Saving All the Bits

Peter J. Denning

I often hear from colleagues in the earth sciences, astronomy, physics and other disciplines that after we start up an expensive instrument or complete a massive computation, we must save all the bits of information generated by the instrument or the computation. The arguments for this practice are, first, that the cost of acquiring the bits is so great that we cannot afford to lose any of them, and, second, that some rare event might be recorded in those bits, and to throw them away would be a great loss for science. Sometimes these points are made with such vehemence that I am left with the impression that saving the bits is not merely a question of cost but is a moral imperative.

Those who accept the imperative of saving the bits are perforce limited to questions about technologies for storing and moving bits. How can we build a communications network with enough bandwidth to carry all the bits? How can we build storage devices to hold them? How can we build retrieval mechanisms that will provide access to them from around the world? Given a determination to save every bit, data compression is worth considering only if it is lossless, or in other words if it is a reversible mapping from the original data to the compressed data. "Smart instruments" that detect patterns in the data and inform us of those patterns are of little interest; indeed, there is sometimes outright hostility to such instruments. It has been claimed, for example, that on-board data processing by weather satellites delayed the discovery of the Antarctic ozone hole by several years.

When the Hubble Space Telescope is operated at full capacity, it sends some 300 million bits per second via NASA's satellite-link network to the Goddard Space Flight Center in Maryland. This data stream will be joined by that from the ACT (advanced communications technology) satellite and several other "Great Observatories." By the late 1990s, NASA will have placed in orbit a network of satellites making up the EOS (earth observing system). These are just a few of the growing number of advanced space-borne instruments, any one of which can produce a data stream of hundreds of millions of bits per second.

Let us do some simple arithmetic with the EOS data stream. This system is expected to produce between 10^{12} and 10^{13} bits per day. These are enormous numbers. If the data are stored on compact optical disks, which hold about four gigabits each, then a day's output will fill up at least 2,500 CDs. Where will all the disks be kept? Will the Goddard center be responsible for recording 2,500 disks daily? Even a national communications network with gigabit-per-second capacity will be inadequate to divert the stream to other sites for recording. And if we succeed in recording all the bits, how will we gain access to them? How will I as a scientist ask for the records that might contain evidence of a particular event? I will have to search 2,500 disks to survey one day's observations, 900,000 disks for a year's, or nine million disks to examine trends over a 10-year period.

Peter J. Denning is Director of the Research Institute for Advanced Computer Science at the NASA Ames Research Center.

Increases in optical storage density may allow the number of disks to be reduced by a factor of 10 or 100 by the time EOS is on line. On the other hand, the volume of data generated by this program and others like it may well expand by a factor of 1,000. Furthermore, these examples do not take into account the data-fusion problem that arises when an investigator attempts to study several data sources simultaneously for correlations. I have heard it said that advanced graphics will allow the investigator to visualize all the bits and see the correlations. But this statement is too glib: it ignores limitations on the bandwidth of networks, the speed of graphics devices, methods of storing and retrieving data, and algorithms for detecting correlations.

Paradigms and Practicality

The imperative to save all the bits forces us into an impossible situation: The rate and volume of information flow overwhelm our networks, storage devices and retrieval systems, as well as the human capacity for comprehension. Why then are so many scientists unwilling to forgo the practice? The answer seems to lie with the paradigm of the scientific method itself, which requires that full disclosure of an experiment and its data be made to the community to allow for independent verification of the results. To shed light on this, I take a short digression into the paradigms of science.

We often use the word paradigm to refer to the framework of preunderstandings in which we interpret the world. We have been taught, and we teach our students, that the great discoveries of science have happened when the discoverer challenged the current paradigm and stepped outside of it. At the same time, as recognized masters of our scientific domains, we resist changes that might leave us in a less-esteemed position. Thus we have a love-hate relationship with paradigms: we like challenging the paradigms of others, but we dislike having others challenge our own.

In *Science in Action*, Bruno Latour painstakingly analyzes the scientific literature before, during and after great discoveries and great inventions [1]. He distinguishes between the simplified story we tell about science when looking back after the fact, and the complex web of conversations, debates and controversies that exist before the discovery is accepted by the community. By tracing the literature, he demonstrates that statements are elevated to the status of facts only after no one has been able to mount a convincing dissent. Thus, he says, science is a process of constructing facts. Not any statement can be accepted as fact; a large community of people must accept the statement and must be incapable with the resources and methods available to them of adducing new evidence that casts doubt on the statement.

