Article for:

Encyclopedia of Biostatistics. P. Armitage and T. Colton (eds), Wiley, 1998.
ISBN 0471 97576 1

Computer-Intensive Methods

One sense of "computer-intensive" statistics is just statistical methodology which makes use of a large amount of computer time. (Examples include the bootstrap, jackknife, smoothing, image analysis and many uses of the EM algorithm.) However, the term is usually used for methods which go beyond the minimum of calculations needed for an illuminating analysis, for example by replacing analytical approximations by computational ones, or requiring numerical optimization or integration over high-dimensional spaces. We introduce the subject by a very simple yet useful example, then consider some of the areas in which computer-intensive methods are used, to give a flavour of current research.

A Simple Example

Let us examine the idea of a Monte Carlo test (see for example [60]). Suppose we have a test statistic T, large values of which indicate a departure from a simple null hypothesis. The traditional analysis is to report *either* the P value $p = P(T > T_{\text{obs}})$ *or* whether p falls into one of the conventional ranges. However, to compute p we do need to know the distribution of T under the null hypothesis. Traditionally either T was chosen because its exact distribution was tractable, or some large-sample approximation was used, perhaps a normal distribution.

The Monte Carlo test depends on our being able to simulate under the null hypothesis; for each of m replications we generate some artificial data from the null hypothesis and compute the value T_i of the test statistic. One idea is to use the empirical distribution of the T_i as an approximation to the null-hypothesis distribution of T, that is to compute $\hat{p} = \#\{i : T_i > T_{\text{obs}}\}/m$. However, Monte Carlo tests use a clever variation. If the null hypothesis is true, we have not m but $m+1$ samples from the null hypothesis, the $T_1, \ldots, T_m$ that we generated plus T itself. A simple counting argument shows that

$$P(T \text{ is amongst the } r \text{ largest}) = r/(m+1)$$

Thus we can obtain an *exact* 5% test by taking $r = 1, m = 19$ or $r = 5, m = 99$ or $r = 25, m = 499, \ldots$.

When Monte Carlo tests were first proposed by Barnard [4] in 1963 they would have been rather slow to compute, and have needed extensive programming. When the idea was rediscovered in 1975 for some problems in spatial statistics (see [59]) increased computing power had made Monte Carlo tests much more attractive, although the choice of m was still severely limited by computing resources. Eventually good analytical approximations were found for those test statistics under the simpler null hypotheses but they are hardly ever used as it has become easy to use the exact Monte Carlo test.

This example has many of the key features of computer-intensive methods: it makes use of a simple calculation repeated many times, it relaxes the distributional assumptions

needed for analytical results, and it is in principle exact given an infinite amount of computation. Rather than considering large amounts of data, we consider large amounts of computation, as the ratio of cost of computation to data collection is continually falling.

It is also important that it allows us to avoid asymptotics. Large-sample results are only useful in practice by providing an approximate distribution theory. In the spatial-statistics context there are "several ways to infinity" [61] – that is several possible asymptotic regimes – and it is only possible to find good enough approximations by combining leading terms from all of them. When we consider neural networks below, we will see it is common to increase the complexity of the model with the amount of data to hand, so large-sample results are never appropriate (and in the real world models are always false).

Graphics

A considerable amount of computer power is used to plot or print graphs, and even more is needed for dynamic graphics, for example to rotate views of data, interactively change the bin size of a histogram or to highlight or identify points se Graphical Displays). We now take this for granted (although systems to do this are described in section on Programming Environments). As computer power grows, researchers are exploring ways to compute large numbers of views and let the software arrange the "interesting" ones to show to the user.

The Trellis system (see Becker et al. [7]; this is now part of S-PLUS [45]) is based on presenting graphical summaries of many "slices" of the data in a systematic way. Asimov's *grand tour* [3, 18] shows the user a continually rotating series of views (projections) of multidimensional data of real-valued variates, but with four or more variables it is almost impossible for the user to screen the series for interesting views. (See the discussion in [38].) Projection pursuit [17, 18, 25, 26, 38, 39, 42, 43] replaces human inspection by optimizing (locally) the "interestingness" of a view. Sometimes the results can be frustrating, but at other times very interesting structures are revealed (see, for example, [64]).

