R

Last Edited: 11/25/2010

1. Module name: R

2. Scope:

a. This module covers use of the R language for performing the statistical analysis needed for several information retrieval (IR) techniques. The R language is extremely extensive and a powerful tool. R will not be covered completely in this module. Rather, we hope to introduce some of the more common tools used in IR, such as matrix and vector manipulations.

3. Learning objectives:

The user should:

a. Be capable of using R to perform various operations on matrices and vectors, including:

- i. standard arithmetic operations (addition, subtraction, division, multiplication)
- ii. calculate the determinant of a matrix
- iii. perform a matrix transpose
- iv. calculate the Pearson and Spearman correlation coefficients between two vectors

b. Be able to use the command line interface to R to perform eigenvalue/eigenvector and singular value decomposition on term-document matrices.

c. Be able to do a low ranking approximation/latent semantic indexing of a term-document matrix.

4. 5S characteristics of the module:

a. **Streams**: R provides a wide variety of statistical methods for easily processing large collections of data. Additionally, R can streamline the visualization process by applying graphical techniques directly to imported data.

b. **Structures**: The user can interact directly with R through the "R environment" on the command line. Single commands can be issued in the "R environment" in a similar way to using a linux BASH shell. Additionally, standalone R packages may be developed for inserting powerful R tools into code written in other programming languages.

c. **Spaces**: R is typically used to manipulate large collections of data that can be stored in text documents which contain matrices or vectors of floating point numbers.

d. **Scenarios**: IR systems may utilize the libraries written in R for performing any necessary mathematical manipulations.

e. **Society**: With the wealth of information available on the world wide web, R can easily be self-taught. Typically issuing a search in Google of "tutorial R <method>", where <method> describes any mathematical operation, will direct the user to useful guides for utilizing R.

5. Level of effort required (in-class and out-of-class time required for students):

a. The module should require 4-6 hours to complete.

i. Out-of-Class: Each user should expect to spend around 3-4 hours studying the module individually, including the completion of each exercise. The language syntax may be the biggest hurdle for most users. Careful investigation of the material in this module should help the users understand the concepts behind R. It is imperative that users complete the exercises individually and before a group meeting. For further information regarding the exercises and helpful links for this module, the user can refer to Sections 10, 11, and 14.

ii. In-Class: Each user should spend 1-2 hours discussing the module with their teammates. Any terminology or concepts that are unclear can be clarified during this discussion. Additionally, users should take this opportunity to discuss their individual results to the exercises in Section 11.

6. Relationships with other modules (flow between modules):

a. Weka is a collection of machine learning tools implemented in the Java programming language. These tools include vector space classification, which relates to the R-Package **class package** that contains various functions for vector space classification (i.e., KNN).

b. Cluto is a collection of libraries for efficient data clustering and analysis. The R-Package **cluster package** and **clv package** can similarly be used for cluster analysis and cluster validation within the R framework.

7. Prerequisite knowledge/skills required:

a. The user should have some previous notion of common methods from linear algebra such as:

i. matrix decomposition

ii. evaluating eigenvalues and eigenvectors

iii. significance of eigenvalues and eigenvectors

b. The user should have already been introduced to basic IR concepts such as:

i. term-document matrix

ii. vector space model

c. The user should be familiar with using a command line interface in a terminal window.

8. Introductory remedial instruction:

Singular Value Decomposition (SVD) computes the term and document vector spaces by decomposing the term-document matrix into three other matrices — a term matrix, a singular values matrix, and document matrix. SVD is the main concept behind Latent Semantic Indexing (LSI), which is performed on the term-document matrix. LSI aims to represent semantic associations between terms contained in the documents in a low dimensional space.

LSI may be used to:

i. Overcome the size of term-document matrix

ii. Compute document similarity more efficiently (on a reduced space)

iii. Reflect the polysemy of terms

iv. Cluster documents by topics.

LSI has been readily applied to:

i. Automated document classification

ii. Text summarization

iii. Matching technical papers and grants with reviewers

iv. Spam filtering

9. Body of knowledge:

a. **Background of R**: R is derived from the S programming language, which was originally developed at Bell Laboratories by Rick Becker, John hambers, and Allan Wilks. The S language was first built to only call Fortran functions, and it was later rewritten to execute C/C++ and Java functions. This revised version of S was named "New S" language. R is currently distributed under the GNU general public license and is publicly available on Windows, Linux, and Mac OS platforms.

b. **Accessing R**:

i. For the purpose of this module, R has been installed on an IBM cloud instance running SUSE Linux. Three pieces of information are necessary to access the IBM cloud via SSH: hostname, IP address, and RSA key for the instance. Your instructor or the module authors can

provide you with the information needed to log into the cloud instance. If you are uncertain about how to SSH into the cloud instance using this information, please contact the instructor or authors for further instructions.

ii. For personal use: You can install R using the appropriate package management system (apt, zypper, rpm, etc.) for your Linux distribution. Windows user can download R from http://www.r-project.org/index.html

c. **Using R:** R is more than just an environment for performing statistical analysis, but it is commonly used for statistics systems. We will not cover the entire spectrum of functionality that R provides. We will focus on a few methods and their applications to various IR techniques.

