Complex Graph Stream Mining

Shirui Pan

Faculty of Engineering and Information Technology

University of Technology Sydney

A thesis submitted for the degree of

Doctor of Philosophy

October 2015

CERTIFICATE OF AUTHORSHIP/ORIGINALITY

I certify that the work in this thesis has not previously been submitted for a degree nor has it been submitted as part of requirements for a degree except as fully acknowledged within the text.

I also certify that the thesis has been written by me. Any help that I have received in my research work and the preparation of the thesis itself has been acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis.

Signature of Student

Acknowledgements

I would like to express my earnest thanks to my supervisors, Professor Chengqi Zhang and Professor Xingquan Zhu, who have provided tremendous support and guidance for my research in the past four years. Prof Zhang provided me an opportunity to study in the stimulating and interactive centre for Quantum Computation and Intelligent Systems (QCIS), where I met and leant a lot from many smart and sharp people. I benefit significantly from his unselfish help and invaluable suggestion on my research career. I would like to thank Prof Xingquan Zhu for his continuous guidance and supervisions during my Ph.D. study. Discussing a problem with him has been always a pleasure and eye-opening experience. He always gives me sufficient freedom and encouragement to think and explore my research interest. His vision, creativeness and enthusiasm in solving challenging problems has greatly encouraged me and inspired my works. Without his endless patience, generous support, and constant guidance, this thesis could not have been accomplished.

I would also like to thank all the people that had a positive influence on my day-to-day enjoyment of the job. My office-mates, past and present: Jia Wu, Yifan Fu, Guodong Long, Jing Jiang, Peng Zhang, Tianyi Zhou, Wei Bian, Wei Wang, Xun Wang, Ting Guo, Lianhua Chi, Meng Fang, Mingsong Mao, Zhibin Hong, Hongshu Chen, Shaoli Huang, Haishuai Wang, Mingming Gong, Sujuan Hou, Qin Zhang, Maoying Qiao, Zhiguo Long, Hua Meng, Zhe Xu, Bozhong Liu, Tongliang Liu, Junyu Xuan, and Jiang Bian. They are the ones who have given me support during both joyful and stressful times, to whom I will always be thankful.

I also wish to express my appreciation to the financial support I gained for my study. Special thanks go to China Scholarship Council (CSC), University of Technology Sydney (UTS), centre for Quantum Computation and Intelligent Systems (QCIS), ICDE student travel grant, and CIKM student travel grant.

Finally, and above all, I want to thank my family for their continuous support. I especially thank my wife, Yu Zheng, who took care of the daily life of our little baby, Yixin Pan, and myself, and shared all my pain, sorrow and joy in every moment of my research. I would like to thank my parents, brothers, and sisters for their unconditional encouragement and support, both emotionally and financially. No words could possibly express my deepest gratitude for their endless love, self-sacrifice and unwavering help. To them I dedicate this dissertation.

Abstract

Recent years have witnessed a dramatic increase of information due to the ever development of modern technologies. The large scale of information makes data analysis, particularly data mining and knowledge discovery tasks, unprecedentedly challenging. First, data is becoming more and more interconnected. In a variety of domains such as social networks, chemical compounds, and XML documents, data is no longer represented by a flat table with instance-feature format, but exhibits complex structures indicating dependency relationships. Second, data is evolving more and more dynamically. Emerging applications such as social networks continuously generate information over time. Third, the learning tasks in many real-life applications become more and more complicated in that there are various constraints on the number of labelled data, class distributions, misclassification costs, or the number of learning tasks etc.

Considering the above challenges, this research aims to investigate theoretical foundations, study new algorithm designs and system frameworks to enable the mining of complex graph streams from three aspects, including (1) Correlated Graph Stream Mining, (2) Graph Stream Classifications, and (3) Complex Task Graph Classification.

