R computing for Business Data Analytics

Instructor: Dr. Howard Hao-Chun Chuang

Assistant Professor, Department of Management Information Systems

National Chengchi University, Taipei, Taiwan 116

chuang@nccu.edu.tw

Last Revised: August 2014

3.1 Basics of probability

- Some necessary definitions

Experiment: An observational process that leads to a number of possible outcomes, which cannot be predicted with full certainty.

Sample space: All possible outcomes of an experiment, usually denoted by S or Ω.

The sample spaces fall into three categories.

Finite: There are only a finite number of possible outcomes.

Infinite discrete: Sample spaces are countably infinite, e,g., positive integers.

Continuous: Sample spaces correspond to finite or infinite interval of real numbers.

Event: Any subset of S, denoted by E.

$P(E)$ denotes the probability of the event E.

- Examples

Toss two coins and look for at least one head.

Experiment:

S:

E:

Inspect IC chips from a manufacturing process repeatedly.

Experiment:

S:

E:

Run a computer program and measure its CPU time.

Experiment:

S:

E:

- Calculating probabilities

A **classical approach to probability** for finite sample spaces:

$$P(E) = \frac{\text{number of points in } E}{\text{number of points in } S} = \frac{n(E)}{n(S)}$$

This definition assumes that each individual outcome *x* has is equally likely (e.g., a fair coin).

$$x \in S \Rightarrow P(x) = \frac{1}{n(S)}$$

Example: A group of four IC chips consists of two good chips and two defective chips. The experiment consists of selecting three chips randomly from this group. Write down S and E. What is the probability that two are defective?

S:

E:

P(*E*):

A **relative frequency approach to probability**, on the other hand, interprets the probability as the relative frequency of the event over a long series of experiment. That is, repeat the experiment over and over again, and then count how many times an event occurs. Define

$$N_n(E) = \text{ the number of times } E \text{ occurs in } n \text{ repetitions}$$

$$\text{where } N_n(E) \in \{0, 1, 2, ..., n\}$$

Accordingly, we can define

$$P(E) = \lim_{n \to \infty} \frac{N_n(E)}{n}$$

The statement of "limit" is for mathematical convenience and NOT supposed to be taken lightly from a computational standpoint. Let's simulate probabilities of tossing a coin.

```
> x=sample (c("H", "T"), 10, replace=TRUE) #n=10
> table(x)
> table(x)/10
> x=sample (c("H", "T"), 100, replace=TRUE) #n=100
> table(x)/100
```

How about increasing the number of tosses to 10,000 times?

Example: Snoopy and Charlie Brown are gambling against each other. A UNfair coin is tossed repeatedly (where P(Head)=0.6 and P(Tail)=0.4). Each time a head comes up, Snoopy wins a dollar from Charlie Brown. Otherwise Snoopy loses a dollar to Charlie Brown. Carry out the experiment 50 times, and estimate the number of times that Snoopy wins in those 50 tosses. How much has Snoopy won or lost?

```
> x=sample (c("H", "T"), 50, replace=T, prob=c(0.6, 0.4))
```

Since a head means +1 for Snoopy and a tail means -1, an equivalent code would be

```
> x=sample (c(1, -1), 50, replace=T, prob=c(0.6, 0.4))
```

To calculate the number of times that Snoopy wins

```
> sum(x==1)
```

To see how much money Snoopy has in its pocket

```
> pocket=c()
> pocket[1]=x[1]
> for (i in 2:50){
> pocket[i]=pocket[i-1]+x[i]
> }
> pocket
```

A plot of this is

```
> num=1:50
> plot(num, pocket, type='o', xlab= "Toss number", ylab= "$")
```

- Permutations

Permutations are ordered samples or sequences of a particular size that can be chosen, without replacement, from a population. The number of ways to choose ordered samples of size k from n is given by

$$P_k^n = \frac{n!}{(n-k)!} = n(n-1)...(n-(k-1))$$

It can be calculated in R using *prod*(*n*: *n*-*k*+*1*)

Example: If the set is of size 3, we may write the points as $\{1, 2, 3\}$. The possible ordered sequences of size 2 are:

Example: A series of 10 jobs arrive at a computing center with 10 processors. Assume that each job is equally likely to go through any of the processors.

(a) What is the probability that all processors are occupied?

(b) What is the probability that at least one processor will receive two or more jobs?

(c) How many processors will be needed if we want to be 90% confident that no processor will receive more than one job?

