Simulation Modeling and Statistical Computing

Robert G. Sargent Editor

The P^2 Algorithm for Dynamic Calculation of Quantiles and Histograms Without Storing Observations

RAJ JAIN and IMRICH CHLAMTAC

ABSTRACT: A heuristic algorithm is proposed for dynamic calculation of the median and other quantiles. The estimates are produced dynamically as the observations are generated. The observations are not stored; therefore, the algorithm has a very small and fixed storage requirement regardless of the number of observations. This makes it ideal for implementing in a quantile chip that can be used in industrial controllers and recorders. The algorithm is further extended to histogram plotting. The accuracy of the algorithm is analyzed.

1. INTRODUCTION

In the field of simulation modeling, there is a trend toward reporting medians or 0.9-quantiles rather than mean and standard deviation alone. (The p-quantile of a distribution is defined as the value below which $100p$ percent of the distribution lies.) However, unlike the mean and standard deviation, calculation of quantiles requires several passes through the data, and therefore, the observations have to be stored. Further, a large number of observations is required to get a good estimate of the quantiles. Thus, the amount of memory required becomes very large. In some cases, physical memory limitations make large numbers of replications impossible, and in others, the shuffling of virtual memory pages slows down the simulation considerably.

Most of the literature on median and other quantile calculations is in the area of computational complexity. Several papers [2, 3, 6, 7] have been published with the aim of reducing this complexity. For example, in these papers, it has been shown that medians and other quantiles can be calculated in linear time and memory space. The lower bound on space required to calculate the p-quantile of a sample of n observations is pn. As the number of observations grows, the space requirement grows and soon the exact calculation becomes infeasible due to storage considerations. To save space, experimenters often group the data in cells. However, this approach leads to many idiosyncrasies as described in [5].

This article addresses the storage problem by calculating quantiles dynamically as the data points are generated. The observations are not stored; instead, a few statistical counters are maintained which help refine the estimate.

Raj Jain is now at Washington University in Saint Louis, jain@cse.wustl.edu http://www.cse.wustl.edu/~jain/

The algorithm is then generalized to produce histograms. It turns out that if many quantiles of the same variable are required (e.g., 0.10, 0.50, 0.90, 0.95, etc.), it may be more efficient as well as more accurate to calculate a complete histogram.

The P^2 algorithm proposed in this article requires a very small number of memory locations and does not require prior knowledge of the range (minimum and maximum values) of observations. It can, therefore, be implemented in a chip and used for display of quantiles and histograms in real-time control applications.

In the next section, we present an intuitive development of the P^2 concept, after which the P^2 algorithm is described. We then analyze the performance of the algorithm. In Section 5, some variations of the P^2 algorithm are described and error behavior is analyzed to confirm the superiority of the P^2 design as compared to other similiar designs.

2. INTUITIVE DEVELOPMENT OF THE P^2 CONCEPT

The problem of quantile estimation can be simply stated as follows: given a sample of n observations $\{x_1, x_2, x_3, \ldots, x_n\}$, find the p-quantile. A straightforward method to solve the problem is to sort the observations in increasing order and to plot a "sample cumulative distribution" function as shown in Figure 1. In the figure, $x_{(i)}$ denotes the ith element in the ordered set. A point estimate of the p-quantile can be obtained from this figure by reading $x_{([(n-1)p+1])}$. Here, $[\cdot]$ denotes rounding to the nearest integer.

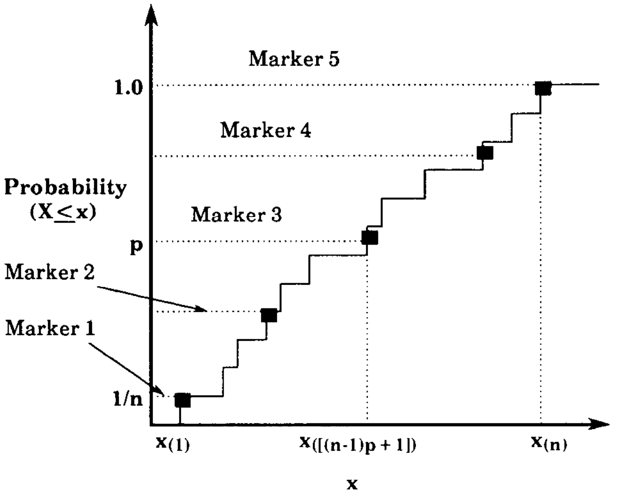


FIGURE 1. One way to calculate a p-quantile is to sort the observations and plot a sample cumulative distribution function. This requires all n observations to be stored. The P^2 algorithm solves this problem by maintaining five markers to store five points on the curve.

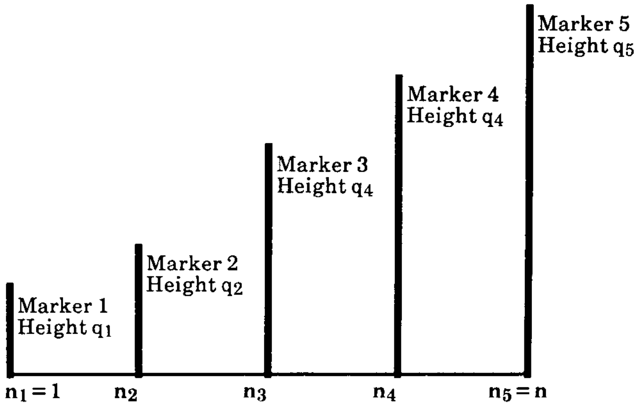


FIGURE 2. The five markers in the P^2 algorithm correspond to the minimum, $p/2$-quantile, p-quantile, $(1 + p)/2$-quantile, and the maximum. The vertical height of each marker is equal to the estimated quantile value.

