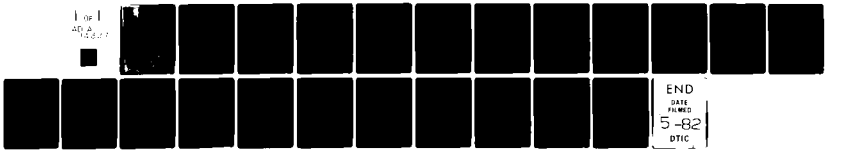
AD-A114 327 TEXAS UNIV AT AUSTIN CENTER FOR CYBERNETIC STUDIES F/G 12/1
THE BEST PARAMETER SUBSET USING THE CHEBYCHEV CURVE FITTING CRI--ETC(U)
SEP 81 A ARMSTRONG, P BECK N00014-75-C-0569
UNCLASSIFIED CCS-412 NL



END
DATE FILMED
5-82
DTIC

CENTER FOR CYBERNETIC STUDIES

The University of Texas
Austin, Texas 78712

(12)

Research Report CCS 412

THE BEST PARAMETER SUBSET USING THE CHEBYCHEV CURVE FITTING CRITERION

by

R. Armstrong*
P. Beck**

September 1981

*The University of Georgia
**The University of Arizona

This research was partly supported by ONR Contract N00014-75-C-0569 with the Center for Cybernetic Studies, The University of Texas at Austin.

CENTER FOR CYBERNETIC STUDIES

A. Charnes, Director
Business-Economics Building, 203E
The University of Texas at Austin
Austin, TX 78712
(512) 471-1821

ABSTRACT

The Chebychev (also Minimax and L^{∞} Norm) criterion has been widely studied as a method for curve fitting. Published computer codes are available to obtain the optimal parameter estimates to fit a linear function to a set of given points under the Chebychev criterion. The purpose of this paper is to study procedures for obtaining the best subset of k parameters from a given set of m parameters where k is less-than-or-equal-to m.

KEY WORDS

Least Absolute Value

Regression

Linear Programming

Best Subset

Introduction

The classical linear curve fitting problem in the L_p norm can be stated as follows. Given $(y_i, x_{i1}\ x_{i1}, \ldots, x_{im})$, $i = 1, 2, \ldots, n$, determine β to solve the problem

$$\text{Minimize } (\sum_{i=1}^{n} |y_i - \sum_{j=1}^{m} x_{ij}\ \beta_j|^p)^{1/p}$$

When p=2, the problem is least squares, when p=1, the problem is least absolute values and when lim $p \to \infty$, the problem is minimize the maximum value which is also called the Chebychev curve fitting problem. A comparison of these three criteria can be found in [1, 2]. Least squares (L_2 norm) is certainly the most popular approach in the statistical community, although under certain conditions least absolute value and Chebychev criteria are preferred. Least absolute values provides a maximum liklihood estimate when the errors have a double-exponential distribution and works well empirically when outliers are present in the data (see Gentle [10] and Dielman and Pfaffenberger [7]). The Chebychev estimates are maximum likelihood when errors are uniformly distributed and work well empirically with most fat-tailed error distributions (see Appa and Smith [1]) and Rabinowitz [13]). This paper will present computational procedures to solve a best subset problem under Chebychev criterion. The problem can be stated as follows.

For q=k, k+1, ..., m, determine values for β and J which

$$\text{Minimize } \{\text{Maximum}|y_i - \sum_{j \in J} x_{ij}\ \beta_j|\},\ [J]=q \qquad (1)$$

where $J \subseteq \{1, 2, \ldots, m\}$ and [J] is the cardinality of the index set J. In other words, consider all possible combinations of exactly q parameters taken

from the original m parameters, and choose the combination yielding the smallest maximum absolute deviation.

The best subset problem arises frequently in statistical analysis where it is desirable to recognize influential variables and study the effect of reducing the number of variables. Draper and Smith [8] give an excellent overview of this modeling technique. Solution algorithms exist for the best subset problem when the least absolute value [5, 12] and least squares [11] criteria are used; however, there appear to be no algorithms available in the public domain when the Chebychev criterion is used.

