The Shogun Detox2 – GSoC 2017 Proposal

WeiJie Lin

Organization: Shogun

Abstract

Shogun is a powerful Machine Learning toolbox, and it's been made by many different developers and geniuses researchers. However, this also implicates the trouble with Shogun that some parts of the codes are less optimized or "only getting things done". So we need to polish the codebase with new feature and new parameter framework to make it more maintainable and readable. This project will include new C++ feature usage, use Shogun internal data representations like SGVector and SGMartix to instead of plain pointer soluation, redesign data structure and its interface and add unit test as much as possible.

Contents

(drop my personal info in here)

- Name:
- Email:
- IRC nick:
- Twitter:
- Skype:
- Country of residence: China
- Timezone: UTC+08:00

Project Proposal

Since last year we had the new parameter framework created by sanuj, we are going to integrate it this summer. Furthermore, we have a lot of things to improve (include but not limit):

- **use SGVector instead of pointer Like this PR**
- **get rid of direct access of fields by using getter and setter #3688**
- **find the places we can use C++1x feature (e.g use nullptr instead of NULL and use smart pointer to avoid explicit type conversion)**
- **Progress bars and premature stopping, and redesign data classes if needed.**

- **Add more unit test to the codebase**.

  Right now the coverage of Shogun is 52%, it will be great if we can improve it to 65%~70% or more! We need to use one or more global fixture to instead of those data_genarator function and test case fixture to test function we want rather than init the module again and again.

Schedule of Deliverables

- **April 22 - May 23:(about four weeks)** Learn more about codebase, and discusswith the mentor and community on its interpretation. Locate the place we want to work with during this summer and clean up the class list.

- **May 23 - June 7:(Two weeks)** Refactor the **classifier** module:
  1. Stop accessing value like this and use getter and setter to instead of it
  2. Use nullptr to instead of NULL like this
  3. Use auto pointer in place like here
  4. All these pointers can replaced with SGVector
  5. Add more test for it.(we only have **5** unit tests with more than 20 modules

* **milestone: Each module under the classifier folder has been polished.**

- **June 8 - June 15:(one week)** Refactor the **clustering** module:
  1. Stop accessing value like this and use getter and setter to instead of it
  2. Use nullptr to instead of NULL like this
  3. Use smart pointer to replace SG_REF like this
  4. All these pointers can replaced with SGVector
  5. Add more unit tests for it.

* **milestone: Each module under the clustering folder has been polished.**

- **June 15 - June 29:(two weeks)** Refactor the **converter** module and **distance** module, polish its code like we did in classifier and clustering module, redesign data class and document it if needed.

* **milestone: Each module under the converter folder and distance folder has been polished.**

- **June 30- July 14:(two weeks)** Refactor the **evaluation** module and **features** module:

Beside to polish its code like we did before, I will also solve these issues related with evaluation and features module:

3612 Stratified Cross Validation with Combined Kernels using Custom Kernels
3727 Parallel cross validation is unsafe to use
3743 parallelize xvalidation
#1991 Possible CSVFile bug with high number of feature vectors

- **July 15 - July 21:(About one week)** Refactor the **metric** and **modelselection** module:
  Polish its code and fix
  # 3706 GradientModelSelection.select_model_ep_inference test fails on buildbot
  # 774 Define python typemaps for model selection

* **milestone: At this point, Mos of the modules in Shogun'scodebase has been polished**

- **July 21 - August 11:(three weeks)** Refactor the **multiclass** and **regression** modules.
  I found these related issues should take a close look at during this summer:
  #3686 Multiclass machines should work with binary labels
  #1880 Documentation of Shogun's multiclass framework

- **August 11 - August 23:(about two weeks)** Clean up the code and finish documentation.

Project Related Skills (1-5: rookie-guru)

- **C/C++ (2)**
  Finish the programing lesson of university. Self-learning on course and get involved into.

The C++11 features I know include:

- Lambda Expressions
- Automatic Type Deduction
- nullptr
- Shared_ptr and unique_ptr

- **Python(4)**
  Self-learning and get involved into several open source projects written by python.
- **Machine Learning(2)**

Probability and Statistics (graduate, grade 87/100)
Matrix theory (graduate, grade 80/100)
Advanced Mathematics (undergraduate, grade 96/100)
Machine Learning, taught by Andrew Ng (Coursera, no licence)

- **CMake(2)**
  Start to use if after working around Shogun and mlpack

- **SWIG(1)**
  No experience about it

Open Source Development Experience

- Accepted PR to Shogun, the rest of pr can been found in [1]:
  # 3724 Refactoring for shogun::memcpy
  # 3701 use get and set in SVM class
  # 3664 Use SGVector instead of pointer in KNN solvers and fix the warning when building KNNSolvers
  # 3641 add tests for KNN and fix an error in KDTree solver
  # 3608 Clean up KNN
  # 3736 clean up memcpy in swig typemaps

- Other open source experience
  I also do some contributions to mlpack**[2]** since couple of months ago. And I had joined into Mozilla Open Source Community for more than one year and get involved into several projects**[3]** in A-team of Mozilla. I used to take part in the Quarter of Contribution as contributor for Perfherder**[4]**, working on further improving the interface for usability and add new interface for it. At the same time, I'm one of active contributors for mozregression**[5]**, most works I do for mozregression are about polish the Gui and improve test coverage for it. In last October, I start my work for Mozci_tool**[4]** and release the Pushlog_client**[6]** as independent package**[7]**. In last summer, I worked on SETA**[8]** project, which is A-team's failure rate analysis tool for tbpl to help us find out which test job is necessary for detecting the failures, as GSoC student. As a part of job for it, we re-write SETA, deploy the whole ouija server on heroku and make it to support Taskcluster as well. All my work for SETA can been found in**[9]**.

Academic Experience

2010 – 2014 North China Institute of Aerospace Engineering, Undergraduate Student, major in Network Engineering

2015 - present Nanchang Hangkong University, graduate student of Software Engineering

Why Me

Like I said above, I have been working on Shogun since last October. So, I am already familiar with the Shogun-toolbox and know how to polish its code. At the same time, I love open source and get passion of it. And I believe the goal of Google summer of code is not only to attract students to get them involved into open source in this summer, but also encourage students to become solid contributors. I promise I will consist in working on Shogun even after this summer because there are ton of things can be learned from it.

Reference

[1] PR for shogun: https://github.com/shogun-toolbox/shogun/pulls/MikeLing

[2] PR for mlpack: https://github.com/mlpack/mlpack/pulls/MikeLing

[3] My bugzilla page:https://bugzilla.mozilla.org/user_profile?login=sabergeass%40gmail.com

[4] PR for Perfherder:https://github.com/mozilla/treeherder/pulls?q=is%3Apr+author%3AMikeLing+is%3Aclosed

[5] PR for mozregression:https://github.com/mozilla/mozregression/pulls?q=is%3Apr+author%3AMikeLing+is%3Aclosed

[6] PR for Mozci_tool:https://github.com/mozilla/mozilla_ci_tools/pulls?q=is%3Apr+author%3AMikeLing+is%3Aclosed

[6] PR for Pushlog_client:https://github.com/mozilla/version-control-tools/pulls?q=is%3Apr+is%3Aclosed

[7] Pushlog_client page:https://pypi.python.org/pypi/pushlog_client

[8] SETA: http://seta-dev.herokuapp.com

[9] Summary of SETA rewrite : https://mikelingblog.wordpress.com/2016/08/12/summary-of-seta-rewrite/