Database Evaluation Using Multiple Regression Techniques

Jane Fedorowicz
Kellogg Graduate School of Management
Northwestern University
Evanston, Illinois 60201

312/492-3427

ABSTRACT

A model of the inverted file of an automated bibliographic system is constructed using the Zipf distribution of word frequency. By ascertaining the parameters of the Zipfian model of the inverted file system, one can estimate the minimum data storage requirements of the database. In addition, given a few additional system parameters, access time for a specified query can be estimated. The estimation procedures are accomplished using logarithmic transformations and multiple regression techniques. This paper introduces the Zipfian models, their regression formulation, and their results and interpretation for application to database evaluation.

Introduction

Database evaluation techniques usually are classified as simulation models, queueing models or monitoring systems. Many of these models are concerned with evaluating the appropriateness of a particular hardware configuration on a mix of programs that will be run on the systems. All entail the construction of complex system models, and variances in their accuracy can be attributed to assumptions made about the system, and to the accuracy and extent of parameterization necessary to construct reasonable estimators. Differences in the cost of building and using these models are also significant.

Detailed simulation models, such as IBM's CSS, FIVE (Nguyen et al. [1980]), ANCICSVS (Seaman [1980]), QSIM (Foster et al. [1974]), and APLOMB (Reiser and Sauer [1978]), are complicated, requiring large initial investments of time and money. These investments are not warranted by a single use; thus these simulators are best suited for development by specific manufacturers, and may not be publicly available. General packages, such as SCERT and CASE, are more useful as flexible evaluation tools (Borovits and Neumann [1979]).

Queueing models are one of the least costly methods of evaluating system performance (Rose [1978], Muntz [1978], Allen [1980]). Most queueing network models require strong simplifying assumptions to be made in order to obtain exact solutions (Chandy and Sauer [1978]). Chandy and Sauer propose approximate solution methods to be used when the simplifying assumptions render exact solution models "unfaithful" to reality.

The third method, monitoring systems, is both the most credible and the most costly of all the tools described here. Hardware monitors are discussed in Borovits and Neumann [1979]. Software monitors range from accounting systems, recording a few simply system statistics, to complicated monitors keeping track of more than 500 system parameters.

All of these techniques evaluate computer systems at a macro level. In contrast, this study has a different focus. We look at the performance of the system at the level of the individual query to the database. We start by examining the physical design properties of the database stored as an inverted file. We then present a model dependent on the characteristics of the database rather than strictly on the hardware. Finally we demonstrate that multiple regression is a very good technique for predicting the access time required to respond to a query.

We first present a brief discussion of multiple regression. Then, the inverted file model is developed and tested using regression techniques. Minimal storage requirements are derived from these results. Following this, the access time model is presented and tested.

Multiple Regression[1]

Multiple regression techniques are usually associated with the methodological area known as econometrics, the study of the relationships between economic variables. The simplest relationship is of the form

$$Y = \alpha + \beta X \qquad (1)$$

which says that the variable Y is determined by a linear transformation on X. We know that this equation will not hold exactly for all (X, Y) pairs, and also that it is probable that not all variation in Y is accounted for by the variation in X. Both of these are handled by the introduction of a stochastic (error) term into equation (1).

$$Y = \alpha + \beta X + \varepsilon \qquad (2)$$

The error term (ε) also allows for a certain amount of randomness to be present in Equation (2). In addition, errors of observation or measurement will be picked up by this term. It should now be obvious that the better the fit of the equation to the actual values of Y, the smaller the variance of ε will be. For example, let Y be the number of publications a researcher has, and X the number of years since the researcher was awarded a Ph.D. We would expect that not all researchers who have had a doctorate for 5 years will have exactly the same number of publications. There would be some variance noted (probably rather high in this simple example). We could try and reduce the variance by adding additional X variables, such as the ranking of the researcher's institution on some scale of "excellence." If the variables we add have some correlation with Y (but not much with each other), we should get a reduction in the variance of ε. This brings us to the general model which is used in this study.

$$Y = \alpha + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_n X_n + \varepsilon \qquad (3)$$

Equation 3 does not limit us to linear relationships between Y and the X's. X_k may be the inverse of a variable, its square, or perhaps its log. A transformation that is important to this study is the logarithmic one, in which we are essentially testing the following relationship:

$$Y = \alpha X_1^{\beta_1} X_2^{\beta_2} \ldots X_n^{\beta_n} \varepsilon \qquad (4)$$

Taking the logs of both sides of equation (4) prior to running the model will allow us to use linear techniques of parameter estimation.

