COMPUTER-INTENSIVE STATISTICAL METHODS

The term "computer-intensive" was first applied to statistical methods in connection with bootstrap techniques [4]. It has since come to describe a set of ideas that depend in some essential way on the availability of high-speed computation. Some other examples include generalized linear models*, nonparametric regression* ("smoothers"), generalized additive models*, classification* and regression trees, Gibbs sampling*, the EM algorithm*, proportional-hazards regression (*see* COX'S REGRESSION MODEL), multiple imputation*, and robust multivariate analysis* (*see also* MINIMUM VOLUME ESTIMATION). A standard linear regression*, even a very large one, isn't included in this definition, because the basic ideas are completely described by classical mathematics.

Of course there is no way to render such a broad definition precise. Methods that seem computer-intensive now may look classical after a few more years of experience, theoretical development, and improvements in computational equipment. In what follows we describe a few examples of current computer-intensive techniques, with no pretense at completeness.

Before the advent of electronic computation one could picture statistics in terms of the tension between two poles, one labeled "mathematics" and the other "applications." The triumph of classical statistics was the production of a mathematically sound theory that was sufficiently realistic to handle a wide variety of applications. Now the picture of statistics is a triangle, with "computation" at the third pole. It isn't that mathematics has disappeared from the world of statistics. Rather, pure mathematical arguments have been augmented with explanations phrased in terms of well-understood computer algorithms.

At their best, computer-intensive statistical methods can be seen as powerful but direct extensions of ideas in mathematical statistics. In our first example below, a bootstrap-t confidence interval (*see* BOOTSTRAPPING II) is introduced as a standard Student's t interval (*see* STUDENT'S t-TESTS), except one for which we need to generate a special t-table (*see* t-DISTRIBUTION) for each new application. This kind of conspicuous computational consumption, unthinkable even a few decades ago, underlies the development of all computer-intensive techniques.

Here we give a brief description of three quite different examples of computer-intensive statistical inference.

BOOTSTRAP-t CONFIDENCE INTERVALS

Our first example is the bootstrap-t method for setting good approximate confidence intervals*, proposed in [6]. Suppose that an observed data set x has yielded an estimate $\hat{\theta} = s(x)$ for a parameter of interest θ, and also an asymptotically accurate standard error* estimate $\hat{\sigma} = \mathrm{se}(x)$ for the uncertainty in $\hat{\theta}$. We could use the *standard intervals* $\hat{\theta} \pm z^{(\alpha)}\hat{\sigma}$ (where $z^{(.95)} = 1.645$ would give the usual 90% coverage probability) as approximate confidence intervals for θ. The bootstrap-t method offers a computationally intensive improvement over the standard intervals.

A generalized form of the usual Student's t-statistic is

$$T = (\hat{\theta} - \theta)/\hat{\sigma}. \qquad (1)$$

In the familiar case where θ is an expectation, $\hat{\theta}$ equals the sample mean* $\bar{x}$, and $\hat{\sigma} = [\sum(x_i - \bar{x})^2/n(n-1)]^{1/2}$, then T equals Student's t-statistic. If we knew the percentiles* $T^{(\alpha)}$ of T, then we could obtain confidence intervals for θ, for example, the two-sided 90% confidence interval

$$\{\theta : \theta \in [\hat{\theta} - \hat{\sigma}T^{(.95)}, \hat{\theta} - \hat{\sigma}T^{(.05)}]\}. \qquad (2)$$

In the genuine Student's t case, where we assume an underlying normal distribution*, $T^{(\alpha)}$ equals $t^{(\alpha)}_{n-1}$, the 100αth percentile point of a Student's t variate with $n-1$ degrees of freedom. However, for most choices of θ, $\hat{\theta}$, and $\hat{\sigma}$ we won't be able to compute the percentiles $T^{(\alpha)}$.

