Correspondence

Partitioning a Sample Using Binary-Type Questions with Ternary Feedback

Amit Cohen, Moshe Kam, and Robert Conn

Abstract— **The problem is to find the largest observation (or the q largest observations) in a random sample of size n by asking binary-type questions of people (or items) in the sample. At each stage of the search, a threshold is calculated and a binary-type question is posed to each member of the sample. This question is of the form "Is your observation greater than the threshold?"**

The threshold is determined from answers given to the previous questions, and no exact data is ever collected, i.e., no member is asked to explicitly provide his observation. Arrow, Pesotchinsky, and Sobel (APS) calculated the optimal threshold sequence for two different objectives: i) minimize the average number of questions required for a solution, and ii) maximize the probability of solving the problem in, at most, r questions. APS have assumed that the number of respondents in the affirmative at each stage of the search is exactly known (this is the case of "full feedback"). There exist applications (e.g., multiuser communications) where the number of affirmative answers is only known to be one member of the set {0, 1, more than 1} ("ternary feedback"). For these applications, we calculate exactly the optimal thresholds, in the sense of maximizing the probability of getting precisely one affirmative answer to the next binary question. We find that, with ternary feedback, the expected number of questions needed to terminate the search is ~2.445 (for large n). Using the same objective function, APS calculated 2.44144 for full feedback.

Application of the threshold-calculation procedure is demonstrated in resolution of packet collisions over multiuser communication channel. At any instant, the procedure maximizes the probability of successful packet transmission during the next time slot.

I. Introduction

In 1981, Arrow, Pesotchinsky, and Sobel (APS) [2] posed and solved the following problem. Let there be n members of a group, for example, users of a communications network or manufacturers of a product. Each member of the group possesses a real observation, $x(k)$ for the kth member. The observations are independently and identically distributed (iid) (or at least exchangeable) with cumulative distribution function (cdf) $F(x)$ which is known. Since the observations are continuous, with probability 1 no two observations are exactly equal. It is desired to find the holder of the largest observation, using only binary-type questions, addressed to each member of a subgroup of the original group. The ith question will be addressed to n_i members, with $n_1 = n$. The questions are designed such that, after the $(i-1)^{st}$ question, we possess knowledge of an interval $[\alpha_i, \beta_i]$ which contains the largest observation, and of n_i, the number of members whose observations lie in that interval. APS have shown that, with a known $F(x)$, the problem can always be reduced to a corresponding one with uniform (0, 1) distribution. We shall therefore assume a uniform (0, 1) distribution in the sequel.

One possible objective is to design the questions such that the *expected length of the search* (= number of questions needed to find the holder of the largest observation) is minimized. Another possible objective is to maximize the *probability of terminating the search in the next step* (= finding the holder of the largest observation after the next question).[1] Anantharam and Varaiya [1] have shown that, for both objective functions, a question of the type "Is $x(k) > c[\alpha_i, \beta_i, n_i]$?" is optimal at the ith stage. In other words, inquiries about belonging to a right-hand interval of the observation space are as good as inquiries about belonging to any other subset of the observation space.

The results obtained by APS are striking. The minimal expected number of questions required to find the holder of the largest observation in a sample is less than 2.427 78. This result holds for any starting sample size n and for any cdf $F(x)$. The procedure that maximizes the probability of terminating in the very next question has a numerically-close answer, namely the expected value of the number of questions required is less than 2.441 44. Both APS procedures were devised for *full feedback*, namely, at each stage it is known exactly how many members answered the last binary question in the affirmative (n and n_i are known).

We proceed to review the APS procedures and then extend them to the case of *ternary feedback*. In this case, the number of members who answered each binary question in the affirmative is known only as 0 (no respondents in the affirmative), 1, or *many* (i.e., more than 1).

