Are matrix-based or node-linked graphs more readable when representing causal relationships for social and health data?

Kristina Lazarova

School of Computing Science
Sir Alwyn Williams Building
University of Glasgow
G12 8QQ

Level 4 Project — March, 2017

Abstract

We show how to produce a level 4 project report using latex and pdflatex using the style file l4proj.cls

Contents

1 Literature Review **1**

2 Introduction **2**

3 Implementation details **3**

3.1 Software tools and technologies 3

3.2 Challenges 3

3.3 Software reliability testing 3

4 Evaluation **4**

4.1 Design 4

4.2 Participants 4

4.3 Procedure 4

4.4 Results 4

4.5 Discussion 4

5 **5**

5.1 First Section in Chapter 5

5.1.1 A subsection 5

6 The Fox and Dog **6**

6.1 The Fox Jumps Over 6

6.2 The Lazy Dog 7

Appendices **8**

A Running the Programs **9**

B Generating Random Graphs **10**

Chapter 1

Literature Review

This is the first chapter where I will introduce data visualisation and and explain how I came up with the idea of this research

Chapter 2

Introduction

Introduction to the specific area of my research

Chapter 3

Implementation details

3.1 Software tools and technologies

A web application framework

3.2 Challenges

```
Spring idea failed
changed to Node.js
Angular compatibility with Node.js
```

In the beginning of this project the Java framework Spring was going to be used in the implementation as it is among the most widely used frameworks in industry [5]. This decision was supported by extensive previous experience with Java from developer's point of view and the applicability of the skills to be acquired. However, one of the reasons why Spring is used in industry is because of the large and complex systems that exist there. The Spring framework works on a very high level of abstraction where you can easily write configuration files to add dependencies from different project. Therefore, it is considered rather unfriendly for small independent projects and developers with limited Spring experience. The reasoning behind this conclusion was provoked after a couple of unsuccessful attempts to set relative paths to different CSS and JavaScript files. The issue was found to be in the web application configuration file. This is how the very simple task of reading a css file turned to be a long tedious debugging process after which the realisation that Spring is unnecessary abstract and complex for this project occurred.

3.3 Software reliability testing

Chapter 4

Evaluation

4.1 Design

4.2 Participants

4.3 Procedure

4.4 Results

4.5 Discussion

Chapter 5

5.1 First Section in Chapter

The quick brown fox jumped over the lazy dog. The quick brown fox jumped over the lazy dog. The quick brown fox jumped over the lazy dog. The quick brown fox jumped over the lazy dog. The quick brown fox jumped over the lazy dog [1]. The quick brown fox jumped over the lazy dog. The quick brown fox jumped over the lazy dog. The quick brown fox jumped over the lazy dog. The quick brown fox jumped over the lazy dog.

5.1.1 A subsection

The quick brown fox jumped over the lazy dog. The quick brown fox jumped over the lazy dog. The quick brown fox jumped over the lazy dog. The quick brown fox jumped over the lazy dog.

The quick brown fox jumped over the lazy dog. The quick brown fox jumped over the lazy dog. The quick brown fox jumped over the lazy dog. The quick brown fox [3] jumped over the lazy dog. The quick brown fox jumped over the lazy dog.

Chapter 6

The Fox and Dog

The quick brown fox jumped over the lazy dog. The quick brown fox jumped over the lazy dog. The quick brown fox jumped over the lazy dog. The quick brown fox jumped over the lazy dog. The quick brown fox jumped over the lazy dog. The quick brown fox jumped over the lazy dog. The quick brown fox jumped over the lazy dog. The quick brown fox jumped over the lazy dog.

6.1 The Fox Jumps Over

The quick brown fox jumped over the lazy dog. The quick brown fox jumped over the lazy dog. The quick brown fox jumped over the lazy dog. The quick brown fox jumped over the lazy dog. The quick brown fox jumped over Uroborus (Figure 6.1). The quick brown fox jumped over the lazy dog.

The quick brown fox jumped over the lazy dog. The quick brown fox jumped over the lazy dog. The quick brown fox jumped over the lazy dog. The quick brown fox jumped over the lazy dog. The quick brown fox jumped over the lazy dog. The quick brown fox jumped over the lazy dog. The quick brown fox jumped over the lazy dog. The quick brown fox jumped over the lazy dog. The quick brown fox jumped over the lazy dog. The quick brown fox jumped over the lazy dog. The quick brown fox jumped over the lazy dog.

The quick brown fox jumped over the lazy dog. The quick brown fox jumped over the lazy dog. The quick

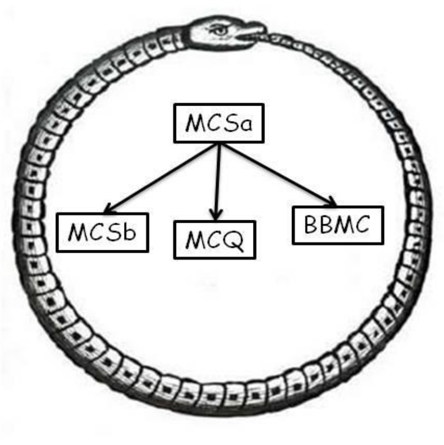

Figure 6.1: An alternative hierarchy of the algorithms.

brown fox jumped over the lazy dog. The quick brown fox jumped over [2] the lazy dog. The quick brown fox jumped over the lazy dog. The quick brown fox jumped over the lazy dog. The quick brown fox jumped over the lazy dog. The quick brown fox jumped over the lazy dog.

6.2 The Lazy Dog

The quick brown fox jumped over the lazy dog. The quick brown fox jumped over the lazy dog. The quick brown fox jumped over the lazy dog.

The quick brown fox jumped over the lazy dog. The quick brown fox [4] jumped over the lazy dog. The quick brown fox jumped over the lazy dog. The quick brown fox jumped over the lazy dog. The quick brown fox jumped over the lazy dog. The quick brown fox jumped over the lazy dog.

Appendices

Appendix A

Running the Programs

An example of running from the command line is as follows:

```
> java MaxClique BBMC1 brock200_1.clq 14400
```

This will apply $BBMC$ with $style = 1$ to the first brock200 DIMACS instance allowing 14400 seconds of cpu time.

Appendix B

Generating Random Graphs

We generate Erdós-Rënyi random graphs $G(n, p)$ where n is the number of vertices and each edge is included in the graph with probability p independent from every other edge. It produces a random graph in DIMACS format with vertices numbered 1 to n inclusive. It can be run from the command line as follows to produce a clq file

```
> java RandomGraph 100 0.9 > 100-90-00.clq
```

Bibliography

[1] DIMACS clique benchmark instances. ftp://dimacs.rutgers.edu/pub/challenge/graph/benchmarks/clique.

[2] Peter Cheeseman, Bob Kanefsky, and William M. Taylor. Where the really hard problems are. In *Proceedings IJCAI'91*, pages 331–337, 1991.

[3] Torsten Fahle. Simple and Fast: Improving a Branch-and-Bound Algorithm for Maximum Clique. In *Proceedings ESA 2002, LNCS 2461*, pages 485–498, 2002.

[4] Brian Hayes. Can't get no satisfaction. *American Scientist*, 85:108–112, 1997.

[5] Kuikui Liuös Xiujin Shiás and Yue Liës. Integrated Architecture for Web Application Development Based on Spring Framework and Activiti Engine. *The International Conference on E-Technologies and Business on the Web (EBW2013)*, pages 52–56, 2013.