In particular, correlated graph stream mining intends to carry out structured pattern search and support the query of similar graphs from a graph stream. Due to the dynamic changing nature of the streaming data and the inherent complexity of the graph query process, treating graph streams as static datasets is computationally infeasible or ineffective. Therefore, we proposed a novel algorithm, CGStream, to identify correlated graphs from a data stream, by using a sliding window, which covers a number of consecutive batches

of stream data records. Experimental results demonstrate that the proposed algorithm is several times, or even an order of magnitude, more efficient than the straightforward algorithms.

Graph stream classification aims to build effective and efficient classification models for graph streams with continuous growing volumes and dynamic changes. We proposed two methods for complex graph stream classification. Due to the inherent complexity of graph structure, labelling graph data is very expensive. To solve this problem, we proposed a gLSU algorithm, which aims to select discriminative subgraph features with minimum redundancy by using both labelled and unlabelled graphs for graph streams. The second approach handles graph streams with imbalanced class distributions and noise. Both frameworks use an instance weighting scheme to capture the underlying concept drifts of graph streams and achieve significant performance gain on benchmark graph streams.

Complex task graph classification aims to address the graph classification problems with complex constraints. We studied two complex task graph classification problems, cost-sensitive graph classification of large-scale graphs and multi-task graph classification. As in medical diagnosis the misclassification cost/risk for different classes is inherently different and large scale graph classification is highly demanded in real-life applications, we proposed a CogBoost algorithm for cost-sensitive classification of large scale graphs. To overcome the limitation of insufficient labelled graphs for a specific learning task, we further proposed effective algorithms to leverage multiple graph learning tasks to select subgraph features and regularize multiple tasks to achieve better generalization performance for all learning tasks.

Contents

Contents i

List of Figures vii

List of Tables xi

Nomenclature xii

1 Introduction 1

1.1 Motivations and Significances 1

1.2 Research Problems 5

1.2.1 Graph Stream Search 5

1.2.2 Graph Stream Classification 6

1.2.3 Complex Task Graph Classification 6

1.3 Thesis Contributions 7

1.3.1 Graph Stream Search 7

1.3.2 Graph Stream Classification 7

1.3.3 Complex Task Graph Classification 8

1.4 Thesis Overview 8

1.5 Publications 10

2 Literature Review 13

2.1 Correlated Graph Search 13

2.2 Graph Classification 14

2.2.1 Similarity-based methods 14

2.2.2 Vector representation-based methods 14

2.2.3 Graph-based Learning for a Single Network 17
2.3 Imbalanced Data Classification 17
2.4 Data Stream and Graph Stream Classification 18
2.5 Cost-sensitive Learning 19
2.6 Multi-task Learning 20
2.7 Key techniques 21
2.7.1 gSpan Algorithm 21
2.7.2 Column generation Algorithm 21
2.7.3 Cutting Plane Algorithm 22

3 Preliminary **23**
3.1 Definitions 23
3.2 Notations 26
3.3 Benchmark Graph Datasets 26

I Graph Stream Query **33**

Graph Stream Search: Overview **35**

4 Continuous Correlated Graph Query for Data Streams **37**
4.1 Introduction 37
4.2 Preliminaries and Problem Definition 40
4.2.1 Preliminaries 40
4.2.2 Problem definition 41
4.2.3 Challenges and Solutions 42
4.3 Frequency lower bound for candidate generation 44
4.3.1 Frequency lower bound 45
4.3.2 Estimation the increment of γ 48
4.4 Correlation upper bound and Heuristic rules for candidate pruning 49
4.4.1 Maximum Value of the Numerator 50
4.4.2 Minimum Value of the Denominator 51
4.4.3 Loose correlation upper bound 52
4.4.4 Heuristic Rule 52

4.5 Algorithm 53
4.6 Experimental Result 56
4.6.1 Experiment setup 56
4.6.2 System runtime performance 57
4.6.3 Query Precision 63
4.7 Conclusions 64

