A4 Issues List

Keycap Guardians

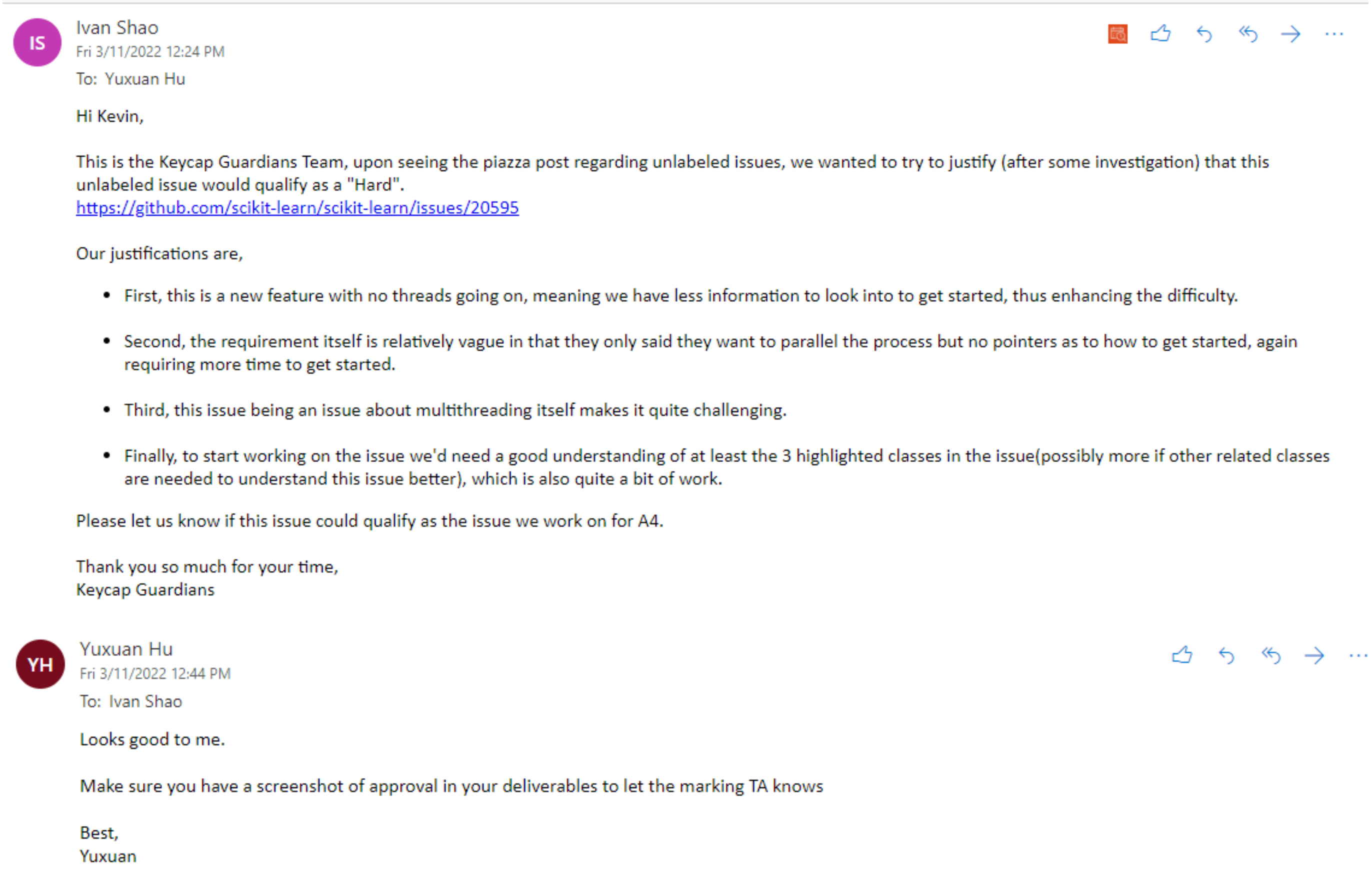
Issue #1(feature/optimization): Parallelization in Locally Linear Embedding

Link to Issue: https://github.com/scikit-learn/scikit-learn/issues/20595

Summary: The issue proposes a new feature that improves performance of a series of computations.

In Locally Linear Embedding, the process of local reconstruction for each of n points (presented in *barycenter_weights* function) is the most computationally costly one. The loop over the points could be easily parallelized to greatly improve the performance, especially since *n_jobs* parameter is already utilized in *barycenter_kneighbors_graph* which calls *barycenter_weights*.

TA Approval of using Issue #20595



IS

Ivan Shao

Fri 3/11/2022 12:24 PM

To: Yuxuan Hu

Hi Kevin,

This is the Keycap Guardians Team, upon seeing the piazza post regarding unlabeled issues, we wanted to try to justify (after some investigation) that this unlabeled issue would qualify as a "Hard".
https://github.com/scikit-learn/scikit-learn/issues/20595

Our justifications are,

- First, this is a new feature with no threads going on, meaning we have less information to look into to get started, thus enhancing the difficulty.
- Second, the requirement itself is relatively vague in that they only said they want to parallel the process but no pointers as to how to get started, again requiring more time to get started.
- Third, this issue being an issue about multithreading itself makes it quite challenging.
- Finally, to start working on the issue we'd need a good understanding of at least the 3 highlighted classes in the issue(possibly more if other related classes are needed to understand this issue better), which is also quite a bit of work.

Please let us know if this issue could qualify as the issue we work on for A4.

Thank you so much for your time,
Keycap Guardians

YH

Yuxuan Hu

Fri 3/11/2022 12:44 PM

To: Ivan Shao

Looks good to me.

Make sure you have a screenshot of approval in your deliverables to let the marking TA knows

Best,
Yuxuan

Issue #2(integration): Improve tests to make them run on variously typed data using the global_dtype fixture

Link to Issue: https://github.com/scikit-learn/scikit-learn/issues/22881

Summary: Introduction of low-level computational routines motivated an extension of tests to run them on 32-bit data.

Currently only a single CI job is used to run tests on 32 bit datasets. A previously resolved issue #22690 introduced a new *global_dtype* fixture as well as a *SKLEARN_RUN_FLOAT32_TESTS* environment variable which makes it possible to run tests on 32-bit datasets for multiple CI jobs.

In this regard, efforts are needed to review current test suites and rewrite appropriate ones with latest fixtures. For instance, tests that check for the exception messages raised when passing invalid inputs must not be converted. Tests using *np.testing.assert_allclose* must now use *sklearn.utils._testing.assert_allclose* as a drop-in replacement.

Issue Chosen: #20595

The reason we chose issue#1 is because it's interesting(involves multi-threading), and more challenging in the sense that this issue was only recently opened and has little existing discussion/comments. Also, it weighs more on programming skills rather than machine learning knowledge which is more suitable to our team’s strengths. Lastly, issue#2 requires examination of dozens of tests, considering the limited timespan we might not be able to provide the best solutions.