From Large to Huge: A Statistician's Reactions to KDD & DM[1]

Peter J. Huber

University of Bayreuth
D-94440 Bayreuth, Germany
peter.huber@uni-bayreuth.de

Abstract

The three distinct data handling cultures (statistics, data base management and artificial intelligence) finally show signs of convergence. Whether you name their common area "data analysis" or "knowledge discovery", the necessary ingredients for success with ever larger data sets are identical: good data, subject area expertise, access to technical know-how in all three cultures, and a good portion of common sense. Curiously, all three cultures have been trying to avoid common sense and hide its lack behind a smoke-screen of technical formalism. Huge data sets usually are not just more of the same, they have to be huge because they are heterogeneous, with more internal structure, such that smaller sets would not do. As a consequence, subsamples and techniques based on them, like the bootstrap, may no longer make sense. The complexity of the data regularly forces the data analyst to fashion simple, but problem- and data-specific tools from basic building blocks, taken from data base management and numerical mathematics. Scaling-up of algorithms is problematic, computational complexity of many procedures explodes with increasing data size; for example, conventional clustering algorithms become unfeasible. The human ability to inspect a data set, or even only a meaningful of part it, breaks down far below terabyte sizes. I believe that attempts to circumvent this by "automating" some aspects of exploratory analysis are futile. The available success stories suggest that the real function of data mining and KDD is not machine discovery of interesting structures by itself, but targeted extraction and reduction of data to a size and format suitable for human inspection. By necessity, such preprocessing is *ad hoc*, data specific and driven by working hypotheses based on subject matter expertise and on trial and error. Statistical common sense – which traps to avoid, handling of random and systematic errors, and where to stop – is more important than specific techniques. The machine assistance we need to step from large to huge sets thus is an integrated computing environment that allows easy improvisation and retooling even with massive data.

Introduction

Knowledge Discovery in Databases (KDD) and Data Analysis (DA) share a common goal, namely to extract meaning from data. The only discernible difference is that the former commonly is regarded as machine centered, the latter as centered on statistical techniques and probability. But there are signs of convergence towards a common, human-centered view on both sides.

Note the comment by Brachman and Anand (1996, p.38): "Overall, then, we see a clear need for more emphasis on a human-centered, process-oriented analysis of KDD". One is curiously reminded of Tukey's (1962) plea, emphasizing the role of human judgment over that of mathematical proof in DA. It seems that in different periods each professional group has been trying to squeeze out human common sense and to hide its lack behind a smoke screen of its own technical formalism. The statistics community appears to be further along on the way, a majority now has acquiesced to the idea that DA ought to be a human-centered process, and I hope the AI community will follow suit towards a happy reunion of resources.

About Data

Data can be experimental (from a designed experiment), observational (with little or no control over the process generating the data), or opportunistic (the data have been collected for an unrelated purpose; such data are sometimes called "samples of convenience"). Massive data sets rarely belong to the first category, since by a clever design the data flow often can be reduced already before it is recorded. But they often belong to the third category for plain reasons of economy.

On the whole, the data mining community, mostly coming from data base management and logic backgrounds, does not yet seem to be sensitized to the specific problems arising with

statistical data, where relationships hold only on average, and where that average can be distorted by selection bias or similar effects.

Retrospective analyses of opportunistic data sets are beset with particularly ugly statistical pitfalls. Not all statisticians are aware of them, and hardly any non-statisticians. Few textbooks even mention the problem; a notable exception is Freedman et al. (1991), p. 506ff. Standard errors and tests of significance usually do not make sense with a sample of convenience, so watch out. In other words: if you try to assess the accuracy and reliability of a fitted model by the usual statistical techniques, you may fool yourself and others. As a data analyst, you need a healthy dose of statistical common sense to recognize the problem, to assess its severity, and to avoid being fooled by it. It also takes stamina (and professional honesty), if you have to tell your sponsor that certain questions cannot be answered with the data at hand, since he can easily find somebody else who will answer them anyway.

Sometimes, data sets are massive because their collection is mandated by law (e.g. census and certain health data), or because they are collected anyway for other purposes (e.g. financial data). Often, however, they have to be massive because smaller sets will not do, and the predominant reason why they will not do is that the data in question are highly structured and heterogeneous for intrinsic reasons. In most cases, structural complexity has to do with the fact that there are many objects, observed by several observers, and the observations are located in space and time. Often, complexity is accompanied by inhomogeneity: standards for measurements, the set of observed variables, and protocols for their recording may change over space and time, between observers and between objects. In addition, there may be unnoticed local, and sometimes even global, lapses of quality control.

As a rule, the data analyst will be confronted not with the data, but primarily with a task (or tasks), hopefully to be solved with the data. Usually, those tasks are, at least initially, poorly formulated and understood.

