Engsci 213 Probability Assignment

Oliver Maclaren
oliver.maclaren@auckland.ac.nz

Due: 29th April 2016 (3pm)

Overview

You are a software engineer and wish to use your knowledge of probability theory to develop an artificial intelligence (AI) system to carry out some simple tasks. You decide to call it BetaStop.

You will first document some simple 'best practice' probability calculations, doing them by hand yourself. This will help you set some benchmarks for the AI to aim for; it will also give you some case studies to think about how probability relates to 'information' and 'decisions', which are (surely!) key concepts for creating a 'learning machine'.

Finally, you will make a simple attempt at teaching the AI to 'learn for itself' by giving it a simple probability model to make predictions about the world and a learning algorithm to update these predictions based on new information.

Task 1 [10 marks]

Background

A key benchmark problem for probability calculations is the processing and interpretation of various types of test results. Luckily, your friend in the Bio-engineering department has just proposed a new test for detecting a disease D; you decide to help evaluate it.

You know that you need to use conditional probability to reason carefully about these sorts of tests.

Problem data

The new test under consideration has the following characteristics

- For people who have disease D it gives a 'positive' (hence correct) diagnosis 97% of the time
- For fully healthy people it gives a 'positive' (hence incorrect) diagnosis 4% of the time
- For people who are not healthy but do not have disease D it shows a 'negative' (correct) diagnosis 92% of the time

- Each test results in either a correct or an incorrect diagnosis

The sample space is the set of people tested (and their associated properties).

Define the events P = "Person has a positive test result", N = "Person has a negative test result", H = "Person is fully healthy", D = "Person has disease D" and S = "Person is sick (not fully healthy) but does not have disease D".

~~The test will initially be applied to a trial population with 95% fully healthy people and 5% not fully healthy people. Of the latter set, 1.3% have disease D.~~

The above text is now replaced by:

The test will initially be applied to a trial population with 95% 'fully healthy' people and 5% 'not fully healthy' people. People with disease D are a subset of the 'not fully healthy' people and comprise 1.3% of the total population.

Questions

(a) If you select a person at random from the above population and administer the test, what is the probability that the person will test positive?

(b) Given that the result of a test is positive for a person, what is the probability that they have the disease?

(c) Given that the result of a test is negative for a person, what is the probability that they do not have the disease?

(d) Is the information that a positive test occurred 'relevant' to knowing whether they have disease D? Justify your answer in terms of probabilities.

(e) Is the information that a negative test occurred 'relevant' to knowing whether they are free of disease D? Justify your answer in terms of probabilities.

Task 2 [10 marks]

Background

Next you decide that your AI should be able to do more than just process given data - it should be able to make 'decisions' that take into account costs and benefits of future outcomes under different decision choices and different predictions for the future.

Emboldened by your success on the previous diagnostic test problem you enquire with a local medical practice whether they could use your services and they provide you with the following problem.

Problem data

The medical practice is concerned about another disease 'C' and needs to decide which of two treatments to use for a given patient having this disease. Every patient has a gene which comes in one of three 'types'. The type of gene carried is known from lab studies published in the research literature to change the probability of the outcomes from the treatment type (the outcome is still uncertain in general). Unfortunately the medical practice that you are consulting for does not have access to the appropriate genetic testing facilities.

The same research literature mentioned above contains a survey used to measure a given patient's 'quality of life' following treatment on a scale of -50 to +50. A value of zero is considered neutral. This survey was carried out on patients at a large clinic which had access to the genetic testing facilities. Results were aggregated (i.e. converted to discrete possibilities) to get an idea of the quality of life score for a typical patient with a given gene variant. These three scores and their probabilities are summarised in the following table

Quality score	'Gene variant 1'	'Gene variant 2'	'Gene variant 3'
Value after treatment 1	45	-10	-40
Value after treatment 2	-5	0	20
Probability of gene variant	0.4	0.5	0.1

You assume that the above table is indicative of a randomly encountered patient in the practice you are consulting for, and define the (discrete) random variable V as assigning a 'quality of life score' to a given patient according to the given (conditional) probabilities.

You decide to use this information to decide on the 'best' courses of action under different scenarios and also to try to quantify the 'value of information' provided by the genetic test.

Questions

(a) What is the expected value of the quality of life score for a randomly selected patient assigned treatment 1 (with no additional test information of any type)?

(b) What is the expected value of the quality of life score for a randomly selected patient assigned treatment 2 (with no additional test information of any type)?

(c) Which treatment would be preferable given no other information?

(d) Suppose you (or the doctor, really!) did have access to the results of the 'perfect' genetic test before choosing a treatment. You could then choose

the best treatment for each possible outcome of the genetic test. What is the expected quality of life score of a randomly chosen patient under this scenario? This is called the 'expected value given perfect information'.

(e) What is the difference between the expected value given perfect information and the expected value without this additional information? This is called the 'expected value of perfect information'.

Task 3 [10 marks]

Background

Now that you've established some basic principles of probabilistic reasoning you want to think about how your AI might be able to 'learn from data' and assimilate new observations into its knowledge base. This could (eventually) allow you to use your AI to diagnose and decide courses of treatments for patients.

To begin, you decide to give the AI a simple probabilistic 'prediction generating model' that it will use to predict outcomes in the real world and then, based on how well these match reality, update its predictions and 'state of knowledge' about the world.

Problem data

You decide to use the Binomial model as a simple 'prediction generating model' for the AI and you set up the following learning scenario:

- You generate some data using the Binomial model for a specific (known to you) choice of parameters p and n.
- You tell the AI that you used $n = 25$ and a Binomial model to generate the data but you don't tell it which probability of success (p^*) you used.
- Instead, you will give it the data you generated. Call this y_0. In particular, you get the result $y_0 = 15$ successes in $n = 25$ Bernoulli trials and will give this to the AI.

Clearly multiple possible values of p are compatible with the given data $y_0 = 15$, so you give the AI the following learning algorithm to infer a 'score' to assign to each possible p. This will represent what the AI thinks the observed data tell us about the 'likelihood' that each p is the 'true' p^* that you used to generate the data:

Learning Algorithm

- For each candidate value of p in the discrete grid [0,0.1,0.2,...,1.0]
  - Generate a large number (100 say) of 'hypothetical' predictions of the data, $y_{new}^{(1)}, y_{new}^{(2)}, ...y_{new}^{(100)}$ where each is a random sample from the Binomial distribution Bin($n = 25$,p).

- Count the number of $y_{new}^{(i)}$ values generated that are equal to the original data y_0.
- Record this count as a 'score' for how well this value of p predicts the original data.

Questions

(a) Write up the above 'learning algorithm' in R or your language of choice. You should submit all your code.

Hint: you can generate random samples from the binomial distribution in R using rbinom (see https://stat.ethz.ch/R-manual/R-devel/library/stats/html/Binomial.html). For other languages you can search the documentation.

(b) Use the data $y_0 = 15$ to generate a count of the number of times each candidate parameter p predicts y_0 (in 100 predictions, where each is a random sample from Bin($n = 25$,p)).

(c) Plot a histogram (graph of counts) showing the number of successful predictions for each candidate value of p.

(d) Which value(s) of p seem like reasonable candidates for the 'true' p^* used to generate the original data? You can justify this informally based on your plot.

(e) If your histogram represents the AI's 'post-data' state of knowledge about the true parameter, what was the AI's 'pre-data' 'state of knowledge' about the true parameter? (i.e. what was the initial assumption about possible values of p before being shown the data y_0). Give this in terms of a probability distribution over p.