Low-Storage Quantile Estimation

Article *in* Computational Statistics · January 1995

CITATIONS
19

READS
42

2 authors:

Catherine Hurley
National University of Ireland, Maynooth
35 PUBLICATIONS **641** CITATIONS
SEE PROFILE

Reza Modarres
George Washington University
83 PUBLICATIONS **865** CITATIONS
SEE PROFILE

20-12-8-10 x416

CLS3086824: Computational statistics.

WR: (ISSN 0943-4062 / Physica-Verlag, [1992]- /)

Requested: 2017-09-09 11:07

Not needed after:

Pickup at: GW George Washington - Gelman

WDD: Web Delivery

Citation: Issue: v.10 () 1995; Article: Low-Storage Quantile Estimation / Hurely C and Modarres R, C; Pages: 311-325

* WRLC Shared Collections Facility - 32882018898885 - - Charged(2012-01-13)

* WRLC Shared Collections Facility - 32882019314585 - - Charged(2012-01-13)

* WRLC Shared Collections Facility - 32882017225437 - - Charged(2012-01-13)

Computational Statistics (1995) 10:311-325

Computational Statistics

Low-Storage Quantile Estimation

C. Hurley and R. Modarres

Department of Statistics, The George Washington University, Washington DC 20052, USA

Summary

When a dataset is too large to be stored in primary memory, standard algorithms for computing sample quantiles are not directly applicable. In this paper, we examine the statistical and computational properties of methods for quantile estimation from a dataset of n observations, which use far fewer than n memory locations. A histogram-based estimator is proposed, which is appealing for its conceptual and computational simplicity. Its good statistical properties are borne out by a simulation study, where it produces estimates which are similar to the sample quantiles, and with greater accuracy than the estimates yielded by other, more complex and computationally intensive methods.

Keywords: Histogram, quantile, space efficiency.

1 Introduction

Estimators that are formed as a result of performing algebra on the observations are not robust. Most robust estimators are based on quantiles, which are not algebraic functions of the observations. Typically, in order to find the sample quantiles of a set of n data values, they must be stored simultaneously in n memory locations.

Extremely large datasets are becoming increasingly commonplace. However, in any computing environment the number of observations that can be stored in primary memory is restricted, because the available memory, though large, is finite (even with virtual memory). For example, at our local SAS installation (SAS Institute, 1990), PROC SORT halts due to insufficient virtual storage with as few as 50,000 numeric observations (on an IBM 4381 main-

0943-4062/95/4/311-325 $2.50

frame under the CMS operating system). Many computing environments also limit the maximum array size allowed, and this may be much smaller and even independent of the available memory. For instance, in Common Lisp (Steele, 1990) the maximum array size is implementation dependent. In one particular Lisp implementation, Macintosh Common Lisp, (Apple Computer, 1992), the maximum array size is just over two million. Even that will not be sufficient for some applications, which can have gigabytes or even terabytes of data. Rousseeuw and Bassett (1990) give various applications where quantiles of large datasets are required, ranging from ERG curves in opthamology to crystallography.

This article is concerned with low-storage quantile estimation. More specifically, assuming the data $x_1, x_2, \ldots, x_n$ are independent observations from a distribution function F, we wish to estimate the number ξ_α such that $F(\xi_\alpha) = \alpha$, when the available number of memory locations is $m << n$. This problem is related to the selection problem, which is concerned with finding the kth smallest of n observations. The kth smallest observation is variously taken as an estimate of the k/n or $k/(n+1)$th quantile. Conversely, an estimate of the αth population quantile provides an approximation to the $\lceil \alpha n \rceil$th largest observation. Baeza-Yates and Gonnet (1991) survey the selection problem and Mahmoud, Modarres and Smythe(1993) analyze a time efficient algorithm for selection of order statistics.

A sample quantile can be selected with many passes through the data. A naive approach would compute the sample median using m memory locations with $\lceil n/2m \rceil$ passes through the data. The inefficiency of this algorithm stems from two sources. First, selecting the sample median in this way requires $O(n)$ passes through the data. Since disk access is far slower than RAM access, an efficient low-storage quantile estimator should rely on a single pass through the data. Second, for fixed m, the algorithm requires $O(n^2)$ comparisons. This compares poorly with the $O(n)$ comparisons required when the data is stored in primary memory (see, for example, Baeza-Yates and Gonnett, 1991). Alternatively, some authors have taken a probabilistic approach to the problem. Munro and Paterson (1980) and Dunn (1991) offer a low-storage, probabilistic, selection algorithm and Hatzinger and Panny (1993) discuss an efficient implementation of Dunn's algorithm.