Simulation-Based Approaches

A *reductio ad absurdum* view of statistical methods is that they reduce to either the optimization or the integration of some function, with Bayesian methods majoring on integration. For reasonably realistic models the integration is often (extremely) computer-intensive. (Evans and Swartz [23] review analytical as well as Monte Carlo methods to approximate integrals in statistics.)

Simulation provides a very simple way to perform an integration such as $\phi = Ef(X)$. Just generate m examples $X_1, \ldots, X_m$ from the distribution of X and report the average of $f(X_i)$. It is not usually a good way to find an accurate estimate of ϕ, for the central limit theorem (if applicable) suggests that the average error decreases at rate $1/\sqrt{m}$. For smooth functions in a small number of dimensions numerical quadrature can do better, and in a moderate number of dimensions it may be better to use non-independent samples X_i, the so-called quasi-Monte Carlo method [24, 50, 60, 66, 67].

In many statistical applications we do not need to know how the integral ϕ at all accurately; in the Monte Carlo test a significance level of 5% is somewhere between, perhaps, 1% and 10%. Thus, in many cases, simulation is a good choice for approximate integration. Remember that a simulation experiment *is* an experiment and there are a number of techniques [23, 60] to design and analyse it to obtain maximum precision for the computer time spent.

Markov Chain Monte Carlo

The simulation-based methods are of course only easy if we have a simple way to simulate from the assumed model. In highly-structured situations we can find that everything depends on everything else. This was first encountered in statistical physics [46] and spatial statistics [30, 59]. Those authors devised iterative algorithms that only used the conditional distributions of small groups of random variables given the rest. As successive samples are not independent but form a Markov chain (on a very large state space) these methods are known as MCMC, short for Markov Chain Monte Carlo.

The same ideas were taken up for hierarchical Bayesian models by Gelfand and Smith [28, 29] some years later, under the name of Gibbs sampling (precisely the algorithm published in 1977 [59] and 1984 [30], and in the Bayesian context by Pearl [51] in 1987) and the earlier work and many of its lessons have been overlooked. In particular, although the algorithms are guaranteed to converge under mild conditions, that convergence can be far too slow to be useful without clever design, and possibly even then; see [65].

Recent reviews of MCMC from several viewpoints are given by [9], [33], [35] and [75].

Simulated annealing

Simulation can also be used to do optimization! Almost optimization methods can only find a local optimum of a function f on a domain $\mathcal{X}$, and can have great difficulty in combinatorial optimization problems (in which some of the components of $\mathcal{X}$ are categories rather than numbers).

Suppose we want to minimize f, and we know $f \geq 0$. Consider drawing samples X_i from a distribution over $\mathcal{X}$ with density proportional to $\exp[-f(x)/T]$ for $T > 0$. As we increase T the distribution becomes more and more concentrated on the region in which $f(x)$ is near its global minimum. Suppose we run a MCMC simulation and decrease T; we might hope that the sample at time N converges to a global minimum. This idea was suggested by Pincus [55, 56] but rediscovered and used by Kirkpatrick et al. [41] under the name of *simulated annealing*. The idea that the sample might converge proved to be too optimistic in practice (convergence occurs at rate $1/\log N$). Aarts and Korst [1] discuss the theory, and Cantoni [15] provides some useful results on how to vary T if a bounded amount of computing is to be used.

Despite the lack of a convincing theory, simulated annealing has been widely used in large-scale optimization problems. Perhaps the most familiar application in statistics is to the segmentation of noisy images [8, 30, 61].

There are a number of other ways to use simulation to approximate maximum likelihood estimates using MCMC, for example in spatial statistics by Pentinnen [53] and Ripley [61] and in genetics by Geyer [33, 34]).

Relaxing Linearity

One major use of computer-intensive methods has taken place largely outside statistics, in a drive to relax linearity and use much more complex models. In 1993 Ripley [62] wrote about neural networks

> Their pervasiveness means that they can not be ignored. In one way their success is a warning to statisticians who have worked in a simply-structured linear world for too long.

The extent of nonlinearity which is not just being contemplated but is being built into consumer hardware goes far beyond nonlinear regression in the sense of, for example, [5].

The most influential developments have been in decision trees [13, 57, 64], Bayesian networks [2, 27, 36, 37, 49, 52, 64, 68] and neural networks [10, 62–64]. The emphasis in the first and last is on good prediction, and the complexity of the models which are built is normally limited only by the amount of data to hand, even when this is 10^4 or more examples.

