

**CONCEPT DRIFT DETECTION FOR
MACHINE LEARNING WITH
STREAM DATA**

Feng Gu

Faculty of Engineering and Information Technology
University of Technology Sydney

A thesis submitted for the Degree of
Doctor of Philosophy

December 2019

CERTIFICATE OF ORIGINAL AUTHORSHIP

I, Feng Gu declare that this thesis, is submitted in fulfilment of the requirements for the award of doctorate, in the Faculty of Engineering and Information Technology at the University of Technology Sydney. This thesis is wholly my own work unless otherwise reference or acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis. This document has not been submitted for qualifications at any other academic institution.

This research is supported by the Australian Government Research Training Program.

Feng Gu

December 2019

Acknowledgements

This has been a memorial and exciting journey. I would like to extend my warm gratitude to the people who inspired and helped me in many ways.

I would like to express my earnest gratitude to my supervisors, Professor Guangquan Zhang and Distinguished Professor Jie Lu, for their knowledgeable suggestions and critical comments. During my doctoral research, their comprehensive guidance always illuminated the way. My discussions with them greatly improved the scientific aspect and quality of my research. Their strict academic attitude and respectful personality benefited my PhD study and will be a great memory throughout my life.

I am honored to have met all the talented researchers of the Decision Systems & e-Service Intelligence Lab (DeSI) at Centre for Artificial Intelligence. I have greatly enjoyed the pleasurable and plentiful research opportunities I shared with them. I would like to give my special thanks to Dr Ning Lu, Dr Anjin Liu and Dr Fan Dong who inspired me to concept drift research, as well as Feng Liu, Yiliao Song and Zhen Fang with whom I engaged in daily discussions that were rewarding and fun.

I kindly thank Ms Jemima Moore and Ms Michele Mooney for polishing the language used in my thesis and publications. I have learnt much about academic writing from them.

I am grateful to the School of Software in the Faculty of Engineering and Information Technology at the University of Technology Sydney. This study was supported by the Australian Research Council (ARC) discovery project DP150101645.

Finally, I would like to express my heartfelt appreciation and gratitude to my family for their love and support.

Abstract

Machine learning in streaming data is often inhibited by arbitrary changes of the data distribution. Particularly, classification boundary change, also known as concept drift, is the major cause of machine learning performance deterioration.

Accurately and efficiently detecting concept drift remains challenging because of inherent limitations of stream data - non-stationarity, velocity and availability of true label data. The non-stationarity of the stream data causes performance degradation of pretrained models and the high velocity of the data generation requires highly efficient prediction algorithms for real time applications. The theoretical foundations of existing drift detection methods - two-sample distribution tests and monitoring classification error rate, both suffer from inherent limitations such as the inability to distinguish virtual drift (changes not affecting the classification boundary, will introduce unnecessary model maintenance), limited statistical power, or high computational cost. Furthermore, no existing detection method can provide information about the trend of the drift, which could be invaluable for model maintenance.

To better address concept drift problems, this thesis first proposes a novel concept drift detection method based on **Neighbor Search Discrepancy (NSD)**, a new statistic that measures the classification boundary difference between two samples. The proposed method uses true label data to detect concept drift with high accuracy while

ignoring virtual drift. It can also indicate the direction of the classification boundary change by identifying invasion or retreat of a certain class, which is also an indicator of separability change between classes.

To improve concept drift adaptation efficiency, based on NSD, this thesis proposes two novel instance selection methods for both concept drift detection - **Decision Region Support Set (DRS)** and classification - **Decision Region Border Set (DRB)**. The unified framework yields reduction instances for both objectives simultaneously without computational overhead. The drift detection method efficiently detects concept drift without relying on resampling technique. The reduction rule based on Neighbor Search better estimates decision boundaries, resulting in improved classification accuracy.

