LEARNING TO CLASSIFY TEXT USING SUPPORT VECTOR MACHINES

Methods, Theory and Algorithms

Thorsten Joachims

Kluwer Academic Publishers

2.005_494-1

2-005-494-1

LEARNING TO CLASSIFY TEXT USING SUPPORT VECTOR MACHINES

جامعة القاهرة
المكتبة المركزية
by

Thorsten Joachims
Cornell University, U.S.A.

Dissertation, Universität Dortmund
Fachbereich Informatik
February 2001

KLUWER ACADEMIC PUBLISHERS
Boston / Dordrecht / London

Contents

Foreword
Prof. Tom Mitchell and Prof. Katharina Morik xi
Preface xiii
Acknowledgments xv
Notation xvii
1. INTRODUCTION 1
1 Challenges 2
2 Goals 3
3 Overview and Structure of the Argument 4
3.1 Theory 4
3.2 Methods 5
3.3 Algorithms 6
4 Summary 6
2. TEXT CLASSIFICATION 7
1 Learning Task 7
1.1 Binary Setting 8
1.2 Multi-Class Setting 9
1.3 Multi-Label Setting 10
2 Representing Text 12
2.1 Word Level 13
2.2 Sub-Word Level 15
2.3 Multi-Word Level 15
2.4 Semantic Level 16
3 Feature Selection 16
3.1 Feature Subset Selection 17

3.2 Feature Construction 19

4 Term Weighting 20

5 Conventional Learning Methods 22

5.1 Naive Bayes Classifier 22

5.2 Rocchio Algorithm 24

5.3 k-Nearest Neighbors 25

5.4 Decision Tree Classifier 25

5.5 Other Methods 26

6 Performance Measures 27

6.1 Error Rate and Asymmetric Cost 28

6.2 Precision and Recall 29

6.3 Precision/Recall Breakeven Point and F_β-Measure 30

6.4 Micro- and Macro-Averaging 30

7 Experimental Setup 31

7.1 Test Collections 31

7.2 Design Choices 32

3. SUPPORT VECTOR MACHINES 35

1 Linear Hard-Margin SVMs 36

2 Soft-Margin SVMs 39

3 Non-Linear SVMs 41

4 Asymmetric Misclassification Cost 43

5 Other Maximum-Margin Methods 43

6 Further Work and Further Information 44

Part Theory

4. A STATISTICAL LEARNING MODEL OF TEXT CLASSIFICATION FOR SVMS 45

1 Properties of Text-Classification Tasks 46

1.1 High-Dimensional Feature Space 46

1.2 Sparse Document Vectors 47

1.3 Heterogeneous Use of Terms 47

1.4 High Level of Redundancy 48

1.5 Frequency Distribution of Words and Zipf's Law 49

2 A Discriminative Model of Text Classification 51

2.1 Step 1: Bounding the Expected Error Based on the Margin 51

2.2 Step 2: Homogeneous TCat-Concepts as a Model of Text-Classification Tasks 53

2.3 Step 3: Learnability of TCat-Concepts 59

3 Comparing the Theoretical Model with Experimental Results 64

4 Sensitivity Analysis: Difficult and Easy Learning Tasks 66

4.1 Influence of Occurrence Frequency 66

4.2 Discriminative Power of Term Sets 68

4.3 Level of Redundancy 68

5 Noisy TCat-Concepts 69

6 Limitations of the Model and Open Questions 72

7 Related Work 72

8 Summary and Conclusions 74

5. EFFICIENT PERFORMANCE ESTIMATORS FOR SVMS 75

1 Generic Performance Estimators 76

1.1 Training Error 76

1.2 Hold-Out Testing 77

1.3 Bootstrap and Jackknife 78

1.4 Cross-Validation and Leave-One-Out 79

2 $\xi\alpha$-Estimators 81

2.1 Error Rate 82

2.2 Recall, Precision, and F_1 89

3 Fast Leave-One-Out Estimation 93

4 Experiments 94

4.1 How Large are Bias and Variance of the $\xi\alpha$-Estimators? 95

4.2 What is the Influence of the Training Set Size? 99

4.3 How Large is the Efficiency Improvement for Exact Leave-One-Out? 101

5 Summary and Conclusions 102

Part Methods

6. INDUCTIVE TEXT CLASSIFICATION 103

1 Learning Task 104

2 Automatic Model and Parameter Selection 105

2.1 Leave-One-Out Estimator of the PRBEP 106

2.2 $\xi\alpha$-Estimator of the PRBEP 106

2.3 Model-Selection Algorithm 108

3 Experiments 108
3.1 Word Weighting, Stemming and Stopword Removal 108
3.2 Trading Off Training Error vs. Complexity 111
3.3 Non-Linear Classification Rules 113
3.4 Comparison with Conventional Methods 113
4 Related Work 116
5 Summary and Conclusions 117

7. TRANSDUCTIVE TEXT CLASSIFICATION 119
1 Learning Task 120
2 Transductive Support Vector Machines 121
3 What Makes TSVMs well Suited for Text Classification? 123
3.1 An Intuitive Example 123
3.2 Transductive Learning of TCat-Concepts 125
4 Experiments 127
5 Constraints on the Transductive Hyperplane 130
6 Relation to Other Approaches Using Unlabeled Data 133
6.1 Probabilistic Approaches using EM 133
6.2 Co-Training 134
6.3 Other Work on Transduction 139
7 Summary and Conclusions 139

Part Algorithms

8. TRAINING INDUCTIVE SUPPORT VECTOR MACHINES 141
1 Problem and Approach 142
2 General Decomposition Algorithm 143
3 Selecting a Good Working Set 145
3.1 Convergence 145
3.2 How to Compute the Working Set 146
4 Shrinking: Reducing the Number of Variables 146
5 Efficient Implementation 148
5.1 Termination Criteria 148
5.2 Computing the Gradient and the Termination Criteria Efficiently 149
5.3 What are the Computational Resources Needed in each Iteration? 150
5.4 Caching Kernel Evaluations 151

5.5 How to Solve the QP on the Working Set 152

6 Related Work 152

7 Experiments 154

7.1 Training Times for Reuters, WebKB, and Ohsumed 154

7.2 How does Training Time Scale with the Number of Training Examples? 154

7.3 What is the Influence of the Working-Set-Selection Strategy? 160

7.4 What is the Influence of Caching? 161

7.5 What is the Influence of Shrinking? 161

8 Summary and Conclusions 162

9. TRAINING TRANSDUCTIVE SUPPORT VECTOR MACHINES 163

1 Problem and Approach 163

2 The TSVM Algorithm 165

3 Analysis of the Algorithm 166

3.1 How does the Algorithm work? 166

3.2 Convergence 168

4 Experiments 169

4.1 Does the Algorithm Effectively Maximize Margin? 169

4.2 Training Times for Reuters, WebKB, and Ohsumed 170

4.3 How does Training Time Scale with the Number of Training Examples? 170

4.4 How does Training Time Scale with the Number of Test Examples? 172

5 Related Work 172

6 Summary and Conclusions 174

10. CONCLUSIONS 175

1 Open Question 177

Bibliography 180

Appendices 197

SVM-Light Commands and Options 197

Index 203