data set and to identify the most influential indicators and (b) compare profiled performance of the considered units to reveal what is driving the composite indicator results, and in particular whether the composite indicator is overly dominated by a small number of indicators.

- **Links to other indicators** identify the relationships between the composite indicator (or its dimensions) and other individual or composite indicators.
- **Visualization of results** should attract audience, presenting composite indicators in a clear and accurate way.

Following the above-mentioned guidelines, the constructors of composite indicators should never forget *that composite indicators should never be seen as a goal per se. They should be seen, instead, as a starting point for initiating discussion and attracting public interest and concern* (Nardo et al. 2005).

However, there is now general agreement about the usefulness of composite indicators: There is a strong belief among the constructors of composite indicators that such summary measures are meaningful and that they can capture the main characteristic of the investigated phenomena. On the other side, there is a scepticism among the critics of this approach, who believe that there is no need to go beyond an appropriate set of individual indicators. Their criticism is focused on the "arbitrary nature of the weighting process" (Sharpe 2004) in construction of the composite indicators.

Cross References

►Aggregation Schemes
►Imputation
►Multiple Imputation
►Multivariate Data Analysis: An Overview
►Multivariate Statistical Analysis
►Scales of Measurement
►Sensitivity Analysis

References and Further Reading

An information server on composite indicators and ranking systems (methods, case studies, events) http://composite-indicators.jrc.ec.europa.eu/FAQ.htm#List_of_Composite_Indicators

Freudenberg M (2003) Composite indicators of country performance: a critical assessment, OECD science, technology and industry working papers, OECD Publishing, 2003/16

OECD, European Commission, Joint Research Centre (2008) Handbook on constructing composite indicators: methodology and user guide. OECD Publishing

OECD, Glossary of statistical terms (http://stats.oecd.org/glossary/index.htm)

OECD (2003) Composite indicators of country performance: a critical assessment, DST/IND(2003)5, Paris

Munda G, Nardo M (2005) Constructing consistent composite indicators: the issue of weights, EUR 21834 EN. Joint Research Centre, Ispra

Nardo M, Saisana M, Saltelli A, Tarantola S (2005) Tools for composite indicators building. european commission, EUR 21682 EN. Joint Research Centre, Ispra, Italy

Saltelli A (2007) Composite indicators between analysis and advocacy. Soc Indic Res 81:65–77

Sharpe A (2004) Literature review of frameworks for macro-indicators. Centre for the Study of Living Standards, Ottawa, Canada

Computational Statistics

Colin Rose
Director
Theoretical Research Institute, Sydney, NSW, Australia

What Is Computational Statistics?

We define *computational statistics* to be: ... 'statistical methods/results that are enabled by using computational methods'. Having set forth a definition, it should be stressed, first, that names such as *computational statistics* and *statistical computing* are essentially semantic constructs that do not have any absolute or rigourous structure within the profession; second, that there are any number of competing definitions on offer. Some are unsatisfactory because they focus purely on data or graphical methods and exclude symbolic/exact methods; others are unsatisfactory because they place undue emphasis on 'computationally-intensive methods' or brute force, almost as if to exclude well-written efficient and elegant algorithms that might be computationally quite simple. Sometimes, the difficulty is not in the execution of an algorithm, but in writing the algorithm itself.

Computational statistics can enable one:

- To work with arbitrary functional forms/distributions, rather than being restricted to traditional known textbook distributions.
- To simulate distributional properties of estimators and test statistics, even if closed-form solutions do not exist (*computational inference* rather than *asymptotic inference*).
- To compare statistical methods under different alternatives.
- To solve problems numerically, even if closed-form solutions are not possible or cannot be derived.
- To derive symbolic solutions to probability, moments, and distributional problems that may never have been solved before, and to do so essentially in real-time.

- To explore multiple different models, rather than just one model.
- To explore potentially good or bad ideas via simulation in just a few seconds.
- To choose methods that are theoretically appropriate, rather than because they are mathematically tractable.
- To check symbolic/exact solutions using numerical methods.
- To bring to life theoretical models that previously were too complicated to evaluate . . .

Journals and Societies

Important journals in the field include:

- Combinatorics, Probability & Computing
- Communications in Statistics – Simulation and Computation
- Computational Statistics
- Computational Statistics and Data Analysis
- Journal of Computational and Graphical Statistics
- Journal of the Japanese Society of Computational Statistics
- Journal of Statistical Computation and Simulation
- Journal of Statistical Software
- SIAM Journal on Scientific Computing
- Statistics and Computing

Societies include: the International Association for Statistical Computing (IASC – a subsection of the ISI), the American Statistical Association (Statistical Computing Section), the Royal Statistical Society (Statistical Computing Section), and the Japanese Society of Computational Statistics (JSCS) . . .

Computational statistics consists of three main areas, namely numerical, graphical and symbolic methods . . .

