HIGH-PERFORMANCE COMPUTING AND PSYCHOLOGICAL MODELS

Chaired by Stephen W. Link
Federation of Behavioral, Psychological and Cognitive Sciences

The art of model development and testing

RICHARD M. SHIFFRIN and PETER A. NOBEL
Indiana University, Bloomington, Indiana

Mathematical and computer simulation modeling are often computationally demanding procedures, so much so that even certain parts of these procedures, such as parameter estimation, exceed the capacities and speed of the best modern computer facilities. A good deal of effort has therefore been dedicated to speeding up and making more efficient programs such as those that are meant to find a global minimum of a parameter space. Our experience, however, is that such well-explored technical procedures in fact represent some of the shortest components of the total set of procedures by which models are developed. In this article we discuss what elements take up the lion's share of development time and speculate on what lessons can be drawn concerning the role of high-performance computing in such enterprises.

Many of us who propose and test multiparameter simulation models have developed a good deal of expertise about the process. Some of this skill has to do with highly technical matters that are well represented in the literature (such as parameter estimation programs). We have found that these sorts of well-researched technical components account for a tiny fraction of the time and effort involved in model development. The great proportion of effort (both on the part of the theorist and probably also in terms of computation time) involves skills that are highly abstract and subjective, and these are seldom discussed in the literature. Thus, we have titled this article the *art* of model development. The discussion will highlight some associated lessons for the role of computation.

The standard story concerning the course of modeling does not really include what we spend most of our time doing, but goes something like the following: The theorist has one or more models, each incorporating some number of free parameters, that are to be assessed by fitting predictions to a set of data. Some metric of goodness-of-fit is selected, and a set of parameter values is found that represents the best fit for each model. Both the selection of a goodness-of-fit metric and the method of

locating the best set of parameter values are highly technical and difficult issues: The data may cross incommensurate domains (like accuracy and response time [RT]), the parameter space may have local minima, and the predictions themselves are often generated by Monte Carlo methods and therefore are inherently noisy. In any event, after some struggle, "best" parameter values are found for each model. If the theorist is lucky, the various models are nested in terms of the parameters (one model is a restriction of, and nested within, another), and this makes it easier to decide whether one is significantly better than another.

There are certainly cases in which such a scenario represents actual practice, as, for example, in situations in which a well-established model is applied to a set of data from a paradigm almost identical to one to which the model has been applied on an earlier occasion. However, such cases are rare, and the actual situation is usually considerably more complex. For one thing, in most cases, the models of interest are not nested and are difficult to compare: Usually none of the models fit the data very well in any statistical sense, so it is difficult to make a case for using statistical evaluation as the basis for preferring one model over another. Instead, preference judgments are based partly on the goodness-of-fit and partly on other, less quantifiable, characteristics of the fit to the data, combined with subjective assessments of the simplicity, elegance, generalizability, and sensibility of the model assumptions. (In addition, and unfortunately, in many cases much of the information that could be used,

This research was supported by NIMH Grant 12717 to R.M.S. P.A.N. is now at Microsoft Corporation, 1 Microsoft Way, Redmond, WA 98052. Correspondence should be addressed to R. M. Shiffrin, Psychology Department, Indiana University, Bloomington, IN 47405 (e-mail: shiffrin@indiana.edu).

Figure 1. A representative timeline for the development of a new multiparameter behavioral model for a complex set of data when predictions are produced by computer simulation.

and ought to be used, to evaluate models is not collected by the theorist, and whether or not collected, is seldom reported in publications; these points will be elaborated later in the article.)

To return to the main point, however, the most common situation is one in which an extant model must be altered, perhaps substantially, or a model must be invented. Many elements of the “standard approach” are of course needed in the course of model development in such a situation. However, in our experience, the elements of the standard approach represent perhaps 1% of the actual development time. Figure 1 depicts an illustrative time line for model development in which parameter estimation and related analyses for the final version of the model occur in the last few days of the project, the writing and running of simulation programs for other variants of the model occupy several weeks scattered through the period of development, and the remaining time, perhaps 99%, is occupied by other “stuff.” In this article, we discuss this other “stuff” and the resultant challenges that are posed for computation.

HOW ARE MODELS INVENTED?