The algorithm for the best subset problem in the Chebychev norm uses a branch-and-bound technique. A binary tree is formed and each node of the tree corresponds to a curve fitting problem with a specified set of parameters included in the model. Problems are solved using the algorithm of Armstrong and Kung [3, 4]; however, because of available bounds, not all problems need be solved to optimality. The framework of the enumeration is similar to that given by Armstrong and Kung [5] for the best subset least absolute value problem. This framework is reviewed in the next section and placed in the Chebychev curve fitting context.

Algorithimic Framework

At any stage of the enumeration procedure a problem of the form given by (1) is being considered. Rewriting (1) in a linear programming equivalent statement yields:

Minimize Z

subject to

$$y_i - Z \leq \sum_{j \in J} X_{ij} \beta_j \leq y_i + Z, \quad i=1, 2, \ldots, n \qquad (2)$$

where, at optimality, z will have the value of the maximum absolute residual.

The linear programming dual of (2) is the following.

$$\text{Maximize } W = \sum_{i=1}^{n} y_i \, \pi_i' + \sum_{i=1}^{n} y_i \, \pi_i'' \tag{3}$$

subject to

$$\sum_{i=1}^{n} x_{ij} \, \pi_i' + \sum_{i=1}^{n} x_{ij} \, \pi_i'' = 0 \qquad j \in J$$

$$\sum_{i=1}^{n} \pi_i' - \sum_{i=1}^{n} \pi_i'' = 1$$

$$\pi_i' \geq 0, \; \pi_i'' \leq 0, \; i=1, 2, \ldots, n$$

It is easily shown that (2) and (3) will always have finite optimal objective values which are equal. Also, the simplex algorithm for linear programming problems will readily provide optimal π values for (3) once (2) is solved and, similarly, the optimal β values are available once (3) is solved. Thus, computational considerations alone should determine whether (2) or (3) should be solved with a primal simplex algorithm.

Special purpose simplex algorithms have been developed for both (2) and (3). Computational experience [13] has indicated that algorithms that maintain a feasible solution to (3) are superior to those that maintain a feasible solution to (2). The algorithm of Armstrong and Kung [3, 4] will be used to solve (3). This uses a reduced basis, may pass through more that one extreme point during an iteration and has a reduced ratio test. The objective value is

monotonically nondecreasing from iteration to iteration and this characteristic is particularly attractive when solving the best subset problem.

An outline of a step-by-step solution procedure for the best subset problem will now be stated which is independent of the method used to solve (3).

STEPS OF ALGORITHM

STEP 1. Set $q = m$; $z_i = \infty$, $i = k, \ldots, m$; $\ell = 0$; $J = \{1, 2, \ldots, m\}$, and $STAT_j = 0$, $j = 1, 2, \ldots, m$

STEP 2. Solve (2) to obtain an optimal solution $(\bar{z}, \bar{\beta})$, where $\bar{\beta}_j = 0$ for $j \notin J$. If $\bar{z} > SAD_q$, then go to STEP 4; otherwise, go to STEP 3.

STEP 3. A better solution has been found for a subset with q parameters included in the model. Set $z_q = \bar{z}$ and save $\bar{\beta}$.

STEP 4. If $q \leq k$ then go to STEP 6, otherwise, set $q \leftarrow q - 1$ and $\ell \leftarrow \ell + 1$.

STEP 5. Find a parameter β_u with $STAT_u = 0$ and form a new subproblem with $\beta_u = 0$. Set $STAT_u = -1$ and remove u from J. Go to STEP 2.

STEP 6. If $PAR_\ell > 0$ then go to STEP 7; otherwise, set $PAR_\ell \leftarrow PAR_\ell \cdot j = PAR_\ell$, $STAT_j = \ell$ and $q \leftarrow q + 1$. Go to STEP 4.

STEP 7. Set $J = PAR_\ell$, $STAT_j = 0$ and $\ell \leftarrow \ell - 1$. If $\ell > 0$ then go to STEP 6; otherwise, terminate the enumeration process.

