A COMPUTATIONAL PROCEDURE FOR SETTING CUTOFF SCORES FOR MULTIPLE TESTS

Zvi Drezner*
George O. Wesolowsky**
Willi Wiesner***

Organizations commonly use several different predictors such as ability tests, personality tests, interviews, and reference checks in selecting employee. In fact, the use of multiple predictors serves as a good way of triangulating on applicants' abilities to do the job and generally provides better prediction of job performance than a single predictor. However, in using multiple predictors, organizational decision makers are faced with the challenge of how to make sense of the various, and sometimes conflicting, sources of information about applicants in order to make an informed decision.

Setting a cutoff on a single predictor is relatively straightforward. However, the subjective procedures used for setting cuttoffs on multiple predictors have not been satisfactory. This study demonstrates the feasibility of a computational procedure for setting multiple cutoffs. Such a procedure should be more cost-effective and accurate than the subjective methods currently used.

Organizations commonly use several different predictors such as ability tests, personality tests, interviews, and reference checks in selecting employees (Bureau of National Affairs, 1988; Karren & Nkomo, 1988). In fact, the use of multiple predictors serves as a good way of triangulating on applicants' abilities to do the job and generally provides better prediction of job performance than a single predictor (Gatewood & Feild, 1994). However, in using multiple predictors, organizational decision makers are faced with the challenge of how to make sense of the various, and sometimes conflicting, sources of information about applicants in order to make an informed decision. One way of dealing with data from multiple predictors is to enter them into a regression equation. The applicant's scores on each predictor (e.g., tests, interviews, reference checks) are

**Zvi Drezner is affiliated with California State University, Fullerton, California.*
***George O. Wesolowsky is affiliated with McMaster University, Hamilton, Ont. Canada.*
**** Willi Wiesner is affiliated with McMaster University, Hamilton, Ont. Canada.*

weighted and summed to yield a total score (e.g., predicted job performance). The appropriate regression weights or *b* values are determined through prior research where the unique contributions of each predictor score (X_i) to predicting job performance (Y) are investigated.

The multiple regression approach assumes that a low score on one predictor can be compensated for by a high score on another predictor. Thus, an applicant could do very poorly in the interview (e.g., receive a score of zero) and still do well if he or she receives high scores on the tests and the reference check. However, this assumption made by the multiple regression model is not necessarily warranted. There might be a minimum level of competence required on each of the predictors for the individual to perform acceptably in the job. For example, a very low interview score might indicate that the applicant has such poor interpersonal and communication skills that he or she cannot function acceptably in retail sales, regardless of high cognitive ability and extraversion scores.

When the assumptions of the multiple regression approach cannot be met, an alternative is to use cutoff scores (whether in multiple cutoff, multiple hurdle, or combination methods. See Gatewood & Feild, 1994). Cutoff scores serve as criteria or thresholds in selection decisions. Applicants who score below the cutoff on any of the predictors are rejected. Thus, cutoff scores ensure that applicants meet some minimum level of ability or qualification to be considered for a job. The cutoff approach assumes that a minimum level is required on each of the attributes measured by the predictors for successful job performance (i.e., it does not assume a linear relationship among the predictors and job performance). The approach also assumes that the predictors are noncompensatory (i.e., it is not possible to compensate for a low score on one predictor with a high score on another predictor).

Setting cutoff scores is relatively straightforward when there is only one predictor. The usual method involves identifying the proportion of applicants who are to be hired and determining how stringent the cutoff score should be to select only the desired number of applicants. First, the expected selection ratio is calculated (number of individuals to be hired divided by the expected number of applicants). Next, the distribution of the applicants' scores on the predictor is estimated by examining the predictor score distributions of past groups of applicants or of current employees (i.e., predictive or concurrent validation data). Finally, the cutoff score is established by applying the selection ratio to the predictor score distribution in order to determine the score that only the top applicants (the proportion to be hired) would attain. For example, if a fire department seeks to hire 5 firefighters and 150 people are expected to apply, the selection ratio will be .03 (5/150). About 3 percent of expected applicants will be accepted or, conversely, about 97 percent of expected applicants will have to be rejected. The cutoff score should therefore be set at the 97^{th} percentile of the distribution of predictor scores (plus or minus one standard error of measurement). That is, the cutoff score is set so that only 3% of applicants would be expected to meet or exceed the score (or 97% would fall below it). As noted above, this approach is limited to setting cutoffs for a single predictor.

