Masaomi Hatakeyama

Reort Summary

- Ruby/GSL library with NMatrix has become available
- Some RSecs have been created and updated

Contents

In this final report, I summarize the project overview, results, social impact and performance evaluation in the entire fellowship term.

Project Overview

This project includes mainly two tasks:

1. Maintain Ruby/GSL with NMatrix library
2. Update RSpecs (Unit-Tests)

The updating of Ruby/GSL library can realize the interoperability between GSL and NMatrix. It makes the potential of NMatrix usage get wider. The preparation of RSpecs definitely helps us to develop NMatrix more efficiently with less bugs.

Development environment

This is below the environment where I have developed:

- Mac OS X 10.8.2 (2.7 GHz Intel Core i7, 16 GB 1600 MHz DDR3)
- Ruby 1.9.3p374 (2013-01-18 revision 38874) [x86_64-darwin12.2.0] (on rvm 1.15.8)
- Ruby 2.0.0p0
- NAarray 0.6.0.7
- NMatrix 0.0.3
- Ruby/GSL 1.14.7

Results

At the beginning of this fellowship term, Ruby/GSL with NMatrix had many compile errors and it was not available at all.

Main results in this fellowship term are listed as follows:

1. All the compile errors of Ruby/GSL with NMatrix have been fixed and it is now available.
2. The total coverage of RSpecs are now reaching to almost 80%.
3. Some bugs have found by updating some RSpecs.
4. I have implemented a benchmark script to compare the efficiency of element-wise operations between GSL::Matrix, NMatrix, and NArray.

Social Impact

With the developing of Ruby/GSL library and RSpecs, the following effects can be taken into account for the social impact.

- Although there must be still some bugs in Ruby/GSL with NMatrix library, it is now on the level to be able to use.
- The benchmark script tells us which library is efficient and it is useful for the further development.
- Keeping RSpecs maintained and behaviour-driven development with RSpecs help us to develop NMatrix with safety and efficiency.

Performance Evaluation

- The success of Ruby/GSL compilation is one of the biggest results in this project.
- It took, however, longer time to debug the compile errors than I expected at the beginning. Therefore, RSpec development became slower than scheduled.
- It may be a good evaluated point that I have found some development styles in the NMatrix project through this fellowship project as follows:
    1. Ruby/GSL should be maintained at the same time with NMatrix development. Otherwise it will be incompatible and be difficult to maintain.
    2. Behaviour-Driven (Test-First) development should be taken into account. Otherwise the number of bugs will increase.

Some other comments and future plan

- Now the patches are merged with the main repository and it is possible to compile them even also in Ruby 2.0.0-p0, but unfortunately I got an error when GSL::Matrix#to_nm method is used. This should be fixed soon.
- As for the RSpecs, it is directly tests the Ruby source code but there must be many other lines and functions which are not tested in C source code level. In my opinion, we should take the test methods to check C functions into account in the future.