Practical machine learning notes

Week 1

**Prediction motivation**

We’ll cover the main techniques and practicalities of doing machine learning on real examples. Implementation will be with the caret package in R. Classes like the other modules in the Data Scientist’s Toolbox are complementary to this course.

Who uses machine learning and prediction? Everyone from tech companies (like Netflix predicting what movies you might like to watch) to insurance companies (who predict what your risk of death is). Prediction can be useful in healthcare, e.g. predicting outcome in breast cancer patients. There are now also lots of data science competitions, e.g. the $1m Netflix prize for improving the predictive algorithm. Kaggle organizes many competitions.

The Elements of Statistical Learning is a recommended book. My experience is that the book is quite tough and heavily mathematical – I would recommend “Introduction to Statistical Learning” by the same authors, which was written precisely to make it more accessible. The “Machine Learning” coursera course by Andrew Ng (we might study it next term!).

The R package caret is a nice package that integrates a lot of different tools related to machine learning.

**What is prediction?**

Broad overview of prediction: We build some sort of prediction function that uses characteristics of samples to predict what category they fall into. Note that this is a classification task, another huge area of ML (machine learning) is regression problems.

Google Flu Trends tried to predict flu epidemics based on what terms people were searching for online. Algorithm was not well adapted to the nuances of people’s usage of search terms, leading to highly inaccurate results. Gives an idea that choosing the right dataset and defining the question are highly important.

In ML, we must start with a well defined question that can be answered with data. Then we have to gather the input data, and from that data we identify the features that we would like to use to predict the output. We choose an appropriate ML algorithm, estimate the parameters of the algorithm, and apply the algorithm on new data.

An example: can we use characteristics of emails to classify them into spam emails and not-spam? To build a predictive frequency, we need to get quantitative features out the emails. For example, we can get frequencies of words for many spam and non-spam (“ham”) emails. Could threshold on frequency of the word “your” (see code).

**Relative importance of steps**

The steps of doing ML include defining a question, finding data, getting features and applying an algorithm. The most important part is defining a good question and collecting the data. Often, the algorithm is the least important component.

Question: it is important to emphasize. An important component of doing prediction is knowing when to give up, when you just don’t have the data to answer the question you want to address.

Input data: “Garbage in, garbage out”. If you have bad data, no matter how good your algorithm, the result still won’t be great. The best results will come from when your training data is basically the same as what you’re trying to predict (e.g. Netflix prize was trying to predict future movie ratings based on old movie ratings, which are very similar things). Having better data is a foolproof way of doing better prediction.

Features: features should be informative, relevant and may be created using specialist domain knowledge. Trying to automate feature selection (including not noticing quirks of the data) can lead to issues. There is a body of research on automatic feature discovery (e.g. there is an algorithm that automatically identified features identifying whether there are cats or humans in a YouTube video).

Algorithms: ML algorithms matter less than you might think. E.g. comparing a basic algorithm (a basic linear discriminant analysis predictor) against the best possible method did not result in massive differences. Using a sensible, basic method will get you a massive part of the way there. The choice of algorithm should also be influenced by interpretability and simplicity, because you’ll probably have to explain it to non-technical people. Scalability and speed are also concerns that fall outside of just the model’s predictive performance. Prediction is about tradeoffs between e.g. interpretability vs. accuracy, or speed versus accuracy. The winning solution of the Netflix £1m prize was never implemented the final algorithm in production because of scalability problems, it was a blend of many ML algorithms and took a long time to compute. Netflix went with an algorithm that was less accurate but was more scalable.

**In and out of sample errors**

This concept is really central to ML. In sample errors (resubstitution error) are the errors that you get on the same data as you used to build your predictor. This estimate of error is optimistic, because the predictor will tune itself to the noise in that particular data set (overfitting). The out of sample error rate (generalization error) is more informative because it is a more realistic estimate of how the predictor will perform on new data.

See code for demonstration in difference between in sample and out sample errors. We find that the simplified predictor does a better job of predicting spam emails when applied to the larger dataset. The goal of a predictor in general is to capture the signal in a dataset and not overfit to the noise. If you tune the predictor too tightly to a dataset, it will not perform as well on a new dataset.