When more than one predictor is used, setting cutoff scores is considerably more difficult (Buck, 1977; Cascio et al., 1988; and Cascio, 1991). Cascio (1991, p. 286) points out that "in general, no satisfactory solution has yet been developed for setting optimal cutoff scores in a multiple cutoff model." Although it is not an ideal approach, one method that has been used to set cutoff scores is to use expert judges. There are several ways in which expert judges can be used to establish cutoffs but they differ only slightly in their methods. We will briefly consider the general approach and readers are encouraged to consult Cascio (1991), Gatewood and Feild (1994), or Schmitt and Klimoski (1991) for more detailed treatments of the various methods. Experienced employees, supervisors, or managers who know the job well or industrial psychologists typically serve as expert judges. Essentially, the expert judges are asked to rate the difficulty of test items (or interview questions) and to indicate what score on each item should be attained by a minimally competent applicant. These ratings are summed across items to yield a pass threshold or cutoff score. Cutoff scores can be established in this manner for each of the predictors used in the selection process.

Although expert judges are able to determine cut-off scores to ensure minimal competence on each of the predictors, they have tremendous difficulty setting optimal cut-off scores for several predictors so that the desired number of applicants is selected (Bazerman, 1986). That is, they must not only determine what scores would indicate minimal competence on each predictor but also what scores on each predictor would generate the desired number of selectees. Rank ordering scores for each predictor is inadequate for these purposes because the rank ordering of each applicant will differ across predictors. For example, an applicant who is ranked 1st on one predictor might be ranked 20th on another predictor and 59th on a third predictor. Clearly, a computational process is the only efficient means of setting multiple cut-off scores.

The purpose of this study is to explore the feasibility of a computational approach to setting multiple cutoff scores. Such an approach would be more objective and more efficient than using expert judges. Moreover, we expect that this approach will generate cutoff scores which will optimize the prediction of successful job performance.

FORMULATION

Assume a matrix of k columns and n rows. This matrix contains scores s_{ij} for $i = 1,K ,n;\ j = 1,K ,k$, representing the score of candidate i in test j. The matrix also contains a vector g_i for $i = 1,K ,n$ indicating whether a candidate's job performance is acceptable ($g_i = 1$) or unacceptable ($g_i = 0$). The problem is to find k cutoff scores, so that a candidate is classified as "acceptable" if he or she is above (or equal to) the cutoff score in at least k-p of the scores, and is classified as "unacceptable" if he or she misses the cutoff it more than p times. (The number p may be zero if a candidate is classified as "acceptable" if he meets the cutoff in all tests.) The objective is to minimize the number of incorrect classifications of candidates.

For each column j the s_{ij} are sorted in increasing order and n_j distinct values $s_1^{(j)} < \text{K} < s_{n_j}^{(j)}$ are obtained. $n_j \le n$ because there may be ties in the scores. Each column j defines a set of n_j 0-1 variables $x_1^{(j)}, \text{K}, x_{n_j}^{(j)}$. If the score is below the cutoff point, then $x_m^{(j)} = 0$, and if the score is above the cutoff point, then $x_m^{(j)} = 1$. In order to guarantee that the x's define a cutoff score, we define the set of constraints for each j:

$$x_1^{(j)} \le x_2^{(j)} \le \text{K} \le x_{n_j}^{(j)} \qquad (1)$$

This set of inequalities guarantees that the vector $x^{(j)}$ starts with a sequence of uninterrupted zeroes and continues with a sequence of uninterrupted ones. It is possible to have either no zeroes at the beginning in which case all the $x^{(j)}$'s are equal to one, or no ones at the end of the vector in which case all the $x^{(j)}$'s are equal to zero. The cutoff point is between the two values where the zeroes end and the ones start (or the first $x_m^{(j)} = 1$). If $x_m^{(j)} = 0$, then the score is below the cutoff point for column j, and if $x_m^{(j)} = 1$, the score is above the cutoff point. Once a solution is obtained, the cutoff points can be determined by examining the solution $x^{(j)}$.