VARIABLE DEFINITIONS

q The number of parameters in the current subproblem.

z_i The current best objective with i parameters.

$$STAT_j = \begin{cases} 0 & \text{j-th parameter is in the model but has not been forced in, i.e., a free parameter} \\ 1 & \text{j-th parameter is forced in the model} \\ -1 & \text{j-th parameter is forced out of the model} \end{cases}$$

PAR_ℓ The parameter restricted at level ℓ of the predecessor path. If PAR_ℓ is negative, the parameter is forced out of the model and if PAR_ℓ is positive, the parameter is forced in the model.

ℓ The current level in the solution tree. The initial problem is at level zero and a node is one level deeper in the tree than the immediate predecessor.

One trivial modification to the algorithm is to force a β_r to be included in every regression. This can be accomplished by setting $STAT_r$ equal to 1 rather than 0 at STEP 1.

The optimal solution of (1) is not used when $\bar{z} \geq z_q$; thus it is not necessary to solve (1) if it can be ascertained that $\bar{z} \geq z_q$ by another test. This additional test is easily implemented when (3) is solved using a primal simplex algorithm. The algorithm will maintain a feasible solution to (3) and the objective value (w) will be monotonically nondecreasing from iteration to iteration. Therefore, the solution process can be terminated whenever the following holds.

$$w \geq z_q \tag{4}$$

A key aspect in any branch-and-bound algorithm is the sequence in which the subproblems are considered. The sequence should be based on the following guidelines.

A) A good solution for a subproblem at any stage should be easily obtained to facilitate the solution of the subproblem.

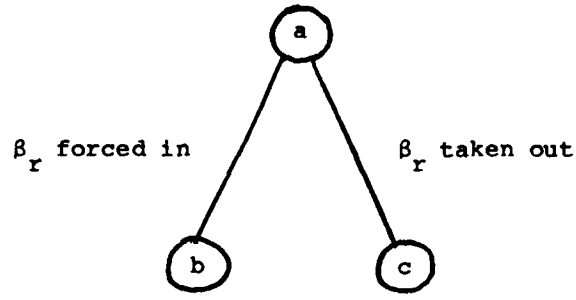
B) A good solution for each subset size should be obtained as soon as possible to maximize the influence of condition (4).

The next section discusses how to implement the branch-and-bound algorithm to accommodate guidelines A and B.

PENALTY CALCULATIONS AND AN ADVANCED START

The previously described branch-and-bound algorithm uses a last-in-first-out (LIFO) branching rule. Viewing the algorithm in a tree format, every node corresponds to a linear programming problem. Two problems are formed by

considering the "current" problem, and



removing a parameter from the model on one branch and forcing the same parameter to be included in the model on the other branch. Once a condition is specified, it must be satisfied in all descendants of the node.

The branch were β_r is forced in the model gives rise to the same linear programming problem in the immediate predecessor node. Thus, the problem at node b of the diagram need not be solved. The problem of concern arises when some β_r is removed from the model. Let (3) represent the problem at node a. Setting $\beta_r = 0$ in (2) is the same as removing the r-th constraint from (3). The problem at node c written by modifying problem at node a is the following.

$$\text{Maximize } w = \sum_{i=1}^{n} y_i \pi_i' + \sum_{i=1}^{n} y_i \pi_i'' \tag{5}$$

subject to

$$\sum_{i=1}^{n} x_{ij} \pi_i' + \sum_{i=1}^{n} x_{ij} \pi_i'' = 0, \quad j \in J, \; j \neq r$$

$$\sum_{i=1}^{n} \pi_i' - \sum_{i=1}^{n} \pi_i'' = 1$$

$$\sum_{i=1}^{n} x_{ir} \pi_i' + \sum_{i=1}^{n} x_{ir} \pi_i'' + S_r = 0$$

$$\pi_i' \geq 0, \; \pi_i'' \leq 0, \; i=1, 2, \ldots, n$$

The logical linear programming variable S_r is unrestricted in sign and, hence, has the effect of eliminating the r-th constraint. Also, S_r will always appear in the basis of an optimal solution to (6). Given a basic feasible solution to the problem at node a, a basic feasible solution to the problem at node c can be formed by "conceptually" performing a simplex iteration to bring S_r into the basis. Once S_r is brought into the basis the r-th constraint is dropped from the problem and the structure is given by (3). The index set J used at node c is formed by taking the index set J used at node a and removing the index r.