Many different approaches have been taken to the problem of low-storage quantile estimation. One approach given by Tierney (1983) is based on stochastic approximation, using algebraic operations to update the quantile estimate as the data values are read. Therefore, it is not necessary to store the data values simultaneously in primary memory. Another approach to quantile estimation is based on trees. Pearl (1981) suggested a minimax tree for estimating an arbitrary quantile. Other authors (Tukey, 1978; Weide, 1978; Rousseeuw and Bassett, 1990) independently proposed another tree-based approach to median estimation, using recursive medians of subsamples. In Section 2 we describe the above methods. We compare their computational

properties, such as storage demands, and the ease and speed of computation, as well as statistical properties such as asymptotic behavior and robustness.

In Section 3 we describe an approach to low-storage quantile estimation based on the sample histogram, which has good computational and statistical properties. While basing estimates on grouped data such as histograms is commonplace, to our knowledge such methods have not been examined in the context of low-storage, single-pass quantile estimation from very large datasets.

Finally, in Section 4 we report on a simulation study, where we compare the efficiency of various low-storage quantile estimators. The study verifies that quantile estimates based on the sample histogram are similar to the sample quantiles, and have greater accuracy than the estimates produced by other, more complex and computationally intensive methods.

2 Approaches to low-storage quantile estimation

In order to simplify the discussion we focus on the problem of estimating the median, which we denote as ξ. In most cases, the properties given for the median generalize easily to other quantiles.

Except for its high storage requirements, the sample median $\hat{\xi}_s$ has good properties: it is consistent and asymptotically normal assuming the density $f(\xi) > 0$ exists at ξ. That is,

$$\sqrt{n}(\hat{\xi}_s - \xi) \xrightarrow{\mathcal{D}} \mathrm{N}\left(0, \frac{1}{4f^2(\xi)}\right). \tag{1}$$

In the absence of further knowledge about F, there is no aymptotically uniformly median-unbiased, translation equivariant estimator of ξ with smaller asymptotic variance (Pfanzagl, 1974). The sample median also has good robustness properties. One measure of robustness is the finite sample breakdown point (see Hoaglin et al., 1983, for example) defined as the smallest fraction of observations which when replaced, can result in an estimator which is arbitrarily large or small. The sample median has the ideal breakdown of $\lceil n/2 \rceil / n \approx 1/2$, while by contrast, the sample mean has breakdown $1/n$. We note that, to calculate the breakdown point for non-permutation invariant estimators, the j values being replaced have indices belonging to the most damaging (breakdown-wise) j-subset of $\{1, 2, \ldots, n\}$.

2.1 Minimax trees

Pearl (1981) proposed a quantile estimator based on minimax trees. We describe this estimator, summarize the properties as given by Pearl (1981), and derive its breakdown point.

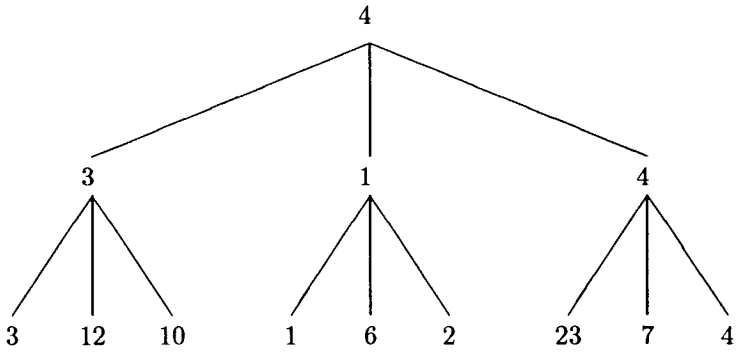

Figure 1: A minimax tree

A minimax tree is a uniform d-ary tree whose terminal nodes are independent observations from F, and whose non-terminal nodes at odd levels of the tree contain the minimum values of its d children while even level nodes contain the maximum values of its d children. (By convention the root is at level 0.) The minimax estimator is based on the observation that the minimax value of the root node converges in probability to a specific quantile of the distribution function F (Pearl, 1981), where the quantile estimated depends on the branching factor d of the minimax tree. The minimax value of the root node is found by traversing the tree from left to right, examining the terminal values and propagating the minimum and alternatively the maximum value upwards through the non-terminal nodes. A tree with height h will have d^h terminal nodes, and so can be used to estimate a quantile from up to d^h observations.

Figure 1, for example, shows a minimax tree with a height h of 2, containing observations 3,12,10,1,6,2,23,7 and 4 at the terminal nodes. The values $3, 1, 4$ at level 1 are obtained by computing the minimum of those at level 2. The root of the minimax tree (level 0) is then the maximum of the level 1 values. This tree has branching factor $d = 3$, which for tall trees gives a root value approximating the 31.7 percentile value of the distribution. If the maximum and minimum nodes are exchanged, the resulting structure is described as a maxmin tree. In this case the root node for $d = 3$ approximates the 68.3^{th} percentile.

