Computational Mathematics for Data Analytics - CS-550

Eugene Pinsky
Department of Computer Science
Metropolitan College, Boston University
Boston, MA 02215
email: epinsky@bu.edu

November 3, 2021

Abstract

Mathematics is fundamental to data science and machine learning. This course reviews essential mathematical concepts and procedures which are fundamental. These concepts are illustrated by Python and/or R code and by many visualizations. This course discusses mathematical concepts and computational methods for data science using simple self-contained examples, intuition and visualization. These examples will help develop intuitive explanations behind mathematical concepts. Extensive visualizations will be used to illustrate core mathematical concepts. The emphasis is both on mathematics and computational algorithms that are at the heart of many algorithms for data analysis and machine learning. This course will advance students mathematical skills that can be used effectively in data analytics and machine learning.

Assignment

In principle, any density can be used to implement Naive bayesian classifier. The default for continuous variables is to use normal distribution. In this assignment, we consider implementing Naive Bayesian using Student-t distribution:

https://en.wikipedia.org/wiki/Student%27s_t-distribution

Student-t distribution is charcterized by 3 parameters:

1. degrees of freedom - dfrac

2. location parameter μ

3. scale parameter s^2 (similar to variance in normal)

We can estimate these parameters in Python as follows:

```python
from scipy import stats
# assume data is in array x
df, location, scale = stats.t.fir(x)
```

To compute the density function for some value z we would use:

```python
value = stats.t.pdf(z, location, scale)
```

When degrees of freedom is large, Student-t distribution approaches normal. When $df \mapsto 0$, we get a distribution similar to normal but with fatter tails. In this assignment, we will investigate whether we can improve our accuracy if we use Student-t distribution and consider fatter tails.

You task is to implement a Naive Bayesian classifier with Student-t with $df = 0.5, 1, 5$). For each week, your feature set is (μ, σ) for that week. Use your labels (you will have 52 labels per year for each week) from year 1 to train your classifier and predict labels for year 2. Use Student-t for the prediction.

Questions:

1. implement a Student-t Naive Bayesian classifier ($df = 0.5, 1, 5$) and compute its accuracy for year 2
2. compute the confusion matrices for year 2
3. what is true positive rate and true negative rate for year 2
4. what is the best value of df? Is it better than normal Naive bayesian
5. for the best value of df, implement a trading strategy based on your labels for year 2 and compare the performance with the "buy-and-hold" strategy. Which strategy results in a larger amount at the end of the year?