Data Mining Hospital Readmissions and Mortality Rates Using Multiagent Random Forest

Andrew Berry

September 8, 2016

I pledge my honor that I have neither given nor received unauthorized aid on this work.

1 Project Overview

My goal for this project is to apply multiagent data mining on a public dataset containing hospital readmission and mortality data. Specifically, I plan to apply a multiagent implementation of the random forest algorithm, an ensemble-learning-based improvement over the decision tree algorithm. While decision trees (and therefore random forests) can be used for multiple machine learning tasks, I intend to use them for classification in this project.

The dataset I am using comes from the Center for Medicare and Medicaid Services (CMS). It contains 64,764 rows of data regarding 4,434 distinct U.S. hospitals. Measures contained in the dataset correspond to 30-day readmission and mortality rates of the following conditions: acute myocardial infarction (i.e. heart attack), coronary artery bypass grafting (i.e. heart surgery), chronic obstructive pulmonary disease, heart failure, pneumonia, and stroke. In addition, there are measures for hip/knee replacement operations readmission rates and hospital-wide readmission rates (regardless of patient condition). Data about the location of the hospital and comparison to national averages are present as well.

Multiple agents will implement the random forest algorithm, learning over a common training set (a specified subset of the original dataset) and using what they learn to predict classifications of new rows of data. In addition to this, the agents will communicate with each other, collaborating on their findings to improve each others' accuracy. Specifically, agents will review some of the same records in the test set and "vote" on which classification to assign to the hospital in question. Whichever classification has the highest number of votes for that record is deemed to be the system's "answer" to the question, and can be evaluated by comparing the system's answer with the actual value from the original dataset.

The specific question I intend to explore is "given the readmission and death metrics of a hospital, can the agents accurately predict what state the hospital is in?" Please note that I will only consider the readmission and death metrics in my analysis; I will ignore the city, county, address, and other obvious geographic data, since those would readily give the answer away to the agents (e.g. if we were analyzing a hospital that has City = "Albuquerque" the system could easily guess New Mexico without even having to

consider the readmission and death metrics). The suspicion that readmission and death metrics have a correlation with the state in which the hospital is located is based on a few factors. First, I hypothesize that doctors in any given hospital are statistically more likely to have graduated from the nearest medical school to that particular hospital (or at least a medical school in the same state) rather than from elsewhere, which may have strengths and weaknesses in its curriculum regarding different medical procedures. Second, the amount of funding for public hospitals will vary by state. Likewise, societal factors in the area (the economy, average education level of citizens) likely have a noticeable effect on health outcomes, with wealthier areas having better access to adequate healthcare, and more educated patients having the health literacy to understand when to consult medical counsel for their symptoms.

To evaluate the system for correctness of implementation, a simple dataset (a "dummy" dataset) with a verified strong correlation between measures and classification categories will be used to verify that the learning algorithms are being properly applied. This will help ensure that, should the final results of the project reveal that there is very little reliability in predicting the state of a hospital, it will be due to lack of correlation between readmission/mortality rates and state, rather than being a product of faulty implementation of the algorithms. The performance of the final system (reliability of the predictions offered by the system) will be measured by calculating the number of times a prediction is correct over the total number of predictions (percentage of predictions correct). This metric can be evaluated on an for the system as a whole, but also can be evaluated on an agent-by-agent basis and any agent found to consistently score lower than desired can have its algorithm re-tuned to improve its performance.

The implementation of this project will use the Python programming language, the pandas library for data analysis, the scikit-learn library for machine learning and data visualization, and the SPADE agent platform for multiagent aspects of the implementation. As development progresses, I may find that other tools and/or frameworks will be very useful in completing the project. Those will be added to this document as they become known.

2 Tentative Project Schedule

- Early-Mid September - Establish a working environment using pandas, scikit-learn, and SPADE. Work through tutorials in each of the tools to gain working familiarity. Apply non-distributed random forest algorithm to the dummy dataset to ensure the algorithm works as intented.

- Mid-Late September - Use pandas to organize the dataset into the appropriate format. Begin using scikit-learn to perform a non-distributed run of the random forest algorithm on the dataset.

- Early-Mid October - Lay groundwork for multiagent setup in SPADE, implement multiagent random forest against the dummy dataset, including a check of percentage of predictions correct to ensure it works as intended.

- Mid-Late October - Apply multiagent random forest algorithm to the hospital dataset

- Early-Mid November - Fine tuning of agents within the system to increase accuracy (if possible); begin work on final report
- Mid-Late November - Finish project and prepare presentation; finish final report writeup for the Nov 29 deadline

3 Relevant Links

- This project's webpage
- The hospital readmission and death dataset.
- Python
- Pandas
- Scikit-learn
- SPADE