

Data skeletons: simultaneous estimation of multiple quantiles for massive streaming datasets with applications to density estimation

James P. McDermott · G. Jogesh Babu ·
John C. Liechty · Dennis K.J. Lin

Published online: 31 July 2007
© Springer Science+Business Media, LLC 2007

Abstract We consider the problem of density estimation when the data is in the form of a continuous stream with no fixed length. In this setting, implementations of the usual methods of density estimation such as kernel density estimation are problematic. We propose a method of density estimation for massive datasets that is based upon taking the derivative of a smooth curve that has been fit through a set of quantile estimates. To achieve this, a low-storage, single-pass, sequential method is proposed for simultaneous estimation of multiple quantiles for massive datasets that form the basis of this method of density estimation. For comparison, we also consider a sequential kernel density estimator. The proposed methods are shown through simulation study to perform well and to have several distinct advantages over existing methods.

Keywords Sequential quantile estimation · Sequential density estimation · Online algorithms · Sequential algorithms · Cubic spline

J.P. McDermott (✉) · G.J. Babu
Department of Statistics, The Pennsylvania State University,
326 Thomas Building, University Park, PA 16802, USA
e-mail: jp.mcdermott@gmail.com

J.C. Liechty
Departments of Marketing and Statistics, The Pennsylvania State
University, 407 Business Building, University Park, PA 16802,
USA

D.K.J. Lin
Department of Supply Chain and Information Systems,
The Pennsylvania State University, 483 Business Building,
University Park, PA 16802, USA

1 Introduction

Massive streaming datasets, especially when the data is in the form of a continuous stream with no fixed length, are becoming more and more common in the modern information age. They arise from sources as diverse as large call centers, internet traffic data, telephone traffic data, sales transactional records, or satellite feeds. These extreme size, sequential data sources present a clear need to be able to process data accurately and efficiently before becoming inundated by a continually growing store of data. We investigate exploratory data techniques that may be applied sequentially to either a static massive dataset of fixed size or a stream of data where the data must be processed and then discarded to free up room for the newly arriving data.

1.1 Density estimation

Density estimation is an important and long studied problem. In this paper, we address the problem of density estimation when the dataset is not of a fixed size, rather it is in the form of a continuous stream of data with a non-fixed sample size.

Applications for density estimates include, but are not limited to, nonparametric density estimation, cluster analysis, and estimation of various quantities that depend on the density such as the hazard rates (Silverman 1998). Additionally, the density can often give a more intuitive picture of such characteristics as the skewness of the distribution or the number of modes. A further advantage of having an estimate of the density is ease of interpretation for nonstatisticians. As noted in Silverman (1998), many statisticians would explain the normal distribution by drawing the familiar bell-shaped curve rather than drawing the cumulative distribution function or writing out the explicit formula for the density of the normal distribution.

However, to employ the usual kernel density estimation methods it is assumed that the size of the dataset is known and the points at which the density is to be estimated must be specified *a priori*. As such, alternative methods are required. Hence we propose estimation of the density at specific quantile values by taking the derivative of a smooth spline fit through a set of quantile values. We will first address the problem of the simultaneous estimation of multiple quantiles and then use this new methodology to address the problem of density estimation.

1.2 Estimation of multiple quantiles

It is well known that given an i.i.d. sample X_1, \dots, X_n with a common distribution function F , the empirical distribution function (edf) is a consistent estimator of the cumulative distribution function (cdf), where we define the edf as

$$F_n(x) = \frac{1}{n} \sum_{i=1}^n I(X_i \leq x), \quad -\infty < x < \infty.$$

Further, by the Glivenko-Cantelli theorem (see Billingsley 1986), we have the stronger consistency result that, with probability 1,

$$\begin{aligned} & \sup_x |F_n(x) - F(x)| \\ &= \max_{1 \leq i \leq n} \left(\max \left(\left| \frac{i}{n} - F(X_{(i)}) \right|, \left| \frac{i-1}{n} - F(X_{(i)}) \right| \right) \right) \\ & \rightarrow 0, \quad \text{as } n \rightarrow \infty, \end{aligned}$$

where $X_{(1)} \leq \dots \leq X_{(n)}$ is the sorted version of X_1, \dots, X_n .

Other nice properties of the edf are its unbiasedness for estimating the cdf and asymptotic normality; that is,

$$\begin{aligned} & \sqrt{n}(F_n(x) - F(x)) \xrightarrow{d} N(0, F(x)[1 - F(x)]), \\ & \text{as } n \rightarrow \infty. \end{aligned}$$

However, when the size of the dataset is such that sorting to obtain the edf is impractical, both in terms of storage space and computation time, we need another method to estimate the cdf that will ideally have some of these same nice properties.

Although there have been several methods proposed for the low-storage sequential estimation of a single quantile (Liechty et al. 2003; Tierney 1983; Dunn 1991; Rousseeuw and Bassett 1990; Pearl 1981; Jain and Chlamtac 1985), only Raatikainen (1987, 1990) gives a method for the simultaneous estimation of more than one quantile. Raatikainen (1987) gives an extension of an algorithm given by Jain and Chlamtac (1985) that is utilized for the simultaneous estimation of multiple quantiles. The method will return a value that has been arithmetically manipulated and tracked as part

of the algorithm instead of an actual observation. It will return only $2k + 3$ quantile estimates for the k prespecified by the user. And finally the method performs poorly for extreme tail quantiles for heavy-tailed distributions such as the Cauchy. In contrast, we propose a method that returns actual observations from the dataset as the estimates, returns km estimates of quantiles for the k prespecified by the user (where m in practice is chosen to be 50), and performs very well for extreme tail quantiles even for heavy-tailed distributions. For these reasons, we will not show a full simulation comparison with this method, but we mention that in all simulation experiments the proposed method outperforms the method of Jain and Chlamtac (1985) particularly in the extreme tails of a Cauchy.

There are three main advantages to simultaneously estimating a set of k quantiles rather than merely running the single quantile estimation method k times. First there is a significant savings in computation time. Second, the proposed method is more accurate and the results less variable than the single quantile method. And a third advantage is that the output gives a total of km points in ascending order from the minimum to the maximum of the entire dataset, each of which will have an estimated rank. These km points with their ranks can then be used to obtain an estimate of the entire cdf through curve fitting techniques. The resulting fitted curve can then be used for other purposes such as density estimation since we have the functional form of the curve.

1.3 Overview and organization of paper

The paper is organized as follows. In Sect. 2, we propose a new method for the simultaneous estimation of an arbitrary set of quantiles from an unknown distribution for massive datasets. We give a brief overview of the algorithm used followed by a detailed description of the method. In Sect. 3, we discuss potential applications. We close with a detailed description of the simulation studies conducted in Sect. 4 followed by concluding comments in Sect. 5.

2 Estimation of multiple quantiles

Modern relational database management systems utilize low-storage quantile summaries for use in query optimization (Manku et al. 1999). There is a great amount of interest in this particular problem in the private sector as evidenced by the large body of current research being produced by such companies as IBM (Poosala et al. 1996), Bell Labs (Chen et al. 2000), and Microsoft Research (Manku et al. 1998).