For scenarios where true label data is unavailable, this thesis first proposes a novel distribution change detection method - **Equal Density Estimation (EDE)** based on the estimation of equal density regions. The aim is to overcome the issues of instability and inefficiency that underlie methods of predefined space partitioning schemes. This method is general, nonparametric and requires no prior knowledge of the data distribution.

Finally, in order to detect concept drift without true label data, this thesis introduces a novel categorization of drift types - maintainable and unmaintainable drift, to describe the necessity of model maintenance in different scenarios. Then we develop a unique drift detection algorithm based on **Probability Percentile Discrepancy (PPD)**, which detects only maintainable drift without relying on true label data.

In summary, this thesis targets a critical issue in modern machine learning research. The approaches taken in the thesis of building effective and efficient concept drift detection algorithms are novel and practical. There has been no

previous study on the theories of neighbor search discrepancy and maintainable concept drift. The findings of this thesis contribute to both scientific research and practical applications.

Table of Contents

| | |
|---|-------------|
| CERTIFICATE OF ORIGINAL AUTHORSHIP | iii |
| Acknowledgements | v |
| Abstract | vii |
| List of Figures | xv |
| List of Tables | xxi |
| List of Abbreviations | xxiv |
| 1 Introduction | 1 |
| 1.1 Background | 1 |
| 1.2 Research Questions and Objectives | 4 |
| 1.3 Research Contributes | 8 |
| 1.4 Research Significance | 9 |
| 1.5 Thesis Structure and Thesis Notations | 10 |
| 1.6 Publications Related to this Thesis | 12 |
| 2 Literature Review | 15 |

| | | |
|----------|--|-----------|
| 2.1 | Problem Description | 15 |
| 2.1.1 | Concept drift definition and the sources | 15 |
| 2.1.2 | The types of concept drift | 17 |
| 2.2 | Concept Drift detection | 19 |
| 2.2.1 | A general framework for drift detection | 19 |
| 2.2.2 | Concept drift detection algorithms | 21 |
| 2.2.2.1 | Error rate-based drift detection | 21 |
| 2.2.2.2 | Data Distribution-based Drift Detection | 25 |
| 2.2.2.3 | Multiple Hypothesis Test Drift Detection | 27 |
| 2.3 | Concept Drift adaptation | 30 |
| 2.3.1 | Retraining models | 31 |
| 2.3.2 | Adaptive models | 33 |
| 2.3.3 | Adaptive ensembles | 35 |
| 3 | Real Concept Drift Detection Based on Neighbor Search Discrepancy | 39 |
| 3.1 | Introduction | 39 |
| 3.2 | A New Measure: Neighbor Search Discrepancy | 42 |
| 3.2.1 | Preliminaries | 43 |
| 3.2.2 | Neighbor Search | 44 |
| 3.2.3 | Neighbor Search Volume Ratio | 49 |
| 3.3 | Neighbor Search Discrepancy Drift Detection | 55 |
| 3.3.1 | Classification Gap | 56 |
| 3.3.2 | Detect Classification Gap Changes | 57 |
| 3.4 | Experimental evaluation | 60 |
| 3.4.1 | Evaluating NSD of a single starting point | 62 |

| | | |
|----------|--|------------|
| 3.4.2 | Evaluating NSD of multiple starting points | 66 |
| 3.4.3 | Evaluating drift detection method | 69 |
| 3.5 | Summary | 76 |
| 4 | Concept Drift-Tolerant Instance Selection | 79 |
| 4.1 | Introduction | 79 |
| 4.2 | Decision region estimation using neighbor search | 81 |
| 4.2.1 | Decision regions | 82 |
| 4.2.2 | Neighbor search | 84 |
| 4.2.3 | Automatically choosing k | 86 |
| 4.2.4 | Noise removal | 90 |
| 4.3 | Concept drift-tolerant instance selection | 91 |
| 4.3.1 | Instance selection for drift detection | 92 |
| 4.3.2 | Instance selection for classification | 98 |
| 4.4 | Experimental evaluation | 102 |
| 4.4.1 | Space and time efficiency | 103 |
| 4.4.2 | Classification with stationary data | 113 |
| 4.4.3 | Classification with concept drift data | 118 |
| 4.5 | Summary | 121 |
| 5 | Concept Drift Detection by Equal Density Estimation | 124 |
| 5.1 | Introduction | 124 |
| 5.2 | Equal density estimation | 126 |
| 5.3 | Change detection method | 128 |
| 5.3.1 | Statistical guarantee | 130 |
| 5.3.2 | Window models | 130 |