Models do not spring up full blown, but arise through a long series of successive approximations. Let us assume that we are discussing complex models to be applied to large sets of data and that simulation methods are required to produce predictions. The way our field is progressing, this situation is becoming the norm. The process of model development in the initial stages—the stages that take the greatest amount of time—involves a good deal of conceptual analysis. This conceptual analysis involves such themes as the development of a consistent framework that generalizes the current task and that is consistent with other data and theoretical mechanisms; it also involves a

great deal of tuning of the theorist’s intuitions—that is, the theorist’s understanding of the model components and mechanisms, and their interactions. At this stage, degree of fit may not play a paramount role, with one important exception: The process of gaining understanding cannot be carried out by pure cogitation, because the models in question are too complex and have components that interact in initially unknown ways. Thus the early stages of tuning one’s intuitions must also utilize simulation techniques. However, because simulation modeling takes so much time on even the fastest computers, it is almost never possible or advisable to research a particular model variant and search its parameter space thoroughly before turning to other variants. This fact imposes a great deal of noise and a good deal of art into the process of model development.

Thus development proceeds in fits and starts: A good deal of thinking leads to an initial model containing what the theorist hopes contains a set of processes necessary and sufficient for the current data set and also consistent with the larger conceptual framework (other models, paradigms, and findings). Preliminary exploration of this model proceeds by simulation. A rather incomplete exploration of the parameter space leads to an apparent failure of prediction. The theorist stops and thinks again, trying to understand the failure (assuming the failure is real and not due to an incomplete parameter search). At this phase, qualitative concerns are paramount, and the logic of the model is at issue. The fact that the failure of the model may be apparent rather than real, because of incomplete simulation searches, is not entirely critical at this stage, since even a noisy simulation result, or one from the “wrong” part of the parameter space, often helps the theorist gain a better intuitive grasp of the interactions of the model components. Having gained some understanding, the theorist may possibly decide to search the

parameter space more thoroughly, or, more usually, decide to combine the new insights with the larger conceptual issues and produce a new model variant. This process then repeats, and the cycle continues for months or even years.

This interplay between simulation results and intuitions is the core of the process of model invention, and it occupies the lion's share of the development time. It consists in delicately balancing the time needed for accurate simulation results, the need to explore different types of models consisting of different interacting processes, and the danger of being misled by noisy or incomplete results. Throughout this process, a great deal of interactivity between theorist and simulation is needed, and this need for interactivity raises important questions about the way in which high-performance computers can be utilized. For example, we have often found it useful to pick a single point in the parameter space, generate predictions (perhaps somewhat noisy), think about the results, choose another single point, and so on, thereby not learning anything about a best fit, but learning something about the logic of the model.

With luck, at the end of this long process, a particular model is deemed worthy of systematic evaluation. At this stage of model development, time constraints typically impose additional limitations on the search of the parameter space. Thus, using ordinary workstations, we have often found it necessary at this final stage to combine human intuition with an assortment of parameter estimation algorithms to make effective progress. In addition, we have found it useful to turn to high-performance computing at this stage.

GOODNESS OF FIT

Goodness-of-fit questions arise at every stage of model development, but the nature of the assessment changes markedly during the process. At first, progress is guided by general conceptual concerns, conceptual concerns that to a certain extent transcend quantitative or qualitative accuracy of prediction. For example, suppose a set of data contains six identifiable and reliable trends. One model is developed that assumes six different independent processes, each based on one free parameter, each designed to produce one of the observed trends. This model is fit to the data and does well. A second model is borrowed from another setting where it has been used successfully; it has six parameters, but there is no obvious connection between its processes and parameters and the observed data. This model is fit to the data, and also fits well, but not quite as well as Model 1. Many theorists would regard Model 1 as uninteresting, would judge its predictions to be ad hoc and post hoc, and would greatly prefer Model 2. Metatheoretical issues of this sort tend to guide theory development at early stages.

When the model's predictions are taken into account, qualitative predictions often play a larger role than quantitative predictions. Consider the situation in Figure 2:

Figure 2. Hypothetical predictions from two models (solid lines) for a hypothetical set of data (unfilled circles). Left panels: data not previously examined in the literature. Right panels: accuracy data replicating a standard finding. Panel A: Model 1 fits the data in the left panel well (least squared error [LSE] = 52), but misses the direction of trend in the right panel (LSE = 23). Panel B: Model 2 fits the data in the left panel poorly (LSE = 112), but predicts the trend in the right panel (LSE = 3). Model 2 is quantitatively inferior (LSE = 115, which is higher than the value for Model 1, LSE = 75), but qualitatively superior.