Network routing decisions, and hence quality of service for the network users (see Kesidis 1999), could be improved by having more accurate summaries of the distributions of

the historical traffic data. As noted in Dunn (1991), another application is in the computation through simulation of critical values and percentile points of new statistics whose distributions are unknown. A further application could be in the summarization of MCMC analysis where simulations routinely generate massive amounts of data.

In this section, we propose a method for simultaneous estimation of multiple quantiles for massive datasets. This proposed method, which we call the *data skeleton* (DS) method, is an extension of the method of proposed by Liechty et al. (2003) where the authors give a low-storage sequential algorithm for the estimation of a single quantile. This method uses estimated ranks and assigned weights to calculate a score for every data point that determines which points to keep and which to drop as each new data point is observed. The estimated ranks are calculated by linear or exponential interpolation (see overview of algorithm below). The weights are determined by taking the nearest neighbor distance in the ranks between adjacent points. The result of this is that a new point falling “close” to an existing point in the tracking array is penalized and is hence more likely to be dropped.

For output, this method will give a set of points “chosen” by the algorithm with each point having an associated *estimated rank*. We may think of this as a “skeleton” of the empirical distribution function. That is, we will have a very small subset of points along the actual edf, but not the entire edf.

2.1 Overview of the data skeleton algorithm

1. Sort the first km points, where k is the number of subarrays of size m being used to estimate the k quantiles, and assign estimated ranks and weights to these points;
2. Find within which of the k subarrays of points the next observation falls and assign an estimated rank and weight to this point accordingly with a point falling between two subarrays being assigned to its nearest neighbor;
3. Calculate scores for all m points in this subarray and for the new point and drop the point with the highest score retaining the ordering in the subarray;
4. Repeat steps 2–3 for all remaining observations;
5. Take the estimates from the final set of observations.

Step 1 consists of loading an array with the first km observations and sorting this array. This initial array is composed of k subarrays of size m . We will refer to this array as the *tracking array*. Each subarray will be dedicated to the estimation of a single quantile. We then assign initial *estimated ranks* to each point equal to its actual rank in this original sample after it has been sorted, i.e. the first element has estimated rank 1 and the last element has estimated rank km . We assign an initial *weight* of 1 to each point. This weight

is a measure of the relative importance or value of a given point. At the first step, all points are equally valuable.

The data skeleton algorithm proceeds in step 2 by observing the next point in the dataset and finding within which of the k subarrays it falls. This determines which quantile the point will be used to estimate. For example, if the point falls anywhere between the 1st and m th points, then it will be used to estimate the lowest quantile being estimated, if it falls anywhere between the $m + 1$ st and $2m$ th points, then it will be used to estimate the 2nd lowest quantile, and so on. If the point falls between two subarrays, e.g. between the m th and $m + 1$ st points, then it is assigned to its nearest neighbor and then used for that associated quantile. Once the proper subarray has been determined, we then find between which two of the m points the new point falls between and assign an estimated rank and weight to the new point. Note here that the estimated ranks of all points among the km that are greater in value than the new point are incremented by 1. The new point, denoted by x_* , is then assigned an estimated rank, denoted by r_* , according to the following linear interpolation formulas.

- If the new point is a new maximum, i.e. $x_* > x_{km}$, the new point becomes the new maximum and the old maximum becomes the new point. Let $r_* = r_{km}$ and then let $r_{km} = r_{km} + 1$.
- If the new point is a new minimum, i.e. $x_* < x_1$, the new point becomes the new minimum and the old minimum becomes the new point. Let $r_* = r_1$ and then let $r_1 = r_1 - 1$.
- If the new point is just less than the maximum, i.e. $x_{km-1} < x_* < x_{km}$, then

$$r_* = r_{km-1} + \frac{r_{km} - r_{km-1}}{1 - \delta} (1 - e^{-\lambda(x_* - x_{km-1})}), \tag{1}$$

where

$$\delta = e^{-\lambda(x_{km} - x_{km-1})}, \quad \lambda = -\frac{\log(1 - q_2(1 - \delta))}{q_1(x_{km} - x_{km-1})},$$

and q_1 and q_2 are set by the researcher (see Liechty et al. 2003).

- If the new point is just greater than the minimum, i.e. $x_1 < x_* < x_2$, then

$$r_* = r_2 + \frac{r_1 - r_2}{1 - \delta} (1 - e^{-\lambda(x_2 - x_*)}), \tag{2}$$

where

$$\delta = e^{-\lambda(x_2 - x_1)}, \quad \lambda = -\frac{\log(1 - q_2(1 - \delta))}{q_1(x_2 - x_1)},$$

and q_1 and q_2 are, again, set by the researcher.

- If the new point falls anywhere else, i.e. $x_2 < x_* < x_{km-1}$, then

$$r_* = r_i + (r_{i+1} - r_i) \frac{x_* - x_i}{x_{i+1} - x_i}.$$

The weight, denoted by w_* , for the new point, denoted by x_* , is assigned according to the following formula.

$$w_* = \min(r_{i+1} - r_*, r_* - r_i),$$

and the score is then calculated as

$$s_* = \frac{|r_* - \text{target}|}{w_*}.$$

Here *target* refers to the target rank at that point, say after observing n' points, and is simply calculated as $n' \cdot p$. When determining which point to drop, the current minimum and maximum are by default always kept.

The nonlinear functions, as described in (1) and (2), are exponential curves which are designed so that the estimated ranks quickly go to the rank associated with the maximum or minimum element as the new point moves towards either of these elements. These nonlinear functions are necessary to account for the tail behavior of extremely heavy-tailed distributions such as the Cauchy. For further discussion of the motivation for the exponential curves, see Liechty et al. (2003).

2.2 Comparison of single and multiple quantile methods

We begin by comparing our proposed DS method that simultaneously computes the quantile estimates to the method given by Liechty et al. (2003) that computes a single quantile. In this comparison, we show that there is a distinct advantage to simultaneously estimating a set of quantiles rather than estimating them one at a time. In our simulation studies, we have considered many commonly studied distributions such as the normal, Cauchy, chi-square, and mixtures of normals and achieve similar results for all distributions investigated. We would like to thank the referee for making many helpful suggestions that have greatly improved the quality of the graphs. We present results for the Cauchy in Figs. 1 and 2. The extreme heavy-tailed nature of the Cauchy presents certain problems for tail quantile estimation. We present these results to demonstrate the effectiveness of our method for the estimation of extreme tail quantiles even for very heavy-tailed distributions, but note that similar performance is observed for all distributions studied. The averaged point estimates of the quantiles for the simultaneous and the single quantile method are not meaningfully different. Although we have chosen to focus on the standard Cauchy distribution for our comparison since this was the most difficult case to handle, our studies have shown

similar results for other common distributions such as the standard normal, chi-square, and mixtures of normals.

The first advantage is that the amount of time needed to estimate the quantiles simultaneously is much less than it would be to estimate them separately. For example, if we let T be the average time in seconds to execute the single quantile estimation method given by Liechty et al. (2003), then to compute k quantiles using the single quantile method will take kT seconds on average. However, using the simultaneous estimation method to compute the same k quantiles will take on average only $T + (k - 1)t$ seconds, where $t \ll T$. That is, for every additional quantile beyond the first, the simultaneous method adds some number of seconds t that is much less than T .

