. 480). If it does, we can replace (8) by $w_l = 1$ where l is the "most relevant" attribute from prior judgment.) The initial basis consists of the column vectors corresponding to the t slack variables $(S_1,S_2, \cdots, S_t)$ (see Figure 1) except that the lth slack vector is replaced by the column vector corresponding to μ. Such a basis defines a dual feasible solution, with weights $w_p^1 = 0$ for $p = 1,2, \cdots, t$, except $w_l^1 = 1/\sum_{(i,k)\in\Omega}(y_{it} - y_{kl})$.

[7] This can be guaranteed almost always by slightly perturbing the attribute values $\{y_{ip}\}$, i.e. defining $y'_{ip} = y_{ip} + [(i-1)t + p]\delta$ where δ is a very small positive number (e.g. $\delta = 10^{-6}$).

Table 1: Computer Storage Requirements for LP1 through LP4

n	t	Storage Requirement in Words Single Precision (Double Precision)			
		LP1	LP2	LP3	LP4
20	5	39100 (77617)	2281 (4165)	420 (615)	340 (585)
50	5	$>10^6$ ($>10^6$)	13816 (25165)	1665 (2040)	700 (1185)
50	10	$>10^6$ ($>10^6$)	20326 (38170)	2105 (2905)	1150 (2070)
100	10	$>10^7$ ($>10^7$)	80426 ($>10^5$)	6430 (7780)	2000 (3570)
200	10	$>10^8$ ($>10^8$)	$>10^5$ ($>10^5$)	22580 (25030)	3700 (6570)
200	20	$>10^8$ ($>10^8$)	$>10^5$ ($>10^6$)	25260 (30360)	6400 (11940)

NOTE The storage requirements in words given above are for data arrays only (i e , the computer program and scalars, which take less than 8000 words of storage, are not included in these calculations)

(iv) For each of the $(n+30)$ observations the "true" value of the dependent variable was computed as

$$s_j' = \sum_{p=1}^{4} w_p' y_{jp}, \quad j = 1,2,\cdots,n+30. \tag{24}$$

(v) Error terms were added to the first n observations used for estimation as follows: Let $\hat{\sigma}^2$ denote the variance of the observations $\{s_1', s_2', \cdots, s_n'\}$. Let E denote the desired proportion of error variance in the "observed" dependent variable. Since $\hat{\sigma}^2$ denotes the proportion $(1-E)$ of the observed variance, $\Delta = E\hat{\sigma}^2/(1-E)$ gives the error variance. The observed dependent variables were then computed as $s_j'' = s_j' + e_j$ for $j = 1,2,\cdots,n$, where e_j, the error term, is drawn randomly from a normal distribution with mean zero and variance Δ.

(vi) The first n observations were then renumbered so that $j=1$ corresponds to the observation with the largest value of s_j'', $j=2$ the second largest, $\cdots$, $j=n$ corresponds to the observation with the smallest s_j''.

(vii) The values $\{y_{jp}\}$ for $j = 1,2,\cdots,n$ (renumbered as above) and $p = 1,2,3,4$ were then input into procedures LP1, LP2, LP3(a), LP3(b), and LP4. The time to obtain the optimal estimates $\{w_p\}$ (exclusive of input and output) were recorded.[8]

(viii) To test the predictive power of the ordinal regression model, the "predicted" dependent variables s_j for $j = n+1, n+2,\cdots,n+30$ (the hold-out sample) were computed as

$$s_j = \sum_{p=1}^{4} w_p y_{jp} \quad \text{for } j = (n+1), (n+2), \cdots, (n+30), \tag{25}$$

where $\{w_p\}$ are the estimated weights in step (vii). (As remarked in Section 3, these estimates will be the same for each of the procedures LP1–LP4.)