Numerical Methods

The numerical approach is discussed in texts such as Gentle (2009), Givens and Hoeting (2005), and Martinez and Martinez (2007); for Bayesian methods, see Bolstad (2009). Numerical methods include: Monte Carlo studies to explore asymptotic properties or finite sample properties, pseudo-random number generation and sampling, parametric density estimation, non-parametric density estimation, ▶bootstrap methods, statistical approaches to software errors, information retrieval, statistics of databases, high-dimensional data, temporal and spatial modeling, ▶data mining, model mining, statistical learning, computational learning theory and optimisation etc. . . . While optimisation itself is an absolutely essential tool in the field, it is very much a field in its own right.

Graphical Methods

Graphical methods are primarily concerned with either (a) viewing theoretical models and/or (b) viewing data/fitted models.

In the case of *theoretical* models, one typically seeks to provide understanding by viewing one, two or three variables, or indeed even four dimensions (using 3-dimensional plots animated over time, translucent graphics etc.).

Visualizing *data* is essential to data analysis and assessing fit; see, for instance, Chen et al. (2008). Special interest topics include smoothing techniques, kernel density estimation, multi-dimensional data visualization, clustering, exploratory data analysis, and a huge range of special statistical plot types. Modern computing power makes handling and interacting with large data sets with millions of values feasible . . . including live interactive manipulations.

Symbolic/Exact Methods

The 21st century has brought with it a conceptually entirely new methodology: symbolic/exact methods. Recent texts applying the symbolic framework include Andrews and Stafford (2000), Rose and Smith (2002), and Drew et al. (2008).

Traditional 20th century computer packages are based on numerical methods that tend to be designed much like a cookbook. That is, they consist of hundreds or even thousands of numerical recipes designed for specific cases. One function is written for one aspect of the Normal distribution, another for the LogNormal, etc. This works very well provided one stays within the constraints of the known common distributions, but unfortunately, it breaks down as soon as one moves outside the catered framework. It might work perfectly for random variable X, but not for X^2, nor $\exp(X)$, nor mixtures, nor truncations, nor reflections, nor folding, nor censoring, nor products, nor sums, nor . . .

By contrast, symbolic/exact methods are built on top of computer algebra systems . . . programs such as *Mathematica* and Maple that understand algebra and mathematics. Accordingly, symbolic algorithms can provide exact general solutions . . . not just for specific distributions/models. Symbolic computational statistical packages include mathStatica (2002–2010, based on top of *Mathematica*) and APPL (based on top of Maple).

Symbolic methods include: automated expectations for arbitrary distributions, probability, combinatorial probability, transformations of random variables, products of random variables, sums and differences of random variables, generating functions, inversion theorems, maxima/minima of random variables, symbolic and numerical maximum likelihood estimation (using exact methods),

curve fitting (using exact methods), non-parametric kernel density estimation (for arbitrary kernels), moment conversion formulae, component-mix and parameter-mix distributions, copulae, pseudo-random number generation for arbitrary distributions, decision theory, asymptotic expansions, ▶order statistics (for identical and non-identical parents), unbiased estimators (h-statistics, k-statistics, polykays), moments of moments, etc.

The Changing Notion of What is Computational Statistics

Just 10 or 20 years ago, it was quite common for people working in computational statistics to write up their own code for almost everything they did. For example, the *Handbook of Statistics 9: Computational Statistics* (see Rao 1993) starts out Chapter 1 by describing algorithms for sorting data. Today, of course, one would expect to find sorting functionality built into any software package one uses . . . indeed even into a word processor. And, of course, the 'bar' keeps on moving and evolving. Even in recent texts such as Gentle (2009), about half of the text (almost all of Part 1) is devoted to computing techniques such as fixed- and floating-point, numerical quadrature, numerical linear algebra, solving non-linear equations, optimisation etc., . . . techniques that Gentle et al. (2004, p. 5) call "*statistical computing*" but which are really just *computing*. Such methods lie firmly within the domain of computational science and/or computational mathematics . . . they are now built into any decent modern statistical/mathematical software package . . . they take years of work to develop into a decent modern product, and they require tens of thousands of lines of code to be done properly . . . all of which means that it is extremely unlikely that any individual would write their own in today's world. Today, one does not tend to build an airplane simply in order to take a flight. And yet many current texts are still firmly based in the older world of 'roll your own', devoting substantial space to routines that are (a) general mathematical tools such as numerical optimisation and (b) which are now standard functionality in modern packages used for computational statistics. While it is, of course, valuable to understand how such methods work (in particular so that one is aware of their limitations), and while such tools are absolutely imperative to carrying out the discipline of computational statistics (indeed, as a computer itself is) – these tools are now general mathematical tools and the days of building one's own are essentially long gone.

Future Directions

It is both interesting and tempting to suggest likely future directions.

(a) *Software packages*: At the present time, the computational statistics software market is catered for from two polar extremes. On the one hand, there are major general mathematical/computational languages such as *Mathematica* and Maple which provide best of breed general computational/numerical/graphical tools, and hundreds of high-level functional programming constructs to expand on same, but they are less than comprehensive in field-specific functionality. It seems likely such packages will further evolve by developing and growing tentacles into specific fields (such as statistics, combinatorics, finance, econometrics, biometrics etc.). At the other extreme, there exist narrow field-specific packages such as S-Plus, Gauss and R which provide considerable depth in field-specific functionality; in order to grow, these packages will likely need to broaden out to develop more general methods/general mathematical functions, up to the standard offered by the major packages. The software industry is nascent and evolving, and it will be interesting to see if the long-run equilibrium allows for both extremes to co-exist. Perhaps, all that is required is for a critical number of users to be reached in order for each eco-system to become self-sustaining.

