Journal of Statistical Software

August 2015, Volume 66, Book Review 2. http://www.jstatsoft.org/

Reviewer: Samuel E. Buttrey
Naval Postgraduate School

Data Mining Algorithms Explained Using R

Paweł Cichosz
John Wiley & Sons, Chichester, 2015.
ISBN 978-1-118-33258-0. 683 pp. USD 64.99.
http://www.wiley.com/WileyCDA/WileyTitle/productCd-111833258X.html

---

Introduction

Data Mining – that intersection of statistics, computer science, and machine learning – is increasingly recognized as a discipline in its own right. Still, its statistical linear is undeniable. Paweł Cichosz's new book helps to keep data mining and statistics close, in part, by describing some of the important pieces of data mining using code in R, a tool with which many statisticians will be very familiar. The book, he says, is not intended for statisticians as such but "for a mixed audience consisting of students of computer science and related fields, researchers...in any area where data analysis capabilities are used [and] analysts...working with data and creating or using predictive models."

The essential idea of the book is to describe the basic data mining algorithms and their components, and this is done by presenting first the building blocks, and then their combinations, in the form of R code. The author notes right away that the book "does not teach [R] nor requires (sic) the readers to learn it. This is because the example code can be run and the results can looked up (sic) with barely any knowledge of R." Cichosz creates even low-level functions (like, for example, code for computing means and medians) to show how they might have been coded in R. These low-level functions are sometimes re-used, but more often he turns to the more efficient but less readable built-in versions later in the book. As another example, he sets up an entire classification tree implementation, complete with pruning, in code that is laid out line by line – then, in later examples, reverts to use of the built-in **rpart** package.

The book breaks its subject into five parts, each represented by between two and six chapters. The opening section, "Preliminaries," describes the set of data mining problems being addressed, establishes the voacabulary, and covers basic statistical ideas like the distribution of an attribute and the relationships between pairs of attributes. The vocabulary, it should be noted, is very much in the data-mining style, using "attributes" and "instances" rather than "variables" and "observations". The author also uses "m-estimation" to refer to something

totally different from the statistician's usual idea of a robust estimator. Still the vocabulary is not at all an obstacle to understanding the content.

The next three parts cover the three basic problems of data mining: classification, regression, and clustering. Each of these is represented by several chapters describing common approaches – in the case of clustering, a chapter is devoted to the idea of distance or dissimilarity measurements – followed by a welcome chapter on evaluation. A fifth part goes over model improvement procedures. These include ensemble methods, the support vector machine and kernel methods, chapters on transformation and discretization, and a chapter on variable selection. One of the strengths of the book is that, as it says, "on many occasions – whenever it is justified by practial utility – it discusses issues that tend to be overlooked or taken lightly..." This is a fair statement: the chapters on distances and evaluation, on discretization and transformation, and on variable selection (as well as an emphasis on handling weighted observations) serve to put focus on a number of issues that are often, if not overlooked, at least treated in an *ad hoc* manner. In addition the description of the uses of training, test and validation sets, and of evaluation through bootstrap and cross-validation, are good and useful.

The book's final chapter is devoted to three extended case studies, using data that users need to download from the UC Irvine repository. Unfortunately, each of these examples suffers from at least minor problems. The code for the first of these contains several errors (two references to a data set by the wrong name and an outright typo); these are easy enough to fix when they are spotted, but they cause consternation. The second requires that the user prepare a separate file with column names (since these are not provided in a convenient form at the repository). Given that the book's web page provides data for the rest of the book, and 21 R packages, the omission of the data for the case studies is hard to understand. Moreover, the second case study also has an outright typo in the code, in the form of an extra parenthesis. The third example produces a line of code that fails with the sort of error message that does not yield much information about the cause. It can be traced to what I characterize as a bug in the **rpart** package's `prune.rpart()` function – but in any case this sort of error should not be permitted to arise in these sorts of examples.

Linguistic style and typography

The book has perhaps more typographical errors than one would normally hope, as observed both in the second paragraph's quotation and in the case studies, but generally they serve to annoy rather than confuse. (For example, "loose" appears in place of "lose" twice, and "burglery" appears four times.) There is at least one error inside some pseudo-code. The book's text is written in a professorial style that seems precise but perhaps formal. A lot of concepts are described well in words and then equally accurately in symbolic notation which, while correct, is bestrewn with sub- and superscripts in a way that I suspect will be intimidating to the substantial number of practitioners unaccustomed to that style. This, I think, is why the inclusion of R code is such a good idea: by laying out the steps by which the algorithms can be implemented, the reader has the opportunity to gain an understanding of the pieces, the mechanisms by which they might be put together, and some of the practical considerations that will apply when analyzing real data. In the next section I discuss some of the issues surrounding this approach of including R code as a substantial part of the book.