In order to enlarge the set of estimable quantiles, different branching factors d_1 and d_2 for the minimum and maximum nodes respectively, are employed. For example, branching factors of 3 and 5 result in an estimate of the 49th percentile, while branching factors of 6 and 44 give an estimate of the median. In general, the minimax tree with branching factors d_1 and d_2 estimates the α^{th} quantile of F, where α is the solution of $\alpha = (1-(1-\alpha)^{d_1})^{d_2}$.

Table 1: Computational properties (storage requirements, ease of implementation, range of n) of median estimators.

Method	Space	Implementation	n
Median	n	difficult	any
Minimax	$2h$	difficult	$(d_1 d_2)^{h/2}$
Remedian	bk	easy	b^k
S.A.	m	moderate	any
Histogram	$m+2$	very easy	any

Table 2: Statistical properties (breakdown, permutation invariance, location-scale equivariance, monotone equivariance and asymptotic normality) of median estimators.

Method	Breakdown	Perm. Invar.	Loc-Scale Equiv.	Mono. Equiv.	Asymp. Normal
Median	$\lceil n/2 \rceil / n$	yes	yes	yes, odd n	yes
Minimax	$[\max(d_1, d_2)]^{-h/2}$	no	no	yes	no
Remedian	$\left(\frac{\lceil b/2 \rceil}{b}\right)^k$	no	no	yes, odd b	no
S.A.	$\lceil m/2 \rceil / n$	no	yes	no	yes
Histogram	$\lceil n/2 \rceil / n$	yes	yes	no	yes

The corresponding maxmin tree estimates the $(1-\alpha)^{th}$ quantile. When such a tree has height h, there will be $(d_1 d_2)^{h/2}$ terminal nodes.

The properties of the minimax tree quantile estimator are summarized in the second row of Tables 1 and 2. Pearl (1981) describes how the estimate can be computed in $2h$ storage locations, and shows that the estimator is consistent for strictly increasing F, at least for the case of $d_1 = d_2$. The asymptotic distribution is non-normal, and the convergence rate is slower than $\sqrt{n}$.

Clearly, the estimator is not permutation invariant. Also, an undesirable property of this median estimator is its lack of location-scale equivariance: the root value of the minimax tree computed on observations whose signs are flipped is the negative of the root value of the maxmin tree computed on the original observations.

Next we derive the breakdown properties of the minimax estimator. In the minimax tree, each successive pair of minimum and maximum levels reduces $d_1 d_2$ values to a single value, by grouping the values into d_2 groups of size d_1, computing the minima of those groups, and then computing the maximum of the d_2 minima. For the $d_1 d_2$ values to be reduced to an arbitrarily large

value, at least one of the d_2 minima must be arbitrarily large, which requires that at all of the values in a group of size d_1 be replaced. Similarly, for the $d_1 d_2$ values to be reduced to an arbitrarily small value, at least one value from each of the d_2 groups must be replaced. With a total of $h/2$ pairs of successive minimum and maximum levels, this leads to a finite sample breakdown point of $\epsilon_{\text{minimax}} = [\min(d_1, d_2)]^{h/2}/n$. When n is exactly $(d_1 d_2)^{h/2}$, this simplifies to $[\max(d_1, d_2)]^{-h/2}$, giving $1/\sqrt{n}$ for the case when $d_1 = d_2$. In fact, for a fixed value of n, the highest breakdown is achieved when $d_1 = d_2$, and so,

$$\frac{2^{h/2}}{n} \leq \epsilon_{\text{minimax}} \leq \frac{1}{\sqrt{n}}. \tag{2}$$

Thus, the breakdown point of minimax quantile estimators, while better than that of the sample mean, tends to zero as the sample size $n \to \infty$.

2.2 The remedian

In this section, we describe another tree-based method for median estimation, which uses recursive medians of subsamples. We show that this method requires more storage than the minimax estimator, but has a higher breakdown point.

The method of recursive medians has appeared several times in both the statistics and computer science literature, see Weide (1978), Tukey (1978), Pearl (1981), (who attributes the idea to Cantor), and Rousseeuw and Bassett (1990). Blum et al (1973) use the method for pivot selection in a one-sided quicksort algorithm for the selection problem. Following Rousseeuw and Bassett (1990), we call this estimator the *remedian*. The method can be extended to other quantiles, as described by Chao and Lin (1993).