(ix) The Pearson Product Moment correlation R was computed between $\{s_j'\}$ and $\{s_j\}$ (eqs. (24) and (25)) for the 30 observations in the hold-out sample.[9] Thus R is a

[8] Since the weights are unconstrained in sign (see step (iii)), the estimation is to be done with the given four attributes plus a fifth attribute with values $y_{j5} = -\sum_{p=1}^{4} y_{jp}$ The optimal estimates are given by $(w_1 - w_5)$, $(w_2 - w_5)$, $(w_3 - w_5)$, and $(w_4 - w_5)$, where $w_p \geq 0$ for $p = 1,2,\cdots,5$ (see Footnote 2)

[9] The Spearman (Rho) rank order correlation coefficients, although not reported in the results to be discussed below, were only about .01, on the average, less than the Pearson R.

Table 2: Comparative Performance of Procedures LP1 - LP4

Number of observations n	Mean Solution Times (standard deviations) in Seconds					Mean Predictive Validity R
	LP1	LP2	LP3(a)	LP3(b)	LP4	
10	.574 (.215)	115 (.067)	031 (.013)	037 (.014)	.049 (.020)	.910 (.081)
14	3.135 (.975)	403 (.104)	075 (.022)	098 (.025)	120 (.036)	.955 (.049)
20	-	1.786 (.420)	243 (.060)	275 (.062)	.261 (.110)	961 (.037)
30	-	8.108 (1.524)	.838 (.187)	791 (.113)	.850 (.350)	972 (.039)
40	-	25.147 (5.231)	2.481 (.574)	1.911 (.357)	2.031 (.874)	.987 (.010)
50	-	-	5.210 (.709)	3.907 (.670)	3.503 (1.079)	.986 (.014)
60	-	-	11.024 (2.125)	6.754 (1.176)	6.278 (2.211)	991 (.008)
70	-	-	-	10.643 (1.173)	8.269 (3.728)	991 (.008)
80	-	-	-	-	13.624 (6.056)	.993 (.004)
90	-	-	-	-	20.008 (7.577)	.996 (.004)
100	-	-	-	-	25.388 (8.467)	.996 (.003)
150	-	-	-	-	93.406 (27.830)	.997 (.003)
200	-	-	-	-	185.256 (75.110)	.998 (.002)

NOTE All problems are with 4 independent variables (t=4) and proportion of error variance E=.2 Computational times are based on 20 randomly generated problems each (except n=150 and n=200 are based on 5 replications each) on the IBM 360/67 (FORTRAN IV, H Compiler) and are exclusive of input and output '-' indicates computations which were not performed

measure of the *predictive validity* of the ordinal regression model using the linear programming estimation procedure.

All computations were performed on the IBM 360/67 (Fortran IV, H Compiler). To maintain accuracy, computations were done in double-precision arithmetic and the current basis inverse was recalculated after every 100 pivots. The computational results are displayed in Table 2, where the mean solution times and standard deviations (displayed in parentheses) are reported based on 20 replications each (for $n=150$ and 200, 5 replications each) with the proportion of error variance E equal to .20 (see step (v)).[10] Some of the computations were not performed (these are marked as "–" in Table 2) because either (i) the required storage exceeded the available storage ($n \geq 20$ for LP1) or (ii) computational times were large and the direction of the results were unambiguously clear even without performing the computations ($n > 40$ for LP2, $n > 60$ for LP3(a), and $n > 70$ for LP3(b)).

There is marked improvement in the solution times in going from LP1 to LP2. For

[10] The values $t=4$ and $E=20$ approximate the values of t and E typically encountered in (multiattribute) marketing applications, an area where some form of ordinal regression has been used in the past The number of replications was chosen as 20 based on the conflicting considerations of accuracy versus computer costs

$n=14$, LP2 took only about $\frac{1}{8}$th of the time required by LP1. Likewise, for $n=40$ there is about a 12:1 reduction in going from LP2 to LP3 or LP4. Thus LP3 and LP4 are significantly superior to LP1 and LP2.