R comes prepacked with numerous functions and datasets stored in packages. The standard R distribution provides 25 "standard" packages. However, hundreds of external packages are contributed by various authors. Each package may be written to fulfill the needs of a particular group of scientists, and these packages can become highly specialized. Below is a sample of some available R packages:

Package Name	**Description**
base	Base R functions
dataset	Base R datasets
graphics	R functions for base graphics
stats	R statistical functions
utils	R utility functions
matrix	Matrix package
class	Functions for classification
cluster	Functions for cluster analysis

Commands:

•To start R, simply execute *R* on the command line

>R

•To quit R, use *q()*

>q()

•To see installed packages, use the *library()* command

>library()

•To load a package, use the *library(class)* command with a parameter

>library(class)

•To start help

> help.start()

•To create a vector of 5 floating point numbers

> x <- c(10.4, 5.6, 3.1, 6.4, 21.7)

•To create a matrix

> x <- array(1:20, dim=c(4,5)) #Generate a 4 by 5 array filled with numbers from 1 to 20.

•To display an object, type the name of the object

>x

•To delete an object, use the *rm* command

>rm x

•To load a data matrix from file, use the *read.table()* command

>HousePrice <- read.table("houses.data")

•To execute an eigen decomposition, use the *eigen()* command

>y <- eigen(x)

•To display eivalues and eigenvectors

>y$val #displays eigenvalues
>y$vec #displays eigenvectors

•To do a Singular Value Decomposition

>svd(x)

•To display the singular values

>svd(x)$d

d. **R and Information Retrieval concepts:** R implements several concepts from information retrieval. The following table presents some of the implementations:

IR Concept	R package
Text preprocessing (stemming, tokenization) Term weighting, scoring	**tm package**: Constructs a term-document matrix, using one of the the following weighting functions **TF** (weightTf), **TF-IDF** (weightTfIdf).
vector space model for scoring	**clv package:** dot.product function returns a **cosine similarity** measure of two vectors.
vector space classification	**class package**: performs a **k-Nearest Neighbour Classification** on a dataset.
Hierarchical clustering	**Cluster package**: computes clusters (**agglomerative hierarchical**) on dataset.
Latent Semantic Indexing	**Base package**: performs **Singular Value Decomposition** on matrix

10. Resources:

a. A wealth of information is available at the official site for the R programming language http://www.r-project.org/. We outline a few of the most important resources on their site:

i. http://cran.r-project.org/doc/manuals/R-intro.html This lengthy introduction to R breaks the use of R down into 13 chapters, each dedicated to a specific element of the R language.

ii. http://cran.r-project.org/doc/manuals/R-data.html Any questions about importing and exporting data that is stored in different formats are likely answered here.

iii. http://cran.r-project.org/doc/manuals/R-lang.html Details about the definition of the R language and its implementation are available here.

iv. http://cran.r-project.org/doc/manuals/R-exts.html This site provides information on developing specialized R packages and/or extending the available functionality of R.

v. http://cran.r-project.org/faqs.html This site contains various frequently asked questions (from both experienced users and beginners) with regard to R.

vi. http://cran.r-project.org/web/packages/class/class.pdf The documentation of the "class" package.

1**1. Exercises / Learning activities:**

a. (20 minutes) Matrix Operations: Load the file "m1.data" (available on the IBM cloud instance) containing the first term-document matrix into an object called "m1". Execute an eigen decomposition of m1. Display the eigenvalues and the eigenvectors. Provide screenshots.

b. (45 minutes) Latent Semantic Indexing and Low Rank Approximation - Vector space classification:

i. Load the content of “m.data” into an object called “m”.

ii. compute the **Singular Value Decomposition** of that object using the “svd” command.

iii. How many eigenvalues were found?

iv. Zero out the three eigenvalues lowest values, and create a diagonal matrix out of the eigenvalue vector.

v. compute the low rank document matrix approximation.

vi. Perform **K-nearest neighbor** clustering on iris3 dataset. The KNN algorithm is provided in the “class package”. Using the class package and R documentation, describe the objects (*train*, *test*, and *cl*) and commands (*rbind, factor,* and *knn*) of the following set of instructions to compute K-nearest neighbor.

