Workshop on Experimental Analysis of Algorithms: Interfaces between the Statistical and Computational Sciences

NISS Building, Research Triangle Park, NC
March 6-7, 2008

ABSTRACTS

Jon Bentley, Avaya Labs Research

Title: Little Experiments on Algorithms

Each of the dozen algorithmic experiments in this talk can be described in a few minutes, and each was conducted in under an hour. Problem domains include computational geometry, bin packing and system utilities such as sorting and memory allocation; code sizes range from dozen-line testbed programs to multiple-staff-decade commercial systems. Analytic techniques are limited to inspection of a few data points and simple graphs of carefully chosen variables. Even so, tiny experiments effectively characterize the behavior of complex systems, such as sorting algorithms in the presence of caching. Particular emphasis is placed on drawing robust conclusions, making fair comparisons, identifying the components of running time, and making plausible extrapolations from experiment to asymptopia.

Marios Hadjieleftheriou, AT&T Labs - Research

Title: One-Pass Streaming Algorithms: Theory and Practice

For this talk I will concentrate on four practical applications related with network monitoring: estimating the distinct number of flows, identifying flows with heavy traffic, estimating frequencies and norms. I will select a small set of randomized and deterministic algorithms for solving these problems in small space and in one pass over the data; all algorithms will provide strong theoretical guarantees in terms of space, accuracy, and failure rates. I will run experiments using real network traffic from a core AT&T router, and compare experimental results under different data distributions with theoretical expectations.

I will show that: a. a given algorithm does not give meaningful results under all data distributions (we need to be careful where and when we apply these algorithm), b. a given algorithm is better at solving certain problems and bad at solving others (irrespective of what theory suggests), c. the theoretical bounds are, sometimes, irrelevant in practice (either the hidden constants make an algorithm impractical, or the algorithm has excellent performance when the space bound is not met).

William Hart, Sandia National Laboratories

Title: Experimenting with Meta-Heuristics

Meta-heuristic search algorithms rarely have provable performance properties, and thus performance characterization of these algorithms is an inherently experimental process. This talk will highlight three areas where statistical issues need to be addressed for meta-heuristics. First, efficient performance assessment and comparison of meta-heuristics remains an outstanding issue. Virtually all performance comparisons consider the distribution of solution quality after a given time-limit. By contrast, what users typically want to know is the effort require to find a near-optimal solution. Analysis of such statistical questions requires the use of statistical techniques related to survival analysis, which have not been used in this community. Second, meta-heuristics typically have many parameters that may be tuned to improve the efficiency of these algorithm (e.g. on a particular application). A variety of experimental design techniques have been proposed, but these require complex experiments that can be difficult to design. Finally, a related issue is the design of meta-heuristics, since these frameworks often have many optional search mechanisms. These design problems require the tuning of nested parameters, which further complicates the experimental design and analysis process.

David Johnson, AT&T Labs - Research

Title: Comparability

In the "Experimental Analysis of Algorithms" we perform experiments to learn about the behavior of algorithms. This seems like an obvious goal, but note that the experiments we perform involve a lot more than the algorithms themselves. What we typically test is a system consisting of an implementation of the algorithm in some programming language, compiled using some compiler to run on a particular operating machine on a particular computer, using particular test instances. If all we are interested in is the quality of solutions the algorithm produces, this extra superstructure can usually be ignored, but often what we are interested in is how fast the algorithm is, in comparison to other algorithms. How can we compare the results of two different studies? Often we can remove one variable by using the same test instances, but we cannot always use the same machine. In particular, if the earlier study to which I wish to compare is 10 years old, the same machine may no longer be available and may be hopelessly slower than what is available today. How can we cope with these issues? How can we make our new studies useful to researchers 10 years from now?

In this talk I will illustrate how this challenge is handled (or ignored) with examples from the literature on the Traveling Salesman Problem (TSP), and talk about our rudimentary attempts to cope with it in the DIMACS TSP Challenge.

Catherine McGeoch, Amherst College

Title: Finding Asymptopia in the Data Set

The most important question asked about an algorithm is: what is its asymptotic cost? The related data analysis question is: given a data sample from an unknown random function on n, with unknown expectation f(n), find an asymptotic upper bound on the order of magnitude of the leading term of f(n). For example, if f(n) = 3n - 2n + 7nlg(n), I would want to learn from the analysis that the growth rate of f(n) is no more than quadratic.

This is the first thing I want to know about any algorithm I am experimenting on, but there appears to be no good data analysis technique for answering this question. There are plenty of techniques for fitting curves, to data, and assessing the quality of the fit, but this is turns out to be a very different problem. Indeed, many textbooks carry specific warnings about the dangers of "extrapolating" from a fitted curve.

Despite the hazards, I want to do it anyway. I have tried to adapt some curve-fitting techniques to the asymptotic curve bounding problem, with mixed results. This talk will illustrate the ideas I have tried, and how well (or poorly) they worked on data sets from algorithmic experiments.