II Graph Stream Classification 65

Graph Stream Classification: Overview 67

5 Graph Stream Classification using Labeled and Unlabeled Graphs 69
5.1 Introduction 69
5.2 Problem Definition & Overall Framework 72
5.2.1 Overall Framework 73
5.3 Minimum Redundancy Subgraph Feature Selection 74
5.3.1 Informativeness of the Feature Set 75
5.3.2 Informative Subgraph Feature Selection 78
5.3.3 Minimum Redundancy Subgraph Feature Selection 79
5.4 gSLU Algorithm 84
5.5 Experiments 87
5.5.1 Experimental Settings 87
5.5.2 Experimental Results 88
5.6 Conclusion 97

6 Imbalanced and Noisy Graph Stream Classification 99
6.1 Introduction 99
6.1.1 Imbalanced Graph Classification 99
6.1.2 Graph Stream Classification 100
6.2 Overall Framework 102
6.3 Learning from a Local Chunk with Noisy and Imbalanced Graphs 105
6.3.1 Framework of Linear Boosting Algorithm 105
6.3.2 gBoost Algorithm for Balanced graph classification 106

6.3.3 Objective Function for Imbalanced and Noisy Data 107
6.3.4 Linear Boosting with Graph Data 108
6.4 gEBoost algorithm 114
6.5 Experiments 117
6.5.1 Experimental Settings 117
6.5.2 Experimental Results 120
6.6 Conclusion 132

III Complex Task Graph Classification 133

Complex Task Classification: Overview 135

7 Cost-sensitive Learning for Large Scale Graph Classification 137
7.1 Introduction 137
7.1.1 Cost-Sensitive Graph Classification 137
7.1.2 Fast Training for Large Scale Graphs 139
7.2 Problem Definition and Overall Framework 141
7.2.1 Overall Framework 143
7.3 Cost-Sensitive Learning for Graph Data 143
7.3.1 Optimal Cost-sensitive Loss Function 144
7.3.2 Cost-Sensitive Formulation for Graphs 146
7.3.3 Boosting for Cost-sensitive Learning on Graphs 147
7.3.4 Cost-sensitive Subgraph Exploration 149
7.4 Fast Training for Large Scale Graphs 151
7.4.1 From l-Slacks to 1-Slack Formulation 151
7.4.2 Cutting-plane Algorithm for Fast Training 152
7.5 Time Complexity Analysis: Theoretical Aspect and Practice . . . 154
7.5.1 Time complexity of Subgraph Mining 154
7.5.2 Time complexity of LP Solving 155
7.6 Experiments 156
7.6.1 Experimental Settings 156
7.6.2 Experimental Results 158
7.7 Conclusion 166

8 Joint Structure Feature Exploration and Regularization for Multi-Task Graph Classification 167

8.1 Introduction . . . 167

8.2 Problem Definition & Preliminaries . . . 172

8.2.1 Preliminaries . . . 172

8.3 Multi-task Graph Classification . . . 173

8.3.1 Regularized Multi-task Graph Classification Formulation . 173

8.3.2 Multi-task Graph Classification: Challenges and Solution Sketch . . . 175

8.3.3 Optimal Subgraph Candidate Exploration . . . 177

8.3.4 Multi-task Graph Classification Algorithm . . . 179

8.3.5 Multi-Task Driven Subgraph Mining . . . 182

8.4 Experiment . . . 185

8.4.1 Experimental Settings . . . 185

8.4.2 Experimental Results . . . 187

8.5 Discussion . . . 195

8.6 Conclusion . . . 197

9 Conclusions and Future Work 199

9.1 Summary of This Thesis . . . 199

9.2 Future Work . . . 200

Appendix A 203

A.1 Duality of Eq.(6.4) . . . 203

A.2 Duality of Eq. (7.7) . . . 204

A.3 Equality of Eq. (7.7) and Eq. (7.10) . . . 204

A.4 Duality of Eq.(7.10) . . . 205

References 207

List of Figures

1.1 Examples of graphs for different applications. 2

1.2 An illustrated example of correlated graph search 3

1.3 Graph representation for a scientific paper 4

2.1 Subgraph-based methods for graph classification 15

3.1 Graph and Subgraph Examples. 24

4.1 A framework of sliding window based correlated graph query for data streams 41

4.2 System runtime consumption with respect to different θ values. . 59

4.3 System accumulative runtime consumption with respect to different θ values. 60

4.4 System accumulative runtime consumption with respect to different w values. 61

4.5 Comparison on different w values. (A) and (B), system runtime in each time point; (C) and (D), system accumulative runtime in each time point. 61