>train <- rbind(iris3[1:25,,1], iris3[1:25,,2], iris3[1:25,,3])

>test <- rbind(iris3[26:50,,1], iris3[26:50,,2], iris3[26:50,,3])

>cl <- factor(c(rep("s",25), rep("c",25), rep("v",25)))

>knn(train, test, cl, k = 3, prob=TRUE)

c. (45 minutes) Computing Pearson and Spearman correlations: The Pearson and Spearman correlation coefficients are methods for analyzing the similarity between two vectors. On the IBM cloud instance, a single matrix file has been provided named “m2.data”. This file contains 10 rows that correspond to 10 items and 5 columns, each corresponding to the value for the item given by its row.

i. Use the “cor” method to calculate the **Spearman correlation** between each pair of columns, and report those correlation values.

ii. Which pair of (non-equal) columns has the **highest Spearman correlation**? What can you infer about this pair of columns?

iii. Which pair of (non-equal) columns has the **lowest Spearman correlation**? What can you infer about this pair of columns?

iv. Use the “cor” cor method to calculate the **Pearson correlation** between each pair of columns, and report those correlation values.

v. Which pair of (non-equal) columns has the **highest Pearson correlation**? What can you infer about this pair of columns?

vi. Which pair of (non-equal) columns has the **lowest Pearson correlation**? What can you infer about this pair of columns?

vii. The correlation values returned for Pearson and Spearman should be slightly different for some pairs of columns. What is the fundamental difference between these two methods for computing correlation?

viii. How might these measures be used to analyze the results from two IR systems?

12. Evaluation of learning objective achievement :

a. Exercise 12-a is intended to expose the user to the eigenvector computation using R, as well as some basic input/output operations within R.

b. Exercise 12-b:

i. Exercise 12-b parts i-v walk the user through the process of making a low rank approximation of the document matrix using Singular Value Decomposition in R. After completion of this exercise, the user should be able to perform various matrix and vector manipulation techniques. Additionally, the user should understand latent semantic indexing and be able to perform LSI on other matrices.

ii. Exercise 12-b part vi illustrates to the user how to perform K-nearest neighbor classification using the "class package". The user will be required to read the documentation for this package (perhaps using the "help" command in R) as well as the text book in order to successfully complete this exercise.

c. Exercise 12-c will familiarize the user with two different methods in R for computing correlation between two vectors.

i. Parts i-vi of Exercise 12-c simply require the user to read a text file, manipulate the data, and report the results.

ii. Part vii-viii asks the user to provide some in depth analysis of the results from i-vi. The user should understand how the correlation between two vectors can be directly applied to IR, and they should provide a convincing argument of ways to incorporate these measures.

13. Glossary:

Eigenvector - A vector in which the direction of the vector remains unchanged while the amplitude changes after multiplication by a (non null) matrix (i.e., $Ax=kx$ where A is a matrix, x the eigenvector and k the eigenvalue).

Eigenvalue - The factor by which the amplitude of the eigenvector is changed.

Latent Semantic Indexing - A representation of a document matrix in a lower dimension (at most the rank of the original matrix) matrix using Singular Value Decomposition.

Low-rank approximations - A representation of a term-document matrix in a lower dimension (at most the rank of the original matrix) matrix using a Singular Value Decomposition.

Matrix rank - The maximal number of linearly independent rows (or columns) of a matrix.

Singular Value Decomposition - Factorization of a matrix into the product of three matrices, with one of these being a diagonal matrix containing the eigenvalues.

Term-document matrix - A matrix representing the terms and their frequency in each document of a collection.

14. Additional useful links:

a. http://rwiki.sciviews.org/doku.php The official R wiki, maintained by the R user community.

b. http://math.illinoisstate.edu/dhkim/rstuff/rtutor.html An R tutorial that focuses on some graphing and other visualization techniques.

c. http://www.stat.pitt.edu/stoffer/tsa2/R_time_series_quick_fix.htm A tutorial that focuses on working with time series data.

d. http://www.his.sunderland.ac.uk/~cs0her/Statistics/UsingLatticeGraphicsInR.htm This tutorial provides details on developing more advanced and sophisticated graphs using the trellis graphics capabilities in R.

15. Contributors:

a. Authors: Eric Fouh, Chris Poirel

b. Reviewers: Dr. Edward A. Fox