Statistical Behavior of Search Keys

Abraham BOOKSTEIN: Graduate Library School, University of Chicago

Editor's note: The editor and author are aware that varying approaches may be taken to the problem presented here. Readers are invited to respond in the form of a paper or a technical communication.

In discussion about search keys, concern has been expressed as to how the number of items retrieved by a single value relates to collection size. This paper creates a statistical model that attempts to give some insight into this behavior. It is concluded that, in general, the observed behavior can be explained as being intrinsically statistical in nature rather than being a property of specific search keys. An attempt is made to relate this model to other research, and to indicate how this model may be made to yield more accurate predictions.

INTRODUCTION

Various experiments suggest that it may be possible to develop, as an access route into a file of bibliographic records, a search key* whose values can be easily derived from such bibliographic data as is likely to be available to its users.[1] Some concern, however, has been expressed regarding the nonuniqueness of these keys: if the number of items retrieved were often to exceed an amount easily handled by a user of the system, the value of this access route would be considerably diminished. Accordingly, an important measure of search key performance is the frequency with which a large number of records is retrieved as the search key is applied to the file. This measure is related, for example, to how many memory accesses will be required, on the average, to retrieve all records satisfying a request; it is also an important consideration in deciding which display device should be installed in a system.[2, 3]

After evaluating such a measure for a search key on a particular file, it is reasonable to ask how that measure will change over time, as the file increases in size. The nature of this variation has already been of concern to researchers in the field. Kilgour, on the basis of a number of experiments carried out at OCLC, notes that "There remains a major problem to be

* By the phrase "search key" we mean a key similar to the 3-3 or 3-1-1-1 keys used at Ohio College Library Center and other places, which is made up by concatenating truncations of bibliographic data elements.

solved and a major question to be answered. The problem is constituted of those replies that contain a number of entries exceeding the optimal maximum. . . . The major question to be answered is how truncated search keys will perform on files ten and a hundred times the size of that used in this experiment."[4] He elsewhere observes that "as a file of bibliographic entries increases, the maximum number of entries per reply does not increase in a one-to-one ratio. . . ."[5] This paper presents a mathematical model that addresses itself to the problem defined by Kilgour and attempts to explain his observation; it is suggested that the gross features of the behavior are statistical in nature and not properties of specific search keys.

A VIEW OF COLLECTION GROWTH

The cause of the phenomenon observed by Kilgour can best be understood by first considering a simple model which, while not itself valid, does cast light on the nature of the behavior. This first model neglects the effect of randomness both in the growth of the collection and in the arrival of requests. It supposes our search key has the following property: regardless of collection size, the fraction of the collection retrieved by a particular search key value, v_i, is *exactly* given by a constant f_i; thus, if the file holds N records, a request for v_i will retrieve $n_i = f_i N$ records. This model similarly assumes that among any sizeable number of requests, the fraction of the time any particular search key value will occur is fixed; thus, for any subset of search key values, it is possible to determine how often members of that subset will occur among a set of requests.

In particular, for any integer n, we can form the set of all the search key values that will retrieve less than n items. We can then determine how often search key values from that set are requested. If, for example, requests for these values occur 99 percent of the time, then we can assert that 99 percent of the time less than n items will be retrieved. If the file contains N items, then these n items constitute the fraction $f = \frac{n}{N}$ of the file. Should the collection size increase to lN, then the model predicts that 99 percent of the time less than $f(lN) = ln$ items would be retrieved. In other words, we have precisely the behavior Kilgour observes does not occur. This argument shows that a simple *deterministic* model does not conform to experience with search keys.

The model breaks down in two ways, which accounts for the discrepancy between the results derived from it and Kilgour's observations:

1. in any actual library, the fraction of the time that a particular request will appear within a sequence of requests will vary; and
2. in comparing two different samples having the same size, the number of items having a given search key value will vary.

