A model search procedure for hierarchical models

George Ostrouchov and Edward L. Frome

Mathematical Sciences Section, Oak Ridge National Laboratory, Oak Ridge, TN 37831-6367, USA

Received February 1990
Revised November 1991

Abstract: Large data sets cross-classified according to multiple factors are available in epidemiology and other disciplines. Their analysis often calls for finding a small set of best hierarchical models to serve as a basis for further analysis. This selection can be based on some well defined model optimality criterion. Fitting all possible models to find a best set is usually not feasible for as few as five factors (7581 possible models). We note that the set of hierarchical models and their relationships can be represented by a graph and develop an algorithm to generate it efficiently. We further develop a graph traversal algorithm that requires fitting of only a fraction of all models to find exactly a best subset of the models. The algorithm classifies as many models as possible on the basis of each fit. A data structure implementing the graph of model nodes keeps track of the information required by the model search algorithm.

Keywords: Information criterion, Bayes criterion, Model optimality, Partial ordering, Cover graph, Spanning tree, Depth first search, Branch and bound.

1. Introduction

In epidemiology, cause-specific mortality data are often cross-classified according to the levels of several risk factors. Hierarchical log-linear models are used to evaluate the potential association between the risk factors and the mortality rates. In exploratory studies, an important objective is to find all of the models that adequately describe the data. One approach to this problem that can be used for small tables (up to three or four factors) is to fit all possible hierarchical models and then use an information type statistic, such as Akaike or Bayes (see [1] and [14]), to select those models that provide a good summary of the data. For higher dimensional tables this approach becomes difficult because of computational complexity. With 4, 5, and 6 factors, the number of possible models is 168, 7,581, and 7,828,354, respectively. Typically, models with high order interactions are difficult to interpret. However, even when we

Correspondence to: Dr. G. Ostrouchov, Mathematical Sciences Section, Oak Ridge National Laboratory, Oak Ridge, TN 37831-6367, USA (E-mail: ost@msr.epm.ornl.gov)

consider only models of order two or less, 5 and 6 factors produce 1,451 and 40,070 models, respectively. The computational complexity lies not only in generating and managing these models, but also in fitting them. As the number of factors increases, the number of possible models grows faster than exponentially and the effort in fitting a single model grows exponentially.

Many model selection procedures select a single model through a series of significance tests on individual or groups of parameters. An overview of several such procedures is given by Wrigley [15]. The selected model will often differ between these procedures even if a common test size is used. This is because, in general, individual parameters are not orthogonal so that marginal effects are not additive.

Optimization approaches with a well defined model optimality criterion are more consistent because the criterion ranks models the same way regardless of the method of search. We emphasize that a search procedure does not select models. It merely finds models already selected by an optimality criterion. As is the case for most optimization problems, some search methods guarantee to find the optimum while others use a heuristic to obtain something “close” to the optimum. These approaches are typically more computationally intensive than those relying on significance tests of parameters.

Model search procedures that take the optimization approach have been developed for regression models (for example, see [8], [12]) where all possible subsets of the regressor variables are considered. These algorithms are based on the fundamental inequality $RSS(A) \leq RSS(B)$, where RSS is the residual sum of squares and model B is contained in model A. In [8] and [12], models are first partitioned according to number of model terms and then best fitting models are found for each partition. Edwards and Havranek [4] have developed a fast procedure for model search in hierarchical models based on one optimality criterion. The procedure is approximate in the sense that it does not guarantee to find all models that belong in the best subset.

We use the same fundamental inequality as [8] and [12], but our optimality criterion is any of a class of functions of both model size and goodness of fit. That is, we select a single set of models that are best with respect to one of a broad class of optimality criteria. Our best set is exact in the sense that any model in the best set is better than any model not in the best set. Also, our models consist only of hierarchical combinations of variables. Under this restriction, the selected set of models is invariant to location and scale transformations of the variables [13].

Motivation for the development of our procedure came from epidemiology; therefore, some of its description and implementation is for log-linear models. The procedure, however, is applicable to most other situations that require finding a best set of hierarchical models from some larger set of hierarchical models.

The basis for our model search algorithm is a graph representation of the relationships between a set of hierarchical models. This representation and its implementation are developed in the next section. The model search algorithm is described in section 3 and its performance is discussed in section 4.