The following illustrative examples have been slightly stylized, but are based on actual consulting experiences. The first one exemplifies pure structural complexity.

Example 1: Air traffic radar data. A typical situation is: half a dozen radar stations observe several hundred aircraft, producing a 64-byte record per radar per aircraft per antenna turn, approximately a megabyte of data per minute. If one is to investigate a near collision, one extracts a subset, defined by a window in space and time surrounding the critical event. If one is to investigate reliability and accuracy of radars under real-life air traffic conditions, one must differentiate between gross errors, systematic errors, and random measurement errors. Outlier detection and interpretation is highly non-trivial. Essentially, one must first connect thousands of dots to individual flight paths (technically, this amounts to sometimes tricky prediction and identification problems). The remaining dots are outliers, which then must be sorted out and identified according to their likely causes (a swarm of birds, a misrecorded azimuth measurement, etc. etc.). In order to assess the measurement accuracy with regard to both systematic and random errors, one must compare measurements of individual radars to flight paths determined from all radars, interpolated for that particular moment of time. The error structure is quite inhomogeneous, depending on the range and topographic features, as well as on crowding conditions. Summary statistics do not enter at all, except at the very end, when the results, say the frequency and pattern of different types of errors, are summarized for presentation. This example also illustrates the role of ancillary data sets: radar installation data, locations of airports, air traffic routes, national and traffic control boundaries, etc.

The next example illustrates problems of inhomogeneity.

Example 2: Highway maintenance. The available data are maintenance records and measurements of road quality in the Federal Republic and Germany, spanning several decades. The ultimate task is to provide a rational basis (a decision support system) for highway maintenance policies. In particular one ought to be able to predict long-range consequences of maintenance decisions. For that, one should find the relationships between road construction, repairs, traffic volume and surface deterioration. Because of the time frame involved, and because of distributed administrative responsibilities, such a data collection cannot possibly be homogeneous. Both the maintenance records and the measurements of road quality are uneven. Most of the data are crude. The same terms may mean different things to different people at different times. Some test routes have been measured in great detail, but only for a limited time. The statistical fluctuations are enormous. Unsuspected gaps in the data (e.g. missing records of maintenance interventions) are far from obvious and may escape even a painstaking scrutiny. It is easier to decide which statistical techniques to use (say between regression and survival analysis) than to decide whether the available data is good enough or whether it needs to be supplemented by additional, targeted measurement programs.

There may be indirect cross-linking between originally unrelated data sets, as in the following example. Typically, such cross-links are not explicit in the data sets and may be difficult to establish.

Example 3: Health and environmental hazards. The task is to investigate long-range health effects of pesticides. The available data are retrospective and consist of several, originally unrelated, opportunistic sets.

- Patient data: hospital, diagnosis, treatment, ...
- Pesticide application data: what? when? where? how much? ...
- Cross-linking: who was potentially exposed to what? when? ...

Summaries and random subsamples may be worse than useless, as illustrated by the following example.

Example 4: Radon levels. It was found (through exploratory data analysis of a large environmental data set) that very high radon levels were tightly localized and occurred in houses sitting on the locations of old mine shafts. Neither the existence nor the location of such shafts was deducible from the data set, they were found by on-site inquiries. In this case, indiscriminate grouping and averaging would have hidden the problem and would have made it impossible to investigate causes and necessary remedies. Random samples would have been useless, too: either they would have missed the exceptional values altogether, or one would have thrown them out as outliers. A traditional statistician, looking for a central tendency, a measure of variability, measures of pairwise association between a number of variables, or the like, would have missed the essential issue.

Naive data mining, that is: grinding a pile of data in a semi-automatic, untutored fashion through a black box, almost inevitably will run into the GIGO-syndrome – Garbage In, Garbage Out. Unfortunately, you may not recognize the output as such. The more opaque a black "data mining" box is, the less likely it is that one will recognize potential problems. A case story from a data analysis exam may serve as a warning.

Example 5: Discriminant analysis. The data exam problem was to distinguish between carriers and non-carriers of a certain genetic disease on the basis of enzyme and other data. A student found that age was the variable that discriminated best between carriers and controls. This was entirely correct, but useless. What he had discovered, but misinterpreted, was that in the process of data collection, carriers and controls had not been properly matched with regard to age. Would you have spotted the problem if the result had been presented not verbally but in the form of a duly cross-validated black box (e.g. a neural network)?

Data Size and Scaling

By now, we believe to understand the issues involved in the interactive analysis of data sets in the megabyte range, and perhaps a little beyond. Somewhere around data sizes of 100 megabytes or so, qualitatively new, very serious scaling problems begin to arise, both on the human and on the algorithmic side. In concrete terms, we thus must be concerned with stepping to gigabyte sizes and beyond.

Human Limitations.