Selecting and Combining Models

Another way that a large amount of computer power has been harnessed is to consider a very large number of models. We should distinguish two uses of statistical models, for *explanation* and for *prediction*. Although statistical training has usually emphasized explanation, that the purpose of the analysis is to discover structure in the the dataset to hand, in many other fields the emphasis is entirely on prediction. Even in a field such as medical diagnosis in which it is helpful to be able to explain the basis of the diagnosis (and this is the aim of work in Bayesian networks) it would be useful to have an accurate diagnosis system without explanation. And for a voice recognition system, to read vehicle licence plates, for biometric security checking and so on all that is required is accuracy and perhaps speed.

This suggests that for prediction we should either choose a model on the basis of its predictions, or even choose to combine the predictions from several models. The traditional way to proceed was to divide the data into training, validation and test sets. The training set is used to optimize the parameters in each of the candidate models, the validation set to choose the best one (or combination) and the test set to estimate the performance of the selected prediction method (necessary as the performance on the validation set will be an optimistic measure of future performance)

Rarely will we have enough data to afford separate training, validation and test sets, and in many problems increasing the size of the training set will result in a significant gain in performance. Enter K-fold *cross-validation*, in which the dataset is divided into K sections of roughly equal size, and each section in turn is used as the validation set for the candidate models trained on the rest of the data. When all K fits are done, we have used the whole dataset as a validation set, and can choose the best model (or combination). To assess the future performance we also have to take into account the variability of the selection procedure, so we nest the K-fold cross-validation inside a V-fold split into training set and training/validation set. Typically K and V will be chosen of the order of 10, so each model is fitted of the order of 100 times, and it is not uncommon to entertain 25 models. The procedure rapidly becomes computer-intensive, but only by repeating a simple building block (fit a model on one dataset, predict on another) many times. Such procedures can easily be done in parallel, or on separate machines.

The idea of using a weighted average $\sum_i w_i f_i(x)$ of the predictions for all the candidate models goes back at least to Stone [71], and arises from Bayesian theory if we believe that one of the models is true, but do not know which one. In the Bayesian context the weights are the posterior probabilities of the models, whose computation which usually involves a very high-dimensional integration over the parameters (see, for example, [54] and [58]). Alternatively, we can use K-fold cross-validation to choose the weights w_i from the performance on the validation sets. Some examples of the use of model averaging in the statistical literature are [21], [31], [32], [44], [48] and [70], but there are many in other fields.

There is no reason why the weights we give to the various models should not vary (slowly) with x, which leads to the idea of regarding the predictions of the candidate models together with x as inputs to some nonlinear function g, so the prediction system becomes

$$g(x, f_i(x), \ldots, f_M(x); \theta)$$

and choosing θ by simultaneous fitting or cross-validation. (In the neural networks literature variants are known as *stacked generalization* [77] and *hierarchical mixtures of experts* [40].)

Many of the nonlinear procedures such as decision tree induction are rather sensitive to the training set used, and many methods have multiple optima and so are sensitive to the starting values. We can apply the ideas of combining *models* to combining the predictions of the same model fitted on different training sets. Breiman [11, 12] called one procedure *bagging*, which averages the predictions of a model fitted to bootstrapped training sets. We can go further, and design subsets of the training data to produce rather different fits to average, a process known as *boosting* [22].

This is a very active area of research, and some of the results of combining simple models have shown very appreciable gains in performance.

Programming Environments

A major attraction of computer-intensive methods is their conceptual simplicity, but most biostatisticians are no keener on computer programming than on deriving asymptotic approximations. The S language (see Becker et al. [6] and Chambers and Hastie [16]) is the preferred working environment of many of the researchers in the field. S is currently marketed exclusively as part of the commercial S-PLUS system (Mathsoft Inc., [45]). Despite a steep initial learning curve, sophisticated analyses can be performed in S-PLUS very easily [76] and many researchers donate their S software to public archives, notably `statlib` [47]. The free XLispStat system (see Tierney [74] and also Cook and Weisberg [19], available from `statlib` [47]) is also popular and has a substantial archive of user-contributed software at UCLA [20]. The package XGobi [14, 17, 18, 72, 73] implements dynamic graphics and projection pursuit, and BUGS [69] is a (currently free) system for using Gibbs sampling for a class of Bayesian methods.