2. Graph representation of a set of hierarchical models

A hierarchical model can be represented by its generating class (for example, see [4]). The generating class is the set of maximal terms not set to zero. For example, the model

$$\mu + a + b + c + ab + ac$$

is represented by $[ab, ac]$, and the model

$$\mu + a + b + c + ac$$

is represented by $[ac, b]$. The representation simply omits every term that is contained in another term. We will call this the maximal representation. This representation is also useful in iterative proportional fitting of a log-linear model since the maximal terms specify the table margins that are involved in the iterations. The maximal terms also define the smallest tables required to represent the fitted values.

An alternative representation, also discussed in [4], is to use the minimal terms that are set to zero. This is called the dual representation. Assuming three factors, dual representations for the two models above are given by $[bc]^d$ and $[ab, bc]^d$, respectively.

Both representations are useful in generating other models from a given model. The maximal representation shows exactly the terms that should be set to zero to obtain hierarchical models that are smaller by one term. Analogously, the dual representation shows exactly the terms that should be added to obtain hierarchical models that are larger by one term. Consider a three factor example. The model $[ab, ac]$ has only one model that is larger by one term, namely $[ab, ac, bc]$, and has two models that are smaller by one term, namely $[ab, c]$ and $[ac, b]$. Thus we remove ab or ac to obtain the smaller models and we add bc to obtain the larger model. This is exactly what the maximal representation $[ab, ac]$ and the dual representation $[bc]^d$ indicate.

Given k factors there are 2^k possible model terms (all subsets of k), but not all model term combinations form hierarchical models. Let $\mathcal{H}_k$ denote the set of all possible hierarchical models for k factors including both the minimal model and the null model. Crude bounds on the number of models in this set are given by

$$2^k <\mid \mathcal{H}_k \mid< 2^{2^k}, \text{for } k > 0.$$

Since a model is a set of model terms, the subset relation $\subset$ imposes a *partial ordering* on the set of all possible hierarchical models. A partial ordering can be

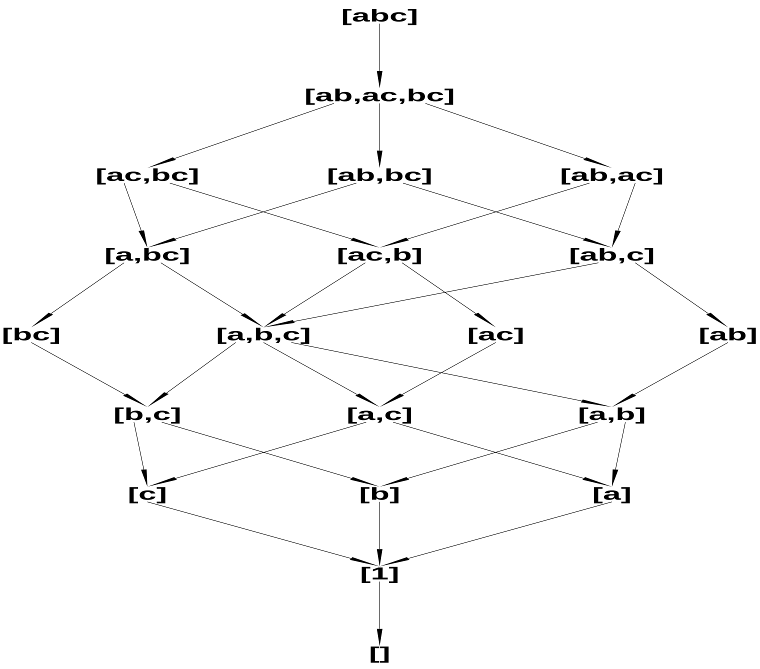

Figure 1: G_3. Cover graph induced by the subset relation on $\mathcal{H}_3$, the set of all hierarchical models for three factors.

represented by a directed graph. A graph consists of a set of *nodes* and a set of *edges*. Two nodes are considered neighbors if there exists an edge between them. In a directed graph, edges have direction. A precise definition of a graph and many algorithms applicable to graphs can be found in a number of books on design of computer algorithms. See for example, [9] or [10].