It is interesting that although we acknowledge the importance of community action while doing science, we quickly adopt a different view as soon as the science is done. Our research papers, for example, describe orderly, systematic investigations proceeding from problem description, to experiment, to data collection and analysis, and finally to conclusions. The paper tells a story that never happened: it fits neatly inside the scientific-method paradigm, whereas the discovery itself is made inside a network of ongoing conversations. We do this also with the history of science. We trace an idea back to its roots, giving the first articulator the full credit. (If the idea is great enough, we give its original articulator a Nobel prize.) The complex, dynamic web of conversations and controversies disappears.

The stories told in our research papers make it seem as if the scientific method were something as fundamental and immutable as the laws of nature. When community action is brought back into view, however, we see that in reality the scientific method is a set of standards for convincing others that a statement ought to be taken as fact. Accordingly, the method is subject to revision: communities change their standards when old standards are no longer practical.

The same reasoning applies to the problem of massive data. The standard of saving all the bits for future reference is clearly not practical in a growing number of cases. We need to step outside the paradigm and accept that there are important cases in which we do not need all the bits. New questions arise. An important one is: What machines can we build that will monitor the data stream of an instrument, or sift through a database of recordings, and propose for us a statistical summary of what's there?

Discovery Machines

Let me give an example of a "discovery machine" under test jointly by the Research Institute for Advanced Computer Science and the Artificial Intelligence Branch at the NASA Ames Research Center. Peter Cheeseman has developed a program called Autoclass that uses Bayesian inference to automatically discover the smallest set of statistically distinguishable classes of objects present in a database [2].