The second advantage is that simultaneous estimation yields improved accuracy over the single quantile estimation method when compared to the true sample quantile. For example, we define the *mse ratio* as the ratio of the given method's mean squared error to the mean squared error of the sample quantile. Ideally we would like a ratio as close to 1 as possible. A ratio close to 1 means the proposed method has approximately the same variability as the sample quantile. In Fig. 1, we can see that the mse ratio is improved in every case by using the simultaneous estimation method. Similarly we define a measure called *mse** as the average squared deviation of the proposed estimator from the sample quantile. This measure differs from the usual mean squared error in that we are interested in deviations from the sample quantile instead of the population quantile. With this measure we are looking at how closely the proposed method approximates the accuracy of the sample quantile. Here we would like to see *mse** as close to 0 as possible. In Fig. 2, we can see that the *mse** is also improved everywhere by using the simultaneous estimation method.

3 Density estimation

Density estimation for massive streaming datasets poses particular problems. The traditional kernel density estimation method requires that the points at which the density is to be estimated be specified a priori before the algorithm is executed. Further, the optimal bandwidth selection depends upon knowing the total sample size. When the data is in the form of a continuous data stream and the total sample size is not known, new estimation methods are needed.

One important thing to note is that while we have very accurate estimates of the k quantiles that were prespecified, we also have information about all of the other points that the algorithm kept but did not choose as estimates. That is, each of the km points that the DS algorithm gives as output has an associated estimated rank. As mentioned in the introduction, we are interested in obtaining an estimate of the entire

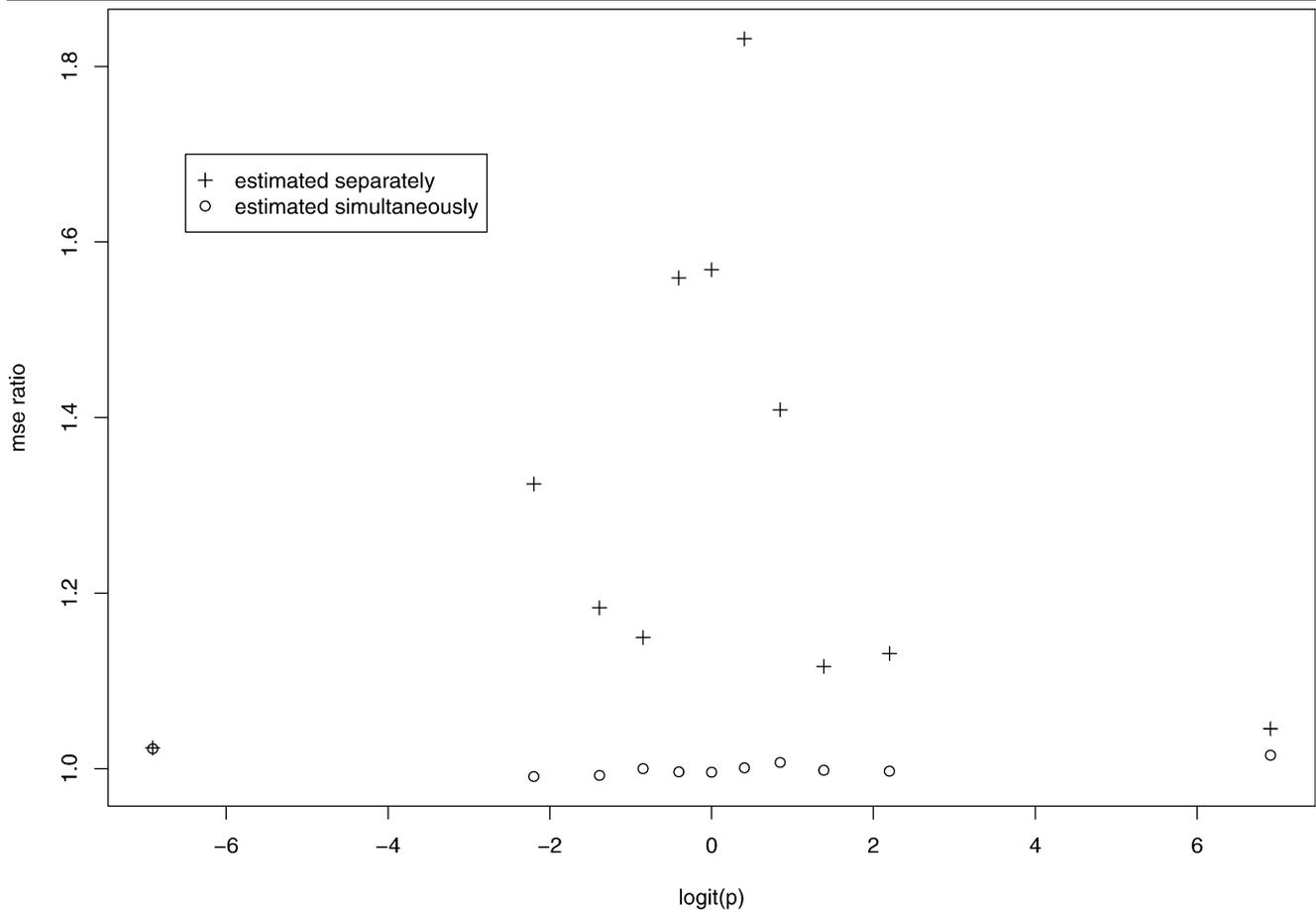


Fig. 1 mse ratio vs. logit(p). Standard Cauchy example

unknown cdf. To that end we propose fitting a cubic spline to the set of km points with associated estimated ranks. By doing so we will have an estimate of the underlying cdf in a functional form which we may then work with analytically.

Further, while these estimated quantiles are useful and informative on their own, some might find it more useful to have information about the density as well since this can give a more intuitive picture of such characteristics as the skewness of the distribution or the number of modes. Hence we will explore the possibility of density estimation through taking the derivative of the cubic spline fit, thereby having access to density estimates over the entire range of the original sample. We also give an alternative method of density estimation that uses a sequential kernel density estimation technique.

3.1 Sequential kernel density estimation

The well known *kernel density estimator* (see Silverman 1998) defined by

$$\hat{f}(x) = \frac{1}{nh} \sum_{i=1}^n K\left(\frac{x - x_i}{h}\right),$$

where h is the *bandwidth*, x is the point at which we are estimating the density, and the x_i are the observations, and the *kernel function* K is some nonnegative function satisfying the condition $\int_{-\infty}^{\infty} K(x) dx = 1$. The optimal bandwidth is determined by some function of the total sample size n . For example, in Silverman (1998), the author recommends using a bandwidth of $h = 0.9An^{-1/5}$, where $A = \min(\text{standarddeviation}, \text{interquartilerange}/1.34)$.