II. Summary of the APS Procedures

Let the question asked in the ith stage of the search for the largest observation be "Is your observation greater than $c[\alpha_i, \beta_i, k_i]$?" Here, $[\alpha_i, \beta_i]$ is the interval where the largest observation is known to reside. The original cdf has been reduced to a uniform cdf over (0, 1). Therefore, at stage i of the search, the observations that reside in $[\alpha_i, \beta_i]$ are also uniformly distributed. Their distribution can, in turn, be reduced to a uniform (0, 1) cdf. With $s = k_i$, let $\pi_{s,j}$ denote the probability that j elements out of s have observations greater than $c[\alpha_i, \beta_i, s]$. Then, $\pi_{s,j} = \binom{s}{j} p^j (1-p)^{s-j}$, where $p = 1 - F(c[\alpha_i, \beta_i, s])$ $(= 1 - c[0, 1, s]$ for uniform (0, 1) cdf).

The expected number of questions needed for finding the largest observation in a sample of n observations is given by

$$\begin{aligned}\mu_n(p_n) &= 1 + \pi_{n,0}\mu_n(p_n) + \sum_{j=2}^{n} \pi_{n,j}\mu_j(p_j) \\ &= 1 + (1-p_n)^n \mu_n(p_n) \\ &\quad + \sum_{j=2}^{n} \binom{n}{j} {p_n}^j (1-p_n)^{n-j} \mu_j(p_j). \end{aligned} \tag{1}$$

In expression (1), $\mu_k(p_k)$ is the expected number of questions needed to terminate the search starting with k members in the group. From (1) we obtain

$$\begin{aligned}\mu_n(p_n) &= \frac{1 + \sum_{j=2}^{n-1} \pi_{n,j}\mu_j(p_n)}{1 - \pi_{n,0} - \pi_{n,n}} \\ &= \frac{1 + \sum_{j=2}^{n-1} \binom{n}{j} {p_n}^j (1-p_n)^{n-j} \mu_j(p_j)}{1 - {p_n}^n - (1-p_n)^n}. \end{aligned} \tag{2}$$

Manuscript received February 13, 1994; revised November 18, 1994. This work was supported by the National Science Foundation under Grant ECS 9057587.

The authors are with the Department of Electrical and Computer Engineering, Data Fusion Laboratory, Drexel University, Philadelphia, PA 19104 USA.

IEEE Log Number 9413265.

[1] It is straightforward to generalize to holders of the q *largest observations* $(q > 1)$ and to maximize the probability of terminating the search in *at most* r *questions* $(r > 1)$.

TABLE I
EXPECTED LENGTH OF SEARCH

Original group size f[0,1]	Full Feedback — PI: average search length — *calculation*	Full Feedback — PI: maximum probability of termination in one step — *calculation*	Ternary Feedback — PI: maximum probability of termination in one step — *simulation*
(1)	(2)	(3)	(4)
2	2	2	2
3	2.16507445	2.16666667	2.174
4	2.23783259	2.24137931	2.248
5	2.27895657	2.28404488	2.292
6	2.30542104	2.31168886	2.323
7	2.32388612	2.33107003	2.344
8	2.33750509	2.34541601	2.360
9	2.34796545	2.35646532	2.369
10	2.35625233	2.36523811	2.378
11	2.36297962	2.37237268	2.385
12	2.36854986	2.37828896	2.392
13	2.37323799	2.38327461	2.397
14	2.37723821	2.38753329	2.401
15	2.38069162	2.39121324	2.406
16	2.38370318	2.39442497	2.409
17	2.38635260	2.39725252	2.411
18	2.38870151	2.39976094	2.413
19	2.39079829	2.40200139	2.415
20	2.39268148	2.40401463	2.418

A sufficient condition for $\mu_n(p_n)$ to be finite is that the binomial probabilities $\pi_{n,0}$ and $\pi_{n,n}$ be bounded away from zero.