After estimates of α, β and the variance of the error term (called σ^2) are obtained, statistical tests are performed to ascertain the appropriateness of the model and its "goodness of fit." The three tests reported in this study are described here: the t-test, F statistic, and R^2.

The t-test determines the extent that each X variable explains some of the variance in the Y variable. The computed t statistic is compared to a critical value determined by the number of degrees of freedom and the level of accuracy desired for the model. If it exceeds the critical value, the X variable has explained a "significant" amount of Y's variance.

The F statistic tests whether the regression equation itself is appropriate for its intended use. A similar critical value test is conducted. Thus, we can test if a linear or logarithmic or some other type of model is better suited to an application.

An R^2 value of 1.0 connotes that the Y variable is perfectly explained by the right side of the equation (the X variables). The higher the R^2 value, the better the explanatory power of the model. A value close to zero shows that the model has little, if any, explanatory power. The R^2 values reported in this paper are adjusted to reflect the proportion by which the estimated variance in Y is reduced when the X's are taken into account.

A Model of the Inverted File

The distribution of term frequency in the inverted file of an automated bibliographic system has been shown to belong to the class of distributions commonly known as Zipf's Law (Zipf, [1979], Fedorowicz [1982a]). Zipf's Law states that if all unique words in a text are arranged (or ranked) in order of decreasing frequency of occurrence, the product of frequency times rank yields a constant which is approximately equal for all words in a text. A reformulation of Zipf's Law which has been adapted to the inverted file was first presented in Fedorowicz [1982b]).

The model was tested on the MEDLINE databases, comprising the automatic bibliographic system of the National Library of Medicine. Two main search mechanisms are available to the MEDLINE users. The first is keyword search, which is a controlled vocabulary. The second mechanism is the free text search, which enables the user to search on any word occurring in the title or abstract in a citation. Boolean operations can be used. All keywords and free text words are stored in an inverted file format, so that the actual citations are not accessed until a print command is issued.

---

[1]This discussion is adapted from Johnston, (1972). The reader is referred to his book for a more complete discussion.

The search mechanisms operate principally via an Inverted Index method. The Inverted Index method is a three step process, each of which consists of a separate file structure. These are the Index File, the Postings File and the Header File.

The Index File is an alphabetic listing of all the search terms which allow access to the citations, including authors, text words, MeSH headings, Computer Assigned Numbers (CAN), etc. An entry in the Index File consists of one search term followed by an indication of the type of search term it is (text word, author, etc.) and a two part number. The first part of the number is the sequence number address, which gives the location of the entries for the search term in the Postings File. The second part gives the number of postings (each citation counts as one posting) containing the term.

The Postings File is comprised of groups of CAN corresponding to the citations associated with each term in the Index File.

The Header File contains the actual citations. It is accessed during print statements by the CAN retrieved by a search on the Postings File.

In order to apply the reformulation of Zipf's Law to the inverted file, the concept of dividing the distribution into groups of frequency counts was adopted. Briefly, each word frequency group, G_m, is approximately equal to the number of unique words found in the Header File which occur between 2^{m-1} and 2^m-1 times. That is,

$$G_m = (kT)^{1/\alpha}\left[\frac{1}{(2^{m-1})^{1/\alpha}} - \frac{1}{(2^m)^{1/\alpha}}\right] \qquad (5)$$

where k is a constant, T is the total number of words and symbols in the Header File, and $\alpha>0$.

