Model Evaluation and Deployment;
How Things Work in Industry

Conner DiPaolo

1 Introduction

This course is focuses on the mathematical approach to solving problems involving a large amount of data, primarily using techniques from statistical machine learning, but also building on non-statistical background such as K-Nearest Neighbors or Support Vector Machines that reignited interest in the field. That said, many people in the course have interest in moving to industry to work with data there. This document is an eclectic collection of techniques that, while not relevant to the learning goals in this course, are must-haves for anyone looking to use machine learning techniques in industry or even performing applied analysis in academia. Note that while these techniques might speed up task completion in industry, following these are not necessarily conducive to generating good research in academia.

2 Model Complexity

This is going to depend on where you are working, but unless your company is doing more advanced work, everyone is likely to only use linear models in production. A notable exception is that when working with images you're almost guaranteed to use a convolutional neural network these days. This dichotomy between the set of models that academia works on an industry works with (except neural networks) comes down to a choice in efficiency. Complicated models are, in general

(i) complicated to train (and therefore implement),

(ii) might not have libraries available to train for you,

(iii) non-trivial to store from a model-representation standpoint,

(iv) and require more effort to maintain.

The product of all of these issues brings us to a cardinal rule when working with a task that seems to require machine learning: **try the simplest models that could possibly work first. If the simple model doesn't work well, try a more complicated model only then.** This is a key departure from academic thinking. In industry, the goal is to add features to a product. It follows that having 95% accuracy (or some other metric) is totally fine, even though using a more complicated model might let you eek out a 2.5% higher score on that metric. At it's core, time (to implement *and* maintain models) is prioritized higher than model performance. For a detailed and well thought out analysis of the hidden costs of machine learning that motivate this intellectual sparsity see Sculley et al.'s work, "Machine Learning: The High-Interest Credit Card of Technical Debt"[1] out of Google.

3 Model Evaluation

3.1 Metrics

3.2 Cross Validation

4 Hyperparameter Optimization

5 Working With Text Data

[1] http://static.googleusercontent.com/media/research.google.com/en//pubs/archive/43146.pdf