The human ability to inspect the whole of a data set, or even only a meaningful part of it, breaks down for datasets in the gigabyte range. This is most emphatically not a problem of the display devices (as some authors seem to believe, cf. Butler and Quarrie 1996), but one of the human visual system. See Wegman 1995. The conclusion is that human inspection of terabyte sets forever will be restricted to very thin subsets or very crude summaries. As mentioned before, with highly structured sets neither random subsamples nor any of the customary summaries or density estimates will be of use.

Computational Complexity.

Scaling up of algorithms is problematic, the computational complexity of many fashionable computer intensive procedures explodes with increasing data size. Computations taking up to about 10^{15} floating point operations are just about feasible nowadays (one gigaflop per second, sustained for two weeks). If computer performance doubles every two years, as in the past, this will go up to 10^{18} in 20 years, but by then, somewhere near sustained teraflop performance, just-in-time management of massive data sets will become problematic (tera = 10^{12}, and in 10^{-12} seconds, light moves merely 0.3 mm). This means that algorithms supposed to work for gigabyte sets and above better do not take more than about $O(n^{3/2})$ operations for n items. To illustrate the practical consequences, assume that a data set containing n items is structured as a matrix with r rows and c columns, $r>c$. Then, for example, regression or principal component algorithms are feasible, they use $O(rc^2)$ operations, but clustering algorithms, using $O(r^2c)$ operations, are out. See Huber 1994, 1996, Wegman 1995.

Workarounds?

Since it is humanly impossible to inspect more than a very limited number of very thin slices of a huge data set, we must select such slices, at least initially, either on the basis of prior model considerations, or assisted by machine search. The idea to have a robot search for interesting, but otherwise unspecified features is alluring, but in my opinion (influenced by experiences with manual and automated projection pursuit) it is a mere day-dream, bound to fail for several separate reasons. First, already for moderately large data sets, blind machine searches appear to have excessive computational

complexity. Second, after we have found some interesting looking features (with structured data, there may be overly many), it is too hard for us to interpret them, unless we are guided by some goals or working hypotheses. But if we already have a goal or working hypothesis, then it makes more sense to search in a targeted fashion. In practical terms, this amounts to targeted extraction and reduction of data to a size and format suitable for human inspection. By necessity, any such pre-processing is *ad hoc*, data and task specific, and driven by working hypotheses, first based on subject matter expertise and then amended by trial and error. Both the formulation of the goals and the interpretation of the finds is a task for a human mind, not for a machine. The problem, as I see it, is not one of replacing human ingenuity by machine intelligence, but one of assisting human ingenuity by all conceivable tools of computer science and artificial intelligence, in particular aiding with the improvisation of search tools and with keeping track of the progress of an analysis. That is, the data analyst needs foremost a good, integrated computing environment.

The following example, which is of historical interest as the first big success story of data mining, may serve as an illustration of targeted pre-processing.

Example 6: Lunar Mascons. The surprising discovery of mass concentrations (Mascons) under the lunar surface is due to Muller and Sjogren (1968), who found them by extracting structure from the residual noise of a Doppler radar ranging experiment. They did this work on their own time, because their superiors at JPL (Jet Propulsion Laboratory) felt they were going to waste their efforts on garbage. I remember Muller joking that evidently one person's junk pile was another's gold mine, so the invention of data mining ought to be credited to them. Their success was due not to a black box approach, but to a combination of several thoughtful actions: First, a careful inspection of residuals, revealing that these did not behave like white noise. Second, a tentative causal interpretation: the cause might be an irregular distribution of lunar mass. Third, modelling this distribution by expanding it into spherical harmonics with unknown coefficients and then estimating these coefficients by least squares. Fourth, a graphical comparison of isodensity contours of the estimated mass distribution with a map of lunar surface features. The discovery literally happened at the moment when the plot emerged from the plotter: it revealed systematic mass concentrations, beneath the lunar *maria*. Interestingly, the persuasive argument in favor of correctness of the result was not probabilistic (i.e. a significance level, or the like), but the convincing agreement between calculated mass concentrations and visible surface features.

Data Structure and Strategy

Almost every major data analysis requires considerable initial data massage. First, it is necessary to bring data from different sources into a consistent format; this is a prerequisite for any efficient analysis. Often, particularly in the Data Warehousing literature, this initial massage is regarded mainly as an issue of data cleaning. However, cleaning often distorts the internal stochastic structure of the data, and with cleaned data it is no longer possible to investigate data quality, so watch out.

Whenever the data have a complex structure, most of the traditional statistical approaches, based on summaries and subsamples, will not work. For example, resampling methods (bootstrap) rarely are applicable. In almost every case, *ad hoc*, data and task specific pre-processing and rearranging of the data is needed (cf. in particular Examples 1 and 6). Usually, some sophisticated, but simple, statistical ideas and algorithms will be built into the pre-processing and processing of the data, combined with tools from numerical analysis and data base operations. All these should therefore be available in the form of re-usable modules or building blocks. The traditional packages, offering such tools in canned form, are much too heavy-handed and not flexible enough.

