Academic career planning using bayesian network

Ray Shulang Lei
200253624
Department of Computer Science
University of Regina
Regina, Saskatchewan, S4S0A2, Canada

April 8, 2011

Abstract

Most of the time, students going into universities without anticipating enough time/energy. Furthermore, they might set unrealistic graduation plans, thus risking their financial/career/relationship well-beings. Universities provide program advisers(experts) to help students, but human resources are too valuable in most situations. On the other hand, face to face consulting does not have good accessibility to students, thus the reason to seek for online automated solutions.

1 Introduction

Undergraduate programs often has prerequisite charts. These charts can be converted to decision trees. However, whether a student will graduated is usually affected by many uncertain factors. For example, some courses are only being offered under specific conditions; Some classes have great difficulties such that they can not be taken simultaneously; Some classes have lab time requirements thus increases students work load, and so on. Using Bayesian network seems to be well-suited for these uncertain situations.

Using node.js, Dlib C++ Library to build an web application that answers student queries about carrer planning questions.

2 Course Prerequisite Inference

Here is a directed acrylic graph A to represent the course prerequisite relations.

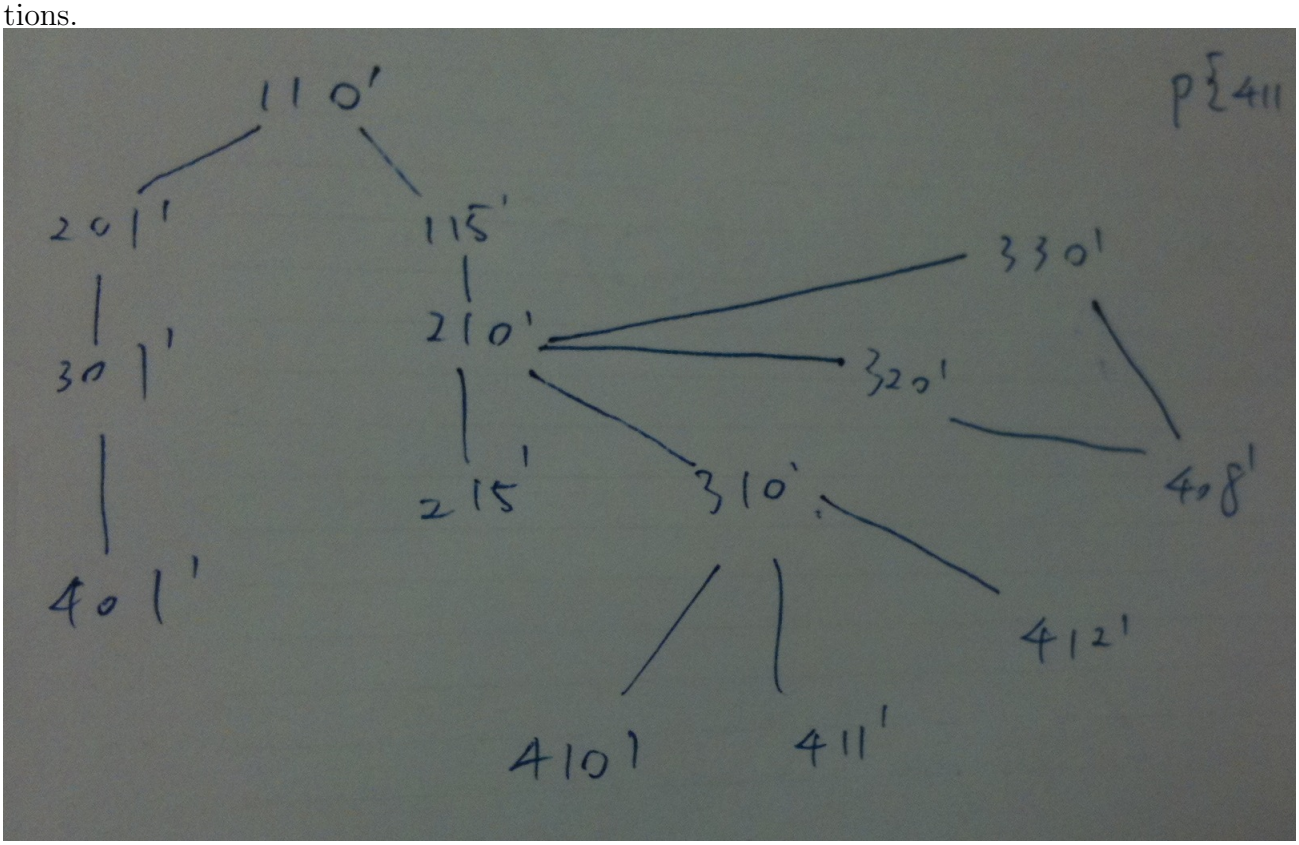

Figure 1.1

i.e Querying the probability of taking CS411 in the future, provided student have taken CS110:

1. Search the shortest path between $110' $ to $411'$ in A
2. Which is $110' -> 115' -> 210' -> 310' -> 411'$

3. The length of this path is 4, which means student needs at least 4 semesters to finish CS411, provided there is no class not bing offered during these semesters.
4. If there are classes in the path not being offered, we need to add extra semesters accordingly.
5. Run JTP on A, query $P\{CS411 = 1 \mid CS110 = 1\}$
6. Provide feedback to student, telling the least semesters he need to finish CS411 with a probability.

3 Knowledge Aquisition

I am working with Wendy Preikchat, who is the program coordinator at Computer Science Department. She has a lot of experience providing students consultations for their academic careers. There are also other people who helped with knowledge acquisitions: Ara Steininger, math department program coordinator; Shalini Mathias, Graduate Research Associate (curriculum design).

4 Install, Run instructions

Here are how to get the program running:

1. git clone gzmask@lancewave.com:/home/gzmask/node_bayes.git (get the source code, password is '121212)
2. go into node_bayes folder
3. run "sh server_start.sh" to compile the source code. you'll need G++, make, node.js to compile.
4. open browser, input the url: localhost:8080/index
5. check your passed course, submit, then the bayesian network will tell you the possibilities that you would be able to pass the future classes
6. executing "git pull" would update the source code to the newest version

And ongoing and running instance can be directly access at: http://142.3.31.224:8080/index Open in Browser directly can avoice the trouble of compiling the source code and setting up the server yourself.

5 Bayesian Network Implementation

I used DLIB (http://dlib.net/) for the Bayesian Network part. The source code is hosted using GIT Repository, thus providing full advantage of easy to download and version management. The address of the GIT Repository is: gzmask@lancewave.com:/home/gzmask/nodebayes.git with password of '121212'.

Here is a simple explaination of the repository structure:

1. README simple instruction on how to run the program
2. app.js node.js starting point
3. bayesnet.cpp bayesian network implementation
4. bayesnet.h header file
5. bayesianbind.cpp binding implementation between node.js and c++
6. build binary files of the c++ compilation results
7. dlib Dlib which provides data struture of Bayesian network and JTP
8. doc this file
9. examples
10. nodemodules node.js libraries
11. oldbayesnet.cpp junk file
12. package.json node.js package info metadata
13. public html web file serving folder
14. serverstart.sh shell script to execute the package
15. views ejs html templates with node.js embeded code

16. wscript c++ compilation linking information

6 User Interface

The implementation of the User Interface is using the Node.js, which is based on the google chrome V8 engine. It provides fast javascript performance with good c++ interfacing abilities. At this point, the UI is still very simple.

7 Results

The Node.js along side with C++ provides good performance and responsive web server. A copy of source code is attached with this document.