The main problem with this and other alternative approaches is that all n observations must be stored. In many situations, n can be very large; also, there may be many variables whose quantiles may be required. It is this space problem that we intend to solve in this article. Instead of storing the complete distribution function, we store only five points on it and update the points as more observations are generated. We show that a very good estimate of any quantile can be obtained by locating points at the minimum, the maximum, and at the current estimates of $(p/2)$-, p-, and $(1 + p)/2$-quantiles. This divides observations into four cells whose boundaries (called marker heights) are adjusted if necessary using a Piecewise-Parabolic (PP or P^2) formula. The algorithm has been tested on many different types of random and nonrandom samples and has been observed to produce quantile estimates almost as precise as those obtained by order statistics.

3. THE P^2 ALGORITHM FOR QUANTILES

The algorithm consists of maintaining five markers: the minimum, the $p/2$-, p-, and $(1 + p)/2$-quantiles, and the maximum. The markers are numbered 1–5. Markers 2 and 4 are also called middle markers because they are midway between the p-quantile (marker 3) and the extremes. As shown in Figure 2 (which is a rotated version of Figure 1), the vertical height of each marker is equal to the corresponding quantile value, and its horizontal location is equal to the number of observations that are less than or equal to the marker. True values of the quantiles are, obviously, not known; at any point in time, the marker heights are the current estimates of the quantiles, and these estimates are updated after every observation. Thus, after n observations

if q_i = height of ith marker $i = 1, 2, \ldots, 5$
and n_i = horizontal position of the ith marker

$$i = 1, 2, \ldots, 5$$

$$= \text{number of observation } x_j \text{ such that}$$

$$x_j \leq q_i \quad j = 1, 2, \ldots, n$$

then q_1 = minimum of the observations so far
q_2 = current estimate of the $p/2$-quantile
q_3 = current estimate of the p-quantile
q_4 = current estimate of the $(1 + p)/2$-quantile
q_5 = maximum of the observations so far.

As a new observation comes in, it is compared with the markers, and all markers higher than the observation are moved one position to the right. The resulting locations are then examined. Ideally, the ith marker should be located at n_i' such that

$$n_1' = 1$$

$$n_2' = (n-1)\frac{p}{2} + 1$$

$$n_3' = (n-1)p + 1$$

$$n_4' = (n-1)\frac{(1+p)}{2} + 1$$

$$n_5' = n.$$

If a marker is off to the left or right of its ideal location n_i' by more than one, then the value (height) and the location of the marker is adjusted using a piecewise-parabolic prediction (PP or P^2) formula. The formula assumes that the curve passing through any three adjacent markers is a parabola of the form $q_i = an_i^2 + bn_i + c$. Thus, if a marker is moved d positions to the right, its new height and location are given by:

$$q_i \leftarrow q_i + \frac{d}{n_{i+1} - n_{i-1}}$$

$$\cdot \left\{(n_i - n_{i-1} + d)\frac{(q_{i+1} - q_i)}{(n_{i+1} - n_i)} + (n_{i+1} - n_i - d)\frac{(q_i - q_{i-1})}{(n_i - n_{i-1})}\right\}$$

$$\cdot\ n_i \leftarrow n_i + d \quad i = 2, 3, 4$$

where d is always either +1 or −1. A one position move to the left corresponds to $d = -1$. A derivation of the P^2 formula is given in the Appendix.

Other points regarding the algorithm:

1. The P^2 formula need not be applied to adjust the minimum and the maximum markers. If an observation is less than (or equal to) the current minimum, then the observation becomes the minimum, its location n_1 remains 1, and the locations of all the other markers are incremented by 1. Similarly, if an observation is more than the current maximum, the fifth marker's location is incremented by 1 (as always) and the locations of the other four markers remain unchanged.

2. Successive markers must be kept at least one observation away, that is,

$$n_i > n_{i-1} \quad i = 2, 3, 4, 5.$$

Thus, a marker may not be moved if that would result in two markers being in the same position.

3. The movement of markers to the left or to the right is always one position only. Thus, if a marker is off from its desired location by more than one position, the adjusting move is only one position. It must be noted that the desired locations n_i' are computed on a continuous (real) scale, while the actual locations n_i are on a discrete (integer) scale.

4. For the algorithm to work correctly, the marker heights must always be in a nondecreasing order, that is, $q_i \geq q_{i-1}$. Therefore, if the P^2 formula predicts a value which will make new q_i less than q_{i-1} or greater than q_{i+1}, then the parabolic prediction is ignored and a linear prediction is used as follows.

For a move by d positions ($d = \pm 1$):

$$q_i = q_i + d\,\frac{(q_{i+d} - q_i)}{n_{i+d} - n_i}$$

$$n_i = n_i + d.$$

Again, positive values of d correspond to right moves and negative values to left moves.

5. The first five observations are sorted and used to initialize the five marker values, and marker locations are initialized to

$$n_i = i \quad i = 1, 2, \ldots, 5.$$

As is obvious by now, the marker q_3 is the estimated quantile. An algorithmic description of the P^2 algorithm is given in Box 1.

Example. An example of median calculation using the P^2 algorithm is shown in Table I (p. 1080–1081). The observations are from an exponential distribution with a mean of 10 (median = 6.931). The first five observations 0.02, 0.5, 0.74, 3.39, and 0.83 are sorted and used to initialize the markers.

The sixth observation is 22.37. It is greater than all five existing markers, and so it "fits after" the last marker. This is mentioned in point 1, above. The fifth marker is moved one position and its height becomes 22.37. No further adjustment is necessary since all marker positions are in the desired range.

The seventh observation of 10.15 fits after marker 4. Therefore, marker 5 is moved one position to the right. The desired marker positions are 1, 2.5, 4, 5.5, and 7. Markers 3 and 4 are off from their desired position by at least one. However, marker 3 cannot be moved at this point because it must remain at least one position away from its adjacent markers (condition $n_i > n_{i-1}$). Marker 4 does not have this problem. It is moved one position to the right; its new height 4.47 is calculated using the P^2 formula.

Errata: The second observation 0.5 in the example above should be 0.15.

The eighth observation of 15.43 again fits after marker 4 and results in the adjustment of markers 3 and 4. This procedure is followed as long as new observations are generated.