The bootstrap-t idea is to estimate the percentiles of T by bootstrap sampling. If $x = (x_1, x_2, \ldots, x_n)$ were obtained by random sampling from some probability distribution F, and if $\hat{F}$ denotes an estimate of F, then

a bootstrap sample $x^* = (x_1^*, x_2^*, \ldots, x_n^*)$ is a random sample of size n from $\hat{F}$. We independently generate a large number B of bootstrap samples, $x^*(1), x^*(2), \ldots, x^*(B)$, with $B = 1000$ being sufficient for most applications, and for each one compute the bootstrap analogue of (1),

$$T^* = \frac{\hat{\theta}^* - \hat{\theta}}{\hat{\sigma}^*} = \frac{s(x^*) - s(x)}{\mathrm{se}(x^*)}. \quad (3)$$

The percentiles $T^{*(.95)}$ and $T^{*(.05)}$ of the values $T^*(1), T^*(2), \ldots, T^*(B)$ are substituted into (2) to give the bootstrap-t approximate confidence interval

$$\{\theta : \theta \in [\hat{\theta} - \hat{\sigma}T^{*(.95)}, \hat{\theta} - \hat{\sigma}T^{*(.05)}]\}. \quad (4)$$

Hall [15] showed that under quite general circumstances the bootstrap-t intervals (3) are an order of magnitude more accurate than the standard intervals $\hat{\theta} \pm 1.645\hat{\sigma}$. The standard intervals have actual probability $.05 + O(1/\sqrt{n})$ (*see* O,o NOTATION) of θ exceeding the upper limit, or lying below the lower limit. The corresponding error probabilities for the bootstrap-t intervals are $.05 + O(1/n)$. This form of second-order accuracy can be quite impressive in practice, as the following example shows.

Figure 1 is a scatter plot* of data $x = (x_1, x_2, \ldots, x_{26})$ from 26 children, each of whom took two tests of spatial cognition. The tests are called A and B, so each data point x_i consists of a pair of measurements $x_i = (A_i, B_i)$ for $i = 1, 2, \ldots, n = 26$. The Pearson sample correlation* coefficient between A and B is $\hat{\theta} = .821$. We can use the bootstrap-t algorithm to generate second-order accurate confidence intervals for the true correlation coefficient θ.

Suppose first that F, the unknown distribution giving the points $x_i = (A_i, B_i)$, is bivariate normal. We estimate F by $\hat{F}$, its maximum-likelihood estimate*, and generate bootstrap samples $x^* = (x_1^*, x_2^*, \ldots, x_{26}^*)$ by random sampling from $\hat{F}$. The histogram* on the left side of Fig. 2 shows 2000 bootstrap replications of $T^* = (\hat{\theta}^* - \hat{\theta})/\hat{\sigma}^*$, where $\hat{\theta}^*$ is the sample correlation based on the bootstrap data x^* and where

$$\hat{\sigma}^* = (1 - \hat{\theta}^{*2})/\sqrt{26}, \quad (5)$$

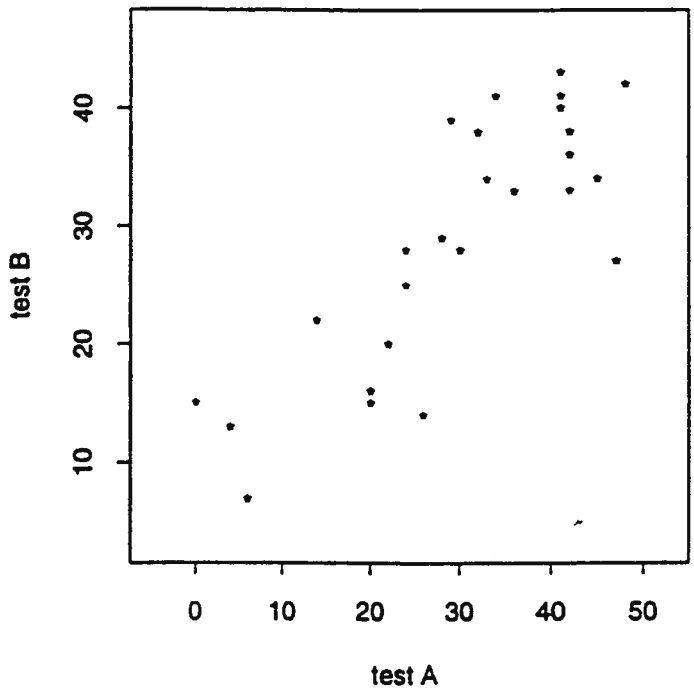

Figure 1 The spatial test data.