For each score s_{ij} there exists an index $\phi(i,j)$ such that $s_{ij} = s_{\phi(i,j)}^{(j)}$. The variable $x_{\phi(i,j)}^{(j)}$ determines whether this score is above the cutoff for that column or not. Therefore, the number of tests which are above the cutoff point for row i is $\sum_{j=1}^{k} x_{\phi(i,j)}^{(j)}$.

If $\sum_{j=1}^{k} x_{\phi(i,j)}^{(j)} \ge k - p$, the candidate is classified as "acceptable", otherwise the candidate is classified as "unacceptable". Consider the expression (recall that $g_i = 1$ means an acceptable candidate and $g_i = 0$ means an unacceptable one):

$$g_i\left[k - p - \sum_{j=1}^{k} x_{\phi(i,j)}^{(j)}\right] + (1 - g_i)\left[\sum_{j=1}^{k} x_{\phi(i,j)}^{(j)} - k + p + 1\right] \qquad (2)$$

1. If candidate i is acceptable ($g_i = 1$), the second term in (2) is zero.
   - If he or she is classified as "acceptable" the expression (2) is non-positive
   - If he or she is classified as "unacceptable" the expression (2) is positive (is at least one)
2. If candidate i is unacceptable ($g_i = 0$), the first term in (2) is zero.
   - If he is classified as "unacceptable" the expression (2) is non-positive
   - If he is classified as "acceptable" the expression (2) is positive (is at least one)

3. Therefore, if the classification is it correct, expression (2) is non-positive, and if the classification is wrong, expression (2) is at least one.
4. The maximum possible value for expression (2) is *k-p* if $g_i = 1$ and *p+1* if $g_i = 0$. This is the same as $g_i(k-p)+(1-g_i)(p+1)$.

n additional 0-1 variables y_i for $i=1,K,n$ are added to the formulation. y_i indicates whether the classification is correct ($y_i = 1$), or incorrect ($y_i = 0$). The objective is to maximize $\sum_{i=1}^{n} y_i$.

By the above discussion, y_i is defined by the constraint:

$$g_i\left[k - p - \sum_{j=1}^{k} x_{\phi(i,j)}^{(j)}\right] + (1-g_i)\left[\sum_{j=1}^{k} x_{\phi(i,j)}^{(j)} - k + p + 1\right] \le [g_i(k-p) + (1-g_i)(p+1)](1-y_i)$$

Rearranging terms leads to:

$$(1-2g_i)\sum_{j=1}^{k} x_{\phi(i,j)}^{(j)} + [g_i(k-p) + (1-g_i)(p+1)]y_i \le k(1-g_i) \qquad (3)$$

The complete formulation is:

$$z = \max_{x,y}\left\{\sum_{i=1}^{n} y_i\right\}$$

$$\text{subject to: } x_1^{(j)} \le x_2^{(j)} \le K \le x_{n_k}^{(j)} \qquad (4)$$

$$(1-2g_i)\sum_{j=1}^{k} x_{\phi(i,j)}^{(j)} + [g_i(k-p) + (1-g_i)(p+1)]y_i \le k(1-g_i)$$

$$x_i^{(j)}, y_i \in \{0,1\}.$$

DISCUSSION

The Case *p=0*

In our investigation we found that for *p=0*: when y_i only are required to be 0-1 variables and $x_i^{(j)}$ may be real numbers, we usually obtain integer values for the *x*'s as well. We did not find an example where this is not true, but it cannot generally be proven. This makes the formulation "integer friendly" for the values of the *x*'s and only *n* integer variables are required. We propose to solve the problem with only the requirement that the n y's are integer, and only if non-integer values are obtained for the x's, to resolve the problem requiring that it all variables are integers. The reason for this property is as follows. Consider the second constraint in (4). There are four possibilities for the values of g_i and y_i summarized in Table 1.

