4.6 Correlation Matrices

Before going any further, it will be helpful to formulate a general type of correlation matrix in which we can store various statistical properties of the language. (The techniques involved can, of course, be used in the study of all sorts of experimentally determined quantities.)

Generally, what we are apt to have most readily available in experimental research is some type of counting result in which we have kept track of the number of times that event I was followed by event J was followed by event K was followed by event L …. This type of quantity can be stored in a multidimensional matrix.

$$M (I, J, K, L, \ldots)$$

which is computed by adding one to the element I, J, K, L, … every time a new sequence I, J, K, L, … is encountered. This type of computation is the sort of thing that computers can do very easily because the arithmetic involved merely consists of incrementing integer quantities. The only problems of significance are ones of core size and access methods.

Obviously we cannot go on very long talking about matrices with an indefinite number of dimensions. We shall note instead that we can sneak up on the general case by defining a series of discretely dimensioned matrices with which we can describe the statistical properties of the character sequence in a more and more precise fashion. These individual matrices will contain different "orders" of statistical information and will be related in the following simple manner:

$$M = \qquad (6)$$

That is , the zeroth-order "matrix" is just the total number of events,

$$M = \qquad (7)$$

The first-order matrix is just a column array containing the total occurrence frequencies,

$$M(I) = \qquad (8)$$

The total second-order matrix giving correlation between successive pairs of characters is determined from the third-order matrix by the sum

$$M(I,J) = \qquad (9)$$

and so on.

Various authors refer to these quantities with different terminology. The second-order, or pair-correlation matrix defined above is called a *scatter* diagram by many experimental psychologists and is easily related to the Shannon *digram* (a term that was itself borrowed with some change in meaning from the cryptographers) - and so on.

We may similarly define a set of normalized probabilities:

$$P(I) = \qquad (10)$$

represents the total probability that the Ith character occurs;

$$P(I,J) = \qquad (11)$$

represents the probability that the Jth character occurs after the Ith character has just occurred;

$$P(I,J,K) = \qquad (12)$$

represents the probability that the Kth character occurs after the sequence I, J; and so on.

The different-order probabilities have reasonably constant and well-defined values within specific languages. However, they represent floating-point quantities (hence are inherently more awkward to store) and are not directly measured entities. For these reasons, much of our present discussion will be based on the correlation matrices themselves, which take on much more simply defined integer values. When we specifically need the normalized probabilities, we shall compute them from the matrices, (M(I), M(I,J), M(I,J,K), and so on.

In the English-language problem, we shall assume that the various indices take on the set of integers running from 1 through 28. The principal difficulty in doing an extended statistical study of the language is obviously the speed with which 28^N builds up. Specifically,

$$28^2 = 784, \quad 28^3 = 21{,}952, \quad 28^4 = 614{,}656, \quad 28^5 = 17 \text{ million, etc.} \quad (13)$$

Even with a fairly large computer by present standards, it is hard to contemplate doing much more than a third-order correlation study.

Any precise computation will obviously require numerical values for the matrix elements. However, it will be helpful to have a quick look at the qualitative structure of the first several correlation matrices in a language such as English before going on to simulate a higher-order Eddington monkey experiment. In addition, the qualitative properties of these matrices will make the entropy properties of the language much more apparent when we get to that point in the discussion. One simply cannot visualize the relative probabilities involved merely by looking at 28, 28x28, and especially 28^3 numbers.

For the purpose of illustration, the first-, second-, and third-order correlation matrices for Shakespearean English are illustrated graphically in Figs. 4-3, 4-4, and 4-5. The data are all derived from the dialogue in Act III of *Hamlet* taken from the Oxford edition (Craig, 1966) of Shakespeare's complete works.

The histogram in Fig. 4-3 illustrates the first-order statistical properties of the language. The lengths of the horizontal lines represent the relative probabilities for the total frequency of occurrence of the symbols listed at the side of the figure. Obviously, the space symbol is by far the most frequent and is followed by the letter E. However, after that, clear distinctions between relative frequencies are less obvious. In this » 35,000-character sample, the letters J, Q, X, and Z occur very rarely. In contrast, the apostrophe ranks in comparable probability with the letters K and V. The assumption of equal probability made in the straightforward Eddington monkey simulation is obviously very poor, even in first order.