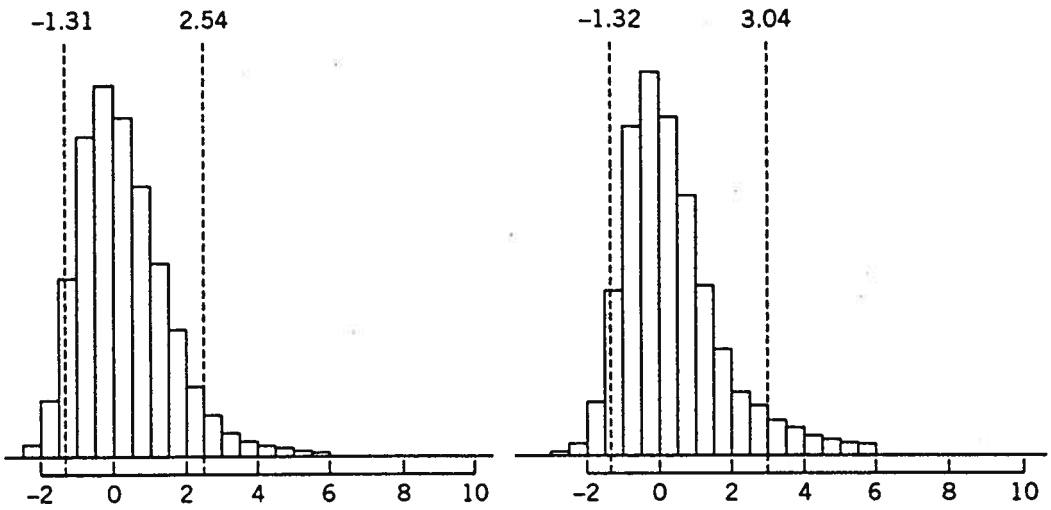

Figure 2 Normal theory (left) and nonparametric (right) bootstrap-t histograms; the 5th and 95th percentiles are indicated.

the delta-method estimate (*see* STATISTICAL DIFFERENTIALS, METHODS OF) of the standard error for $\hat{\theta}^*$. The 5th and 95th percentiles of the T^*-distribution are $(-1.31,2.54)$, very different from the corresponding percentiles $(-1.71,1.71)$ for a standard t_{25}-distribution. The bootstrap-t approximate confidence interval (4) is quite similar to the exact confidence interval for θ, as shown on the left side of Table 1. The increase in accuracy over the standard interval is striking.

In this case, of course, we don't need the bootstrap-t intervals, since a classical exact solution exists. The virtue of computer-intensive methods is their ability to extend classical results to situations which are mathematically intractable. Suppose we don't wish to assume that F is bivariate normal (a poor assumption in this case). Second-order accuracy of the bootstrap-t intervals holds under quite general conditions. The right side of Fig. 2 is the *nonparametric* bootstrap-t distribution of $T^* = (\hat{\theta}^* - \hat{\theta})/\hat{\sigma}^*$. Now $\hat{F}$ is the empirical distribution of x, the distribution putting probability $\frac{1}{26}$ on each point in Fig. 1. The standard error estimate $\hat{\sigma}^*$ is based on the nonparametric delta method applied to the bootstrap sample correlation coefficient $\hat{\theta}^*$; $\hat{\sigma}^*$ can be obtained either by numerical differentiation or by substitution into a generalized version of (5), e.g. formula (27.8.1) of ref. [3]. The upper tail of the T^*-distribution is longer in the nonparametric case, giving $T^{*(.95)} = 3.04$.

Table 1 Two-Sided .90 Confidence Intervals for the Correlation Coefficient, Spatial Test Data

	Interval	
Approximation	Normal Theory	Nonparametric
Exact	(.665,.902)	?
Boot T	(.653,.905)	(.627,.905)
BC_a	(.668,.901)	(.675,.892)
Standard	(.716,.926)	(.726,.916)

Table 1 also shows approximate confidence limits based on another bootstrap method, the BC_a (bias-corrected and accelerated) bootstrap [7, 5]. The BC_a method is also second-order accurate, and has some advantages over the bootstrap-t procedure. It does not require calculation of a standard error estimate $\hat{\sigma}^*$. It is transformation*-invariant, so, for example, the BC_a limits for $R = \sqrt{1 - \theta^2}$ are obtained from the same transformation on the limits for θ. In practice the BC_a method seems to perform more stably in nonparametric situations. There is no gold standard on the right side of Table 1, but the BC_a intervals are probably preferable for general nonparametric situations.

