

# Statistical methodology for massive datasets and model selection

G. Jogesh Babu\* and James P. McDermott

Department of Statistics, 326 Thomas Building,  
The Pennsylvania State University, University Park, PA 16802-2111, USA.

## ABSTRACT

Astronomy is facing a revolution in data collection, storage, analysis, and interpretation of large datasets. The data volumes here are several orders of magnitude larger than what astronomers and statisticians are used to dealing with, and the old methods simply do not work. The National Virtual Observatory (NVO) initiative has recently emerged in recognition of this need and to federate numerous large digital sky archives, both ground based and space based, and develop tools to explore and understand these vast volumes of data.

In this paper, we address some of the critically important statistical challenges raised by the NVO. In particular a low-storage, single-pass, sequential method for simultaneous estimation of multiple quantiles for massive datasets will be presented. Density estimation based on this procedure and a multivariate extension will also be discussed.

The NVO also requires statistical tools to analyze moderate size databases. Model selection is an important issue for many astrophysical databases. We present a simple likelihood based 'leave one out' method to select the best among the several possible alternatives. The performance of the method is compared to those based on Akaike Information Criterion and Bayesian Information Criterion.

**Keywords:** Akaike Information Criterion, Bayesian Information Criterion, streaming data, convex hull, log-likelihood, maximum likelihood, leave one out jackknife type method

## 1. INTRODUCTION

We are at the start of a new era of information-rich astronomy. Several ongoing sky surveys over a range of wavelengths are now generating data sets measured in the tens of Terabytes. These surveys are creating catalogs of objects (stars, galaxies, quasars, etc.) numbering in billions, with up to a hundred measured numbers for each object. Yet, this is just a fore-taste of the much larger datasets to come. Thus astronomy faces a revolution in data collection, storage, analysis, and interpretation of large datasets. Data are already streaming in from surveys such as the Two Micron All Sky Survey and the Sloan Digital Sky Survey, which are providing maps of the sky at infrared and optical wavelengths, respectively. The data volumes here are several orders of magnitude larger than what astronomers and statisticians are used to dealing with. This great opportunity comes with a commensurate technological challenge: how to optimally store, manage, combine, analyze and explore these vast amounts of complex information, and to do it quickly and efficiently?

Some methods of statistical inference based on massive datasets will be presented. In particular the estimation of several quantiles simultaneously using a low-storage method from streaming data will be discussed. This is useful in the estimation of the density when the data is streaming. The method uses estimated ranks, assigned weights, and a scoring function that determines the most attractive candidate data points for estimates of the quantiles. The method uses a small fixed amount of storage and its computation time is  $O(n)$ . Simulation studies show that the estimates are as accurate as the sample quantiles.

The procedure is useful in the approximation of the unknown underlying cumulative distribution function by fitting a cubic spline through the estimates obtained by this extension. The derivative of this spline fit provides an estimate of the probability density function. A multivariate extension is also briefly discussed.

---

\* E-mail: babu@stat.psu.edu; phone: 1 814 863 2837; fax: 1 814 863 7114

The Virtual Observatory also requires statistical tools to analyze moderate size databases. Model selection is an important issue for many astrophysical databases. Approximation methods such as Akaike Information Criterion (AIC) and Bayesian Information Criterion (BIC) methods are very popular among astronomers. We present a simple likelihood based 'leave one out' model selection method. This works well in comparison to AIC and BIC, even when the parameters of the curves are not known completely. Implementation of this method is very simple and does not require any new software.

## 2. LOW-STORAGE QUANTILE ESTIMATION FOR MASSIVE DATASETS

A very simple statistic like the sample mean can be estimated sequentially by updating the sum of all the observations. On the other hand, other simple statistical measures such as the median have no such easily implemented sequentially updatable representation. Theoretically, a sample quantile can be obtained by sorting the data and taking the appropriate order statistic. So sorting is needed for estimation of even simple statistical quantities such as the median (which is more robust than the mean) and the quantiles, which themselves are indispensable for estimating the parameters and their confidence bounds.

Standard sorting algorithms often require that the entire dataset be placed into memory. When confronted with databases containing millions or billions of objects, this may be impossible due to limitations on memory storage and CPU.