Although Equation (5) itself is non-linear, it was possible to perform the estimation using regression after taking the natural log of Equation (5):

$$\ln G_m = 1/\alpha \ln k + 1/\alpha \ln T + \ln (2^{1/\alpha}-1) - 1/\alpha\, m \ln 2 \qquad (6)$$

This is equivalent to a linear regression of the form where

$$Y = \delta_o + \delta_1 X$$

where
$$Y = \ln G_m$$
$$\delta_1 = 1/\alpha$$
$$\delta_o = 1/\alpha \ln k + \ln (2^{1/\alpha}-1)$$
$$X = \ln T - m \ln 2$$

The estimated regression equation is then:

$$\ln G_m = -2.785 + 1.016 (\ln T - m \ln 2)$$

t statistics (-24.14) (94.72) or,

$$G_m = (.0631T)^{\frac{1}{.9843}}\left[\frac{1}{(2^{m-1})^{1/.9843}} - \frac{1}{(2^m)^{1/.9843}}\right]$$

An R^2 value of .98 was computed, and the F-statistic with one and 210 degrees of freedom is 8970, surpassing the critical value of 6.7 at the 0.01 significance level. This means (loosely) that an F-statistic value greater than 6.7 is 99% assured of being a good estimator. The t-statistics also surpass their critical value of 2.326.

Storage Requirements

This model gives much information on the size of the individual files comprising the inverted file (Fedorowicz [1981]). The Index File will have approximately ($\Sigma_m G_m$) terms in it. The length of the Postings File can be estimated by

$$\sum_m G_m x \text{ (the number of occurrences in } G_m),$$

where the mean of G_m is computed as

$$\text{MEAN}(m) = \frac{\frac{2^{m-1}}{2^{(m-1)/\alpha}} + \left[\sum_{j=2^{m-1}+1}^{2^m-1} \frac{1}{j^{1/\alpha}}\right] - \frac{2^m-1}{2^{m/\alpha}}}{\frac{1}{2^{(m-1)/\alpha}} - \frac{1}{2^{m/\alpha}}}$$

The Header File, which is essentially a list of bibliographic citations, will be of length T.

The next step is to establish the relationship between these estimates and the physical storage requirements (in blocks) of the inverted file elements. These results have been reported in Fedorowicz (1981).

The Access Time Model

The access time model to be summarized here requires the use of a good deal of terminology, which is described in Table 1. All pertinent definitions are found in the table.

The Index File blocks referred to in the first six steps of the model are the files that were developed in the Storage Requirements model. The following describes the access time model for a three layer inverted file structure. Adaptation of the model to a larger file structure is easily realized.

Step 1: Read Master File Index

Time requirement: T_t

The total time required is the time it takes to access a single block of data.

Step 2: Search Master File Index block for the location of the term of interest in the secondary Index File.[2]

$$T_c * \left\lceil \log_2 \left\lceil \frac{NKEYS}{BFI*BF12} \right\rceil \right\rceil$$

TABLE 1

DEFINITIONS OF TERMS USED IN THE ACCESS TIME AND RESPONSE TIME MODELS

ACCTM	Access time (minimum time; one user on system)
AND_i	0-1 variable describing whether the i^{th} Boolean operation is an AND
BFI	Blocking factor for the Index File
BFI2	Blocking factor for the Master Index Files
BFP	Blocking factor for the Postings File
BLOCKC	Block length in characters
CPUTIME	Amount of CPU time used in the search for the search procedure only
EST-HITS	Estimated number of hits for a particular search statement
HITAND	Number of hits expected when two blocks of citations are ANDed together
HITOR	Number of hits expected when two blocks of citations are ORed together
IOCNT	Number of I/O's required for a search
NKEYS	Number of access keys in the data base
NUMCOMP	Expected number of comparisons for a search
NUMHITS	Expected number of hits from a search
NUMKEY	Number of keys in the search statement
NREC	Total number of records in the data base
OR_i	0-1 variable stipulating whether the i^{th} Boolean operation is an OR
P_i	Number of postings for the i^{th} key
RECAVE	Average length of a record (in characters)
SRI(j)	Storage Requirements in blocks for the j^{th} Index File layer
T_c	Average time to compare an access key to one block
T_i	Average time to merge (OR) two blocks of record numbers
T_t	Average time to access a block
X_d	Average number of header file blocks accessed per search
X_1	Number of SEARCHERS equivalent to one OTHER user
X_2	Amount of response time required per I/O
σ	Optimization factor when intersecting two blocks of Boolean operation AND (e.g., extreme values of list are examined; if value exceeded, then there is no need to continue search at that point).

