MATH 728D: Machine Learning Lab #12: Naive Bayes Classification

John Burkardt

January 27, 2019

Can I estimate complicated probabilities if I only have simple observations?

The naive Bayes classifier makes the assumption that the values of each feature are independent. That means that you estimate the probability that two features have particular values by computing their separate probabilities and multiplying. As a simple first approximation, this seems reasonable, but it can lead to some peculiar results. If you sampled 1000 people, and for the feature "gender" half were female, while for the feature "has a beard" half listed "True", then this procedure would estimate the chances of a bearded female as 25%! However, the naive Bayes classifier is often preferred because of its simplicity, and its ability to process very large amounts of data rapidly.

1 A Jumbled Deck of Cards

Suppose we find a drawer full of hundreds of playing cards, and we randomly collect 50 of them into a "deck". Cards have a rank between 1 and 13. Number the suits hearts=1, clubs=2, diamonds=3, and spades=4. Since the mixture of cards is uneven, we have to do a little work to answer questions about the probability of certain events.

When you answer the following questions, assume that, one at a time, five people each pick a card from then deck, look at it, and then replace it.

Useful MATLAB commands:

- Number of diamonds: sum (s==3);
- Number of cards greater than 6: sum (r>6);
- Number of clubs less than 5: sum (s==2 & r<5);
- Probability of a 9 for suits 1:4: sum (r==9 & s==[1:4]) ./ sum(s==[1:4]);
- Probability that a 9 is suit 1:4: sum (r==9 & s==[1:4]) / sum (r==9);
- Probability of a 4 of hearts: sum(r==4 & s==1)/sum(r==r)/sum(s==s);
- Naive probability of a 4 of hearts: sum(r==4)*sum(s==1)/sum(r==r)/sum(s==s):

Exercise 1:

- Use `csvread()`, skipping row 1 and column 1, to create `data`;
- Copy columns 1, 2 and 3 of `data` into arrays `r, s, o` for rank(1-13), suit(1-4), and order(1-52);
- Use `histogram` on the `o` array to get an idea of how scattered the cards are;
- Alice draws a card from the deck. What are the chances it beats a 10?
- Bob draws a card; it is a club. What are the chances it beats a 10?
- Charlie draws a 10. What is the likeliest suit of his card?
- Dave picks a card that beats an 8. What is its most likely suit?
- Ed picks a heart; What is its most likely rank?

- What is the probability of picking a 2 of diamonds?
- What is the naive probability of picking a 2 of diamonds?

2 Computer Center Hotline Reports

Assume that you are working the computer center hotline. Users are calling you about a program, available in C and Python, which may be causing some systems to crash. You have already answered 20 calls, recording in a file the operating system, the programming language in use, the browser, and whether or not there was actually a crash.

We want to use this data to build a probabilistic model of what is going on.

Exercise 2:

1. Use `csvread()` to read *crash_data.csv* into `data`, skipping row 1 and column 1;
2. Extract columns 1, 2, 3, 4 of `data` to:
   (a) os: the operating system (1=Linux, 2=OSX, 3=Windows);
   (b) lang: program language (1=C, 2=Python);
   (c) browser: (1=Chrome, 2=Explorer, 3=Firefox, 4=Safari);
   (d) crash: was there a crash? (0=No, 1=Yes);
3. Use this information, and naive Bayes techniques, to answer:
   (a) What is the most probable operating system of your caller?
   (b) If your caller says they use the Chrome browser, what is the most probable operating system they are using?
   (c) If your caller says they use the Chrome browser, and they were running the C program, estimate the probability that they had a crash.
   (d) Answer the previous question again, but using a naive Bayes approach;
   (e) If your caller uses Chrome and C, but is willing to make one switch. which combination, Chrome+Python, or Firefox+C, seems a lower probability of crashing?
   (f) Answer the previous question again, but using a naive Bayes approach;
   (g) The caller has a friend whose computer crashed. What is the probability the friend was using Windows and Python?
   (h) Answer the previous question again, but using a naive Bayes approach;

3 Fraud in Loan Applications

A bank has records of 20 loan applications, including

1. ID: 1-20
2. CH: credit history, "none", "paid", "current", "arrears";
3. GC: guarantor; "none", "guarantor", "coapplicant";
4. AC: accommodation: "own", "rent", "free";
5. FR: fraud? "true", "false";

Use the data to determine various necessary probabilities. Then decide whether a new applicant should be suspected of intending fraud.

Exercise 3:

- Set `data = readtable ( loan_data.csv );`

- Extract columns of the cell array data:

```
CH = data{:,2};
GC = data{:,3};
AC = data{:,4};
FR = data{:,5};
```

- determine `pFRt` and `pFRf`, the probabilities of fraud being "true" or "false";
- determine `pCHnFRt, pCHpFRt, pCHcFRt, pCHaFRt`, the probabilities that, if fraud is "true", the credit history is "none", "paid", "current", or "arrears" respectively;
- similarly determine `pCHnFRf, pCHpFRf, pCHcFRf, pCHaFRf`;
- determine `pGCnFRt, pGCgFRt, pGCcFRt, pGCnFRf, pGCgFRf, pGCcFRf`;
- determine `pACoFRt, pACrFRt, pACfFRt, pACoFRf, pACrFRf, pACfFRf`;
- Use these probabilities to determine whether to suspect an applicant of fraudulent intent, given CH="paid", GC="none" and AC="rent";

4 Tax Cheaters

So far, the datasets we have examined in this exercise have been discrete (a small collection of integer possibitilies) or categorical ("single", "married", "divorced"). In the following exercise, involving tax returns, one of the features measures reported income, and thus is essentially a continuous variable. In order to compute a probability for incomes, we will model them as normally distributed. Thus, we will separate compute a mean and variance for incomes in the Cheating="true" and Cheating="false" classes. The probability of any income can then be determined by evaluating the normal PDF.

Exercise 4:

1. Set `data = readtable ( tax_data.csv );`
2. Extract columns of the cell array data:

```
RF = data{:,2};              %  Refund requested
ST = data{:,3};              %  Marital status
IN = double ( data{:,4} ); %  Income in thousands
CH = data{:,5};              %  Taxpayer determined to be cheating?
```

3. Determine `pCHt`, the probability that a given taxpayer was cheating;
4. Determine `pRFtCHt, pRFfCHt`, the probabilities that a cheating taxpayer requested a refund;
5. Similarly, determine `pSTsCHt, pSTmCHt, pSTdCHt`, the probabilities that a cheating taxpayer was single, married, or divorced;
6. Determine `INmuCHt, INstCHt`, the mean and standard deviation of the incomes of cheating taxpayers;
7. Similarly, determine probabilities and other quantities `pCHf, pRFtCHf, pRFfCHf, pSTsCHf, pSTmCHf, pSTdCHf, INmuCHf, INstCHf` associated with a noncheating taxpayer;
8. Given, for a new taxpayer, the data `RF="false"; ST="married", IN=120;`, use your computed probabilities to determine whether the taxpayer is cheating or not. In particular, report the two probabilities that you compute and compare.