The objective function change incurred by bringing S_r into the basis provides a penalty on the restriction $\beta_r = 0$. In other words, this is a lower bound on the total objective change when going from node a to node c.

The penalty can be calculated using a reduced basis from the immediate predecessor and a modification of the ratio procedures described in [4]. A formal statement of the penalty procedures will not be given as it requires

excessive notation and a firm knowledge of the algorithm in [4]. For purposes of the presentation here it is necessary to realize that the penalty calculation provides the following two important pieces of information.

1. The objective change during the iteration which brings S_r into the basis is determined.
2. The dual variable (π) to leave the basis during this iteration is determined. The feasibility of (3) is maintained after S_r enters the basis.

(In the computer implementation of the algorithm, S_r is never created explicitly, rather, the dimension of the basis is decreased by one.)

The information obtained from the penalty calculations can be used to develop the solution tree. This topic will be discussed in the next section.

Implementation and Computational Testing

The algorithm for the best parameter subset using a Chebychev curve fitting criterion was coded in FORTRAN and various implementations were tested. The initial implementation had the following characteristics.

1. The parameter chosen to restrict at a node was the free parameter with the smallest index.
2. Each subproblem was solved to optimality without utilizing any information from the problems solved at preceding nodes.

The first variation made to the algorithm was to drop the requirement that each subproblem be solved. If the objective value of the current subproblem was not less than the best objective value found thus far for the associated subset size, then the algorithm returned to the branching process. The comparison was made immediately before updating the linear programming basis.

The solution time was cut by more than one half by this simple check. Thus, all future testing included this feature.

The second variation was to use the last solution from the immediate predecessor as a starting solution for the current subproblem. The procedure outlined in the previous section was implemented to determine the variable to remove from the basis when a constraint is removed from (3). This required saving the indices of the variables in the final basis of each subproblem so the LU decomposition [6] of this basis could be reconstructed.

Since the algorithm used a last-in-first-out branching rule, the reconstruction of the basis was only necessary when backtracking took place. This advanced start also cut solution times by more than one half and was included in all future versions.

The final alterations to the algorithm involved the use of the penalties to guide in the construction of the solution tree. It was hypothesized that the maximum benefit from comparing the objective value of current subproblem against the incumbent objective value would be derived by obtaining the best solutions early in the enumeration process. Thus, assuming the objective change during the first pivot reflected the overall objective change, the free parameter with the smallest penalty should be chosen to be restricted.

Table 1 gives a comparison of run times for solving a set of randomly generated problems with the smallest penalty and first penalty branding rule implementation. All problems were randomly generated with the errors having a uniform distribution and solved on the 170/750 Dual Cyber at the University of Texas at Austin. All times are reported for solution only and do not include input-output. The iteration count reported gives for iterations within the algorithm of [3] for solving the subproblems. All variables were single precision with the Cyber's 60-bit word and the tolerance value for zero was

set at 1.E-8. Choosing to restrict the free parameter with smallest penalty was, overall, not as good a strategy as choosing the first free parameter. The superiority of the first free parameter rule became more pronounced as the problem dimensions increased. It is felt that the poor performance of the smallest penalty rule came from the extra work required to determine the parameter to restrict and from the purely local information given by the penalty. A similar result has been observed in the penalties from integer programming [9].

The next phase of testing considered the effect of limiting the smallest subset size and not requiring the verification of optimality. Table 2 shows the results with the two larger values of m and a solution within 95, 98 or 100 percent of optimality guaranteed. The use of the smallest penalty branching rule did not seem to provide any better suboptimal solutions than the first free parameter option for this problem size. However, for the smaller dimension problems the smallest penalty rule frequently provided the optimal solutions. Table 3 shows the effect of limiting the number of parameters in the smallest subset (k) to 5, 10 and 15 rather than 1. The growth of solution times is approximately exponential with the decrease in the value of k. This is to be expected because of the tree search strategy.