The theoretical studies of quantiles have been documented extensively in the statistical literature, but generally the implementation is a problem when the entire dataset cannot be placed in memory. Recently, several methods such as remedian, histogram-based estimators, minimax trees, stochastic approximation, and other similar methods have been considered<sup>1</sup> for estimation of the median from a large data set. Some of the existing methods and their drawbacks in handling massive/streaming datasets are briefly mentioned below.

- Minimax trees<sup>2</sup>

- This is a recursive tree structure where minima and maxima are taken alternatively at successive levels of the tree. In the first tier of the tree, the first  $m$  points are stored, and then their minimum is computed and stored in the second tier. This process is repeated until the second tier is full with  $M$  points, and then their maximum is taken and stored in the third tier which will again be a minimum tier. This procedure is repeated until all the data is exhausted.

*Drawback: Restriction on the sample size that is dependent on the choice of parameters  $m$  and  $M$ . It cannot be adopted to streaming data where the total data size is not known in advance.*

- Stochastic Approximation<sup>3,4</sup>

- This procedure involves starting with a pilot sample to get a preliminary estimate and then updating it sequentially. This method utilizes an estimate of the density at each step as part of its updating scheme. It can be slow to react to changes in the source of the data. Chen, *et al.*<sup>4</sup> address this problem.

*Drawback: Accuracy is dependent on the initial sample.*

- Remedian<sup>5</sup>

- Recursive tree structure where the median of  $b$  points at each of  $k$  levels are taken. The first  $b$  points are stored in the first tier of the tree structure and their median is stored in the second tier. This is repeated until the second tier is full and then the median of the  $b$  points in the second tier is stored in the third tier, and so on until all the data is exhausted. Thus the method proceeds by computing medians of groups of  $b$  observations, yielding  $b^{k-1}$  estimates on which this procedure is iterated, and so on, until only a single estimate remains. When implemented properly, this method merely needs  $k$  arrays of size  $b$  that are continuously reused, where  $n = b^k$  denotes the data size.

*Drawback: Data size must be  $b^k$ . Therefore it is not suitable for handling streaming data where the total number of data points is not known in advance. It requires storage space of the order of  $b^k$ .*

- Histogram-type<sup>1</sup>

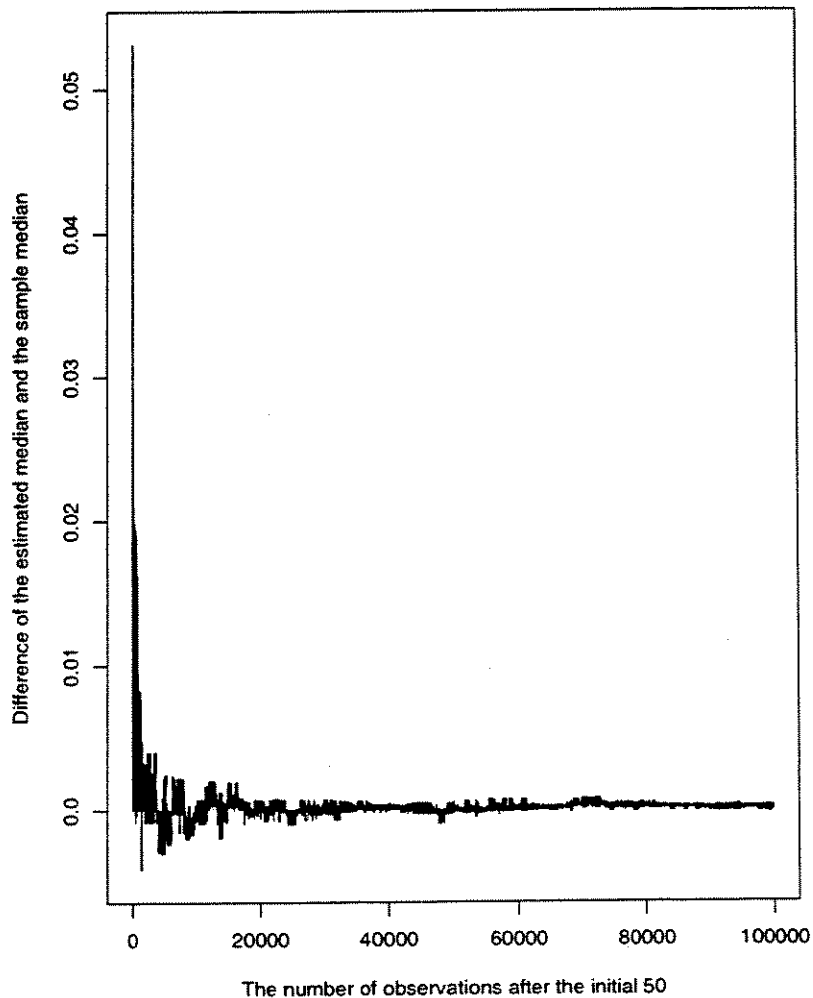
- Involves taking an initial sample and creating histogram-like bins and then sequentially updating the counts.

