Code Review: clkhash

V0.11.0 - Commit hash 4adcb908ea7f2be689471df048e1c7123ef9b222 (Tue 10 Apr)

This report details the findings of a half-day review of clkhash by Arthur Street (Data61).

Summary

No major issues identified. The code looks well structured. The two most important issues would be clearing up the python version compatibility, and checking that the type errors are not pointing to any uncovered edge cases. An architectural suggestion would be to remove the dependence on the tqdm progress bar from clk.py, and move it to cli.py.

Python version ambiguity

- The ReadMe says “Supports Python versions 2.7+, 3.4+”. PyCharm tells me `concurrent.futures` isn’t available in Python 2.7. Also the CLI is pinned to Python 3.6 (the first line is #!/usr/bin/env python3.6).
- You could add “classifiers” to setup.py to specify which python versions are supported.

Type errors

- I can’t see immediately how to check types.
- I installed mypy and ran:

  ```
  mypy clkhash --ignore-missing-imports --strict-optional --disallow-untyped-defs
  ```

  which raises a number of errors, including:

  ```
  clkhash/field_formats.py:235: error: Unsupported operand types for > ("int" and "Optional[int]")
  clkhash/field_formats.py:245: error: Value of type variable "AnyStr" of "re_compile_full" cannot be "Optional[str]"
  clkhash/clk.py:136: error: "None" not callable
  ```

  ...Plus lots of “`Function is missing a type annotation`” messages.

Architecture

- Assuming clk.py is meant to be common code that could support a number of different interfaces, then tqdm’s progress bars (which are specific to a CLI) should be handled in cli rather than clk - perhaps by passing a progress-bar callback to clk rather than a “progress_bar” boolean. (Or, could at least reduce code duplication by having only one call to generate_clks, with callback=None if no progress_bar.)

Tests

- Test coverage is excellent at 95%, and easy to run via pytest (although two env vars need to be set to avoid skipping some tests).
- Although the coverage report (pytest --cov-report html --cov=clkhash ; open htmlcov/index.html) says clk has only one line not covered by the tests, test_clk.py only seems to test that clk.generate_clk_from_csv doesn't crash. Would it be worth checking that for a given random number seed, the result is stable for some inputs?

Minor

- [PR already merged]: I commented out `stats` on lines 131 and 146 of clk.py and it all still worked – it looks unnecessary.
- PyCharm tells me there are quite a few imports that don't get used, eg. in "clk.py", platform, sys, Any, Union. (These particular ones removed in the above PR.)
- Some docstring param lines are out-of-date, eg. bloomfilter.py's stream_bloom_filters `refers to parameters "schema_types" and "xor_folds" which are not present.

Style

- I prefer not to create new attributes in methods outside __init__, eg. randomnames.load_names creates self.all_male_first_names etc. (ie. give them the value None in __init__). According to this link, it may be a PEP8 thing, but I can't find it there myself.
- Mix of single and double quotes, eg.

```
with tqdm(desc="generating CLKs", ..., unit='clk', ...
```

- I like your generic type variable T = TypeVar('T') .
- I haven't seen strings inserted into the code before, eg. in NgramEncodings.

Other

- Did you try using greenlet/gevent for multiprocessing? (I haven't used it myself, just curious if it applies here.)
- Do you need to attribute where you got the names data from?
- Are there any known vulnerabilities (eg. in the CVE database) for any of the dependencies?

Requirements

- There is a more recent version of "cryptography" available (2.2.2) which resolves a bug in HKDF, which may be relevant.

- When I did `pip install clkhash` in a new virtualenv (actually I used an anaconda environment, so it might be that), the packages listed by `pip freeze` (below) did not match those in requirements.txt – there are some additions and deletions. For some reason jsonschema, mypy_extensions, pytest and pytest-cov are missing from my list, although I see them in setup.py.

```
asn1crypto==0.24.0
bitarray==0.8.1
cffi==1.11.5
chardet==3.0.4
click==6.7
clkhash==0.10.1
cryptography==2.1.3
future==0.16.0
idna==2.6
pycparser==2.18
requests==2.18.4
six==1.11.0
tqdm==4.19.4
typing==3.6.4
urllib3==1.22
```