| | | |
|----------|---|------------|
| 5.4 | Empirical Evaluation | 131 |
| 5.4.1 | Evaluating ω | 131 |
| 5.4.2 | Evaluating the change detection method | 133 |
| 5.5 | Summary | 141 |
| 6 | Maintainable Concept Drift Detection without True Labels | 142 |
| 6.1 | Introduction | 142 |
| 6.2 | Maintainable Concept Drift | 143 |
| 6.3 | Prediction-based Drift Detection | 148 |
| 6.4 | Experimental evaluation | 151 |
| 6.4.1 | Various drift types | 153 |
| 6.4.2 | High dimensional data | 157 |
| 6.4.3 | Real world data set | 159 |
| 6.5 | Summary | 160 |
| 7 | Conclusion and Future Research | 163 |
| 7.1 | Conclusions | 163 |
| 7.2 | Future Study | 166 |
| | Bibliography | 168 |

List of Figures

| | | |
|-----|--|----|
| 1.1 | Concept drift in mobile phone usage (data used in figure are for demonstration only). [73] | 2 |
| 1.2 | Framework for handling concept drift in machine learning. [73] | 3 |
| 1.3 | Thesis structure | 11 |
| 2.1 | Three sources of concept drift. [73] | 17 |
| 2.2 | An example of concept drift types. [73] | 18 |
| 2.3 | An overall framework for concept drift detection | 20 |
| 2.4 | Landmark time window for drift detection. The starting point of the window is fixed, while the end point of the window will be extended after a new data instance has been received. | 22 |
| 2.5 | Two time windows for concept drift detection. The New Data window has to be defined by the user. | 24 |
| 2.6 | Two sliding time windows, of fixed size. The Historical Data window will be fixed while the New Data window will keep moving. | 26 |
| 2.7 | Parallel multiple hypothesis test drift detection. | 27 |
| 2.8 | Hierarchical multiple hypothesis test drift detection. | 29 |

| | | |
|------|--|----|
| 2.9 | A new model is trained with latest data to replace the old model when a concept drift is detected. | 31 |
| 2.10 | A decision tree node is replaced with a new one as its performance deteriorates when a concept drift occurs in a subregion. | 33 |
| 2.11 | A new base classifier is added to the ensemble when a concept drift occurs. | 35 |
| 3.1 | Examples of finding 3 nearest neighbors and their corresponding search steps following neighbor searches of different shapes. | 46 |
| 3.2 | The PDF of $\mathcal{V}(k, n, \lambda)$ converges to that of $\text{Gamma}(k, \lambda)$ as n becomes greater than k | 50 |
| 3.3 | Neighbor search discrepancy equals area under curve of Beta PDF between $[0, 0.5]$ | 54 |
| 3.4 | Neighbor search discrepancy $\text{NSD}(10, 4)$ describes the probability of no more than 3 points from sample X_2 falling in $S_{(k_1=10)}$, which is determined by the 10th neighbor points from sample X_1 | 55 |
| 3.5 | Different types of real concept drift and virtual drift reflecting on classification gap change. | 56 |
| 3.6 | Joint neighbor search is constructed by combining simple spherical searches on sample X_1 , and then invasion and retreat of classification gap change is tested on sample X_2 | 60 |
| 3.7 | Comparison of the calculated NSD (solid horizontal lines) with Monte Carlo simulation (colored lines). The simulation result converges to NSD as sample size increases. | 63 |