Box 1

P² Algorithm: To calculate the p-quantile of $\{x_1, \ldots, x_n\}$

```
A. Initialization: Sort the first five observations
   $\{x_1, x_2, x_3, x_4, x_5\}$ and set
   Marker heights    $q_i \leftarrow x_{(i)}$;  $i = 1, \ldots, 5$
   Marker positions  $n_i \leftarrow i$;  $i = 1, \ldots, 5$
   Desired marker positions
      $n'_1 \leftarrow 1$;  $n'_2 \leftarrow 1 + 2p$;  $n'_3 \leftarrow 1 + 4p$;
            $n'_4 \leftarrow 3 + 2p$;  $n'_5 \leftarrow 5$;

   Note that $n'_i$ are real variables, while $n_i$ are
   integers.
   To reduce CPU overhead, calculate and store
   the increment $dn'_i$ in the desired marker positions:

      $dn'_1 \leftarrow 0$;  $dn'_2 \leftarrow p/2$;  $dn'_3 \leftarrow p$;

      $dn'_4 \leftarrow (1 + p)/2$;  $dn'_5 \leftarrow 1$;

B. For each subsequent observation $x_j$, $j \geq 6$, perform the following:
   1. Find cell $k$ such that $q_k \leq x_j < q_{k+1}$ and
      adjust extreme values ($q_1$ and $q_5$) if necessary, that is,
         CASE of $x_j$
         $[x_j < q_1]$             $q_1 \leftarrow x_j$; $k \leftarrow 1$;
         $[q_1 \leq x_j < q_2]$    $k \leftarrow 1$;
         $[q_2 \leq x_j < q_3]$    $k \leftarrow 2$;
         $[q_3 \leq x_j < q_4]$    $k \leftarrow 3$;
         $[q_4 \leq x_j \leq q_5]$ $k \leftarrow 4$;
         $[q_5 < x_j]$             $q_5 \leftarrow x_j$; $k \leftarrow 4$;
         END CASE;

   2. Increment positions of markers $k + 1$
      through 5:

         $n_i \leftarrow n_i + 1 \quad i = k, \ldots, 5$

      Update desired positions for all markers:

         $n'_i \leftarrow n'_i + dn'_i \quad i = 1, \ldots, 5$

   3. Adjust heights of markers 2–4 if necessary:
      FOR $i$ = 2 TO 4 DO
        BEGIN
          $d_i \leftarrow n'_i - n_i$
          IF $\{(d_i \geq 1$ and $n_{i+1} - n_i > 1)$ or
              $(d_i \leq -1$ and $n_{i-1} - n_i < -1)\}$
          BEGIN
            $d_i \leftarrow \text{sign}(d_i)$
            Try adjusting $q_i$ using P² formula:
            $q'_i \leftarrow q_i$ from parabolic formula
            IF $\{q_{i-1} < q'_i < q_{i+1}\}$
            THEN $q_i \leftarrow q_i$
            ELSE use linear formula:
                  $q_i \leftarrow q_i$ from linear formula;
            $n_i \leftarrow n_i + d_i$,
          END IF;
        END DO;

C. Return $q_3$ as the current estimate of $p$-quantile.
```

4. PERFORMANCE OF THE P² ALGORITHM

The performance of the P² algorithm is measured by how close the estimated quantile comes to the parameter being estimated. Given a set of n observations, let T_{p^2} be the P²-quantile (the quantile calculated by the P² algorithm). Sort these n observations and take the $[(n - 1)p + 1]$th element; we call this the "sample-quantile" T_s, or the quantile obtained by the order statistics.

For random sequences (from a given distribution) both the P²-quantile and the sample-quantile would be random estimates of the parent quantile θ. In such cases, the goodness of an estimator is measured by its mean squared error, bias, consistency, and efficiency [4]. The following is a brief explanation of these four terms taken from [4], to which the reader is referred for details.

In estimating a parameter θ by the statistic (or estimator) T, the difference $T - \theta$ is called the error, and mean squared error (MSE) is $E[(T - \theta)^2]$.

The MSE can be decomposed into two parts: the variance of the estimator and a nonnegative term:

$$E[(T - \theta)^2] = \text{var } T + (E[T] - \theta)^2.$$

The quantity $E[T] - \theta$ is called the *bias* in T, and an estimator with zero bias is said to be *unbiased.*

The goodness of an estimator depends on the sample size, and it is reasonable to expect that the larger the sample, the better the inference one could expect to make. If the mean squared error of the estimator T_n (based on a sample of size n) decreases to 0 as more and more observations are incorporated into its computation; that is, if

$$\lim_{n \to \infty} E[(T_n - \theta)^2] = 0$$

then the estimator is said to be *consistent in quadratic mean.* This holds, of course, if and only if both the variance of T_n and the bias tend to 0 as n becomes infinite.

If an estimator T has a mean squared error that is smaller than the mean squared error of another estimator T' in estimating θ from a given sample, the estimator T is thought of as making more "efficient" use of the observations. The relative efficiency of T' with respect to T is the ratio

$$e(T', T) = \frac{E[(T - \theta)^2]}{E[(T' - \theta)^2]}.$$

All of the above criteria for goodness of an estimator are related to its mean squared error. We, therefore, choose MSE as our primary performance metric and compare a P² estimate of a quantile with that obtained by sorting the observations. To estimate MSE, we gen-