REFERENCES

[1] Aarts, E. and Korst, J. (1989). *Simulated Annealing and Boltzmann Machines*. Wiley, New York.

[2] Almond, R. G. (1995). *Graphical Belief Modeling*. Chapman & Hall, London.

[3] Asimov, D. (1985). The grand tour: A tool for viewing multidimensional data. *SIAM Journal on Scientific and Statistical Computing* **6**, 128–143.

[4] Barnard, G. (1963). Contribution to the discussion of Bartlett's paper. *Journal of the Royal Statistical Society series B* **25**, 294.

[5] Bates, D. M. and Watts, D. G. (1988). *Nonlinear Regression Analysis and its Applications*. Wiley, New York.

[6] Becker, R. A., Chambers, J. M., and Wilks, A. R. (1988). *The NEW S Language*. Chapman and Hall, New York.

[7] Becker, R. A., Cleveland, W. S., and Shyu, M.-J. (1996). The visual design and control of Trellis display. *Journal of Computational and Graphical Statistics* **5**, 123–155.

[8] Besag, J. (1986). The statistical analysis of dirty pictures (with discussion). *Journal of the Royal Statistical Society series B* **48**, 259–302.

[9] Besag, J., Green, P., Higdon, D., and Mengersen, K. (1995). Bayesian computation and stochastic systems (with discussion). *Statistical Science* **10**, 3–66.

[10] Bishop, C. M. (1995). *Neural Networks for Pattern Recognition*. Clarendon Press, Oxford.

[11] Breiman, L. (1996). Bagging predictors. *Machine Learning* **24**, 123–140.

[12] Breiman, L. (1996). The heuristics of instability in model selection. *Annals of Statistics* **24**, 2350–2383.

[13] Breiman, L., Friedman, J. H., Olshen, R. A., and Stone, C. J. (1984). *Classification and Regression Trees*. Wadsworth and Brooks/Cole, Monterey, CA.

[14] Buja, A., Cook, D., and Swayne, D. F. (1996). Interactive high-dimensional data visualization. *Journal of Computational and Graphical Statistics* **5**, 78–99.

[15] Cantoni, O. (1992). Rough large deviation estimates for simulated annealing. Application to exponential schedules. *Annals of Probability* **20**, 1109–1146.

[16] Chambers, J. M. and Hastie, T. J. (eds.) (1992). *Statistical Models in S*. Wadsworth and Brooks/Cole, Pacific Grove, CA.

[17] Cook, D., Buja, A., and Cabrera, J. (1993). Projection pursuit indices based on orthonormal function expansions. *Journal of Computational and Graphical Statistics* **2**, 225–250.

[18] Cook, D., Buja, A., Cabrera, J., and Hurley, C. (1995). Grand tour and projection pursuit. *Journal of Computational and Graphical Statistics* **4**, 155–172.

[19] Cook, R. D. and Weisberg, S. (1994). *An Introduction to Regression Graphics*. Wiley.

[20] de Leeuw, J. Archive of `XLispStat` software. Use WWW or anonymous ftp to `www.stat.ucla.edu`.

[21] Draper, D. (1995). Assessment and propagation of model uncertainty (with discussion). *Journal of the Royal Statistical Society series B* **57**, 45–97.

[22] Drucker, H., Cortes, C., Jaeckel, L. D., LeCun, Y., and Vapnik, V. (1994). Boosting and other ensemble methods. *Neural Computation* **6**, 1289–1301.

[23] Evans, M. and Swartz, T. (1995). Methods for approximating integrals in statistics with special emphasis on Bayesian integration problems. *Statistical Science* **10**, 254–272.

[24] Fang, K.-T. and Wang, Y. (1994). *Number-theoretic Methods in Statistics*. Chapman & Hall, London.

[25] Friedman, J. H. (1987). Exploratory projection pursuit. *Journal of the American Statistical Association* **82**, 249–266.

[26] Friedman, J. H. and Tukey, J. W. (1974). A projection pursuit algorithm for exploratory data analysis. *IEEE Transactions on Computers* **23**, 881–890.