The final computational results displayed in tabular form compares the solution times for the Chebychev best subset problem with times for the least absolute value subset problem. The algorithm of [5], called L1LU, was used for the least absolute value problems. The code L1LU was consistent in required close to three times the CPU seconds than the first free parameter code for the Chebychev norm problem. This time differential is similar to that observed for solving a single curve fitting problem with the two norms.

Other branching strategies were tested without any notable results. The rules attempted were the following.

1. The parameter chosen to restrict was the one yielding the largest penalty.
2. During the first descent of the tree, the parameter chosen to restrict yielded the smallest penalty, thereafter, the parameter yielding the largest penalty was chosen.
3. The pseudo-cost procedure of [9] was modified to the problem at hand and used to choose the parameter the restrict.

The maximum penalty performed poorly and the other two strategies were at times better than the smallest penalty; however, the first free parameter branching rule remained the best.

Conclusions

This paper has presented an algorithm for the best parameter subset using a Chebychev curve fitting criterion. Computational results with variations of the fundamental branch-and-bound procedure indicate that the use of penalties to develop the tree is not worth the additional labor for most problems. Solution times grow exponentially with the number of parameters but show a slow linear growth based on the number of observations. The largest problem solved during the study had 20 parameters, 300 observations and the smallest subset size considered was 10. It seems, at this time, prohibitive to consider the smallest subset size to be one and determine the best subset for problems with m greater-than-or-equal-to 20.

One modification that would certainly increase the speed of the algorithm is to save the complete LU decomposition at each node rather than just the indices of columns in the basis. For large problems, a significant amount of

time is spent reconstructing previously obtained LU decompositions. The additional storage required to save previous LU decompositions would, however, limit the size of problems that could be solved. In our implementations it was felt that the savings of space was more important than the savings in time.

Curve fitting with a Chebychev criterion is often a desirable alternative to other curve fitting criteria. The ability to analyze the best parameter subsets using the Chebychev criterion is provided by the algorithm presented here. Although this paper has only dealt with algorithmic procedures for the best subset problem, the foundation for simulation and empirical studies of curve fitting problems is made available.

The computer code version of the algorithm presented in this paper is available from the authors.

TABLE 1

	n = 200		n = 250		n = 300	
	Smallest Penalty	First Penalty	Smallest Penalty	First Penalty	Smallest Penalty	First Penalty
m = 5	.09 (21)	.12 (27)	.14 (24)	.13 (26)	.15 (23)	.16 (24)
m = 10	2.42 (154)	2.23 (186)	3.18 (240)	2.78 (205)	4.12 (281)	3.44 (251)
m = 15	104 (2921)	79 (1504)	151 (5805)	136 (5515)	156 (4718)	99 (2124)

A computational comparison of two implementations of the best subset algorithm for the Chebychev norm is given. The upper entry in each cell is the mean CPU time in seconds and the lower entry is mean number of iterations. Three problems were solved with each combination of m and n when m equals 5 or 10, a single problem was solved when m equals 15.

TABLE 2

	m = 15, k = 1		m = 20, k = 10
	Smallest Penalty	First Penalty	First Penalty
95%	84 (1072)	74 (808)	2195 (1083)
98%	121 (2793)	85 1299)	2415 (6400)
100%	156 (4718)	99 (2124)	2678 (10244)

A computational comparison of two implementations of the best subset algorithm for the Chebychev norm with three percentages of optimality guaranteed is given. The upper entry in each cell is the CPU time in seconds and the lower entry is the number of iterations. All problems had the value of n set at 300.

TABLE 3

	m = 15		m = 20	
	Smallest Penalty	First Penalty	Smallest Penalty	First Penalty
k = 5	130 (5205)	90.3 (2285)	DNR	DNR
k = 10	36.1 (1391)	25 (660)	DNR	2678 (10244)
k = 15	.28 (35)	.28 (35)	201 (96)	174 (452)

DNR = Did Not Run

A computational comparison of two implementations of the best subset algorithm for the Chebychev norm with three minimum set sizes is given. The upper entry in each cell is CPU time in seconds and the lower entry is number of iterations. All problems had the value of n set at 300 and 100% of optimality guaranteed.