The first of these factors is easily dealt with and its analysis will suggest the number of requests to use in a test of search key behavior in a given library. For a particular collection, let S denote the set of search key values for which, say, twenty or more items are retrieved. We would like to find the fraction of the time that a request in S occurs in the long run; suppose this value is in fact q. Then among M requests, the probability that m members of S occur is given by the binomial distribution $f_B(m|q,M)$. This distribution has a mean of qM and a variance of $qM(1 - q)$. Should we desire to estimate the actual fraction of the time that twenty or more items will be retrieved, we can take a sample of M requests and compute $\hat{q}$, the fraction of the requests with search key values in S; if we do so, we will usually get a value for $\hat{q}$ between $q - \frac{2}{\sqrt{M}}\sqrt{q(1-q)}$ and $q + \frac{2}{\sqrt{M}}\sqrt{q(1-q)}$. If for example, $q = .01$ and $M = 10{,}000$, we would tend to find $\hat{q}$ in the interval .01 ± .002. Thus the effect of randomness in the arrival of requests can easily be controlled by increasing the number of requests considered; furthermore, the size of error can be predicted.

We next introduce the second factor; its analysis will suggest how the behavior of search keys will change as the collection grows in size. For this purpose we adopt a model of collection growth which assumes that as items arrive, they are randomly distributed among the search key values in accordance with some probability distribution. If we suppose that the probability of an item being assigned a specified search key value, v_i, is p_i, then in a collection of N items we may conclude that the probability of n items having that value is given by the binomial distribution:

$$f_B(n|p_i,N) = \left(\frac{N}{n}\right) p_i^{\,n}(1-p_i)^{N-n}.$$

If $g'(v_i)$ is the probability that the value v_i is selected from the request population, then the probability that the "next" request retrieve n items is given by

$$\Sigma_i\, g'(v_i)\, f_B(n|p_i,N) \approx \int g(p)\, f_B(n|p,N)dp;\; g(p)\, dp \overset{\text{def}}{=} \sum_{p \leq p_i \leq p+dp} g'(v_i)$$

is the probability that a request arrive with value p_i in the interval $(p,p + dp)$, and will be treated as a continuous function.[**] Since the expectation of the binomial distribution is given by pN, we have

$N\int pg(p)dp \overset{\text{def}}{=} N\bar{p}$ as the expected number of items retrieved by a random request; since this is proportional to N, *doubling the size of the collection will, on the average, double the amount of material retrieved.* Similarly, the variance, σ^2, is given by $N^2(\overline{p^2} - \bar{p}^{\,2}) + N\int p(1-p)g(p)dp$. Should $\overline{p^2} - \bar{p}^{\,2}$, the variance of p, be small, this reduces to $N\int p(1-p)g(p)dp \overset{\text{def}}{=} \tilde{\sigma}^2 N$, so that approximately 95 percent of the time the amount of material retrieved would be less than

$$N\bar{p} + 2\sqrt{N}\,\tilde{\sigma} = N\left(\bar{p} + \frac{2\,\tilde{\sigma}}{\sqrt{N}}\right).$$

[**] This result would more precisely be expressed as $\int f_B(n|p,N)dG(p)$, which has the form of a Stieltjes integral. The expression used in the text is simpler and reasonably valid because of the vast number of values the search key can take.

It is the factor

$$\bar{p} + \frac{2\,\tilde{\sigma}}{\sqrt{N}},$$

and its dependence on N, that may account for Kilgour's nonlinearity, and not any property intrinsic in the nature of any type of search key. Thus, to the extent that this model reflects what is really happening, the 95 percent point increases roughly proportionately with file size; the "constant" of proportionality, however, is the sum of two terms: the first is a true constant, and the second is a term that approaches zero as the file gets larger. In particular, this model suggests that we will never reach a leveling off point—as the file increases in size, the number of items retrieved will also increase, and the pattern of increase will become increasingly linear.

Up to this point this discussion has been qualitative in nature, being based upon general statistical considerations and making use of the normal approximation to some unknown distribution; its broad conclusions are, however, consistent with the findings of earlier workers and can explain certain unanticipated properties of search keys. To proceed further it will be necessary to restrict the form of the function g(p); this will be attempted in the following section of this paper.

RELATIONSHIP OF MODEL TO EARLIER RESEARCH

Interest in access methods that are appropriate for files of bibliographic data has generated a considerable amount of empirical research on search key behavior. Of necessity, this pioneering work has been of a descriptive nature, resulting in data showing search key behavior in specific environments. While these efforts have lent a good deal of insight into the nature of search keys, the basic weakness of such research lies in the difficulty of extending these findings to other situations. One purpose of a mathematical model such as the one being developed here is to provide this increased generality by representing in a concise and easily manipulated form the results of previous research. It is accordingly of interest to indicate the relationship between previous work on search keys and our model.

