J. Statist. Comput. Simul., 1982, Vol. 15, pp. 17–25
0094-9655/82/1501-0017 $06.50/0

Printed in Great Britain

Algorithmic Complexity: Three *NP*-Hard Problems in Computational Statistics

WILLIAM J. WELCH†

Department of Mathematics, Imperial College, London SW7 2BZ, Great Britain

(*Received September 23, 1980*)

A theory of algorithmic complexity from computer science allows one to examine the properties of algorithms for the solution of computational problems and, in particular, the relationship between the size of an instance of the generic problem and the time required for computation. The theory identifies certain problems as *NP*-complete or *NP*-hard, two classes of problems thought to be intractable. We show that three problems from computational statistics are *NP*-hard: cluster analysis, subset selection in regression and *D*-optimal exact design of experiments.

KEY WORDS: Algorithm, algorithmic complexity, *NP*-complete, *NP*-hard, cluster analysis, regression, subset selection, design of experiments, *D*-optimal.

1. INTRODUCTION

Consider the following scenario. An algorithm has been devised to solve a computational problem, for definiteness say clustering of entities with some optimality criterion, and a program has been implemented which we assume is correct. The algorithm has successfully solved several test examples but, when presented with a more realistic, practical-sized case with many entities, the program will not complete execution within an acceptable amount of computer time. This undesirable situation may be attributed to an inefficient algorithm, an inherently difficult computational problem, or both. In the first case other algorithms might be considered, whereas in the second one is warned that the problem is intrinsically complex and a reformulation may be necessary.

In this paper we shall be concerned with the time-complexity of algorithms to solve certain problems: that is, the relationship between the size of an instance of the generic problem and the time required to solve

†Present address: British Telecom Headquarters, Cheapside House, Room 213, London EC2V 6JH, Great Britain.

it. By employing the theory of *NP*-completeness and *NP*-hardness, reviewed in Section 2 since it is probably not well known to statisticians, we may identify certain computational problems as members of a class thought to be intractable. Three computational problems in statistics, namely optimal clustering, optimal subset selection in regression and *D*-optimal exact design of experiments, are shown in Sections 3, 4 and 5 to be members of this class.

2. *NP*-COMPLETENESS AND *NP*-HARDNESS

We now give some definitions and a brief review of the essential concepts of the theory of *NP*-completeness and *NP*-hardness. For a comprehensive review the reader is referred to Garey and Johnson (1979), whereas for a more concise account see Tarjan (1978).

The theory is directed primarily towards yes/no decision problems usually possessing some parameters to which values are assigned to define instances of the generic problem. The problems of interest to the statistician, whilst often not of this form, usually have decision analogues. For example, consider the following problem, to be examined further in Section 4.

Optimal subset selection in regression

The parameters are three positive integers n, k and $q(q \leqq k)$, an $n \times 1$ data vector $\mathbf{Y}$ and an $n \times k$ design matrix $\mathbf{X}$ and the objective is to find a $k \times 1$ vector $\hat{\boldsymbol{\beta}}$ with only q nonzero entries such that the residual sum of squares $R(\hat{\boldsymbol{\beta}}) = (\mathbf{Y} - \mathbf{X}\hat{\boldsymbol{\beta}})^T(\mathbf{Y} - \mathbf{X}\hat{\boldsymbol{\beta}})$ is minimized. An instance is defined by giving values to n, k, q, $\mathbf{Y}$ and $\mathbf{X}$.

The yes/no decision analogue may be formed in this case by adding a further parameter B to those above and, instead of minimizing $R(\hat{\boldsymbol{\beta}})$, asking whether there exists a $k \times 1$ vector $\hat{\boldsymbol{\beta}}$ with only q nonzero entries such that $R(\hat{\boldsymbol{\beta}}) \leqq \mathbf{B}$.

