LiberTEM : Distributed algorithms for dimensionality reduction methods on scanning transmission electron microscopy (STEM) data

Org : Python Foundation

Sub-org : LiberTEM

Participant Name : Jaeweon (Jae) Shin

Mentors : Alex (@sk1p), Dieter (@ueulle)

About me

I’m a third year student in mathematics currently studying temporarily at ETH Zurich in the Department of Mathematics. At the moment, I’m focusing on statistics and optimization theories, and I’m particularly interested in statistical inference and computational biology. I have been programming since the first year in college and mainly used Python as my primary language. In the past, I participated in developing a bioinformatics application at a genomics lab, called “juicebox,” which is a tool for computational biologist to visualize Hi-C genomics data based on pairwise interactions among gene locus. Also, I have participated in various data science projects and am familiar with the overall flow of a typical data science project.

I’ve decided to join GSOC this year, and in particular LiberTEM, because I wanted to learn more about backend-side of developing open-source projects while working on rewarding and interesting tasks. To the best of my knowledge, electron microscopy is a widely-used tool in computational biology; as a person who is interested in computational biology, I cannot think of a better opportunity to gain hands-on experience on how electron microscopy data gets processed. Furthermore, distributed processing of data is a recurring theme in many branches of science and I hope to learn more about it by working on a project with LiberTEM. Lastly, managing open-source projects are becoming more and more important in science and engineering and is producing/maintaining open-source tools are getting acknowledged as almost as important as developing new scientific theories. Therefore, I hope to learn more about the overall distributed data processing pipeline as well as how different components of an open-source project is maintained by partaking in GSOC at LiberTEM this year.

Details

1. University / Program / year / expected graduation date : ETH Zurich / Department of Mathematics / 3rd year / 2020 May
2. Time Zone : Central European Time (UTC + 1)
3. Current Country of Residence : Switzerland

Project Abstract

Dimensionality reduction techniques are useful methods that allow us to gain crucial insights about the given dataset. For instance, high-dimensional dataset is often impossible to visualize, thereby making human-level comprehension hard. By using dimensionality reduction techniques, one can reduce the dimension of the dataset to 2-dimensional or 3-dimensional so that visualization becomes a feasible task. Furthermore, dimensionality reduction serves many other purposes, including but not limited to feature extraction and noise reduction. Unfortunately, such methods become computationally intensive when dealing with large scale dataset such as the ones that LiberTEM handles. To deal with complexity issues, one possible approach is to implement algorithms in a distributed fashion, which is the main focus of my project for the summer.

After discussing with mentors, I have narrowed down to three tentative candidates for dimensionality reduction methods that could be applicable to LiberTEM : Principal Component Analysis (PCA), Independent Component Analysis (ICA), and UMAP (Uniform Manifold Approximation and Projection). To briefly discuss each of the methods, PCA is one of the most commonly used dimensionality reduction technique that extracts the lower dimensional structure in the data. ICA is a simple yet powerful signal processing technique that can separate multivariate signal that comprises the data as well as identifying artifacts (noise) that are embedded in the image data. Lastly, UMAP is a recently developed dimensionality reduction method that can handle non-linear dataset (PCA and ICA assumes linearity in the data). As for the prospect of being implemented in a distributed way, a lot of research is available for PCA, since it is a well-established technique that has been studied for a long time. Compared to PCA, ICA is a less-studied from distributed algorithm perspective, but there are some researches that have proposed distributed implementation of ICA. Being such a new technique (published in 2018), UMAP is yet to have any distributed implementation available. Nonetheless, a lot of computations within UMAP is parallelizable (e.g. stochastic gradient descent), suggesting a glimpse of hope for distributed implementation.

In the initial phase of the project, I will carefully go over each of these methods (and potentially other methods) to select one method that I will pursue for the summer. To do so, I will conduct cost and benefit analysis for each of these methods, with regards to how attainable the implementation of the method would be for the summer as well as how useful it would be for the LiberTEM community. Furthermore, I will be prototyping the distributed version of each of these methods to see how the implementations for the methods can benefit from the existing LiberTEM distributed processing framework. Having done that, I will work on the actual incorporation of the distributed algorithm for the chosen dimensionality reduction method into LiberTEM.

Ideally, users of LiberTEM can benefit from implementation of these algorithms that they can run through a simple pipeline. User-defined functions in LiberTEM, which is a new component that is in active development, allow the users to run functions with their desired functionality without having to worry about parallelization, which is done under the hood by LiberTEM. My project will be concerning both the distributed implementation of dimensionality reduction technique as well as improving the general User-defined functions framework.