The theorist considers the data trend in the right panel to be reliable and important (the observed trend being one that is reliably found in other experiments), whereas the data in the left panel may be important but are new. Model 1 in panel A predicts the majority of data quite well, but predicts the trend in question to go in the wrong direction; Model 2 in panel B does not predict the data as well quantitatively, but does predict the trend in question. The theorist would usually weight the qualitative misprediction more than the quantitative mispredictions: The first model would be modified (if possible), or rejected. The second model would probably be slated for further development.

As theory development proceeds further, quantitative measures become increasingly important, not least because those models that produce qualitative failures have already been deleted from consideration. At this stage, the precise choice of measure (e.g., maximum likelihood, chi-square, least square) becomes critical, as does the way in which error measures for different aspects of the data are combined (e.g., how best to combine prediction of accuracy with prediction of RT). By the time publication occurs, the models that have survived are those that have passed the earlier conceptual and qualitative tests (except, of course, for those models that the theorist is trying to reject, usually retained as "straw men" candidates). Because most of the models that are bypassed do

Figure 3. Confidence region 95% (2-dimensional) for two selected parameters (intercept and growth rate) fit separately to signal-to-respond data collected from three different conditions (labeled A, B, and C) from Nobel (1996).

not appear in the article, the implications to be drawn from their failure tend to be lost and are not available to the reader.

REPORTING A MODEL

Let us assume that all of the steps alluded to above are complete, and that a model has been developed and fit successfully to the data; that is, for a given model, a set of parameter values has been found that produces adequate predictions for the data (adequate in a "judgment" sense rather than a statistical sense). In practice, this is often the end of the story, as well as the culminating point of the article reporting the model. In fact, much more work is needed, and a good part of the project remains. The problem is that the theorist has presumably gained a grasp of the essentials of the model and the interactions of its components, but the reader has none of the information needed to gain the same level of understanding. It would generally not be sensible to try to lead the reader through the set of false starts and partial fits that eventually led to the development of the model being reported. Some of the models rejected during the process of development are certainly hopelessly inept false leads. However, the nature of the process of model invention usually means that the rejected variants are explored superficially and incompletely; thus, it would be difficult to verify firm conclusions concerning their viability.

What it is possible to do, and what should be done, are extensive additional analyses concerning the structure of the final model. Many important conceptual ideas ought to be conveyed to the reader: Which are the model components essential to produce the most important findings? Which combination of components produces the ob-

served patterns? What is the shape of the parameter space? What are the correlations and dependencies among parameters? For example, the model can be selectively lesioned by successive removal of its components, one at a time; a best fit is obtained for each submodel, and much can be learned by the pattern of degradations in predictions that occurs (examples of this sort include McClelland & Rumelhart, 1980, and Raaijmakers & Shiffrin, 1980). As another example, a parameter can be fixed at a value, and best fits obtained by varying the other parameters; by doing this for a range of values of the fixed parameter, something like a standard error can be obtained for that parameter value. In cases in which parameters are correlated, the same can be done for parameter pairs, as is illustrated with results from Nobel (1996) that are given in Figure 3. The answers obtained from such manipulations are important as an aid to understanding, not only for the reader, but also for the author. Regardless of the additional effort needed to obtain the relevant data and to carry out the associated analyses, they are omitted from publications far too often. It should be noted that since the final version of the model is precisely specified, such analyses are well suited for exploration by high-performance computational devices.

AN ILLUSTRATED EXAMPLE

Nobel's (1996) dissertation research provides a reasonable case study to serve as the basis for discussion: There was a very large database to which models had to be applied, and the models were not already specified, but either had to be built by extending existing models in new directions or had to be invented entirely.

In brief, Nobel presented lists of word pairs for study, followed by recognition or cued recall tests. Subjects either responded freely, as quickly and accurately as possible, or were given a signal to respond at various times after presentation of the test word. The lists varied in both length and study time. RTs were collected for all responses, and sufficient data were collected that entire RT distributions were compiled for each type of response in each condition. Faced with an enormous amount of data, we decided to start by modeling the cued recall data in the free response condition (about $\frac{1}{8}$ of the data). Even so, as is illustrated in Figure 4 (panels A–F), there were more than enough data to justify the use of a complex model with many free parameters.

