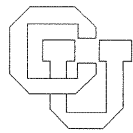


Software Aid for Optimizing 0-1 Matrices

Michael Perry*

CU-CS-242-82 December 1982



University of Colorado at Boulder

DEPARTMENT OF COMPUTER SCIENCE

* This research was supported by ONR grant number N00014-78-C-0433

ANY OPINIONS, FINDINGS, AND CONCLUSIONS OR RECOMMENDATIONS EXPRESSED IN THIS PUBLICATION ARE THOSE OF THE AUTHOR(S) AND DO NOT NECESSARILY REFLECT THE VIEWS OF THE AGENCIES NAMED IN THE ACKNOWLEDGMENTS SECTION.

Software Aids for Optimizing 0-1 Matrices
R. Michael Perry

Abstract

Here we consider the problem, given a real m -vector b and an integer $n < m$, of finding an $m \times n$ matrix A such that the least-squares residual norm, $\|b - Ax\|$ where $x = (A^T A)^{-1} A^T b$, is minimized, subject to the constraint that the entries of A must be 0's or 1's. This problem has arisen in a study of the retention of information from visual and verbal sources. Mathematically it is likely to be a difficult problem, however, and thus only "good" not optimal solutions are expected. A software package has been written to assist a human operator in searching for such desirable matrices. This is described here, and its use in the study is reviewed.

1. Introduction

In the standard linear least-squares problem we are given an m -dimensional vector b and an $m \times n$ matrix A , with $n < m$, and asked to find an n -dimensional vector x such that Ax approximates b in a least-squares sense, that is, so that the Euclidean 2-norm residual, $\|b - Ax\|$, is minimized. The solution vector x is given (ignoring possible conditioning problems with matrix A) by $x = (A^T A)^{-1} A^T b$. The resulting residual norm $\|b - Ax\|$ will be denoted by $r(b, A)$. It should be noted that computation of the solution x and the residual $r(b, A)$ are straightforward and can be performed reasonably efficiently, that is, in time that is a fairly small polynomial in the parameters m and n that determine the size of the problem.

Here we will consider the problem, given a real m -vector b and an integer $n < m$, of finding an $m \times n$ matrix A such that $r(b, A)$ is minimized, subject to the constraint that the entries of A must be 0's or 1's. This problem has arisen in a psychological study of the retention of information from visual and verbal sources [1]; a further discussion is given later. Mathematically the problem is known to be *NP*-complete in some forms [2]. Thus, unlike the least-squares problem, no efficient algorithms for solving it are known for most cases of interest and quite possibly none exist. That is, although the problem could be solved by exhaustively considering all the 2^{mn} possible 0-1 matrices with dimensions $m \times n$, it is doubtful if there exists any algorithm that always finds a best matrix in time that is polynomial in m and n . In practical terms the optimization problem is likely to be unsolvable even for small values of m and n (say, for $mn < 100$).

Often, however, one is not interested in an optimal matrix exclusively, but would simply like to find the best matrix possible, that is, the one with the smallest residual, subject to a reasonable limit on the time spent searching. Complex strategies using more basic heuristics may prove useful. Thus user interaction is desirable so that the search can be guided by intelligent decision-making.

In the work described here, several software tools were created for constructing and modifying 0-1 matrices. Among these are heuristics to find a matrix with low residual norm or to modify a previously found matrix to reduce the residual. Most of the procedures are straightforward and are not described in detail but are mentioned in the section on implementation. The section on algorithms that follows will deal with the two major heuristics that are used, both of which were suggested by Andrzej Ehrenfeucht.

2. Algorithms

There are two important algorithms used in generating low-residual matrices. One adds a "best" column, the other modifies the rows to reduce the residual. Thus to use these heuristics it is necessary to have a matrix to start with. In the application [1] it is assumed that the first column of any matrix is filled with 1's, so that an initial matrix is always given. Strictly speaking, this changes the optimization problem from what was defined above, since the first column of the matrix is not allowed to vary, but the modified problem is still likely to be intractable, so that similar considerations will apply.

