*Python for Probability, Statistics, and Machine Learning*. Unpingco, José. 2019. San Diego: Springer International Publishing.

By Gene Callahan and Yujia Zhang

CSS Terms:

D.3 Programming Languages

D.1 Programming Techniques;

G.3 Probability And Statistics

I.2 Artificial Intelligence

The aim of *Python for Probability, Statistics and Machine Learning* is to offer programmers a tutorial on how to use python libraries like Numpy, Matplotlib, Pandas, Scipy, and Sympy to perform probability evaluations and statistical analysis, as the foundations for studying machine learning. It then gives a summary of various machine learning strategies, and introduces some further libraries, such as TensorFlow, to support that work.

The book is divided into four parts: it starts with a brief introduction to some of the main mathematical/scientific Python libraries, the iPython shell, Jupyter notebooks, and various IDEs. Next, the author discusses random variables, various distributions, and sampling methods in python. Then, it includes some python statistical modules and gives a tutorial on how to test hypotheses, evaluate confidence levels, and perform linear regressions in python. Finally, it gives an overview of machine learning concepts like decision trees, neural networks, dimensionality reduction through Primary Component Analysis, deep learning, the general steps in building a machine learning model with simple neural networks, and ways to train and test models.

The book uses Python 3.6 for its examples. The author expects the reader to understand simple loops, lists, matrix operations, Python math operations, input/output operations, and the module import system. Since Python 3 has good documentation online, it is not very hard to find explanations for any of these features that one does not know. The book introduces the use of a more advanced python libraries with small blocks of codes, brief comments and sample output. Thus, the book is aimed primarily at intermediate or advanced python programmers, although a beginner could struggle through it with some effort.

In terms of mathematics, since the author is trying to give a brief background on every probability and statistic concepts appear, the book is dense with math formulas. As this is not an introduction to probability and statistics in general, there is limited space for giving examples of datasets and problem scenarios, and therefore the mathematics may not be easy to understand and apply if the reader is not very familiar with intermediate probability and statistics, and with linear algebra.

However, besides a lot of complicated formulas given, each variable involved within the formula is clearly defined. And the alphabetical notion in the formula in the math formula is used to name the variable, as a result, increase the readability of the example codes. After a result is printed, the author usually clarifies what the number means, which makes sense in the context. However, there is still some use of constants that can be improved. For example, in the illustration of the *Chi-square* distribution with a cumulative distribution function, after calculating the *z-score* according to the formula, the author writes:

1. statis.chi2(2).cdf(z)

Based on the previous example and information, we infer that by writing

1-statis.chi2(2).cdf(z)

instead of

statis.chi2(2).cdf(z)

the code is calculating the tail bound of the Chi-square distribution, but that could be clarified with a comment. Moreover, since it passes 2 as a parameter to the Chi-square method, and, in the dataset provided, there are 3 categories, we might assume 2 represents the degrees of freedom. But all of this would be easier for the reader to understand if the author had written something like:

deg\_of\_freedom = num\_categories - 1  
1 - statis.chi2(deg\_of\_freedom).cdf(z) # get the tail bound

In the machine learning section of the book, the author makes good use of visuals to explain machine learning concepts. The mathematical methods mentioned in this part are often new, rather than deeply related to the previous chapters. Some general principles, such as linear regression may apply to some middle steps like calculating gradient descent, but different libraries and methods are used. It is good in the aspect for the reader to read the catalog and directly study the part needed, but if the reader is expecting to learn how to do a machine learning study integrated with various statistical methods, this book may not provide enough guidance. In general, the machine learning section is great for a beginner in machine learning and can be independently read from other parts.

This book needed better editing, to avoid sentences like, "The pip installer does not check for such conflicts checks only if the proposed package already has its dependencies installed and will install them if not or remove existing incompatible modules" (3) or "Visual Studio Community Edition... usually has enough to compile many C-codes" (3). The author recommends dividing by "the fractional powers of 1/2" (14) when he means "the powers of the fraction 1/2." Later he says an increase in dimensionality would necessitate "1000 more data points" instead of "1000 times more data points" (218). Of course these are just careless errors -- surely the author meant to write the latter version in both of the latter cases -- but such errors can seriously hamper a novice trying to master a new subject. Nevertheless, this work is a generally sound and comprehensive overview of the areas it covers, and we recommend it to the Python programmer interested in growing in these areas, or the expert in these areas interested in learning how to deal with them in Python.