IST687 – Word Association - Homework

Chapter 17 of the textbook ("Hi Ho, Hi Ho – Data Mining We Go") introduces association rules mining – a very flexible and easy to understand form of unsupervised machine learning. Association rules mining is also sometimes called market basket analysis, and indeed the chapter works an example that focuses on groceries. Yet association rules mining can be applied to a variety of types of data, basically any data set where you have a list of "containers" and each container has a list of stuff inside it. Association rules mining looks for commonalities across these containers – what are the combinations of items that frequently occur together.

If you think about it for a minute, you might see that this idea applies to documents (e.g., emails or web pages) and the words that appear in them. Each document can be thought of as a container of words, and within each container certain combinations of words may appear together. In a previous chapter, we explored this idea by creating a "term-document" matrix. In this exercise, we are going to apply association rules mining to a term-document matrix.

You can create and/or find all of the code you need to accomplish these steps:

1. There is a nice, manageable term-document matrix that Yanchang Zhao has created based on a set of tweets about data mining that he extracted: http://www.rdatamining.com/data/termDocMatrix.rdata

   If for some reason hat link doesn't work, you should visit this page and download the file entitled "termDocMatrix.rdata" onto your computer:

   http://www.rdatamining.com/data

   As you have guessed from the file extension, this is a dataset that is already prepared for opening in R. Run R-Studio on your laptop and use the Open File command to load this file. After answering the confirmation message affirmatively, you will find that a data object called "termDocMatrix" appears in your environment window. Inspect this data object.

2. For association rules mining (and more specifically the apriori() command) to work properly, you want items in containers/baskets to be your columns and the rows to be your containers. You will find that this data object is set up the opposite way: It has the terms as rows and the documents as columns. You will need to transpose the data set, and fortunately R has a command to do that very easily. Do some research to find that command and how to use it. Then transpose your data set and place it in a new data object.

3. Next, apply all of the techniques you learning from chapter 17. This means that you will have to load the arules package, run apriori(), set the parameters correctly, inspect the results, visualize the results using the arulesViz package, and make sense

out of what you find. You should set your parameters so you generate at least 20 rules.

4. At the end of your code file for this exercise, write a few sentences interpreting the results of this analysis and describing how this technique might be valuable in making sense out of large sets of documents (e.g., emails).

Learning Goals for this activity:

A. Consider how a simple data mining technique can be applied to a variety of kinds of source data.
B. Provide practice in conditioning data to prepare for analysis.
C. Develop skill in the setup, execution, and interpretation of association rules mining.
D. Increase familiarity with bringing external data sets into R.
E. Increase familiarity with sources of advice and ideas on R source code.

Essential Guide for All IST687 Activities (appears at the end of all activity guides)

1. All IST687 activities work on what some people call a "constructivist learning" model. By developing a product on your own, testing it to find flaws, improving it, and comparing your solution to the solutions of other people, you can obtain a deeper understanding of a problem, the tools that might solve that problem, and a range of solutions that those tools may facilitate. The constructivist model only works to the extent that the student/learner has the drive to explore a problem, be frustrated, fail, try again, possibly fail again, and finally push through to a satisfactory level of understanding.
2. Each IST687 activity builds on skills and knowledge developed in the previous activities, so your success across the span of the course depends at each stage on your investment in earlier stages. Take the time to experiment, play, try new things, practice, improve, and learn as much as possible. These investments will pay off later.
3. Using the expertise of others, the Internet, and other sources of information is not only acceptable - it is expected. You must ***always, always, always*** give credit to your sources. For example, if you find a chunk of code from r-bloggers.com that helps you with developing a solution, by all means borrow that chunk of code, but make sure to use a comment in your code to document the source of the borrowed code chunk. The discussion boards in the learning management system have been setup to encourage appropriate sharing of knowledge and wisdom among peers. Feel free to ask a question or pose a solution on these boards.
4. Building on the previous point, when submitting code as your solution to the activity, the comments matter at least as much, if not more than the code itself. A good rule of thumb is that every line of code should have a comment, and every meaningful block of code should be preceded by a comment block that is just about as long as the code itself. As noted above, you can use comments to give proper credit to your sources and you can use comments to identify your submission as your own.

5. Sometimes the building process reveals unexpected results that are themselves very informative in learning. When you completed the exercise above, what did you find that was unexpected? What did you do about trying to understand what had happened? Did you do further exploration? What did that further exploration reveal?
6. Here’s a new bonus item: Frustration is actually a powerful source of learning, if you can push through to the “other side” (i.e., you can ultimately work around the source of the frustration). The challenge layer in this exercise is an important tool in this regard: Combining the skills from previous lessons with new skills and applying them to a difficult and novel problem will almost inevitably lead to glitches in the process of constructing your artifact (in this case the R code to solve the challenge). Embrace that frustration and see if you can get through it to deeper learning.