4.6 System accumulative runtime consumption with different $|D_j|$ values. 62

4.7 Comparison on different m values. (A) system runtime, (B) system accumulative runtime. 62

4.8 Query precisions with respect to different parameters settings. . . 63

5.1 An example demonstrating subgraph correlations 70

5.2 A framework for semi-supervised graph data stream classification 73

5.3 An illustrated example for subgraph selection with minimum redundancy 81
5.4 Comparison of the proposed minimum redundancy subgraph feature selection with other algorithms. 89
5.5 Accuracy *w.r.t.* different number of labeled graphs on *DBLP stream* 90
5.6 Accuracy *w.r.t.* different seizes of labeled graphs on *NCI stream* . 91
5.7 Average accuracy (and standard deviation) *v.s.* labeled graph sizes $|D_t^l|$ with chunk size $|D_t|$=800, feature size $m = 20$. 92
5.8 Accuracy *w.r.t.* different number of features on *DBLP stream* . . 93
5.9 Accuracy *w.r.t.* different number of features on *NCI stream* with each chunk containing 800 graphs, and the size of labeled data in each chunk is 30. The number of features selected in each chunk: (A) 10; (B) 20; (C) 30. 94
5.10 Averaged accuracy (and standard deviation) *v.s.* number of features m, with chunk size $|D_t|$=800, feature size $m = 20$. 94
5.11 Accuracy *w.r.t.* different chunk sizes on *DBLP stream* with each chunk containing 30 labeled graphs, and the number of features in each chunk is 50. The batch sizes vary as: (A) 1000; (B) 800; (C) 600. 95
5.12 System accumulated runtime *v.s.* number of graphs processed over stream. ($|D_t| = 800$, $|D_t^l| = 10\%|D_t|$, and m=20). 96
5.13 System accumulated runtime *v.s.* different chunk sizes $|D_t|$, $|D_t^l| = 10\%|D_t|$. m=20; (A) Results on DBLP stream; (B) Results on NCI stream. 96

6.1 A framework for imbalanced noisy graph stream classification. . . 103
6.2 The proposed boosting framework for learning from noisy and imbalanced graphs in each chunk 104
6.3 A conceptual view of graph weighting scheme for imbalanced graph stream classification 115
6.4 Comparison of different algorithms for imbalanced graph stream classification 121

6.5 AUC *w.r.t.* different noise levels on *NCI stream* with ensemble size k=10 and chunk size D_t= 1500. (A)$Z = 5$; (B) $Z = 15$. 123

6.6 AUC *w.r.t.* different noise levels on *DBLP stream* with ensemble size k=10 and chunk size D_t= 800. (A)$Z = 5$; (B) $Z = 15$. . . . 123

6.7 AUC *w.r.t.* different noise levels on *Twitter stream*. Figures on the left panel are plotted with respect to uniform intervals of chunks in the x-axis, and figures on the right panel are plotted with respect to uniform intervals of weeks in the x-axis. 124

6.8 Averaged AUC values (and standard deviation) *v.s.* different noise degrees Z, with ensemble size k=10. 126

6.9 AUC *w.r.t.* different ensemble sizes on *DBLP stream* with chunk size $|D_t|$=800. 127

6.10 AUC *w.r.t.* different ensemble sizes on *Twitter stream*. Figures on the left panel are plotted with respect to uniform intervals of chunks in the x-axis, and figures on the right panel are plotted with respect to uniform intervals of weeks in the x-axis. 128