*Drawback: Accuracy is dependent on the initial sample.*

## 2.1. Streaming quantile estimation

We recently have been investigating a sequential procedure<sup>6</sup> to estimate a  $p$ -th quantile ( $0 < p < 1$ ). The case  $p = \frac{1}{2}$  corresponds to the median. It is a low-storage sequential algorithm using estimated ranks and weights to calculate scores which determine the most attractive candidate data points to keep as the estimate of the quantile.

Consider a very large dataset with  $n$  points of which a fixed number, say  $m$ , points can be placed into memory for sorting and ranking. Initially, each of these points are given a weight and a score based on  $p$ . Now



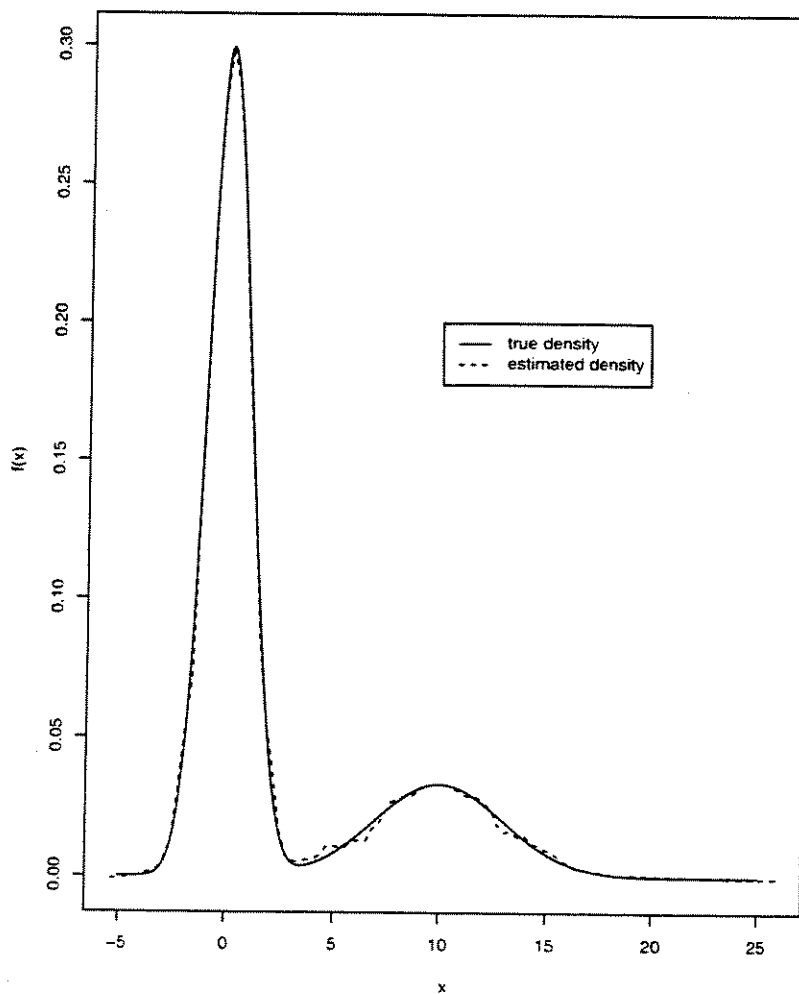
**Figure 1.** Sequential plot of the difference between the median estimated using the low-storage method and the sample median for a Cauchy dataset based on 100,000 points.

a new point from the dataset is put in the array and all the points in the existing array above it will have their ranks increased by 1. The weights and scores are updated for these  $m + 1$  points. The point with the largest score will then be dropped from the array, and the process is repeated. Once all the data points are run through the procedure, the data point with rank closest to  $np$  will be taken as an estimate of the  $p$ -th quantile. See Liechty *et al.*<sup>6</sup> for the details.