Suppose that n is of the form b^k where b and k are integers. The remedian algorithm processes the observations sequentially in groups of size b. A median is computed for each group, yielding b^{k-1} medians at the first stage. This step is repeated recursively until a single estimate is found (see Figure 2). A remedian can be computed using k arrays of size b requiring kb storage locations.

Rousseeuw and Bassett (1990) have investigated the computational and statistical properties properties of the remedian, which we summarize in Tables 1 and 2. We note that implementation is easy as long as n has the form b^k. For fixed b as $k \to \infty$, the estimator is consistent, converging at a sub-optimal rate to a non-normal asymptotic distribution. Recently, Chao and Lin (1993) investigated the asymptotic properties of the remedian under more general conditions. The estimator has good breakdown properties, but the demands of robustness are in conflict with the requirement that storage be kept low: as the storage increases so does the finite sample breakdown point.

Next we compare the storage requirements of the minimax and remedian methods. From Rousseeuw and Bassett (1990), when $b = 3$ the remedian

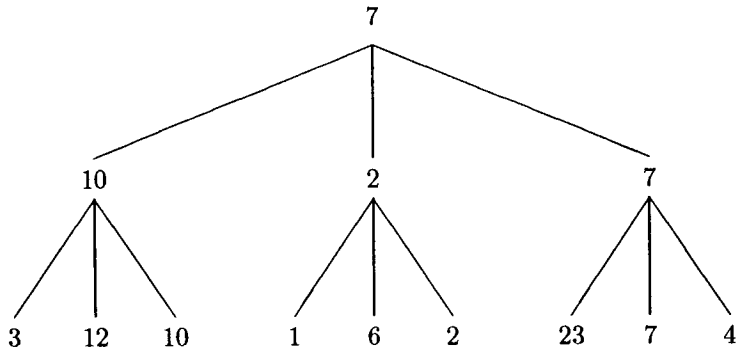

Figure 2: A remedian tree

uses the smallest possible amount of storage, which is $3\log_3 n$. Now, unless both d_1 and d_2 are 2,

$$3\log_3 n > 2\log_{\sqrt{d_1 d_2}} n = 2h,$$

which is the storage used by the minimax tree.

The remedian requires more storage locations, but with the exception of the degenerate $b = 2$ case, has better breakdown properties than the minimax estimator, as we now show. The remedian has breakdown point $\epsilon_{\text{remedian}} = (\lceil b/2 \rceil / b)^k \geq 1/2^k$ (Rousseeuw and Bassett, 1990). When $b \geq 4$,

$$k = 2\log_b \sqrt{n} = 2\frac{\log_2 \sqrt{n}}{\log_2 b} \leq \log_2 \sqrt{n},$$

and so $1/2^k \geq 1/\sqrt{n}$. When $b = 3$, the breakdown $\epsilon_{\text{remedian}} = (2/3)^k > 1/\sqrt{n}$, using steps similar to above. And so, using (2), we have that, for $b \geq 3$,

$$\epsilon_{\text{remedian}} \geq \frac{1}{\sqrt{n}} \geq \epsilon_{\text{minimax}}.$$

In the $b = 2$ case, the remedian collapses to the sample mean, and its breakdown is $\epsilon_{\text{remedian}} = 1/n < \epsilon_{\text{minimax}}$, again using (2).

2.3 Stochastic approximation

Tierney (1983) proposed a stochastic approximation (S.A.) algorithm for estimating the αth quantile ξ_α. A starting estimate $\hat{\xi}_{\alpha,m}$ is obtained using the first m observations, say. Then as observations x_i are read in, the estimate is updated according to the following formula:

$$\hat{\xi}_{\alpha,i+1} = \hat{\xi}_{\alpha,i} - c_i \left(I_{[x_{i+1} \leq \hat{\xi}_{\alpha,i}]} - \alpha \right), \tag{3}$$

where $I_{[\,]}$ is the indicator function, and c_i is chosen so that the resulting estimator has minimum asymptotic variance.

The formula (3) looks deceptively simple; in fact the quantity c_i involves an estimate of the density at the population quantile, and computing c_i at each step requires quite a few floating point operations. The space complexity and breakdown properties for the S.A. estimator (see Tables 1 and 2) depend on the number of observations used in obtaining the starting estimate $\hat{\xi}_{\alpha,m}$. Interestingly, since the c_i's are finite, it is not possible to breakdown this estimator by replacing any or all of $x_{m+1}, x_{m+2}, \ldots, x_n$. Tierney(1983) shows that the estimator has the same large-sample behavior as the sample median. Like the remedian and minimax estimators, this estimator is not permutation invariant.

3 Histogram Quantile Estimates

We propose the following histogram estimator of the population quantile. For ease of presentation, we focus on the median in the following discussion.

