Team 5

Technology Review

Introduction

My project title is: Joint Analysis of Hotel Review & Historical Local Economy Metrics for Causal Topics. As detailed in my project proposal, the tool I plan to build would involve extracting topics from hotel review data and understanding those topics in the context of economic indicators for local regions.

The ingestion of hotel review text data to find meaningful topics would require the most effort and involve the greatest application from this course. It would need to provide topic modeling. At first I considered three packages that would help me: mscstexta4r, topicmodels, and syuzhet. These three packages are each R packages. It turns out that the first package, mscstexa4r, was a package that enabled use of available Microsoft technology through Azure. I had difficulty in getting that package to work consistently. Also, after some additional thought, I realized that I did not want to perform sentiment analysis as part of my project so I ruled out using syuzhet. Fortunately I found another R package useful for topic modeling. Therefore, this report will cover the topicmodels and stm R packages in the Topic Modeling Packages Review section.

In addition, the project will involve employing the Granger Causality Test. I will review briefly the lmtest and MSBVAR R packages and what they have to offer for my project in the Granger Causality Test Packages Review section.

Topic Modeling Packages Review

Package: topicmodels

The topicmodels package builds on the package tm. The tm package provides relevant methods to read in text data within R to build what is called a corpus.[1] The tm package has the capability to transform the text data in useful ways like removing extra whites spaces, converting to all lower case, removing stopwords and stemming. The tm package can express the corpus in what is called a DocumentTermMatrix. The DocumentTermMatrix expresses the underlying text in matrix form whereby each row is identified by a document identifier and each column represents the term frequency for that document for each unique term. As can be imagined, for a large set of documents, the matrix form of the text collection can be quite large and result in a sparse matrix. Fortunately, the tm package is capable of displaying the DocumentTermMatrix in a dense representation.

It is this DocumentTermMatrix that can be directly evaluated by the topicsmodels package. The topicsmodel R package enables evaluation of the corpus by means of the Latent Dirichlet Allocation (LDA) model and another model called the Correlated Topics Model (CTM).[2] The main difference between the two models is that LDA assumes no correlation between topics extracted from a corpus whereas with CTM, correlations between topics are allowed. The topicsmodels R package allows for evaluating a corpus as in the DocumentTermMatrix format that the tm R package creates. The topicsmodels R package can also perform some of the useful transformations that the tm R package can conduct. For example, the topicsmodel R package has built-in capabilities to remove stop words, stemming, and removing punctuation. The challenge with the topicmodels R package is that it does not provide tools to guide the selection of the appropriate number of topics. The number of topics is a parameter that is defined by the user. Depending on the number of topics pre-defined by the user, the set of topic models that are generated can differ widely.

Package: stm

The stm R package builds on the work that underlies the development of LDA and CTM.[3] The stm R package is named after the Structural Topic Model. The Structural Topic Model (STM) is distinct from LDA and CTM is that STM allows users to incorporate arbitrary metadata into generating the topic models. The stm R package calls functions from the tm R package to perform text transformations like removing stop words, stemming and dropping punctuation.

The feature of the stm R package that I found most useful was the ability to provide guidance about choosing the appropriate number of topics. The stm R package has a function called searchK. The searchK function takes in as input the text corpus and an array of k values where k is the number of pre-defined topics. The searchK function returns a table like the one shown below.

```
> searchK_results
$results
   K   exclus    semcoh   heldout residual    bound   lbound em.its
1  5 9.751650 -60.45724 -6.793233 6.418723 -1657797 -1657792     17
2 10 9.357365 -68.94170 -6.686866 4.457200 -1624263 -1624247    451
3 20 9.533295 -75.76209 -6.647662 3.541775 -1602934 -1602891     97
4 25 9.400818 -75.39965 -6.670972 3.079778 -1594367 -1594309     66
```

In this example, I provided an array of 5, 10, 20 and 25 as the number of topics (k) I wanted to evaluate for the same text corpus. The example table indicates that for k = 5, the number of iterations in the expectationmaximization steps (the last column, em.its) was only seven whereas for k = 10, the number of iterations shoots up to 451. The residual column provides the residual check outcome. This value represents the overdispersion of the variance of the multinomial. In practicing with the stm R package, I look at the residual and em.its columns to provide guidance on the number of topics I should pre-select. The topicsmodels R package does not seem to have similar functionality. I doubt I can use the results of the searchK function from the strm R package to inform my choice of number of topics when I run the topicmodels R package.

Granger Causality Test Packages Review

Package: lmtest

The lmtest R package contains a function called grangertest. The function takes two arrays distinct univariate arrays as inputs and provides a measure of how well one variable can be inferred to cause the other. The assumption is that the two arrays represent two variables of the same time interval increment. Let's say there are two variables, x and y. In essence if there is greater accuracy in predicting y from the past of x and y than from the past of y alone, x can be inferred to cause y. The grangertest function is actually an implementation of the Wald test. The grangertest function returns the critical anova test results which contains the residual degrees of freedom, the difference in degrees of freedom, Wald statistic and corresponding p value.

Package: MSBVAR

The MSBVAR R package contains a function called granger.test. The granger.test function takes an input a matrix that contains the two variables as well as the lag term. The granger.test function returns a matrix of a two-column matrix. The first column contains the F-statistic values. The second column contains the p-values for the F-tests.

[1] Feinerer, Ingo, "Introduction to the tm Package Text Mining in R", 3/2/2017 (link here)
[2] Grün, Bettinna and Hornik, Kurt, "topicsmodels: An R Package for Fitting Topic Models" (link here)
[3] Roberts, Margaret; Stewart, Brandon; Tingley, Dustin, "stm: R Package for Structural Topic Models", *Journal of Statistical Software* (link here)