It took a decade of hard work to produce the original Student's *t*-tables. Now it takes only a few minutes to generate a bootstrap "*t*-table" that applies to the particular data set at hand. Efron and Tibshirani [9] give a general introduction to the bootstrap and related statistical methods. As shown by this example, computer-intensive statistical theory has developed in response to the challenge of making effective use of modern computational equipment. We will present two more examples.

CLASSIFICATION AND REGRESSION TREES

In this section we describe the tree-based approach to classification*, as developed in the CART (Classification and Regression Trees) methodology of Breiman et al. [2]. We illustrate CART with a real example.

In an experiment designed to provide information about the causes of duodenal ulcers (see Giampaolo et al. [14]), a sample of 745 rats were each administered one of 56 model alkyl nucleophiles. Each rat was later autopsied for the development of duodenal ulcer, and the outcome was classified as 1, 2, or 3 in increasing order of severity. There were 535 class 1, 90 class 2, and 120 class 3 outcomes. The objective in the analysis of these data was to ascertain which of 67 characteristics of these compounds were associated with the development of duodenal ulcers. When applied to these data, the CART procedure produced the classification tree shown in Fig. 3.

At each node of the tree a rule is given, and observations which satisfy the rule are assigned to the left branch while the others go to the right branch. The shaded leaves of the tree shown in Fig. 3 are called *terminal nodes*. Each observation is assigned to one of the terminal nodes according to the answers to the questions. For example, a rat that received a compound with dipole moment ≤ 3.56 and melting point >98.1 would go left then right and end up in the terminal node marked [13,7,41]. Triplets of numbers such as [13,7,41] below each terminal node indicate the membership at that node; i.e., there are 13 class 1, 7 class 2, and 41 class 3 observations at this terminal node.

Before discussing how the CART procedure built this tree, consider how it is used for classification. Each terminal node is assigned a class

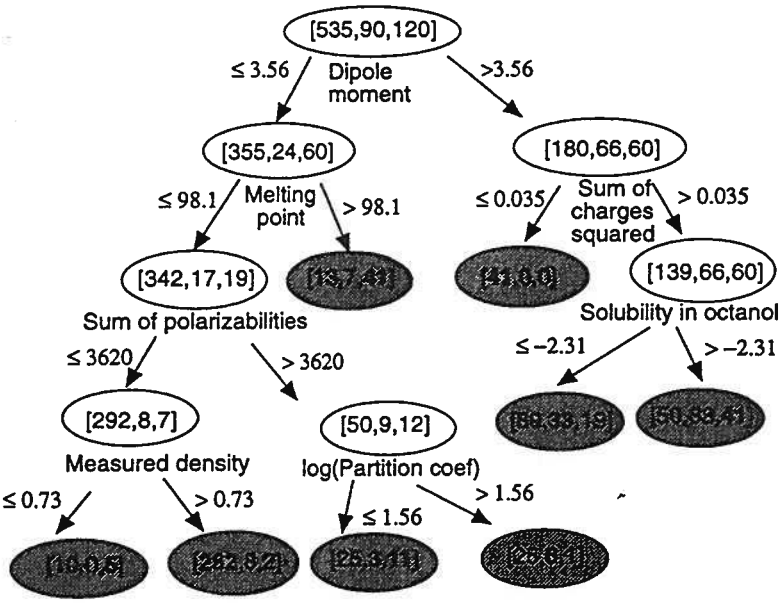

Figure 3 CART tree. Classification tree from the CART analysis of data on duodenal ulcers. At each node of the tree a rule is given, and observations which satisfy the rule are assigned to the left branch while the others go to the right branch. The shaded nodes are the terminal nodes (or leaves) of the tree.

(1, 2, or 3). The most obvious way to assign classes to the terminal nodes would be to use a majority rule and assign the class that is most numerous in the node. Using a majority rule, the node marked [13,7,41] would be assigned to class 3, and all of the other terminal nodes would be assigned to class 1. In this study, however, the investigators decided that it is worse to misclassify an animal with a severe ulcer than one with a milder ulcer, and hence they prescribed a higher penalty for those errors. Using the prescribed penalties, a best rule for each terminal node can then be worked out. In Fig. 3 the assigned class is indicated by the boldface number at each terminal node; for example, the node at the bottom left marked [10,0,5] has the **5** in boldface and hence is a class 3 node.