To add a "best" column we start with an $m \times (n-1)$ matrix A_0 , assumed to contain a "constant" column filled with 1's, and look at the residual vector $v = b - A_0 x_0$ obtained by subtracting the least-squares approximation $A_0 x_0$ from b . The column to be added will be the one which, when linearly combined with the constant column, secures a least-squares fit to the residual vector. Thus in essence the column sought is a basis function which secures a best fit when linearly combined with a constant function. Since the column or basis function is limited in its values to 0 and 1, the fitting function obtained by linear combination must also be two-valued, though in this case the values can be arbitrary real numbers. The best column can thus be determined by finding the best two-valued approximation of the vector v . That is, if f is a two-valued function that secures a least-squares fit to v (over the set of all two-valued functions), then the desired column will be obtained if the i th entry of the column is set to 0 whenever $f(i)$ is the smaller value, and to 1 otherwise. (Or the 0 entries could correspond to the larger values $f(i)$ and the 1's to the smaller values. The existence of the constant column allows any other column in the matrix to be complemented without affecting the residual norm.)

To find the best-fitting two-valued function f , we first sort the entries of v in the order of increasing size. This will greatly simplify the problem, and the desired solution for the original case can then be determined by a straightforward unscrambling.

Assuming then that the entries v_i of v are sorted as indicated, that is, so that $v_j \geq v_i$ whenever $j \geq i$, it remains to determine a best-fitting function f_i . This will not be difficult once it is established that f too can be an increasing function, i.e., that for some best-fitting f , $f(j) \geq f(i)$ whenever $j \geq i$. To show this in turn, suppose that f is a function such that $f(j) \leq f(i)$ for some $j \geq i$. Then a function that fits just as well can be defined by interchanging the values of f so that $f(j)$ is assigned to i and $f(i)$ to j . In other words we claim that the discrepancy in fitting for the new function is no worse than that for the old, or that

$$(v_i - f(j))^2 + (v_j - f(i))^2 \leq (v_i - f(i))^2 + (v_j - f(j))^2. \quad (1)$$

To show this in turn, let $p=v_i$, $q=v_j$, $r=f(j)$, $s=f(i)$. Then $p \leq q$, $r \leq s$, and we claim that

$$(p-r)^2 + (q-s)^2 \leq (p-s)^2 + (q-r)^2. \quad (2)$$

By expanding terms the above inequality is equivalent to

$$pr + qs \geq ps + qr. \quad (3)$$

Clearly (2) holds when $p = r$ because for this case p is \leq both q and s , thus $|q-s| \leq |p-s|$; (3) must hold also. Next, suppose that (2) and (3) hold for some values p, q, r , and s . Then (3) must also hold for $p, q, r+t, s+t$, where t is an arbitrary real constant, as can be seen by expanding terms. From this it follows that (2) and (3) must hold for arbitrary p, q, r, s when the initial inequalities are satisfied. From this in turn we can assume without loss of generality that the function f is increasing.

Since, on the other hand, f has only two values it must have the form $f(j) = r$ whenever $1 \leq j \leq i$ for some $i < m$ and $f(j) = s$ for $i < j \leq m$. (Here we assume $m \geq 2$; also note that the case that v is constant, which will only occur if $v=0$ in view of the constant column, is handled by allowing $r=s=0$.) The best choice for f will be one that minimizes the discrepancy with v , which in turn is given by

$$\sum_{j=1}^i (v_j - r)^2 + \sum_{j=i+1}^m (v_j - s)^2. \quad (4)$$

For a given value i , the best choices of r and s are respectively the means of v_j over the intervals $1 \leq j \leq i$ and $i+1 \leq j \leq m$; thus

$$r = \frac{1}{i} \sum_{j=1}^i v_j; \quad s = \frac{1}{m-i} \sum_{j=i+1}^m v_j. \quad (5)$$

The best choice for f , then, is found by selecting the value i that achieves the minimum discrepancy according to (4), using the values for r and s given in (5). By expansion of terms the expression to be minimized becomes

$$\sum_{j=1}^m v_j^2 - \frac{1}{i} \left(\sum_{j=1}^i v_j \right)^2 - \frac{1}{m-i} \left(\sum_{j=i+1}^m v_j \right)^2, \quad (6)$$

which is convenient for computation.

The best f can then be decoded as indicated earlier (including unscrambling) to obtain a best column to add to matrix A_0 . In this way a matrix A can be built up column by column. Although the matrix will not in general be optimal the heuristic has proved highly useful, particularly when combined with other heuristics, the most important of which will now be described.

This heuristic modifies the rows of A in an attempt to find a better matrix. Initially we are given vector b , matrix A , and the least-squares solution vector x . Each entry b_i of b is approximated by taking the inner product of the i th row of A and the vector x . A better matrix will result if this i th row is replaced by another row whose inner product with x gives a better approximation to b_i . Rows of A can be modified independently of each other to find improvements in this way. The result (assuming some improvements are found) will be a matrix A' such that $\|b - A'x\| < \|b - Ax\|$. x , however, will not in general be the least-squares solution vector for A' ; this latter vector, call it x' , must then be computed and will give a still better fit to b . The heuristic can then be reapplied to the rows of A' using the new vector x' .