The smallest directed graph that represents a partial ordering (containing the least number of edges) is called a *cover graph* of the partial ordering. Let the cover graph induced by $\subset$ on $\mathcal{H}_k$ be denoted by G_k. This cover graph consists of one node for each model in $\mathcal{H}_k$ and an edge between any two models that differ by exactly one term. The term by which two neighboring models differ is associated with the edge that connects them and the edge is directed towards the smaller model. Examples of G_3 and G_4 are shown in Fig. 1 and Fig. 2, respectively. Models are shown in their maximal representation and each edge emanates from the term that is being deleted to form the smaller model. Note that the dual representation of a model is obtained by collecting the terms deleted on all incoming edges. For example, the dual representation of $[a,b,c]$ is $[ab,ac,bc]^d$. Also note that G_3 is a subgraph of G_4 and that in general G_i is a subgraph of G_j for $i < j$.

The graph representation G_k of a set of hierarchical models clearly divides the models into *levels*; edges exist only between models in adjacent levels. Any two

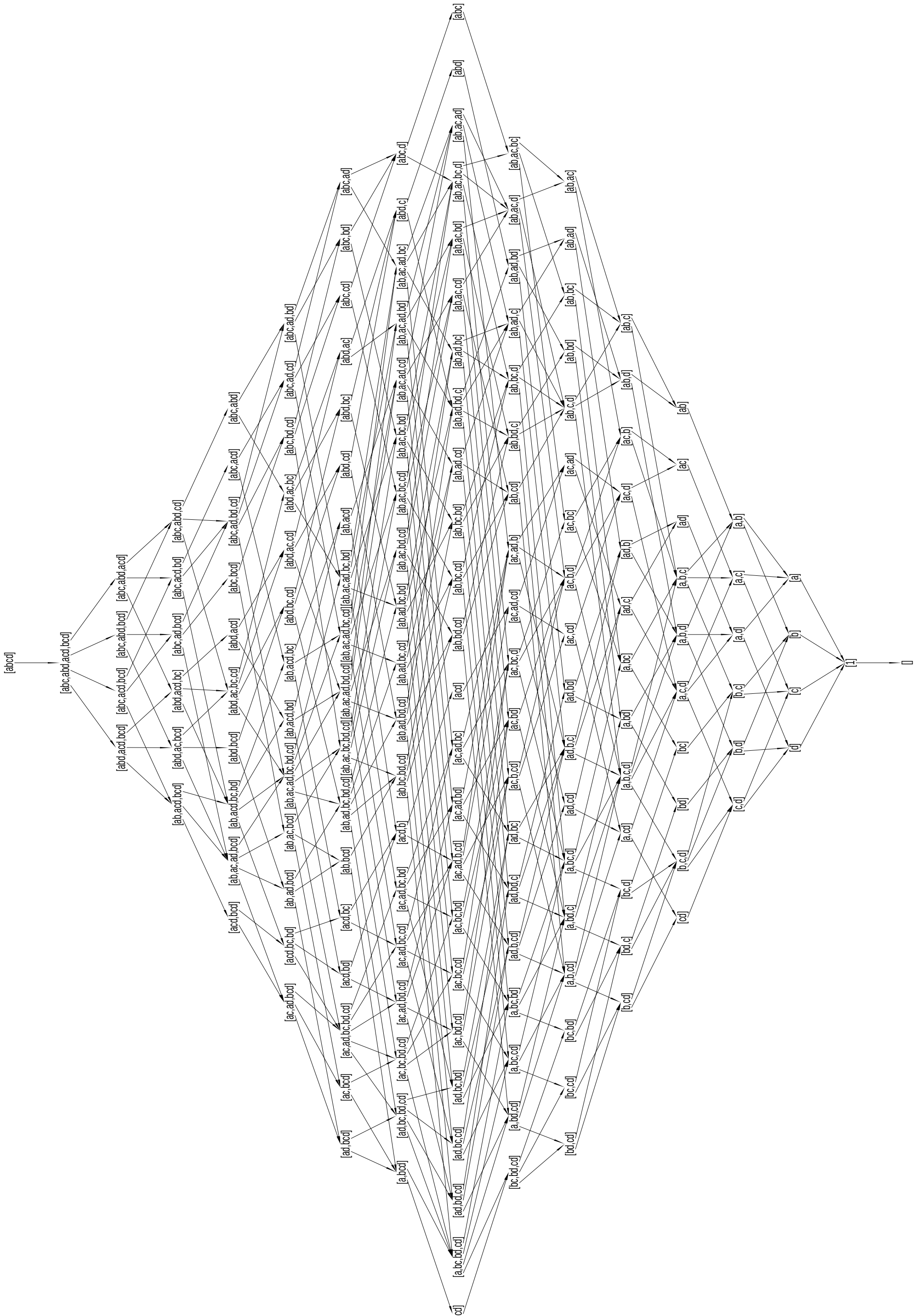

Figure 2: G_4, cover graph induced by the subset relation on the set of all hierarchical models for four factors.

