Development of machine learning infrastructures for Ruby ecosystem

Kenta Murata
Ruby World Conference 2016
Acknowledgement

- SciRuby JP survey members
  - Kozo Nishida
  - Makoto Hiramatsu
  - Yoshihiro Ashida
  - Yusuke Sangenya
  - Shinobu Kimura (ITOC)
  - Takuya Funo (ITOC)

- SciRuby JP survey sponsor
  - ITOC: Shimane IT Open-innovation Centor
  - Media Technology Lab., Recruit Holdings Co., Ltd.
Topics

• Why Ruby is not applicable for data science and machine learning tasks?

• How to make Ruby applicable for them?
Using Ruby for data science and machine learning

- I want to use Ruby for data science and machine learning works
- I use Ruby for almost all works for several years
- It is helpful if Ruby can be used for those types of works
Current status

• Ruby isn't applicable for data science and machine learning works

• Python is the first major programming language for machine learning

• What's the cause?
Why people select Python?

• Python has all necessary tools

• numpy, scipy, pandas, jupyter notebook, matplotlib, seaborn, scikit-learn, gensim, chainer, keras

• Infrastructure for computation, visualization, notebook, machine learning, deep learning are completed on Python

• They are well integrated via numpy array
Ruby?

• There are several libraries on Ruby:
  • numo-narray, nmatrix, daru, nyaplot, iruby, statsamples, etc.

• Two incompatible numerical array libraries prohibit to make integration among utilities

• Less functions

• Slow and incomplete functions

• Not production level quality
Why Python utilities are well integrated?

• IMO, the reason is Python community selected numpy as the only one numerical array library on Python in 2005
  
  • [http://www.slideshare.net/shoheihido/scipyhistory](http://www.slideshare.net/shoheihido/scipyhistory)

• There were two incompatible numerical array libraries so far

• Ruby's current situation is over 11 years behind
Other languages for data science

- R
- Julia
R

- R is the most powerful programming language for statistics including time-series analysis.

- It is also applied to machine learning, but Python is better than R.

- Data frames was first introduced as a first-class data type in R, but currently Python is the best for manipulating data frames due to pandas.

- R is general purpose programming language, but it isn't easy to use as Ruby and Python.
Julia

• A high-level, high-performance dynamic programming language for technical computing

• Julia has many attractive features for scientific computing: multiple dispatch, dynamic type system, lisp-like macros, parallel and distribute programming, high-performance JIT compiler

• I believe Julia will be the most major programming language for scientific computing 5 years after
**Figure**: benchmark times relative to C (smaller is better, C performance = 1.0).

C and Fortran compiled with gcc 5.1.1. C timing is the best timing from all optimization levels (-O0 through -O3). C, Fortran and Julia use OpenBLAS v0.2.14. The Python implementations of rand_mat_stat and rand_mat_mul use NumPy (v1.9.2) functions; the rest are pure Python implementations. Plot created with Gadfly and IJulia from this notebook.
Ruby

- Ruby is great programming language for implementing Web system because of Rails
- But Ruby is unsuitable for implementing algorithms for data science
- Python is also unsuitable, but Python libraries are implemented by C/C++ and Cython
What will be happen with the situation as it is?

• Python will take Ruby's market share on web

• Because the importances of data science and machine learning technologies get higher in businesses

• Python, especially pandas and scikit-learn, will be more important than Ruby and Rails in business

• Python engineers use Django or Bottle instead of Rails or Sinatra for building up Web system

• How to prevent this worst future?
Ruby's current situation

• Ruby is over 11 years behind Python:
  • Two incompatible numerical array libraries
  • Less integrated libraries, less features, low quality features

• Will it be improved by unifying numerical array libraries?
  • No, I don't think so
The biggest cause of problem: Negative feedback

- No tools
- No users
- No developers
Tools for data science

• Necessary features:
  • Useful numerical array operations
  • Large sparse matrix operations
  • Fast and complicated data frame operations
  • A wide variety of data visualizations
  • Well integrated GPU calculation

• The unified numerical array library is necessary, but not enough
Another problem is Time

• Unifying numerical array libraries is not easy task, need some months or over 1 year by the current SciRuby community

• We need not only to unify numerical array libraries, but also we need to change other utility libraries against the unification.

• Finishing to unify and rewire is not a goal, but just start line.
Breaking the negative feedback

- We should realize the environment that can be used for data science works in the real world for about 1 year

- And we should keep the environment up to date as Python and R so that users get established in a community

- How can we do that?
Stands on the shoulders of the giants

- Giants are Python, R, Julia, and so on
- In this way, I give up to make utilities for Ruby by myself
- Instead, I utilize the existing utilities of the giants
Stands on the shoulders of the giants

- gem libraries I'm going to make in this plan
  - num_buffer.gem
  - pycall.gem
  - pandas.gem
  - scikit-learn.gem
  - xgboost.gem
  - gensim.gem
  - matplotlib.gem
  - rcall.gem
  - julia.gem
  - etc.

- They makes the resources of Python, R, and Julia as a libraries made for Ruby
Schedule

• Until end of Dec. 2015

• pycall.gem version 0.2, including numpy integration

• scikit-learn.gem version 0.2, including LinearRegression, RandomForestClassifier, KFold, GridSearchCV, etc.

• rcall.gem version 0.2, including plotting support with iRuby integration
Schedule

• Until the end of Mar. 2017

• scikit-learn.gem version 0.4, including almost models in sklearn.linear_model and sklearn.ensemble, and some models in sklearn.cluster

• pandas.gem version 0.2 with basic data frame operations, and integration with daru

• julia.gem version 0.2 with basic operations

• I want to call for few contributors around of this period
More on Slack

• Let's continue this discussion in SciRuby slack

  https://sciruby-slack.herokuapp.com/

• I've given up to make our own utilities for Ruby, but almost all SciRuby slack members not

• I hope SciRuby community to get more lively
And ITOC booth
Conclusion

• Ruby is not applicable for data science and machine learning

• I'm working on development of utilities such as pycall.gem to realize the integration with existing great utilities of Python, R, and Julia

• I hope you are interested in this topic, come to SciRuby Slack, and discuss this topic