The generic problem of interest will typically have many possible instances and we are concerned with the relationship between the "size" of an instance and the time required for execution. We only use an informal notion of the size $s(I)$ of an instance I: the input length presented to the computer by a reasonable encoding of I. Similarly, the computer model need not concern us. The time-complexity function $T(n)$ of an algorithm solving a given generic problem is defined as

$$T(n) = \max_{I:s(I)=n} \{\text{time required to solve } I\}$$

and we note that this definition is based upon the worst-time case.

Edmonds (1965) emphasized the important distinction between polynomial time algorithms, with $T(n)$ bounded above by a polynomial, and exponential algorithms, where such a bound does not exist. Problems solvable by a polynomial time algorithm constitute the class P, whereas those only solvable in exponential time have been termed intractable as exponential algorithms quickly require such vast amounts of computer time to execute as n increases that, frequently, even moderate-sized instances are infeasible. The theory is concerned with identifying intractability.

In order to provide a general, mathematically convenient algorithmic framework, sufficiently powerful to include exponential time, Cook (1971) introduced the idea of a nondeterministic polynomial time algorithm: a nondeterministic "guessing" stage repeatedly guesses structures and a polynomial time checking stage verifies whether the proposed structure proves the answer to be "yes" or not. The number of guesses need not be polynomially bounded and many complex problems may be solved by such an algorithm using an exponential number of guesses. These algorithms are not intended as a practical means of solution, but are a conceptual device for defining the class *NP* of problems that can be solved by nondeterministic polynomial time algorithms. For our subset selection in regression decision analogue, the guessing stage might choose a subset of q elements of $\hat{\boldsymbol{\beta}}$ allowed to take nonzero values and the checking stage would elucidate whether the least squares fit achieves the bound B.

Clearly, P is contained in *NP*, since a problem solvable in polynomial time could have a nondeterministic polynomial algorithm with a null guessing stage and a polynomial verification stage providing the solution. However, though intuitively reasonable, it remains to be proved that the complement of P in *NP*, *NP-P*, is not empty: not one of the very extensive list of problems in *NP* has been shown to be outside P and such a proof seems remote. Hence, Cook's weaker theory attempts to prove that a problem is "as hard" as any other in *NP* and, therefore, if *NP-P* is not empty (as is widely believed) it is as hard as those belonging to *NP-P* and also has no polynomial time algorithm. More precisely, a problem Π is *NP*-complete if (i) $\Pi \in NP$ and (ii) any other problem $\Pi' \in NP$ is transformable to Π in polynomial time; that is, given any instance of Π' there exists a polynomial time algorithm (for instances of Π') that will transform it into an instance of Π such that the answer to Π' is "yes" if and only if the answer to Π is "yes". If Π' is polynomially transformable to Π and no polynomial time algorithm exists for Π' then Π must also be intractable since, otherwise, the transformation would provide a method of solving Π' in polynomial time. It is in this sense that Π is at least as hard as Π', the *NP*-complete problems are the hardest in *NP*, and all *NP*-

complete problems are intractable if any one is (*NP-P* is not empty). As polynomial transformability is transitive, given one problem Π' known to be *NP*-complete a simpler method of showing Π to be also *NP*-complete is to prove that $\Pi \in NP$ and Π' is polynomially transformable to Π. Cook supplied the first *NP*-complete problem, Satisfiability, and Karp (1972) added some practical combinatorial problems by transformation.

We may generalize the idea of *NP*-completeness to certain problems outside *NP*. If a problem Π, not necessarily in *NP*, is such that any instance of an *NP*-complete problem Π' can be polynomially transformed into an instance of Π and the solution of Π (now not necessarily of the yes/no type) determines the answer to Π', then Π is at least as hard as the *NP*-complete problems, Π cannot be solved in polynomial time unless *NP-P* is empty, and Π is termed *NP*-hard. Rather than show that the decision analogues of the examples from statistical computation are *NP*-complete, we now prove that the original optimality problems are *NP*-hard.