In 1987 Autoclass was applied to the 5,425 records of

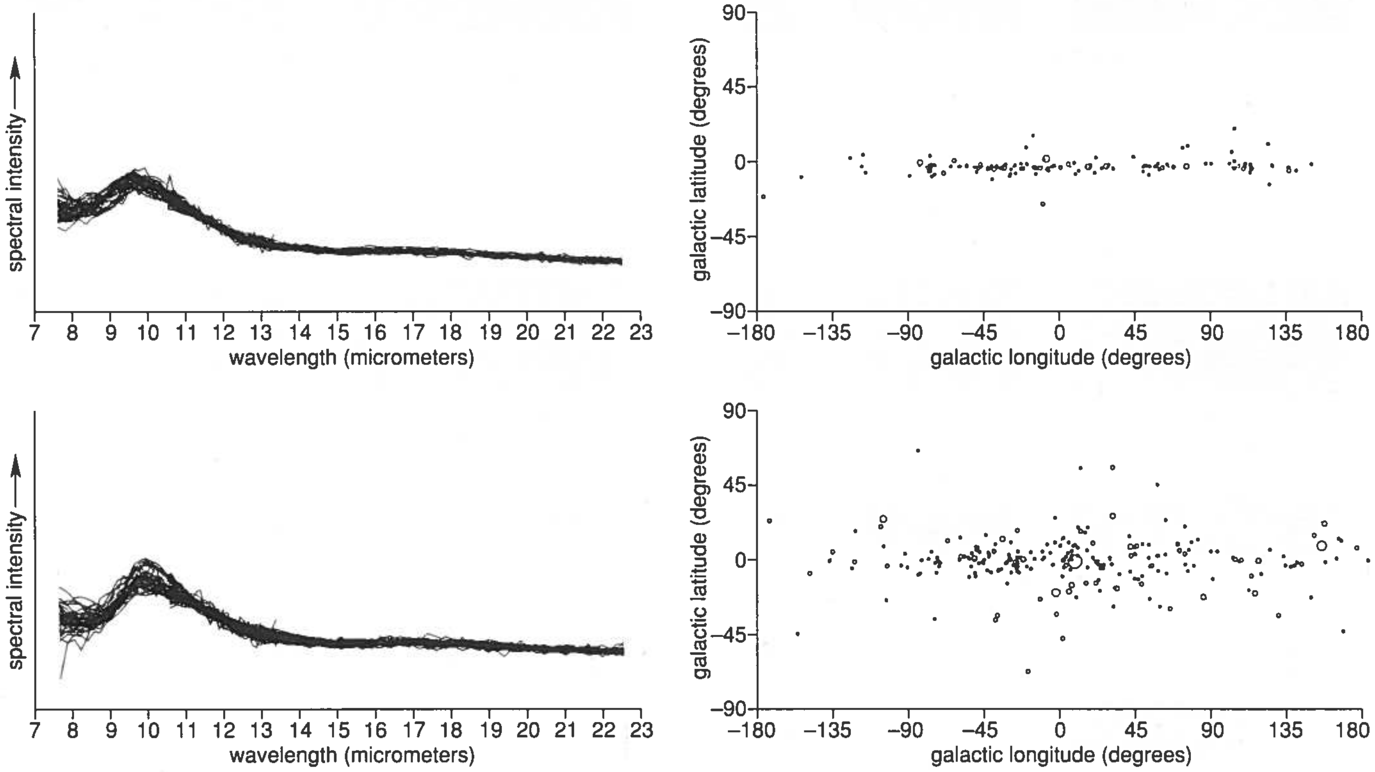


Figure 1. Automated detection of patterns in data is accomplished by the computer program Autoclass, which employs the method of Bayesian inference to group data into classes. The data in this case are infrared spectra of 5,425 stellar objects, recorded in 1983 and 1984 by the Infrared Astronomical Satellite (IRAS). Previous analysis had identified within these records a set of 297 objects with strong silicate spectra. Autoclass partitioned this set into two parts. One class *(upper left)* consists of 171 objects whose spectra have a peak at a wavelength of 9.7 micrometers. The second class *(lower left)* includes 126 objects with peak intensity at 10.0 micrometers. When the objects in each set are plotted on a star map *(right)*, the upper set shows a tendency to cluster around the galactic plane, whereas the objects in the lower set are more widely scattered, confirming that the classification represents real differences between the sets of objects. Autoclass did not use the celestial coordinates in forming the classes.

spectra observed by the Infrared Astronomical Satellite (IRAS) in 1983 and 1984. Each record included two celestial coordinates and 94 intensities at selected wavelengths in the range from 7 to 23 micrometers. Autoclass reported most of the classes previously observed by astronomers, and most of the differences between the Autoclass results and prior understanding were acknowledged by astronomers as clearly representing unknown physical phenomena. In 1989 NASA re-issued the star catalogue for the IRAS objects based on Autoclass's results.

There is nothing magic about Autoclass. It is a program that takes a large set of records of many-dimensional data and groups them into similarity classes using Bayesian inference. It is thus an instrument that allows finer resolution than is possible with the unaided human eye. It does not need to know anything about the discipline in which the data were collected; it works directly on the raw data. The important point illustrated by Autoclass is that a machine can isolate a pattern that otherwise would have escaped notice by human observers.

Cheeseman suggests that an Autoclass analyzer could be attached to an instrument, where it would monitor the data stream and form its own assay of distinguishable classes. It would transmit the class descriptions to human observers on the ground, at significant reductions in bandwidth. A human observer wanting to see all the details of specific ob-

