

Content of Lecture 5

Computational Statistics
Lecture 5: Simulating from arbitrary empirical random distributions

Raymond Bisdorff

University of Luxembourg

23 novembre 2017

1. Computing quantiles from statistical data
Example of quantile computation
Enhancing exponential tail quantiles'accuracy
2. Single-Pass estimation of arbitrary quantiles
Selecting the Mth largest
Tracking the *M* largest in a single pass
Single-pass estimation of a quantile
3. Practical aspects of the iqagent
Programming exercises
Using the iqagent for Monte Carlo simulations

Observed quantile computation

Consider the following stem and leaf diagram showing the time in minutes measured for 51 units of health care actions in hospital :

| | |
|---|----------------------------|
| 2 | 144 |
| 3 | 00001111223344444466667789 |
| 4 | 001112233566678 |
| 5 | 011566 |
| 6 | 22 |

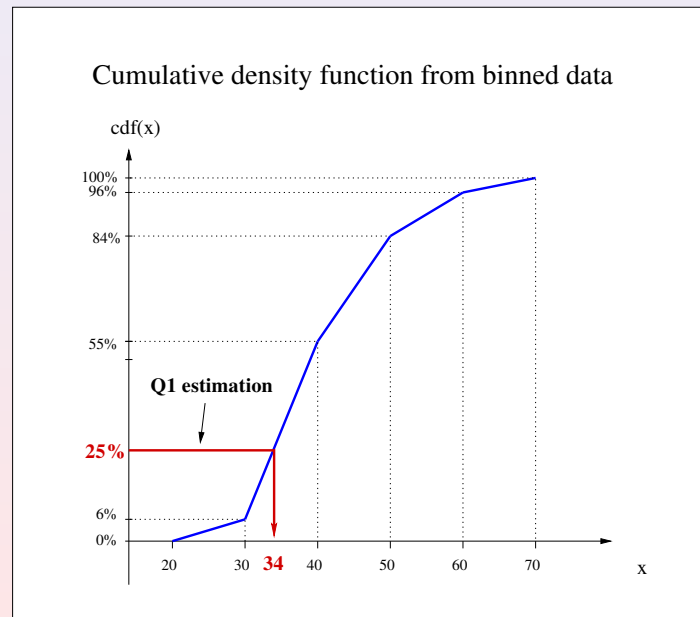
Compute the quartiles Q_0 , Q_1 , Q_2 , Q_3 and Q_4 from the observed data.

Binned quantile interpolation

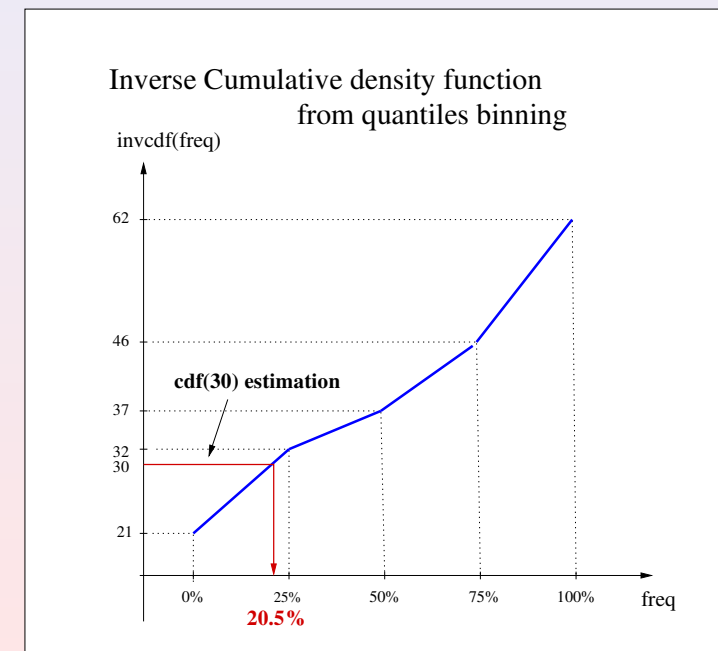
| bin | low | high | center | freq. | f% | F ↑ | F* ↓ |
|-----|-----|------|--------|-------|-----|------|------|
| 1 | [20 | 30[| 25 | 3 | 6% | 6% | 100% |
| 2 | [30 | 40[| 35 | 25 | 49% | 55% | 94% |
| 3 | [40 | 50[| 45 | 15 | 29% | 84% | 45% |
| 4 | [50 | 60[| 55 | 6 | 12% | 96% | 19% |
| 5 | [60 | 70[| 65 | 2 | 4% | 100% | 4% |

Compute the same quartiles from the binned data.