The starting point for model development was the SAM model (see, e.g., Gillund & Shiffrin, 1984; Raaijmakers & Shiffrin, 1980, 1981). This model had proved most successful in fitting both qualitative trends and quantitative data such as probabilities of response, but had been applied largely to accuracy data. Considerable development was needed to extend the model to handle RTs.

Although the details of the SAM model are not essential for the present exposition, a very brief description may nevertheless provide some useful context. The SAM model assumed a long-term memory with separate traces termed "images," each image containing sensory information,

Figure 4. Accuracy and response time (RT) data from a cued recall study by Nobel (1996). The solid and dashed curves in each panel give the RT distributions for the type of response given in the labels at the top of each panel. For each type of response, the panel also gives the probability of each type of response (p), the mean RT (μ), the standard deviation of the RT (σ), and the median RT (m). Panels A and B, correct responses; panels C and D, intrusions; panels E and F, give-ups. Panels A, C, and E, short and long lists; panels B, D, and F, fast and slow lists.

semantic and content information, and context information (such as the setting in which the to-be-remembered items were studied). The amount of information stored in an image varied with the amount and type of rehearsal and coding at study. In an explicit memory test, some test information is provided to the subject (such as a word and a set of instructions telling the subject to decide whether that test word was on the list just studied). The subject constructs a memory probe typically consisting of content (e.g., the word) and context information (e.g., the current context cues in short-term memory). Each cue in the probe has a retrieval strength to each image in memory, higher if the cue and the other information in that image have been studied together longer. Each image is activated to a degree determined by multiplying the retrieval strengths between it and the cues in the probe. For *old/new* recognition, a decision is based on the sum of the activations across the images. To carry out cued recall (e.g., one member of a studied pair of words is presented as a cue, and the subject is to try to generate the other word of the pair), the subject engages in a sequential search of memory. In each step of the search, an image is sampled randomly, with replacement, with a probability proportional to the activation of the image divided by the sum of the activations across images. At each step (some of) the contents of the sampled image are examined, and if one or more words in that image are identifiable, a decision is made whether to output a word or to continue the search. Applied to accuracy data, this model usually had from four to eight parameters to be estimated, depending on the paradigm.

In extending the SAM model to handle RT data as well as accuracy data, we were led to models with 12–16 parameters; in no cases could we have accurately intuited the way in which the processes and parameters of the model produced detailed predictions. Thus we learned the predictions of each variant by running a simulation. The predictions were obtained through Monte Carlo simulations: there were many choice points at which divergent actions were determined by probabilities, and the simulations used a random number generator to make each such choice, generating pseudodata corresponding to every trial in every condition.

It was immediately evident that the simulation procedure operated at a painfully slow rate, making it a poor strategy to try to explore the parameter space for each variant in any systematic fashion. For example, a reasonably accurate set of predictions (so that most simulated points would be within, say, 5% of the correct *predicted* value) might have taken 5 min on a modern state-of-the-art workstation, for a single set of parameter values. In the following, we will ignore improvements in coding of the simulation program, which often are helpful, but require a great deal of effort on the part of the programmer and seldom improve times by more than 50% to 100%. That is, the speed problem is endemic. Could matters be helped by transfer to a high-performance computer? Perhaps, or perhaps not; we will return later to this issue. In any event, we operated under these constraints in a highly interactive and intuitive manner, obtaining predictions for relatively few parameter sets, thinking about the results, choosing a few additional parameter val-

ues for exploration, and so forth. The hope, of course, is that we would learn by examining these partial simulation results how the processes of the model interact to produce patterns of predictions. This would eventually lead us to develop models that would be serious contenders for fitting our data. Then, when we wished to obtain a "best fit" for any serious contender, we hoped that a knowledge of the mechanisms of the model and their interactions would allow us to approach a best fit after a very limited exploration of the parameter space, a much more limited exploration than is normally required by even the most efficient search algorithms.

This approach to the general problem can be defended, but not solely on the basis of time needed to reach a good fit. It turns out that the amount of thinking time between simulation results is often so large (perhaps a day to a week) that a rather extensive algorithmic search of the parameter space could have been carried out in that time. Nonetheless, we tended to mix small amounts of simulation with large amounts of thinking, for two very good reasons: First, a chief goal of the simulation work is the training of our intuitive understanding of the model's structure. Second, there was at every stage of this process an extremely high likelihood that some insight would be gained that would lead us to abandon the current model and replace it with some other variant. Thus we proceeded in this way, mixing long periods of cogitation with shorter periods of computation, for some months, until a reasonably successful model variant had been obtained.