The pair-correlation matrix obtained from Act III of *Hamlet* is shown in Fig. 4-4. Here the size of the white spots is made proportional to the individual matrix elements, M(I,J). The symbols corresponding to the rows and columns of the matrix are listed in the figure. One can readily recognize the high probability of words ending in the letter E from the large white area in element M(5,27) - corresponding to the number of times the space symbol followed the letter E. Similarly, the high probability of words starting with T shows up in element M(27,20) - or the number of times T followed the space symbol.

One can also readily spot the extremely high probabilities for the letter sequences TH, HE, and so on, along with less frequently occurring, but highly correlated, pairs such as QU and EX.

In Fig. 4-4 the dark spaces are almost as important as the bright spots. Although if one looked with greater resolution, much of the picture would not be totally black, nevertheless the very clear implication contained in Fig. 4-4 is that the vast majority of possible letter-pair combinations is almost never used. (There are » 291 appreciable matrix elements out of a total possible number of 784 in the figure.) Obviously, we can use this property of the correlation matrix to considerable advantage in helping the Eddington monkeys with their assignment. Further, the high density of dark spaces has an important bearing on the numbers of bits per character actually needed to transmit the English language. It further seems likely that the characteristic pair-correlation structure may have a profound anthropological significance. (Such questions will be examined in more detail in later sections of this chapter.)

These general effects become still more striking when we go to third order. The data shown in Fig. 4-5 are again based on Act III of *Hamlet*. Here we have broken up the 28 x 28 x 28 (= 21,952)-element third order correlation matrix into 28 separate pair-correlation matrices of the type discussed previously in connection with Fig. 4-4. The difference is that the data displayed in Fig. 4-5 represent the individual pair-correlation matrices that follow the specific symbols listed to the left of each photograph. The photograph in the upper left-hand corner corresponds to the pair matrix that would follow the occurrence of the letter A; the next one to the right corresponds to the pair matrix that would follow the letter B; and so on. (The same labeling of rows and columns given in Fig. 4-4 is tacitly implied in each of the photographs in fig. 4-5.) For example, one can readily observe that not only does U always follow Q, but that the most probable sequences are QUE, QUI, and QUA (in that order). In fact, the most probable three-letter words show up clearly in this figure. Thus the bright spot in the matrix following the letter T is the well-known, most probable word in English,

THE. Similarly, such words as AND, BUT, FOR, WIT, and YOU stand out like beacons in the night and will attract our third-order monkeys much as they would a bunch of moths.

4.7 Second Order Monkeys

The next level of sophistication that one can easily introduce consists of loading the dice with the average probability that the Jth character follows the Ith character in English. Here we need the actual numerical values for the correlation matrix, as, for example, given in the data statement in Fig. 4-6 (based on the dialogue from Act III of *Hamlet*). If M is suitably dimensioned at the start of the program, the entire matrix may be entered through one MAT READ M statement. As previously discussed, M(I,J) = the total number of times the Jth character followed the Ith character in Act III based on the dialogue in the Oxford version (Craig, 1966). The notation on the rows and columns corresponds to the same convention used in subrouting 5ÆÆ. For example, the first row of the matrix implies that

A followed A zero times
B followed A 19 times
C followed A 63 times etc.

Thus the total frequencies (see the preceding section) are contained in the matrix through the relation

F (I) =

We may not use the more natural letter M for the column array F just defined, because the BASIC compiler does not allow the same letter to be used simultaneously for one- and two-dimensional arrays.

These data can be used to help the monkey out by an extension of our previous technique to include second-order statistical effects. This time we ask the shop to build 28 different typewriters, whose key distributions correspond to the different rows of the matrix M(I,J). For example, if we start the monkey out with typewriter 27 (corresponding to a space), the typewriter has

F(27) =

of which there are 627, A's, 329 B's, 218 C's, …, 0 space keys, and 28

apostrophes. (We deliberately defined M(27, 27) = 0 to avoid long sequences of spaces.)

We let the monkey hit one key (i.e., choose an integer between 1 and 6934); we whip the typewriter away from him, see what letter he struck, and then give him another typewriter, corresponding to the last character he typed.

