Software Engineering for Machine Learning
NIPS 2014 Workshop

Ryan Turner
Northrop Grumman Corp.
ryan.turner@ngc.com

Joaquin Quiñonero Candela
Facebook
joaquinq@fb.com

Xavier Amatriain
Netflix
xavier@netflix.com

We are organizing a one day NIPS 2014 workshop that will cover topics at the intersection of machine learning and software architecture/engineering. This intersection is a critical area for deploying machine learning methods in practice, but is often overlooked in the literature. As a result, much of the publicly available code for download is disorganized, undocumented, and buggy. Therefore, it cannot serve as an example of how actual deployed machine-learning-heavy software should be written. Those looking to implement actual software could greatly benefit from a workshop that can provide guidance on software practices.

There are several topics this workshop will cover through contributed and invited talks:

1. Scaling machine learning: Solutions to practical issues involving taking single machine algorithms and making them ready for “big data” by distributing them with Spark or Hadoop/MapReduce are welcome here.

2. Accelerating machine learning prototypes: Methods and tips for moving single machine Matlab/R/Python code to C/C++ code, as well as GPU acceleration.

3. Software paradigms: When is it best to work in an object oriented, procedural, or functional framework when developing machine learning software?

4. When to use probabilistic programming environments? If so, which tool (e.g. Infer.NET, Stan, Church, etc.) is most appropriate for your project requirements?

5. Systematic testing: This is often overlooked but important area for the workshop to cover. Can we develop better methods for systematically testing our methods to make sure they are implemented correctly? This includes unit testing and regression testing.
   (a) There is a perception among some practitioners that systematic methods like unit tests are not applicable to machine learning because “The whole reason we are doing the computation in the first place is that we do not know the answer.” One goal of this workshop is to try and change that perception with guidance and examples.
   (b) What are some of the common ways to break down a machine learning project into units where unit testing is possible? Monte Carlo unit tests: Unlike most projects many unit tests in machine learning are Monte Carlo tests.
   (c) Different inference methods will have their own methods that can be used to test their implementation correctness: VB, EP, MCMC [3; 1], etc.

6. Documentation: How should people in machine learning be doing a better job at documenting their code? Do the usual guidelines for software documentation need to be augmented or modified for machine learning software? Could tools such as literate programming [4] be more useful than the typical documentation tools (e.g. Doxygen or Javadoc)? We could examine issues involving requirements documents [2] for machine learning algorithms.

7. Advice for machine learning people in interfacing with traditional software designers. What are common misunderstandings and things we should be ready to explain?

8. Collaboration on machine learning projects. For instance, platforms that make it easy for engineers to reuse features and code from other teams make every feature engineer much more impactful.
9. Issues with regard to open source in machine learning. Talks involving intellectual property issues in machine learning would also be welcome.

10. Getting data into the software development process is also a possible talk. Handling organization restrictions with regard to security and privacy issues is an important area.

11. Building automatic benchmarking systems. A critical part of machine learning project is to first setup an independent evaluation system to benchmark the current version of the software. This system can ensure that software is not accidentally “peaking” at the test data. Other subtle issues include excessive benchmarking against a test set which could result in overfitting, or not placing any confidence intervals on the benchmarks used. Machine learning competitions can provide some guidance here.

12. Methods for testing and ensuring numerical stability. How do we deal with numerical stability in deployed, or real-time, software systems?

13. Differences between using machine learning in client side vs. server side software. Processing the training set client side, and outside of the designers control, poses many more challenges than only processing the test set on a client machine.

14. Reproducibility: What advice do we have for making machine learning experiments completely reproducible? This is largely an extension of using revision control systems and procedures for logging results.

15. Design patterns: What advice is there for utilizing ideas, for example from Gamma et al. [2], in machine learning projects?

Many of the above items are about utilizing and adapting advice from tradition software development, such as from McConnell [5].

References


Current Speakers (Confirmed)

1. Hussein Mehanna and Stuart Bowers (Facebook): *Facebook Machine Learning Platform*
2. Anmol Bhasin (LinkedIn): *Leveraging Machine Learning Open Source at LinkedIn*
3. Andreas Müller (University of Bonn): *scikit-learn*
4. Daniel Lee (Columbia University): *The Stan Project*
5. Justin Basilico and Aish Fenton (Netflix): *Machine Learning at Netflix*

Further Speakers: We are seeking further invited speakers who have developed machine learning systems that have been deployed in actual products used by thousands (or millions) or users. We seek to have a diverse set of speakers: those who have worked in open source, probabilistic programming, server side machine, or client side machine learning projects. We are interested in contributed talks from speakers who can provide novel insight into the 15 topics mentioned above.
Schedule
08:30  Workshop Intro (Ryan Turner and Joaquín Quiñonero Candela)
09:30  Leveraging Machine Learning Open Source at LinkedIn (Anmol Bhasin)
10:00  Facebook Machine Learning Platform (Hussein Mehanna and Stuart Bowers)
10:30  Break
11:00  scikit-learn (Andreas Müller)
11:30  Contributed Talk
12:00  Lunch Session and Contributed Posters
15:00  The Stan Project (Daniel Lee)
15:30  Reproducibility/Intellectual Property Issues (Victoria Stodden)
16:00  Machine Learning at Netflix (Justin Basilico and Aish Fenton)
16:30  Break
17:00  Contributed Talk
17:30  Contributed Talk
18:00  Panel Discussion
18:30  End of Workshop