A. Minimizing the Average Number of Questions

To find p_n (and the corresponding threshold) that will minimize the average number of questions, APS proposed to view (2) as a recurrence relation, and minimize $\mu_n(p_n)$ at each step. A table of the resulting optimal $p_n{}^{(q)}$ (for finding the q largest observations with n observations in the interval) appears in [2, p. 406] and can be generated easily using a numerical minimization routine. The resulting optimal average numbers of questions needed to terminate the search appear (for $q = 1$ and $n = 1, 2, \cdots, 20$) in Table I, column 2.[2] As $n \to \infty$, the average number of questions using the APS procedure tends to be 2.427 78.

B. Maximizing the Probability of Terminating the Search in the Next Step

For maximizing the termination-probability in the next step, APS have found that optimally $p_n{}^{(q)} = \frac{q}{n}$. Consequently, for k_i uniformly distributed observations over $[\alpha_i, \beta_i]$, the optimal threshold is $c[\alpha_i, \beta_i, k_i] = \frac{k_i - 1}{k_i}\beta_i + \frac{1}{k_i}\alpha_i$. The corresponding average numbers of questions needed to terminate the search (for $q = 1$ and $n = 1, 2, \cdots, 20$) appear in Table I, column 3.[2] We note that the procedure for maximizing the probability of one-step termination is much simpler than the procedure needed for minimizing the average search length. The price is a very small increase in the average length of the search (compare columns 2 and 3 in Table I). As $n \to \infty$, the average number of questions using the simpler procedure tends to be 2.441 44.

Example: We are looking for the largest observation with $n = 5$. The observations are independently and uniformly distributed on $[10, 50]$. For the first question, we choose $p = \frac{1}{5}$, and get $c[10, 50, 5] = \frac{4}{5}50 + \frac{1}{5}10 = 42$. The probability of success with this threshold is $\pi_{5,1} = 5p(1-p)^4$, which is 0.4096. Suppose that four elements had observations that are greater than 42 (the probability of that event is $\binom{5}{4}0.2^4 0.8 = 0.0064$). We shall design $c[42, 50, 4]$ so as to maximize the probability of finding the largest observation in the next question. This leads to $c[42, 50, 4] = 48$. We shall continue to calculate new thresholds until we get a single affirmative answer.

[2] Additional entries for $q > 1$ can be found in [2, p. 406].

III. MAXIMIZING PROBABILITY OF SEARCH-TERMINATION IN THE NEXT STEP WITH TERNARY FEEDBACK

There exist applications (especially in multiuser communications [5]) where the number of respondents that answered in the affirmative to our binary question is not known exactly. We know only that it was "none," "exactly one," or "more than one." This is the case of *ternary feedback*, which has been studied extensively in the context of random-access communications (see, e.g., [3, pp. 289–304], [6]–[8]). An optimal solution with ternary feedback requires modifications of the original APS procedures.

Again, we use a cdf $F(x)$ which is uniform on (0, 1). For simplicity, we replace the notation $c[\alpha_i, \beta_i, k_i]$ by c_i. At each stage of the search, an interval $(t, w]$ is known to contain more than one observation, while $(w, 1]$ is empty. Initially, $t = 0$ and $w = 1$. If the problem is not solved after the first question, then either $t = 0$ and $w = c_1$, or $t = c_1$ and $w = 1$. Let $f[\alpha, \beta]$ designate the number of observations in the interval $[\alpha, \beta]$. It is easy to show that

$$\begin{aligned}\mathrm{PR}\{f[t,w] = i \mid f[t,w] > 1,\ f[\gamma,w] > 1,\ t > \gamma\} \\ = \mathrm{PR}\{f[t,w] = i \mid f[t,w] > 1\}.\end{aligned} \tag{3}$$