TABLE I. An example of median calculation using P² Algorithm

Observation #	Value	Fits after marker	Marker positions after the observation					Desired marker positions				
6	22.37	5	1	2	3	4	6	1.00	2.25	3.50	4.75	6.00
7	10.15	4	1	2	3	4	7	1.00	2.50	4.00	5.50	7.00
8	15.43	4	1	2	3	5	8	1.00	2.75	4.50	6.25	8.00
9	38.62	5	1	2	4	6	9	1.00	3.00	5.00	7.00	9.00
10	15.92	4	1	3	5	7	10	1.00	3.25	5.50	7.75	10.00
11	34.60	4	1	3	5	7	11	1.00	3.50	6.00	8.50	11.00
12	10.28	3	1	3	6	9	12	1.00	3.75	6.50	9.25	12.00
13	1.47	2	1	3	7	10	13	1.00	4.00	7.00	10.00	13.00
14	0.40	1	1	5	8	11	14	1.00	4.25	7.50	10.75	14.00
15	0.05	1	1	6	9	12	15	1.00	4.50	8.00	11.50	15.00
16	11.39	3	1	5	8	13	16	1.00	4.75	8.50	12.25	16.00
17	0.27	1	1	6	9	14	17	1.00	5.00	9.00	13.00	17.00
18	0.42	1	1	6	10	14	18	1.00	5.25	9.50	13.75	18.00
19	0.09	1	1	7	11	15	19	1.00	5.50	10.00	14.50	19.00
20	11.37	3	1	6	10	16	20	1.00	5.75	10.50	15.25	20.00

erate r random samples of size n each from a probability distribution with a known population-quantile θ; we calculate the sample-quantile T_{si} and the P²-quantile T_{p^2i} for the ith set. The MSE for the sample-quantile is then empirically estimated to be

$$\mathrm{MSE}_s = \frac{1}{r}\sum_{i=1}^{r}(T_{si}-\theta)^2.$$

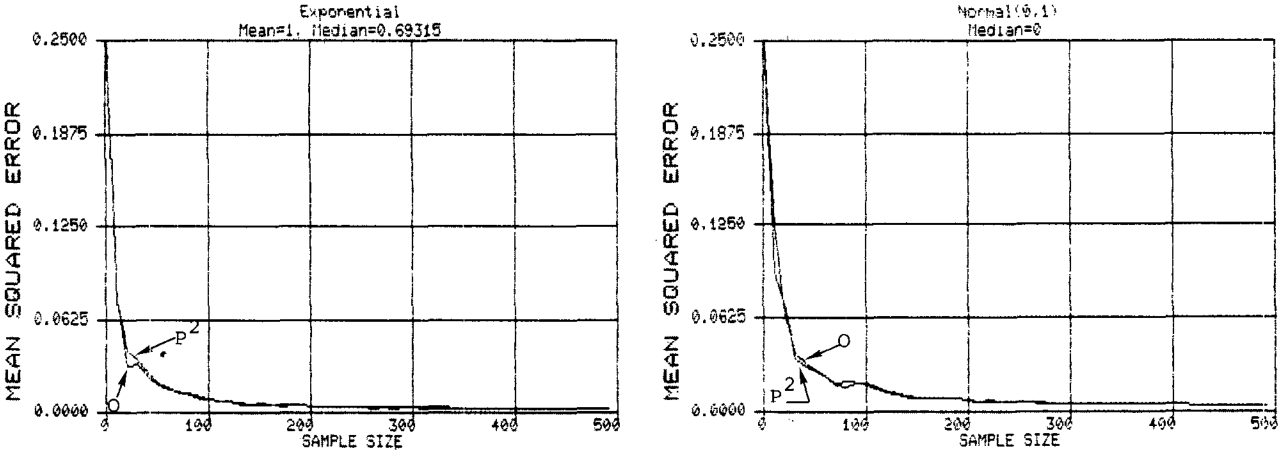
Similarly, MSE for the P²-quantile is estimated to be

$$\mathrm{MSE}_{P^2} = \frac{1}{r}\sum_{i=1}^{r}(T_{P^2i}-\theta)^2.$$

Figure 3 shows MSE's for medians of four different distributions: exponential, normal, log-normal, and uniform. Each curve is based on 50 samples of the given distribution. Similar curves were obtained for 0.10-, 0.90-, 0.95-, and 0.99-quantiles of these distributions. For all cases, the MSE of the P²-quantile is comparable to that obtained by order statistics and that both tend to zero as sample size is increased.

Figure 4 (p. 1082) shows the relative efficiency ($\mathrm{MSE}_s/\mathrm{MSE}_{P^2}$) of the P²-quantiles with respect to the sample-quantiles. For all cases tested, the P²-quantiles seem to be almost as efficient as the sample-quantiles.

Figure 5 (p. 1083) shows the MSE curves for a 3-point discrete distribution. The purpose of this figure is to show that the P² algorithm works for noncontinuous distributions. (The algorithm works perfectly on constant sequences.) However, if the number of possible values the distribution can take is small, one may obtain better quantile estimates by keeping a count for each value than by using P². Also, we do not recom-



Graph 1

Graph 2

FIGURE 3. Mean squared error (MSE) for medians. The MSE's for the medians estimated by the P² algorithm and the order statistics

calculation using P² Algorithm

Adjust markers			New marker positions					New marker heights				
			1	2	3	4	6	0.02	0.15	0.74	0.83	22.37
		4	1	2	3	5	7	0.02	0.15	0.74	4.47	22.37
	3	4	1	2	4	6	8	0.02	0.15	2.18	8.60	22.37
2	3	4	1	3	5	7	9	0.02	0.87	4.75	15.52	38.62
			1	3	5	7	10	0.02	0.87	4.75	15.52	38.62
	3	4	1	3	6	8	11	0.02	0.87	9.28	21.58	38.62
			1	3	6	9	12	0.02	0.87	9.28	21.58	38.62
2			1	4	7	10	13	0.02	2.14	9.28	21.58	38.62
			1	5	8	11	14	0.02	2.14	9.28	21.58	38.62
2	3		1	5	8	12	15	0.02	0.74	6.30	21.58	38.62
			1	5	8	13	16	0.02	0.74	6.30	21.58	38.62
2		4	1	5	9	13	17	0.02	0.59	6.30	17.22	38.62
			1	6	10	14	18	0.02	0.59	6.30	17.22	38.62
2	3		1	6	10	15	19	0.02	0.50	4.44	17.22	38.62
			1	6	10	16	20	0.02	0.50	4.44	17.22	38.62

mend using the algorithm for distributions with discontinuities close to the quantile being computed.