- Combinations

Combinations are unordered samples or sequences of a particular size that can be chosen, without replacement, from a population. Generally,

$$C_k^n = \binom{n}{k} = \frac{n!}{k!(n-k)!} = \frac{P_k^n}{P_k^k}$$

It can be calculated in *R* using *choose*(n, k)

Example: Pick elements for $\{1, 2, 3\}$. The number of unordered sequences of size 2 is:

Example: If a box contains 75 good IC chips and 25 defective chips from which 12 chips are selected at random, find the probability that all selected chips are good.

Experiment:

S:

E:

3.2 Rules of probability

- Probability and sets

Union: $\cup$

Example: E_1=Spade, E_2=Ace, $E_1 \cup E_2 =$ Spade *or* Ace. $P(E_1 \cup E_2) =$

Intersection: $\cap$

Example: E_1=Spade, E_2=Ace, $E_1 \cap E_2 =$ Spade *and* Ace. $P(E_1 \cap E_2) =$

Mutually exclusive events

Example: E_1=Spade, E_2=Heart, $E_1 \cap E_2 = \varnothing$. $P(E_1 \cap E_2) =$

Complementary events

Example: E=Ace, $\overline{E}$= All those cards that are not aces. $P(E \cup \overline{E}) =$

- Axioms of probability

A probability function P, defined on subsets of the sample space S, satisfies the three axioms

1. $P(E) \geq 0$, for all $E \subset S$
2. If $E_1 \cap E_2 = \varnothing$, then $P(E_1 \cup E_2) = P(E_1) + P(E_2)$
3. $P(E \cup \overline{E}) = P(S) = 1$

Following the three axioms, we can derive several properties (which you can prove).

1. For each $E \subset S$, $P(\overline{E}) = 1 - P(E)$
2. $P(\varnothing) = 0$ where $\varnothing$ is the empty set
3. If $E_1 \subset E_2 \subset S$, then $P(E_1) \leq P(E_2)$
4. For each $E \subset S$, $0 \leq P(E) \leq 1$
5. If $E_1 \subset S \,\&\, E_2 \subset S$, then $P(E_1 \cup E_2) = P(E_1) + P(E_2) - P(E_1 \cap E_2)$

Example: A computer system uses passwords of five characters, each being one of a-z or 0-9. The first character must be a letter. Assume passwords are NOT case sensitive.
Find the probabilities that (A) a password begins with a vowel (a, e, i, o, u); (B) ends with an odd number (1, 3, 5, 7, 9); and (C) begins with vowel or ends with an odd number.

3.3 Conditional probability

- Definition

The *conditional probability* of an event E_2 given an event E_1 is defined as

$$P(E_2 \mid E_1) = \frac{P(E_2 \cap E_1)}{P(E_1)}$$

where $P(E_1 \cap E_2)$ is the *joint probability* and $P(E_1)$ is the *marginal probability*.

Example: Four firms A, B, C, D, are bidding for a contract. The probabilities to win are:

$$P(A) = 0.35,\ P(B) = 0.15,\ P(C) = 0.3,\ P(D) = 0.2$$

Find $P(A \mid \overline{B}),\ P(C \mid \overline{B}), P(D \mid \overline{B})$ and check that $P(A \mid \overline{B}) + P(C \mid \overline{B}) + P(D \mid \overline{B}) = 1$

- Multiplication law of probability

Following the definition of conditional probability, we can show that

$$P(E_1 \cap E_2) = \begin{cases} P(E_1)P(E_2 \mid E_1) \\ P(E_2)P(E_1 \mid E_2) \end{cases}$$

The law can be further generalized to:

$$P(E_1 \cap E_2 \cap \cdots \cap E_k) = P(E_1)P(E_2 \mid E_1)P(E_3 \mid E_1 \cap E_2) \cdots P(E_k \mid E_1 \cap E_2 \cap \cdots \cap E_{k-1})$$

Example: Draw three cards from a deck *without* replacement, the probability all are black

- Independent events

E_2 is said to be independent of E_1 if

$$P(E_2 \mid E_1) = P(E_2)$$

For k>2 independent events

$$P(E_1 \cap E_2 ... E_k) = P(E_1)P(E_2)...P(E_k)$$

Example: Draw three cards from a deck *with* replacement, the probability all are black

- Theorem: Law of total probability

If a sample space S can be partitioned into k mutually exclusive and exhaustive events, A_1, A_2,… A_k, then for any event E

$$P(E) = P(A_1)P(E \mid A_1) + P(A_2)P(E \mid A_2) + \cdots + P(A_k)P(E \mid A_k)$$