The tree can be interpreted as follows. The top (or root) node was split on dipole moment. A high dipole moment indicates the presence of electronegative groups. This split separates the class 1 and 2 compounds; the ratio of class 2 to class 1 in the right split, 66/190, is more than 5 times as large as the ratio 24/355 in the left split. However, the class 3 compounds are divided equally, 60 on each side of the split. If, in addition, the sum of squared atomic charges is low, then CART finds that all compounds are class 1. Hence ionization is a major determinant of biologic action in compounds with high dipole moments. Moving further down the right side of the tree, the solubility in octanol then (partially) separates class 3 from class 2 compounds. High octanol solubility probably reflects the ability of the compound to cross membranes and to enter the central nervous system.

On the left side of the root node, compounds with low dipole moment and high melting point were found to be class 3 severe. Compounds at this terminal node are related to cysteamine. Compounds with low melting points and high polarizability, all thiols in this study, were classified as class 2 or 3 with the partition coefficient separating these two classes. Of those chemicals with low polarizability, those of high density are class 1. These chemicals have high molecular weight and volume, and this terminal node contains the largest number of observations. On the low-density side of the split are all short-chain amines.

The data set of 745 observations is called the *learning sample*. We can work out the misclassification rate for each class when the tree in Fig. 3 is applied to the learning sample. Looking at the terminal nodes that predict classes 2 or 3, the number of errors for class 1 is $13 + 89 + 50 + 10 + 25 + 25 = 212$, so the apparent misclassification rate for class 1 is $212/535 = 39.6\%$. Similarly, the apparent misclassification rates for classes 2 and 3 are 56.7% and 18.3%, and the overall apparent misclassification rate is 38.2%. But this is misleading, since misclassification rates in the learning sample can be badly biased downward, for reasons discussed below.

How does CART build a tree like that in Fig. 3? CART is a fully automatic procedure that chooses the splitting variables and splitting points that best discriminate between the outcome classes. For example, dipole moment ≤ 3.56 is the split that was determined to best separate the data with respect to the outcome classes. CART chose both the splitting variable (dipole moment) and the splitting value (3.56). Having found the first splitting rule, new splitting rules are selected for each of the two resulting groups, and this process is repeated.

Rather than stopping when the tree is some reasonable size, a large tree is constructed and then pruned from the bottom. This latter approach is more effective in discovering interactions* that involve several variables.

This brings up an important question: How large should the tree be? If we were to build a very large tree with only one observation in each terminal node, then the apparent misclassification rate would be 0%. However, this tree would probably do a poor job predicting the outcomes for a new sample of rats. The reason is that the tree is too specific to the learning sample; in statistics (especially regression analysis), this problem is called *overfit*.

The best-sized tree would be the one that had the lowest misclassification rate for some new data. Thus if we had a second data set available (a test sample), we could apply the trees of various sizes to it and then choose the one with lowest misclassification rate.

Of course in most situations we do not have extra data to work with. Data are so precious that we want to use all of them to estimate the best possible tree. The method of cross-validation* is what CART uses to choose the tree size, a procedure that attempts to mimic the use of a test sample. It works by dividing the data up into ten groups of equal size, building a tree on 90% of the data, and then assessing its misclassification rate on the remaining 10% of the data. This is done for each of the ten groups in turn, and the total misclassification rate is computed over the ten runs. The best tree size is then that which gives the lowest misclassification rate. This is the size used in constructing the final tree from all of the data. The crucial feature of cross-validation is the separation of data for building and assessing the trees; each one-tenth of the data acts as a test sample for the other nine-tenths.

The process of cross-validation not only provides an estimate of the best tree size, it also gives a realistic estimate of the misclassification rate of the final tree. The apparent rates computed above are often unrealistically low because the training sample is used both for building and assessing the tree. For the tree in Fig. 3, the cross-validated misclassification rate was about 48%, or 10% higher than the learning-sampling misclassification rate. It is the cross-validated error rate that provides an accurate assessment of how effective the tree will be in classifying a new sample of animals.

CART is one of an increasing number of flexible regression and classification methods that have recently been developed. Other related methods are generalized additive models* and multivariate additive regression splines* [10]. All of these proposals exploit the power of the computer to discover structure in high-dimensional multivariate data.