3. CLUSTER ANALYSIS

The objective of cluster analysis is to partition a given set of n entities into a specified number, K, of initially undefined, mutually exclusive and exhaustive groups or clusters according to some optimality criterion. We only consider the situation where the available information comprises "distances", not necessarily Euclidean, and the criterion reflects the desire to achieve homogeneous, well separated clusters in this metric. Possible criteria include

i) minimize the maximum within cluster distance,
ii) minimize the sum of average within cluster squared distances, and
iii) minimize the total within cluster distance.

For reviews of cluster analysis see Cormack (1971) and Everitt (1974).

As the number of possible partitions is approximately K^n, the computational requirements of a naive examination of all partitions are usually prohibitive and most algorithms find good but not necessarily optimal solutions [see, for example, Friedman and Rubin (1967)]. Hansen and Delattre (1978) have shown that the decision problem analogue of the optimality problem with criterion (i) is *NP*-complete and, further, give a branch and bound algorithm which greatly reduces the number of partitions requiring examination to guarantee an exact solution. We extend the Hansen and Delattre result to show that optimal clustering is *NP*-hard for a much wider class of criteria, including (i), (ii) and (iii) above. Let a partition P_K into K clusters be called perfect if all within cluster distances are zero (note, however, that the clusters need not be well separated as no conditions are imposed on the between cluster distances). Consider the class of criteria minimize $d(P_K)$, where $d(P_K)$ is a function of the within cluster distances such that $d(P_K) \geqq 0$ with $d(P_K) = 0$ if and only if P_K is perfect.

THEOREM 1 *Optimal clustering is NP-hard for any criterion minimize* $d(P_K)$ *such that* $d(P_K) \geqq 0$ *and* $d(P_K) = 0$ *if and only if* P_K *is a perfect partition.*

Proof Following Hansen and Delattre we polynomially transform one of Karp's *NP*-complete problems: Graph K-Colourability (*GKC*), also known as chromatic number. An instance of *GKC* is provided by a graph $G = (V, E)$ and a positive integer K and the question posed asks whether G is K-colourable; that is, does there exist a function f mapping every vertex $v \in V$ onto the integers $\{1, \ldots, K\}$ such that $f(u) \neq f(v)$ whenever the edge $\{u, v\} \in E$.

Take an instance of *GKC*. The n vertices of G will correspond to entities and the presence or absence of edges between vertices to distances. We define the distance $d(u, v)$ between entities u and v as $d(u, v) = 1$ if $\{u, v\} \in E$ and 0 otherwise. As there are only $\frac{1}{2}n(n-1)$ distances to compute the transformation can be achieved in polynomial time. Now consider the solution of the optimal clustering problem with K clusters: if minimum $d(P_K) = 0$ the perfect partition found demonstrates that G is K-colourable whereas if minimum $d(P_K) > 0$ then no perfect partition exists and G is not K-colourable. Hence, optimal clustering for the defined class of criteria is *NP*-hard and no polynomial time algorithm exists unless *NP-P* is empty.

4. SUBSET SELECTION IN REGRESSION ANALYSIS

Optimal subset selection in regression analysis is a well-known problem in computational statistics reviewed by Hocking (1976). For a definition of the computational problem see Section 2. In practice, we might be interested in trying various values of q, the parameter defining the subset size, but this leaves the problem essentially unchanged for our purposes.

As the possible subsets number $\binom{k}{q}$ and increase exponentially with k, efficient algorithms have attempted to find an optimal subset by employing branch and bound to explore only a fraction of the total. Algorithms suitable for most practical examples are described by, for example, Beale, Kendall and Mann (1967), Lamotte and Hocking (1970) and Furnival and Wilson (1974). That these methods can cope with realistic examples is a reflection of the relatively small values of k typically

found (less than 50 say) since the optimality problem is in fact, *NP*-hard as we demonstrate below.

