A Perspective on Statistical Tools for Data Mining Applications

David M. Rocke
Center for Image Processing and Integrated Computing
University of California, Davis

Statistics and Data Mining

Statistics is about the analysis of data. Some statistical ideas are designed for problems in which well-formulated prior hypotheses are evaluated by the collection and analysis of data, but other currents of thought in the field are aimed at more exploratory ends. In this sense, data mining (defined as the exploratory analysis of large data sets) should be a branch of statistics. Yet the field of data mining has evolved almost independently of the community of statistical researchers. Why is this?

One reason for the separate development of data mining is that the methods were developed by those who needed to solve the problems, and these rarely included researchers whose primary areas of interest were statistical theory and methodology. Several authors have recently pointed out ways in which statistical ideas are relevant to data mining (see Elder and Pregibon 1996; Friedman 1997; Glymour et al. 1996, 1997; as well as many contributors to the NRC CATS report 1996). If the only reason for the lack of use of standard statistical and data analytic methods were unfamiliarity, this approach would be sufficient. But perhaps there are other problems.

Statistical methods have traditionally been developed to make the greatest use of relatively sparse data. The concept of statistical efficiency, for example, is crucial when data are expensive, and less important when additional data can be obtained for the price of fetching them from the disk. On the other hand, computational efficiency has always played a smaller role, especially since the advent of electronic computers. When analyzing potentially vast data sets, the importance of various considerations is changed or reversed compared to what statisticians are traditionally used to. It is perhaps this that has made the use of standard statistical methods less common in the field of data mining than it might otherwise have been.

What is Difficult about Data Mining?

The problem is not just that there is a large amount of data, or that the goal is exploratory. Statisticians (among other disciplines) have developed many tools for the exploration of data. For many exploratory statistical problems, the answers become clearer as the size of the data set become larger. Finding means and medians, as well as regression coefficients, fall into this easy category. Suppose, for example, that one wished to predict account delinquincies from a fixed set of 25 predictors that exist in a database, and suppose that previous experience showed that a linear logistic regression specification worked well. In this case, the only uncertainty lies in the values of the regression coefficients. If a particular coefficient were estimated to be 2.0 with a standard error of 1.3 from a sample of 100 cases, there would be considerable uncertainty as to its value. If the sample size was increased to 1,000,000 cases, the standard error of the coefficient would be only .0013, or essentially no uncertainty. Furthermore, the calculations for this larger regression would not be very expensive in the context of normal data mining applications.

But many problems in data mining have what might be called *subtle structure*, defined as those structures that are difficult to find even in large samples. Multivariate outlier detection and cluster finding fall into this category. This means that considerable searching is required to determine how the data break out into clusters, and more data do not necessarily make this easier. If the cluster structure is hidden from easy detection because of the orientation of the clusters, no amount of data will make this immediately apparent.

Subtle structure is often definable as the global optimum of a function with many local optima. The best global optimum may be hard to find with small data sets, but good computational procedures with small data sets may be impractical with large ones. Cluster finding can be cast in this mold if one defines clusters, for example, by means of the normal likelihood or a similar criterion function. (One example would be to compute the pooled covariance matrix of all clusters together, in which each point is centered at its cluster mean). Such criterion functions may have many local optima, and finding the correct one is often a difficult problem.

Computational Complexity of Statistical Methods for Data Mining Applications

We consider this issue in the context of two fundamentally important methods in data mining: data cleaning and data segmentation. Data cleaning here will mean the identification of anomalous data (outliers for short) that may need to be removed or separately addressed in an analysis. Variable-by-variable data cleaning is straightforward, but often anomalies only appear when many attributes of a data point are simultaneously considered. Multivariate data cleaning is more difficult, but is an essential step in a complete analysis. We will avoid technical details of these methods (see Rocke and Woodruff 1996), but the essential nature of the methods is to identify the main "shape" of the data and identify as outliers those data points that lie too far from the main body of data.

Data segmentation is here taken to mean the division of the data into nonoverlapping subsets that are relatively similar within a subset. Some points may be outliers, and so belong to no group. We do not assume that any reliable prior knowledge exists as to which points belong together, or even how many groups there may be. Statistical methods for this problem often go by the name "cluster analysis." In both cases of data cleaning and data segmentation, consideration will be restricted to measurement data, rather than categorical data. This will simplify the discussion so that many separate sub-cases do not need to be separately described.

It seems like an obvious point that the computational effort cannot rise too fast with the size of the dataset, otherwise processing large datasets would be impossible. If there are n data points, each with p associated measurements, then many statistical methods naively used have a computational complexity of $O(np^3)$, while more complex methods may in principle be high order polynomial, or even exponential in n or p. This is obviously not satisfactory for larger data sets.

In the quest for low computational effort, one limit is that there must of necessity be a piece that is linear in *n*. This is because we cannot identify outliers unless each point is examined. It will be important that the linear part has a low constant; that is, the effort will rise proportional to *n*, but the constant of proportionality must be small. This will be the case if we simply plug each point into a quick outlier identification routine. In no case, however, should we tolerate calculations that rise more than linearly with *n*.