Figure 1 shows simulation results for estimation of the median. Here  $m$  is taken as 50. It gives the difference between the low-storage estimate and the sample median based on the points seen up to any given stage. In this example a data of size  $n = 100,000$  points is generated from a standard Cauchy distribution. In addition to the estimates of the median, the sample median is also computed sequentially at each stage for comparison purposes. The graph indicates that the estimates obtained by the low-storage sequential estimation method is very accurate and converges very fast. By repeating the procedure a large number of times, we also found that the errors in estimation are practically negligible.

## 2.2. Simultaneous estimation of multiple quantiles and density estimation

The procedure described above is extended to estimate multiple quantiles simultaneously. This extension proceeds in similar fashion as the single quantile method except that we now track a number of points, say



**Figure 2.** Sample dataset: Mixture of 2 normals with  $n=10,000,000$ . Estimation of density (including the tails) in  $O(n)$  operations.

$m$ , for each quantile. Thus if we are interested in estimating  $k$  quantiles we need to store  $m \times k$  points and sequentially process the data as before.

The output from this algorithm is a set of  $m \times k$  points, each with an associated estimated rank. Hence instead of having the entire empirical cumulative distribution function we have a collection of points located along this function.

While the estimated quantiles are useful and informative on their own, it is often more useful to have information about the density as well. The probability density function can give a more intuitive picture of such characteristics as the skewness of the distribution or the number of modes. Any of the many standard curve fitting methods can now be employed to obtain an estimate of the cumulative distribution function. To get a smooth curve, a logical choice would be to apply an interpolating cubic spline to the estimated quantiles. If the estimated quantile points are used as the knots of the spline, then the interpolating spline will force the piecewise cubic function to go through these points. This gives an estimate  $\hat{F}$  of the true unknown distribution function  $F$ . As noted in Wahba,<sup>7</sup> the derivative  $\hat{F}'$  of this function provides a pointwise estimator of the density  $\hat{f}$ , i.e.  $\hat{f}(x) = \hat{F}'(x)$ . This work is in progress in collaboration with Liechty and Lin at Penn State.

Figure 2 illustrates estimation of the density function using estimates of  $k = 23$  quantiles from a dataset of size  $n = 10,000,000$ . The data is generated from a mixture of two normals ( $0.75N(0, 1) + 0.25N(10, 3)$ , where  $N(\mu, \sigma)$  denotes the Gaussian density with mean  $\mu$  and standard deviation (s.d.)  $\sigma$ ). Here  $m = 50$ , so that the cubic spline is forced to go through  $50 \times 23 = 1150$  points. The figure shows that our method provides a very accurate estimate of the density even at the tail region.

Estimates of the density at quantiles are useful in obtaining the asymptotic variances and covariances of the quantiles as well. For example, given  $k$  sample quantiles  $\hat{\xi}_1, \dots, \hat{\xi}_k$  corresponding to  $0 < p_1 < \dots < p_k < 1$ , respectively, the joint asymptotic distribution of

$$y_n = \sqrt{n} \left( \hat{\xi}_1 - \xi_1, \dots, \hat{\xi}_k - \xi_k \right) \quad (1)$$

is  $k$ -variate normal with mean vector zero and covariance matrix  $\Sigma = (\sigma_{ij})$ , where

$$\sigma_{ij} = \frac{p_i(1-p_j)}{f(\xi_i)f(\xi_j)} \quad (2)$$

for all  $i \leq j$ .

We are currently working on a two-sample goodness-of-fit tests utilizing the estimates of quantiles from streaming data.