Project Goal

The core goal of the project for the summer is the implementation of the distributed algorithms for dimensionality reduction methods. The ultimate goal of the project (potentially beyond this summer) is the implementation of such algorithms through LiberTEM back-end facilities so that the algorithms benefit from the existing distributed frameworks and allow users to deploy these algorithms through UDF. Therefore, the main focus of the project will be implementation of dimensionality reduction methods as well as making improvements on UDFs and LiberTEM back-end to facilitate efficient dimensionality reduction.

1) Distributed Algorithms for dimensionality reduction methods

a) Read about the current research works on this topic

b) Produce prototype implementations of such algorithms under LiberTEM settings where the data is partitioned and each partition is kept on fast local storage on separate cluster nodes.

c) (If possible) Make it compatible with User-defined functions framework so that users can call these algorithms at their disposal

2) User Defined Functions

a) Make improvements on the current implementation of run_udf. For instance, we can only perform computation over each individual frames in the current framework of run_udf. If we can perform computation over a small partition/batch of frames, then we can potentially save costs and speed up the computation

b) Make thorough documentation about User-defined functions in general

Code Contribution

https://github.com/LiberTEM/LiberTEM/pull/308

The above pull request is an implementation of one-pass algorithms for computing variance, standard deviation, sum, and mean of the data. First, I implemented the merging functionality in udf/stddev.py, which merges two (or more) partitioned sets of frames with different number of frames and computes variance and sum over joint set of frames. Then, using the run_udf function, which partitions the dataset and maps the function, which is given as a parameter to run_udf, to each partition, I iteratively merged the partitioned frames and computed the joint variance and sum over the entire dataset. Having implemented all the necessary functions for computing the statistic, I've made run_stddev, which is served as a top-level interface with which LiberTEM users can call to obtain variance, standard deviation, sum, and mean of a dataset, without having to go through the hidden function calls. Lastly, I implemented a testing function in tests/udf/test_stddev.py, where I tested the "merging" function with randomly generated dataset.

While I was doing the above pull request, I filed an enhancement issue about BufferWrapper, which stores intermediate results of run_udf function (currently resolved) :
https://github.com/LiberTEM/LiberTEM/issues/315
Basically, the issue was that the previous BufferWrapper didn't allow storing a single constant value. For instance, if I wanted to store the number of frames, which is an integer (1-dim array), I had to propagate the entire n-dimensional array (same as the dimension of the dataset) with the desired integer. Currently, this issue is resolved with kind="single," which allows us to define the input type of the buffer as a single constant value of 1-dimensional array.

One improvement that I hope to make about the implementation of computing various statistic concerns the mapping of UDF on large subsets of data (mentioned above). Currently, "merging" operation is done iteratively over individual frames; if we can perform computation over a large frames subset, it would potentially speed up the computation (mentioned above in the Project Goal/User Defined Functions 2b)

Some further potential room for improvements are as follows:

1) Replace collections.namedtuple with other data structure since namedtuple seems to lead to messier code

2) More parallelism and optimization, potentially using numba

Weekly Timeline

Below is a tentative schedule for the summer and is subject to change.

Community Bonding (May 7 - 26) :

- Discuss with mentors further ideas about dimensionality reduction methods that have useful applications in STEM data
- Identify and study three dimensionality reduction method candidates to pursue for the summer (Currently considering ICA, PCA, UMAP as potential candidates as mentioned above)
- Become familiar with the codebase for User-defined functions on LiberTEM
- Read various issues raised in the past related to User-defined functions to get better understanding of different components of codebase and to get hands wet on the project (refer Issues)

Week 1 (May 27 - 31) :

- Get more familiar with the API and the overall codebase
- Gain better understanding of the state-of-the-art distributed algorithms the chosen dimensionality reduction method candidates

Week 2 (June 3 - 7):

- Gain better understanding of the state-of-the-art distributed algorithms the chosen dimensionality reduction method candidates

- Given the three candidates for dimensionality reduction methods, write up prototype codes for each of the three candidates

Week 3 (June 10 - 14) ~ Week 4 (June 17 - 21) :

- Given the three candidates for dimensionality reduction methods, write up prototype codes for each of the three candidates
- Write up short reports for each of the three candidate dimensionality reduction methods, discussing how practical/attainable each methods are for the summer
- Discuss with mentors on reviewing the prototype implementations for three dimensionality reduction candidates and select a method that would be most appropriate for full implementation into LiberTEM
- Communicate with STEM application experts as well as LiberTEM users to gain feedbacks about the current progress on dimensionality reduction. Discuss the potential use cases of dimensionality reduction method. Also, discuss use cases for other promising statistical methods, such as K-means clustering, that would be beneficial to the community.

