Proceedings of the 2023 Winter Simulation Conference
C. G. Corlu, S. R. Hunter, H. Lam, B. S. Onggo, J. Shortle, and B. Biller, eds.

SCALING CROSS-RELATIONS WITH LARGER DATASET

Victor Diakov

Simfoni Ltd.
450 Townsend St.
San Francisco, CA 94107, USA

ABSTRACT

Simulation and optimization of procurements might employ clustering dataset elements to exclude possible duplicates and improve processing resiliency. This study presents a case of applying scaling methods to reduce computation time of clustering between a smaller and a larger dataset. In this example (of selecting close supplier names), computation time scales as square of N (the number of elements), and the presented approach in effect brings computing time to be linear in N. As a result, computation time in our case is reduced by over an order of magnitude.

1 INTRODUCTION

Resilient simulation and optimization of procurements often requires to cluster similar supplier names from vendors dataset. Clusters of (new) sets of vendor names are routinely analyzed (Diakov and Anandpara 2022) in conjunction with known (legacy) dataset and, in our case, only clusters that include vendors from the new set are of interest. The goal is to find cross-relations between two datasets - a smaller ('new') set and a larger ('legacy') one. Our plans involve integrating the routine into procurement and spend analysis solutions (areas of commercial application are described at https://simfoni.com/spend-intelligence/).

The presented method is a shell (an 'envelope') for speeding-up updates of processed information where smaller sets of data are examined for clustering (or for other cross-relations) with the larger processed dataset. Various areas of application for clustering are listed elsewhere (Oyewole and Thopil 2023).

2 APPROACH

Here, clustering is treated as a 'black box', we are only varying the input dataset contents, and the task for our shell process is to slice the legacy dataset to pieces so as to minimize computing time. The smaller dataset is then analyzed for clustering with each slice of the larger legacy dataset, and yet the overall computation time decreases despite smaller dataset being analyzed multiple times. Computation time scaling as a higher power of dataset size makes such slicing possible.

When sliced, parts of the large 'legacy' dataset clusters might end up in different slices. The method presented here assumes that if an element of the 'new' dataset gets clustered with a legacy cluster, then it necessarily will get clustered with at least one of the sliced parts of that cluster.

The clustering routine for the case presented here was downloaded from
https://scikit-learn.org/stable/modules/generated/sklearn.cluster.DBSCAN.html

The results are shown for a 'legacy' dataset containing 243695 vendor names and a 'new' set consisting of 1361 vendors none of which is an exact match with legacy vendors.

2.1 Downscaling Computation Time

A simple test shows that computation time in our case scales as the square of the number of elements being examined for clustering. We'll have these notations: n is the number of 'new' elements,

N is the number of 'legacy' ones, m is the number of chunks of data that N is sliced into. Each slice of N/m legacy elements is analyzed for clustering with the 'new' n elements, meaning there are m computations of $n + N/m$ elements. At given N and n, to minimize

$$m \cdot (N/m + n)^2 \rightarrow \min, \tag{1}$$

m should be N/n, and the computing time scales as $n \cdot N$ (vs. N^2 when clustering all in one big chunk).

Aggregation of clusters from different slices is needed to 're-assemble' clusters whose elements end up in different slices of the legacy data. This postprocessing operation scales with the number of clusters (rather than with N) and in our case does not affect the overall computing time.

It is important to note that the method applicability is not restricted to clustering, the approach can speedup computing other cross-relations between smaller and larger datasets elements. The method applies when i) cross-relations computation time scales as higher power (N^p, $p > 1$) of the dataset size, and ii) the postprocessing time is small.

2.2 Results

The results (Figure 1) confirm that faster processing is achieved when clustering the dataset 'by parts' rather than in one big 'chunk'.

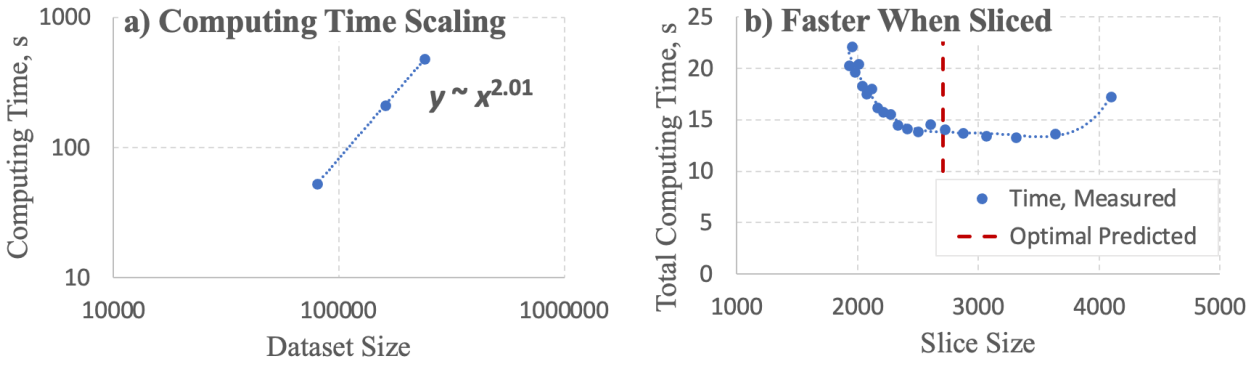

Figure 1: Downscaling computation time. For a (clustering) process with computation time scaling as the square of dataset size (a), the overall computing time is reduced (b) by slicing the larger legacy dataset and clustering slices one-by-one with the new dataset. In the examined case, the optimal predicted number of slices is 179 and the optimal predicted Slice Size (legacy plus new data) is 2712. The overall computing time is reduced significantly, from 500 s to under 15 s.

3 CONCLUSION

A scaling approach is applied to speedup computation of relations between a smaller and a larger datasets. The application involves processing the large dataset in pieces, which would suggest further improvements are possible with use of parallel computing.

REFERENCES

Oyewole, G. J., and G. A. Thopil. 2023. "Data Clustering: Application and Trends". *Artificial Intelligence Review* 56:6439-6475.

Diakov, V., and T. Anandpara. 2022. "Referenced Filtering: a Case for Avoiding Each-to-Each Computations". In *Proceedings of the 2022 Winter Simulation Conference*, edited by B. Feng, G. Pedrielli, Y. Peng, S. Shashaani, E. Song, C.G. Corlu, L.H. Lee, E.P. Chew, T. Roeder, and P. Lendermann. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.