In other words, only the smallest known right-hand interval that contains more than one observation is of interest, not the history of arriving there. With n members initially in the group and with $u \in (t, w)$, let the next question be "Is your observation greater than u?" In that case, the probability of terminating after the next question is

$$\begin{aligned}\mathrm{PR}\{f[u,w] = 1 \mid f[t,w] > 1] \\ &= \frac{\mathrm{PR}\{f[u,w] = 1\}}{\mathrm{PR}\{f[t,w] > 1\}} \\ &\quad \times (1 - \mathrm{PR}\{f[0,t] = n-1 \mid f[0,u] = n-1\}) \\ &= \frac{n\left(\frac{u}{w}\right)^{n-1}\left(1 - \frac{u}{w}\right)\left[1 - \left(\frac{t}{u}\right)^{n-1}\right]}{\mathrm{PR}\{f[t,w] > 1\}}.\end{aligned} \tag{4}$$

Let $r(u) = u^{n-1}\left(1 - \frac{u}{w}\right)(1 - t^{n-1}u^{1-n})$. A necessary condition for maximizing (4) is $\frac{dr}{du} = 0$, which, after some algebra, reduces to solving

$$r_d(u) = w(n-1)u^{n-2} - nu^{n-1} + t^{n-1} = 0. \tag{5}$$

Thus, at step i we calculate u using (5), and set $c_i = u$. The first question (with $t = 0$ and $w = 1$) always uses $c_1 = \frac{n-1}{n}$, as does the full-feedback APS procedure. If $n = 2$, and if the number of affirmative answers to the first question was greater than 1, then $c_2 = \frac{3}{4}$, viz., we divide the interval $(\frac{1}{2}, 1]$, which contains two observations, exactly in the middle. If $n = 3$, under the same conditions, $c_2 = \frac{1}{3} + \frac{1}{3}\sqrt{\frac{7}{3}} = 0.8425\ldots$. Note that for $n = 3$, we do not know whether two or three members of the group have answered the first question in the affirmative. Had we known that, we would have used $c_2 = \frac{5}{6}$ (for two respondents) or $c_2 = \frac{8}{9}$ (for three respondents). The optimal value that we used, knowing only that 'more than one' affirmative answers were given ($c_2 = 0.8425\ldots$), is

much closer to the optimum for two respondents than to the optimum for three. Indeed, it is more likely that when more than one affirmative answer was given, it was due to two (not three) respondents. The proximity of the calculated value of c_2 to $\frac{5}{6}$ reflects this fact.

Numerical methods to maximize $r(u)$, or to find the roots of $r_d(u)$, are expected to be very effective, since the solution to (5) (for $i > 1$) is known to be close to $\frac{t+w}{2}$ in the interval $(t, w]$.

Expected Number of Questions: Column 2 of Table I shows the APS average number of questions required to terminate the search under the following conditions: i) full feedback is available, and ii) the objective is to minimize the expected number of questions. Column 3 shows the full-feedback APS average when the probability to terminate in the next step is maximized. Column 4 shows the corresponding ternary-feedback average, as obtained, using (5), from Monte–Carlo simulations. For the same original group size ($f[0, 1]$), the numbers are very close to each other. Very little is lost by changing the objective function from average number of questions to probability of termination in the next step. Only a little more is lost when moving from full feedback to ternary feedback.

IV. Application to Multiuser Communications

We consider a single communication channel that serves a large number of users [5]. Each user receives messages for transmission from a source outside the channel. The times of reception of these messages, known as *arrival times*, are assumed statistically independent, and identically-distributed according to a Poisson distribution. The mean of the overall arrival process, known as the *arrival rate*, is λ. The duration of time needed for transmission of a message is assumed to be fixed, and serves as the time unit. Users are allowed to begin transmission of a message only at integer times, $t = 0, 1, 2, \ldots$ and are assumed to be synchronized. The time intervals $(0, 1], (1, 2], (2, 3], \ldots$ are referred to as *slots* (time slot i is the interval $(i, i + 1]$), and the channel is referred to as a *slotted channel*. If, during a certain slot, more than one user transmits a message, a conflict, or *collision* is said to have occurred. In this case, the colliding messages have to be retransmitted at some future time. A collision is considered *resolved* once all messages involved in it were successfully transmitted. The *collision-resolution interval* (CRI) is the epoch from the time of an initial collision to the time of its resolution. The *throughput* of the channel is the fraction of time during which successful transmissions occur. The *maximum stable throughput* is the maximum throughput that a channel can achieve, as long as the expected delay time between message arrival and its successful transmission is finite. The arrival process implies that the probability of k arrivals during an interval $[t_1, t_2]$ is

