Parallel Massive Dataset Cleaning

Jianmeng Yu

4th Year Project Report

Computer Science

School of Informatics

University of Edinburgh

2018

Abstract

This project applies the decision algorithm[1] developed by Matt Pugh in 2015 on a massive parallel scale, to remove the large amount of False Positive fish detections in the Fish4Knowledge (F4K) dataset[2], without losing too many True Positives.

Also, according to Qiqi Yu's assessed runtime[3], the cleaning process will take more than 1000 days to complete on a 40-core machine. Simply putting the process onto parallel scale will not be sufficient, optimization of the code is also essential for making the processing more feasible.

This document describes the detail of various approach to reduce unnecessary work during pre-processing, improve the cleaning algorithm, and evaluating efficiency of different implementations of the machine learning techniques used. A more detailed roadmap this project is provided in the Chapter 1.

Acknowledgements

I would like to thank my project supervisor, Prof. Fisher, for his constant, patient support throughout the year. Without his expert knowledge in the field, it would be impossible for me to navigate through all of the data source and prior work of the Fish4Knowledge project.

I would also like to thank Mr. Matthew Pugh for finding out time answering my questions on the project, and precious advices on the implementation of his algorithms.

I must also extend gratitude to my friends, and my family back in China, for all their help and encouragement during my study.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(*Jianmeng Yu*)

Table of Contents

Chapter 1

Introduction

This project applies the previous work of Pugh[1] and Yu[3] on massive parallel scale (detail in Chapter 2), while trying to reduce the computational cost of the cleaning algorithm. The main goal of this project is to produce a cleaned subset of Fish4Knowledge Original Data-Set (FDS), so researchers in the future may extract more reliable information from it.

An introduction on background of the F4K project, related background of the decision algorithm, framework design, and libraries used are included in Chapter 2.

It is identified that the most time-consuming part of the cleaning is the extraction of data from storage. Details of the data sources used are described in Chapter 3.

Chapter 4 describes the first stages of the cleaning: early detection removal, feature extraction, preprocessing for classification in the next stage, and also approaches like translating Matlab code into Python to reducing runtime and workload.

Chapter 5 discusses the final classifiers used in the cleaning, with evaluation of the results and comparison between different algorithms. Conclusions and possible future work needed are included in Chapter 6.

1.1 Big Data

chui bi

chui bi chui bi

blah

Chapter 2

Background

2.1 Fish4Knowledge Data Set

Say something here

2.2 Classification Schema

2.3 Original Pipeline Classifier

2.4 Message Passing Interface (MPI)

Chapter 3

Data Source

Under limitations of disk space and access speed, loading a large SQL database dump file into server and performing 400,000 queries is very unnecessary and time consuming, hence making it the slowest part of the cleaning. Since each record needed for the cleaning are independent, an alternative is to use python script with stdin/out pipeline to parse and partition the dump file directly into usable csv files, the details of the pipeline and data sources are described in Chapter 3.

Chapter 4

Preprocessing

Chapter 5

Classifiers

Chapter 6

Conclusion

The document structure should include:

- The title page in the format used above.
- An optional acknowledgements page.
- The table of contents.
- The report text divided into chapters as appropriate.
- The bibliography.

Commands for generating the title page appear in the skeleton file and are self explanatory. The file also includes commands to choose your report type (project report, thesis or dissertation) and degree. These will be placed in the appropriate place in the title page.

The default behaviour of the documentclass is to produce documents typeset in 12 point. Regardless of the formatting system you use, it is recommended that you submit your thesis printed (or copied) double sided.

The report should be printed single-spaced. It should be 30 to 60 pages long, and preferably no shorter than 20 pages. Appendices are in addition to this and you should place detail here which may be too much or not strictly necessary when reading the relevant section.

6.1 Using Sections

Divide your chapters into sub-parts as appropriate.

6.2 Citations

Note that citations can be generated using `BibTeX` or by using the `thebibliography` environment. This makes sure that the table of contents includes an entry for the bibliography. Of course you may use any other method as well.

6.3 Options

There are various documentclass options, see the documentation. Here we are using an option (`bsc` or `minf`) to choose the degree type, plus:

- `frontabs` (recommended) to put the abstract on the front page;
- `twoside` (recommended) to format for two-sided printing, with each chapter starting on a right-hand page;
- `singlespacing` (required) for single-spaced formating; and
- `parskip` (a matter of taste) which alters the paragraph formatting so that paragraphs are separated by a vertical space, and there is no indentation at the start of each paragraph.

Bibliography

[1] Matthew Pugh. Removing false detections from a large fish image data-set. Msc dissertation, The University of Edinburgh, 2015.

[2] Fish4knowledge homepage. http://fish4knowledge.eu/. Accessed: 2018-01-08.

[3] Qiqi Yu. Adding temporal constraints to a large data cleaning problem. Msc dissertation, The University of Edinburgh, 2016.