Suppose for simplicity l, r are fixed and known to contain the sample median. Later we relax this assumption and address the case where l and r are estimated from the data. Divide $(l, r]$ into m bins each of width $(r - l)/m = w/m$. Then count the number of observations falling in each bin $(l + iw/m,\ l + (i+1)w/m], i = 0, 1, \ldots, m-1$, and the number falling at or below l and beyond r. We define the histogram estimator of the population median as

$$\hat{\xi} = a\,\frac{F_n(b) - 1/2}{F_n(b) - F_n(a)} + b\,\frac{1/2 - F_n(a)}{F_n(b) - F_n(a)}, \tag{4}$$

where $F_n(x)$ is the empirical distribution function and the bin $(a, b]$ contains the sample median so that

$$F_n(a) < 1/2 \leq F_n(b).$$

Note that the estimator $\hat{\xi}$ is simply obtained by linear interpolation of the bin endpoints.

This estimator requires m storage locations and is computed in $O(n)$ steps. Since

$$\hat{\xi} = \hat{\xi}_s + O\left(\frac{1}{m}\right),$$

where $\hat{\xi}_s$ is the sample median, it follows that the $\hat{\xi}$ has the same asymptotic distribution (1) as the sample median, when $n = o(m^2)$. Alternatively, for fixed a and b, the estimator (4) has an asymptotic normal distribution, but it is not necessarily a consistent estimator of ξ. From the joint asymptotic normality of the empirical c.d.f. evaluated at a and b, and the multivariate

delta method (see, for example Rao, 1973 pp. 386–387), we have that

$$\sqrt{n}(\hat{\xi}-\tilde{\xi}) \xrightarrow{\mathcal{D}} \mathrm{N}\left(0, \frac{(b-a)^2}{4\left(F(b)-F(a)\right)^2} - \frac{(b-a)^2(1/2-F(a))(F(b)-1/2)}{(F(b)-F(a))^3}\right),$$

where

$$\tilde{\xi} = a\,\frac{F(b)-1/2}{F(b)-F(a)} + b\,\frac{1/2-F(a)}{F(b)-F(a)}.$$

Properties of this histogram estimator are summarized in the fifth lines of Tables 1 and 2. Unlike previous methods, this procedure leads to an estimator which is permutation invariant. This property is important when there are dependencies among the data values, which frequently occurs when they are collected over time. The lack of monotone equivariance of the histogram estimator has two sources: the first is the use of equispaced bins in the histogram construction, and the second is the use of linear interpolation between the endpoints of the bin containing the sample median.

The histogram estimate of the population median is based on a compact, frequency table representation of the n data values. Once the table is constructed, any population parameter may be estimated in time which is independent of n. This contrasts with the other low-storage estimators presented, where each quantile estimated causes the amount of work performed to increase by an amount proportional to n.

Our discussion so far has assumed that an interval $(l, r]$ containing the sample median is known. In many situations the context of the data will suggest such an interval, but it may be too wide to be of practical use. Fortunately, if it happens that l and r are incorrectly specified and do not contain the sample median, this is detected by the algorithm.

We now describe how suitable values of l and r may be obtained from the data. The simplest and most conservative strategy uses the first m values to estimate l and r by the minimum and maximum. However, especially for extremely heavy-tailed distributions such as the cauchy distribution where wild observations occur, this strategy can lead to a wide $(l, r]$ interval and thus an unreliable median estimator.

More generally, one could let l be the $j = \lceil qm \rceil$th smallest and r the jth biggest among the first m values, where $0 \leq q < 1/2$. We would like the resulting interval to be narrow, but with a high probability of containing the sample median. We note that

$$P\left(\hat{\xi}_s \in (l, r]\right) = P\left(M \geq j \text{ and } m - M \geq j\right),$$

where M is the number of values from the first m which are less than $\hat{\xi}_s$. Assuming that the first m values can be regarded as a random sample from the totality of n values, $M \sim \text{Hypergeometric}(m, n, n_1)$, where n_1 is the total number of values which are less than $\hat{\xi}_s$. When there are no ties ($n_1 = \lceil n/2 \rceil$)

and n is large, M is approximately distributed as $\mathrm{N}(m/2, m/4)$. Therefore,

$$\begin{aligned} P\left(\hat{\xi}_s \in (l,r]\right) &= P(j \le M \le m-j) \\ &\approx 1 - 2\Phi\left(\frac{j-m/2}{\sqrt{m/4}}\right) \\ &\approx 1 - 2\Phi\left(2\sqrt{m}\left(q-\frac{1}{2}\right)\right). \end{aligned} \tag{5}$$

The lower and upper quartiles of the first m values ($q = 1/4$) are thus a reasonable choice for l and r; the resulting interval has a high probability of containing the true sample median (more than 99.84% for m of 40 or beyond).