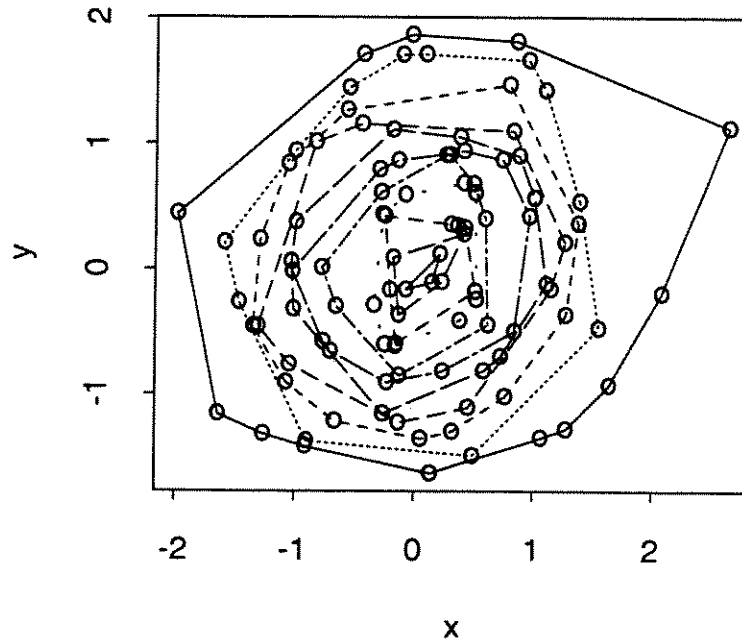
### 2.3. Multivariate methods: convex hull peeling

So far we have concentrated on univariate data and we now turn to multivariate datasets. In the multivariate setting, there is no universal definition of a median or a quantile. Several definitions of multivariate medians are proposed in the statistical literature<sup>8,9</sup> using method such as data depth. These methods are developed for fixed moderate sized datasets. These procedures are almost impossible to implement for streaming multivariate massive datasets.

The concept of convex hull peeling is useful in developing procedures for median in 2 or more dimensions. The convex hull of a dataset is the minimal convex set of points that contains the entire dataset. The convex hull based multivariate median is obtained by successively peeling outer layers until the dataset cannot be peeled any further. The centroid of the resulting set is taken as the multivariate median. Similarly, a multivariate interquartile range is obtained by successively peeling convex hull surfaces until approximately 50% of the data is contained within a hull. This hull is then taken as the multivariate interquartile range. See Barnett<sup>10</sup> for a nice overview.

The drawback with this method is the same problem we have with the univariate quantile: we still have to store the entire dataset to execute the algorithm. To avoid the storage demands of performing convex hull peeling on a massive dataset, a sequential convex hull peeling algorithm is proposed here to deal with streaming

## Convex Hull Peels



**Figure 3.** Example of convex hull peels for a bivariate normal sample of size 100

massive datasets. An example in two dimensions is presented here to give an idea of how the algorithm works, but it can easily be extended to higher dimensions.

To begin we take the first  $m$  (a fixed even number) points from the dataset and peel the outermost layers until approximately  $m/2$  points are remaining. We then add enough points from the remaining dataset to bring the number up to  $m$  again. We would repeat this process until the dataset is exhausted and the resulting outer hull is taken as an estimate of the multivariate interquartile range. A similar procedure could be adapted for the multivariate median. This method avoids the problem of having to store the entire dataset in local memory to perform the peels and it can be extended to dimensions higher than 2.

Figure 3 shows successive convex hull surfaces for a bivariate normal sample of size 100.

### 3. MODEL SELECTION

The Virtual Observatory also requires statistical tools to analyze moderate size databases. Classification and clustering procedures are important tools in analyzing astronomical databases. Several complications may arise in classification problems. The object classes forming multivariate “clouds” in the parameter space may have a power-law or exponential tails in some or all of the dimensions, and some may have sharp cutoffs, etc. The clouds may be well separated in some of the dimensions, but not in others. How can we objectively decide which dimensions are irrelevant, and which ones are useful? The *topology* of clustering may not be simple: there may be clusters within clusters, holes in the data distribution, multiply-connected clusters, etc.