The next step consisted of switching our sights to the free response recognition data, an even larger set of data than that available from cued recall. We hoped, of course, to use the model just developed for cued recall as the basis for the recognition model. At first we were optimistic because the simulation programs for recognition operated more quickly than those for cued recall, but it proved difficult to obtain an acceptable fit to the data. Many model variants were explored over several months. Eventually, we identified the apparent difficulty: Our variations of length and strength had produced accuracy differences, but virtually no RT differences. However, our choices of lengths and strengths had produced relatively small differences even in the accuracy measure, making it difficult to come to firm conclusions. Of course we were reluctant to abandon the entire class of SAM models without crystal clear data. We therefore redid the entire study, making a few improvements in design, but employing much larger variations in length and strength. Figure 5 (panels a and b) illustrates the main result by showing a representative portion of the data: Large accuracy differences occurred between conditions, but the RT distributions did not change.

These results indeed seemed at odds with the class of models based on SAM, but with a successful history of applying SAM in many domains, we were reluctant to reject this approach without a fight. We therefore embarked on an exploration of increasingly complex variants of SAM models (e.g., adding various random walk pro-

Figure 5. Partial results from a recognition study by Nobel (1996). The solid and dashed curves in each panel give the response time (RT) distributions for the type of response given in the labels at the top of each panel. For each type of response, the panel also gives the probability of each type of response (p), the mean RT (μ), the standard deviation of the RT (σ), and the median RT (m). RT distributions for hits do not differ as a function of list length (panel A) or list strength (panel B), even though probability differs considerably. A similar pattern was obtained for correct rejections, misses, and false alarms.

cesses with different properties). We were finally forced to accept the necessity of a new approach, at which point the whole process of model invention had to begin anew for recognition. Simultaneously, the new study produced a new set of cued recall data, generally in agreement with the previous data, but somewhat clearer and more diagnostic. When the previously successful cued recall model was applied to these new data, some relatively small failures of prediction occurred, failures that would nonetheless require further modeling to become rectified.

At this point we were faced with two different problems: First, model development was needed for cued recall; these amendments and extensions were relatively well defined (we were reasonably sure we knew what alterations were needed), but hard to carry out because of the slowness of the simulation programs. As mentioned earlier, as much as 5 min on a new workstation were needed to obtain a not-quite-accurate set of simulation predictions for a single set of parameter values, and we were dealing with a cued recall model having about 14–16 parameters. Second, we needed to develop an entirely new recognition model; we had hopes that it would

be in a form whose simulation would run quickly, but this was not guaranteed at the outset. We therefore proceeded simultaneously on two fronts. First, for cued recall, Nobel decided to switch both the software and hardware in order to improve the rate at which a global best fit could be found. Second, we began a course of model invention for recognition, using our available workstations. We describe these in turn.

High-Performance Parameter Estimation

Up to this point, the simulations and parameter estimation routines were implemented on the workstations (mainly DEC Alphas) available in our lab. For parameter estimation, a program developed at the CERN in Geneva was used: Among other algorithms, it includes a routine that uses combinations of the downhill simplex method (Nelder & Mead, 1965) and the Davidson–Fletcher–Powell (DFP) method (Fletcher & Powell, 1963). The downhill simplex method requires only function evaluations and not derivatives (as in the DFP method), and when the computational burden is small, it can work quickly; it is a relatively efficient algorithm (Press, Teukolsky, Vetterling, & Flannery, 1992). The DFP method is generally described as one of the most sophisticated search techniques available (Pierre, 1986).

The cued recall models that we were fitting to the data had grown more and more complex until they included as many as 20 highly interdependent free parameters. At this point, the simplex and DFP methods took approximately 3 days of continuous time as a single user on one of our workstations (a DEC Alphastation 255/300; 300-MHz CPU; 64 MB of RAM) to converge on something close to a best fit. We could perhaps have lived with such limits but for a new problem that developed along with the complexity of the models: The results were highly dependent on the order of the parameter definitions. When given starting values of the parameters, the minimization routines step through the multidimensional space while fixing some parameters and varying some others. Initially, the first parameters are varied, and when the error is at some minimum, those parameters are fixed, and the next set is varied. When the set of free parameters is large and interdependent, and when the parameters that come late in the order do not affect the error value, the program “gets stuck” in a local minimum.