Of course, when the first m values are used to choose l and r, the estimator loses the permutation invariance property and the breakdown drops from the optimal $\lceil n/2 \rceil / n$ to $\lceil qm \rceil / n$. However, to break the estimator down, the $j = \lceil qm \rceil$ extreme values would have to lie among the first m values. Assuming all values are equally likely to be outliers, the probability that j values cause the estimator to breakdown is

$$\frac{\binom{n-j}{m-j}}{\binom{n}{m}} = \frac{m(m-1)\cdots(m-j+1)}{n(n-1)\cdots(n-j+1)} < \left(\frac{m}{n}\right)^j,$$

which is small.

4 Simulation Study

In the previous two sections, we examined the asymptotic properties of several methods for quantile estimation. However, it is worthwhile to investigate the properties of these estimators for finite sample sizes. Ideally, we would like to consider sample sizes of several million observations, but given the limitations of computing resources, we chose to perform a simulation based on 1000 repetitions with a sample size of 50,625. The random number generator used was described by Marsaglia (1972).

In our simulation study, we compare the performance of the five methods of median estimation listed in Tables 1 and 2. The sample median is included in the study in order to compare the low-storage methods to the estimator typically used when the storage is unrestricted. Since both the remedian and minimax methods place limitations on the form of the sample size, we set b (buffer size) to 15 and k (the number of buffers) to 4, for the remedian algorithm. For the minimax method, we choose the branching factors d_1 and d_2 to be 3 and 5, respectively, and the tree height h to be 8. These parameter choices result in a sample of size $n = 50,625$, with remedian using 60 memory locations and the minimax method using 16. To make the memory requirements of the other methods comparable, we also set m to 60.

Table 3: Simulation Results. For the low-storage estimators, the ratio of the MSE to the MSE of the sample median is given, for the normal, contaminated normal, cauchy and χ^2_1 distributions.

Method	Normal	Contam.	Cauchy	χ^2_1
Minimax	76.95	79.61	73.88	75.327
Remedian	3.291	3.453	3.134	3.332
S.A.	1.004	1.013	1.025	1.001
Histogram	0.992	0.995	0.994	0.991

For the histogram method, we use the first m values to estimate l and r by the upper and lower quartiles. (From Equation (5) with $m = 60$, such an interval contains the actual sample median with a probability of 0.9999.)

We compare the four low-storage estimators for four continuous distributions, three exhibiting increasing amounts of heavy-tailedness, namely the standard normal, a contaminated normal and the cauchy distributions, and the chi-square distribution with one degree of freedom, which exhibits a high-degree of skewness. The contaminated distribution used is $N(0, 1)$ contaminated by 10% $N(0, 9)$. For each of the five methods, we estimate the population median. In Table 3, we report the ratio of the mean square error for each estimator to the mean square error of the sample median.

The minimax estimator consistently performs very poorly over the range of distributions considered. The poor performance can be traced to its asymptotic behavior, which requires the tree height $h \to \infty$. In our simulation, h was merely 8. A moderately large h value of 150 for instance, would require a sample size n of more than $1.6\text{x}10^{88}$. With $d_1 = 3$ and $d_2 = 5$, the maxmin tree used in the simulation is actually estimating the 51st percentile of the parent distribution, rather than the median. However, the contribution of the bias term to the mean square error of the estimator is negligible. The branching factors of 6 and 44 required to estimate the median, even with a tree height as small as 8, would require a sample size prohibitively large for the purposes of simulation.

The performance of the other tree-based estimator, the remedian, while much better than that of the minimax estimator, is less than ideal. For each of the distributions considered, its mean square error is more than three times that of the sample median. Figure 3 shows plots of the remedian for the cauchy and χ^2_1 samples only (the plots for the other distributions are similar). Once again, we see that the variability of the remedian far exceeds that of the sample median. We also notice that the correlation between the remedian and the sample median is low.