In data segmentation, it might seem difficult to avoid a piece that is proportional to n^2, since pairwise distances are often needed for classification. We avoid this difficulty by employing sampling (in the first of two ways). If *n* is very large, the basic structure of a data set can be estimated using a much smaller number of points than *n*, and that basic structure can be used to classify the remaining points. For example, if a complex algorithm for data segmentation is $O(n^3)$, but we instead perform the calculation on a subset of size proportional to $n^{1/4}$ (for large *n*), the net complexity is a sublinear $O(n^{3/4})$.

An additional advantage of estimating the structure of the data set on a subset of the data is that it can be done more than once and the results compared. This allows for a kind of independent verification of results and avoids making conclusions based on accidental appearances. If each subcalculation is sublinear, then repeating it any fixed number of times that does not rise with *n* is also sublinear.

Sampling has an additional role to play in estimating complex structures within a sample. Some estimation methods have a computational effort that rises exponentially with *n* if done naively. An example of this is a method of data cleaning that depends on finding the most compact half of the data and then evaluating each point against this half. Specifically, the Minimum Covariance Determinant (MCD) method finds that half of the data for which the determinant of the covariance matrix of the data is smallest. Since the space that must be searched to find even a gross approximation of the MCD rises exponentially with *n*, there is a danger that the computational effort to find a solution within given quality bounds will also rise rapidly (Woodruff and Rocke 1993). A solution (Woodruff and Rocke 1994) is to conduct most computations only on the cells of a partition of the data, and then use the full sample only for the final stages.

Properties of Algorithms for Data Mining

Computer-science theorists usually strive to find methods that are deterministic, get the right answer every time, and run in worst-case polynomial time. The problems that we face cannot apparently be accomplished without giving up something from this list—most likely, we will need to give up both determinism and sure correctness. We clearly cannot use methods that fail to be polynomial; in fact a low-constant linear portion plus a sublinear remainder is required. Probabilistic algorithms provide a way to obtain good, if not provably optimal, answers for even very large problems.

Global Optimization

Many problems have both a discrete and a continuous formulation. For example, if the data are thought to fall into two clusters, we can define this by the (discrete) cluster membership (an integer vector of length *n*) or by the (continuous) location and shape of the clusters (two continuous vectors and two continuous matrices). Often continuous formulations lead to better answers if the “correct” local optimum is found, but this may be difficult to find. Discrete methods can yield search techniques with no obvious or easy continuous analog, thus improving the search for the local optimum. These include swap neighborhoods, constructive neighborhoods with highly random starting point (small number of points), tabu lists, etc. Often it is very effective to use a discrete heuristic search method to provide one or more starting points for locating optima of a continuous global optimization problem. Among heuristic optimization methods, genetic algorithms (GA) seem to work well for optimization only when a descent step is added (so that the GA serves mainly to diversify the starting points). Simulated annealing can perform well if tuned properly, but we have had more success with other methods. Steepest descent with random restarts (perhaps with constructive start) is hard to beat for many problems.

The Role of Algorithms in Statistical Science

At one time, it would have been feasible to argue that algorithms were not central to statistical science because all a better algorithm did was somewhat speed up arriving at essentially the same estimate. Things have changed. More and more, we are using estimators that have a substantial stochastic component, in which the "theoretical estimator" is not achievable, and all we get are various approximations of varying degrees of exactness. To avoid extra complexity, we will describe this phenomenon in terms of an example—*S*-estimators of multivariate location and scale—but it clearly applies equally to Markov chain Monte Carlo, optimal experimental design, and other important areas of statistical science.

An *S*-estimator of multivariate location and scale consists of that choice of a location vector ***T*** and PDS matrix ***C*** which minimizes |***C***| subject to a data-based constraint (omitted here) which is designed to standardize the estimator. The *S*-estimator can always be shown to exist in some theoretical sense, but there exists no known algorithm that can always produce it, even in an arbitrarily long computation. The theoretical *S*-estimator must be a solution to a set of associated *M*-estimating equations; however, there is no known method of determining how many such local solutions there are. Given a list of such local solutions, the best of them is a candidate for the theoretical *S*-estimator, but there is no known method of determining whether there is another solution with smaller determinant. Furthermore, there can be very large differences between the best solution (the theoretical *S*-estimator), and the second-best solution (Rocke and Woodruff 1993, 1997).

Under these circumstances, which occur in more and more modern statistical methods, the algorithm used to find local solutions is critical in the quality of the approximation to the theoretical *S*-estimator. Use of a less effective algorithm can result, not in a small degradation of performance, but in a very large one (Woodruff and Rocke 1994; Rocke and Woodruff, 1996). In a very real sense, "the algorithm is the estimator," and different algorithms result in essentially different estimators, which can have very different finite-sample properties.