The relative performance of LP3(a), LP3(b), and LP4 depends on the value of n. For $n \leq 20$, the "most positive indicator rule" (LP3(a)) performs best. For such small values of n, the search effort for computing all the N reduced prices is quite small, and hence it seems desirable to get the maximum improvement per pivot. For $n \geq 30$, the search effort becomes proportionately larger, and LP3(b) (most positive for an observation rule) takes over as the best procedure. For $n \geq 50$, LP4 performs the best. Even in the range of n where LP4 is not the best, its solution time differs from the best by less than 0.12 seconds/problem. This coupled with the considerably smaller storage requirements for LP4 (See Table 1) makes LP4 a very attractive solution procedure. LP4 also has the advantage that at every stage of computation, a feasible $\{w_p\}$ is at hand so that if we had to stop the procedure before attaining the global optimality because the computation times are too large (say, for $n=500$) we can still hope to get a "good" feasible solution within reasonable computational effort. Finally, although the above analysis of the superiority of LP4 is based on $t=4$ and $E=.2$, a number of "spot checks" for other values of t and E showed that LP4 continued to be the best for $n \geq 50$. For these reasons it seems clear that LP4 should be chosen as the best solution procedure.

The relationship of the mean computational time of LP4 as a function of n was studied from the data of Table 2. A log-linear regression fitted to this data yielded

$$T = .0665n^{2.796}. \tag{26}$$

The adjusted R^2 (coefficient of determination adjusted for degrees of freedom) was .998, and both coefficients were statistically significant beyond the .001 level.

The constant term .0665 in eq. (26) is for the particular case of $t=4$ and $E=.2$. To study the effect of n, t, and E on the mean solution time of LP4, a $3\times3\times3$ design was used with n at the three levels 20, 40, 60, t at 2, 4, 6, and E at .25, 0.5, 1. An $E=1$ means that all the variance in the dependent variable is error variance. This can be accomplished by taking the "true" weights $w_1' = w_2' = \cdots = w_t' = 0$ in step (iii) and $\Delta=1$ in step (v) of the simulation procedure outlined earlier. Each of the 27 possible parameter combinations were tested with LP4. For each parameter combination, 10 problems were solved and the mean solution times and standard deviations based on these 10 replications are reported in Table 3.

A log-linear multiple regression relating the mean computation time T to the parameters n, t, and E yielded

$$T = .0039n^{2.795}t^{1.365}E^{.230}. \tag{27}$$

The adjusted R^2 was .988 and all the coefficients were statistically significant beyond

Table 3: Effects of n, t and E on the Solution Time of LP4

n \ t	Mean Solution Times (standard deviations) of LP4 in Seconds								
	2	4	6	2	4	6	2	4	6
20	.085 (.052)	.255 (.094)	.455 (.173)	.093 (.055)	.294 (.127)	.521 (.151)	.127 (.059)	.387 (.146)	.525 (.179)
40	.503 (.534)	1.987 (.934)	2.811 (.908)	.599 (.519)	1.936 (.800)	2.968 (.912)	.898 (.656)	2.067 (.989)	3.178 (1.307)
60	1.980 (1.891)	5.692 (1.727)	9.304 (2.933)	2.216 (1.989)	5.428 (1.978)	10.089 (3.800)	4.572 (2.490)	6.349 (2.885)	11.659 (4.245)
	E = .25			E = .50			E = 1.0		

NOTE 1) The computational times reported above are based on 10 randomly generated problems each on the IBM 360/67 (FORTRAN IV, H Compiler) and are exclusive of input and output.

2) n = number of observations, t = number of independent variables, and E = proportion of error variance in the dependent variable

Table 4: Effects of n, t and E on Predictive Validity

n \ t	Means (and standard deviations) of Predictive Validity R					
	2	4	6	2	4	6
20	989 (.013)	978 (.024)	913 (.062)	.971 (.033)	924 (.066)	705 (.230)
40	996 (.006)	983 (.014)	967 (.025)	.987 (.017)	.959 (.030)	915 (.059)
60	.998 (.002)	987 (.012)	971 (.026)	.994 (.007)	968 (.028)	921 (.080)
	E = .25			E = .50		

NOTE 1) The means (and standard deviations) are based on 10 randomly generated problems each.

2) n = number of observations, t = number of independent variables, and E = proportion of error variance in the dependent variable

the .001 level. From the exponents in (27), we find that n has the largest effect on T, and t the next largest; E has a considerably smaller effect on T. It is interesting to note that the exponents of n in the eqs. (26) and (27) are remarkably consistent.