[27] Fung, R. and Del Favarro, B. (1995). Applying Bayesian networks to information retrival. *Communications of the ACM* **38**, 42–48, 57.

[28] Gelfand, A. E. and Smith, A. F. M. (1990). Sampling-based approaches to calculating marginal densities. *Journal of the American Statistical Association* **85**, 398–409.

[29] Gelfand, A. E., Hills, S. E., Racine-Poon, A., and Smith, A. F. M. (1990). Illustration of Bayesian inference in normal data models using Gibbs sampling. *Journal of the American Statistical Association* **85**, 972–985.

[30] Geman, S. and Geman, D. (1984). Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **6**, 721–741.

[31] George, E. I. and McCulloch, R. E. (1993). Variable selection via Gibbs sampling. *Journal of the American Statistical Association* **88**, 881–889.

[32] George, E. I. and McCulloch, R. E. (1996). Stochastic search variable selection. In Gilks et al. [35], pp. 203–214.

[33] Geyer, C. (1992). Practical Markov chain Monte Carlo (with discussion). *Statistical Science* **7**, 473–511.

[34] Geyer, C. J. (1996). Estimation and optimization of functions. In Gilks et al. [35], pp. 241–258.

[35] Gilks, W. R., Richardson, S., and Spiegelhalter, D. J. (eds.) (1996). *Markov Chain Monte Carlo in Practice*. Chapman & Hall, London.

[36] Heckerman, D. and Wellman, M. P. (1995). Bayesian networks. *Communications of the ACM* **38**, 26–30.

[37] Heckerman, D., Breese, J. S., and Rommelse, K. (1995). Decision-theoretic troubleshooting. *Communications of the ACM* **38**, 49–57.

[38] Huber, P. J. (1985). Projection pursuit (with discussion). *Annals of Statistics* **13**, 435–525.

[39] Jones, M. C. and Sibson, R. (1987). What is projection pursuit (with discussion)? *Journal of the Royal Statistical Society series A* **150**, 1–36.

[40] Jordan, M. I. and Jacobs, R. A. (1994). Hierarchical mixtures of experts and the EM algorithm. *Neural Computation* **6**, 181–214.

[41] Kirkpatrick, S., Gellat, C. D., Jr, and Vecchi, M. P. (1983). Optimization by simulated annealing. *Science* **220**, 671–680.

[42] Kruskal, J. B. (1969). Toward a practical method which helps uncover the structure of a set of multivariate observations by finding the linear transformation which optimizes a new 'index of condensation'. In *Statistical Computation*, R. C. Milton and J. A. Nelder, eds. New York. Academic Press, pp. 427–440.

[43] Kruskal, J. B. (1972). Linear transformation of multivariate data to reveal clustering. In *Multidimensional Scaling: Theory and Application in the Behavioural Sciences*, R. N. Shephard, A. K. Romney, and S. K. Nerlove, eds. New York. Seminar Press, pp. 179–191.

[44] Madigan, D. and Raftery, A. E. (1994). Model selection and accounting for model uncertainty in graphical models using Occam's window. *Journal of the American Statistical Association* **89**, 1535–1546.

[45] MathSoft Inc. S-PLUS. Data Analysis Products Division, MathSoft Inc., Seattle.

[46] Metropolis, N., Rosenbluth, A., Rosenbluth, M., Teller, A., and Teller, E. (1953). Equations of state calculations by fast computing machines. *Journal of Chemical Physics* **21**, 1087–1091.

[47] Meyer, M. J. `statlib`. on-line archive of data and computer software. Use WWW, anonymous ftp or Gopher to `lib.stat.cmu.edu`.

[48] Moulton, B. R. (1991). A Bayesian-approach to regression selection and estimation with application to a price-index for radio services. *Journal of Econometrics* **49**, 169–193.

[49] Neapolitan, E. (1990). *Probabilistic Reasoning in Expert Systems. Theory and Algorithms*. Wiley, New York.

[50] Niederreiter, H. (1992). *Random Number Generation and Quasi-Monte Carlo Methods*. SIAM, Philadelphia, PA.

[51] Pearl, J. (1987). Evidential reasoning using stochastic simulation of causal models. *Artificial Intelligence* **32**, 245–257.

[52] Pearl, J. (1988). *Probabilistic Inference in Intelligent Systems. Networks of Plausible Inference*. Morgan Kaufmann, San Mateo, CA.