Model selection is an important issue for many astrophysical databases. Many of the clustering problems may be treated as problems in model selection. Model selection is key in answering questions such as:

- How many statistically distinct classes of objects are in a given dataset, and which objects are to be assigned to which classes, along with association probabilities? Are previously unknown classes of objects present?
- Are there interesting correlations among the properties of objects in any given class, and what are the optimal analytical expressions of such correlations? An example may be the “Fundamental Plane” of elliptical galaxies, a set of bivariate correlations obeyed by this Hubble type, but no other types of galaxies. Some of the correlations may be spurious (e.g., driven by sample selection effects), or simply uninteresting (e.g., objects brighter in one optical bandpass will tend to be brighter in another optical bandpass).
- How to select among the many possible alternative astrophysical models?
- How to select among the many alternative curve fittings? Is it a line (linear regression) or a cubic (regression) curve?

Standard penalized likelihood based methods such as Bayesian Information Criterion and Akaike Information Criterion are very popular among astronomers. These approximation methods have been in use for analyzing the so called nested models. The main notion behind these methods is the reduction of statistical bias. However, these methods still suffer from some statistical bias and other theoretical problems.

We present a simple likelihood based ‘leave one out’ jackknife type method for model selection. This works well even when the parameters of the curves are not known completely. Implementation of this method is very simple and does not require any new software.

To describe the method, let  $X_1, \dots, X_n$  be independent identically distributed multidimensional data from an unknown distribution  $F$ . We may be interested in knowing whether the data has three clusters or five clusters. In general, we have no clue as to what  $F$  is. What we are interested in is a model that represents the unknown underlying data generating mechanism as closely as possible. Suppose we need to decide between two possible models, say,  $\{f(\theta; \cdot), \theta \in \Theta_1\}$  or  $\{g(\eta; \cdot), \eta \in \Theta_2\}$ , where  $f$  could be a mixture of three Gaussians and  $g$  could be a mixture of five Gaussians. Here the dimensions of  $\Theta_1$  and  $\Theta_2$  could be different. If we are trying to discriminate between  $f(\theta; \cdot)$  and  $g(\eta; \cdot)$  with fixed  $\theta$  and  $\eta$ , one could compute the log-likelihoods

$$\sum_{i=1}^n \log f(\theta; X_i) \quad \text{and} \quad \sum_{i=1}^n \log g(\eta; X_i), \quad (3)$$

and see which one is larger. If  $\theta$  and  $\eta$  are not fixed and unknown, then they need to be estimated using the data. We could replace  $\theta$  and  $\eta$  by their maximum likelihood estimators  $\hat{\theta}$  and  $\hat{\eta}$ . That is we are in a sense looking for an element of the family  $\{f(\theta; \cdot), \theta \in \Theta_1\}$  that is closest to  $F$  in the so called Kullback-Leibler distance. Now compute

$$\sum_{i=1}^n \log f(\hat{\theta}; X_i) \quad \text{and} \quad \sum_{i=1}^n \log g(\hat{\eta}; X_i) \quad (4)$$

to see which one is larger. But this introduces a bias. Methods such as Bayesian Information Criterion and Akaike Information Criterion use bias correction based on the number of estimated parameters. But these still do not reduce the bias enough. If the leave one out jackknife type method is used, then it can be shown theoretically, under some regularity conditions, that the bias would be substantially reduced. The method consists of computing maximum likelihood estimates  $\hat{\theta}_1, \dots, \hat{\theta}_n, \hat{\eta}_1, \dots, \hat{\eta}_n$ , where  $\hat{\theta}_1$  and  $\hat{\eta}_1$  are based on  $X_2, \dots, X_n$  leaving the first observation  $X_1$  out. Similarly  $\hat{\theta}_j$  and  $\hat{\eta}_j$  are based on  $X_1, \dots, X_{j-1}, X_{j+1}, \dots, X_n$  leaving the  $j$ -th observation  $X_j$  out. Then the estimated likelihoods

$$\sum_{i=1}^n \log f(\hat{\theta}_i; X_i) \quad \text{and} \quad \sum_{i=1}^n \log g(\hat{\eta}_i; X_i) \quad (5)$$

are computed and compared. For large samples the performance is further improved by comparing

$$\sum_{i=1}^n \left( \sum_{j=1}^n \log f(\hat{\theta}; X_j) - \sum_{j=1, j \neq i}^n \log f(\hat{\theta}_i; X_j) \right) \quad \text{and} \quad \sum_{i=1}^n \left( \sum_{j=1}^n \log g(\hat{\eta}; X_j) - \sum_{j=1, j \neq i}^n \log g(\hat{\eta}_i; X_j) \right). \quad (6)$$