Furthermore, if the only available algorithm for an estimator has exponential computational complexity, estimation in large or even medium sized samples is impossible. For example, the exact computation of a robust multivariate estimator, the Minimum Covariance Determinant Estimator (MCD) is only possible in samples of size 30 or less. No conceivable improvement in computational speed could raise this level higher than 100. Consider the case of a sample of size 200 in which we are interested in examining all subsamples of size 100. There are about 10^{59} such subsamples. Suppose that one had a million-processor parallel computer, each processor of which could examine a billion configurations per second (this is perhaps 1000 Gflops, considerably faster than any existing processor). It would still take 10^{33} *millenia* for the process to finish!

For massive data sets, even polynomial-time algorithms may be far too slow. An $O(n^3)$ algorithm may be feasible in samples of size 1000, but not in samples of size 1,000,000. Linear, or even sub-linear, algorithms are required to deal with massive data sets.

Conclusion

A statistical perspective on data mining can yield important benefits if the data mining perspective on statistical methods is kept in mind during their development. In the data mining perspective, statistical efficiency is not a particularly important goal, whereas computational efficiency is critical. Statistical methods must be used that do not depend on prior knowledge of the exact structure of the data; the questions to be asked as well as the answers to be derived must be free to depend on the outcome of the analysis.

Some specific recommendations include:

- Perspectives from statistics, computer science, and machine learning must contribute to an understanding of algorithms for data mining.
- Global optimization is often necessary to solve data mining problems, which often have subtle structure.
- Attention to the performance of heuristic search algorithms is critical.
- Sampling is needed to keep complexity linear or sublinear. Partitioning to find a starting point by combinatorial search for a smooth algorithm is often a viable strategy.
- Statistical asymptotics must now focus on computational complexity as well as statistical convergence.
- Although data mining has developed mostly independently of statistics as a discipline, a fusion of the ideas from these two fields will lead to better methods for the analysis of massive data sets.

References

Banfield, J. D. and Raftery, A. E. (1993), “Model-Based Gaussian and Non-Gaussian Clustering,” *Biometrics*, **49**, 803–821.

Cabena, Peter, Hadjinian, Pablo, Stadler, Rolf, Verhees, Jaap, Zanasi, Alessandro (1998) *Discovering Data Mining: From Concept to Implementation*, Upper Saddle River, NJ: Prentice Hall.

Elder IV, John and Pregibon, Daryl (1996) “A Statistical Perspective on Knowledge Discovery in Databases,” in *Advances in Knowledge Discovery and Data Mining*, Usama Fayyad, Gregory Piatetsky-Shapiro, Padhraic Smyth, and Ramasamy Uthurusamy (eds.) Cambridge, MA: MIT Press

Fayyad, Usama M., Piatetsky-Shapiro, Gregory, Smyth, Padhraic, and Uthurusamy, Ramasamy (eds.) (1996) *Advances in Knowledge Discovery and Data Mining*, Cambridge, MA: MIT Press.

Friedman, Jerome H. (1997) “On Bias, Variance, 0/1-Loss, and the Curse-of-Dimensionality,” *Data Mining and Knowledge Discovery*, **1**, 55–78.

Glymour, Clark, Madigan, David, Pregibon, Daryl, and Smyth, Padhraic (1996) “Statistical Inference and Data Mining” *Communications of the ACM*, **39**, 35–41.

Glymour, Clark, Madigan, David, Pregibon, Daryl, and Smyth, Padhraic (1997) “Statistical Themes and Lessons for Data Mining,” *Data Mining and Knowledge Discovery*, **1**, 1–28.

National Research Council, Board on Mathematical Sciences, Committee on Applied and Theoretical Statistics (1996) *Massive Data Sets: Proceedings of a Workshop*, Washington, DC: National Academy Press.

Rocke, D. M. (1996) “Robustness Properties of *S*-Estimators of Multivariate Location and Shape in High Dimension,” *Annals of Statistics*, **24**, 1327–1345.

Rocke, D. M. and Woodruff, D. L. (1993) “Computation of Robust Estimates of Multivariate Location and Shape,” *Statistica Neerlandica*, **47**, 27–42.

Rocke, D. M. and Woodruff, D. L. (1996) “Identification of Outliers in Multivariate Data,” *Journal of the American Statistical Association*, **91**, 1047–1061.

Rocke, D. M. and Woodruff, D. L. (1997) “Robust Estimation of Multivariate Location and Shape,” *Journal of Statistical Planning and Inference*, **57**, 245–255.

Woodruff, D. L., and Rocke D. M. (1993) “Heuristic Search Algorithms for the Minimum Volume Ellipsoid,” *Journal of Computational and Graphical Statistics*, **2**, 69–95.

Woodruff, D. L. and Rocke, D. M. (1994) “Computable Robust Estimation of Multivariate Location and Shape in High Dimension using Compound Estimators,” *Journal of the American Statistical Association*, **89**, 888–896.