THEOREM 2 *Optimal subset selection in regression is NP-hard.*

Proof We polynomially transform Minimum Weight Solution to Linear Equations (MWSTLE), shown by Garey and Johnson to be *NP*-complete. An instance comprises an $n \times k$ matrix $\mathbf{X}$, an $n \times 1$ vector $\mathbf{Y}$ and a positive integer $q \leqq k$ and we ask whether there exists a $k \times 1$ vector $\hat{\boldsymbol{\beta}}$ with at most q non-zero entries such that $\mathbf{X}\hat{\boldsymbol{\beta}} = \mathbf{Y}$. The transformation is trivial: take the parameters n, k, q, $\mathbf{X}$, $\mathbf{Y}$ of any instance of MWSTLE and solve the same optimal subset selection in regression instance. If we achieve $R(\hat{\boldsymbol{\beta}}) = 0$, then $\mathbf{X}\hat{\boldsymbol{\beta}} = \mathbf{Y}$ and the answer to the instance of MWSTLE is "yes", whereas if $R(\hat{\boldsymbol{\beta}}) > 0$ then the answer is "no". Therefore, optimal subset selection in regression is *NP*-hard.

5. *D*-OPTIMAL EXACT DESIGN OF EXPERIMENTS

Here we are concerned with choosing n experimental settings or design points $x_{(1)}, \ldots, x_{(n)}$, not necessarily distinct and possibly vector-valued, from a design space $\mathcal{X}$ of r candidate sites $x_1, \ldots, x_r$. At the design points we shall observe yields $Y_1, \ldots, Y_n$ according to the linear model

$$Y_i = \mathbf{f}(x_{(i)})^T \boldsymbol{\beta} + \varepsilon_i, \quad i = 1, \ldots, n$$

or, in matrix notation,

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}.$$

The matrix $\mathbf{X}$ is an $n \times k$ design matrix ($k \leqq n$) with row i containing $\mathbf{f}(x_{(i)})^T$ and $\mathbf{f}$ is a $k \times 1$ vector of linearly independent functions on $\mathcal{X}$, $\boldsymbol{\beta}$ is a $k \times 1$ vector of unknown parameters, and $\boldsymbol{\varepsilon}$ is an $n \times 1$ vector of independently distributed errors, each with mean zero and constant variance. A *D*-optimal design is a choice of the n design points such that $\det(\mathbf{X}^T\mathbf{X})$ is maximized, whereupon the generalized variance of the least squares estimates of $\boldsymbol{\beta}$ is minimized (see St. John and Draper (1975) for a review of *D*-optimality). If resource constraints fix n at a small value, as we assume, such designs are called exact and the computationally simple, approximate design theory of Kiefer and Wolfowitz (1959) is inappropriate.

There are $\binom{r+n-1}{n}$ possible designs over which to maximize $\det(\mathbf{X}^T\mathbf{X})$ and most algorithms have only attempted to find good but not necessarily optimal solutions; the DETMAX algorithm of Mitchell (1974) is probably the most successful. More recently, Welch (1982) describes a branch and bound algorithm guaranteeing an optimal solution, but limited to moderate-sized problems (r, $n < 30$, say). We now show that the optimality problem is *NP*-hard.

THEOREM 3 *D-optimal exact design of experiments is NP-hard.*

Proof The target problem for polynomial transformation is another proved by Karp to be *NP*-complete, namely Hamiltonian Circuit (HC). An instance of HC is specified by a graph $G = (V, E)$ with k vertices and r edges and we ask whether G contains a Hamiltonian circuit, that is, an ordering $v_{\Pi(1)}, \ldots, v_{\Pi(k)}$ of the vertices $v_1, \ldots, v_k$ of G such that $\{v_{\Pi(k)}, v_{\Pi(1)}\} \in E$ and $\{v_{\Pi(i)}, v_{\Pi(i+1)}\} \in E$ $(1 \leqq i < k)$.