R code

The R code could, in my opinion, have been this book's strongest point, elevating it from a strong exposition of important data mining tools to a learning tool that brings data miners to R and statisticians to data mining. Unfortunately, the code is also the book's greatest weakness. The difficulty starts with the installation of the packages, available at the book's website, which have a complicated dependency structure. (Advice to readers: install all of the external packages first, then keep looping over the book's packages, and, although the book doesn't say to, be sure to pass the `type = "source"` argument to `install.packages()`.)

The code is very far from self-explanatory. I expect it might be incomprehensible to a novice, as it jumps right in to formula notation, lists, functions like `mapply()`, and other complicated concepts. Indeed it often uses R concepts that even an intermediate user might not be aware of, without any discussion or explanation. For example, in the section on naive Bayes (`"nbc"`), one function includes these lines (with indentation modified here).

```
`class<-`(list(prior=cc/sum(cc),
                    cond=sapply (avc, function (avc1)
                            t(apply(avc1, 1, "/", colSums (avc1))))),
              "nbc")
```

To make things worse this segment happens to extend over a page break. With a little thought the experienced R user can detect that `avc` is a list of tables, and that the interior `apply()` computes, for each table, the ratio of each entry to its column total (though the `apply()` operation works on the rows! This is a consequence of R's recycling mechanism, and its column-wise matrix storage); that the result needs to be transposed because of the way `apply()` works; that the result is a list of $n \times 2$ tables; and that that list should be incorporated into a bigger list containing the prior probabilities, which is given class "`nbc`". How many R users are aware that the `` `class<-` `` function, when enclosed in backticks, can be called directly and returns its argument with the new class in place? I imagine this function is not widely used. If the intent were to make the code readable, I would have used something like

```
avc.new <- avc # make copy for holding conditional probs.
for (i in 1:length (avc.new)) {
    cs <- colSums (avc.new[[i]]) # get column sums and divide by them
    avc.new[[i]] <- avc[[i]] / rep (cs, nrow (avc.new[[i]]))
}
outlist <- list (prior = cc/sum(cc), cond = avc.new) # create list
class (outlist) <- "nbc" # add class
return (outlist) # and return
```

These complications and other constructs uncommon in regular R code (like, for example, wide use of the global assignment operator, `<<-`), appear fairly frequently. Now, to repeat, Cichosz does not focus on readability. He acknowledges that "[s]ince the primary role of the code is didactic and illustrative, it is written without any care for efficiency and error handling, and it may not always demonstrate good R programming style." Fair enough; I certainly agree that omitting the efficiency and error handling makes for clearer code. But if the goal of including the code is to allow it be understandable (otherwise, why not use

pre-built code in the first place?), it should be able to be understood. For example, it should have spacing, so that we see `rpart (x ~ .)` instead of `rpart(x~.)`.

In a similar way the author notes that "the documentation is particularly lacking...but this is forgivable given the fact that they (sic) are not distributed as standalone software tools." This point is well taken when it comes to help pages (although I remain persuaded that help pages would be usefull). It is less convincing when it comes to inserting comments into the code so that the reader can clearly tell what each line is supposed to be doing. Code should have comments – or at the very least, detailed explanations in the text. The lack of comments can make even readers experienced with both R and the method stop and wonder about the details of the approach – exactly the opposite result from what was intended, which was that the code would clarify the method. To be clear, these comments come from the perspective of a statistician with a great deal of R experience – but I venture to suggest that beginners would agree.

Conclusion

Paweł Cichosz's new book is a useful bridge between data mining and its implementation in R. A reader who ignored the code would probably learn quite a lot about the building blocks and algorithms presented; one who looked closely might learn a lot more. I am particularly gratified by the ample consideration of evaluating and comparing models. The major concern is the R code, which is included so as to clarify the steps of the algorithms. Instead, the code, while usually error-free, is hard enough to read and understand that it can form an impediment to understanding. I would love to see a second edition in which the code is made clearer and easier to read. Such a book would be a valuable resource to students new to data mining and those seeking to unify the ideas of regression, classification and clustering, and to be equipped with a set of tools to create and interpret data-mining models. It would also be a useful addition to the bookshelves of researchers and analysts who need to build and use predictive models. I look forward to seeing that book.

Reviewer:

Samuel E. Buttrey
Naval Postgraduate School
Department of Operations Research
Monterey, California 93943, United States of America
E-mail: buttrey@nps.edu

Journal of Statistical Software http://www.jstatsoft.org/
published by the American Statistical Association http://www.amstat.org/
Volume 66, Book Review 2 *Published:* 2015-08-06
August 2015