The mean square errors of both the S.A. and histogram estimators are

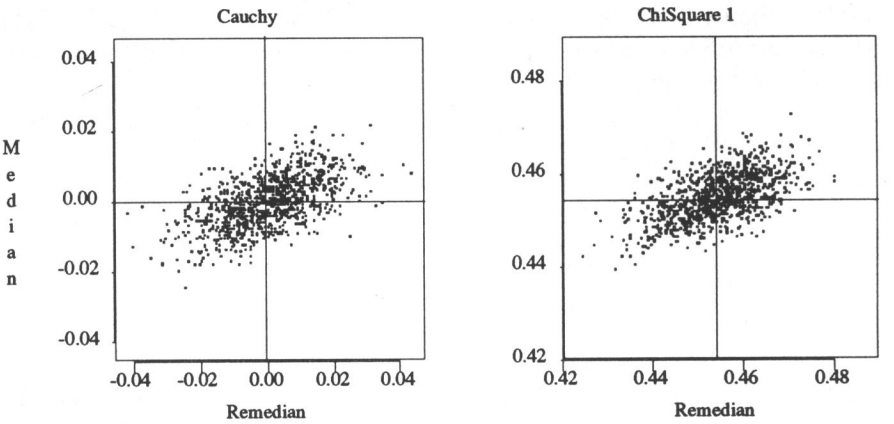

Figure 3: Plots of remedian estimates. The x-axis shows the remedian, and the y-axis shows the the sample median, for the cauchy and χ^2_1 samples.

very close to that of the sample median, for each of the distributions considered, with the histogram estimator having slightly smaller MSE in each case. For a closer examination, we turn to Figure 4 which shows the cauchy and χ^2_1 samples only (the plots for the other distributions are similar). Both methods yield estimates which have very high correlation with the sample median, therefore it is more informative to plot deviations from the sample median on the y-axis. Here we see that for both the cauchy and χ^2_1 samples, the methods give us estimates that are almost always within .02 of the true population median. Overall, it appears that the histogram estimates track the sample median a little more closely than do the stochastic approximation estimates. In the case of the χ^2_1 distribution, the histogram estimator shows evidence of a very slight, positive bias. This is not surprising, given that we are performing linear interpolation in intervals where the density is decreasing.

5 Discussion

In this investigation, we have examined the computational and statistical properties of four methods of quantile estimation. For clarity of presentation we focused on median estimation, but except in the case of the remedian, the extension to other quantiles is immediate.

While the method of Pearl (1981) has very low storage requirements, our simulation study demonstrated that it is very unreliable, at least for the moderately large sample size considered. The remedian algorithm is far simpler, has higher breakdown, and performed better in simulation than the minimax method. However, both tree-based methods performed substantially worse

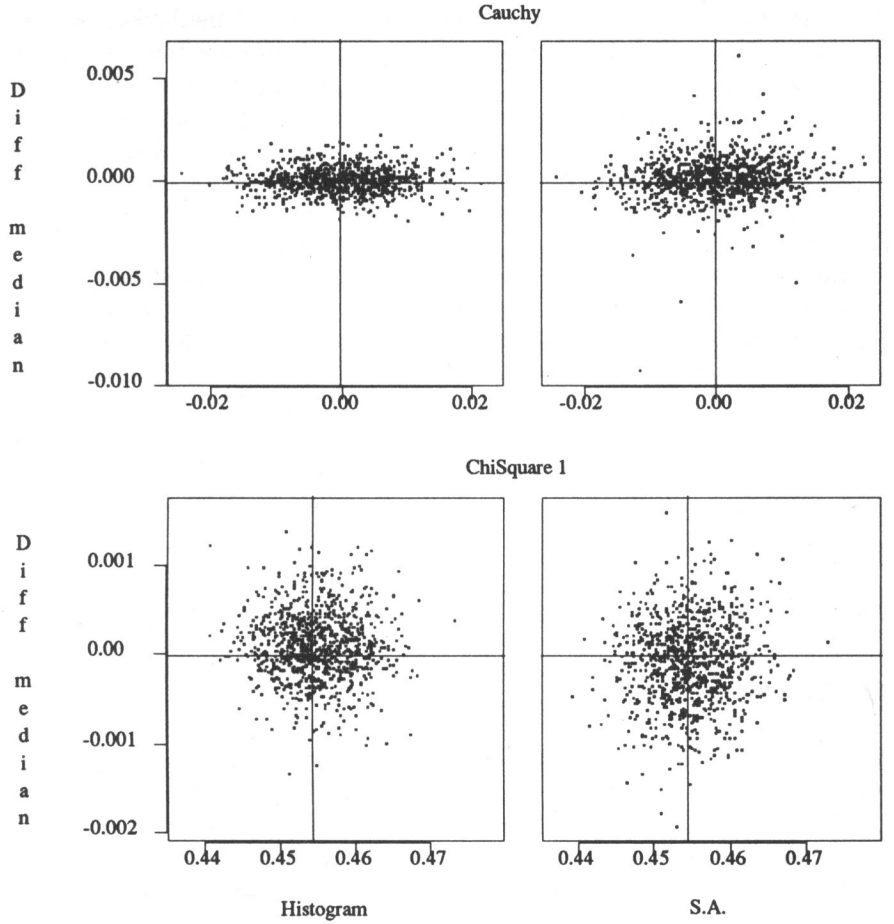

Figure 4: Comparision of the histogram and S.A. methods. The plots in the first row compare the methods on cauchy data, while the second row uses the χ_1^2 data. In each panel, the x-axis shows the estimated median, while the y-axis shows the difference between the estimate and the sample median.

than the stochastic approximation and histogram-based methods. Both of these methods produced estimates matching the sample median in their accuracy, and, data values used for startup aside, both have high breakdown. Our expermiments have shown similar results for other quantiles. We believe the histogram method is to be preferred for its conceptual simplicity, ease of implementation and computational efficiency.