To remedy this problem, we developed a simple genetic algorithm (see, e.g., Whitley, 1993). In our particular instantiation, the initial parameter values are drawn from normal distributions: The mean and standard deviation for each parameter are specified in an input file, as are the step sizes between values for each parameter; this allows us to set up a search as wide or as narrow as needed. The main advantage of this approach is that when given a large enough population (starting samples of the search space), the algorithm is able to “jump out” of local minima. At the same time, this introduces a new problem: The number of simulations required rises dramatically (7,500 for a relatively narrow search, and 40,000 for a rela-

tively wide search, vs. about 1,500 for the simplex and DFP methods).

To solve the speed issue we tried two approaches, both involving high-performance computers. First, we obtained an account on a supercomputer (Cray Y-MP) through the National Center for Supercomputing Applications. Since our simulations mainly involve probabilistic processes that depend heavily on random number generators instead of vector or matrix operations, the increase in speed was nominal and disappeared as the number of other users on the machine increased. Second, we used the parallel computer (Intel Paragon) through the Center for Innovative Computer Applications at Indiana University. With the help of one of the consultants, the genetic algorithm was modified to run on a partition of multiple nodes. Although each individual node is slower than our workstation as a single user, the use of n simultaneous nodes improved the solution rate by a factor of about n . Since the partitions on the parallel machine were not available all the time, or in large numbers, we used it to do relatively large searches of the parameter space and used the results as starting parameters for the simplex and DFP methods carried out on our workstations.

Figure 6. Predictions of an acceptable model for recognition response time (RT) distributions, for 2 of the 16 conditions in Nobel (1996). The solid and dashed curves in each panel give the observed and predicted RT distributions for correct rejections (panel A) and misses (panel B) for the short and rapidly presented lists. The panels also give the observed and predicted probabilities (p), mean RTs (μ), the standard deviations of the RTs (σ), and the median RTs (m). Predictions for other conditions were of similar quality.

Development of a New Recognition Model

The process of inventing a new recognition model began with many weeks of occasional thinking without interference by simulations of any kind. Eventually a critical insight led to a new and rather different class of models, based on element sampling. The corresponding simulations operated rapidly, and convergence in the parameter space was not too difficult to obtain. Not too many iterations of model variants were necessary before a remarkably successful model and fit were obtained. (An example of 1 of 16 figures is shown in Figure 6; the goodness-of-fit is typical.)

Unfortunately, this model was not entirely satisfactory on the conceptual level. Using the insights gained from the successful fit, therefore, we tried to go back to the SAM framework and build a new model incorporating what we now thought we knew to be the necessary properties. Many more months went by as we explored one variant after another, mostly attempting to marry a version of SAM with a random walk model. Unfortunately, none of these attempts proved successful. Eventually we changed tacks once again, and found a less conceptually satisfying way to combine SAM and the element sampling model that nonetheless produced a good fit. At this point, the main portion of the project was sufficiently complete so that we felt ready to publish the results.

Models of Other Theorists

There is of course more to the story. We next had to explore models of other theorists in the field; none of these had been developed for our paradigms, and they therefore had to be extended and amended, and fit to our data, again raising all the problems we have just described. We will skip this discussion for brevity's sake.

The Final Model

We have still to carry out the extensive analyses on our final model that will enable us to convey to the reader what we think we understand of its structure. For example, we will fix various parameters at extreme or default values and look to see what happens to the best fit that can be found with those restrictions; we will selectively delete certain processes from the model and find best fits for each such case; we will obtain confidence regions about each best parameter value and perhaps certain combinations of parameters. We have done relatively little of this as yet, but high-performance computing software and hardware would provide critical help in this enterprise.

CONCLUSIONS AND SUGGESTIONS

We hope we have illustrated the kinds of activities that go on in the long period between the initiation of model development and the fitting of the final model to the data. Our descriptions should make it clear that almost all of the processes of development, including design of experiments, analysis of data, design of models based on concepts and the literature, modification of models based on initial data analysis, learning how the processes built into the models affected predictions, and qualitative and quantitative fitting of models to data through the use of simulations, took place in a kind of parallel mixing pot, with each new discovery triggering further activities in other areas.

During this long process, an enormous number of models were investigated, in various degrees of depth. Briefly, we looked into Nobel's computer directories and made a count of the classes of models he investigated and the variants explored within each class. We found the result extraordinary—it is given in Figure 7.