The predictive validities R (see step (ix) of the Monte Carlo procedure outlined earlier) are reported in the last column of Table 2 and in Table 4. For $E=1$, as remarked earlier, the true weights $w_p'=0$ for $p=1,2,\cdots,t$, so that the true dependent variable $s_j'=0$ for every j (eq. (24)). Consequently $R=0$, as it should be, since there is no systematic variation in the dependent variable. Consequently Table 4 reports the values of R only for $E=.25$ and $E=.50$.

The correlation coefficient R is above .9 almost always (the only exception is for $n=20$, $t=6$, and $E=.5$ in Table 4, where R drops to .705). It is interesting that the predictive validity is high even for problems such as $n=10$, $t=4$, $E=.2$; $n=20$, $t=6$, $E=.25$; $n=20$, $t=4$, $E=.5$. Admittedly, the tests conducted do not exhaust all possibilities in terms of n, t, E, other distributions for $\{y_{jp}\}$, and the like. Nevertheless, the ordinal regression procedure coupled with the linear programming solution procedure LP4 offers considerable promise as a valid and computationally feasible procedure and provides a viable alternative to the standard multiple regression for problems where the data for the dependent variable satisfy only ordinal properties.

REFERENCES

1 CARROLL, J D Individual differences and multi-dimensional scaling In *Multidimensional Scaling: Theory and Applications in the Behavioral Sciences, Vol I Theory*, R N. Shepard, A.K Romney, and S Nerlove, Eds., Seminar Press, New York, 1972, pp 105-155

2 CHAPMAN, R G , WEINSTEIN, A G , AND TURNER, R.E An investigation of student ranking procedures. Working paper, Grad Sch Ind Admn , Carnegie-Mellon U , Pittsburgh, Pa , 1973

3 CHARNES, A , AND COOPER, W W *Management Models and Industrial Applications of Linear Programming, Vol I* Wiley, New York, 1961.

4 DANTZIG, G B. *Linear Programming and Extensions* Princeton U Press, Princeton, N J , 1963

5 JOHNSTON, J *Econometric Methods*, 2nd ed McGraw-Hill, New York, 1972

6 KRUSKAL, J B Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis *Psychometrika 29* (1964), 1-27

7 MOOD, A M , AND GRAYBILL, F A *Introduction to the Theory of Statistics* McGraw-Hill, New York, 1963

8 PEKELMAN, D , AND SEN, S K Mathematical programming models for the determination of attribute weights *Manag Sci 20* (1974), 1217-1229

9 SHEPARD, R N Metric structures in ordinal data *J. Math Psych. 3* (1966), 287-315

10. SHOCKER, A.D A methodological approach to the identification of a feasible mass transit configuration for maximal satisfaction to users In Proc of the 1971 AIDS Conf , M Hopfe, ed , American Institute of Decision Sciences, St Louis, Mo , 1971, pp 31-35

11 SHOCKER, A.D , AND SRINIVASAN, V. A consumer-based methodology for the identification of new product ideas *Manag Sci 20* (1974), 921-937

12 SIMMONARD, M. *Linear Programming* Prentice-Hall, Englewood Cliffs, N J , 1966

13 SRINIVASAN, V , AND SHOCKER A D Linear programming techniques for multidimensional analysis of preferences *Psychometrika 38* (1973), 337-369

14 SRINIVASAN, V , AND SHOCKER, A D Estimating the weights for multiple attributes in a composite criterion using pairwise judgments *Psychometrika 38* (1973), 473–493
15 SRINIVASAN, V , AND THOMPSON, G L Benefit-cost analysis of coding techniques for the primal transportation algorithm *J ACM 20*, 2 (April 1973), 194–213
16 WAGNER, H M The dual simplex algorithm for bounded variables *Naval Res Logist Quart 5* (1958), 257–261
17 WAGNER, H M *Principles of Operations Research*, 2nd ed Prentice-Hall, Englewood Cliffs, N J , 1975

RECEIVED NOVEMBER 1974, REVISED AUGUST 1975