This equation says that the total time to search a Master File Index block is a function of the number of search keys which are stored on the block. The number of search keys on the Master File Index is equal to the number of blocks on which the search keys for the Secondary Index blocks are found. This in turn is equal to the number of blocks on which the citation data is stored. The blocking factors reflect how many records are recorded on each block.

Step 3: Read secondary Index Block.

$$T_t$$

Step 4: Search secondary Index block for appropriate final Index block.

$$T_c * \left\lceil \log_2 SRI(3) \right\rceil$$

The storage requirements for the data portion of the citation database are equal to SRI(3), which is computed as $\frac{\sum_m G_m}{BFI}$

Step 5: Read Index File block.

$$T_t$$

Step 6: Search Index File block for key of interest.

$$T_c * \left\lceil \log_2 BFI \right\rceil$$

This search is done on a single block which has BFI records stored on it.

NOTE: Steps 1, 3 and 5 can be omitted if the Index Files reside in main memory.

Step 7: Repeat steps 1 through 6 for all terms in the search statement.

Step 8: Read, merge (OR), and intersect (AND) intermediate lists of record numbers.

$$\frac{NUMCOMP}{BFP} * T_i$$

where the expected number of comparisons, NUMCOMP, is computed as:

for two terms:

$$NUMCOMP = (P_1 + P_2 - 1)\ (OR_1 + \sigma\ AND_1)$$

for three terms:

Let $P_1 \leq P_2 \leq P_3$

$$EST\text{-}HITS_2 = (P_1 + P_2)\ (HITOR)\ (OR_1) + (P_1)\ (1 + \frac{P_2 - P_1}{NREC})\ (HITAND)\ (AND_1)$$

[2] $\lceil X \rceil$ is read the smallest integer greater than or equal to X.

$NUMCOMP = (p_1 + P_2 - 1)\ (OR_1 + \sigma\ AND_1) + (P_3 + EST\text{-}HITS_2 - 1)\ (OR_2 + \sigma\ AND_2)$

$EST\text{-}HITS_2$ incorporates the expected hit rate for 2-term searches, with the first half of the equation in effect for a merging of two lists of search keys and the second half if it is an intersection of keys. The term $\frac{P2-P1}{NREC}$ reflects any large disparities in the length of the two lists in the comparison. The estimated number of hits is then used to compute the number of comparisons in the 3-term query.

for four terms: Let $P_1 \leq P_2 \leq P_3 \leq P_4$

$$y = \min\ \{EST\text{-}HITS_2,\ P_3\}$$

$$\bar{y} = \max\ \{EST\text{-}HITS_2,\ P_3\}$$

$$EST\text{-}HITS_3 = (P_3 + EST\text{-}HITS_2)\ (HITOR)\ (OR_2) + (y)\ \left(1 + \frac{\bar{y} - y}{NREC}\right)\ (HITAND)\ (AND_2)$$

$$NUMCOMP = (P_1 + P_2 - 1)\ (OR_1 + \sigma\ AND_1) + (P_3 + EST\text{-}HITS_2 - 1)\ (OR_2 + \sigma\ AND_2) + (P_4 + EST\text{-}HITS_3 - 1)\ (OR_3 + \sigma\ AND_3)$$

and so on.

Step 9: Read data blocks.

$$T_t * X_d$$

$$\text{where } X_d = \left\lceil \frac{RECAVE}{BLOCKC} \right\rceil * NUMHITS$$

X_d will give an indication of the number of blocks that can be expected to be accessed for a particular search. Step 9 is not necessary when responding to the type of requests looking for only the total number of citations pertinent to a particular query. The summation of steps 1 through 9 gives an estimation of the access time as

$$ACCTM = (3*NUMKEY + (\frac{NUMCOMP}{BFP}) + X_d) * T_t + (NUMKEY * (\lceil \log_2 \lceil \frac{NKEYS}{BFI*BF12} \rceil \rceil + \lceil \log_2 SRI(3) \rceil + \lceil \log_2 BFI \rceil)) * T_c + (\frac{NUMCOMP}{BFP}) * T_i \quad (7)$$

ACCTM is an approximation of the minimum CPU time and I/O time required for a search. In order for the model to prove useful in a typical computing environment, the additional time requirements of time-sharing in an online environment are incorporated. This work is described in Fedorowicz (1983c).