Research on search key performance has been of two kinds. The first kind seeks to answer the question: for any number, n, how many search key values retrieve n items? The answer to this question depends only on the search key and the collection; it is independent of the pattern of request arrivals. The second kind of research involves the actual arrival of requests; it tries to answer the question: for any number n, how frequently will requests resulting in the retrieval of n items occur?

To discuss this research in terms of our model requires a closer examination of the function g(p) previously defined. We recall that $g(p)\ dp \overset{\text{def}}{=} \sum_{p \leqq p_i \leqq p+dp} g'(v_i)$, with dp being a small number. Thus g(p) is determined by two factors:

a. The number of search key values in the interval $(p, p + dp)$. Let us denote this value by $f(p)dp$, so $f(p)$ is the density of search keys at p. We make use here of the fact that although the number of possible search key values is finite, the number is very large, so their distribution can be thought of as continuous.

b. The average probability of search keys, with values p_i near p, being requested. We shall refer to this quantity as $g''(p)$. By combining these factors we have $g(p) = g''(p)f(p)$.

In terms of this discussion, the first type of research described above is in fact estimating $f(p)$: if there are s search key values that retrieve n items from a collection of N items, then s is an *estimate* of

$$\frac{1}{N}\ f\left(\frac{n}{N}\right);$$

this relation uses

$$n \doteq pN, \text{ and } dp = \frac{n + \frac{1}{2}}{N} - \frac{n - \frac{1}{2}}{N} = \frac{1}{N}.$$

The second kind of research directly estimates $g(p)$. Guthrie, in a recent paper, provides a bridge between the two types of research by discussing his findings in terms of two models.[6] One of his models, which asserts that each search key value has an equal chance of being requested, is equivalent to the assumption that $g''(p) = 1$, and $g(p) = f(p)$. Guthrie finds that this is not an adequate representation of his data.

Guthrie's second model asserts that each item has an equal chance of being requested. In our terms this becomes $g''(p) \propto p$, and $g(p) \propto pf(p)$. This model, while an improvement over the first, still disagrees with the data. Furthermore, these models do not estimate $f(p)$; even if Guthrie's model were correct, we would not know the probability that n items would be retrieved until we were told how many search key values contained n items. In the next section we will try to remedy this situation by means of a two parameter representation of $g(p)$.

A REPRESENTATION OF $f(p)$

To get a more detailed account of search key behavior by experiment is difficult since the two aspects of randomness already discussed are confounded; the experimenter only sees the combined effect. We will, however, try to estimate the distribution $g(p)$ by a distribution of the form

$$\frac{(\alpha + \beta + 1)!}{\alpha!\,\beta!}\ p^{\alpha}(1 - p)^{\beta}.$$

We believe that such an attempt is reasonable on three grounds:

a. It is not possible to find $g(p)$ exactly, and moreover, it is not clear that this would be desirable. We are interested in a reasonable approximation that is satisfactory for decision-making purposes;

b. The above distribution assumes a wide variety of shapes as α and β vary; it seems likely that values of α and β can be found for which this distribution is close enough to g(p); and

c. This distribution is mathematically tractable.

If we proceed using the above approximation for g(p), we find:

(i) the probability, P(n), of n items being retrieved is given by

$$1.\quad P(n) = \left(\frac{N}{n}\right) \frac{(\alpha + \beta + 1)\,!}{\alpha\,!\;\beta\,!}\;\frac{(\alpha + n)\,!\quad(N - n + \beta)\,!}{(\alpha + \beta + N + 1)\,!}$$

(ii) the expected number of items retrieved, E, is given by

$$2.\quad E = N\,\frac{\alpha + 1}{\alpha + \beta + 2}\;;\text{ and}$$

(iii) the variance, V, of the number of items retrieved is given by

$$3.\quad V = N\,\frac{\alpha + 1}{\alpha + \beta + 2}\;\frac{\beta + 1}{\alpha + \beta + 3}\,(1 + \frac{N}{\alpha + \beta + 2}).$$