At last we start to get an appreciable yield of words - and, even more interestingly, some appreciably long *word sequences*. The latter is a little surprising because we have only incorporated the statistical correlation between *pairs* of letters. Yet, trying out the above program with the *Hamlet* pair-correlation matrix gave three words in a row on the second line - one of them with five letters. Specifically, the second-order Shakespearean monkeys started off:

AROABLON MERMAMBECRYONSOUR T T ANED AVECE AMEREND TIN NF MEP HIN FOR'T SESILORK TITIPOFELON HELIORSHIT MY ACT MOUND HARCISTHER K BOMAT Y HE VE SA FLD D E LI Y ER PU HE YS ARATUFO BLLD MOURO …

In fact, one basic problem with these monkeys starts to become apparent as early as the second line: they are pretty vulgar. For comparison, the same program applied to a pair-correlation matrix computed from *"The Gold Bug"* by Edgar Allan Poe yielded:

ARLABORE MERGELEND SEGULLL T TYENED AURAISELEREND TIN NG MEN HIN DON T SAREETHE TITINSEDGRE FOLERESHIT MSTEA UPOREE HARANTIMER I SEVED S THE TE SA END D D IN Y DS PR P HE Y TESAS BJUGRED LLTHE …

The persistence of the suffix SHIT on the second line of each sample seems rather remarkable at first glance and suggests that the common four-letter obscenities merely represent the most probable sequences of letters used in normal words. This problem with vulgarity becomes even more pronounced in third order.

At the pair-correlation level one also begins to recognize characteristics differences between individual languages in the monkey simulation program. Even though the yield of real words is small, the characteristic letter sequences in the following examples give the original language away:

Second-Order Italian Monkeys:

A T I A B E D O V E T I C E N I C O C C H E I S T O A R E L I A LALLANDERSENTRETRINTIOR E E DESUTTOISENORE SI ITOLANON DEPEVE CI VE MACO LLLEN ENOLE LCHE GNA CCO VONE SA PA DELIGNDUIO VILE N SESSUE AVA NCHIDIOMPIVORE LITOMO TI POLINANCE DA AVA ULLLAN SSA TA IR SACO CCALA QUSTIA UE PA RI BANOSERSI PRMBO PRI TESE O QUSE E CON QUATUANDI HE …

Second-Order German Monkeys:

A N S A B I N E I L I L B E I G E T U E L L E R N T S AMEILAUNDERALENENDISSPRSIRNIG ERISENI US ANEINGER HUNSTEIERE DELENINER WESTEBUSTSTEITEINDEROFOL GSCHEIS ZWEMPRAT A DEIMATE GE ZUHERT VIGT ETERASTEN DEND IN FR IMM DR WERUNDENDEIEREINDIES GENAL T CH D IN VEBRUFFADAT DR JA WEWICHTS BEMIMEN IS WIES R M WENE N SM E ESCHEUNGAN BEKS …
(note the long words)

Second-Order French Monkeys:

ARIABLIL'HESTSERDEL OILLE L'OUS ANGESA LAISERESINE QUN LE LES'S E DES'UVICILEXINT JONS CENTE DERETIRE PURS BA SYS DE ENSET LESS GOIRENUS QUIS AUSA DEMEPRE GI VILE MOUME VE BLAT CHUETIE LLSST LEUSE PTIS NETELENE DE BLE UNSTAL'QUE SJURI SECONSENAGAUSE S A UMOUE QU'AGESTES LUS PE PPRI TINFUS PHON E DUIT EFI CEPLUNE …

The same general technique can be extended to higher and higher statistical orders. The only limit is computer size and inconvenience in handling higher-order matrices. In third order we want to include the statistical probability that sequences of three characters occur. Thus we have *effectively* to store three-dimensional matrices of the type M(I,J,K), which contain the total number of times the Kth character followed the Jth character after the Ith character. The main difficulty is that there are 28 x 28 x 28 = 21,952 different matrix elements to include, and one starts to feel memory limitations in the data-storage allocation on modest-sized computers.

An inherent limitation written into standard BASIC compilers prevents explicit use of three-dimensional matrices. That is, a dimension statement such as