Equally binned data



Regular data quantiles



Content of the lecture Computing quantiles from statistical data Single-Pass estimation of arbitrary quantiles Practical aspects of

Enhancing exponential tail quantiles' accuracy

Exponential tail quantiles' accuracy may be improved by applying non linear, i.e. a logit interpolation.

1. The **logit interpolation** of cumulated probability $F(q)$ of quantile q in bin $[q_0; q_1]$ with cumulated probabilities $F(q_0)$ and $F(q_1)$ is defined as follows :

$$F(q) = g^{-1}\left(g(F(q_0)) + (g(F(q_1)) - g(F(q_0))) \cdot \frac{q - q_0}{q_1 - q_0}\right)$$

where $g(p) = \log(p/(1 - p))$, and $g^{-1}(x) = (1 + \exp(x))^{-1}$.

2. **Inversely**, quantile q_α for $p(q) = \alpha$ and bin $[q_0; q_1]$ with $F(q_0) < \alpha < F(q_1)$, is defined as follows :
 $q_\alpha = \rho q_0 + (1 - \rho)q_1$, where

$$\rho = \frac{g(F(q_1)) - g(\alpha)}{g(F(q_1)) - g(F(q_0))}$$

1. Computing quantiles from statistical data
 Example of quantile computation
 Enhancing exponential tail quantiles' accuracy

2. Single-Pass estimation of arbitrary quantiles
 Selecting the Mth largest
 Tracking the M largest in a single pass
 Single-pass estimation of a quantile

3. Practical aspects of the iqagent
 Programming exercises
 Using the iqagent for Monte Carlo simulations

Selecting the k th smallest or $N - 1 - k$ largest

What is the k th smallest, equivalently the $N - 1 - k$ th largest element out of N preordered (with possible ties) elements $x_{(i)}$ with $i = 0, \dots, N - 1$?

Here k may take on values between 0 and $N - 1$, so $k = 0$ gives the minimum, and $k = N - 1$ the maximum value.

The most common use of selection is in statistical characterization of a set of data, as for instance obtained through the simulation of a random generator.

The quartiles $q_0 = x_{0\%}$, $q_1 = x_{25\%}$, $q_3 = x_{50\%}$ (the median), $q_3 = x_{75\%}$, and $q_4 = x_{100\%}$ are the most commonly used.

9 / 21

Selecting via partitioning

The fastest method for selection, allowing rearrangement, is partitioning, exactly as is done in the Quicksort algorithm.

Selecting a random element, one marches through the array, forcing smaller elements to the left and larger elements to the right.

One can ignore one subset, and continue only with the subset containing the desired k th element. Selection therefore does not need a stack of pending operations and its operations count scales as N .

For a C++/nr3 implementaton see the `sort.h` code.

10 / 21

Tracking the M largest in a single pass

- The previous partitioning approach should not be used for finding the largest or smallest element in an array.
- When one is looking for the M largest elements, where M is modest compared to N , the number of elements of the array, a good approach is to keep a **heap** of the M largest values.
- This approach is implemented as a `HeapSelect` class with :
 - a constructor where you specify M , the size of the heap,
 - an `add` method allowing to add new incoming data values one by one, and
 - a `report` method for getting the k th largest seen so far ($1 \leq k \leq M$).

11 / 21

Heap select –continue

The heap has to be sorted when reporting, but all k values may be given without resorting when no new data value is added meanwhile.

A special case is that getting the $M - 1$ st largest is always cheap, since it is always at the top of the heap.

So if you look for a single favorite k , it is best to choose M such that $M - 1 = k$.

For a C++/nr3 implementaton see the `sort.h` code.

12 / 21

Single-pass estimation of a quantile

Working conditions :

1. The data values fly by in a stream.
2. You get to look at each value once, and do a constant-time process on it.
3. You only have a fixed amount of storage memory.
4. From time to time arbitrary quantiles of the data values seen so far have to be reported.

With conditions stated, only an approximate answer about the exact quantiles of the observed data may be given.

The incremental quantile estimation algorithm

John Chambers et al. (see moodle resources) have given a robust, and extremely fast, algorithm they call IQ agent, that adaptively adjusts a set of bins so that they converge to the data values of specified quantiles (centiles, quartiles, etc).