GIBBS SAMPLING FOR BAYESIAN ANALYSIS

The statistical techniques discussed so far have been frequentist procedures. That is, the unknown parameters are treated as constants rather than random variables. In recent years, computer-intensive methodology has also led to some significant advancements in Bayesian inference*.

Efron and Feldman [8] analyze data from the Stanford arm of a large clinical trial designed to test the efficiency of the cholesterol-reducing drug cholestyramine. The data consist of two measurements on 164 men: a cholesterol reduction score R and compliance score C, the proportion of the intended dose each man actually took (measured by counting packets of unused cholestyramine returned to the clinic). The data are shown in Fig. 4, along with a least-squares* fit of the quadratic model

$$R = \beta_0 + \beta_1 C + \beta_2 C^2.$$

The least-squares estimates are $\hat{\beta}_0 = 4.705$, $\hat{\beta}_1 = 0.134$, and $\hat{\beta}_2 = 0.004$. We see that better compliance tends to be associated with a greater reduction in cholesterol, just as we might hope.

One of the main challenges in the Bayesian approach is to compute marginal posterior distributions* for the individual parameters of interest. Suppose, for example, that we want to make inferences about the coefficient of compliance (β_1) and squared compliance (β_2) in the quadratic regression model example above. The techniques that we describe here are not needed

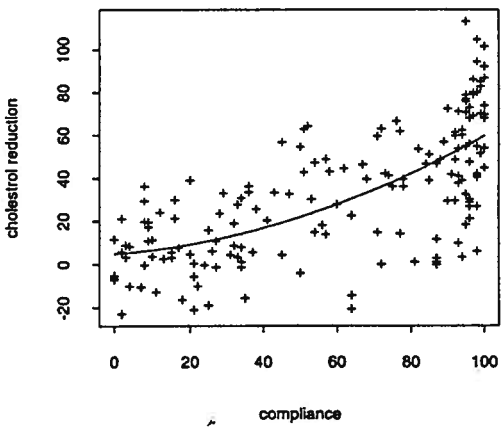

Figure 4 164 men in the Stanford arm of experiment LRC–CPPT: the vertical axis is cholesterol reduction score; the horizontal axis is compliance, measured as the percentage of intended cholestyramine dose actually taken. The average compliance was 60%. The smooth curve is the quadratic regression fit to the 164 points by least squares.

when there are only two parameters of interest, but this choice simplifies the explanation. The top panel of Fig. 5 shows the probability contours of a typical posterior distribution for the two parameters. Here's how that posterior distribution comes above. We start with a prior distribution* for β_1 and β_2, denoted by $\pi(\beta_1, \beta_2)$. The prior distribution reflects our knowledge about the parameters before we collect the data. After we collect the data, the quadratic regression model defines a likelihood* $f(\beta_1, \beta_2)$ that specifies how the distribution of the data depends on the parameters. Finally, we combine prior and likelihood into the posterior using Bayes' theorem*:

$$p(\beta_1, \beta_2) = \frac{\pi(\beta_1, \beta_2) f(\beta_1, \beta_2)}{\int \pi(\beta_1', \beta_2') f(\beta_1', \beta_2')\, d\beta_1'\, d\beta_2'}.$$

For simplicity we have chosen the prior and likelihood to correspond to normal distributions, and hence the posterior distribution is bivariate normal (and thus the contours in Fig. 5 are elliptical).