The details of the method based on (6) are currently under investigation in collaboration with C. R. Rao at Penn State.

### 3.1. Application to least squares regression

We now illustrate the method in the case of regression. Consider the two regression models, linear and quadratic.

**Model 1:** Linear regression,

$$y_i = a + bx_i + \epsilon_i, \quad \text{where } \epsilon_i \text{ are Gaussian residuals.}$$

**Model 2:** Quadratic regression,

$$y_i = \alpha + \beta x_i + \gamma x_i^2 + \epsilon_i, \quad \text{where } \epsilon_i \text{ are Gaussian residuals.}$$

Let

$$R(M_1) = \sum_{i=1}^n (y_i - \hat{a} - \hat{b}x_i)^2 \quad (7)$$

and

$$R(M_2) = \sum_{i=1}^n (y_i - \hat{\alpha} - \hat{\beta}x_i - \hat{\gamma}x_i^2)^2 \quad (8)$$

denote the residual sum of squares for the two models, where  $\hat{a}, \hat{b}, \hat{\alpha}, \hat{\beta}, \hat{\gamma}$  denote the least squares estimates of  $a, b, \alpha, \beta, \gamma$ . The quantities  $R(M_1)$  and  $R(M_2)$  correspond to the log-likelihoods in (4), except for a negative constant multiplier and an additive constant. According to AIC, we identify the correct model as the one with minimum value among

$$n \log \left( \frac{R(M_1)}{n} \right) + 2 \times 3 \quad (9)$$

and

$$n \log \left( \frac{R(M_2)}{n} \right) + 2 \times 4. \quad (10)$$

The numbers 3 and 4 above correspond to the number of parameters estimated.

On the other hand, the leave one out method of (5), correspond to identifying the correct model as the one with minimum value among

$$\sum_{i=1}^n \log \left( \frac{R(M_1, i)}{n-1} \right) + (y_i - \hat{a}_i - \hat{b}_i x_i)^2 ((n-1)/R(M_1, i)) \quad (11)$$

and

$$\sum_{i=1}^n \log \left( \frac{R(M_2, i)}{n-1} \right) + (y_i - \hat{\alpha}_i - \hat{\beta}_i x_i - \hat{\gamma}_i x_i^2)^2 ((n-1)/R(M_2, i)), \quad (12)$$

where  $\hat{a}, \hat{b}, \hat{\alpha}_i, \hat{\beta}_i, \hat{\gamma}_i$  are the least squares estimates derived by omitting the  $i$ -th data  $(y_i, x_i)$ , and  $R(M_1, i)$  and  $R(M_2, i)$  respectively denote the residual sum of squares for models 1 and 2.

### 3.2. Simulations

We ran many simulations to compare AIC, BIC and the leave one out method. Here the residuals  $\epsilon$ 's are generated from Gaussian distributions with zero means and standard deviations  $\sigma = 0.1, 2, 50$ . The design points  $x_i$  are generated from Gamma with scale parameter 5, Weibull (5, 0.5), and Cauchy with parameter 2. The comparisons are made with two different data sizes  $n = 15$  and  $n = 30$ . The results are presented in Tables 1, 2, 3. The entries in the tables represent the number of times the correct model is picked out of 500 simulations. Even with data sizes as small as 15 or 30, our method picked the correct model more often than AIC or BIC.

