Bayes Marches On (U)

BY F. T. LEAHY

Top Secret Daunt

This article describes one of the more important mathematical techniques used by cryptanalysts today. It is based on an address given by the author before the assembled members of the Crypto-Mathematical Institute at their regular meeting in April, 1959.

There are two general mathematical methods of measurements that are used in almost all the successful cryptanalysis at the National Security Agency. The first is the χ^2, or its companions, the δ, α, or cross IC, and $\Sigma \log f!$ When it is appropriate to use any of these formulas – in other words, when we have a frequency count of different kinds of "objects"–the questions that the cryptanalyst is asking himself generally are: (1) Is the source of this data a random sampling from a flat universe? Or, less frequently, (2) Are these samples so similar to each other that they appear to be merely a larger sample randomly broken into parts? If the mathematical answer to these questions is in the general direction of being "yes" (we know that hairline distinctions between "yes" and "no" answers do not exist) the cryptanalyst would generally be disappointed, and would have to start to search elsewhere for whatever clues he needs to further the solution of his problem. Whenever the answer tends to look like "no", there is probably a reason – possibly one that can be ascertained — which caused the distribution of figures under study to have its apparent causal characteristics.

The main point we are making is that with a χ^2 type of statistic, exactly one hypothesis is advanced; namely, that no particular cause exists to make the data appear *unlike* a randomly selected sample.

A typical example of the usage of a δ I.C. would be to spot a cipher message consisting of a simple substitution of plain text, since we know that in such a cipher the frequencies of the letters encountered would reflect the widely different plaintext probabilities. Again, the δ can be used just as effectively (if the cipher message is longer by an appropriate amount) to spot a cyclic polyalphabetic substitution of plain text. (We would simultaneously discover that we had a polyalphabetic substitution of plain text, and we'd learn the exact number of alphabets.) There are of course other examples, too numerous to mention, used daily in our cryptanalysis, in which the χ^2 or the δ locates the one unusual situation, separating it from a mass of potential contenders, and thus points the way to the next (perhaps the final) step in the decipherment of the message under attack.

However, this discussion will deal with another method of measurement which is often far more powerful and useful than any of the χ^2 family of statistical tests, and which also is instrumental in playing a major part in much of the successful cryptanalysis of our agency.

The reference is to Bayes' Theorem, which in turn gives rise to what are known as Bayes factors. Who the original Mr. Bayes was, I do not know, as his name does not appear in the usual list of famous men. However, my own private researches lead me to believe that he was a brilliant 18th century Irish scholar. The latest edition of Van Nostrand's Scientific Encyclopedia, after stating Bayes' Theorem itself, petulantly remarks,

> "When all conditions of the theorem are fulfilled, there is no objection to it. The difficulties in applying the theorem depend upon the fact that the a priori probabilities are not known, and are assumed to be equal in the absence of other knowledge . . . However, the theorem has been found to be unscientific, to give rise to various inconsistencies, and to be unnecessary. The modern theory of testing hypotheses makes no use of it."

In the rest of this article we will make Mr. Van Nostrand eat these very words. For despite the statement just quoted, we believe that Bayes' Theorem is not only useful, but in fact leads to the *only* correct formulas for solving a large number of our cryptanalytic problems. Incidentally, only a handful of mathematicians at N.S.A. know about *all* the ways that Bayes factors can be employed, or how to prepare the formulas in every case.

As you know, Bayes' Theorem and Bayes factors are scarcely mentioned in many books on statistics and probability, and in fact one of my first encounters with them arose in connection with a brief Navy paper I wrote some years ago, in which I spelled it BAYE'S. At that time, we had a secretary who was very skillful in correcting all our spelling and punctuating errors, and even in suitably emending the mathematical formulas in the papers she happened to be typing. The secretary changed my spelling to BAYES', but I wasn't convinced that she was right. Finally, I dug up an old Navy paper, written during World War II, wherein Bayes' Theorem was described, and – sure enough – it was spelled BAYE'S. Thinking that I was vindicated, I started to read the next sentence. It read, "*Its* purpose is to . . . ." Ever since then, we've all agreed that BAYES' is the correct spelling.

This settled, we can turn to the purpose of Bayes' Theorem, which is to yield a Bayes factor. This factor alters the odds in favor of one hypothesis over another, in view of a given set of "pieces of evidence." In some usages of the theorem, there are a multiplicity of hypotheses instead of two, in which cases the Bayes factor can produce the final odds in favor of any one of the hypotheses against all the others. Even with a Bayes factor, however, the a priori odds in favor of each hypothesis must often be taken into account, particularly when the a priori probabilities of the various hypotheses are different from each other. Otherwise we are not obtaining the maximum benefit from its use. For ease in computation, we at N.S.A. almost always use the logarithms of the numbers we are dealing with, and hence we arrive at *log* Bayes factors, which, when multiplied by the a priori odds, form the *final odds*.