Examining Table 1 shows that when $y_i = 0$ the constraint is superfluous because it is always satisfied whether the x's are fractions or integers. This should be expected because y_i can only be 0 or 1 and since we maximize the sum of the y's, y_i will be equal to 1 at the maximum unless the constraint involving it is not satisfied. In this case, if the constraint for $y_i = 0$ is not satisfied, we have an infeasible solution. On the other hand, if $g_i = 1$ and $y_i = 1$, then all the x's on the left hand side of this constraint must all equal to 1. All the constraints for which $g_i = y_i = 1$ force many x's to be equal to 1. Therefore, a constraint for which $g_i = 0$, $y_i = 1$ will usually have some of its x's on the left hand side equal to one because of other constraints. If *k-1* of them are equal to 1, then the remaining one must be equal to zero. By this reasoning many of the x's are forced to be zero or one (mostly one). The only case where we may get non-integer x's is when for a constraint of the type $g_i = 0$, $y_i = 1$ fewer than *k-1* x's are fixed to zero or one by the above reasoning by other constraints. And even then it is possible that the solution will be zero or one for these "free" x's. We would also like to note that for practical problems (and especially near the optimum) there are many $y_i = 1$ because $y_i = 1$ means that we predict correctly candidate i and we expect to predict correctly many of them. That means that the majority of the constraints will be for $y_i = 1$ which yields many "non-free" x's that are forced to get the value zero or 1.

Table 1: Possible Constraints

g_i	y_i	Constraint
0	0	$\sum_{j=1}^{k} x^{(j)}_{\phi(i,j)} \leq k$
0	1	$\sum_{j=1}^{k} x^{(j)}_{\phi(i,j)} \leq k - p - 1$
1	0	$\sum_{j=1}^{k} x^{(j)}_{\phi(i,j)} \geq 0$
1	1	$\sum_{j=1}^{k} x^{(j)}_{\phi(i,j)} \geq k - p$

This argumentation is not effective for $p>0$ because even for a constraint of the type $g_i = y_i = 1$ no x's are surely fixed to 1.

Computational Experience

The integer programming formulation was programmed in AMPL (Fourer et al.,1993) and given in the appendix. The program is available from the first author.

We constructed two illustrative problems: one problem with n=30 candidates and k=5 tests, and the other with n=100 candidates and k=3 tests. The first problem was investigated more extensively. The raw data for the first problem have been already sorted for each test and common scores grouped into one variable. We had nine different scores among the 30 candidates for each test. The data used are given in Table 2.

The first problem is based on 45 x-variables and 30 y-variables. There are 40 constraints of the first type and 30 constraints of the second type.

We solved the problem for p=0, 1, 2, 3, 4 by linear programming (no integrality constraints), only 0-1 y's and all integer variables. The solution for this particular problem was obtained in a couple of seconds of computer time on a 486 33MHz computer. The response was so fast that we did not notice a difference in run time between the linear programming solution, only 0-1 y's solution, and all integer solution. The results are summarized in Table 3.

Table 2: The First Example Problem

i	g_i	Test1	Test2	Test3	Test4	Test5
1	0	2	4	6	3	1
2	1	4	5	7	9	4
3	0	3	7	4	5	1
4	0	5	2	2	1	9
5	1	7	3	6	8	7
6	0	6	7	2	7	2
7	0	3	7	7	2	1
8	0	1	7	1	5	3
9	1	6	1	2	9	4
10	0	3	7	3	2	8
11	1	8	7	5	3	5
12	1	5	6	3	7	6
13	1	1	7	4	8	9
14	0	5	9	5	4	2
15	1	8	8	6	6	1
16	1	9	3	9	6	3
17	1	9	1	9	8	6
18	0	2	6	7	1	4
19	0	3	7	8	3	8
20	0	3	4	2	5	9
21	0	5	8	1	6	2
22	1	9	9	4	6	7
23	0	4	3	3	8	4
24	1	5	3	8	9	6
25	0	1	6	7	1	3
26	0	4	5	4	2	9
27	0	9	1	9	2	6
28	0	3	6	1	5	9
29	1	8	7	5	7	1
30	0	5	8	6	3	1