Figure 7. A partial count of the number of different models for which parameter minimization was attempted during the process of developing the models reported in Nobel (1996). The tree shows the breakdown of the models into broad categories, following the breakdown of types in Nobel's file directory. Numbers at the terminal branches indicate number of models of each type.

The counts at the terminal branches of this tree are the number of model *variants* for which there was at least some attempt at parameter estimation for the purposes of best fit. These 64 models are actually an underestimate, since some minor variations simply replaced their predecessors. Each such variant had to have its parameter space explored to at least some degree; most of the parameter spaces were from about 14-dimensional to 20-dimensional, with many local minima. Each single point in a given parameter space required from several seconds to several minutes to produce a set of reasonably accurate predictions. Nor is this chart the entire story; another 30 models were fit to the signal-to-respond data (that we have not discussed).

What lessons might there be for high-performance computing applications (both hardware and software)? The procedures that we have outlined could not possibly have been reduced to any sort of rote set of estimation procedures, no matter how fast the computational devices. Learning the interaction of the mechanisms that made up the models was essential to the progress that was made. Nonetheless, at almost every stage of this process, it would have been enormously helpful to have had faster estimation procedures available, with one critical proviso: Faster computation would have helped only if it had been available in a form useful to the theorist. For example, batch processing, no matter how fast, would have been of little use. If high-performance computing had been available in *immediate, interactive* form, the entire project could have been cut to months instead of years, and could well have produced better results. Why? The speed constraints with which we operated caused us to stop the search of the parameter space well short of the point at which we had any certainty concerning the acceptability of the model, and we often collected so little pseudodata that the predictions were not entirely accurate. Thus we were probably led to incorrect intuitions on a number of occasions and probably were led to reject (at least temporarily) model variants that might have been successful.

High-performance computation in batch mode has an important role, of course. This occurs, for example, when-

ever a particular model is to be taken seriously, and it is important to find either a globally best fit or an acceptable fit. In addition, high-performance computation in batch mode would be most helpful in exploring the structure of a given model and the shape of the parameter space in the region of the best parameters.

To summarize, we estimate the advantages we gained and could have gained by high-performance computation at the final stage of the project (a stage carried out in batch mode) to be on the order of weeks; perhaps high-performance computing in batch mode could have reduced the time needed for this final stage from weeks to days. If interactive, immediate, high-performance computing had been available to us during the model invention stages of our project, our time line might have been cut from years to months.

REFERENCES

- FLETCHER, R., & POWELL, M. J. D. (1963). A rapidly convergent descent method for minimization. *Computer Journal*, **6**, 163-168.
- GILLUND, G., & SHIFFRIN, R. M. (1984). A retrieval model for both recognition and recall. *Psychological Review*, **91**, 1-67.
- MCCLELLAND, J., & RUMELHART, D. (1981). An interactive activation model of context effects in letter perception: Part I. An account of basic findings. *Psychological Review*, **88**, 375-407.
- NELDER, J. A., & MEAD, R. (1965). A SIMPLEX method for function minimization. *Computer Journal*, **7**, 308-313.
- NOBEL, P. A. (1996). *Response times in recognition and recall*. Unpublished doctoral dissertation, Indiana University, Bloomington.
- NOBEL, P. A., & SHIFFRIN, R. M. (1997). *Retrieval processes in cued recall and recognition*. Manuscript submitted for publication.
- PIERRE, D. A. (1986). *Optimization theory with applications*. Mineola, NY: Dover.
- PRESS, W. H., TEUKOLSKY, S. A., VETTERLING, W. T., & FLANNERY, B. P. (1992). *Numerical recipes in C: The art of scientific computing* (2nd ed.). Cambridge, MA: Harvard University Press.
- RAAIJMAKERS, J. G. W., & SHIFFRIN, R. M. (1980). SAM: A theory of probabilistic search of associative memory. In G. H. Bower (Ed.), *The psychology of learning and motivation: Advances in research and theory* (Vol. 14, pp. 207-262). New York: Academic Press.
- RAAIJMAKERS, J. G. W., & SHIFFRIN, R. M. (1981). Search of associative memory. *Psychological Review*, **88**, 93-134.
- WHITLEY, D. (1993). *A genetic algorithm tutorial* (Tech. Rep. No. CS-93-103). Fort Collins: Colorado State University, Department of Computer Science.