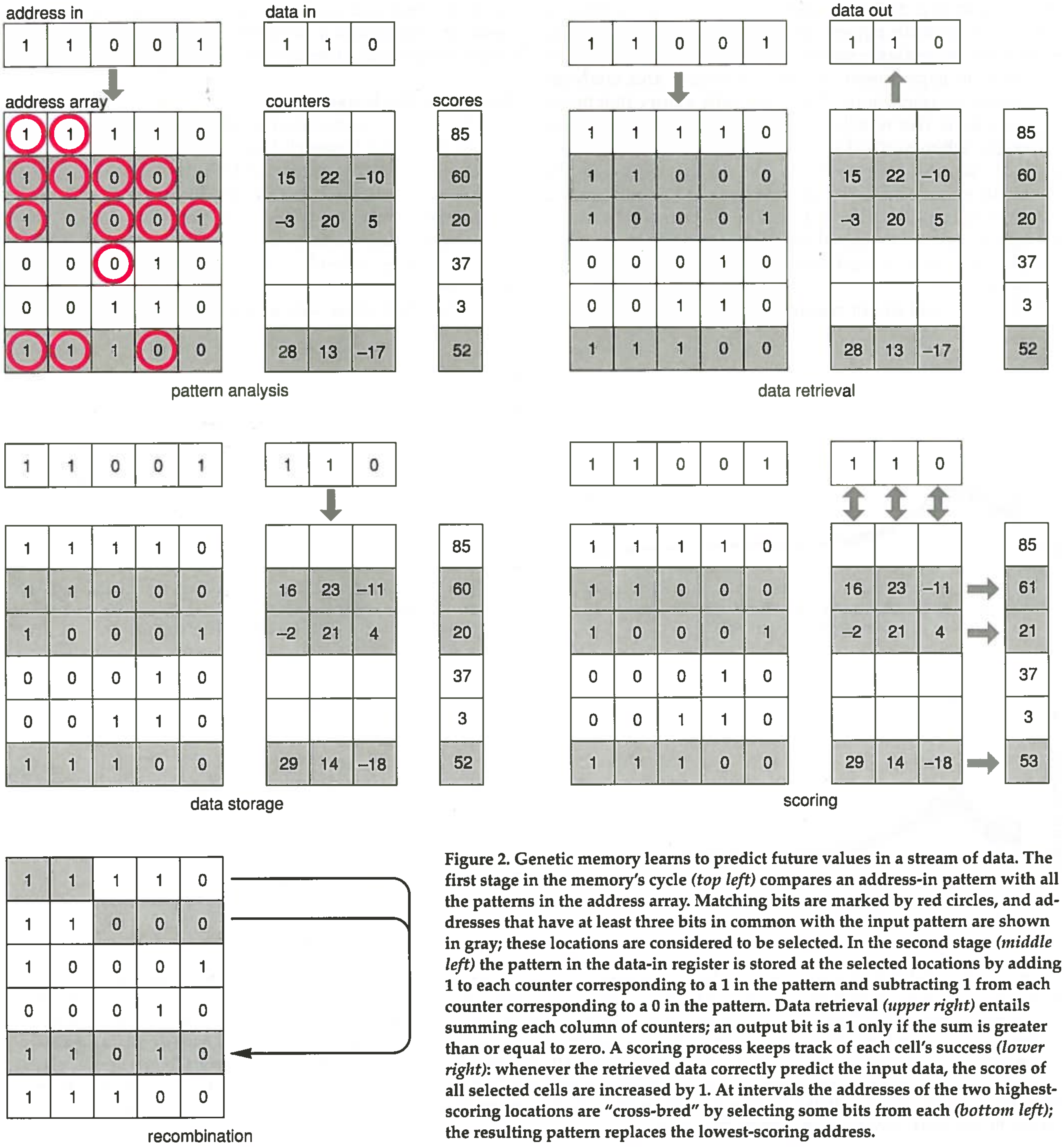


Figure 2. Genetic memory learns to predict future values in a stream of data. The first stage in the memory's cycle *(top left)* compares an address-in pattern with all the patterns in the address array. Matching bits are marked by red circles, and addresses that have at least three bits in common with the input pattern are shown in gray; these locations are considered to be selected. In the second stage *(middle left)* the pattern in the data-in register is stored at the selected locations by adding 1 to each counter corresponding to a 1 in the pattern and subtracting 1 from each counter corresponding to a 0 in the pattern. Data retrieval *(upper right)* entails summing each column of counters; an output bit is a 1 only if the sum is greater than or equal to zero. A scoring process keeps track of each cell's success *(lower right)*: whenever the retrieved data correctly predict the input data, the scores of all selected cells are increased by 1. At intervals the addresses of the two highest-scoring locations are "cross-bred" by selecting some bits from each *(bottom left)*; the resulting pattern replaces the lowest-scoring address.

jects could command the analyzer to pipe all the bits straight through.

Let me give a second example. Also at the Research Institute for Advanced Computer Science we have been studying an associative-memory architecture called SDM (sparse distributed memory) [3, 4]. In a conventional computer memory each data item is stored at a particular location, or address, and it can be retrieved only by specifying that address. An associative memory affords access by content rather than by location. In the SDM each memory cell has an address field (a vector of bits) and a data field (a vector of counters). When an address pattern is presented, decoders at all the cells simultaneously determine whether the given address and their own stored address are similar; similarity is determined by some measure such as Hamming distance, which counts the number of bits that would have to be changed to make two patterns identical. All the cells in which the stored address is an acceptable match for the supplied pattern participate in the read or write operation requested. Writing is accomplished by adding an image of the data vector to the selected counters; reading is done by statistically reconstructing a bit vector from these counters.

In one experiment David Rogers sought to learn if a variant of SDM could learn correlations between measurements and desired results. He fed an SDM simulator a stream of 58,000 records of weather data from a station in Australia. Each record included 12 measurements and a bit indicating whether rain fell in the measurement period. The measurements were encoded into a 256-bit vector, and the rain bit of the *next* period was used as data. Just before the actual next-period rain bit was stored, the SDM was asked to retrieve its version of the bit. If the retrieved bit agreed with the bit about to be written, each selected cell had 1 added to its "score." At intervals the two highest-scoring memory locations were cross-bred by combining pieces of their addresses; the new address thus created replaced the address in the lowest-scoring location. This is the principle used in genetic algorithms, and Rogers calls his variant of the SDM the genetic memory.