In the corresponding experiment the k vertices are associated with k unknown treatment parameters $\beta_1, \ldots, \beta_k$ and we may only make observations on the differences between certain pairs of parameters, those allowed being determined by the edges E. In fact, we define the r candidate sites as the set $\mathcal{X}$ where $\mathbf{x}_{ij} \in \mathcal{X}$ if and only if $\{v_i, v_j\} \in E$ and $\mathbf{x}_{ij}$ is a $k \times 1$ vector of zeros except for $+1$ in position i and -1 in position j (the ordering of i and j is unimportant) and we are limited to taking only k observations. The linear model

$$Y_i = \mathbf{x}_{(i)}^T \boldsymbol{\beta} + \varepsilon_i \quad i = 1, \ldots, k$$

would be over-parameterized since we can only make observations on the differences between parameters, but one reparameterization is

$$Y_i = \mathbf{f}(\mathbf{x}_{(i)})^T \boldsymbol{\Upsilon} + \varepsilon_i, \quad i = 1, \ldots, k$$

where $\mathbf{f}(\mathbf{x}_{(i)})$ is a $(k-1) \times 1$ vector of the first $k-1$ elements of $\mathbf{x}_{(i)}$ and $\boldsymbol{\Upsilon}$ is the $(k-1) \times 1$ vector $(\beta_1 - \beta_k, \ldots, \beta_{k-1} - \beta_k)$. The matrix $\mathbf{X}$ is therefore $k \times (k-1)$ with row i containing $\mathbf{f}(\mathbf{x}_{(i)})^T$. We now show that $\max \det(\mathbf{X}^T\mathbf{X}) = k$ if and only if G contains a Hamiltonian circuit.

If G does contain a Hamiltonian circuit $v_{\Pi(1)}, \ldots, v_{\Pi(k)}$ then the points $\mathbf{x}_{\Pi(k)\Pi(1)}$ and $\mathbf{x}_{\Pi(i)\Pi(i+1)}$ $(1 \leqq i < k)$ belong to $\mathcal{X}$ and they may be chosen as the k design points. By considering the structure imposed on the $\mathbf{X}$ matrix by such a choice we find that $\det(\mathbf{X}^T\mathbf{X}) = k$. Conversely, suppose $\det(\mathbf{X}^T\mathbf{X}) = k$ for a design matrix $\mathbf{X}$. Now

$$\det(\mathbf{X}^T\mathbf{X}) = \sum_{i=1}^{k} \det(\mathbf{X}_{-i}^T\mathbf{X}_{-i}) = \sum_{i=1}^{k} [\det(\mathbf{X}_{-i})]^2$$

where $\mathbf{X}_{-i}$ is the $(k-1) \times (k-1)$ matrix formed by deleting row i from $\mathbf{X}$. The application of row and column operations not changing $\det(\mathbf{X}_{-i})$

demonstrates that det $(\mathbf{X}_{-i}) = \pm 1$ for any nonsingular $\mathbf{X}_{-i}$ and, therefore, $\det(\mathbf{X}^T\mathbf{X}) = k$ implies that $\mathbf{X}_{-i}$ is nonsingular $(1 \leqq i \leqq k)$. Three necessary conditions for all $\mathbf{X}_{-i}$ to be nonsingular $(1 \leqq i \leqq k)$ are:

i) all rows of $\mathbf{X}_{-i}$ are distinct and hence all rows of $\mathbf{X}$ are distinct;
ii) no column of $\mathbf{X}_{-i}$ consists entirely of zeros and hence $\mathbf{X}$ must include at least two rows with ± 1 in column $j (1 \leqq j < k)$ (if there were only one such row it would be deleted for one $\mathbf{X}_{-i}$), and
iii) for at least one row of $\mathbf{X}_{-i}$ the row sum is not zero and, hence, at least two row sums of $\mathbf{X}$ must be nonzero.

Therefore, if $\det(\mathbf{X}^T\mathbf{X}) = k$, from (i) all design points must be distinct; by (ii) they include at least two comparisons of treatment j with another $(1 \leqq j < k)$ and by (iii) they must also include at least two comparisons of treatment k with another (only such design points have a nonzero row sum in X). The design points must, therefore, correspond to edges forming a Hamiltonian circuit of the graph G. Therefore, G contains a Hamiltonian circuit if and only if max $\det(X^TX) = k$ and D-optimal exact design of experiments is NP-hard.