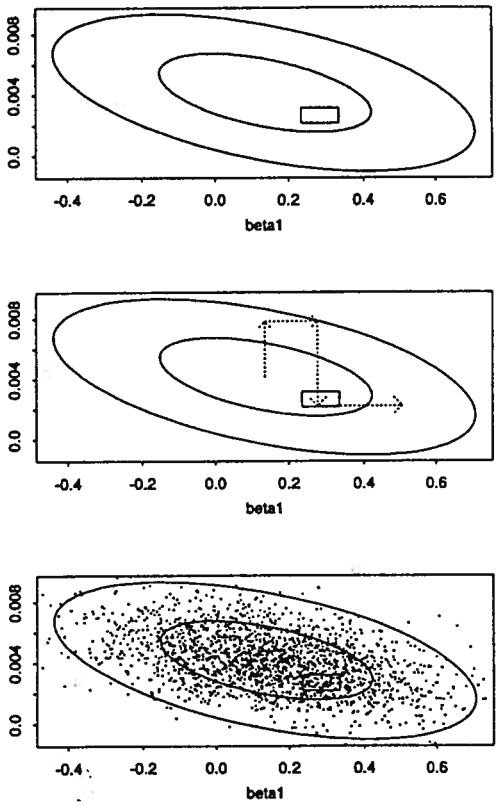

Figure 5 Illustration of Gibbs sampling. The top panel shows the probability contours (ellipses) for a hypothetical posterior distribution of the coefficients of compliance (β_1) and squared compliance (β_2). The goal is to find the probability of the rectangular region. The middle panel shows the steps taken in a typical Gibbs-sampling run. The bottom panel displays the results of 1000 runs. The probability of the rectangular region is estimated by the proportion of points falling in the rectangle.

Given a posterior distribution, we might ask such questions as "What is the probability that β_1 is less than .13?" or "What is the probability that $.23 \leq \beta_1 \leq .33$ and $.002 \leq \beta_2 \leq .003$?" The answer to this latter question corresponds to probability content of the rectangle in Fig. 5. A direct approach for answering such questions would involve numerical integration of the posterior distribution over the region of interest. When there are many parameters, this can be a computationally difficult problem. But often there is enough information about the problem that it is easy to sample from the conditional distributions of each parameter given the rest. It turns out that by successive sampling from these conditional distributions, we end up with a sample that has approximately the desired distribution. The procedure is known as Gibbs sampling*, and is due to Gelfand and Smith [12], following work of Geman and Geman [13] and Tanner and Wong [20].

To illustrate Gibbs sampling, look back at the top panel of Fig. 5. Pretend that we are given only the conditional probabilities for this distribution; here they have simple normal forms. How can we obtain a sample of β_1 and β_2 values from the probability distribution of Fig. 5?

Gibbs sampling answers this by first taking some starting value of β_1, for example $\beta_1 = .01$. Then it generates a random β_2 from the conditional distribution of β_2 given $\beta_1 = .01$. Suppose it obtains $\beta_2 = .007$. Then it generates a random β_1 from the conditional distribution of β_1 given $\beta_2 = .007$. If it obtains, for example, $\beta_1 = .03$, then it generates a random β_2 from the conditional distribution of β_2 given $\beta_1 = .03$, and so on. Continuing this, say B times, gives a final pair β_1 and β_2; call these β_1^1 and β_2^1.

This process of alternating conditional sampling is depicted in the middle panel of Fig. 5.

The sequence of β_1 and β_2 values is shown by the arrows.

Suppose now that we repeat this entire process for $B = 1000$ times; thus we end up with a sample of 1000 pairs $(\beta_1^1, \beta_2^1), \ldots, (\beta_1^{1000}, \beta_2^{1000})$. These are represented by the points in the bottom panel of Fig. 5. Notice how these 1000 values have a distribution that approximately matches the bivariate normal contours.

Finally, to estimate the probability that (β_1, β_2) falls in the rectangle, we simply count up the number of points in the rectangle and divide by 1000. Table 2 shows how this approximation improves as the length B of the chain increases. The approximation is often quite accurate when the chain reaches 50 in length.

In probabilistic terms, the Gibbs sampling approach effectively creates a Markov process* whose stationary distribution is the posterior distribution of interest. It is an attractive method when it is easy to sample from the distribution of each parameter given the others. In other problems there are alternative approaches to Markov-chain sampling of the posterior, for example the Metropolis–Hastings procedure described by Hastings [18] (*see* MARKOV CHAIN MONTE CARLO ALGORITHMS).

Gibbs sampling and its relatives offer a simple, powerful approach to Bayesian integration problems. They have already proved to be useful in other disciplines such as physics and computer science and are now being applied to problems of statistical inference such as those described above. Like the bootstrap, they do not require sophisticated mathematical work for each new application. Current research focuses on the refinement of these procedures to make them suitable for routine statistical practice.

Table 2 Gibbs Sampling Approximation

Length of Chain	No. in rectangle[a]
1	24
5	23
25	34
50	29

[a]Out of 1000. Exact probability = .0292.