At the end of the experiment, Rogers found that the memory gave accurate predictions of rain. By examining the address fields of all the memory cells, he was able to determine which subset of the measurements were the most highly correlated with the occurrence of rain in the next measurement period.

The genetic memory is a machine that can be fed a stream of data. It organizes itself to become a consistent predictor of a specified pattern. It opens up new methods of discovering the predictors of a pattern in a large field of data elements.

Knowbots

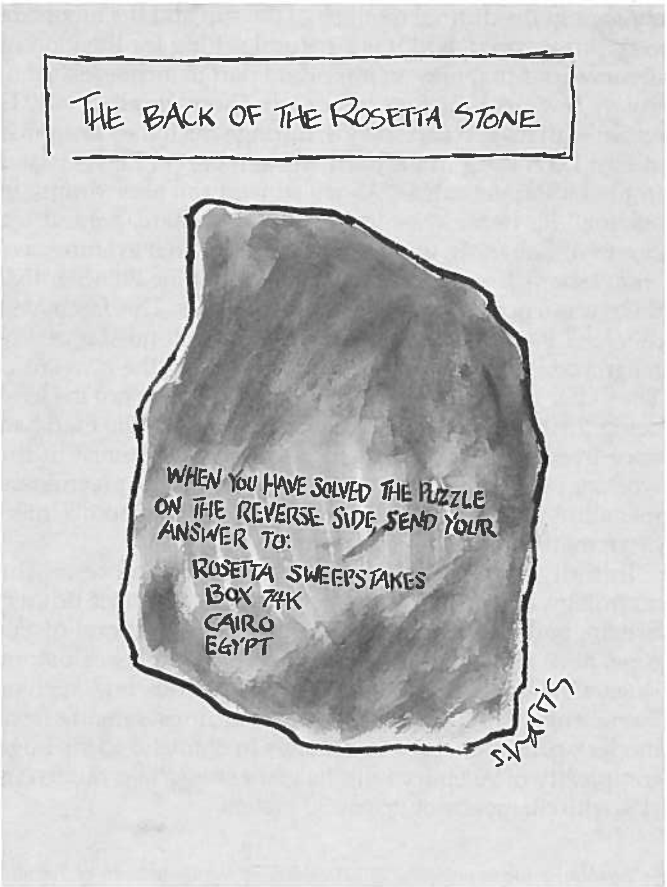
Both Autoclass and the genetic memory show that it is possible to build machines that can recognize or predict patterns in data without knowing the meaning of the patterns. Such machines may eventually be fast enough to deal with large data streams in real time. By the end of the decade they may well be advanced enough to serve on space probes and space-borne instruments, where they can monitor streams that would be incomprehensible to us directly. With these machines, we can significantly reduce the number of bits that must be saved, and we can reduce the hazard of losing latent discoveries from burial in an immense database. The same machines can also pore through existing databases looking for patterns and forming class descriptions for all the bits we've already saved.

I am not alone in this conclusion. Writing in *Science* recently, journalist M. Mitchell Waldrop documents the rising concern in the science community about the volume of data that will be generated by supercomputers and by instruments [5]. He likens the coming situation to drinking from a fire hose: Instant access to far-flung databases could soon be a reality, but how will we swallow a trillion bytes a day? He is drawn to a proposal by Robert E. Kahn and Vinton G. Cerf of the Corporation for National Research Initiatives to create surrogate processes that would canvass the networks looking for data of a particular kind, returning home with their findings. Called knowbots (short for knowledge-collecting robots), these processes would scour the networks for answers to questions.

Waldrop's article ends without saying how knowbots might work. What might go inside? Machines that perform automatic discovery, pattern matching and prediction.

References

1. Bruno Latour. 1987. *Science in Action*. Harvard University Press.
2. Peter J. Denning. 1989. "Bayesian Learning." *American Scientist* 77:216–218 (May–June).
3. Pentti Kanerva. 1988. *Sparse Distributed Memory*. MIT Press/Bradford Books.
4. Peter J. Denning. 1989. "Sparse Distributed Memory." *American Scientist* 77:333–335 (July–August).
5. M. Mitchell Waldrop. 1990. "Learning to drink from a fire hose." *Science* 248:674–675 (11 May).


THE BACK OF THE ROSETTA STONE
WHEN YOU HAVE SOLVED THE PUZZLE ON THE REVERSE SIDE, SEND YOUR ANSWER TO:
ROSETTA SWEEPSTAKES
BOX 74K
CAIRO
EGYPT