TABLE 4

	n = 200			n = 300		
	Smallest Penalty	First Penalty	L1LU	Smallest Penalty	First Penalty	L1LU
m = 10	2.42 (154)	2.23 (186)	7.11 (3410)	4.12 (218)	3.44 (251)	10.23 (3987)
m = 15	104 (2921)	79 (1504)	145 (13515)	156 (4718)	99 (2124)	308 (69271)

An algorithm for the best subset least absolute value problem is compared against an algorithm for the best subset Chebychev norm problem. The upper entry in each cell is mean CPU time and the lower entry is mean number of iterations. Three problems were solved when m equaled 10 and a single problem when m equals 15. All solutions had 100% of optimality was guaranteed.

REFERENCES

1. G. Apa and C. Smith, On L_1 and Chebychev Estimation, Mathematical Programming 5, 1973, 73-87.

2. Armstrong, R.D., E.L. Frome and M.G. Sklar, "Linear Programming in Exploratory Data Analysis," Journal of Educational Statistics, 1980, Vol. 5, No. 4, 293k-307.

3. Armstrong, R.D. and D.S. Kung, "Min-Max Estimates for a Linear Multiple Regression Problem," Applied Statistics, Vol. 28, No. 1 (1979), pp. 93-100.

4. R.D. Armstrong and D.S. Kung, A Dual Method for Discrete Chebychev Curve Fitting, Mathematical Programming 19, 1980, 186-199.

5. Armstrong, R.D. and M.T. Kung, "An Algorithm to Select the Best Subset for a Least Absolute Value Regression Problem," Center for Cybernetic Studies Report No. 396, May 1981.

6. R.H. Bartels and G.H. Golub, The Simplex Method of Linear Programming Using LU Decomposition. Communications of the ACM 12, 1969, 266-268.

7. Dielman, T. and R. Pfaffenberger, "LAV (Least Absolute Value) Estimation in Linear Regression: A Review," Optimization in Statistics Volume, TIMS Studies of the Management Sciences to appear.

8. Draper, N.R. and H. Smith, Applied Regression Analysis, (1966) New York, John Wiley and Sons, Inc.

9. Gauthier, J.M. and G. Ribière," Experiments in Mixed-Integer Linear Programming Using Pseudo-costs," Mathematical Programming, Vol. 12 (1977), 26-47.

10. Gentle, J.E., "Least Absolute Values Estimation: An Introduction," Communications in Statistics, Simulation and Computation, 1977, B6 (4), 313-328.

11. Kennedy, W.J. and J.E. Gentle (1980) Statistical Computing, New York, Marcel Dekker.

12. Narulo, S. and J.F. Wellington, "Selection of Variables in Linear Regression Using the Minimum Sum of Weighted Absolute Errors Criterion," Technometrics, Vol. 21, (1979), 299-306.

13. P. Rabinowitz, Application of Linear Programming to Numberical Analysis, SIAM REV 10, 1968, 121-159.

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE	READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER: CCS 412 / 2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle): The Best Parameter Subset Using the Chebychev Curve Fitting Criterion.	5. TYPE OF REPORT & PERIOD COVERED
	6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s): R. Armstrong and P. Beck	8. CONTRACT OR GRANT NUMBER(s): N00014-75-C-0569
9. PERFORMING ORGANIZATION NAME AND ADDRESS: Center for Cybernetic Studies, UT Austin, Austin, Texas 78712	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS: Office of Naval Research (Code 434), Washington, D. C.	12. REPORT DATE: September 1981
	13. NUMBER OF PAGES: 21
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)

This document has been approved for public release and sale; its distribution is unlimited.

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 absolute value, regression, linear programming, best subset.

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The Chebychev (also Minimax and L_∞ Norm) criterion has been widely studied as a method for curve fitting. Published computer codes are available to obtain the optimal parameter estimates to fit a linear function to a set of given points under the Chebychev criterion. The purpose of this paper is to study procedures for obtaining the best subset of k parameters from a given set of m parameters where k is less-than-or-equal-to m.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 68 IS OBSOLETE
S/N 0102-014-6601

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)