In practice the new rows are found simply by exhaustive searching. In particular, since the m rows of A can be modified independently of each other, there are only $m \cdot 2^n$ combinations that must be considered for the most general

case, rather than all the 2^{mn} possible matrices. Thus if n , the number of columns, is not too large the rows can be searched exhaustively to find the best A' . (This of course will not guarantee an optimal matrix but like the other heuristic it has proved useful.) In practice it has generally been desirable to limit the columns to be modified in searching for the best rows; for example a constant column has usually remained fixed. By suitably limiting the columns in this way, search times can be kept reasonable even when n is large.

3. Implementation

A software package has been created to assist a human operator in searching for 0-1 matrices with small residual norms. A number of user-defined constraints can be imposed on the matrices that are to be found. Thus a variety of problems can be defined and the search can be guided by intelligent decision-making.

To begin, then, a b -vector is written to a specified file; this vector, of course, will remain fixed during computation. The user will then attempt to find a 0-1 matrix A in the appropriate form having low residual $r(b, A)$. To assist this process there are (1) routines for defining A directly, that is, by setting entries individually or by such operations as redefining specified rows or columns, (2) a procedure for determining the least-squares vector x , given b and A , and (3) heuristics which automatically modify or add columns to A or which suggest other possible improvements. The software is extensively documented and should be usable without difficulty. The various components will now be described briefly. The documentation should be consulted for further details.

Matrix A must be written to a specified file so that the least-squares vector x and the residual $r(b, A)$ can be determined. (At present this file is fixed, as are those for the b -vector and for other information that may be needed, though this could easily be changed.) The matrix entries can be keyed in directly using one of the system's editors. In addition there is a "modify" routine in which instructions added to the matrix file are executed to change the entries. The available instructions include "delete", "replace", "insert", "union", and "complement", each of which performs the corresponding operation involving one or more columns of the matrix. ("Union" replaces a specified column with the bitwise "or" of two or more columns, while "complement" takes the bitwise complement of one column.) In addition a row of the matrix can be replaced using the "rerow" instruction. Use of the modify routine can greatly reduce the labor (and error) of making alterations in a matrix by hand.

When the b -vector and A -matrix have been set as desired, the routine "solve" can be called to determine the least-squares fitting vector x . This is calculated by a straightforward application of Cholesky decomposition using the normal-equations matrix $A^T A$, thereby obtaining $x = (A^T A)^{-1} A^T b$ [3]. The residual norm $r(b, A)$ is also derived. Although this method can lead to conditioning problems it is relatively fast and is stable in the cases of interest to date, that is, for nonsingular 0-1 matrices of fairly small dimensions. The solve routine also notes linearly dependent columns for a singular matrix.

Currently there are five other routines that solve the linear least-squares problem, "asolve", "bsolve", "csolve", "dsolve", and "nsolve". These, however, are used mainly to find a better matrix A with a smaller residual $r(b, A)$. asolve applies the first heuristic of the previous section, adding a best column to A . bsolve, using the same heuristic, adds columns iteratively until $r(b, A)$ falls below a specified tolerance. csolve modifies the rows of A using the second heuristic, printing the best solution and also alternate versions of rows that achieved a particularly good fit. dsolve is a simplified version of csolve in that

only two versions of each row are considered, namely, the original row and its complement. However it prints results of least-squares solving for each row when complemented individually. These results are sorted "best-first", that is, in the order of increasing residuals. In addition, all rows which, when complemented individually gave a better fit, are complemented simultaneously and the resulting solution and residual are printed. nsolve deletes columns of the matrix individually, solves the least-squares problem for each resulting matrix, and sorts the results best-first. In this manner columns which make relatively little contribution to the fitting can be identified and deleted, giving a smaller matrix with nearly the same residual.

The software package was coded in Franz LISP, under the UNIX operating system, and is now running on a VAX 11/780 computer at the University of Colorado Computer Science Department. LISP was found to be a convenient language for coding, particularly for the modify routine and for such features as dynamic allocation of arrays. It should be noted, however, that only small matrices have been considered (typically about 17×5) so that execution efficiency was not of primary concern. Typically about 30 sec. was required for one run of the modify routine, with 3-10 min. being the rule for one of the solving routines.