Week 5 (June 24 - 28):

- Investigate Issue # 262, on adding support for type checks with UDF functions, and work towards a fix. For this task, I will mostly build up on top of the existing discussions among LiberTEM developers around this topic and explore other open-source projects that currently has this support such as astropy (see Issues)
- Begin implementing the selected dimensionality reduction method into LiberTEM

Week 6 (July 1 - 5) ~ Week 7 (July 8 - 12) :

- Work on minibatch-level mapping for UDF. Currently, only frame-level mapping of functions is supported in UDF, but for various reasons (including aforementioned computational cost reason; see Project Goal), it can be advantageous to have mapping for UDF on large subsets of data
- Write test cases and add documentations
- Continue working on implementing the selected dimensionality reduction method into LiberTEM

Week 8 (July 15 - 19) ~ Week 9 (July 22 - 26):

- Experiment with numba.njit method to add auto-vectorization of loops and variable elimination features to UDF functions to improve performance
- Continue working on implementing the selected dimensionality reduction method into LiberTEM
- Once implemented, add unit testing for the implemented dimensionality reduction method

Week 10 (July 29 - August 2):

- Continue working on implementing the selected dimensionality reduction method into LiberTEM
- Write test cases and add documentation, if necessary
- Complete any missing documentations or tests for implemented functions
- (If time permits) Port the implemented dimensionality reduction method as a User-defined function

Week 11 (August 5 - 9):

- Complete any missing documentations or tests for implemented functions
- Complete/Wrap up any unfinished tasks that I have been working on during the summer
- (If time permits) Port the implemented dimensionality reduction method as a User-defined function

Week 12 (August 12 - 16):

- (If time permits) Port the implemented dimensionality reduction method as a User-defined function
- Complete/Wrap up any unfinished tasks that I have been working on during the summer
- Write a thorough report and documentation for the works I have done during the summer, especially for tasks that I have attempted but not finished or that I have thought about but didn’t have the time to finish. Such a report could be valuable resource to anyone who wishes to take over the batton or continue working on this project
- Submit report for the project

Issues

Here are some existing issues related to User-defined functions that other developers at LiberTEM have worked on. These issues could be a useful benchmark for checking the needed functionality for User-defined functions (UDF) on LiberTEM. Also, as mentioned above in the weekly timeline, these issues will be used as an introduction to the existing framework during the community bonding period.

- https://github.com/LiberTEM/LiberTEM/issues/306 : Documentations needed for UDF

- https://github.com/LiberTEM/LiberTEM/issues/289 : Creating new dataset for UDF
- https://github.com/LiberTEM/LiberTEM/issues/262 : Type checking for UDF
- https://github.com/LiberTEM/LiberTEM/issues/287 : Support for more parameter data for frames
- https://github.com/LiberTEM/LiberTEM/issues/288 : Run functions on cluster without reading an input dataset

References

https://www.nature.com/articles/srep26348 : Big Data Analytics for Scanning Transmission Electron Microscopy Ptychography

https://dbs.ifi.uni-heidelberg.de/files/Team/eschubert/publications/SSDBM18-covariance-authorcopy.pdf : Numerically Stable Parallel Computation of (co-) variances

https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5657031/ COINSTAC : Decentralizing the future of brain imaging analysis

https://ieeexplore.ieee.org/document/7324344 Large scale collaboration with autonomy : Decentralized data ICA

https://arxiv.org/pdf/1811.00080.pdf Manifold learning for four-dimensional scanning transmission electron microscopy

Availability/Other Commitments/GSOC

11At ETH Zurich, session examination periods (i.e., final exams) starts in the second week of August. However, I don't see that as a major hurdle, since GSOC will be mostly done by the time the exam starts at ETH. While the detailed examination schedule is not out yet, should there be any conflicts, I will make sure to work on the weekends, etc, so that it wouldn't interfere with the project that I'm working on with LiberTEM. Also, the classes ends at the end of May, so there will not be any major conflict with classes during the summer. I'm willing to spend 30 + hours on the GSOC project since I don't have any other official commitments during the GSOC period. At the moment, I don't have any plans for traveling; if I happen to move around during working days, I will make sure to finish the designated amount of works for the week either before or immediately after to make sure that I’m on schedule for the project. Also, this is my first time participating in GSOC, and LiberTEM is the only organization that I’m applying to.