The data for the access time models were collected for a four-day period in August, 1979. All data were made available by the National Library of Medicine. The estimates for the other model parameters were obtained from a number of sources. Some parameters were derived from IBM specifications since all computation and storage devices at NLM are produced by that company. Parameters that are system bound were estimated with NLM data. A complete description of the data collection procedures is given in Fedorowicz (1982b,1983). The estimates for the model parameters and their sources and statistics can be found in Table 2. The results of the access time model itself can be found in Table 3, in the column labeled "Predicted Access time," along with the actual parameter values. A scatter plot of the actual and predicted values is depicted in Figure 1. The access time data was obtained by performing the actual MEDLINE searches in a mode that allowed both the CPU time and I/O's required by the search to be recorded as if the searcher was the only MEDLINE user on the system at that time. The twenty-nine searches were repeated once for this phase of the experiment, so that search numbers 1 and 30 in Table 3 represent the same search statement. The access time model tested is represented by the following equation:

Access time = IOCNT * T_t + (number of Index File Blocks accessed) * T_c + (NUMCOMP due to OR operations) * T_I + (NUMCOMP due to AND operations) * T_I

where $\sigma T_I < T_I$, since there are some checks built into the AND comparison time (e.g., if the record number at the top of list 1 exceeds the record number at the bottom of list 2, then it is obvious that no overlap exists between the two lists). NUMCOMP is equal to $NUMCOMP_{AND} + NUMCOMP_{OR}$.

The number of index files accessed is constant for all searches in a given month which contain the same number of terms, and is equal to

$$NUMST * \left\lceil \log \left\lceil \frac{NTERMS}{2 \lceil BFI*BF12 \rceil} \right\rceil \right\rceil + \lceil \log_2 SRI(3) \rceil + \lceil \log_2 BFI \rceil$$

TABLE 2

ACCESS TIME MODEL PARAMETER ESTIMATE

Parameter	Estimate	Estimate Sources	T-statistic	Standard Error
HITAND	.96	sample searches	-	-
HITOR	.105	sample searches	-	-
RECAVE	931 char.	supplied by NLM	-	-
NCIT	512,444	supplied by NLM	-	-
T_t	.0095	sample searches	8.60	.0011
t_c	.0095	derived from data other than sample searches	4.33	.0004
T_i	.00000384	sample searches	14.75	.00000026
σT_i	.00000115	sample searches	7.56	.00000015

TABLE 3

ACCESS TIME MODEL - ACTUAL AND PREDICTED SEARCH RESULTS

Search Number	IOCNT	NUMCOMP (AND)	NUMCOMP (OR)	Actual* Access time	Predicted* Across Time
1	10	0	3648	.2500	.1890
2	11	0	6252	.2490	.2084
3	10	0	820	.2400	.1781
4	40	0	134574	.7490	.9754
5	22	36948	0	.4410	.3309
6	13	14481	0	.2930	.2200
7	12	3996	0	.2500	.1985
8	37	222492	0	.7080	.6858
9	22	110690	0	.5350	.4155
10	13	3617	0	.2800	.2075
11	12	0	3698	.2800	.2081
12	103	843329	0	1.9320	2.0227
13	54	360466	5333	1.0720	1.0254
14	84	661523	23162	1.6040	1.7232
15	19	20298	2163	.4060	.2917
16	32	163245	2375	.6410	.5796
17	28	78073	4402	.5040	.4518
18	68	466464	367292	3.0880	2.6686
19	24	0	44865	.4610	.4796
20	39	0	108630	.7690	.8663
21	37	22474	47183	.7670	.6374
22	97	743561	23802	1.9470	1.9428
23	107	743561	23802	1.9560	2.0374
24	37	0	139188	.8050	.9647
25	22	0	9632	.4920	.3255
26	40	76979	49581	.9020	.7375
27	51	0	254952	1.2190	1.5415
28	28	52023	2450	.5290	.4144
29	30	52023	2450	.6220	.4333
30	10	0	3648	.2990	.1890
31	11	0	6252	.2860	.2084
32	10	0	820	.2710	.1781
33	40	0	134574	.7610	.9754
34	22	36948	0	.4330	.3309
35	13	14481	0	.2860	.2200
36	12	3996	0	.2990	.1985
37	37	222492	0	.7250	.6858
38	22	110690	0	.5150	.4155
39	13	3617	0	.2760	.2075
40	12	0	3698	.2750	.2081
41	103	843329	0	1.9290	2.0227
42	54	360466	5333	1.0220	1.0254
43	84	661523	23162	1.6420	1.7232
44	21	20298	2163	.4570	.3107
45	32	163245	2375	.6310	.5796
46	24	78073	4402	.4440	.4139
47	68	466464	367292	2.9760	2.6686
48	24	0	44865	.5500	.4796
49	40	0	108630	.8310	.8758
50	35	22474	47183	.7200	.6184
51	98	743561	23802	1.8330	1.9522
52	105	743561	23802	1.9380	2.0185
53	37	0	139188	.7970	.9647
54	22	0	9632	.4940	.3255
55	44	76979	49581	.9230	.7754
56	50	0	254952	1.1790	1.5320
57	27	52023	2450	.5080	.4049
58	30	52023	2450	.6250	.4333