4. Application

In the one major application to date [1] a study was made of the retention of information in the human memory. Subjects were shown an educational movie, some being presented with both narration (or written text) and with the visual portion, while in other cases the verbal or visual component was omitted. Other subjects were given verbal information followed by visual presentation without sound, or vice versa. The subjects then were tested for retention of information, (1) immediately and (2) after a one-week delay. In this manner 17 test scores were obtained measuring the amount of information retained under varying conditions of acquisition and testing delay.

The next step was to find a sensible explanation of these results, and it seemed natural to interpret them in terms of features that were either definitely present or definitely absent in each of the 17 subject categories. One obvious feature of this type, for example, was the "visual" one that was present in those categories in which the visual portion of the movie was shown, and absent in the others.

By selecting the right set of features, then, it was hoped that every test score would be accounted for by the features present or absent in each particular group. That is, it was assumed that each feature would contribute a specific numerical amount to the test scores of all categories in which it was present, with zero contribution if absent. Each feature, then, would be assigned a value, positive or negative, by which it would affect a test score if present. Ideally, then, the test score of a particular category would be exactly reproduced by adding the values of the features that were present. This would require an apt choice of features and a correct assignment of values as well. It was recognized, however, that there should be some toleration of discrepancies, as for example, if the actual and calculated scores did not differ by a statistically significant amount.

In each case the values assigned for a particular choice of features were those that achieved a least-squares fit to the test scores. The problem then was to find a "reasonable" set of features that would give a reasonable least-squares fit. Mathematically, then, the 17 test scores formed an m -vector b with $m = 17$

while a set of n features comprised an $m \times n$ 0-1 matrix A , each feature contributing one column. The values of the features were then contained in the n -vector x obtained by least-squares fitting.

A reasonable solution of the problem would be a small number of physically meaningful features that gave a good fit to the 17 scores. The software package described in this report was used in searching for such a set of features, and finally five features were chosen that satisfied all requirements. (One of these was the "baseline" or constant column that figured in the previous section.) It is important to note, however, that, since the features had to be "physically meaningful" it was not sufficient to simply find a matrix with low residual. Instead there were further constraints that were difficult to delineate mathematically. Thus the human operator was crucial, both in finding solutions and in rejecting those that were unrealistic. At any rate, a satisfactory solution was eventually obtained, and the paper's conclusions could then be stated. Among these was the interesting observation that "in a show and tell presentation, one should not tell first and show second".

The results, then, were obtained by a lengthy interaction of person and machine, both of which were indispensable.

REFERENCES

- [1] Baggett, P. and Ehrenfeucht, A., Encoding and retaining information in the visuals and verbals of an educational movie. Technical Report #108-ONR (Sept., 1981, also to appear in Educational Communication and Technology).
- [2] Ehrenfeucht, A., private communication.
- [3] Lawson, C., and Hanson, R., Solving least squares problems, Prentice-Hall, Inc., Englewood Cliffs, N. J., 1974.

Navy

- 1 Robert Ahlers
Code N711
Human Factors Laboratory
NAVTRAEQUIPCEN
Orlando, FL 32813
- 1 Dr. Richard Cantone
Navy Research Laboratory
Code 7510
Washington, DC 20375
- 1 Dr. Stanley Collyer
Office of Naval Technology
800 N. Quincy Street
Arlington, VA 22217
- 1 Dr. Jim Hollan
Code 304
Navy Personnel R & D Center
San Diego, CA 92152
- 1 Dr. Ed Hutchins
Navy Personnel R&D Center
San Diego, CA 92152
- 1 Dr. Peter Kincaid
Training Analysis & Evaluation Group
Dept. of the Navy
Orlando, FL 32813
- 1 Dr William Montague
NPRDC Code 13
San Diego, CA 92152
- 1 Library, Code P201L
Navy Personnel R&D Center
San Diego, CA 92152
- 1 Technical Director
Navy Personnel R&D Center
San Diego, CA 92152
- 6 Commanding Officer
Naval Research Laboratory
Code 2627
Washington, DC 20390
- 1 Office of Naval Research
Code 433
800 N. Quincy SStreet
Arlington, VA 22217