Hence if we wish to use the traditional kernel density estimation method to obtain estimates of the density at the prespecified but unknown quantiles, ξ_1, \dots, ξ_k , we need to know n in order to choose a reasonable bandwidth and we need good estimates of the ξ_i . However, even if these quantities were available to us, the traditional kernel density estimator would still be undesirable because we have to make more than one pass through the data: one pass to get good estimates of the quantiles and a second pass to estimate the density at those points. Ideally we would like to be able to sequentially update both the quantile estimates and the estimates of the density at those quantile values to avoid either storing the whole dataset or making more than one pass through the dataset.

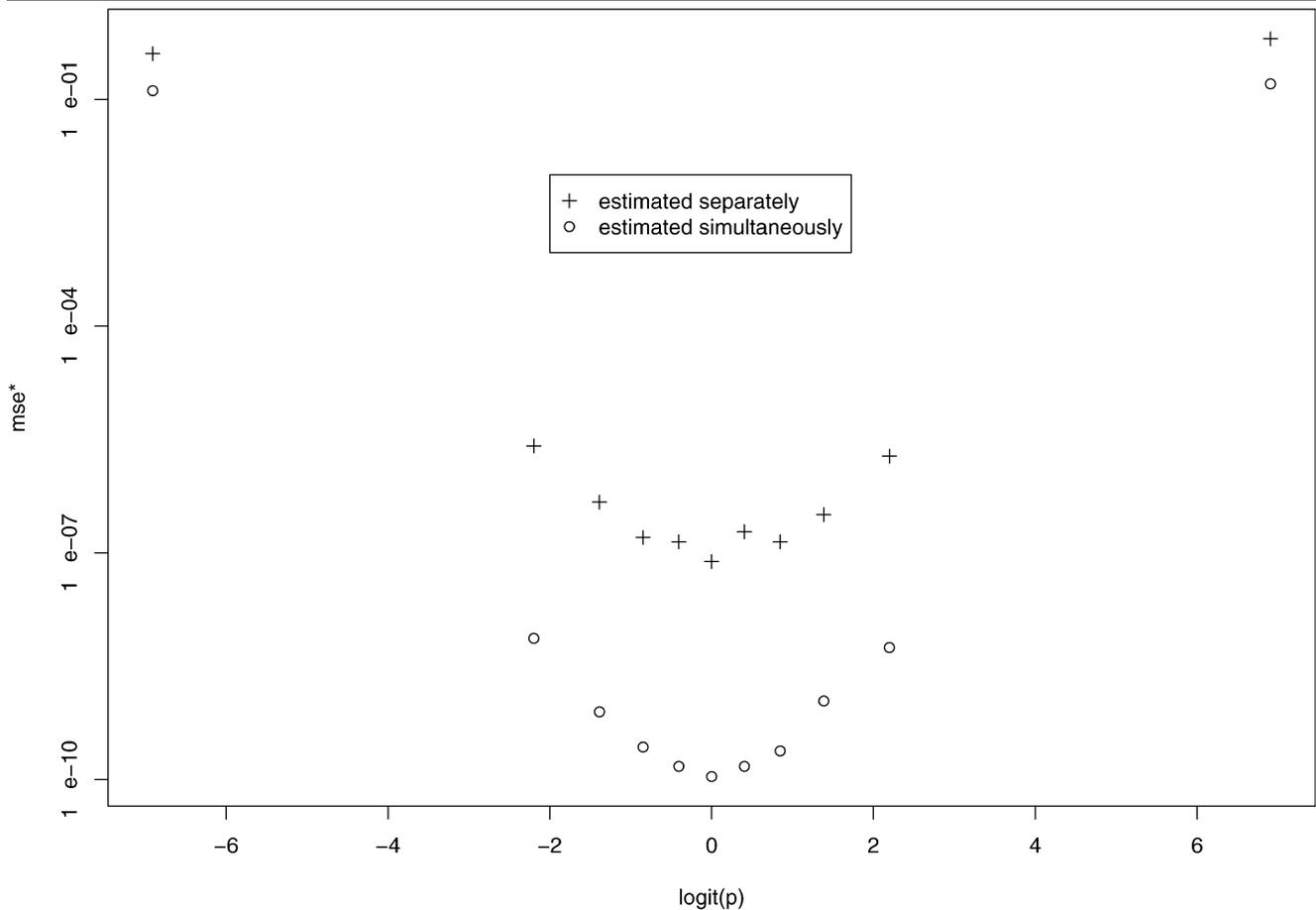


Fig. 2 mse* vs. logit(p). Standard Cauchy example

Tierney (1983) presents a sequential version of the kernel density estimation method. He uses the sequential density estimates as part of a sequential quantile estimation method and proves the convergence of the density estimates. The bandwidth is treated as a sequence of bandwidths that tends to zero rather than a fixed constant. He defines a sequence $\{h_n\}$ to be of the form $\{n^{-\beta}\}$, where $0 < \beta < 1$. Although Tierney sets h_n to be the sequence $\{n^{-1/2}\}$, we will use $\{n^{-1/5}\}$ to incorporate the recommended bandwidth suggested in Silverman (1998), as it gives us the optimal rate of convergence. As in Tierney (1983), we will use the rectangular kernel defined by

$$K(t) = \begin{cases} \frac{1}{2} & \text{if } |t| < 1, \\ 0 & \text{otherwise.} \end{cases}$$

Hence we define our sequential kernel density estimator as

$$\hat{f}_n(x) = \frac{1}{n} \sum_{i=1}^n K\left(\frac{x - x_i}{h_i}\right) = \frac{1}{n} \sum_{i=1}^n I(x, x_i, h_i)/(2h_i),$$

where

$$I(x, x_i, h_i) = \begin{cases} 1 & \text{if } |x - x_i| \leq h_i, \\ 0 & \text{if } |x - x_i| > h_i \end{cases}$$

and x is the point at which the density estimate is being calculated.

3.2 Smoothing and interpolating cubic splines

A cubic smoothing spline, \hat{F} is defined as the unique function, over all those functions with continuous first and second derivatives, that minimizes the penalized sum of squares

$$\sum_{i=1}^{km} \{y_i - F(x_i)\}^2 + \alpha \int_a^b \{F''(t)\}^2 dt,$$

where α is the smoothing parameter (see Green and Silverman 1994).