Many variations on the histogram estimator presented here are possible. For example, Krieger and Gastwirth (1984), show that, at least for unimodal distributions, retaining bin means as well as counts allows one to narrow the search for a sample quantile to a bin sub-interval. Rather than the simple linear interpolation scheme we described, a quadratic or any higher-order interpolation of the empirical c.d.f. at the bin endpoints could be performed to obtain the quantile estimate. Attempts could also be made to use the first m observations to construct non-equispaced bins, which are denser in the region where the target quantile is likely to fall.

Acknowledgement

We wish to thank an anonymous referee for useful comments on this paper.

References

Apple Computer Inc. (1992) *Macintosh Common Lisp Reference.*

Baeza-Yates, R., and Gonnet, G.(1991) *Handbook of Algorithms and Data Structures*, 2nd edition, Addison-Wesley, Reading, Mass.

Blum, M., Floyd, R.W., Pratt, V., Rivest, R.L. and Tarjan R.E. (1973) Time-Bounds for Selection, *Journal of Computer and System Sciences*, 7(4), 448–461.

Chao, M.T and Lin, G.D. (1993) The Asymptotic Distributions of the Remedians, *Journal of Statistical Planning and Inference*, 37, 1–11.

Dunn, C. L. (1991) Precise Simulated Percentiles in a Pinch. *The American Statistician*, 45, 207–211.

Hatzinger, R. and Panny, W. (1993) Single and Twin Heaps as Natural Data Structures for Percentile Point Simulation Algorithms. *Statistics and Computing*, 3, 163–170.

Hoaglin, D.C., Mosteller, F. and Tukey, J.W. (1983) (editors) *Understanding Robust and Exploratory Data Analysis*, John Wiley, New York.

Krieger, A.M., and Gastwirth, J.L. (1984) Interpolation from Grouped Data for Unimodal Densities, *Econometrica*, 52(2), 419–426.

Marsaglia, G. (1972) The Structure of Linear Congruential Sequences, *Applications of Number Theory to Numerical Analysis*, S.K. Zaremba, ed., Academic Press, London, pp. 249–285.

Mahmoud, H., Modarres, R. and Smythe, R. (1994) Analysis of Quickselect: An Algorithm for Order Statistics, *RAIRO, Theoretical Informatics and its Applications*, to appear.

Munro, J.I., and Paterson, M.S. (1980) Selection and Sorting with Limited Storage, *Theoretical Computer Science*, 12, 315–323.

Pearl, J. (1981) A Space-Efficient On-Line Method of Computing Quantile Estimates, *Journal of Algorithms*, 2, 164-177.

Pfanzagl, J. (1974) Investigating the Quantile of an Unkown Distribution, *Contributions to Applied Statistics*, W.J. Ziegler, ed. Birkhauser Verlag, Basel, pp. 111–126.

Rao, C.R., (1973) *Linear Statistical Inference and its Applications*, John Wiley, New York.

Rousseeuw, P.J., and Bassett, G.W (1990) The Remedian: A Robust Averaging Method for Large Datasets, *Journal of the American Statistical Association* 85(409), 97–104.

SAS Institute, (1991) *SAS User's Guide* , SAS Institute, Cary, NC.

Steele, G.L. (1990) *Common Lisp, The Language*, 2nd ed. Digital Press, Bedford Mass.

Tierney, L. (1983) A Space-Efficient Recursive Procedure for Estimating a Quantile of an Unknown Distribution, *SIAM Journal on Scientific and Statistical Computing*, 4(4): 706–711.

Tukey, J.W. (1978) The Ninther: A Technique for Low-Effort Robust (Resistant) Location on Large Samples, in *Contributions to Survey Sampling and Applied Statistics in honor of H.O. Hartley*, H.A. David (editor), Academic Press, New York, pp. 251–257.

Weide, B.W. (1978) Space-Efficient On-Line Selection Algorithms, in *Computer Science and Statistics: Proceedings of the 11th Symposium on the Interface*, 308–311.