Alternative computational engines for PyMC3

Proposal For Google Summer of Code 2018

Mentor - Thomas Wiecki, Chris Fonnesbeck

Contact information:

Name : Sharan RY

University : BITS Pilani

Email-id : sharanyalburgi@gmail.com

Telephone: +91 91464 54088

GitHub/Gitter : sharanry

Discourse: sharan

Blog : sharanry.github.io

Time-zone : Indian Standard Time (UTC + 5:30)

Organization: NumFOCUS

Sub-organization: PyMC3

Abstract

PyMC3 is based on Theano, and uses it for creating and computing the graph that comprises the probabilistic model. Given the discontinuation of support for Theano, we are exploring using alternative libraries for future major versions of PyMC. We aim to port or re-implement some of the distributions currently present in PyMC3 using the selected framework while keeping the API, output and performance consistent.

Technical Details

Several frameworks have been considered as possible computational engine. Here is a summary of the discussions over the past few months.

Tensorflow

Backed by Google.

With the ready availability of TF.distributions. It will make it much more simple to integrate with tensorflow backend with the advantages provided by Edward2.

Edward2

Edward is a Python library for probabilistic modeling, inference, and criticism. It is a testbed for fast experimentation and research with probabilistic models, ranging from classical hierarchical models on small data sets to complex deep probabilistic models on large data sets.

With the latest commit, adding make_log_joint_fn to Edward2, it should make the transition towards TensorFlow based computational engine easier.

WIth TensorFlow being backed by Google the future of the frameworks is in safe hands. With Edwards's team's willingness to help and coordinate, future aspects of this project can be done smoothly.

PyTorch

Backed by Facebook.

With the recent availability of torch.distributions, we can easily port these to PyMC and apply a higher level API on top of this. However, performance and future viability needs to be investigated.

Dynamic graphs, numpy like syntax and easy GPU acceleration are some of the advantages of Pytorch

However, "MCMC can be hard to implement because of pytorch design. For now we collect all variables and compute joint_logp, that is tricky if you have dynamic graph. It would require us to change how models are defined" (Source)

Pyro

Backed by Uber. Pyro is a flexible, scalable deep probabilistic programming library built on PyTorch.

With the addition of HMC and other MCMC algorithms Pyro it is worthy option to consider. (Source)

CNTK

The Microsoft Cognitive Toolkit (https://cntk.ai), is a unified deep-learning toolkit that describes neural networks as a series of computational steps via a directed graph.

Small user base and lack of user familiarity, lack of surity of long term survival of the framework are major disadvantages of this.

MxNet

Availability of static and dynamic graphs, automatic parallelisation are some of the advantages.

However, small user groups and lack of some relevant features are the few disadvantages. (Source)

Implementation Details

Since the basic distributions have already been implemented in frameworks like Edward and Pyro. Our main focus is to match the API.

The main steps in this project will be:

1. Implement basic version of Model class first.
2. Port the basic distributions with existing PyMC3 API for ease of use for the existing users. (Try to maintain consistency with the current user experience)
3. Test these basic distributions exhaustively.
4. Move onto distributions like beta-binomial distribution distribution.
5. Test these distributions exhaustively.

Timeline

Pre-GSoC

Exploring the pros and cons of different frameworks. The ones which were considered are PyTorch, CNTK, TensorFlow and MxNet.

During GSoC

Note: My summer vacations start on May 13th just before the start of the GSoC coding period. This will enable me to turn my focus entirely towards this project.

Community Bonding	● Familiarize myself with he existing codebase. ● Reproduce simple example.

April 23, 2018 to May 14, 2018		• Deciding on which framework to use. • Sharing knowledge and experience is an integral part of the open source culture. I will be initiating my dedicated GSoC blog here.
Coding Period May 14, 2018 to August 6, 2018	Week 1 May 14 - 20	• Implement basic Model class like the one found here
	Week 2 May 21 - 27	• Port basic distributions like uniform, flat normal, exponential with the existing distributions in the chose framework preserving the structure of the current API.
	Week 3 May 28 - June 3	• Setup logp and dlogp. • Test them exhaustively.
	Week 4 June 3 - 10	• Test these port exhaustively for performance issues, and consistency using the existing python samplers.
	Evaluation 1 June 11 - 15	• Review the implemented API along with their tests. • Make modifications to the API/code implementation if necessary
	Week 6 June 16 - 23	• Extend the API to Beta, Gamma
	Week 7 June 24 - 30	• Continue with the implementation. Keep this buffer period for unexpected problems with the implementation.
	Week 8 July 1 - 8	• Test the implementation of the second set of models exhaustively using the different sampling methods. • Compare the results with existing models.
	Evaluation 2 July 9 - 13	• Review implementation and testing of Beta and Gamma. • Make modifications to the API/code implementation if necessary • Document the API
	Week 10 July 14 - 21	• There will be a lot of low-priority issues that will pop up during the

	development phase, which will be addressed during this period.
Week 11 July 22 - 28	● Polish the code further with suggestions from the maintainers This would be a buffer time for any unforeseen contingencies that can occur.
Week 12 July 29 - August 5	● Document the changes to the API ● Write a final blog about the release.

Submit Code and Final Evaluations August 6 - 14, 2018	● Have the PRs merged :)

Post GSoC

This phase will be mainly involve:

1. Help implement the rest of the distributions.
2. To maintain the code.
3. Fix any potential bugs.
4. Answer communities queries.

Development Experience

(Models Implemented, SOP, SWD, ARC)

Currently working on two research oriented study projects:

Conditional Random Fields for Image Segmentation [Link]

This has till now been a study oriented project in the field of conditional random fields. We are trying to understand their significance in the field of semantic segmentation of images. With this understanding we aim to contribute to furthering of its capabilities and use to the field.

Nested Mini-Batch K-Means [GitHub]

We are exploring the use of this new and improved unsupervised learning technique in the domain of visual recognition.

Student Welfare Division [GitHub]

A large scale application by our institute for the everyday usage of the students to increase ease of access of campuses facilities. Aim was to make it future proof.

Academic Research Division [GitHub]

A course management tool for the use of institute. This automates repeated tasks and increases the productivity of the staff.

Models Implemented [GitHub 1] [GitHub 2] [GitHub 3]

Links to some of the models which I have implemented.

Open Source Contributions:

I have previously contributed to PyMC and Scikit-Learn and Gensim.

My PRs are available here.

Pre-GSoC Involvement

Here is a list of PRs I have contributed to:

[MERGED]Fix examples in docs of mixture.py, bound.py and continuous.py

[MERGED]Fix effective sample size estimation (Partially Contributed)

[Open PR]Make Basic tensorflow.distributions model #4

Why Me?

I am an enthusiastic self-learnt student of machine learning and a growing open source enthusiast. I have experience working with large code bases and small teams of developers. I have previous experience in building deep learning models, and so I am familiar with popular frameworks. I feel it is these small but put together significant things that makes me a good fit to take on such an ambitious project.

Why PyMC3?

I am fascinated by bayesian and probabilistic learning and I love to code. This project enables me to learn and do both of these things at the same time. Being Math and CS major I find that I am in ideal position to take up project like this.

Acknowledgement

Thanks to Thomas Wiecki and Junpeng Lao for their time and guidance.

Appendix and References

[1] https://discourse.pymc.io/t/tensorflow-backend-for-pymc4/409/13

[2] https://arxiv.org/abs/1711.10604