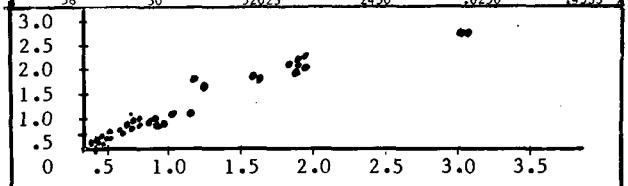

Figure 1: Actual vs. Predicted Search Results

The access time model results (equation 7) are, then,

$$\text{Access time} = .0095\ X_1 + .0015X_2 + .00000384X_3 + .00000115X_4$$

where

$$X_1 = 3*\text{NUMST} + \left(\frac{\text{NUMCOMP}}{\text{BFP}}\right) + X_d$$

$$X_2 = \text{NUMST}*\left\lceil \log_2 \left\lceil \frac{\text{NKEYS}}{\text{BFI}*\text{BFI2}} \right\rceil \right\rceil + \lceil \log_2 \text{SRI}(3) \rceil + \lceil \log_2 \text{BFI} \rceil$$

*These figures include the CPU time required to read and write data on the disk, as well as that required to manipulate the data for the search.

$$X_3 = \frac{\text{NUMCOMP}_{OR}}{\text{BFP}}$$

$$X_4 = \frac{\text{NUMCOMP}_{AND}}{\text{BFP}}$$

Values for X_1 were captured in the data collection process. X_2, X_3 and X_4 were computed from the values in Tables 2 and 3. The model produced an F-ratio of 574 with 2 and 55 degrees of freedom, surpassing the critical value of 5.08 at the 0.01 significance level. It had an R^2 value of .95.

In order to show the increased accuracy of this access time model over a similar one which assumes a uniform distribution of term occurrence, the following examples have been constructed. If a uniform distribution of term postings is assumed for the samples searches, all 2-term searches could be expected to be completed in approximately .2114 sec, for an OR operation, or .1845 sec for an AND operation. Four term searches would take .3091 sec for three OR operations, or .2991 sec for three AND operations. The actual times recorded for two term searches range from .240 to .761 sec, and four-term searches ranged from .406 to 3.088 sec. Of course, these searches generally are constructed from terms with more than the average 56.3 postings, which explains the discrepancies on the low end of the estimates. Nonetheless, the point must be made that the use of the actual posting counts, and the projections made possible by the use of the Zipf parameters, make a substantial difference in the access time predictions.

Summary and Implications

It is shown that multiple regression techniques can be applied to the area of database evaluation to determine the file characteristics and storage requirements of a database. These database characteristics are then used to estimate the access time required to respond to an individual user's request for information retrieval.