[53] Pentinnen, A. (1984). *Modelling Interaction in Spatial Point Patterns: Parameter Estimation in the Maximum Likelihood Method*. Jyväskylä Studies in Computer Science, Economics and Statistics **7**.

[54] Philips, D. B. and Smith, A. F. M. (1996). Bayesian model comparison via jump diffusions. In Gilks et al. [35], pp. 215–239.

[55] Pincus, M. (1968). A closed form solution of certain programming problems. *Operations Research* **16**, 690–694.

[56] Pincus, M. (1970). A Monte Carlo method for the approximate solution of certain types of constrained optimization problems. *Operations Research* **18**, 1225–1228.

[57] Quinlan, J. R. (1993). *C4.5: Programs for Machine Learning*. Morgan Kaufmann, San Mateo, CA.

[58] Raftery, A. E. (1996). Hypothesis testing and model selection. In Gilks et al. [35], pp. 163–187.

[59] Ripley, B. D. (1977). Modelling spatial patterns (with discussion). *Journal of the Royal Statistical Society series B* **39**, 172–212.

[60] Ripley, B. D. (1987). *Stochastic Simulation*. Wiley, New York.

[61] Ripley, B. D. (1988). *Statistical Inference for Spatial Processes*. Cambridge University Press, Cambridge.

[62] Ripley, B. D. (1993). Statistical aspects of neural networks. In *Networks and Chaos—Statistical and Probabilistic Aspects*, O. E. Barndorff-Nielsen, J. L. Jensen, and W. S. Kendall, eds. London. Chapman & Hall, pp. 40–123.

[63] Ripley, B. D. (1994). Neural networks and related methods for classification (with discussion). *Journal of the Royal Statistical Society series B* **56**, 409–456.

[64] Ripley, B. D. (1996). *Pattern Recognition and Neural Networks*. Cambridge University Press, Cambridge.

[65] Ripley, B. D. and Kirkland, M. D. (1990). Iterative simulation methods. *Journal of Computational and Applied Mathematics* **31**, 165–172.

[66] Shaw, J. E. H. (1988). A quasirandom approach to integration in Bayesian statistics. *Annals of Statistics* **16**, 895–914.

[67] Spanier, J. and Maize, E. H. (1994). Quasi-random methods for estimating integrals using relatively small samples. *SIAM Review* **36**, 18–44.

[68] Spiegelhalter, D. J., Dawid, A. P., Lauritzen, S. L., and Cowell, R. G. (1993). Bayesian analysis in expert systems (with discussion). *Statistical Science* **8**, 219–283.

[69] Spiegelhalter, D. J., Thomas, A., Best, N. G., and Gilks, W. R. BUGS. Bayesian inference Using Gibbs Sampling. Version 0.5. MRC Biostatistics Unit, Cambridge, UK. Available from URL `http://www.mrc-bsu.cam.ac.uk` or by anonymous ftp from `ftp.mrc-bsu.cam.ac.uk`.

[70] Stewart, L. (1987). Hierarchical Bayesian analysis using Monte Carlo integration: computing posterior distributions when there are many possible models. *The Statistician* **36**, 211–219.

[71] Stone, M. (1974). Cross-validatory choice and assessment of statistical predictions (with discussion). *Journal of the Royal Statistical Society series B* **36**, 111–147.

[72] Swayne, D. F., Cook, D., and Buja, A. XGobi. Use WWW, anonymous ftp or Gopher to `lib.stat.cmu.edu`.

[73] Swayne, D. F., Cook, D., and Buja, A. (1991). XGobi: interactive dynamic graphics in the X window system with a link to S. In *Proceedings of the ASA Section on Statistical Graphics*. Alexandria, VA. American Statistical Association, pp. 1–8.

[74] Tierney, L. (1990). *LISP-STAT*. Wiley, New York.

[75] Tierney, L. (1994). Markov chains for exploring posterior distributions (with discussion). *Annals of Statistics* **22**, 1701–1762.

[76] Venables, W. N. and Ripley, B. D. (1997). *Modern Applied Statistics with S-Plus*. Second edition. Springer, New York.

[77] Wolpert, D. H. (1992). Stacked generalization. *Neural Networks* **5**, 241–259.

B.D. RIPLEY