6. CONCLUSIONS

We have proved that three optimality problems found in computational statistics are NP-hard; that is, unless NP-P is empty they cannot be solved in polynomial time. However, the reader may be perturbed by the use of transformations in all three proofs that led to a mapping of instances of the target problem into instances of the statistical problem that are pathological or at least not typical of those commonly encountered. There is, indeed, some truth in this argument since the worst-case definition of time-complexity employed here and elsewhere would allow an atypical, small subset of the optimality problem to dominate the time-complexity function. A redefinition of time-complexity in terms of an average case would lead to great difficulties, though, firstly in specifying a probability distribution over instances and, secondly, in proving intractibility. Unfortunately, this is the state of the art.

On the other hand, some practical benefits can ensue from such results: if an optimality problem is known to be NP-hard we are warned that inherently complex cases can arise within the general framework, so prompting a re-examination of the problem, perhaps limiting the objective of an algorithm to finding good but not necessarily optimal solutions adequate for most practical purposes. If an exact solution is required to an optimality problem known to be NP-hard then, as in the three examples cited here, efficient approaches usually involve branch and bound or some other partial enumeration technique.

Acknowledgements

This work was funded by Warren Spring Laboratory, Stevenage, England. The author would like to thank A. C. Atkinson for several valuable comments.

References

Beale, E. M. L., Kendall, M. G. and Mann, D. W. (1967). The discarding of variables in multivariate analysis. *Biometrika* **54**, 357–366.

Cook, S. A. (1971). The complexity of theorem-proving procedures. *Proc. Third Annual ACM Symp. on Theory of Computing*, Association for Computing Machinery, New York, 151–158.

Cormack, R. M. (1971). A review of classification. *J. Roy. Statist. Soc. Ser.A* **134**, 321–367.

Edmonds, J. (1965). Paths, trees, and flowers. *Canad. J. Math.* **17**, 449–467.

Everitt, B. S. (1974). *Cluster analysis*. Heinemann, London.

Friedman, H. P. and Rubin, J. (1967). On some invariant criteria for grouping data. *J. Amer. Statist. Assoc.* **62**, 1159–1178

Furnival, G. M. and Wilson, R. W. (1974). Regression by leaps and bounds. *Technometrics* **16**, 499–511.

Garey, M. R. and Johnson, D. S. (1979). *Computers and Intractability: A Guide to the Theory of NP-Completeness*. Freeman, San Francisco.

Hansen, P. and Delattre, M. (1978). Complete-link cluster analysis by graph coloring. *J. Amer. Statist Assoc.* **73**, 397–403.

Hocking, R. R. (1976). The analysis and selection of variables in linear regression. *Biometrics* **32**, 1–49.

Karp, R. M. (1972). Reducibility among combinatorial problems, in R. E. Miller and J. W. Thatcher (Eds.), *Complexity of Computer Computations*, Plenum Press, New York, 85–103.

Kiefer, J. and Wolfowitz, J. (1959). Optimum designs in regression problems. *Ann. Math. Statist.* **30**. 271–294.

LaMotte, L. R. and Hocking, R. R. (1970). Computational efficiency in the selection of regression variables. *Technometrics* **12**, 83–93.

Mitchell, T. J. (1974). An algorithm for the construction of D-optimal experimental designs. *Technometrics* **16**, 203–210.

St. John, R. C. and Draper, N. R. (1975). D-optimality for regression designs: a review. *Technometrics* **17**, 15–23.

Tarjan, R. E. (1978). Complexity of combinational algorithms. *SIAM Rev.* **20**, 457–491.

Welch, W. J. (1982). Branch-and-bound search for experimental designs based upon D optimality and other criteria. *Technometrics*.