For p=0, 27 candidates were categorized correctly (the objective function is 27, and all $y_i = 1$ except 9, 11, 23). The cutoff points were 1, 1, 3, 6, 1, respectively, for the five tests. These cutoff scores were determined by checking the solutions for the x's. For j=1, 2, and 5 all the x's were equal to 1 and therefore the cutoff score is the lowest value. For

j=3 the first two x's were equal to 0 and the rest equal to 1, and for j=4 the first five x's were equal to 0. A candidate is classified "acceptable" if he scored at least these values in all five tests and is classified "unacceptable" if he failed to get at least the particular score on any of the five tests. Note that tests #1, #2 and #5 can be discarded from consideration because the "passing" score is the minimum possible score and all candidates (whether "acceptable" or "unacceptable") passed these tests successfully. The testing procedure can be simplified to include only tests #3 and #4 because the other three tests do not provide any information that helps to distinguish between acceptable and unacceptable candidates. They are dominated by tests #3 and #4.

The same solution was obtained when we require only the y's to be integer for all p's except for p=1. For p=1 we obtained the same y's but some of the x's were non-integer. The linear programming solution was non-integer in all cases.

Table 3: The First Example Problem Solved by Linear and Integer Programming

	p=0	p=1	p=2	p=3	p=4
L.P. Solution (z)	29.4	29.75	29.667	29.5	29.4
0-1 y Solution	27	29	29	28	27
Are x's Integer?	yes	no	yes	yes	yes
All integer	27	29	29	28	27
$y_i = 0$	9,11,23	9	27	12,27	12,13,27
Cutoff Points	1,1,3,6,1	4,6,4,6,3	5,10,4,6,4	6,10,4,8,10	7,10,9,9,10

The second problem with 100 candidates and three tests is about the largest problem that can be solved by the *students' version* of AMPL which can handle problems with up to 300 variables and 300 constraints. The full-size version of AMPL can efficiently handle much larger problems. The data for the second problem are given in Table 4. In order to prepare the data for AMPL, the scores for each test were sorted, and the lowest score was entered as "1", the second lowest as a "2" and so on.

The student version of AMPL (Fourer et al., 1993) found a solution to the second problem in a couple of seconds. The cut-off points are 99, 28, and 52.Ninety seven of the applicants were classified correctly. Only applicants #6, #11, and #20 were classified incorrectly. Both cases when only the y's are integer or all the variables are integer were solved in about the same computer time and yielded the same solution. The linear programming relaxation of the problem resulted in a solution with an objective function of 99.

CONCLUSION

Our program was able to quickly generate multiple cutoff scores with a very high degree of accuracy. This computational method requires considerably less time and effort than the usual expert judgement approach and has the potential of providing substantial savings to the hiring organization. Moreover, this approach is more likely to provide accurate cutoff scores because it is keyed to job performance scores rather than subjective judgements made by job experts.