One of the main advantages of this method is that it does not require any new software. One can use one's favorite software to compute likelihoods and other parameters. The code for our method sits on top of it with only few lines to combine the leave one out likelihoods. The expressions in (5) help in writing down the computer code in any programming language.

| Normal | Data size $n = 15$ |     |     | Data size $n = 30$ |     |     |
|--------|--------------------|-----|-----|--------------------|-----|-----|
|        | s.d.               | AIC | BIC | LEAVE 1 OUT        | AIC | BIC |
| 0.1    | 161                | 240 | 444 | 207                | 319 | 426 |
| 2      | 168                | 253 | 437 | 199                | 309 | 433 |
| 50     | 170                | 243 | 446 | 194                | 329 | 438 |

**Table 1.** The entries indicate the total number of times the correct model is picked out of 500 simulations. The residuals are from normal with standard deviations 0.1, 2 and 50. The design points  $x$ 's are generated from Gamma with scale parameter 5.

| Normal | Data size $n = 15$ |     |     | Data size $n = 30$ |     |     |
|--------|--------------------|-----|-----|--------------------|-----|-----|
|        | s.d.               | AIC | BIC | LEAVE 1 OUT        | AIC | BIC |
| 0.1    | 182                | 227 | 441 | 191                | 313 | 420 |
| 2      | 165                | 234 | 432 | 215                | 291 | 443 |
| 50     | 162                | 226 | 450 | 198                | 326 | 440 |

**Table 2.** The entries indicate the total number of times the correct model is picked out of 500 simulations. The residuals are from normal with standard deviations 0.1, 2 and 50. The design points  $x$ 's are generated from Weibull (5, 0.5).

| Normal<br>s.d. | Data size $n = 15$ |     |             | Data size $n = 30$ |     |             |
|----------------|--------------------|-----|-------------|--------------------|-----|-------------|
|                | AIC                | BIC | LEAVE 1 OUT | AIC                | BIC | LEAVE 1 OUT |
| 0.1            | 192                | 251 | 450         | 187                | 315 | 424         |
| 2              | 177                | 228 | 442         | 221                | 330 | 455         |
| 50             | 182                | 240 | 452         | 209                | 321 | 451         |

**Table 3.** The entries indicate the total number of times the correct model is picked out of 500 simulations. The residuals are from normal with standard deviations 0.1, 2 and 50. The design points  $x$ 's are generated from Cauchy with parameter 2.

### ACKNOWLEDGMENTS

The research is supported in part by National Science Foundation grant DMS-0101360. Hyun-sook Lee helped with the simulations.

### REFERENCES

1. C. Hurley, and R. Modarres, "Low-Storage Quantile Estimation", *Computational Statistics*, **10**(4), pp. 311-325, 1995.
2. J. Pearl, "A Space-Efficient On-Line Method of Computing Quantile Estimates", *Journal of Algorithms*, **2**, pp. 164-177, 1981.
3. L. Tierney, "A Space-Efficient Recursive Procedure for Estimating a Quantile of an Unknown Distribution", *SIAM Journal on Scientific and Statistical Computing*, **4** (4), pp. 706-711, 1983.
4. F. Chen, D. Lambert, and J. C. Pinheiro, "Incremental quantile estimation for massive tracking," in *Knowledge Discovery and Data Mining*, pp. 516-522, 2000.
5. P. J. Rousseeuw, and G. W. Bassett, "The Remedian: A Robust Averaging Method for Large Datasets" *Journal of the American Statistical Association*, **85** (409), pp. 97-104, 1990.
6. J. C. Liechty, D. K. J. Lin, and J. P. McDermott, "Single-pass low-storage arbitrary quantile estimation for massive datasets", *Statistics and Computing*, To appear, 2002.
7. G. Wahba, "Interpolating Spline Methods for Density Estimation I. Equi-Spaced Knots," *Annals of Statistics* **3**, pp. 30-48, 1975.
8. R. Y. Liu, J. M. Parelius, and K. Singh, "Multivariate analysis by data depth: descriptive statistics, graphics and inference", *Ann. Statist.*, **27** (3), pp. 783-858, 1999.
9. Y. Vardi, C.-H. Zhang, "The multivariate  $L_1$ -median and associated data depth", *Proc. Natl. Acad. Sci. USA*, **97** (4), pp. 1423-1426, 2000.
10. V. Barnett, *Interpreting Multivariate Data*, Wiley, New York, 1981.