5. FEATURES OF THE P² ALGORITHM COMPARED WITH ALTERNATIVE DESIGNS

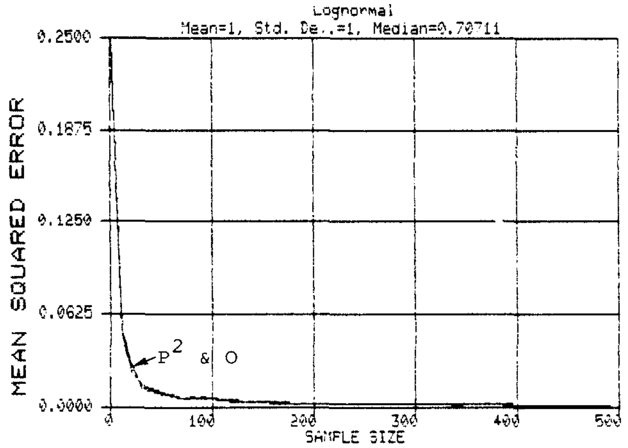
The P² algorithm produces estimates close to those obtained by order statistics. It evolved after a series of trials with other similar designs. In this section, we briefly describe those designs and their observed performance. We justify the current choice of features of the P² algorithm.

The three key features of P² are: five markers, centrally located middle markers, and the piecewise-parabolic prediction algorithm. Let us now change these features one by one and analyze the impact on the performance.

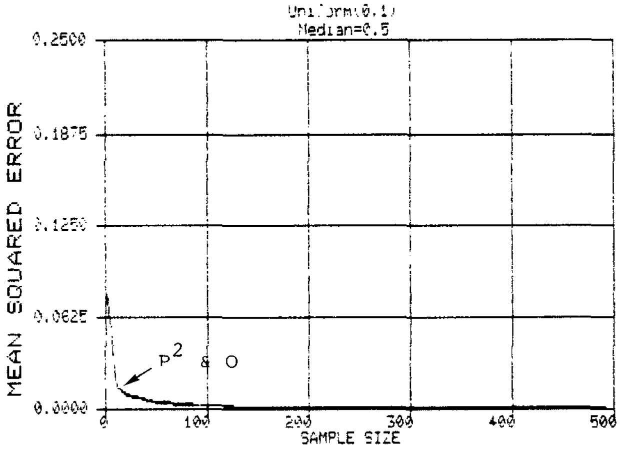
5.1 Three Markers

Suppose we use only three markers instead of five, without changing the remaining features in the P² design. We locate the markers at the minimum, the p-quantile, and the maximum. The second marker is adjusted whenever its position differs from the desired location by more than one. The new value is predicted as a parabolic function of the minimum, the p-quantile, and the maximum. This design gives much larger errors than the five marker design.

Another weakness of this design is that a single outlier observation may cause the error to jump to a large

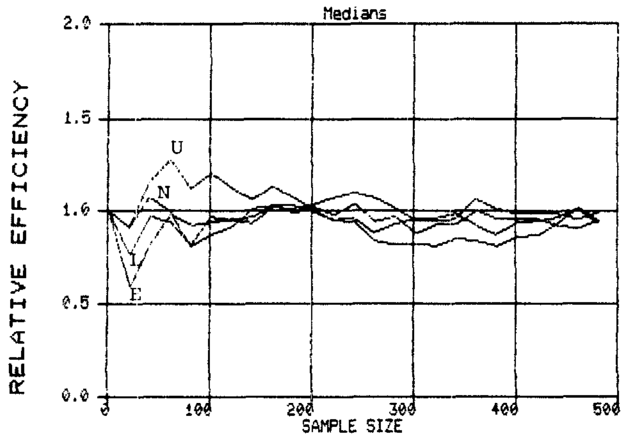


Graph 3

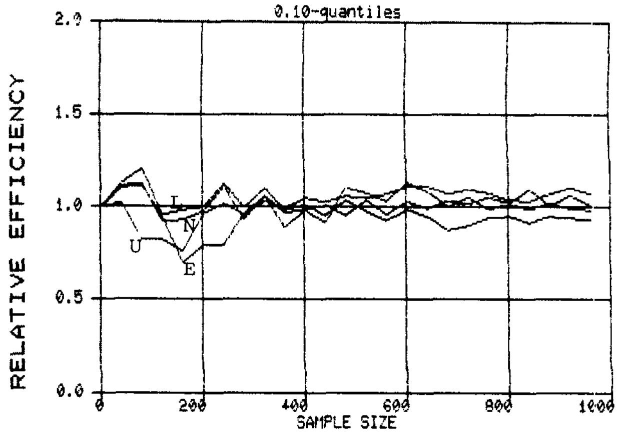


Graph 4

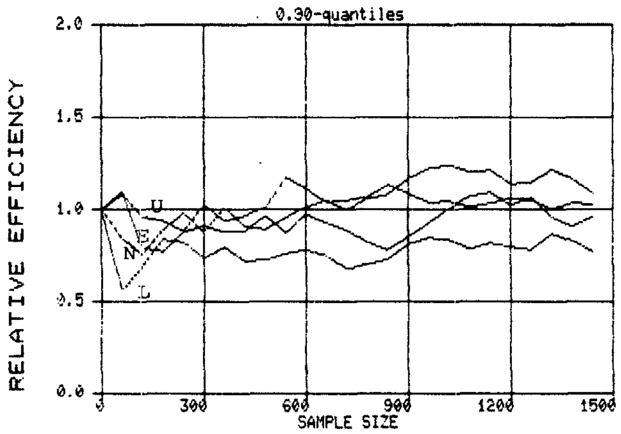
are indicated by P² and O, respectively. In each case, MSE was calculated from 50 samples of the given size.