$$\mathrm{PR}\{k \text{ messages arrived during } [t_1, t_2]\} = e^{-\lambda(t_2-t_1)}\frac{[\lambda(t_2-t_1)]^k}{k!}. \tag{6}$$

At the beginning of the i^{th} time slot (which occurs at time i), we would like to allow all messages that have arrived during a certain interval $(t_1, \tau]$ (such that $t_1 < \tau \leq i$) to be transmitted. Let $f[t_1, \tau]$ be the number of messages that have arrived during the interval $(t_1, \tau]$. Then, due to the ternary feedback, we shall get either $f[t_1, \tau] = 0$, $f[t_1, \tau] = 1$, or $f[t_1, \tau] = e$ (the latter indicates all cases when more than one message has been transmitted during $(t_1, \tau]$, i.e., a collision has occurred).

Based on previous transmission attempts and past questions, at the beginning of the ith time slot the channel is in one of three states:

i) *Latest collision resolved*: there exists a time $t < i$ such that the last collision that has occurred was resolved (by at least two successful transmissions of messages) prior to time t. In this case, no prior knowledge exists about the number of actual arrivals in $(t, i]$.

ii) *Partially resolved collision*: there exists a time $t < i$ such that the most recent collision occurred for messages that arrived in the interval $(r, s]$ where $r < t < s < i$, and such that, following the collision, all the messages that have arrived during $(w, t]$, for some $r \leq w < t$, were allowed to be transmitted; a single successful transmission has occurred in $(w, t]$. In this case $f[t, s] > 0$.

iii) *Unresolved collision*: there exists an interval $(t, s]$ $t < s < i$ such that the most recent collision has occurred between messages that arrived in $(t, s]$, and no successful transmissions followed this collision. Hence, $f[t, s] > 1$.

In case i), we shall place τ such that $t_1 = t < \tau \leq i$.
In cases ii) and iii), we shall place τ such that $t_1 = t < \tau \leq s$.
For case i), the placement of τ will be determined by maximizing the unconditional probability that $f[t, \tau] = 1$, namely

$$\mathrm{PR}\{f[t, \tau] = 1\} = e^{-\lambda(\tau-t)}\lambda(\tau - t). \tag{7}$$

Not surprisingly, the maximum of (7) occurs at[3]

$$\tau = t + \frac{1}{\lambda}. \tag{8}$$

In case ii), we maximize (over τ) the probability

$$\mathrm{PR}\{f[t, \tau] = 1 \mid f[t, s] > 0\} \tag{9}$$

and find that, if the length of the interval $[t, s]$ is greater than $\frac{1}{\lambda}$, the maximum is obtained at

$$\tau = t + \frac{1}{\lambda}. \tag{10a}$$

Otherwise, the maximum is found at

$$\tau = s. \tag{10b}$$

In case iii), we shall maximize (over τ) the probability

$$\mathrm{PR}\{f[t, \tau] = 1 \mid f[t, s] > 1\} = \frac{\mathrm{PR}\{f[\tau, s] > 0\}\mathrm{PR}\{f[t, \tau] = 1\}}{\mathrm{PR}\{f[t, s] > 1\}} = \frac{\lambda e^{\lambda t}[e^{-\lambda\tau} - e^{-\lambda s}](\tau - t)}{\mathrm{PR}\{f[t, s] > 1\}}. \tag{11}$$