The idea is to :

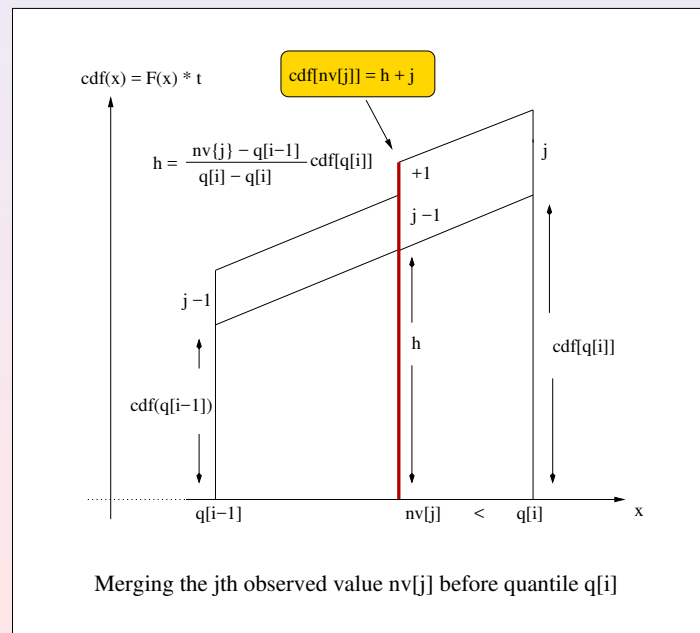
1. accumulate incoming data into batches,
2. update a stored, piecewise linear, cumulative distribution function (cdf) by :
 - 2.1 adding a batch's cdf, and
 - 2.2 interpolate back to the fixed set of quantiles.
3. obtain arbitrary quantiles by linear interpolation from the stored cdf.

For a C++/nr3 implementation see the `iqagent.h` code.

13 / 21

14 / 21

Adding a new observation



Major steps in the IQ algorithm

1. Suppose that T data values have been processed so far.
2. The quantile buffer Q holds the estimated quantiles $[q_{p_1}, q_{p_2}, \dots, q_{p_m}]$ for p -values : $[p_1 = 0.01, p_2 = 0.02, \dots, p_m = 0.99]$.
3. Refill the data buffer $D = [d_1, d_2, \dots, d_N]$ with N new data values.
4. If D is full or at prespecified times, D is converted into an empirical CDF $F_D(x)$.
5. The quantile buffer Q is converted into a CDF $F_Q(x)$.
6. For all values $x \in Q \cup D$, a weighted average CDF : $T/(T+N) \cdot F_Q(x) + N/(T+N) \cdot F_D(x)$ is computed.
7. Quantiles $[p_1 = 0.01, p_2 = 0.02, \dots, p_m = 0.99]$ of the average CDF are used to update Q .

16 / 21

Incremental Python and/or R quantile agent

1. Computing quantiles from statistical data

Example of quantile computation
Enhancing exponential tail quantiles' accuracy

2. Single-Pass estimation of arbitrary quantiles

Selecting the Mth largest
Tracking the M largest in a single pass
Single-pass estimation of a quantile

3. Practical aspects of the iqagent

Programming exercises
Using the iqagent for Monte Carlo simulations

Exercise

1. Re-implement the C++/nr3 iqagent.h code in Python or R,
2. Design a suitable Monte Carlo simulation experience for verifying the re-implementation
3. Compare the run times of the iqagent in each one of your implementations.

17 / 21

18 / 21

Empirical CDF agent

Exercise

Notice that the state of the incremental quantile agent represents in fact the empirical cumulated density function (cdf) constructed on the fly from an incoming random data stream.

1. Add save and restore methods to the iqagent (C++/nr3 and R) that allow to save and restore the state of the agent in/from a file.
2. Add a cdf method to the iqagent in C++/nr3 and R rendering the propability $P(X \leq q)$ of a given quantile(q).

Use the iqagent for simlation problems

Exercise

Notice that previous incremental cdf agent may readily be used for Monte carlo simulation purposes :

1. Save an empirical cdf from a sample of 10000 random normal numbers of mean 50 and standard deviation 20
2. Compare the previous cdf estimation with the theoretical random variable $\mathcal{N}(50, 20)$.

19 / 21

20 / 21

Use the iqagent for simulation problems – continue

Exercise

The size of the data buffer has a certain influence on the accuracy of the iqagent estimations.

1. *Estimate quantiles with the iqagent from a continuous stream of values generated from a known probability distribution, by varying the size of the data buffer D .*
2. *What is the lowest size for D such that the accuracy stays within the 90% confidence interval of the χ^2 test of difference between the estimated and the real distribution.*