For the case of an interpolating cubic spline, our goal is to find a smooth curve \hat{F} such that \hat{F} interpolates the points (x_i, y_i) , i.e. $\hat{F}(x_i) = y_i$. That is, we wish to find an interpolating cubic spline such that $\hat{F}(x_i) = r_i/n$ for $i = 1, \dots, km$

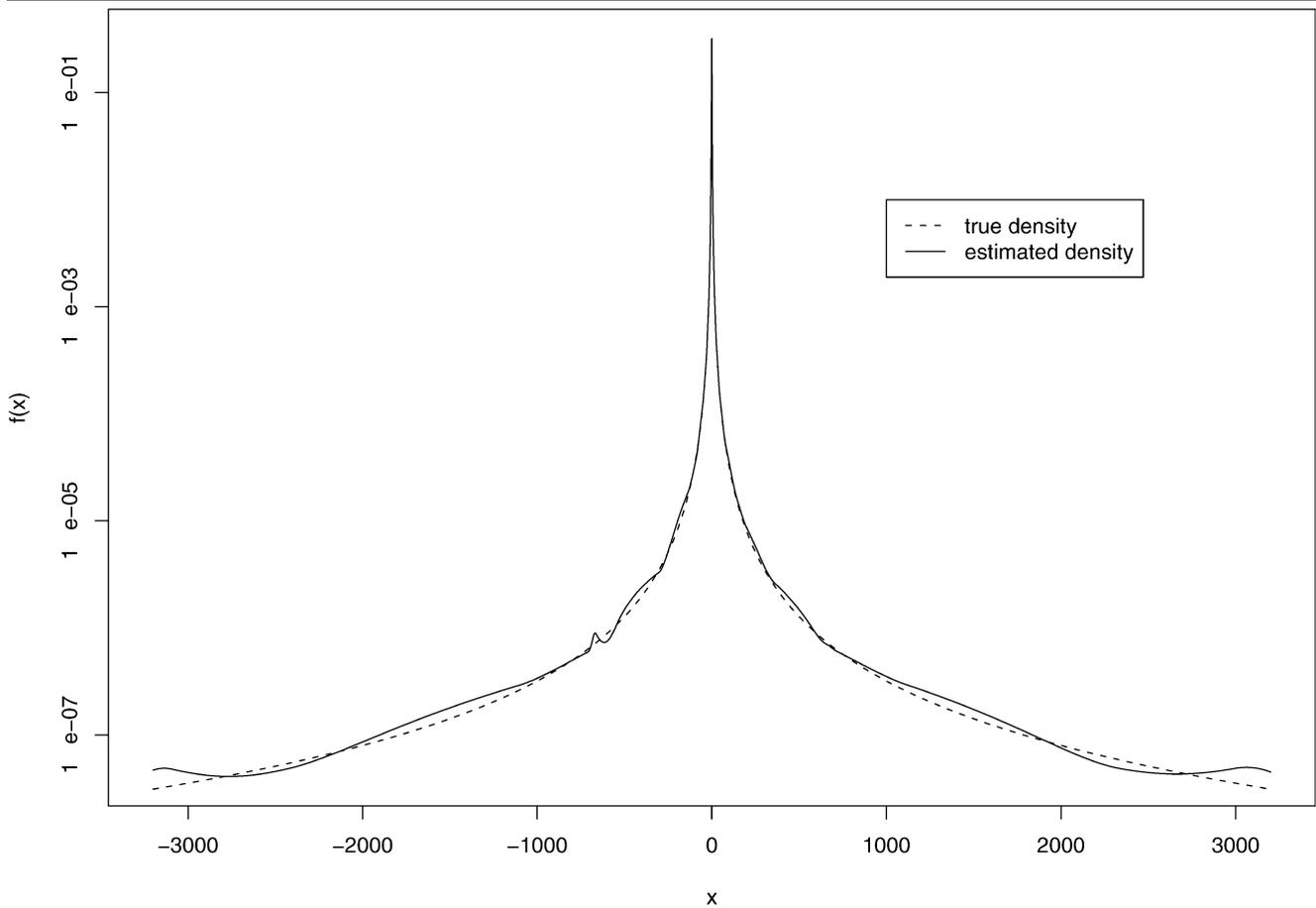


Fig. 3 Density estimation—standard Cauchy example (10,000,000 observations)

and where the r_i are the estimated ranks obtained by the DS algorithm.

In general, suppose we are given values y_1, \dots, y_r at the points x_1, \dots, x_r . We wish to fit a smooth curve \hat{F} through the points (x_i, y_i) . In the cubic spline literature (see Green and Silverman 1994) the points x_i are called the *knots*. The function \hat{F} defined on some interval $[a, b]$ such that $a < y_1 < \dots < y_r < b$ is a cubic spline if \hat{F} is a cubic polynomial on each of the intervals $(a, y_1), (y_1, y_2), \dots, (y_{r-1}, y_r), (y_r, b)$ and if \hat{F} has a continuous 1st and 2nd derivative at each x_i . We will explore two types of cubic splines: smoothing and interpolating.

As a result of running the DS algorithm, we will have as output a set of km points covering the entire range of the data that was seen, where k is the number of quantiles estimated and m is the number of points in each subarray dedicated to the estimation of a single quantile. Each point will have an estimated rank associated with it. The first point and last points will be the minimum and the maximum of all points observed with the first point having rank 1 and the last point having rank n . Hence our goal will be to fit a cubic spline through the set of km points given as output by the DS al-

gorithm. To that end, we will treat the x_i from this set of km points as the knots that we will use to fit either a smoothing or an interpolating cubic spline. A desirable property of both cubic splines is that by definition both their first- and second-derivatives exist. In other words, assuming the data comes from a continuous distribution, we may obtain estimates of the unknown density by taking the 1st derivative of the cubic spline fit, \hat{F} . For both smoothing and interpolating cubic splines in our application, we are given km values y_1, \dots, y_{km} at the points x_1, \dots, x_{km} . Here the x_i are the km points from the tracking array and the y_i 's are the r_i/n 's that are given as output from the DS algorithm.

It is commonly known that $f(x) = F'(x)$, assuming of course that the density f exists. By definition, the cubic spline has a first derivative that exists everywhere along the curve, including at the knots themselves. As in Wahba (1975), we take the derivative of this function, $\hat{F}'(x)$, to obtain a pointwise estimator of the density, $\hat{f}(x)$; i.e. $\hat{f}(x) = \hat{F}'(x)$.

To guarantee a positive density, we would merely have to first guarantee a strictly increasing estimate of the CDF. This would guarantee that the resulting derivative of the CDF,