The overall issue is one of strategy, that is: the free use of the prepared means for the purposes of a campaign, adapting them to individual needs (Huber 1997, quoting Clausewitz). To facilitate such a combination of tools, they must be presented in the form of an integrated, extensible, high-level data analysis language.

Presentation of Results

The larger the data sets are, the more difficult it is to present the conclusions. With massive data sets, the sets of conclusions become massive too, and it is simply no longer possible to answer all potentially relevant questions. We found that a kind of sophisticated decision support system (DSS), that is: a customized software system to generate answers to questions of the customers, almost always is a better solution than a thick volume of precomputed tables and graphs. It is straightforward to design a system duplicating the functions of such a volume, and it is easy to go a little beyond, for example by providing hypertext features or facilities for zooming in on graphs. But the appetite grows with the eating, trickier problems will arise, and the DSS then begins to develop into a full-fledged, sophisticated, customized data analysis system adapted to the particular data set(s).

Actually, with massive data sets the need for customized data analysis systems arises already earlier in the analysis, namely whenever several people with similar needs must work with the same data, or the same kind of data, over an extended period of time. It is humanly impossible to pre-

specify a customized system in advance, one must learn by trial and error. Close cooperation and feedback between data analysts, subject area specialists and end-users are required, and whenever possible, the latter must be involved from the very beginning. Th. Huber and M. Nagel (1996) have described the methodology for preparing such systems under the name *Data Based Prototyping*. The process is driven by the data and by its on-going, evolving analysis; it is a task which must be done by people analyzing the data, and it cannot be left to mere programmers.

Conclusions

The following list of conclusions appears already in Huber (1996):

- With the analysis of massive data sets, one has to expect extensive, application- and task-specific pre-processing. We need tools for efficient *ad hoc* programming.
- It is necessary to provide a high-level data analysis language, a programming environment and facilities for data-based prototyping.
- Subset manipulation and other data base operations, in particular the linking of originally unrelated data sets, are very important. We need a data base management system with characteristics rather different from those of a traditional DBMS.
- The need for summaries arises not at the beginning, but toward the end of the analysis.
- Individual massive data sets require customized data analysis systems tailored specifically toward them, first for the analysis, and then for the presentation of results.
- Pay attention to heterogeneity in the data.
- Pay attention to computational complexity; keep it below $O(n^{3/2})$, or forget about the algorithm.
- The main software challenge: we should build a pilot data analysis system working according to the above principles on massively parallel machines.

References

Brachman, R. J., and Anand, T. 1996. The Process of Knowledge Discovery in Databases: A Human-Centered Approach. In: *Advances in Knowledge Discovery and Data Mining*. Fayyad, U. M., Piatetsky-Shapiro, G., Smyth, P., and Uthurusamy, R., (eds.). M.I.T. Press.

Butler, J. N., and Quarrie, D. R. 1996. Data acquisition and analysis in extremely high data rate experiments. *Physics Today*, Oct. 1996, 50-56.

Freedman, D.; Pisani, R.; Purves, R., and Adhikari, A. 1991. *Statistics*. 2nd edition. New York: Norton & Company.

Huber, P. J. 1994. Huge Data Sets. In: *Proceedings of the 1994 COMPSTAT Meeting*. Dutter, R., and Grossmann, W. (eds.). Physica-Verlag, Heidelberg.

Huber, P. J. 1996. Massive Data Sets Workshop: The Morning After. In: *Massive Data Sets. Proceedings of a Workshop*. Kettenring, J., and Pregibon, D. (eds.). National Academy Press, Washington, D.C.

Huber, P. J. 1997. Strategy Issues in Data Analysis. In: *Proceedings of the Conference on Statistical Science honoring the bicentennial of Stefano Franscini's birth, Monte Verità, Switzerland*. Malaguerra, C.; Morgenthaler, S., and Ronchetti, E. (eds.). Basel, Birkhäuser Verlag.

Huber, Th. M., and Nagel, M. 1996. Data Based Prototyping. In: *Robust Statistics, Data Analysis, and Computer Intensive Methods*. Rieder, H. (ed.), Springer, New York.

Muller, P. M., and Sjogren, W. L. 1968. Mascons: Lunar Mass Concentrations. *Science*, 161, 680-684.

Tukey, J. W. 1962. The future of data analysis. *Ann. Math. Statist.* 33, 1-67.

Wegman, E. 1995. Huge data sets and the frontiers of computational feasibility. *J. of Computational and Graphical Statistics*, 4, 281-295.