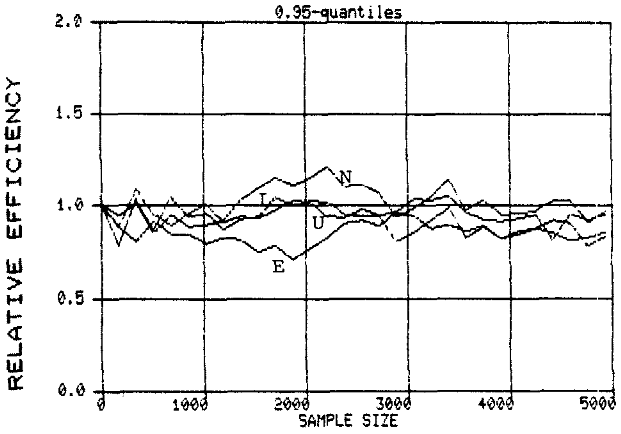
Graph 1



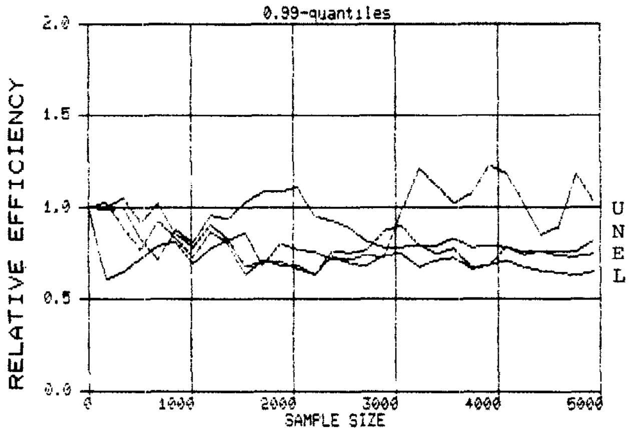
Graph 2



Graph 3



Graph 4



Graph 5

FIGURE 4. Relative efficiency of P^2-quantiles with respect to those obtained by order statistics. Relative efficiency is defined as the ratio of MSE for sample-quantiles and MSE for P^2-quantiles. The following four distributions are shown: E = exponential with mean 1, L = lognormal with mean = 1, and standard deviation = 1, N = normal (0, 1), U = uniform (0, 1).

value. This is due to the fact that for many statistical distributions, the minimum and maximum are statistically "unbounded" variables in the sense that they generally do not attain a stable value as the number of observations increases. Therefore, a single outlier can cause a sudden jump in the maximum or minimum value; that is immediately reflected in the predicted value of the quantile. Putting two more markers around the quantile makes the algorithm less susceptible to extreme values. A sudden jump in the maximum causes a small jump in the fourth marker and still a smaller jump in the quantile. Thus, the middle-markers in the P^2-design serve as "outlier guards."

Interestingly enough, despite the reduced accuracy, the three marker algorithm does not necessarily entail less computation. In fact, in many cases the three marker algorithm results in more frequent adjustment of markers than the five marker algorithm, and thus consumes more CPU time.

It is possible to increase the accuracy by using seven (or more) markers. Extension of P^2 to a higher number of markers is straightforward. Such a design, however, requires considerable increase in CPU time as well as storage overhead. In our empirical tests, five markers were found to give sufficient accuracy (as demonstrated in Section 4).

5.2 Noncentral Middle Markers

In the P^2 algorithm, the two additional markers are kept at $p/2$- and $(1 + p)/2$-quantiles, exactly halfway between the minimum and the desired quantile, and between the quantile and the maximum. When these markers are moved closer to the p-quantile marker (say at 30 percent and 70 percent for the median), the variance of the quantile estimators increases and, in the limit, when the two markers are in locations adjacent to the quantile, the algorithm behaves similarly to the three marker algorithm, that is, it becomes outlier-sensitive. If the points are moved closer to the boundaries (minimum and maximum), their variance (and hence the variance of the quantile estimator) increases until finally the algorithm again tends to behave like the three marker algorithm.

Although any five marker algorithm is superior to the three marker design regardless of where the middle markers are placed, the central location of middle markers between the quantile and the extremes (minimum or maximum) was empirically found to be the best.

5.3 Linear Prediction

In the P^2 algorithm, the curves passing through any three adjacent markers are taken to be parabolic. What happens if a linear prediction is used instead? Our experiments show that the error increases; this result can be explained as follows: the curve passing through the markers is really the cumulative distribution function. For different distributions, this curve is different. A piecewise-parabolic curve provides a second-order approximation. For most distributions (including discrete distributions), this is considerably better than a piecewise linear curve. On the same principle, it follows that a piecewise cubic prediction may provide a better approximation than a parabolic prediction. However, fitting a cubic curve is quite cumbersome and the improvement is not worth the cost. Thus, the piecewise-parabolic prediction provides a good trade-off between complexity and accuracy.

As with the three marker algorithm, the piecewise linear prediction does not necessarily save computation. In fact, in most cases, the piecewise linear prediction results in more frequent adjustments of markers than the P^2 design and, thus, consumes more CPU time.

Figure 6 shows the relative efficiency in median

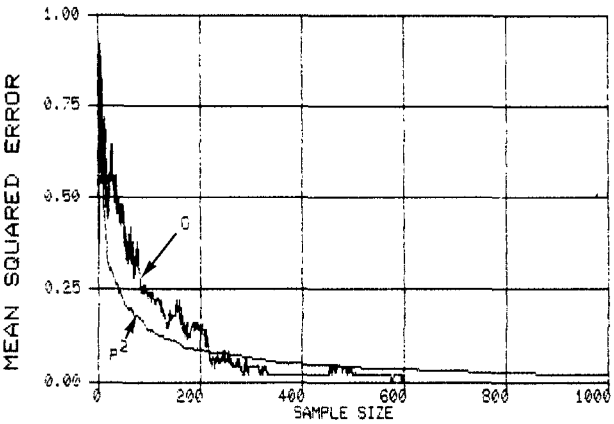


FIGURE 5. Mean squared error (MSE) in median estimation for a discrete distribution. The variables take three values: 0 with probability 0.45, 1 with probability 0.10, and 2 with probability 0.45.