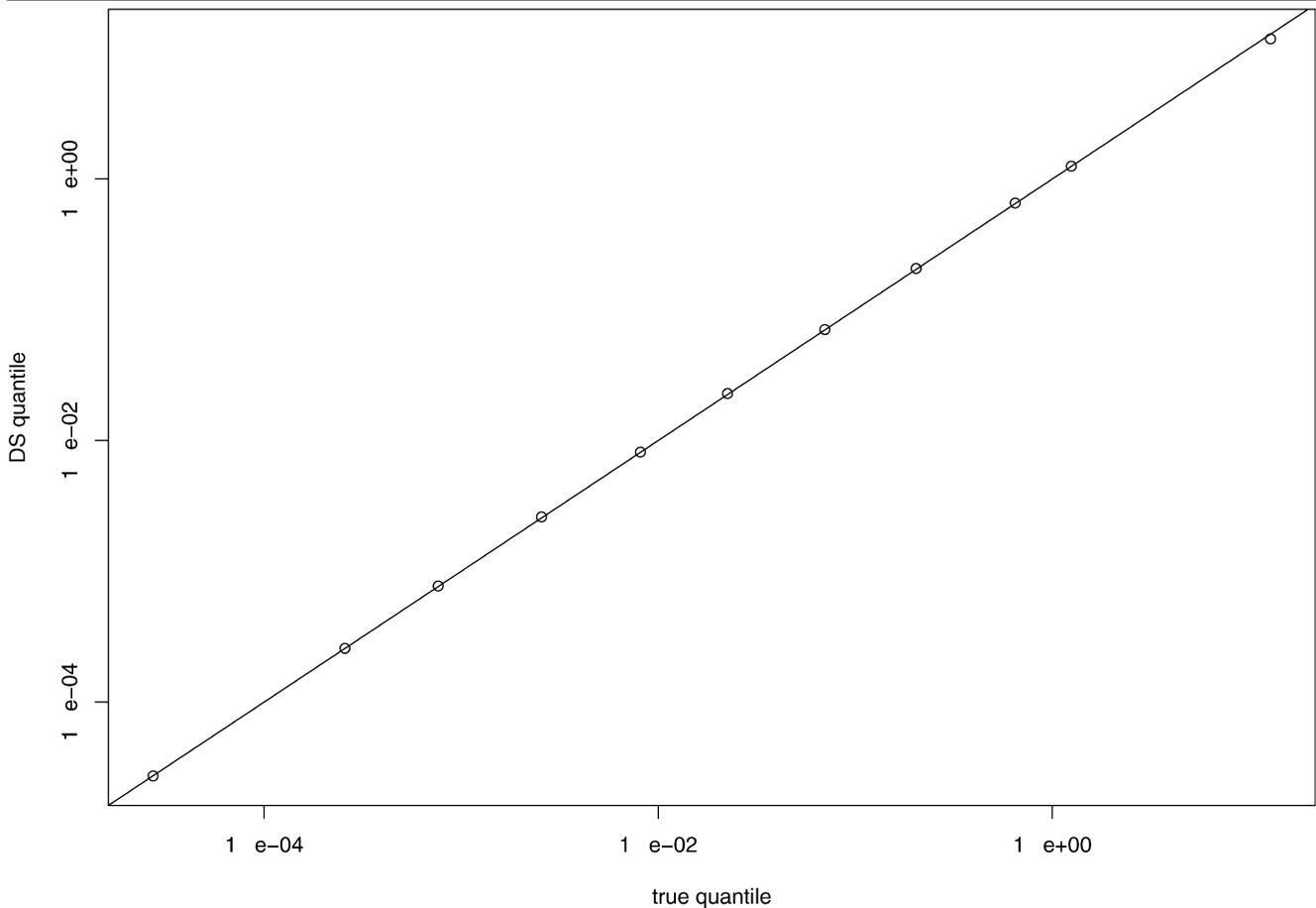


Fig. 4 Log-scaled Q–Q plot of DS method vs. true quantiles. Standard normal with 10,000,000 observations and 1000 replications

the density, is strictly positive. One possible approach to this are to use the COBS library in R (standing for Constrained B-Splines). This method allows for the imposition of constraints to a spline fit. In particular, this method allows to constraint the fitted spline to be strictly increasing. This method also allows the fit to be constrained to have a minimum of 0 and a maximum of 1, as with a true CDF for a continuous distribution. Another possible approach is to use the MGCV library in R (standing for Multiple smoothing parameter estimation by Generalized Cross Validation). This also allows the fit to be constrained to be strictly increasing. Since we never encountered these situations in our work with the cubic smoothing splines, we did not find it necessary to employ these other more complex and computationally intensive methods.

3.3 Comparison of two density estimation methods

One advantage of this method of density estimation over the sequential kernel density method proposed in the previous section is that we are not restricted to density estimates just at the chosen quantile values. Since we now have an estimate of the cdf over the entire range of the dataset that has

two continuous derivatives by definition, we may take the derivative and obtain an estimate of the density at any point.

A second advantage is that the cubic spline derivative method requires no additional computation: the set of points through which we fit the spline is a byproduct of the DS algorithm. In contrast, the sequential kernel method requires additional computation upon seeing each new data point. For example, using a sample of size 10,000,000 and estimating 13 quantiles and the associated density estimates at those quantiles, the sequential kernel method takes over 10 times as long to compute as the cubic spline derivative method takes. In fact, the cubic spline derivative method adds no appreciable computation time to the DS algorithm computation time since all that is required is to fit a cubic spline and take its derivative at the desired points *after* the DS algorithm has been run.

An additional study conducted was a comparison between the two density estimation methods presented: the sequential kernel density estimation method and the method utilizing the derivative of a cubic spline fit. We simulated data from the standard normal distribution: the sample size was 10,000,000 and the number of replications was 100. For

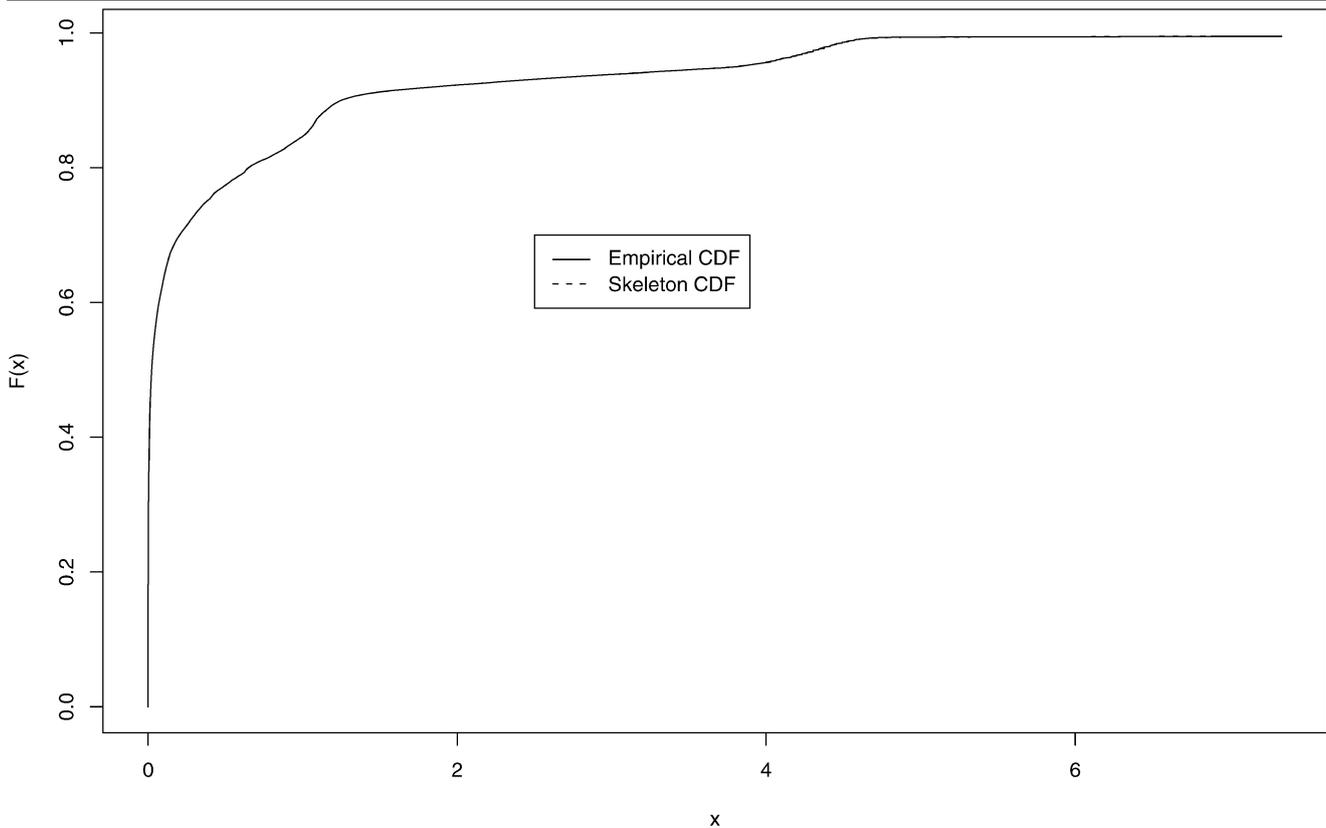


Fig. 5 Comparison of empirical cdf and skeleton cdf (LBL-PKT-4 dataset)