-
- 4.1 Different shapes of decision region when the distributions of samples of two classes have different relative positions - (a) separate, (b) joint and (c) overlapping. 84
- 4.2 Joint neighbor search is constructed by combining spherical searches with different increasing rate and is used to estimate the decision region. 86
- 4.3 Neighbor searches for estimating different types of decision regions. 87
- 4.4 Decision region support set reduction process. (a) Points from the origin set (dots) are grouped by their neighboring points in the reference set (circles); (b) in each group, the points except the furthest one are removed, meanwhile marginally reducing the volume of the search step; (c) the search step increases to find the next neighbor so that the lost volume is corrected. 98
- 4.5 Decision region border for different distributions of two sample classes. 100
- 4.6 DRS reduction result for binary classification sample sets of different distribution. (a) 2D normal distribution; (b) linear decision boundary; (c) circular non-linear decision boundary. 104
- 4.7 Instance selection results on sample set of 2D normal distribution. The algorithms for comparison include: Decision Region Boundary (DRB, proposed), Class Conditional Instance Selection (CCIS, [79]), Incremental Reduction Optimization Procedure 3 (DROP3, [109]), Fast Condensed Nearest Neighbor (FCNN, [10]), Instance-Based Learning 3 (IB3, [1]), Iterative Case Filtering Algorithm (ICF, [23]) and Template Reduction for KNN (TRKNN, [36]). 105

| | | |
|------|---|-----|
| 4.8 | Instance selection results on sample set of 2D linearly separable distribution. The algorithms for comparison include: Decision Region Boundary (DRB, proposed), Class Conditional Instance Selection (CCIS, [79]), Decremental Reduction Optimization Procedure 3 (DROP3, [109]), Fast Condensed Nearest Neighbor (FCNN, [10]), Instance-Based Learning 3 (IB3, [1]), Iterative Case Filtering Algorithm (ICF, [23]) and Template Reduction for KNN (TRKNN, [36]) | 106 |
| 4.9 | Instance selection results on sample set of 2D non-linearly (circularly) separable distribution. The algorithms for comparison include: Decision Region Boundary (DRB, proposed), Class Conditional Instance Selection (CCIS, [79]), Decremental Reduction Optimization Procedure 3 (DROP3, [109]), Fast Condensed Nearest Neighbor (FCNN, [10]), Instance-Based Learning 3 (IB3, [1]), Iterative Case Filtering Algorithm (ICF, [23]) and Template Reduction for KNN (TRKNN, [36]) | 107 |
| 4.10 | Instance selection space efficiency with different sample sizes. (a-c) Number of selected instances with respect to sample size. (d) Average retention rate for all datasets. | 108 |
| 4.11 | Instance selection space efficiency with different dimensions. (a-c) Number of selected instances with respect to dimension. (d) Average retention rates for all datasets. | 110 |
| 4.12 | Instance selection time efficiency with different sample sizes. (a-c) Computation time (s) of instance selection methods with respect to sample size. (d) Average computation time (ms) per instance. | 112 |

| | | |
|------|---|-----|
| 4.13 | Instance selection time efficiency with different dimensions. (a-c) Computation time (s) of instance selection methods with respect to dimension. (d) Average computation time (ms) per instance. | 113 |
| 5.1 | Partitioning by fixed regions vs. equal density regions. | 127 |
| 5.2 | <i>DensityScale</i> increases when two data sets have different distribution. | 129 |
| 5.3 | ω varies as the mean μ of normally distributed data changes. | 132 |
| 5.4 | ω varies as the standard deviation σ of normally distributed data changes. | 133 |
| 5.5 | ω varies as the correlation ρ of normally distributed data changes. . | 134 |
| 5.6 | Pairwise reduction on two normally distributed data sets of the same mean. | 141 |
| 5.7 | Pairwise reduction on two normally distributed data sets of different mean. | 141 |
| 6.1 | Different types of $P(y X)$ changes with respect to decision boundary change. | 146 |
| 6.2 | $E(y^-)$ changes when IR drift occurs (class+ invading, class- re-treating). | 150 |
| 6.3 | Type (I) drift (maintainable) detection accuracy | 154 |
| 6.4 | Type (R) drift (maintainable) detection accuracy. | 155 |
| 6.5 | Type (IR) (maintainable) drift detection accuracy. | 156 |
| 6.6 | Type (II) drift (unmaintainable) false detection rate (lower is better). | 156 |
| 6.7 | Type (RR) drift (unmaintainable) false detection rate (lower is better). | 157 |