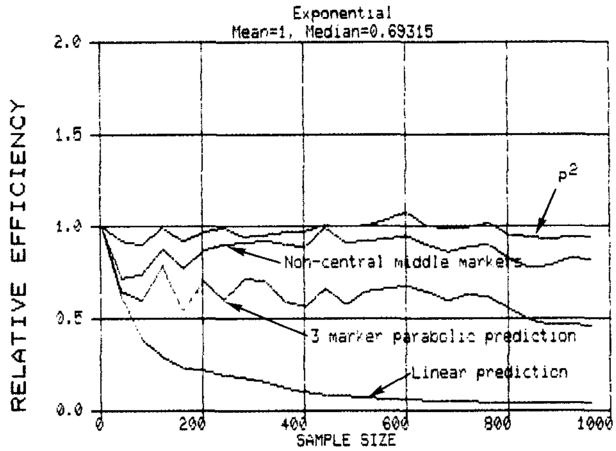


FIGURE 6. Relative efficiency of P^2 algorithm and alternative designs with respect to median estimation using order statistics. The P^2 design appears statistically most efficient.

computation for an exponentially distributed random variable using the above variations of the P^2 design. This figure clearly shows that the P^2 design has the highest relative efficiency among these variants. As discussed in Section 4, high relative efficiency implies low mean squared error.

Box 2

P^2 Algorithm: To calculate a b-cell histogram of $\{x_1, \ldots, x_n\}$

```
A. Initialization: Sort the first $b + 1$ observations
   $\{x_1, x_2, \ldots, x_{b+1}\}$ and set
   Marker heights   $q_i \leftarrow x_{(i)}$;
   Marker positions $n_i \leftarrow i$;        $i = 1, \ldots, b + 1$

B. For each subsequent observation $x_j, j \geq b + 2$,
   perform the following:

   1. Find cell $k$ such that $q_k \leq x_j < q_{k+1}$ and
      adjust extreme values ($q_1$ and $q_{b+1}$) if necessary, that is,
      CASE OF $x_j$
      $[x_j < q_1]$                 : $q_1 \leftarrow x_j; k \leftarrow 1$;
      $[q_i \leq x_j < q_{i+1}]$    : $k \leftarrow i, i = 1, \ldots, b-1$;
      $[q_b \leq x_j \leq q_{b+1}]$ : $k \leftarrow b$;
      $[q_{b+1} < x_j]$             : $q_{b+1} \leftarrow x_j; k \leftarrow b$;
      END CASE;

   2. Increment positions of markers $k + 1$
      through $b + 1$:
         $n_i \leftarrow n_i + 1 \quad i = k + 1, \ldots, b + 1$

   3. Adjust heights of markers 2-$b$ if necessary:
     FOR $i = 2$ TO $b$ DO
       BEGIN
         Calculate desired marker position
         $n' \leftarrow 1 + (i - 1)(n - 1)/b$;*
         $d_i = n' - n_i$
         IF $\{(d_i \geq 1$ and $n_{i+1} - n_i > 1)$ or
            $(d_i \leq -1$ and $n_{i-1} - n_i < -1)\}$
         BEGIN
           $d_i \leftarrow \text{sign}(d_i)$
           Try adjusting $q_i$ using P$^2$ formula:
           $q_i' \leftarrow q_i$ from parabolic formula
           IF $\{q_{i-1} < q_i' < q_{i+1}\}$
           THEN $q_i \leftarrow q_i'$
           ELSE use linear formula:
                $q_i \leftarrow q_i$ from linear formula;
           $n_i \leftarrow n_i + d_i$;
         END IF;
       END DO;

C. A plot of $n_i/n$ on $y$-axis and $q_i$ on $x$-axis gives
   the cumulative histogram.
```

* Note: Some savings in CPU time may be obtained (at the cost of increased storage) by calculating increments in desired marker positions during initialization and maintaining separate counters for the desired positions. Also note that n' is a real variable, while n_i are integers.

6. THE P^2 ALGORITHM FOR HISTOGRAMS

A common problem in plotting histograms of data is choosing proper cell size. If the cells are too narrow, enough observations may not fall in some cells. If cells are too wide, information is lost and the histogram does not show sufficient detail. One way to circumvent this problem is to use equiprobable cells. Particularly when the aim is to fit a distribution function, equiprobable cells are better than equal-size cells [1].

To plot a histogram using b equiprobable cells, all we need are values of $b - 1$ quantiles; namely, the $1/b$-, $2/b$-, $3/b$-, ... $(b - 1)/b$-quantiles, along with the minimum and maximum. Thus, one way to calculate histograms would be to dynamically calculate these quantiles using the P^2 algorithm. Each quantile would require its own set of five markers. Although the space and time requirements for this method would not be large, an even more efficient and accurate method is obtained by adapting the design as follows.

To plot a $b + 1$ point histogram, we make b cells bounded by $b + 1$ equidistant markers with values equal to the current estimates of the minimum, $1/b$-quantile, $2/b$-quantile, ... $(b - 1)/b$ quantile, and the maximum (see Figure 7). The first $b + 1$ observations are sorted to initialize these $b + 1$ markers. Then,

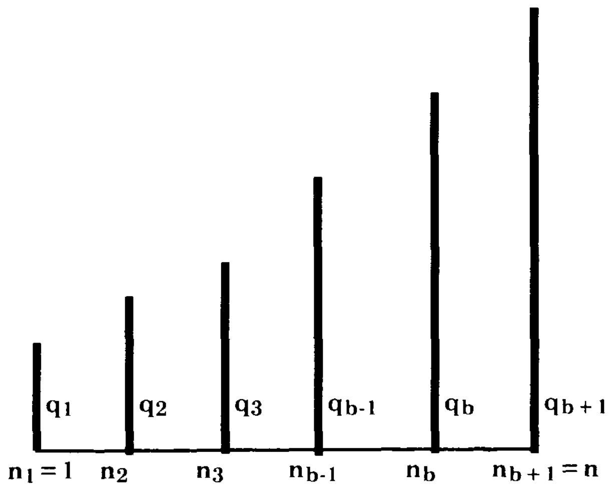


FIGURE 7. The P^2 algorithm for calculating a b-cell histogram. This design consists of maintaining $b + 1$ equidistant markers with their heights corresponding to the minimum, $1/b$-quantile, $2/b$-quantile, ..., $(b - 1)/b$-quantile, and the maximum.