each dataset we computed the sequential kernel density estimate presented above at each of the 9 deciles and at 4 tail quantiles, i.e. at the .001th, .01th, .1th through .9th, .99th, and .999th quantiles. Since we do not know the values of these quantiles, we must update the estimates of them sequentially and plug them into the sequential kernel density estimator. Thus after observing each new point we must first update our estimates of the quantiles and then use these estimates to update our estimates of the density at these quantile values. These estimates are then compared to the cubic spline derivative estimates of the density.

Both methods give consistent estimates of the density at the quantiles. The sequential kernel density estimates are slightly more accurate, as exhibited by a marginally lower mean squared error. However this increased accuracy is offset by the tenfold increase in computation time required to execute this method. A further drawback for the sequential kernel method is that the values at which the density is to be estimated must be specified before the algorithm is run and we will get density estimates only at those points. In contrast, the cubic spline derivative method does not require that the points at which the density is to be estimated be specified *a priori*.

In Fig. 3, we give an example of the density estimate obtained by the DS algorithm for a standard Cauchy. The

example is computed by running the DS algorithm on a sample of size 10,000,000 with the 29 quantiles specified corresponding to $\mathbf{p} = (0.0001, 0.0005, 0.001, 0.005, 0.01, 0.05, 0.1, 0.15, 0.2, 0.25, \dots, 0.75, 0.8, 0.85, 0.9, 0.95, 0.99, 0.995, 0.999, 0.9995, 0.9999)$. Hence the resulting smoothing cubic spline fit is obtained by fitting the curve through $km = (29)(50) = 1450$ points. The density estimates are plotted on the log scale to demonstrate the accuracy in the tails.

4 Example: Internet traffic data

In this section we apply our DS algorithm to a real-world dataset in an attempt to see how accurately the method can handle non-simulated data. The data we are considering is internet traffic data. As an example we will analyze a dataset containing internet traffic data from the Lawrence Berkeley Laboratories. The website, <http://ita.ee.lbl.gov/html/contrib/LBL-PKT.html>, contains the dataset used in this example. A detailed description of this dataset, or trace, can be found in Paxson and Floyd (1995) along with a list of references. The trace was collected from 2PM to 4PM in January 1994. The dataset is named LBL-PKT-4. There were approximately 863,000 packets collected in this trace of which ap-

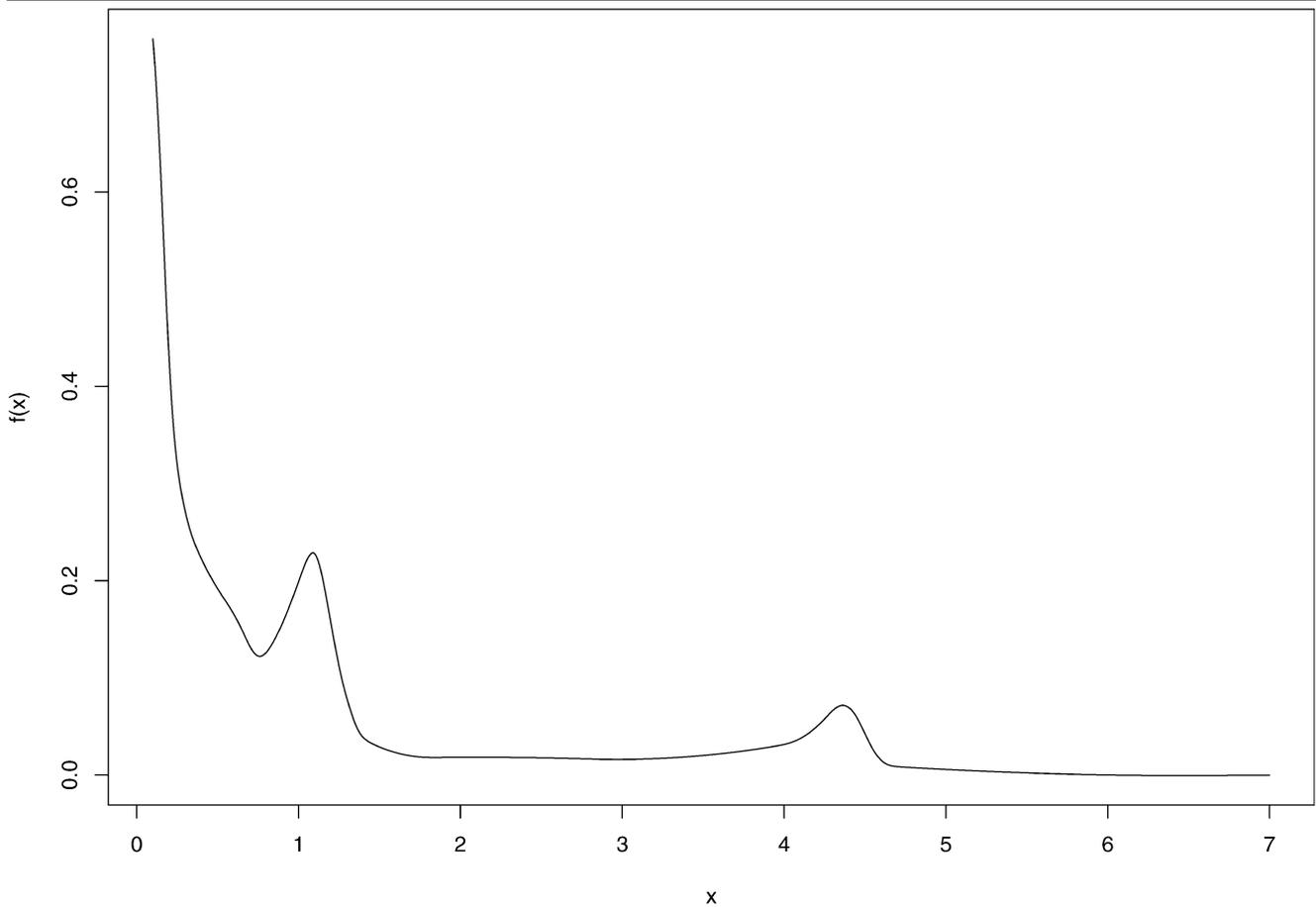


Fig. 6 Density estimate from derivatives of the cubic spline fit. LBL-PKT-4 dataset

proximately 276,000 packets were of size zero. The size zero packets are used for control purposes to let outside sources know that a packet has been received. For the purposes of studying throughput these packets can be ignored. Hence we have approximately 587,000 non-zero packets in LBL-PKT-4. Each packet will have a timestamp associated with it identifying the start time of the transfer. A derived measure of interest for this data is something called *throughput*. This is defined as the size of the packet divided by the time it took to travel to its destination. Throughput is commonly studied as a measure of network efficiency. We will be looking at the distribution of the packet-level throughput quantities for this dataset.

