coursera COURSES ABOUT HUANGHAO

coursera
Web Intelligence and Big Data

Gautam Shroff
TCS Innovation Labs

Home
Video Lectures
Discussion Forums
Feedback Surveys
Quizzes
Homework Assignments
Programming Assignments
Course Outline
Course Logistics
References
Fall 2012 Course Schedule
About the Instructor

Join a Meetup
Course Wiki

Programming Assignments

Assignment C (Programming / Data Analysis) to be used for HW 7 (Week 8)

Sales volumes of hundreds of products in the first month after each is launched (M), as well as the total sales of each product in the first year after launch (S), are given in the data set HW7Data.csv. Each product is characterized by five numerical features: G,Q1,Q2,Q3,and Q4.

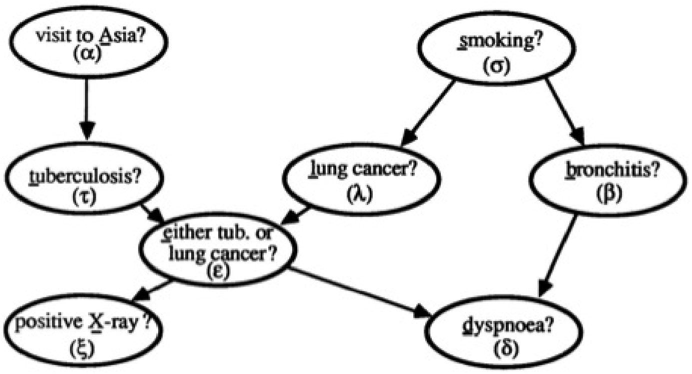
Determine the least-squares linear relationships (i.e. linear regressions) between:
(i) M and *each* of the features G,Q1,Q2,Q3,Q4 individually
(ii) M and *all* of the features G,Q1,Q2,Q3,Q4 together
(iii) S and each of the features G,Q1,Q2,Q3,Q4 individually
(iv) S and *all* of the features G,Q1,Q2,Q3,Q4 together
In each case determine the regression coefficients, i.e. the vector **f**, as well as the **r-squared** measure of 'goodness of fit'.

Note: You can choose to program your solution directly using any programming language or use a package such as R, Octave, Matlab or even Statplus (which works with Excel). (In the former case, be sure to use a matrix manipulation library that includes a linear system solver.)

Once you have completed tasks (i) .. (iv), answer some questions about the data in **Homework 7**.

Homework 6 (Programming Assignment B)

Write a program that encodes the Bayesian network given below. Your program should take as input evidence regarding the patients behavior or symptoms, and compute the a-posterior probabilities of the disease variables, i.e., tuberculosis, lung-cancer, or bronchitis. (You can also download this diagram from: here).



α:	$p(a) = .01$	ε:	$p(e \mid l, t) = 1$ $p(e \mid l, \bar{t}) = 1$ $p(e \mid \bar{l}, t) = 1$ $p(e \mid \bar{l}, \bar{t}) = 0$
τ:	$p(t \mid a) = .05$ $p(t \mid \bar{a}) = .01$		
σ:	$p(s) = .50$	ξ:	$p(x \mid e) = .98$ $p(x \mid \bar{e}) = .05$
λ:	$p(l \mid s) = .10$ $p(l \mid \bar{s}) = .01$	δ:	$p(d \mid e, b) = .90$ $p(d \mid e, \bar{b}) = .70$ $p(d \mid \bar{e}, b) = .80$ $p(d \mid \bar{e}, \bar{b}) = .10$
β:	$p(b \mid s) = .60$ $p(b \mid \bar{s}) = .30$		

You will need to run your program to answer the questions posed to you in Homework 6; as before, the exercise will be timed, so that manual computation is not an option!

Homework 3 (Programming Assignment A)

Download data files bundled as a .zip file from hw3data.zip

Each file in this archive contains entries that look like:

journals/cl/SantoNR90:::Michele Di Santo::Libero Nigro::Wilma Russo:::Programmer-Defined Control Abstractions in Modula-2.

that represent bibliographic information about publications, formatted as follows:

paper-id:::author1::author2::…. ::authorN:::title

Your task is to compute how many times every term occurs across titles, for *each* author.

For example, the author Alberto Pettorossi the following terms occur in titles with the indicated cumulative frequencies (across all his papers): program:3, transformation:2, transforming:2, using:2, programs:2, and logic:2.

Remember that an author might have written multiple papers, which might be listed in multiple files. Further notice that ‘terms’ must exclude common stop-words, such as prepositions etc. For the purpose of this assignment, the stop-words that need to be omitted are listed in the script stopwords.py. In addition, single letter words, such as "a" can be ignored; also hyphens can be ignored (i.e. deleted). Lastly, periods, commas, etc. need to be ignored; in other words, only alphabets and numbers can be part of a title term: Thus, “program” and “program.” should both be counted as the term ‘program’, and "map-reduce" should be taken as 'map reduce'. Note: You do *not* need to do stemming, i.e. "algorithm" and "algorithms" can be treated as separate terms.

The assignment is to write a parallel map-reduce program for the above task using octo.py, which is a lightweight map-reduce implementation written in Python available from http://code.google.com/p/octopy/.

Once you have computed the output, i.e. the terms-frequencies per author, go attempt Homework 3 where you will be asked questions that can be simply answered using your computed output, such as the top terms that occur for some particular author.

Note: There is no need to submit the code; I assume you will experiment using octo.py to learn how to program using map-reduce. Of course, you can always write a serial program for the task at hand, but then you won’t learn anything about map-reduce.

Lastly, please note that octo.py is a rather inefficient implementation of map-reduce. Some of you might want to delve into the code to figure out exactly why. At the same time, this inefficiency is likely to amplify any errors you make in formulating the map and reduce functions for the task at hand. So if your code starts taking too long, say more than an hour to run, there is probably something wrong.

Created Wed 19 Sep 2012 5:30:22 AM PDT
Last Modified Tue 23 Oct 2012 10:37:10 PM PDT