TEST REPORT DOCUMENT

Numerical Linear Algebra meets Machine Learning

Fabian Koffer

Simon Hanselmann

Yannick Funk

Dennis Leon Grötzinger

Anna Katharina Ricker

Supervisors

Hartwig Anzt Markus Götz

March 12, 2019

Contents

1 Overview

In this last section of our project, we wanted to get our project ready for usage by others. Therefore we first took a look at all the things we wanted to get finished before the project is being closed. After we got a list of tasks together, we created issues on GitHub together with a detailed descriptions on what needs to be done. This didn't just include bugs, but we also wanted to increase test coverage, remove code issues and set up some necessary things like a Wiki and documentation. After we created this issues, everybody could assign himself the issues he wanted to take care of. While doing so we also discovered some bugs. For these bugs we either created a issue or directly fixed them and wrote them into the bugs-report-table.

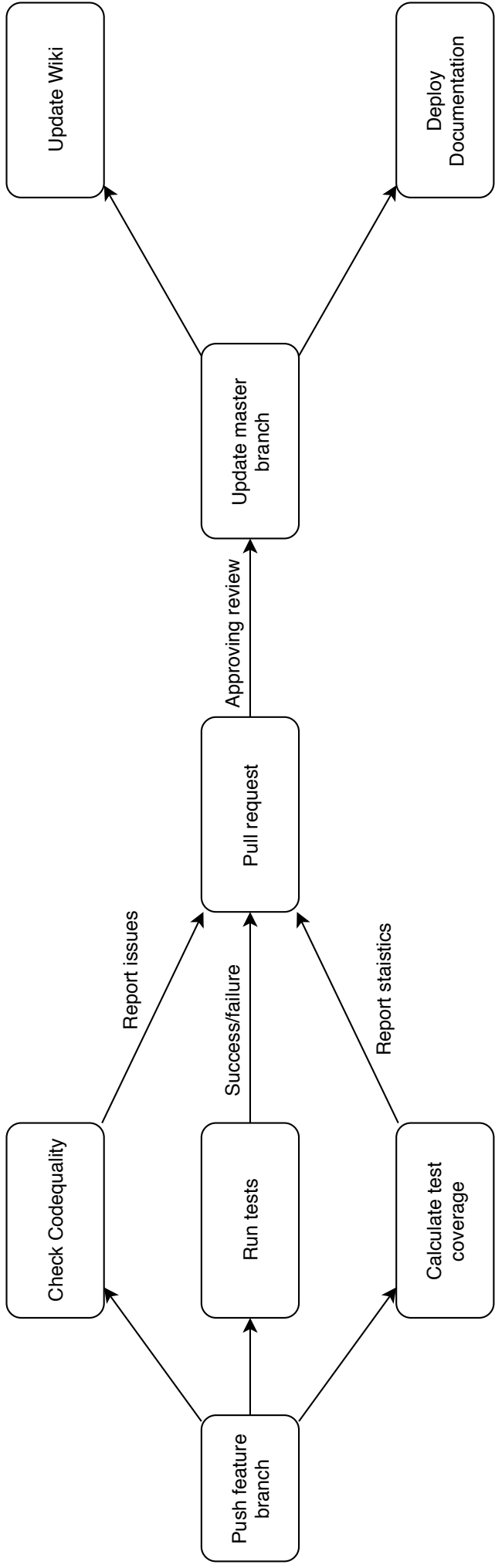# 2 Statistics

Lines of Code:

Python:

C++:

others:

Test Coverage:

3 Continuous Integration

To ensure a high quality on our code, we increased the building process of our code.

When pushing a branch, not just the test are ran on Travis, but also we generate a coverage report that will be sent to CodeClimate. This way you get a small overview on each pull request about the current coverage and the coverage on the newly committed lines. Added to that, you can get a complete overview on the CodeClimate Webpage.