Navy

- 6 Personnel & Training Research Group
Code 442PT
Office of Naval Research
Arlington, VA 22217
- 1 Psychologist
ONR Branch Office
1030 East Green Street
Pasadena, CA 91101
- 1 Office of the Chief of Naval Operations
Research Development & Studies Branch
OP 115
Washington, DC 20350
- 1 LT Frank C. Petho, MSC, USN (Ph.D)
CNET (N-432)
NAS
Pensacola, FL 32508
- 1 Dr. Alfred F. Smode, Director
Training Analysis & Evaluation Group
Dept. of the Navy
Orlando, FL 32813
- 1 Mr. Brad Sympton
Navy Personnel R&D Center
San Diego, CA 92152
- 1 Dr. Edward Wegman
Office of Naval Research (Code 411S&P)
800 North Quincy Street
Arlington, VA 22217
- 1 Roger Weissinger-Baylon
Department of Administrative Sciences
Naval Postgraduate School
Monterey, CA 93940
- 1 Mr John H. Wolfe
Navy Personnel R&D Center
San Diego, CA 92152

Marine Corps

- 1 H. William Greenup
Education Advisor (E031)
Education Center, MCDEC
Quantico, VA 22134
- 1 Special Assistant for Marine
Corps Matters
Code 100M
Office of Naval Research
800 N. Quincy St.
Arlington, VA 22217
- 1 DR. A.L. SLAFKOSKY
SCIENTIFIC ADVISOR (CODE RD-1)
HQ, U.S. MARINE CORPS
WASHINGTON, DC 20380

Army

- 1 Technical Director
U. S. Army Research Institute for the
Behavioral and Social Sciences
5001 Eisenhower Avenue
Alexandria, VA 22333
- 1 Dr. Robert Sasmor
U. S. Army Research Institute for the
Behavioral and Social Sciences
5001 Eisenhower Avenue
Alexandria, VA 22333
- 1 Dr. Robert Wisher
Army Research Institute
5001 Eisenhower Avenue
Alexandria, VA 22333

Air Force

Department of Defense

- 1 AFHRL/LRS
Attn: Susan Ewing
WPAFB
WPAFB, OH 45433
- 1 U.S. Air Force Office of Scientific
Research
Life Sciences Directorate, NL
Bolling Air Force Base
Washington, DC 20332
- 1 Air University Library
AUL/LSE 76/443
Maxwell AFB, AL 36112
- 1 Mr. Raymond E. Christal
AFHRL/MOE
Brooks AFB, TX 78235
- 1 Dr. David R. Hunter
AFHRL/MO
Brooks AFB, TX 78235
- 1 Dr. Joseph Yasatuke
AFHRL/OT
Williams AFB, AZ 58224
- 12 Defense Technical Information Center
Cameron Station, Bldg 5
Alexandria, VA 22314
Attn: TC
- 1 Military Assistant for Training and
Personnel Technology
Office of the Under Secretary of Defense
for Research & Engineering
Room 3D129, The Pentagon
Washington, DC 20301

