Deploying predictive models with the Actor framework

Brian Gawalt

October 15, 2015

Abstract

The majority of data science and machine learning tutorials focus on generating models: assembling a dataset; splitting the data into training, validation, and testing subsets; building the model; and demonstrating its generalizability. But when it's time to repeat the analogous steps when using the model in production, issues of high latency can arise. To an end user, the cost of too much time spent featurizing raw data and evaluating a model over features can wind up erasing any gains a smarter prediction can offer. Exposing concurrency in these model-usage steps, and then capitalizing on that concurrency, can cut down on that latency cost. This work describes how the Actor framework can be used to bring a predictive model to a real-time setting. Two case-study examples are described: a simple text classifier (with accompanying code), and a live deployment built for the freelancing platform Upwork.

1 Machine learning in production

When a firm hires a machine learning specialist, you can be pretty sure that the firm would like to start generating predictions about the future. Accurate predictions, to the extent possible. Whose face is appearing in this recently uploaded photograph? Which of these credit card transactions are fraudulent? Who among our inactive users is likely to log back in, should we send them a "We Miss You!" email?

The internet is chock full of resources to help someone hone their skills in answering these challenges. Coursera courses can provide excellent primer material on up through graduate-level instruction. Kaggle contests let one practice those learned techniques on real data, making real predictions. To move oneself to the state of the art, one can follow the proceedings at NIPS, KDD, ICML, SIGIR – plenty of venues to describe new modeling techniques and algorithms.