One of the simplest possible illustrations of the effective use of Bayes factors will now be given. Suppose I exhibit a stretch of twenty letters, and ask how you can tell whether this is an English sentence, or a collection of letters pulled out of a grabbag (where all letters have equal probabilities). Your answer would be: "I just look at it." An automatic Bayes factor computer which is built inside all human beings would almost instantly tell you the correct answer. But what is a very simple problem for a person can become a much more difficult problem for a computer or for special-purpose cryptanalytic devices. The latter devices must take the first character in the stretch of twenty, and by means of a log Bayes factor obtain a figure representing the log odds in favor of this character's having arisen from English plain text, rather than from a flat-random collection of letters of the alphabet. Then the device proceeds in turn to each of the remaining nineteen letters, pulls out a log Bayes factor for each, adds them all together (this being the equivalent of multiplying the Bayes factors), and obtains a final *score*. This score represents the log odds in favor of hypothesis I (that the characters are English plain text) over hypothesis II (that the characters are random, i.e. equiprobable before they were selected). In this illustration, we have supposed that each hypothesis had an equal a priori probability. When we know that this is not so, special allowances therefor must be made.

Where or how does a computer or special purpose device get hold of its log Bayes factors? A log factor for each letter of the alphabet, A through Z, must be stored in the memory of the computer in advance. The probability of each of the twenty-six letters of plain text is first obtained from a large frequency count (at least as large as practicable). Then each log Bayes factor is found by looking up the logarithm of 26 times the probability of the letter in question. (The multiplier is 26 because there are 26 distinct classes of characters, or letters, in English plain text.) Now, the computer can easily sum the proper values for all the letters in a stretch of "putative plain text." Besides the log Bayes factors, we can compute the size of the expected total score in the "correct" or "plain text" case, for a stretch of twenty characters, and the expected score in the "wrong" or "random" case; and, if we so desire, can order the computer to print out the score only if it exceeds a preassigned threshold. (Normally, of course, we will have other relevant information printed out along with the score.)

Several noteworthy remarks are now is order. The first is that the log factors can be rounded off to one or two digit accuracy, even if they originally were computed with a five-digit log table. Then, these rounded values can be subjected to any linear transformation, such as multiplying them all by one arbitrary constant, and adding a second arbitrary constant to each. When this process has been completed, the values obtained are called *log weights*, or better still, just plain *weights*. It is easy to see that any set of values (as for example the 26 log factors for English plain text) can be condensed into a selected number of weights (say 8 or 32), by making the smallest value 0 and the largest 7 or 31, as the case may be; all other values are determined by linear proportion. The *base* of the logarithms we have been using was not mentioned, because any base is permissible.

Two interesting observations in connection with weight-making can now be made. The first is that, for problems analogous to the foregoing, *only weights made by this process* (or which could have been made by this process) are correct. Weights made by using other methods may range from being very nearly correct (and hence in practice undoubtedly just about as useful) to being so distorted that they are not doing half the job they should be doing. The lesson here is that all weights should be log weights, which in turn are (at most) rounded off and/or linearly transformed log Bayes factors.

The second observation is that altering the log weights by the process outlined serves no theoretical purpose whatever, and can only *weaken* what would have otherwise been the scores. [redacted] However, I had better quickly add that correctly prepared weights of 32, or even fewer, categories are virtually as accurate for practical purposes as the original 5-digit logarithmic factors, at least in all customary situations. But, the important point is that, statistically, we never *benefit* by trying to form weights out of logarithmic Bayes factors, let alone by conjuring up weights by some other process.

I should add, as a footnote to all this, that if our information as to the probabilities involved happens to be erroneous, then improperly prepared weights just might work better than correct ones. In this sense, 3 might turn out to be a better approximation to π than 3.1416 in obtaining the area of a circle *provided* that we had overestimated its diameter. But no one would argue that 3 might be an improved value for π in deriving areas of circles, even though this is actually a true statement under such circumstances as just described.

Next, we consider the problem of what we must do in a particular case if the log Bayes factors are not regularly giving us large enough scores; in other words, not yielding what is referred to as a sufficient statistical separation between right and wrong answers. This occurs when some of the right answers fail to reach our preset separating threshold, while many of the wrong answers exceed this same threshold and hence are mistaken by the computer for correct answers (in our illustration, for English plain text). A difficulty of this type is always aggravated when there is a high ratio of wrong to right cases under consideration.