If the experiment is performed on a small sample, the expectation and variance can be computed and the values of α and β estimated from the relations

$$4.\quad \beta = \frac{\alpha\,(1 - \frac{E}{N}) + 1}{\frac{E}{N}} - 2,\text{ and}$$

$$5.\quad \alpha = \frac{E - \dfrac{\frac{V}{N}}{1 - \frac{E}{N}}}{\dfrac{\frac{V}{E}}{1 - \frac{E}{N}} - 1} - 1$$

Usually $\frac{E}{N}$ will be much smaller than one; in this case we may use the approximations:

$$4'.\quad \beta = (\alpha + 1)\,\frac{N}{E}\,,\text{ and}$$

$$5'.\quad \alpha = E\,\frac{E}{N} - 1.$$

Once α and β have been evaluated, we can compute the probabilities P(n) for files of arbitrary size, and with these values we can make assertions regarding the probability of, say, more than 30 items being retrieved. A relation that can be derived from Formula 1 and may be of use when comparing this model with experiment is:

$$\frac{P(n)}{P(n + 1)} = \frac{1 + \frac{\beta}{N - n}}{1 + \frac{\alpha}{n + 1}}$$

The probability of zero retrievals is likely to be an extraordinary point in the distributions g(p) and P(n) since it is influenced by the knowledge that a user may have of the collection; this effect is likely to be encountered in a sampling process in which the requests have to be generated artificially. In such cases it would be advisable to treat P(0) as an empirically derived parameter, θ, and use the modified formula

$$6.\quad P'(n) = \begin{cases} \theta \text{ if } n = 0 \\ (1 - \theta)\dfrac{P(n)}{1 - P(0)} \text{ if } n \neq 0. \end{cases}$$

The value of θ can be estimated by the fraction of requests retrieving zero items; for sampling techniques using only productive requests, θ will be zero. α and β can be calculated as before from the mean and variance of the sample.

CONCLUSION

The above discussion is intended as an attempt to provide some theoretical understanding of the puzzling behavior discovered in the use of search keys and also to provide some guide for those experimenting with samples of such files. We do, however, urge caution for the latter uses.

An analysis similar to the above can be useful under several different circumstances, such as: determining the future behavior expected of a search key in a single library as the collection grows; determining the behavior for one library based upon experiments conducted on a different but similar library; and extrapolating from the performance of a search key in a sample of the collection to its performance in the full collection.

If one wishes to compare two different libraries, one can note that as far as search key values are concerned, a particular library's collection can be thought of as a random sample of the larger population from which it selects its material, and accordingly the formula for P(n) should be valid. In this case, if two different collections are drawn from the same population, the g(p) refers to this population and the libraries are distinguished by the parameter N; when we are considering samples from a single library, then N is the sample size and g(p) refers to the library itself.

No theoretical basis exists at present for estimating to what extent the populations being considered depend upon the type of library, if any, so this problem must be dealt with empirically. We have assumed here that these populations are similar with regard to search key values. Should these populations in fact vary, it is possible that they can be broken down, e.g., by language, into subpopulations that *are* stable and for each of which the analysis is valid.

ACKNOWLEDGMENTS

This work was made possible by CLR/NEH Grant No. EO-262-70-4658. I would like to express my gratitude to members of the University of Chicago Systems Development Office for their many comments and suggestions on this work.

REFERENCES

1. Frederick G. Kilgour, Philip L. Long, Eugene B. Leiderman, and Alan L. Landgraf, "Title-Only Entries Retrieved by Use of Truncated Search Key," *Journal of Library Automation* 4:207-10 (Dec. 1971).
2. A. Bookstein, "Double Hashing," *Journal of the American Society for Information Science* 23:402-25 (Nov.-Dec. 1972).
3. A. Bookstein, "Hash Coding with a Non-Unique Search Key," to be published in the *Journal of American Society for Information Science.*
4. Frederick G. Kilgour, Philip L. Long, Eugene B. Leiderman, and Alan L. Landgraf, "Retrieval of Bibliographic Entries from a Name-Title Catalog by Use of Truncated Search Keys." preprint.
5. Kilgour, Long, Leiderman, and Landgraf, "Title-Only Entries," p.209-10.
6. Gerry P. Guthrie and Steven D. Slifko, "Analysis of Search Key Retrieval on a Large Bibliographic File," *Journal of Library Automation* 5:96-100 (June 1972).