DOT-K: Distributed Online Top-K Elements Algorithm with Extreme Value Statistics

Nick Carey, Tamás Budavári, Yanif Ahmad, Alexander Szalay
Johns Hopkins University
Department of Computer Science
ncarey4@jhu.edu
Context

• Simple Top-k query – selecting the largest ‘k’ data elements
• Peta-scale and above datasets row-partitioned over many nodes
• Naïve, centralized solutions quickly become untenable at scale
Top-K Query Research

- Most work in the field is based on variants of the Threshold Algorithm, selecting the Top-K of a monotonic aggregation function over row elements.

- We target the simple Top-K query, and our approach is generic and widely applicable.

Structure

• Overview of relevant Extreme Value Statistics
• Outline of DOT-K Algorithm
• Experimental results
Extreme Value Statistics

- EVS is concerned with characterizing the tail distributions, or extreme values, of random variables.
- Traditionally used to describe extreme environmental phenomena as well as weakest-links in reliability modeling.
Pickands, Balkema, de Haan Theorem

• The distribution of threshold exceedances of a sequence of independent and identically-distributed random variables with a common continuous underlying distribution function is approximated by the Generalized Pareto Distribution, and that the approximation converges as the tail threshold rises.

• The ‘k’ largest values of a dataset may be well approximated by the Generalized Pareto Distribution provided the ‘k’th order statistic is appropriately high.
Bias-Variance Trade-off

- Selecting a threshold from which to model threshold exceedances
- A lower threshold results in a worse theoretical GPD approximation of the data
- A higher threshold limits the amount of available threshold exceedances leading to greater parameter estimation uncertainty
- Fortunately for our context, this becomes less of a problem as dataset size increases
Generalized Pareto Distribution

\[ p(x|\xi, \sigma, \mu) = \frac{1}{\sigma} \left[ 1 + \xi \cdot \left( \frac{x-\mu}{\sigma} \right) \right]^{-\frac{1}{\xi}} \]

**Equation 1.** GPD probability density function including parameters $\xi$ (shape) $\sigma$ (scale) and $\mu$ (location, or threshold)
Estimating GPD Parameters in Practice

• Variety of published methods for estimating GPD parameters that best fit a set of threshold exceedances
• Various strengths and weaknesses in computational complexity and accuracy
• Crucial to the DOT-K algorithm, as good parameter fit greatly affects query accuracy
• For our purposes, we use a computationally intense yet relatively accurate Maximum Likelihood Estimator
• **Equation 2.** Coles’ M-Observation Return Level equation. \( \zeta_{\mu} \) is a constant estimated by the number of observations exceeding \( \mu \) divided by total observations.

\[
\zeta_{\mu} \left[ 1 + \xi \cdot \left( \frac{x_m - \mu}{\sigma} \right) \right]^{-\frac{1}{\xi}} = \frac{1}{m}
\]

• For a given GPD, one may calculate the threshold \( x_m \) that is exceeded on average once every \( m \) observations.

• By relating ‘\( m \)’ to the dataset size, we can estimate various order statistics.
DOT-K Algorithm Objective

• Assuming a numerical dataset row-partitioned across many nodes, our goal is to estimate the k’th largest element and subsequently retrieve all elements greater than the estimate.
**DOT-K Algorithm**

1. Each distributed node collects its largest ‘k’ local values and calculates the GPD parameters that best fit the local data partition.

2. By relating the GPD parameters collected from each data partition node, the query issuer estimates the global k’th largest element by numerically solving Equation 3 (next slide).

3. The k’th order statistic estimate is communicated to the distributed nodes and the exceedances are relayed back to the query issuer.
Our Contribution

\[
\sum_{i=1}^{p} n_i \xi \mu_i \left[ 1 + \xi_i \cdot \left( \frac{x_m - \mu_i}{\sigma_i} \right) \right]^{-\frac{1}{\xi_i}} = k
\]

**Equation 3.** Our modification of Coles’ M-Observation Return Level. Numerically solving for \(x_m\), this equation estimates each distributed data partition’s expected contribution to the top-\(k\) query result.

Note that this equation is also useful for estimating many upper order statistics by varying ‘\(k\)’; \(x_m\) is the estimate for the ‘\(k\)’th global order statistic
Communications Overhead

• Four series of messages
  • Query Issuer sends message to each dataset partition node, starting query and communicating the query parameter ‘k’
  • Dataset partition nodes forward local GPD parameter estimates to central Query Issuer
  • Query Issuer relays global k’th order statistic estimate to each dataset partition
  • Dataset partitions forward k’th order statistic exceedances to Query Issuer forming the query result

• Ideal DOT-K implementation transmits 4*P total messages between all nodes with approximately 6*P + ~k total real values communicated