There are two remedial steps that can be taken for improving the scores. The first method is to increase the number of letters (or characters), say by weighing 40 of them instead of 20, in deriving the score. The second method is to bring in additional information, which, in the plain text example, could come from a digraphic instead of a monographic evaluation of the letters present. This involves preparing and storing 676 log weights instead of the 26 mentioned previously, and hence may begin to assume practical disadvantages. *Caution: Never try to prepare digraphic log weights unless you know exactly how it's done.* And, of course, trigraphic or polygraphic weights are statistically even stronger than digraphic weights, but are generally not considered because they present grave problems in computer storage, and in the need for accurate preparation of such a large number of weights. This is an interesting sidelight on the "inefficiency" of a computer compared with a human being, the latter having almost instant access to literally billions of letter combinations and their plausibilities.

Having suggested two methods for improving our scores, it is important to point out that what we *cannot* do is find another statistic or mathematical function that is better than a Bayes factor. In fact, this is a very important point. A lot of people have been spending a lot of time trying to do this. At best, a different statistic offered for consideration in this type of problem can be *almost* as strong as a Bayes factor. Later on, however, we will discuss cases where it is impractical to carry out all the necessary Bayes factor calculations, and therefore simpler statistics are of necessity substituted.

In connection with alternative statistics, [redacted]

I fully subscribe to [redacted] assertion that the correct statistic to be used is not a matter of opinion. [redacted] then points out that any difficulty in assessing the a priori probabilities (a problem often besetting the cryptanalyst) casts the same shadow over any statistical method whatsoever that is being employed. [redacted] adds that even such famous statisticians as Karl Pearson and Keynes were either "confusing" or "confused" in discussing Bayes' Theorem; and he ends by noting that some other statisticians have claimed that Bayes weights are useless, because "problems in which the probabilities can be calculated" do not occur. However true this may be in agricultural experimentation, concludes [redacted] it is certainly *not* true in cryptanalysis.

[redacted]

Before going on, I'd like to mention, but not fully discuss, a relatively minor problem that occasionally arises when establishing "plain text" probabilities, or the equivalent, which are used in preparing log weights. Let us suppose that 1000 characters of French telegraphic plain text have been frequency-counted, with the thought that we can divide each frequency by 1000 to obtain the approximate probability of the letter in question. But the sample of 1000 by chance had no letter W which as we all know is a rather uncommon letter in French. Does this mean that we assign this letter a probability of zero? The answer is certainly not, as this would eventually result in a log weight of minus infinity, and this would mean, to a computer at least, that any stretch of twenty letters containing a W (even if it happened to be Je-vais-a-Washington) could not be plain text.

This difficulty, when it arises, can be readily taken care of, if not perhaps strictly solved in a theoretical sense, by Dr. Getchell or by myself, among others, for anyone who isn't sure of the best way to circumvent this apparent obstacle. However, a warning is in order at this point: To escape from this dilemma it is very definitely *not* necessary to laboriously assemble an enormous sample of plaintext from which to obtain "refined" probabilities. Perhaps we should not mention one unfortunate case in which a sample of half a million trigraphs were counted over a period of two years in an utterly futile attempt to obtain "more accurate" probabilities, to be used for log weights.

[redacted]

[redacted]

There are many pitfalls to be carefully avoided in the preparation and use of Bayes factors. For example, if we are trying to line up adjacent columns in a simple transposition, the methods already described for recognizing plain text are not applicable at all. For the characters we are now dealing with are nothing but plaintext letters in their proper proportions, and the Bayes factors we are going to use must be based upon the *digraphic* plaintext probabilities of the appropriate language. (We are now, as before, attempting to decide between two hypotheses.) Having placed two possible columns side by side, do the pairs of adjacent letters appear to be plaintext digraphs, or do they appear to be two separated plaintext characters? The Bayes factor, therefore, tests the hypothesis that the digraphs were originally together rather than separate. Such digraphs as EA will have very low transposition-type log weights, while a digraph like CH will have a high log weight quite dissimilar to our earlier weights. One set of French transposition log weights, arranged in order of descending size, was headed by "WY", in fact. Digraphs made up of very common letters like AN often have a neutral log weight. The combined weights of all the digraphs in the paired columns naturally make up the score used to determine whether these particular columns are in fact adjacent or non-adjacent. In the right case, the columns must not only be adjacent but must occupy their proper left and right relative positions. But anyone who has worked on transpositions knows that the scores thus obtained, while helpful, by no means afford conclusive evidence of the pairing or the non-pairing of columns. The relatively small number of digraphs available for scoring in any given pair of columns allow false answers rather frequently to attain the same scores as that of the average right answer.

[redacted]

EO 1.4.(c)
P.L. 86-36

EO 1.4.(c)
P.L. 86-36

[redacted]

EO 1.4.(c)
P.L. 86-36

[redacted]

EO 1.4.(c)
EO 1.4.(d)
P.L. 86-36

[redacted]

[redacted]

EO 1.4.(c)
EO 1.4.(d)
P.L. 86-36

[redacted]

EO 1.4.(c)
P.L. 86-36