Austin Ban and Patrick Haller

Dr. William Hooper

Artificial Intelligence

December 3, 2016

Bayesian Probability

To begin our overview of Bayesian probability, we will begin with a quick overview of its conception and original use. Bayes Algorithm was named after Reverend Thomas Bayes (1702–61), but was not originally met with renown and awe. Thomas Bayes did not publish his work during his lifetime, though his friend Richard Price edited and presented this work in 1763. The scientific world took notice of Bayes work only after publishing Price's late friend's paper under the title "An Essay towards solving a Problem in the Doctrine of Chances." It is worth noting that the tragic story of genius only being recognized post-mortem is repeated, which gives hope to us struggling scientists that someday our genius will be recognized and spread amidst the masses.

After Richard Price presented Reverend Thomas Bayes' work, there were dozens of key characters who helped to progress the understanding and application of Bayesian probability, leading to a dramatic growth of research in the 1980s when the discovery of Markov chain Monte Carlo methods eliminated a lot of computational problems that existed beforehand, allowing Bayesian probability to be used in expanded ways. These breakthroughs led to the eventual application of Bayesian probability in machine learning, which is why this paper is being written in the first place.

This project uses a naive Bayesian algorithm, which is an implementation of Bayes theorem that assumes independence between the data, or predictors. It is the simplicity of a naive Bayes network that makes it so powerful. Because of the limited moving parts, it is very adept at working with larger datasets. While the data we used with our project is not nearly large enough to qualify as “big data,” the speed of calculation and ease of use are still apparent and useful in the implementation. In fact, this machine learning is done in linear time, making it hard to surpass in terms of speed.

The application of Bayesian algorithms is limited by the nature of its often inaccurate results, but again, its simplicity allows for it to be used practically and accurately in the right situation. The nature of its function means each query can be estimated as a one-dimensional query, solving issues that arise from exponentially complex networks that require more dimensions as the number of queries or datasets increase, known as the curse of dimensionality. It is because of all of these features that made it perfect for this class project.

This project was done using CORGIS (The Collection of Really Great, Interesting, Situated) datasets categorized by state. There were several databases loaded in, providing state based information on crime data, including assault rate, murder rate, rape rate, robbery rate, burglary rate, larceny rate, and motor theft rate. There was also data on wealth, including median household income and persons below poverty level. Education includes state funding, attendance rate, bachelor's degree or higher, and high school or higher. Ethnicity data includes American Indian and Alaska native alone, Asian alone, black alone, hispanic or latino, native Hawaiian and other Pacific Islander alone, two or more races, and white alone.

Using that data, we created a dictionary to house all the fields that we thought would be interesting. We modeled it after the "joint" dictionary that was used in the aartiste folder. After loading everything from three different databases into one joint dictionary, we had to create an intersection dictionary that would get rid of all the records that didn't have data from all three datasets. There were states in some of them that weren't in others. One of the datasets had a total record with the state named "United States." After getting the intersection working, we just had to add it into the dataframes. We started by creating a generic dataframe but eventually implemented one called "YourExample" that had all of the data readily available. All we had to do to swap the used fields was comment out the old ones and remove the comment on the new ones. We took the print_table function that was included and changed it to also print the data into an html file. The html file has the file name of the example, and it contains all the data that would have printed out so that we could then easily copy it later into charts and link them together. With more work, we could have implemented a GUI input system that would allow the user to select the fields they want to show and the html file that it spits out would automatically create itself into a chart.

When it comes to how to display this information gathered by the Bayes search, a wild goose chase began that led us to partially install a few Python based GUI systems that wound up being more trouble than good. The one that we spent the most time testing is Plotly (https://plot.ly/python/). Plotly had great looking tests and seemed promising to be able to quickly plot information onto a publically accessible, interactive plot with a unique URL. It seemed like the dream situation, and above all else, it was free. Unfortunately, it became rather complex quite quickly, and due to time constraints, we decided to scrap the idea, but keep it in

mind for future projects that have a greater timeline attached to them. Deciding to play to our strengths, outputting the Python to HTML became the path of choice. Using the robust and flexible charts.js (http://www.chartjs.org/) library, we suddenly had a surplus of ways to display the information, ranging from scatter plots to bar charts with everything in between. The issue arose when trying to output data live to the HTML. Though it was possible, the only way that we could think to do such a thing would be to create HTML partials that would all compile down into a single HTML file that renders the chart. This would unfortunately require running a partial engine in tandem with the python body of the project. This would be too complex for the timeline with too many moving parts to expect it to work consistently without robust testing, so static HTML files became the poison of choice by the end of the project.

This project yielded some interesting results, showing some surprising correlation between the independent datasets, as well as taught us how to work with Bayesian probability significantly better. This class also had the great benefit of providing either an introduction or reinforcement of Github, which benefited us greatly while working on this project remotely from each other.

Annotated Bibliography

Russell, Stuart J., and Peter Norvig. Artificial Intelligence: A Modern Approach. Upper Saddle River: Prentice-Hall, 2010. Print.

The general concept of the Bayesian Network is what we used from this book. This was our main reference for learning how the Bayes Network operates. We also pulled an image onto our poster that helped to explain the concept to the people at SURS.

"Welcome to the Kennel." CORGIS Datasets. N.p., n.d. Web. 30 Nov. 2016.

We used this site to pull three different datasets. The state demographics dataset, the state crime dataset, and the education dataset. We combined those three to create one bigger dataset for our project. They put the work in to make sure the datasets were all formatted relatively the same.

"Bayes Theorem (aka, Bayes Rule)." *Bayes' Theorem*. N.p., n.d. Web. 30 Nov. 2016.

On this website, we used the part with Example 1. This section really helped our poster be able to explain the process better. Under the “Let’s get technical” section on our poster, we tried to go into enough detail to explain the bayes rule and theorem to the people that walked up.

"An Example." *An Example | STAT 414 / 415*. N.p., n.d. Web. 30 Nov. 2016.

We included this specific example on the second section of our poster. This was used as a more detailed version of the first example. This one focuses more on the Bayes Theorem that uses the prior probability.

User7997. "Origin of the Naïve Bayes Classifier?" *Stack Exchange*. N.p., n.d. Web. 30 Nov. 2016.

We were able to learn more about the history of Bayes Networks with this post. This was in the cross reference section of stack exchange. It covers the classifier and the theorem used in the early and modern Bayes systems.