We can write E as $E \cap S$, *and since the entire* $S = A_1 \cup A_2 \cdots \cup A_k$, *we have*

$$\begin{aligned} E &= E \cap S \\ &= E \cap (A_1 \cup A_2 \cdots \cup A_k) \\ &= (E \cap A_1) \cup (E \cap A_2) \cdots \cup (E \cap A_k) \end{aligned}$$

The events $E \cap A_1, E \cap A_2, \ldots, E \cap A_k$ *are mutually exclusive sine A_1, A_2,… A_k are. Hence,*

$$\begin{aligned} P(E) &= P(E \cap A_1) + P(E \cap A_2) + \cdots + P(E \cap A_k) \\ &= P(A_1)P(E \mid A_1) + P(A_2)P(E \mid A_2) + \cdots + P(A_k)P(E \mid A_k) \end{aligned}$$

Example: In a certain company, 50% of documents are written in Word, 30% in Latex, and 20% in Html. It is also known that

40% of the Word documents exceed 10 pages;

20% of the Latex documents exceed 10 pages;

20% of the Html documents exceed 10 pages.

Let *E* be the event that a randomly selected document exceeds 10 pages. What is *P*(*E*)?

- Applied probability

The Intel fiasco

In October 1994, the Pentium chip was found to produce an incorrect result when dividing two numbers, despite Intel claimed that "such an error would occur once in 9 billion divides" for a typical user. However, the chip was withdrawn eventually. Why did that happen?

Let *E* be the event that an error will occur

```
> error=1/9000000000
> noerror=1-error
> noerror
```

What about no error occurs in two divides?

> (noerror)^2

What Intel had not anticipated was that, for heavy software and/or users, billions of divisions over a short time span are not unusual.

What is the probability that at least one error occurs in 1 billion divisions?

How about 2 billion divides?

Trees

For a binary communication channel, 70% of messages are transmitted as 0 and 30% of messages are transmitted as 1. A transmitted 0 is correctly received with probability 0.95 and a transmitted 1 is correctly received with probability 0.75. Define

R_0: The event that a 0 is received. T_0: The event that a 0 is transmitted.

R_1: The event that a 1 is received. T_1: The event that a 1 is transmitted.

Find the probability that (a) a 0 was received; (b) an error occurred.

3.4 Bayes theorem and posterior probability

- Bayes' rule for two events

Consider two events A and B. From the multiplication law we know

$$P(A \cap B) = P(A)P(B \mid A)$$
$$P(B \cap A) = P(B)P(A \mid B)$$

Since $A \cap B = B \cap A$

$$P(A \cap B) = P(B \cap A)$$

Therefore

$$P(A)P(B \mid A) = P(B)P(A \mid B)$$

We can rearrange the terms

$$P(A \mid B) = \frac{P(A)P(B \mid A)}{P(B)}$$

This is called Bayes' rule for two events, and it enables us to compute the probability of A after B has occurred.

Example: Go back to the binary communication channel. 70% of messages are transmitted as 0 and 30% of messages are transmitted as 1. A transmitted 0 is correctly received with probability 0.95 and a transmitted 1 is correctly received with probability 0.75. Define

R_0: The event that a 0 is received. T_0: The event that a 0 is transmitted.

R_1: The event that a 1 is received. T_1: The event that a 1 is transmitted.

Given a 0 was received, what is the probability that it was transmitted as a 0?

- Bayes' theorem

If a sample space S can be partitioned into k mutually exclusive and exhaustive events, A_1, A_2,… A_k, *then*

$$P(A_i \mid E) = \frac{P(A_i)P(E \mid A_i)}{P(E)} = \frac{P(A_i)P(E \mid A_i)}{P(A_1)P(E \mid A_1) + P(A_2)P(E \mid A_2) + \cdots + P(A_k)P(E \mid A_k)}$$

Example: In a certain company, 50% of documents are written in Word, 30% in Latex, and 20% in Html. It is also known that

40% of the Word documents exceed 10 pages;

20% of the Latex documents exceed 10 pages;

20% of the Html documents exceed 10 pages.

Let E be the event that a randomly selected document exceeds 10 pages. What the probability that it has been written in Latex?

- A Bayesian Application

Machine learning

There are two classes, y=1 and y=2 into which we can classify items. If the system has already learnt probabilities from the past, we can use Bayes' theorem to classify f, a new value of the item.

There is not so much *R* in this lecture per se. Nonetheless, probability theory is crucial for you to understand data analysis techniques in *R*. You can almost always apply probability distributions (which will be covered in lecture 4) to model observations in order to *learn something from data*.