After a successful merge to the master, we also introduced two new deployment steps. The first is that the wiki pages on GitHub are build again in case a change happened on this files. The second step is that we generate a code documentation that is generated and uploaded to GitHub pages.

```mermaid
flowchart

UpdateWiki[Update Wiki]
DeployDoc[Deploy Documentation]
UpdateMaster[Update master branch]
PullRequest[Pull request]
CheckCode[Check Codequality]
RunTests[Run tests]
CalcCoverage[Calculate test coverage]
PushBranch[Push feature branch]

UpdateMaster --> UpdateWiki
UpdateMaster --> DeployDoc
PullRequest -->|Approving review| UpdateMaster
CheckCode -->|Report issues| PullRequest
RunTests -->|Success/failure| PullRequest
CalcCoverage -->|Report staistics| PullRequest
PushBranch --> CheckCode
PushBranch --> RunTests
PushBranch --> CalcCoverage
```

4 Code Documentation

In the first phases we already decided to comment our code with the Doxygen-Syntax. In the implementation phase we took good care in documenting our code properly. The problem now was that everyone would have to generate this documentation for himself, which is a lot of work.

To easy things up, we added a deployment step, which generates this documentation and deploys it to GitHub pages.

This way everybody can have a look at the latest version of the documentation only be opening a link.

5 Wiki

We already collected some documentations for working with our project in the previous phases and decided to make them available in the GitHub wiki. Because the GitHub wiki holds its own GIT structure and we wanted to keep all our files in one place, we decided to integrate the wiki in our build process.

Therefore we created a folder in the projects root which holds all the wiki's entries in markdown files together with an configuration file. This configuration file is used to dynamically generate a sidebar that holds a navigation for the wiki entries. All the contents of this folder together with the generated sidebar are then copied to the wiki's git.

After this is done, the new wiki entries can be found on GitHub.

6 Bugs

Fault Symptom	Reason	Fix
Command line input with more than one space between arguments resulted in program crash	Input string would be split into a string list which had empty elements that caused problems	Changed the parameters of the string-split function for the expected behavior (remove spaces)
A corrupted configuration file caused the program to crash	Errors that are thrown while opening the file were not caught	Opening config file now happens in a try-except block and errors are reported to user (no crash)
Labeling or collection on operating systems other than linux resulted in crashes	The operating system was not checked when using the labeling or collecting module	Current operating system is checked at start of module and wrong operating systems are reported to user
Changing the size of the collected matrices was not possible	User entered size parameter was not used in collector	Removed static size declaration and started using the size input parameter
Default parameters are not correctly passed to the modules	The configuration file had a wrong format and could not be read properly	The configuration file got restructured and has it's default keys for each module
When not entering anything, the program crashes	The command parser tried to access a element in an empty array	Added extra check to prevent the invalid operation and displaying error message if check fails
Passing a not regular matrix to the classifier causes unexpected behavior	The regularity was not checked on the received matrix in the classifier	We added the regularity check to the classifier and a failure is reported to the screen is this check fails
Trying to open a not existing file in the loader results in a crash	The errors when opening a file where not caught properly	The loader now catches all errors and raises a IOException that can be caught when using the loader
Trying to enter different commands without setting the name flag, the program crashes	There is no default value for the name, so no default value was set	We added a method that sets a default name as a combination of a defaut name from the config file and the current date and time
The collector prints his own warnings	numpy prints his own warnings in the regularity calculator	Suppressed numpy warnings

7 Glossary

Glossary

CodeClimate A tool that monitors statistics about your code like coverage and displays it on each pull request on GitHub.

Doxygen A tool which uses you comments in the code to generate a documentation of the code.

GIT A version control system that can be used for tracking changes in a code repository.

GitHub A web based hosting service for the versioning control system GIT.

GitHub pages A web storage hosted by GitHub where you get a personal domain for your GitHub repository.

operating system A system software that manages computer hardware and software resources.

Travis A tool for continuous integration that is easy to integrate with GitHub.