The maximum occurs at

$$\tau = t + \frac{1 - e^{-\lambda(s-\tau)}}{\lambda} \tag{12a}$$

with a solution for the optimum τ at

$$\tau = t + \frac{1 - W\left(e^{1+\lambda(t-s)}\right)}{\lambda} \tag{12b}$$

and where $W(\cdot)$ is Lambert's W function [4]. For $\lambda > 0$ and $s > t$, the value of $W\left(e^{1+\lambda(t-s)}\right)$ is restricted to be in the interval (0, 1). We note that, unless we deviate from the optimal rules, the condition $s - t > \frac{1}{\lambda}$ will never occur, and therefore (10a) will never be applied.

Suppose there are no unresolved collisions at time 0. At the end of time slot i, stations possess the previous channel feedback, $f[t_1, c_i]$, and a time, $T_c = c_k$, where k is the most recent time slot during

[3]However, λ may not be explicitly known; in that case, λ can be either estimated from the arrival rate of past events, or assumed to possess a certain value on the basis of an optimization criterion. Often, the *maximum stable throughput* of the algorithm (if known) is used for λ.

which a collision has occurred. Each station executes the following steps:

0) Initialization: $i \leftarrow 0$; $c_0 \leftarrow 0$; $T_c \leftarrow (-1)$.
1) If a message is waiting for transmission, and its arrival time is less than c_i, transmit during the current slot.
2) At the end of the slot, obtain the channel feedback, $f[t_1, c_i] \in \{0, 1, e\}$.

   a) If $f[t_1, c_i] \neq e$, and $T_c < t_1, c_{i+1} \leftarrow \min[c_i + \frac{1}{\lambda}, i+1]$.
   b) If $f[t_1, c_i] = 0$ and $T_c > t_1$, $c_{i+1} \leftarrow c_i + \frac{1-W\{\exp[1+\lambda(c_i - T_c)]\}}{\lambda}$.
   c) If $f[t_1, c_i] = 1$ and $T_c > t_1$, $c_{i+1} \leftarrow \min[c_i + \frac{1}{\lambda}, T_c]$.
   d) If $f[t_1, c_i] = e$, $c_{i+1} \leftarrow t_1 + \frac{1-W\{\exp[1+\lambda(t_1 - c_i)]\}}{\lambda}$; $T_c \leftarrow c_i$.
   e) If $f[t_1, c_i] \neq e$, $t_1 \leftarrow c_i$.

3) $i \leftarrow i + 1$; Go to 1.

V. Comments on Relations with Other Collision-Resolution Algorithms

Our algorithm is optimal in the sense of maximizing the probability of successful transmission of a single message during the next time slot. Its maximum stable throughput (obtained in Monte–Carlo simulations) is 0.480.... This throughput is about 1.7% less than the highest throughput achieved to date (0.48776 [4], [5]). In many real channels, the traffic rate changes significantly over time. These channels exhibit short-term, but not long-term, stationarity. For them, short-term solutions (like ours) may have an advantage over long-term maximization of average throughput.

For the first question, our algorithm opens an interval of length $1/\lambda$, while most max-throughput algorithms open an interval of α/λ for some $\alpha > 1$. The advantage of our algorithm, in terms of probability of search termination in the very first step, is by a factor of $\alpha e^{1-\alpha}$. Comparing our collision resolution algorithm to Gallager's (who uses $\alpha = 1.266$ [5]), we have an advantage of about 3% in terms of the probability of success at the first try.

