Classifiers with Bayes

 (https://github.com/learn-co-curriculum/dsc-classifiers-with-bayes) (https://github.com/learn-co-curriculum/dsc-classifiers-with-bayes/issues/new/choose)

Introduction

Now that you're familiar with Bayes' theorem and foundational concepts of Bayesian statistics, you'll take a look at how to implement some of these ideas for machine learning. Classification tasks can be a natural application of Bayes' theorem since you are looking to predict some label given other information, which can be conceptualized through conditional probability.

Objectives

You will be able to:

- Explain the assumption that leads to Naive Bayes being "naive"
- Explain how to use the probabilities generated by Naive Bayes to make a classification

Naive Bayes

Naive Bayes algorithms extend Bayes' formula to multiple variables by assuming that these features are independent of one another. This then allows you to estimate an overall probability by multiplying the conditional probabilities for each of the independent features.

For example, extending the previous medical examples of Bayes' theorem, a researcher might examine multiple patient measurements to better predict whether or not an individual has a given disease. Provided that these measurements are independent (and uncorrelated from one another), one can then examine the conditional probability of each of these metrics and apply Bayes' theorem to determine a relative probability of having the disease or not. Combining these probabilities can then give an overall confidence of a patient having the disease given all the information. From this, one can then make a prediction for whether or not you believe an individual has the disease or not based on which probability is higher.

Mathematically, if Y is a class you wish to predict (such as having a disease) and $X_1, X_2, \ldots, X_n$ are the various measurements for the given individual or case, then the probability of class Y can be written as:

$$P(Y|X_1, X_2, \ldots, X_n) = \frac{P(X_1|Y) \cdot P(X_2|Y) \cdot \ldots \cdot P(X_n|Y)}{P(X_1, X_2, \ldots, X_n)} P(Y)$$

Again, note that multiplying the conditional probabilities is based on the assumption that these probabilities (and their underlying features) are independent -- and it is this assumption that the Naive Bayes algorithm is considered naive, or simple, because this is almost never true. However, Naives Bayes can prove to be quite efficient given the right circumstances, as you will see in the upcoming lessons.

In practice, calculating the denominator, $P(X_1, X_2, \ldots, X_n)$ is often impractical or impossible as this exact combination of features may not have been previously observed. However, doing so is often not required. This is because when implementing a classifier, the exact probabilities themselves are not required to generate a prediction. Instead, you must simply answer which option is the most probable. To do this, you would calculate $P(Y_0)$, the probability of not having the disease as well as $P(Y_1)$, the probability of having the disease. Furthermore, since the denominator, $P(X_1, X_2, \ldots, X_n)$, is equal for both $P(Y_0)$ and $P(Y_1)$, you can compare the numerators, as these will be proportional to the overall probability. You'll investigate this further as you code some Naive Bayes classification algorithms yourself in the upcoming lessons.

Summary

In this lesson, you briefly explored how Bayes' theorem can be used to build classification algorithms. In the upcoming lessons and labs you'll investigate particular implementations of Naive Bayes classifiers which differ in how the individual conditional probabilities themselves are constructed. As you will see, Naive Bayes can be extremely effective or trivially useful depending on the context and implementation.

How do you feel about this lesson?

Have specific feedback?

Tell us here! (https://github.com/learn-co-curriculum/dsc-classifiers-with-bayes/issues/new/choose)