DISCUSSION

This entry highlights some of the new computer-intensive methods that have been developed in the statistical field. But what we have discussed in this short article is just the tip of the iceberg. Some of the other interesting developments include projection pursuit* [11]; its cousin in artificial intelligence, neural networks* (see, e.g., ref [19]); and the ACE [1] and AVAS [21] algorithms for transformations. More computationally intensive bootstrap methods have been proposed, most notably bootstrap iteration [16].

With the introduction of more ambitious statistical tools comes the challenge of how to use them effectively in statistical practice. Questions such as "When should I use a complex model?" and "What inferences can I draw from my analysis?" become more difficult to answer. Computer-intensive methods provide a pressing motive for addressing these problems, but much work still needs to be done. At this point in time, the theory of inference has failed to keep pace with the development of new techniques.

References

[1] Breiman, L. and Friedman, J. (1985). Estimating optimal transformation for multiple regression and correlation (with discussion). *J. Amer. Statist. Ass.*, **80**, 580–619.

[2] Breiman, L., Friedman, J., Olshen, R., and Stone, C. (1984). *Classification and Regression Trees.* Wadsworth, Belmont, Calif.

[3] Cramér, H. (1946). *Mathematical Methods of Statistics.* Princeton University Press.

[4] Diaconis, P. and Efron, B. (1983). Computer intensive methods in statistics. *Sci. Amer.*, **248**, 115–130.

[5] DiCiccio, T. and Efron, B. (1992). More accurate confidence limits in exponential families. *Biometrika*, **79**, 231–245.

[6] Efron, B. (1979). Bootstrap methods: another look at the jackknife. *Ann. Statist.*, **7**, 1–26.

[7] Efron, B. (1987). Better bootstrap confidence intervals (with discussion). *J. Amer. Statist. Ass.*, **82**, 171–200.

[8] Efron, B. and Feldman, D. (1991). Compliance as an explanatory variable in clinical trials. *J. Amer. Statist. Ass.*, **86**, 9–26.

[9] Efron, B. and Tibshirani, R. (1993). *An Introduction to the Bootstrap*. Chapman & Hall, New York.

[10] Friedman, J. (1991). Multivariate adaptive regression splines (with discussion). *Ann. Statist.*, **19**, 1–141.

[11] Friedman, J. and Tukey, J. (1974). A projection pursuit algorithm for exploratory data analysis. *IEEE Trans. Comput. C*, **23**, 881–889.

[12] Gelfand, A. E. and Smith, A. F. M. (1990). Sampling based approaches to calculating marginal densities. *J. Amer. Statist. Ass.*, **85**, 398–409.

[13] Geman, S. and Geman, D. (1984). Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images. *IEEE Trans. Pattern Anal. Machine Intell.*, **6**, 721–741.

[14] Giampaolo, C., Gray, A., Olshen, R., and Szabo, S. (1991). Predicting induced duodenal ulcer and adrenal necrosis with classification trees. *Proc. Nat. Acad. Sci. U.S.A.*

[15] Hall, P. (1988). Theoretical comparison of bootstrap confidence intervals (with discussion). *Ann. Statist.*, **16**, 953–985.

[16] Hall, P. and Martin, M. (1988). On bootstrap resampling and iteration. *Biometrika*, **75**, 667–671.

[17] Hastie, T. and Tibshirani, R. (1990). *Generalized Additive Models*. Chapman & Hall, New York.

[18] Hastings, W. K. (1970). Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, **57**, 97–109.

[19] Hertz, J., Krogh, A., and Palmer, R. (1991). *Introduction to the Theory of Neural Computation*. Addison-Wesley, Redwood City, Calif.

[20] Tanner, M. and Wong, W. (1987). The calculation of posterior distributions by data augmentation (with discussion). *J. Amer. Statist. Ass.*, **82**, 528–550.

[21] Tibshirani, R. (1988). Estimating transformations for regression via additivity and variance stabilization. *J. Amer. Statist. Ass.*, **83**, 394–405.

B. EFRON
R. TIBSHIRANI

(BOOTSTRAPPING I, II
CLASSIFICATION
CURVE FITTING
GENERALIZED ADDITIVE MODELS
GIBBS SAMPLING
LOCAL REGRESSION
MARKOV-CHAIN MONTE CARLO ALGORITHMS
NEURAL NETWORKS
NONPARAMETRIC REGRESSION
STATLIB)