| | | |
|-----|---|-----|
| 6.8 | Classification accuracy on real world concept drift data set with different drift detection algorithms. The drift locations are marked as dashes. | 161 |
|-----|---|-----|

List of Tables

| | | |
|-----|--|----|
| 1.1 | Notations. | 13 |
| 3.1 | Comparison of concept drift detection accuracy (detection rate/false alarm rate) between proposed method (NSD) and competence model-based method (CM). | 72 |
| 3.2 | Comparison of running times (in seconds, unless indicated otherwise) of different dimensions and window sizes for proposed method (NSD) and the competence model-based method (CM). | 73 |
| 3.3 | Parameters of concept drift handling methods for real world datasets. | 75 |
| 3.4 | Classification accuracy of concept drift detection methods on real-world datasets with different base learners. The rank of the accuracy on each dataset is shown in brackets. The average rank indicates the overall performance of each algorithm (lower is better). | 77 |

| | | |
|-----|--|-----|
| 4.1 | Classification accuracy of instance selection methods with fixed sample size. The rank of the accuracy on each dataset is shown in brackets. The reduction result size is displayed in italics. The average rank indicates the overall performance of each algorithm (lower is better)). | 117 |
| 4.2 | Classification accuracy of instance selection methods with same reduction result size. The rank of the accuracy on each dataset is shown in brackets. The average rank indicates the overall performance of each algorithm (lower is better). | 119 |
| 4.3 | Parameters of concept drift handling methods for real-world datasets. | 120 |
| 4.4 | Classification accuracy of concept drift detection methods on real-world datasets with different base learners. The rank of the accuracy on each dataset is shown in brackets. The average rank indicates the overall performance of each algorithm (lower is better). | 122 |
| 5.1 | Additional parameters of comparison methods used in the experiments. | 135 |
| 5.2 | Drift detection result on 2D normal distribution data with window size 10,000. | 136 |
| 5.3 | Drift detection result on 2D normal distribution data with window size 5,000. | 137 |
| 5.4 | Drift detection result on high-dimensional normal distribution data with window size 10,000. | 138 |
| 5.5 | Drift detection result on 2D Poisson distribution with window size 10000. | 139 |
| 5.6 | Running time with different dimensions and window sizes. | 140 |

| | | |
|-----|---|-----|
| 5.7 | Temporal complexities of model construction and incremental updating. | 140 |
| 6.1 | The applicable drift scenarios of different drift detection methods . . | 148 |
| 6.2 | Error set changes for different types of drift. | 150 |
| 6.3 | Additional parameters of comparison methods used in the experiments. | 153 |
| 6.4 | Drift detection result on high-dimensional data | 158 |
| 6.5 | Running time of drift detection with different dimensions and window sizes. | 159 |

List of Abbreviations

- DRB** **D**ecision **R**egion **B**order Set pp. viii, 8
- DRS** **D**ecision **R**egion **S**upport Set pp. viii, 8
- EDE** **E**qual **D**ensity **E**stimation pp. viii, 8
- NSD** **N**eighbor **S**earch **D**iscrepancy pp. vii, 8
- PPD** **P**robability **P**ercentile **D**iscrepancy pp. viii, 9