The study summarized in this paper constitutes a model of MEDLINE, the automatic bibliographic system of the National Library of Medicine. In an automatic bibliographic system, most elements in a citation can be used to search a database comprised of journal citations. Retrieval is accomplished by means of a search of an inverted file which comprises an Index File, a Postings File and a Header File. The relationship between the Index File and the Postings File is analogous to that of the Zipfian relationship which associates the number of different search terms with their frequency of occurrence, based on the length of the text, or the total number of word occurrences in the Header File.

A Zipfian model of term dispersion is applied to MEDLINE, and a regression formulation of that model is presented, which is shown to

have a high degree of accuracy. The model of inverted file access time is then presented, and again the reported results are good.

Well-accepted regression techniques are employed to test the models, with the usual assumptions of variable independence and normality. Logarithmic transformations are applied in the case of the Zipfian model of the database. The access time model can be tested in its initial formulation by substituting single variables for the composite ones of the original model.

Multiple regression techniques should be useful in other areas of computer and information science research as well. The literature reflects a lack of expertise among computer researchers, as exhibited through a dearth of articles taking advantage of the sophisticated modeling capabilities of these techniques, and by the naive (and sometimes inaccurate) applications which have reached publication (Cale, et al. [1979], and the correction in Fedorowicz [1981]). The two-page summary of regression techniques in this paper give a flavor for the types of modeling possible. Interested researchers are advised to pursue further study before trying their hand in the area.

References

Allen, A.O., "Queueing Models of Computer Systems," Computer, Vol. 13, No. 4, April, 1980, pp. 13-24.

Borovits, I. and Neumann, S., Computer Systems Performance Evaluation, D.C. Heath and Co., Lexington, MA 1979.

Cale, E.G., Gremillon, L.L., and McKenney, J.L., "Price Performance Patterns of U.S. Computer Systems", Communications of the ACM, April, 1979, Vol. 22, No. 4, pp. 225-233.

Fedorowicz, J.E., "The Theoretical Foundation of Zipf's Law and Its Application to the Bibliographic Data Base Environment", Journal of the American Society for Information Science, Vol. 33, No. 5, September 1982(a), pp. 284-292.

Fedorowicz, J.E., "A Zipfian Model of An Automatic Bibliographic System: An Application to MEDLINE," Journal of the American Society for Information Science" Vol. 33, No. 4, July, 1982(b), pp. 223-232.

Fedorowicz, J.E., "Data Base Performance Evaluation in an Inverted File Environment," submitted to the Transactions on Database Systems, 1983.

Fedorowicz, J.E., "Comments On Price/Performance Patterns of U.S. Computer Systems", Communications of the ACM, September, 1981, Vol. 24, No. 9, pp. 585-586.

Foster, D.V., McGehearty, P.F., Sauer, C.H. and Waggoner, C.N., "A Language for Analysis of Queueing Models," in Proceedings Fifth Annual Pittsburgh Modeling and Simulation Conference, 1974, pp. 381-386.

Johnston, J., Econometric Methods, 2nd edition, McGraw-Hill, New York, 1972.

Muntz, R.R., "Queueing Networks: A Critique of the State of the Art and Directions for the Future," Computing Surveys, Vol. 10, No. 3, September 1978, pp. 353-359.

Nguyen, H.C., Ockene, A., Revell, R. and Skwish, W.J., "The Role of Detailed Simulation in Capacity Planning," IBM Systems Journal, Vol. 19, No. 1, 1980, pp. 81-101.

Reiser, M. and Sauer, C.H., "Queueing Network Models: Methods of Solution and Their Program Implementation," in Current Trends in Programming Methodology, Vol. III: Software Modeling and Its Impact on Performance, K.M. Chandy and R. T. Yeh (eds.), Prentice-Hall, Inc., Englewood-Cliffs, N. J., 1978, pp. 115-167.

Rose, C.A., "A Measurement Procedure for Queueing Network Models of Computer Systems," Computing Surveys, Vol. 10, No. 3, September, 1978, pp. 263-280.

Seaman, P.H., "Modeling Considerations for Predicting Performance of CICS/VS Systems," IBM Systems Journal, Vol. 19, No. 1, 1980, pp. 68-80.

Zipf, G.K., Human Behavior and the Principle of Least Effort, Cambridge, Addison-Wesley, 1949.