adjacent levels thus form a *bipartite graph*. Each level represents a set of hierarchical models that have a given number of model terms. Since 2^k is the number of possible model terms from k factors and we include both the minimal model and the null model, G_k has $2^k + 1$ levels.

Because G_k is a cover graph of a partial ordering induced by the subset relation, it has some interesting properties. Keeping in mind that a model is a set of terms, the intersection model of two models is given by a model that is on the highest possible level and is reachable from the two models. Similarly, the union of two models is given by a model that is on the lowest possible level from which both models can be reached. Some examples in Fig. 1 are: $[ab] \cap [a, bc] = [a, b]$, $[ab] \cup [a, bc] = [ab, bc]$, $[ab, ac] \cap [ac, bc] = [ac, b]$, and $[ab, ac] \cup [ac, bc] = [ab, ac, bc]$.

Generation of G_k or a subgraph of G_k on a computer is accomplished in either a top-down or a bottom-up fashion. We start with a single model and generate either all its submodels or all models that are larger and have a given number of factors. Each level is constructed by either deleting or adding single terms in each model on the previous level, as guided by the maximal or dual representations, respectively. The union and intersection properties are used to establish all edges of a new model node to previously generated model nodes. This allows us to proceed in any order within a level and ensures that no duplicate models are generated.

We give the top-down algorithm and note that the bottom-up algorithm is obtained simply by interchanging union with intersection and maximal with dual. The algorithm in Fig. 3 describes the construction of one level given the previous level L. This construction involves the creation of nodes of the next level as well as the creation of all edges pointing to those nodes. In this algorithm, A is a node of the graph, and $[A]$ and $[A]^d$ are the maximal and dual representations of the model at node A. A node A has an edge associated with each term in $[A]$ pointing towards the smaller model created by deleting that term. The edge associated with term t is denoted by $\xrightarrow{t}$ and the underlying data structure is assumed to allow traversal of edges in both directions.

No duplicate models are created by this algorithm, because each new model is immediately connected (in the innermost **For** loop) to all larger models that can potentially generate it. Note that $S \xleftarrow{u} S \cup C \xrightarrow{t} C$ uses established edges and requires the level above level L, which contains the node with model $[S] \cup [C]$. When we start with a single model on the top level, such connections are needed only after completing the next level.

Also note that the order for selection of nodes and terms in all **For** statements is not specified, because any order will work. However, we have found that if dictionary order is used for terms in constructing smaller models (second **For** statement), then the connections to existing models (third **For** statement) order the edges in re-

```
For each node $S$ on level L {
    For each term $t$ in $[S]$ that does not have an edge {
        Create $M$ with no edges and let $[M] = [S] - t$.
        For each term $u$ in $[M]^d$ {
            Let $[C] = [M] + u$.
            Find $C$ using established edges $S \xleftarrow{u} S \cup C \xrightarrow{t} C$.
            Create edge associated with $u$ of $C$.
            Point the edge to $M$ (i.e. establish $M \xleftarrow{u} C$ connection).
        }
    }
}
```

Figure 3: Algorithm for constructing the next lower level from level L.

verse dictionary order. That is, dictionary order for terms in maximal representation induces reverse dictionary order for terms in dual representation. For example, dictionary order for terms from three factors is a, ab, abc, ac, b, bc, c. Dictionary order also brings out symmetries in the graph.

To generate G_k we start with the saturated k-factor model and generate all its submodels. Starting with a smaller model generates a subgraph of G_k. For example, starting with the model $[a, b, c]$ in Fig. 1 generates the subgraph of G_3 that contains only main effect models. A more interesting example is a subgraph of G_5 containing only models with up to order two interactions. This is generated from the all two-way interactions model $[ab, ac, ad, ae, bc, bd, be, cd, ce, de]$ and contains 1451 models (nodes) and 6776 edges (too many to display in a figure). It illustrates the quick growth in complexity as the number of factors increases to five.