In Fig. 4, we give a log-scaled Q-Q plot of the true sample quantiles versus the DS estimated quantiles for the LBL-PKT-4 dataset. Graphically, the DS estimate of the quantiles are very close to the sample quantiles obtained by actually sorting the dataset. For this example, we compute the deciles plus two tail quantiles: the .001th and .999th quantiles. This example demonstrates the applicability of the proposed method to non-simulated data. Note here that although the size of our example datasets are less than

1,000,000 observations, they are also only a days worth of traffic data on a single data stream at a site with not very heavy traffic. One can easily imagine datasets growing into the gigabyte or terabyte or petabyte range as the number of days and the number of streams and the rate of traffic grow.

In Fig. 5, we present a graphical comparison of the empirical cdf of the LBL-PKT-4 dataset and the skeleton cdf of the same dataset. Visually, we are unable to detect a difference between the two. Looking at the graph, one might guess that the number of modes for the underlying population is three: one located very close to 0, a second located close to 1, and a third located between 4 and 5. Hence we might think of this datastream as originating from a mixture of three subpopulations.

Although having estimates of the quantiles might lead one to guess at the number and location of the modes, having a picture of the density gives a clearer, more intuitive understanding of the shape of the distribution. Hence in Fig. 6 we present a graphical representation of our density estimate obtained by taking derivatives of the cubic spline that was fit to the data skeleton. Here we can clearly see the three dis-

tinct modes. We note here that the locations of the modes agrees with the histogram of the dataset.

One possible application for these methods is as a diagnostic tool for detecting if a new subpopulation has entered into the datastream or if the locations of the existing modes have shifted. For example, if we continued to monitor this datastream and replotted the density estimate every hour, we may find that a fourth mode has emerged or that the mode that is currently located around 4 may have shifted up to around 5.

5 Concluding comments

We have demonstrated the effectiveness of our methods through simulation study and application to real datasets. We believe that the resulting estimates from our approach provide useful insights into the structure of massive datasets. However, there are some related issues such as the choice of which quantiles to specify for estimation and the choice of curve fitting techniques.

The choices of which quantiles and how many quantiles to be estimated will depend on the specific application to which this method is applied. For example, if one were interested mainly in the extreme tail of a distribution, one could choose to concentrate the quantiles in that area and devote less computation to other areas of the distribution. Poor choices could be made for a given application and care should be taken to choose appropriately based on the situation. The points at which the sequential kernel method estimates are to be computed need to be prespecified, but which values should one choose? *A priori*, we do not know what values the distribution will take and hence we may choose poor values at which to estimate the density. For example, we may choose equally spaced values from -1000 to 1000 , but the actual data may be distributed as Chi-square and only take positive values, or the data may be distributed as $N(0,1)$ and so we will very rarely (if ever) see values near -1000 or 1000 . The benefit of the DS method is that the range of actual values is not needed to obtain a good estimate of the entire range of the observed data.

Another issue is the situation where the distribution from which the data is arising changes over time. To track changes over time, one could possibly have multiple instances of the algorithm running with each one having been started at different times. For example, one could restart a new instance of the algorithm hourly or daily and compare the resulting estimates as each instance progresses. Similarly, as in our internet traffic example, one could track the position and number of modes of the distribution over time to see how the distribution is changing.

The code for the simulation studies was implemented using the C programming language for the computationally

intensive components, such as sorting large datasets or executing the DS algorithm, and the R statistical programming language for the analysis, such as fitting the cubic splines or taking the derivative of the splines to obtain density estimates.

Acknowledgements The authors would like to thank the referee for many helpful suggestions that have greatly improved the quality of the paper.

References

- Billingsley, P.: Probability and Measure. Wiley, New York (1986)
- Chen, F., Lambert, D., Pinheiro, J.C.: Incremental quantile estimation for massive tracking. In: Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Association for Computing Machinery, p. 10 (2000)
- Dunn, C.L.: Precise simulated percentiles in a pinch. *Am. Stat.* **45**(3), 207–211 (1991)
- Green, P.J., Silverman, B.W.: Nonparametric Regression and Generalized Linear Models. Chapman & Hall, London (1994)
- Jain, R., Chlamtac, I.: The p-square algorithm for dynamic calculation of quantiles and histograms without storing observations. *Commun. ACM* **28**(10), 1076–1085 (1985)
- Kesidis, G.: Bandwidth adjustments using on-line packet-level adjustments. In: SPIE Conference on Performance and Control of Network Systems, Boston, Sept. 19–22, 1999
- Liechty, J.C., Lin, D.K.J., McDermott, J.P.: Single-pass low-storage arbitrary quantile estimation for massive datasets. *Stat. Comput.* **13**(2), 91–100 (2003)
- Manku, G.S., Rajagopalan, S., Lindsay, B.G.: Approximate medians and other quantiles in one pass and with limited memory. In: Proceedings ACM SIGMOD International Conference on Management of Data, June, pp. 426–435 (1998)
- Manku, G.S., Rajagopalan, S., Lindsay, B.G.: Random sampling techniques for space efficient online computation of order statistics of large databases. In: Proceedings of the 1999 ACM SIGMOD International Conference on Management of Data, pp. 251–262 (1999)
- Paxson, V., Floyd, S.: Wide-area traffic: the failure of Poisson modeling. *IEEE/ACM Trans. Netw.*, pp. 226–244 (1995)
- Pearl, J.: A space-efficient on-line method of computing quantile estimates. *J. Algorithms* **2**, 164–177 (1981)
- Poosala, V., Ioannidis, Y.E., Haas, P.J., Shekita, E.J.: Improved histograms for selectivity estimation of range predicates. In: Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data, COMAD, pp. 294–305 (1996)
- Raatikainen, K.E.E.: Simultaneous estimation of several percentiles. *Simulation* **49**(4), 159–164 (1987)
- Raatikainen, K.E.E.: Sequential procedure for simultaneous estimation of several percentiles. *Trans. Soc. Comput. Simul.* **7**(1), 21–44 (1990)
- Rousseeuw, P.J., Bassett, G.W.: The mediant: a robust averaging method for large datasets. *J. Am. Stat. Assoc.* **85**(409), 97–104 (1990)
- Silverman, B.W.: Density Estimation for Statistics and Data Analysis. Chapman & Hall/CRC, Boca Raton (1998)
- Tierney, L.: A space-efficient recursive procedure for estimating a quantile of an unknown distribution. *SIAM J. Sci. Stat. Comput.* **4**(4), 706–711 (1983)
- Wahba, G.: Interpolating spline methods for density estimation I. Equi-spaced knots. *Ann. Stat.* **3**, 30–48 (1975)