Civilian Agencies

1 Chief, Psychological Reserch Branch
U. S. Coast Guard (G-P-1/2/TP42)
Washington, DC 20593

1 Dr. Joseph L. Young, Director
Memory & Cognitive Processes
National Science Foundation
Washington, DC 20550

Private Sector

1 Dr. John R. Anderson
Department of Psychology
Carnegie-Mellon University
Pittsburgh, PA 15213

1 Psychological Research Unit
Dept. of Defense (Army Office)
Campbell Park Offices
Canberra ACT 2600
AUSTRALIA

1 Dr. Alan Baddeley
Medical Research Council
Applied Psychology Unit
15 Chaucer Road
Cambridge CB2 2EF
ENGLAND

1 Dr. Jonathan Baron
80 Glenn Avenue
Berwyn, PA 19312

1 Dr. John Black
Yale University
Box 11A, Yale Station
New Haven, CT 06520

1 Dr. John S. Brown
XEROX Palo Alto Research Center
3333 Coyote Road
Palo Alto, CA 94304

1 Dr. Jeffrey Elman
University of California, San Diego
Department of Linguistics
La Jolla, CA 92093

1 ERIC Facility-Acquisitions
4833 Rugby Avenue
Bethesda, MD 20014

1 Professor Reuven Feuerstein
HWCRI Rehov Karmon 6
Bet Hakerem
Jerusalem
Israel

1 Mr. Wallace Feurzeig
Department of Educational Technology
Bolt Beranek & Newman
10 Moulton St.
Cambridge, MA 02238

Private Sector

- 1 Dr. Dexter Fletcher
WICAT Research Institute
1875 S. State St.
Orem, UT 22333
- 1 Dr. John R. Frederiksen
Bolt Beranek & Newman
50 Moulton Street
Cambridge, MA 02138
- 1 Dr. Bert Green
Johns Hopkins University
Department of Psychology
Charles & 34th Street
Baltimore, MD 21218
- 1 Dr. Earl Hunt
Dept. of Psychology
University of Washington
Seattle, WA 98105
- 1 Dr. Steven W. Keele
Dept. of Psychology
University of Oregon
Eugene, OR 97403
- 1 Dr. David Kieras
Department of Psychology
University of Arizona
Tuscon, AZ 85721
- 1 Dr. Stephen Kosslyn
Department of Psychology
Brandeis University
Waltham, MA 02254
- 1 Dr. Pat Langley
Carnegie-Mellon University
Pittsburgh, PA 15213
- 1 Dr. Marcy Lansman
The L. L. Thurstone Psychometric
Laboratory
University of North Carolina
Davie Hall 013A
Chapel Hill, NC 27514
- 1 Dr. Jim Levin
University of California
at San Diego
Laboratory fof Comparative
Human Cognition - D003A
La Jolla, CA 92093

Private Sector

- 1 Dr. Michael Levine
Department of Educational Psychology
210 Education Bldg.
University of Illinois
Champaign, IL 61801
- 1 Dr. Charles Lewis
Faculteit Sociale Wetenschappen
Rijksuniversiteit Groningen
Oude Boteringestraat 23
9712GC Groningen
Netherlands
- 1 Dr. Jay McClelland
Department of Psychology
MIT
Cambridge, MA 02139
- 1 Dr. James A. Paulson
Portland State University
P.O. Box 751
Portland, OR 97207
- 1 Dr. Ernst Z. Rothkopf
Bell Laboratories
Murray Hill, NJ 07974
- 1 Dr. David Rumelhart
Center for Human Information Processing
Univ. of California, San Diego
La Jolla, CA 92093
- 1 Dr. William Stout
University of Illinois
Department of Mathematics
Urbana, IL 61801
- 1 Dr. Kikumi Tatsuoka
Computer Based Education Research Lab
252 Engineering Research Laboratory
Urbana, IL 61801
- 1 Dr. Douglas Towne
Univ. of So. California
Behavioral Technology Labs
1845 S. Elena Ave.
Redondo Beach, CA 90277
- 1 Dr. Benton J. Underwood
Dept. of Psychology
Northwestern University
Evanston, IL 60201

Private Sector

- 1 Dr. Kurt Van Lehn
Zerex PARC
3333 Coyote Hill Road
Palo Alto, CA 94304
- 1 DR. GERSHON WELTMAN
PERCEPTRONICS INC.
6271 VARIEL AVE.
WOODLAND HILLS, CA 91367
- 1 Dr. Keith T. Wescourt
Perceptronics, Inc.
545 Middlefield Road, Suite 140
Menlo Park, CA 94025
- 1 DR. SUSAN E. WHITELY
PSYCHOLOGY DEPARTMENT
UNIVERSITY OF KANSAS
Lawrence, KS 66045
- 1 William B. Whitten
Bell Laboratories
2D-610
Holmdel, NJ 07733
- 1 Dr. Mike Williams
Zerex PARC
3333 Coyote Hill Road
Palo Alto, CA 94304

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Software Aids for Optimizing 0-1 Matrices		5. TYPE OF REPORT & PERIOD COVERED Technical Report
		6. PERFORMING ORG. REPORT NUMBER ONR
7. AUTHOR(s) R. Michael Perry		8. CONTRACT OR GRANT NUMBER(s) N00014-78-C-0433
9. PERFORMING ORGANIZATION NAME AND ADDRESS Department of Computer Science Campus Box 430 University of Colorado, Boulder, CO 80309		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS NR 157-422
11. CONTROLLING OFFICE NAME AND ADDRESS Personnel and Training Research Programs Office of Naval Research, Code 458 Arlington, VA 22217		12. REPORT DATE November 1982
		13. NUMBER OF PAGES 18
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report)		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Least-squares, fitting, 0-1 matrices, vectors, algorithms, heuristics, optimization, NP-completeness, software		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) Here we consider the problem, given a real m -vector b and an integer $n < m$, of finding an $m \times n$ matrix A such that the least-squares residual norm, $\ b - Ax\ $ where $x = (A^T A)^{-1} A^T b$, is minimized, subject to the constraint that the entries of A must be 0's or 1's. This problem has arisen in a study of the retention of information from visual and verbal sources. Mathematically it is likely to be a difficult problem, however, and thus only "good" not optimal solutions are expected. A software package has been written to assist a human operator in searching for such desirable matrices. This is described here, and its use in the study is reviewed.		