The data structure for the graph representation is of a standard variety typically used for graphs. Two adjacency lists are maintained for each node, giving larger and smaller neighbors that correspond to terms of the dual and maximal representations, respectively. This allows edge traversal in both directions. Models are represented as sets of model terms with a single bit indicating the presence or absence of a model term. Similarly, a model term is a set of factors and the presence or absence of a factor is also indicated by a single bit. For example, the model term abd is represented by binary 01011 or by decimal 11. Its presence in a model is indicated by bit eleven of a model representation. Thus for k factors we require 2^k bits to represent a model.

3. Model search

To obtain an exact procedure for finding a best subset of models that avoids fitting all models, we require an optimality criterion that allows a bounding mechanism. Atkinson [1] discusses two groups of criteria. He shows that criteria based on information theory and Bayesian criteria all have the same general form. Model A is best if

$$-L_A + \frac{\alpha}{2} p_A$$

is minimum. Here L_A is the log likelihood for model A maximized over its p_A parameters and the coefficient α may be a constant or a function of the number of observations. We can rewrite this in terms of deviance, $D_A = -2(L_A - L_0)$, where L_0 is the log likelihood of the saturated model maximized over its parameters. It is equivalent to say that model A is best if

$$D_A + \alpha p_A$$

is minimum.

To obtain a bounding mechanism, we use the fundamental inequality that $D_A \leq D_B$ whenever model B is contained in model A. Suppose we fit model A and obtain its deviance, D_A. Then

$$D_B + \alpha p_B \geq D_A + \alpha p_B. \tag{1}$$

Conversely, if we fit model B, then

$$D_A + \alpha p_A \leq D_B + \alpha p_A. \tag{2}$$

Thus, by fitting one model we obtain lower bounds on the optimality criterion of smaller models and upper bounds on larger models. This enables us to accept some larger models or reject some smaller models without fitting.

Let c denote our optimality criterion. Because we only use the monotonicity property of a criterion in constructing the bounds (1) and (2), our procedure can admit a more general form of an optimality criterion

$$c = c(D, p) \tag{3}$$

that is monotone non-decreasing in D for a fixed p. The monotonicity property gives us inequalities analogous to (1) and (2). For model B contained in model A, fitting A gives

$$c(D_B, p_B) \geq c(D_A, p_B) \tag{4}$$

and fitting B gives

$$c(D_A, p_A) \leq c(D_B, p_A). \tag{5}$$

We further require that D should be by far the most difficult part of $c(\cdot,\cdot)$ to compute in order to realize any savings in bounding models instead of fitting models. Of course, we also require that the criterion be reasonable so that the resultant ordering of models is useful.

The criteria described by Atkinson [1] are given by

$$c(D, p) = D + \alpha p.$$

As an example of a different criterion that fits the form (3), consider the procedure of Edwards and Havranek [4] that finds most (and possibly all) *minimal acceptable* models. A model is *acceptable* if it is not rejected by a goodness-of-fit test at some nominal level of significance, and it is *minimal acceptable* if it does not contain any other acceptable model. To select similar models, we can define an optimality criterion that orders all *acceptable* models by their number of parameters followed by all models that are not acceptable. We assume that the goodness-of-fit test is a monotone function of D. This ordering is obtained with

$$c(D, p) = p + (p_0 - p)\mathrm{I}_{[rejected]}$$

where $I_{[rejected]}$ is 1 if the model is rejected by a goodness-of-fit test and is 0 otherwise, and p_0 is the number of parameters in the saturated model.

Given a criterion c and a criterion threshold c_0, our model search procedure fits a sequence of models and on the basis of each fit it classifies other models as *in the best set* ($c \leq c_0$), *not in the best set* ($c > c_0$), and *undetermined* (don't know if $c \leq c_0$ or $c > c_0$) by using inequalities (5) and (4). Each classification employs a depth-first search in a spanning tree of the graph (for example, see [10]). Such a search is efficient because a branch and bound approach can be employed in a tree. Whenever a node is classified as *undetermined*, the rest of that branch in the spanning tree is bounded. G_k for $k > 3$ has considerably less depth (number of levels) than breadth, so that depth-first search is more storage efficient than breadth-first search. We proceed to fit and classify models until no *undetermined* models remain. The choice of models to fit is crucial in determining the total number of fits required to complete this process.

