Probability distributions: Course manual

EBP038A05

Nicky D. van Foreest

2021-01-14

Contents

1 Material

The book's homepage provides an electronic copy of the book, video lectures by Blizstein, and a solution manual for part of the problems. I expect you to watch Blizstein's lectures for the overview and general background of the book.

I advise you to buy the book as it makes studying easier, and it is very practical to look up things later.

Blitzstein gave me the advice to use *named* results, as this sticks much better. I fully agree with him, so here are some more names:

- VOS law (Ch 7): Variance of sums: $\mathsf{V}[X+Y] = \mathsf{Cov}[X+Y, X+Y] = \mathsf{V}[X] + \mathsf{V}[Y] + \mathsf{Cov}[X,Y] + \mathsf{Cov}[Y,X]$.
- LOTE (Ch 9): Law of total expectation: $\mathsf{E}[X] = \sum_i \mathsf{E}[X \mid A_i] \, \mathsf{P}[A_i]$.

The document `memoryless excursions.pdf` (henceforth referred to as ME) discusses the core concepts of BH.7.1 (BH = Blitzstein and Hwang), and contain worked solutions to show you the type of computations you are supposed to master.

The document `notes.pdf` contains notes that I took while reading the book and lots of hints for the exercises of BH. It contains also hints for exercises that are not in the list below. It's up to you to do these other exercises. I worked hard on making this set of hints, and I include many remarks and recipes (general strategies) I used to solve the exercises. As you'll notice, I prefer to stay clear of quick and/or slick solutions. Sure, such solutions can provide fast and elegant answers to hard questions, but such tricks often don't work in more general settings, and they depend on knowledge that takes a long time to acquire. In fact, tricks give people (students and me alike) the feeling that they will never learn probability because they don't know these 'shortcuts'. So, it's better to avoid that altogether, and for that reason I tend to use plain and straightforward, but perhaps long and tedious, derivations. You should learn and master the computational skills as you will need them time and again in many other courses.

With a bit of searching you can find lots of information on the web, such as here,

- `https://github.com/buruzaemon/stats-110`
- `https://github.com/buruzaemon/IntroductionToProbabilityPy`

However, use this at your own risk, I haven't checked it.

2 Schedule

In our on-line lectures, we will illustrate the material by solving problems. It's not necessary that you have solved the problem before the lecture, but you should have at least read them, and have watched the relevant lectures of Blizstein and are up to date with the book.

Table 1: schedule

Lecture	section	Exercises
1	7.1, ME	1, 10, all exercises of ME
	7.1, ME	11, 12, 13, 15, 26,
3	7.2, 7.3	33, 46, 48, 58, 64
4	7.4, 7.5	65, 66, 73 77, 86
5	8.1	1, 11, 12, 14, 16
6	8.2	18, 22, 23, 27
7	8.3, 8.4	33, 34, 36, 37
8	8.5	40, 52, 54
9	9.1, 9.2	1, 3, 7, 12, 15
10	9.2, 9.3	16, 18, 19, 21, 25
11	9.5, 9.6	37, 39, 42, 52, 55
12	9.6, 10.1	56, 58, 1, 2, 4
13	10.2	6, 9, 15, 21, 26
14	10.3, 10.4	27, 29, 30, 35, 39

During the lectures we will discuss problems and show how to solve them. There will be one tutorial group during which you can ask question. This group is indicated as 'Probability Distributions Pr. gr.01' in the schedule.

5 opgaven per college lijkt me redelijk. Kost 3 h

3 On making exercises

The selection of exercises in the table above are the bare minimum; I advice you to do more. To assure you, I found the problems quite hard at times; probability never 'comes for free'; not for you, not for me, not for anybody. You can expect to spend between 30 minutes and 1 hour per problem; if you are serious.

Here is a list of good, and important, advice when making the exercises. (As a student I did not always do this, partly because I was not aware about how useful this advice is. Hopefully you are smart enough to avoid making the same mistakes as I did as a student.)

- Read an example in the book. Close the book, and try to redo the example. When I try, I often fail. Why is that? Simple, I did not really *think* about the example while just reading it, I *skimmed* it. Instead, *reading* requires pen and paper.
- Before trying to solve an exercise, read all parts of it, i.e., part a, b, etc. Ensure you *understand the problem.*
- Before actually solving an exercise, *make a plan on how to solve it*. A first step is to look for simple corner cases (set things to zero, make certain probabilities equal to one, and so on), make extra assumptions that simplify the problem, and solve the problem under these simplifying (stronger) assumptions. Then drop an assumption, and try to generalize to a pattern or some property you expect to hold. You'll be astonished to see how many problems you can actually solve by following this strategy. And even if you cannot solve it with this approach, the corner cases help to check throughout whether you're still working in the right direction. Also, reduce the problem to simpler cases you do understand. Try to solve the simpler problem first, and then generalize.
- Carry out your plan. In my hints, you should notice that I often do not directly aim to solve just the exercise. Instead, I 'play', I develop intuition, I try different ideas, I discover new things. And, I *relax*, even if I cannot directly find the answer.
- Look back right after solving the problem, and try to find a general pattern you used to solve the problem. Can you use this for other problems too?
- Look back again at the problem some time later. In other words, do not solve a problem just once, but also a few weeks later again. This is often very revealing.
- Work every day a reasonable amount of time. This is much more effective than working 10 h on one day, and not at all the next. The concept is often called 'Kaizen', try to improve every day a little bit. Over the course of time, you'll be amazed how much you can achieve.
- When I am stuck, this piece of advice of Jim Rohn (an author on personal development) helps: 'Don't wish it was easier, wish you were better.'