6.11 AUC *w.r.t.* different chunk size on *NCI stream* with ensemble size k=10. (A) $|D_t| = 1000$; (B) $|D_t| = 2000$. 129

6.12 AUC *w.r.t.* different chunk size on *DBLP stream* with ensemble size k=10. (A) $|D_t| = 600$; (B) $|D_t| = 1000$. 129

6.13 System accumulated runtime *v.s.* number of graphs processed over stream. (A) NCI stream; (B) DBLP stream; (C) Twitter stream. . 131

6.14 System accumulated runtime *v.s.* different chunk sizes $|D_t|$. 132

7.1 Training time *w.r.t.* different number of graphs on NCI-1 dataset for gBoost [120] and igBoost algorithm [109]. Runtime of existing graph classification algorithms exponentially grows *w.r.t.* the increase of the training set size. 139

7.2 The proposed fast cost-sensitive boosting for graph classification framework 142

7.3 Different loss functions and formulations with respect to support vector machines (SVMs): (A) Standard Hinge Loss, (B) Cost-sensitive Hinge Loss with $C_1 = 4$ and $C_{-1} = 2$, and (C) Different SVM formulations with Standard Hinge Loss and Cost-sensitive Hinge Loss (**cf.**[91]). 145
7.4 Experimental Results. (A) Average cost, (B) Time Complexity. 159
7.5 Runtime performance in each iterations. Runtime consumption for (A) gBoost, (B) igBoost, (C) CogBoost-a, and (D) CogBoost-1. . 162
7.6 Average Cost with respect to different C_1 value 163
7.7 Average cost (left y-axis) and algorithm runtime (right y-axis) with respect to different ϵ values ($x-$axis). (A) NCI-1, and (B) NCI-33 164

8.1 The comparisons of the Top 5 most discriminative subgraphs for each graph classification task 169
8.2 Accuracy comparisons on training and test graphs with 50 training graphs for each task 170
8.3 The classification accuracy of each single task *w.r.t.* the number of training graphs in each task. 189
8.4 The AUC values of each single task *w.r.t.* the number of training graphs in each task. 190
8.5 Pruning effectiveness with different pruning modules on NCI tasks for subgraph mining. A) Running time; B) Number of enumerated subgraphs. 194

List of Tables

1.1 Structure of the thesis with reference to the chapters. 9

3.1 Important notations used in the chapter 27
3.2 Description of Graph Datasets Used in the Thesis 28
3.3 DBLP-balanced used in this thesis 29
3.4 DBLP-imbalanced graph stream used in experiments 30

4.1 Effectiveness of Pruning in CGStream with $\theta = 0.8$ (seconds)[Acc. Time - accumulative runtime] 58

5.1 Pairwise t-test results with labeled graph sizes $|D_t^l|$. A, B, and C denote gSLU, gSemi+Stream, and IG+Stream, respectively. . . . 92
5.2 Pairwise t-test results with various feature size m. A, B, and C denote gSLU, gSemi+Stream, and IG+Stream, respectively. . . . 95

6.1 NCI cancer screen datasets used in the experiments 118
6.2 Average AUC values and standard deviations on DBLP Streams w.r.t Different Imbalance Degrees 130

7.1 Average Time Consumption in Each Iteration (Seconds) 164

8.1 Accuracies on 9 NCI graph classification tasks *w.r.t* different numbers of training graphs in each task 188
8.2 AUC values on 9 NCI graph classification tasks *w.r.t* different numbers of training graphs in each task 188
8.3 Accuracies on PTC tasks 191
8.4 AUC values on PTC tasks 191

8.5 Running statistics *w.r.t* different K values for MTG-ℓ_{21} (50 training graphs for each task, $S_{max} = 150$) 192

8.6 Results *w.r.t.* different γ values for MTG-ℓ_1 (50 training graphs for each task, $S_{max} = 15$) 193