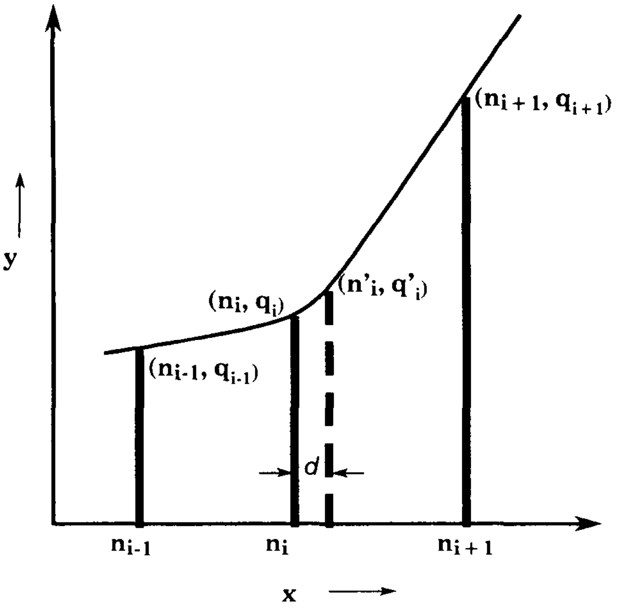


FIGURE 8. The P² formula assumes a piecewise-parabolic curve passing through three adjacent markers.

as a new observation comes in, we increment the locations of markers higher than the observation.

If any marker deviates by more than 1 from the desired position, its position is adjusted. A parabola passing through the two adjacent markers and this marker are then used to adjust the marker. An algorithmic description of the method to obtain a cumulative histogram is given in Box 2.

7. CONCLUSION

In this article, we have proposed a heuristic algorithm for estimating quantiles. The estimates produced are generally as good as those obtained by order statistics, that is, by storing all observations and then sorting them. The advantage of the P^2 algorithm is that observations need not be stored and no prior knowledge of the range of values is required.

The storage requirement of the proposed algorithm is small and fixed, regardless of the number of observations. Thus, it opens a way for the construction of a "quantile chip" to be used in industrial controllers and recorders.

The P^2 algorithm has further been extended to produce histograms. If many quantiles are desired, the calculation of complete histograms is more accurate and computationally more efficient.

APPENDIX

Derivation of the P^2 Formula

As shown in Figure 8, the P^2 formula assumes that the curve passing through (n_{i-1}, q_{i-1}), (n_i, q_i), and (n_{i+1}, q_{i+1}) is a parabola of the form

$$y = ax^2 + bx + c$$

where (x, y) are coordinates (n, q). The coefficients a, b, and c can be determined by solving the following three equations:

$$q_{i-1} = an_{i-1}^2 + bn_{i-1} + c. \tag{1}$$

$$q_i = an_i^2 + bn_i + c. \tag{2}$$

$$q_{i+1} = an_{i+1}^2 + bn_{i+1} + c. \tag{3}$$

Once a, b, c have been determined, it is straightforward to show that the ordinate at $x = n'_i = n_i + d$ is

$$q'_i = an'^2_i + bn'_i + c. \tag{4}$$

$$q'_i = q_i + \frac{d}{n_{i+1} - n_{i-1}} \cdot \left[(n_i - n_{i-1} + d)\frac{q_{i+1} - q_i}{n_{i+1} - n_i} + (n_{i+1} - n_i - d)\frac{q_i - q_{i-1}}{n_i - n_{i-1}}\right]. \tag{5}$$

This is the P^2 formula.

Acknowledgments. The authors wish to thank the referees for their valuable comments which helped enhance the quality of this article in terms of its technical contents and presentation.

REFERENCES

1. Breimann, L. *Statistics: With a View Towards Applications.* Houghton Mifflin, Boston, 1973, p. 207.
2. Dobkin, D., and Munroe, J.I. Optimal time minimal space selection algorithms. *J. ACM 28*, 3 (July 1981), 454–463.
3. Floyd, R.W., and Rivest, R.L. Expected time bounds for selection. *Commun. ACM 18*, 3 (Mar. 1975), 165–173.
4. Lindgren, B.W. *Statistical Theory.* Section 5.1. Macmillan, New York, 1976.
5. Schmeiser, B.W., and Deutsch, S.J. Quantile estimation from grouped data: The cell midpoint. *Commun. Stat. B: Simulation Comput. 6*, 3 (1977), 221–234.
6. Schonhage, A., Paterson, M., and Pippenger, N. Finding the median. *J. Comput. Syst. Sci. 13* (1976), 184–199.
7. Yap, C.K. New upper bounds for selection. *Commun. ACM 19*, 9 (Sept. 1976), 501–508.

CR Categories and Subject Descriptors: D.1.1 [**Programming Techniques**]: Applicable (Functional) Programming; G.3 [**Mathematics of Computing**]: Probability and Statistics—*statistical software*; I.6 [**Computing Methodologies**]: Simulation and Modeling—*general*; J.7 [Computer Architecture]: Computers in Other Systems—*real time*
General Terms: Algorithms
Additional Key Words and Phrases: quantiles, percentiles, median, simulation, histogram, selection problems, real-time estimation

Received 5/82; revised 3/85; accepted 5/85

Authors' Present Addresses: Raj Jain, Eastern Research Lab, Digital Equipment Corporation, 77 Reed Road (HLO 2-3/NO3), Hudson, MA 01749. Imrich Chlamtac, Department of Computer Science, Technion Israel Institute of Technology, Haifa 32000, Israel.