4 Assignments

There are 6 assignments, one for each lecture week, except for the first week. The guidelines are in `assignments.pdf`.

For each week we will make new groups of two students, hence every week you have to work together with another student. The reason behind this is to help all of you expand your network, which we find very important in the current situation. We realize this is perhaps not what all of you like, but this seems the best we can do to really help you get started and work with other students. You should also know that in professional life you will often have to work in different teams and with many different people. You'd better learn such skills early in your carreer.

The due dates are simple. In lecture week n, $2 \leq n \leq 7$, you turn in assignment $n-1$ before Friday 24h.

Ruben: 6 of 7 assignments?

5 Questions

Post your questions on the discussion board. We or fellow students will answer them during the lectures or in the discussion board.

6 Exam

At the exam we will pay attention to details and computational errors. Why? Because *You should learn to check*. One reason for this is that checks require the application of many different methods and strategies to solve probability problems. Hence, you will look at the same problem from different angles, so that you learn a lot. A second reason is that not checking thoroughly is, simply put, unacceptable. To see why, consider this example: you bring your car to a mechanic to have the tires changed. The mechanic is too lazy to check whether the bolts are tight. As a result, you get an accident, and when you wake up in hospital, your left arm has to be amputated. The anesthesiologist does not see the need to check the type of anesthetic nor the dose you need, so you kidneys are permanently damaged. The surgeon prefers to take a few beers before the operation starts, rather than checking what body part to amputate, so s/he removes your right leg instead of your left arm. The nurses are busy with their phones during the operation, because they find check work sooo boring. . . Other example, the programs by which your pension is computed over the years is extremely buggy, because the programmer did not like writing tests for the code. As a result, your lose 500 000 Euro on your final pension. I guess you get the point by now. As all people, *you* find it *unacceptable* when the mechanic, surgeon, and so on, don't check their work. Well, the same principle applies to you. Not checking is unacceptable, for you, for me, for anybody

7 Work load

The estimated work load is as in the table.

Activity	Load	Hours
Lectures by Blitzstein	14 × 1.5	21
Lectures by us	14 × 1	14
Study book	14 × 2	28
Assignments book	6 × 2	28
Exercises	14 × 3	42
Exam		3
Total		136

8 Grading

For each assignment you have to turn in parts 1, 2, and 3, i.e. the parts that are below and at exam level and the coding skills. You are allowed to skip part 4, the challenges. Each assignment will be graded as a 1, 6, 8, or 10. If you skip the challenges, the highest grade you can get is an 8. For a 10 you have to do all parts of the assignment (and have the right answers). If you don't turn in an assignment, the grade will default to 1.

Let $a = \sum_{i=1}^{6} a_i/6$ where a_i is the grade of your ith assignment. Let e be your grade for the exam or the resit. Then we compute your final grade g with the code:

```
def compute_grade(a, e):
    if e < 5:
        g = e
    elif a >= 6:
        g = max(0.8 * e + 0.2 * a, e)
    else:
        g = 0.8 * e + 0.2 * a

    return int(g + 0.5) # rounding
```

It is intentional that if you do a lousy job on the assignments, your final grade g is most surely lower than your exam grade e.

9 Contact info

- dr. N.D van Foreest, Duis 666, 050-363 51 78, n.d.van.foreest@rug.nl
- E.R. van Beesten
- Teaching assistent

10 Interesting other literature

There are a number of books that you might like too. (From experience I can tell that reading different types of explanation can be very helpfulp.)

1. R.B. Ash: Basic probability theory, free online
2. C.M. Grinsted and J. Laurie Snell: Introduction to probablity, also free online
3. F.M. Dekking, et al.: A Modern Introduction to Probability and Statistics, Understanding Why and How.

After the course you might be interested in the following two books that I liked a lot.

1. D.V. Lindley, Understanding Uncertainty. This book explains why probability theory is the way it is. There are three rules that any coherent system of probabilities has to satisfy.

   (a) For any event E, $\mathsf{P}[E] \in [0,1]$;

   (b) $\mathsf{P}[E \text{ or } F] = \mathsf{P}[E] + \mathsf{P}[F] - \mathsf{P}[EF]$;

   (c) $\mathsf{P}[EF] = \mathsf{P}[F \mid E]\, \mathsf{P}[E]$.

   If you want to understand probability in terms of betting, any sensible strategy you can imagine should satify these rules, for otherwise people can use arbitrage (an essential idea in the financial theory and asset and option pricing) to consistently make money from you.

2. E.T. Jaynes, Probability Theory: The Logic of Science. It is hard at times, but very interesting. it discusses applications and ideas behind probability and statistics, and how to think about these topics as a sensible person (not just as a theoretician).
3. C. Bishop, Pattern Recognition and Machine Learning. This is a really nice book on data analysis and machine learning. After the course you can read the first two chapters. At the end of the master, you can read most of the book.