How to get good cuts when you can not store the full sample?

R. Frühwirth, J. Lettenbichler, T. Madlener

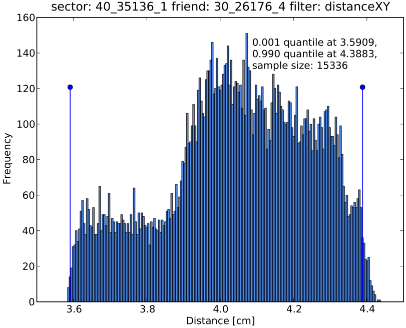
Institute of High Energy Physics
Austrian Academy of Sciences

February 20, 2015

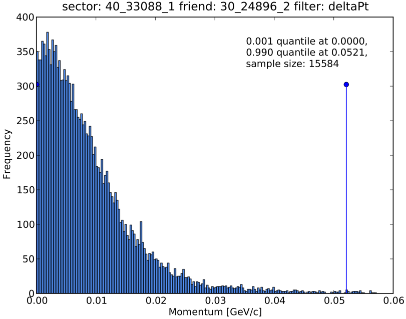
why RAM is the bottleneck for secMapCreation

The Problem:

- getting good cuts for the secMap is crucial
- reliable cuts need big sample sizes which need to be stored
- rough estimation: nTracks * charges * phiRange * thetaRange * momentumRange = nTracks * 2 *360*140*3500 = nTracks * 350 Mio
- easy solution: iterative algorithms to determine cuts
- for gaussian-like distributions: estimate expectation value and standard deviation → you can easily estimate any quantile you like
- unfortunately we can not assume gaussian-like distributions
- requests:
    - determine exact quantile of sample, if possible
    - shall work independently from shape of distribution
    - allow merging of different samples of the same sector-combination


sector: 40_35136_1 friend: 30_26176_4 filter: distanceXY

0.001 quantile at 3.5909,
0.990 quantile at 4.3883,
sample size: 15336


sector: 40_33088_1 friend: 30_24896_2 filter: deltaPt

0.001 quantile at 0.0000,
0.990 quantile at 0.0521,
sample size: 15584

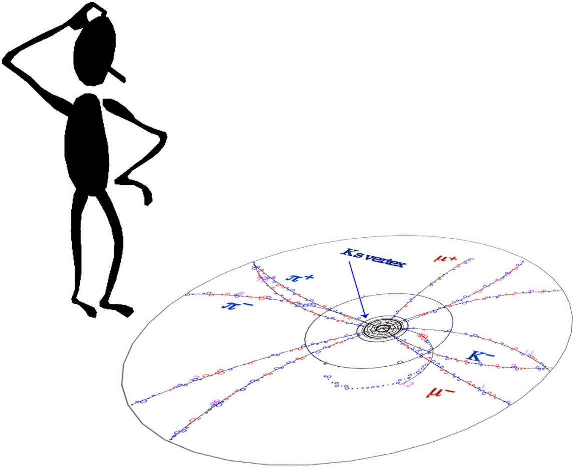
how to get quantiles

- so far:
  - collect full sample
  - sort it
  - retrieve quantiles (min & max cutoff)
- proposal:
  - instead of storing full sample, two sortable containers (one for lower and one for upper cutoff) are used.
  - the fact that the quantiles are near the min and max quantile can be used for reducing the footprint

Example: case 1% and 99% percentile

- start with 2 small sortable containers one for lower (filled with datatype::max), one for upper (filled with datatype::min) cutoff.
- new measurements are sorted into these containers, if they are bigger than the smallest entry (for upper cuts), or smaller than the biggest entry (for lower cuts)
- if measurement was accepted e.g. from upper cut container, lowest value for upper cut is discarded
- if measurement was not accepted by the containers, measurement is discarded and sample-size-counter increased
- size of container grows with the sample size, for 1% percentile → store smallest 2% of measurements.
- if two samples shall be merged, the lower and the upper containers of both samples are fully merged and their counters added.
- with the upper safety margin, for given quantiles and big sample sizes memory consumption is decreased by a factor of 25.
- does practically never lose relevant data and works with samples of any shape

that's all, folks!



Any suggestions, ideas or requests?
Jakob.Lettenbichler@oeaw.ac.at