Table 4: The Data for the Second Problem

#	Scores			g_i	#	Scores			g_i	#	Scores			g_i
1	130	51	56	1	34	95	33	60	0	67	114	43	69	1
2	85	40	71	0	35	95	48	67	0	68	120	40	54	1
3	100	37	77	1	36	92	42	39	0	69	122	46	56	1
4	90	52	58	0	37	131	43	63	1	70	99	43	53	1
5	105	42	69	1	38	110	38	63	1	71	131	45	59	1
6	91	39	79	1	39	103	39	73	1	72	134	35	66	1
7	114	39	75	1	40	121	34	61	1	73	114	45	65	1
8	129	43	82	1	41	86	22	52	0	74	111	38	60	1
9	120	33	54	1	42	100	42	51	0	75	119	38	63	1
10	122	44	52	1	43	102	50	76	1	76	136	41	61	1
11	94	28	66	1	44	103	36	75	1	77	113	37	60	1
12	120	53	72	1	45	115	40	66	1	78	117	39	61	1
13	122	32	73	1	46	90	40	80	0	79	101	34	56	1
14	103	39	63	1	47	99	29	69	1	80	88	32	73	0
15	118	39	73	1	48	124	43	62	1	81	125	43	72	1
16	100	44	43	0	49	131	46	68	1	82	127	38	76	1
17	92	44	47	0	50	128	30	59	1	83	117	30	57	1
18	99	34	62	1	51	91	38	75	0	84	132	37	85	1
19	102	46	66	1	52	131	40	72	1	85	112	30	81	1
20	92	40	71	1	53	105	43	77	1	86	111	40	68	1
21	101	49	62	1	54	90	33	68	0	87	122	41	42	0
22	89	31	74	0	55	83	35	71	0	88	94	35	54	0
23	111	33	69	1	56	143	39	47	0	89	123	40	73	1
24	120	35	54	1	57	136	49	70	1	90	84	45	54	0
25	90	50	63	0	58	113	37	75	1	91	101	28	71	1
26	119	38	76	1	59	114	46	72	1	92	131	40	76	1
27	113	50	65	1	60	115	38	62	1	93	102	37	64	1
28	103	41	64	1	61	103	34	65	1	94	110	40	71	1
29	125	46	73	1	62	85	39	77	0	95	140	38	55	1
30	105	43	53	1	63	117	40	61	1	96	98	34	52	0
31	102	35	50	0	64	107	34	52	1	97	116	42	67	1
32	118	45	68	1	65	117	48	73	1	98	112	38	78	1
33	90	41	46	0	66	119	40	66	1	99	113	54	87	1
										100	119	47	65	1

APPENDIX: THE AMPL PROGRAM AND DATA FILE

The AMPL program:

```
option solver cplex;
param n;
param k;
param p;
set K := {1..k};
set I := {1..n};
param g {i in I};
param s {i in I, j in K};
param N {j in K};
set M {j in K} := {1..N[j]};
var x {j in K, i in M[j]} integer >=0 <=1;
var y {i in I} integer >=0 <=1;
maximize objective: sum {i in I} y[i];
subject to
const1 {j in K, i in M[j]: i<N[j]}: x[j,i] <= x[j,i+1];
const2 {i in I}: (1-2*g[i])*(sum {j in K} x[j, s[i,j]])
+((k-p)*g[i]+(p+1)*(1-g[i]))*y[i] <= k*(1-g[i]);
```

Note that if one is interested in a solution with continuous x variables, then one should just remove the definition "integer" from the variable definition of x.

The data file for the first example problem is:

```
param n=30;
param k=5;
param p=0;
param g :=
1 0
2 1
M
29 1
30 0;
param s :=
1 1 2
2 1 4
3 1 3
M
28 5 9
29 5 1
30 5 1;
param N :=
1 9
2 9
3 9
4 9
5 9;
```

REFERENCES

Bazerman, M. H. (1986). ***Judgement in Managerial Decision Making***. New York: Wiley.

Buck, L. S. (1977). ***Guide to the Setting of Appropriate Cutting Scores for Written Tests: A Summary of the Concerns and Procedures*** (Report No. TM-77-4). Washington, DC: U.S. Civil Service Commission.

Bureau of National Affairs. (1988). ***Recruiting and Selection Procedures*** (PFF Survey No. 146). Washington, DC: Bureau of National Affairs, Inc.

Cascio, W. F. (1991). ***Applied Psychology in Personnel Management*** (4th ed.). Engelwood Cliffs, NJ: Prentice-Hall.

Cascio, W. F., Alexander, R. A., & Barrett, G. V. (1988). "Setting Cutoff Scores: Legal, Psychometric, and Professional Issues and Guidelines." ***Personnel Psychology***, 44, 1-24.

Fourer, R., D.M. Gay, and B.W. Kernighan (1993). AMPL: ***A Modeling Language for Mathematical Programming***. San-Francisco: The Scientific Press

Gatewood, R. D., & Feild, H. S. (1994). ***Human Resource Selection*** (3rd ed.). Fort Worth, TX: The Dryden Press.

Karren, R. J., & Nkomo, S. M. (1988). "So you want to work for us." ***Personnel Administrator***, 38, 88-92.

Schmitt, N. W., & Klimoski, R. J. (1991). ***Research Methods in Human Resources Management.*** Cincinnati OH: South-Western Publishing.