(b) *Methodology*: It seems likely that the field will see a continuing shift or growth from *statistical inference* to *structural inference*, . . . from *data mining* to *model mining*, . . . from *asymptotic inference* to *computational inference*.

(c) *Parallel computing*: Multicore processors have already become mainstream, while, at the same time, the growth in CPU speeds appears to be stagnating. It seems likely then that parallel computing will become vastly more important in evolving computational statistics into the future. Future computational statistical software may also take advantage of GPUs (graphical processing units), though it should be cautioned that the latter are constrained in serious statistical work by the extremely poor numerical precision of current GPUs.

(d) *Symbolic methods*: Symbolic methods are still somewhat in their infancy and show great promise as knowledge engines i.e., algorithms that can derive exact theoretical results for arbitrary random variables.

(e) *On knowledge and proof*: Symbolic algorithms can derive solutions to problems that have never been posed before – they place enormous technological power into the hands of end-users. Of course, it is possible (though rare) that an error may occur (say in integration, or by mis-entering a model). In a sense,

this is no different to traditional reference texts and journal papers which are also not infallible, and which are often surprisingly peppered with typographical or other errors.

In this regard, the computational approach offers both greater exposure to danger, as well as the tools to avoid it. The "danger" is that it has become extremely easy to generate output in real-time. The sheer scale and volume of calculation has magnified, so that the average user is more likely to encounter an error, just as someone who drives a lot is more likely to encounter an accident. *Proving* that the computer's output is actually correct can be very tricky, or impractical, or indeed impossible for the average practitioner to do, just as the very same practitioner will tend to accept a journal result at face value, without properly checking it, even if they could do so. The philosopher, Karl Popper, argued that the aim of science should not be to prove things, but to seek to refute them. Indeed, the advantage of the computational statistical approach is that it is often possible to check one's work using two completely different methods: both numerical and symbolic. Here, numerical methods take on a new role of checking symbolic results. One can throw in some numbers in place of symbolic parameters, and one can then check if the solution obtained using symbolic methods (the exact theoretical solution) matches the solution obtained using numerical methods (typically, ►numerical integration or Monte Carlo methods, see ►Monte Carlo Methods in Statistics). If the numerical and symbolic solutions do *not* match, there is an obvious problem and we can generally immediately reject the theoretical solution (*a la* Popper). On the other hand, if the two approaches match up, we still do not have a proof of correctness . . . all we have is just one point of agreement in parameter space. We can repeat and repeat and repeat the exercise with different parameter values . . . and as we do so, we effectively build up, not an absolute proof in the traditional sense, but, appropriately for the statistics profession, an ever increasing degree of confidence . . . effectively a proof by probabilistic induction . . . that the theoretical solution is indeed correct. This is an extremely valuable (though time-consuming) skill to develop, not only when working with computers, but equally with textbooks and journal papers.

About the Author

For biography *see* the entry ►Bivariate distributions.

Cross References

►Bootstrap Asymptotics
►Bootstrap Methods
►Data Mining
►Monte Carlo Methods in Statistics
►Nonparametric Density Estimation
►Non-Uniform Random Variate Generations
►Numerical Integration
►Numerical Methods for Stochastic Differential Equations
►R Language
►Statistical Software: An Overview
►Uniform Random Number Generators

References and Further Reading

Andrews DF, Stafford JEH (2000) Symbolic computation for statistical inference. Oxford University Press, New York
Bolstad WM (2009) Understanding computational Bayesian statistics. Wiley, USA
Chen C, Härdle W, Unwin A (2008) Handbook of data visualization. Springer, Berlin
Drew JH, Evans DL, Glen AG, Leemis LM (2008) Computational probability. Springer, New York
Gentle JE (2009) Computational statistics. Springer, New York
Gentle JE, Härdle W, Mori Y (eds) (2004) Handbook of computational statistics: concepts and methods. Springer, Berlin
Givens GH, Hoeting JA (2005) Computational statistics. Wiley, New Jersey
Martinez WL, Martinez AR (2007) Computational statistics handbook with MATLAB, 2nd edn. Chapman & Hall, New York
mathStatica (2002–2010), www.mathStatica.com
Rao CR (1993) Handbook of statistics 9: computational statistics. Elsevier, Amsterdam
Rose C, Smith MD (2002) Mathematical statistics with Mathematica. Springer, New York

Conditional Expectation and Probability

Takis Konstantopoulos
Professor
Heriot-Watt University, Edinburgh, UK

In its most elementary form, the conditional probability $P(A|B)$ of an event A given an event B is defined by

$$P(A|B) = \frac{B(A \cap B)}{P(B)},$$