The simplest automatic procedure for choosing models to fit starts at the largest model and tries to eliminate as many models as possible on the basis of model parameters alone (since deviance is bounded by zero below). Next, the procedure goes down through each level fitting any models that are still *undetermined.* For each model that is fit it attempts to eliminate more models using the lower bound (5). With this simple strategy only some models that are not in the best set will require fitting, however all models that are in the best set will require fitting. This is not unreasonable since the subset of interest is typically small.

A simple way to set the c_0 threshold for a given optimality criterion is to take the smallest criterion value of the minimal, the main effects, and all two-way interaction models. This results in relatively small subsets for data that do not have strong high order interactions. If a given threshold produces a best set that is either too large or too small, c_0 can be changed. In this case all models are reset to *undetermined* status and models that are already fit are reused in the classification process. Further fitting may be required to classify remaining *undetermined* models. If the above automatic procedure is used to choose models to fit, it is better to start with c_0 that is too low rather than too high, because all models in the best set are fit.

4. Discussion

Our implementation of the search procedure, including the graph representation data structure, is written in the C programming language with an interface to X windows. With this implementation we are able to conveniently control the model search process both by choosing individual models to fit and using the automatic top-down procedure. This tool is available by electronic mail from Statlib by sending *send hmodel from general* as the body of the message to *statlib@temper.stat.cmu.edu*. Because of our interest in categorical data, our implementation is currently limited to iterative proportional fitting of log-linear models.

Since we describe a model search procedure that finds models selected by an optimality criterion, we report on performance of the search process rather than describing the models selected. We do not advocate the use of one optimality criterion over another and merely state that a wide range of criteria can be used with our procedure. The reader is refered to several good articles discussing the merits of various optimality criteria ([1], [14], [5], and [11]).

We examine two data sets with our model search procedure. The first is a data set on mortality among a cohort of World War II nuclear industry workers that is reported in Frome et al. [7]. The second is a data set from Dyke and Patterson [3] on knowledge of cancer that is also reported in [6]. Performance is reported for the automatic top-down procedure as the number of model fits required to find all models better than a criterion threshold.

It would be interesting to know the minimum number of fits required for a given data set and a criterion threshold. This would first require fitting all models and for each model finding which other models it classifies as *in the best set* or *not in the best set*. Since each fit has a set of models that it classifies and we wish to classify all models, the minimum number of fits problem reduces to the *minimum set cover* problem. Unfortunately the *minimum set cover* problem is NP-hard (see [10]). Thus, it is not feasible to solve this problem for a large number of models. Instead we perform a few repetitions of the model search with different choices of models fit. We base these choices on the results of previous repetitions in an attempt to reduce the number of models fit and report the repetition with least fits. The intent is simply to show that the automatic top-down procedure can be improved rather than to make any statement about the minimum fits required.

Two tables are constructed from the mortality data [7]. One with four factors at 2, 3, 2, and 4 levels giving 48 cells. And the other with five factors at 2, 3, 2, 3, and 4 levels giving 144 cells. Frome et al. [7] used a model screening procedure developed by Brown [2] to evaluate the importance of each of the factors and to investigate the possibility of interactions. We have found that $D + 2p$ (the Akaike Information Criterion) selects models that are implied by interactions reported in [7]. Therefore, we report the search performance for $D + 2p$.

The simple top-down automatic procedure with a threshold given by the main effects model selects four models that are better out of the possible 168 by fitting only 37 models. We also found that it is possible to choose the models to fit so that only 29 are required to find the best five.

$\mathcal{H}_5$, the set of all possible models for the larger table, contains 7,581 models. The main effects model criterion value is again used as the threshold. It turns out that there is only one model that is better. The top-down automatic procedure finds the two best models after fitting 533 models. Our best choice of models to fit (found in a few repetitions of the search) needed 349 fits.

If we start with only models of order two or less (no 3-way and higher interactions), there are 1,451 possible models and they contain the two best models found above. The top-down automatic procedure required 160 fits. By more deliberate choice of models to fit we found that this can be reduced to 105 fits.