References

[1] V. Anantharam and P. Varaiya, "An optimal strategy for a conflict resolution problem," in *Proc. 24th Conf. on Decision and Control*, Ft. Lauderdale, FL, vol. 2, pp. 1113–1114, 1985.
[2] K. J. Arrow, L. Pesotchinsky, and M. Sobel, "On partitioning a sample with binary-type questions in lieu of collecting observations," *J. Amer. Statistical Assoc.*, vol. 76, no. 374, pp. 402–409, 1981.
[3] D. Bertsekas and R. G. Gallager, *Data Networks*, 2nd ed. Englewood Cliffs, NJ: Prentice-Hall, 1992.
[4] R. M. Corless, G. H. Gonnet, D. E. G. Hare, and D. J. Jeffrey, "On Lambert's W function," Dept. Appl. Math., Univ. of Western Ontario, London, Ont., Canada, Int. Rep., 1993.
[5] R. G. Gallager, "A perspective on multiaccess channels," *IEEE Trans. Inform. Theory*, vol. IT-31, no. 2, pp. 124–142, 1985.
[6] J. L. Massey, "Collision resolution algorithms and random access communications," Univ. of California, Los Angeles, Rep. UCLA-ENG-8016, 1980.
[7] J. Mosley and P. A. Humblet, "A class of efficient contention resolution algorithms for multiple access channels," *IEEE Trans. Commun.*, vol. COM-33, no. 2, pp. 145–151, 1985.
[8] I. Stravanakis and D. Kazakos, "A limited sensing protocol for multiuser packet radio systems," *IEEE Trans. Commun.*, vol. 37, no. 4, pp. 353–359, 1989.

Fuzzy Typographical Analysis for Character Preclassification

Shy-Shyan Chen, Frank Y. Shih, and Peter A. Ng

***Abstract*— This correspondence presents a fuzzy-logic approach for analyzing typographical structures of textual blocks in order to be used for character preclassification. An efficient baseline detection method embedded with tolerance analysis is developed for locating precisely the baseline. Fuzzy logic is taken into account when the decision ambiguity of typographical categorization is occurred. The constraints on the fuzzy membership functions are formulated. Their boundary conditions are considered to preserve the continuity. An improved character recognition rate can be achieved by means of the typographical categorization.**

I. Introduction

Computerized document processing has been growing up rapidly since the 1980's because of the exponentially increasing amount of daily received documents and the more powerful and affordable computer systems. Intuitively, the conversion of textual blocks into ASCII codes represents one of the most important tasks in document processing. Our strategy of preclassifying character is to incorporate the typographical structure analysis which categorizes characters in the first step, and therefore it reduces the scope of character recognition. As illustrated in Fig. 1, a text line can be decomposed into three stripes: the upper, middle, and lower zones. They are delimited by a top line, an upper baseline, a baseline, and an underline. The middle zone, being the primary part of a text line, is about twice as high as the other two zones and can be further split by a mid-line.

Explicitly, the baseline appears in a text line. The upper baseline may not present in the case of a short text composed of ascenders only, and the top line and the underline may not exist if only centered characters appear. Therefore, it is essential to locate the baseline. The baseline can also be used for document skew normalization and for determining interline spacing to be more computationally efficient than the traditional Hough and Fourier transform approaches [4], [7].

Our baseline detection algorithm based on a line of text is more reliable and efficient than the one based on a single word [6]. The remaining virtual reference lines are extracted by a clustering technique [3]. To allow the unpredictable noise and deformation, the tolerance analysis is included. To ensure the robustness and flexibility, a fuzzy-logic approach [5], [9] is used to assign a membership to each typographical category for ambiguous classes. A linear mapping function is adopted and its boundary conditions are derived to preserve the continuity.

The typographical categorization is a part of our document processing system [10] as shown in Fig. 2. The office documents are first digitized and thresholded into binary images by a scanner. The textual blocks are distinguished from graphics and pictures by the preprocessing which includes a block segmentation and classification [10]. The unsupervised character classification classifies characters into a set of fuzzy prototypes based on a nonlinear weighted similarity function [1]. The optical character recognition is intended to recognize the set of fuzzy prototypes. Finally, the

Manuscript received February 27, 1994; revised November 18, 1994.

The authors are with the Institute for Integrated Systems, Department of Computer and Information Science, New Jersey Institute of Technology, Newark, NJ 07102 USA.

IEEE Log Number 9413266.