Peter McKay, Daniel Walinsky

2015-02-19

Project Proposal

For our final project, we intend to tackle the following Kaggle problem: the Forest Cover Type Prediction. In brief, we wish to, given a collection of geographic and cartographic variables that apply to a given 30 x 30 meter chunk of forest, determine what kind of tree coverage will be most prevalent in that region.

Examples of data fields we have access to include the soil type (Cryaquolis, Granile, Rock outcrop), the elevation, and average slope.

We are faced, then, with a multi-class classification problem. From such data fields in our testing file, we wish to discover whether a particular cell can be categorized as Spruce/Fir, Lodgepole Pine, Ponderosa Pine, Cottonwood/Willow, Aspen, Douglas-fir, or Krummholz.

Our project timeline, at a high level is as follows:

1. Acquire data, ensure integrity thereof, massage into useful data structures.
2. Explore data, discover correlations, store hypotheses for later use
3. Test different machine learning algorithms based on the previous steps.
4. Settle on an approach (or combination of approaches) that return a maximally effective result based on the a comparison to the testing data.
5. Draw conclusions about the nature of our data, and problem as a whole, based on the data from all earlier steps.

As we proceed through this process, we intend to maintain notes at each stage, in order to parallelize, as much as possible, the experimentation and paper writing processes.

Our first step must be to acquire the data and examine it, in order to discover what sort of model would best suit our data. To that end, we must first settle upon a set of tools to assist us in this matter. We plan to carry out most of our research using Python. More specifically, Python 3.4.2, with the help of a number of libraries designed to assist in machine learning and data analysis.

pandas: Python Data Analysis Library: We will make use of data structures and input mechanisms introduced in pandas. These data structures, while containing a great deal of complex functionality, extend iterables with sufficient cleanliness to allow for compatibility with most other python libraries.

matplotlib: The first of these such libraries that sounds helpful is matplotlib. matplotlib extends python with some matlab-like functionality, allowing us to generate graphs and examine the data in question. Should we need to scale the data, or remove a number of corrupted entries, matplotlib should give us an idea of where to start.

numpy: numpy provides a collection of very useful tools for analysis, which will come in handy as we examine the variables, and any correlations they may have with one another. numpy and matplotlib play quite nice together, and this will allow us to plot a number of interesting functions over the space of the data, testing hypotheses before we get to the proper machine learning part of the assignment.

scikit-learn: In an effort to avoid duplicating the wheel (and running out of RAM while we try to get that wheel to load), we will be making use of scikit-learn for implementations of the canonical algorithms as we test their efficacy.

While it may be poetically appropriate to make use of, say, an ensemble of random forests, decision trees, and nearest-neighbor by way of cover-tree, we will be deciding on our choice of algorithm(s) later in the course of our research.