The Dyke and Patterson [3] data set has five factors each at two levels. We have found that the criterion $D+2p$ ranked models with several interactions of order three as better than the models selected by Fienberg [6], but the criterion $D+\log(n)p$ (the Bayes Information Criterion) put these models among the five best. We report the search performance for the latter criterion. Our threshold is the criterion value for the all two-way interaction model.

This data set presented a special problem for the automatic top-down procedure. Because all of the five factors have exactly two levels, each model has as many parameters as it has terms. This means that all models on level i of G_5 have exactly i parameters. The automatic procedure first eliminates models on the basis of their parameters, so it eliminates a number of the top levels leaving no models eliminated on the levels below. Thus all models on the next level will be fit. For this reason, the automatic procedure required 769 fits to find the 81 models (out of 7,581 possible) that are better than the all two-way interaction model. By more careful selection of models to fit, we were able to reduce this to 383 fits. If we reduce the criterion threshold so that only five models are selected, they can be found after only 164 fits. This is, of course, possible only after we know the criterion value for the best models, but it illustrates that fewer fits are required when the best subset is small.

We conclude that even the automatic top-down strategy requires the fitting of only a fraction of all models to determine exactly a small set of the best models. On our test data sets, this fraction decreases as the number of models under consideration increases. On the other hand, the automatic strategy is not optimal and can be considerably improved by more careful selection of models to fit.

We are investigating other strategies for choosing models to be fit. Better strategies seem to involve fitting some key models which are related to unions of models so far not classified. We are also investigating the use of parallel processing to extend the model search process to larger data sets and more complex models.

Acknowledgments

This work was supported by the Office of Health and Environmental Research, U.S. Department of Energy, under contract DE-AC05-84OR21400 with Martin Marrietta Energy Systems, Inc. and under contract DE-AC05-76OR00033 with the Center for Epidemiologic Research, Oak Ridge Associated Universities, and also by the Applied Mathematical Sciences subprogram of the Office of Energy Research, U. S. Department of Energy under contract DE-AC05-84OR21400 with Martin Marietta Energy Systems, Inc. The authors are grateful to an associate editor and two anonymous referees for helpful comments on an earlier draft of the manuscript.

References

[1] A. C. Atkinson. Likelihood ratios, posterior odds and information criteria. *Journal of Econometrics*, 16:15–20, 1981.

[2] M. B. Brown. Screening effects in multidimensional contingency tables. *Applied Statistics*, 25:37–46, 1976.

[3] G. V. Dyke and H. D. Patterson. Analysis of factorial arrangements when the data are proportions. *Biometrics*, 8:1–12, 1952.

[4] David Edwards and Tomáš Havránek. A fast procedure for model search in multidimensional contingency tables. *Biometrika*, 72:339–351, 1985.

[5] Bradley Efron. How biased is the apparent error rate of a prediction rule. *J. Amer. Statist. Assoc.*, 81:461–470, 1986.

[6] Stephen E. Fienberg. *The analysis of cross-classified categorical data*. The MIT Press, Cambridge, MA, 1977.

[7] Edward L. Frome, Donna L. Cragle, and Richard W. McLain. Poisson regression analysis of the mortality among a cohort of World War II nuclear industry workers. *Radiation Research*, 123:138–152, 1990.

[8] George M. Furnival and Robert W. Wilson, Jr. Regression by leaps and bounds. *Technometrics*, 16:499–511, 1974.

[9] Alan Gibbons. *Algorithmic Graph Theory*. Cambridge University Press, Cambridge, 1985.

[10] Ellis Horowitz and Sartaj Sahni. *Fundamentals of computer algorithms*. Computer Science Press, Inc., Rockville, MD, 1978.

[11] Anne B. Koehler and Emily S. Murphree. A comparison of the Akaike and Schwarz criteria for selecting model order. *Appl. Statist.*, 37:187–195, 1988.

[12] J. F. Lawless and K. Singhal. Efficient screening of nonnormal models. *Biometrics*, 34:318–327, 1978.

[13] Julio L. Peixoto. A property of well-formulated polynomial regression models. *Amer. Statist.*, 44:26–30, 1990.

[14] A. E. Raftery. A note on Bayes factors for log-linear contingency table models with vague prior information. *J. R. Statist. Soc. B*, 48:249–250, 1986.

[15] Neil Wrigley. *Categorical data analysis for geographers and environmental scientists*. Longman, London, 1985.