

Copyright is owned by the Author of the thesis. Permission is given for a copy to be downloaded by an individual for the purpose of research and private study only. The thesis may not be reproduced elsewhere without the permission of the Author.

ACCELERATING CLASSIFIER TRAINING USING  
ADABOOST WITHIN CASCADES OF BOOSTED  
ENSEMBLES

A thesis presented in partial  
fulfillment of the requirements

for the degree of

Master of Science in Computer Sciences

at Massey University, Auckland, New Zealand

Teo Susnjak

2009



# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Brief Introduction to Machine Learning . . . . .	2
1.2	Research Scope and Objectives . . . . .	5
1.3	Contributions . . . . .	6
1.4	Structure of the Thesis . . . . .	6
<b>2</b>	<b>Classification Theory</b>	<b>9</b>
2.1	Features . . . . .	9
2.1.1	Feature Types . . . . .	11
2.1.2	Feature Extraction . . . . .	12
2.1.3	Feature Selection . . . . .	13
2.2	Boosting . . . . .	13
2.2.1	Discrete AdaBoost . . . . .	14
2.2.2	Variant Boosting Algorithms . . . . .	16
2.2.3	Weak Learner . . . . .	17
2.2.4	Strong Classifier Type . . . . .	18
2.3	Classifier Training Structures . . . . .	19
2.3.1	Monolithic Ensemble . . . . .	21
2.3.2	Cascaded . . . . .	21
2.3.3	Parallel . . . . .	24
2.3.4	Tree . . . . .	25
2.3.5	Soft Cascades . . . . .	27
2.4	Training Procedures . . . . .	27
2.5	Summary . . . . .	29

<b>3</b>	<b>Training Classifiers</b>	<b>31</b>
3.1	Feature Space Size . . . . .	32
3.2	Cascade Threshold Adjustments . . . . .	33
3.3	Slow Learning Convergence . . . . .	34
3.4	Cascade Optimization . . . . .	36
3.5	Catastrophic Interference . . . . .	38
3.6	Discussion . . . . .	39
<b>4</b>	<b>Accelerated Classifier Training</b>	<b>41</b>
4.1	The PSL Structure . . . . .	41
4.1.1	PSL Discussion . . . . .	45
4.2	Variant PSL Structure . . . . .	47
4.2.1	Variant PSL Discussion . . . . .	49
4.3	Incremental Learning with IPSL . . . . .	49
4.3.1	IPSL Discussion . . . . .	52
4.4	Summary . . . . .	53
<b>5</b>	<b>Evaluating the PSL Structure</b>	<b>55</b>
5.1	Comparative Experiments . . . . .	56
5.2	Technical Specifications . . . . .	57
5.3	Datasets . . . . .	58
5.3.1	UCI PenDigit and OptDigit Datasets . . . . .	58
5.3.2	UCI Letter Dataset . . . . .	59
5.4	Training Procedures . . . . .	60
5.5	Performance and Accuracy Testing Methodology . . . . .	64
5.5.1	ROC Graphs . . . . .	64
5.5.2	Classifier Training Convergence . . . . .	66
5.5.3	Simple Classification Accuracy . . . . .	67
5.5.4	Performance Runtimes . . . . .	67
<b>6</b>	<b>Experimental Results</b>	<b>69</b>
6.1	Preliminary PSL Experimental Results . . . . .	69
6.1.1	Training Phase Analysis . . . . .	70

CONTENTS	V
6.1.2 Hit Rate . . . . .	73
6.1.3 False Detection Rate . . . . .	74
6.1.4 Total Error Rate . . . . .	75
6.1.5 Runtime Performance . . . . .	76
6.2 Further PSL Experimental Results . . . . .	77
6.2.1 Training Phase Analysis . . . . .	78
6.2.2 Hit Rate . . . . .	80
6.2.3 False Detection Rate . . . . .	81
6.2.4 Total Error Rate . . . . .	83
6.2.5 Runtime Performance . . . . .	83
6.3 Results of the Variant PSL Structure . . . . .	84
6.3.1 Training Phase Analysis . . . . .	84
6.3.2 Hit Rate . . . . .	86
6.3.3 False Detection Rate . . . . .	87
6.3.4 Total Error Rate . . . . .	88
6.3.5 Runtime Performance . . . . .	88
6.4 Summary . . . . .	89
<b>7 Incremental Learning Results</b>	<b>91</b>
7.1 IPSL Incremental Learning Analysis . . . . .	91
7.1.1 Training Phase Analysis . . . . .	93
7.1.2 Hit Rate . . . . .	96
7.1.3 False Detection Rate . . . . .	97
7.1.4 Total Error Rate . . . . .	100
7.1.5 Runtime Performance . . . . .	100
7.2 Discussion . . . . .	101
<b>8 Conclusion</b>	<b>107</b>
8.1 Future Work . . . . .	109
<b>Appendices</b>	<b>113</b>
<b>A ROC Graph Curves For IPSL on UCI Letter Dataset</b>	<b>113</b>

<b>Bibliography</b>	<b>117</b>
<b>Glossary</b>	<b>122</b>

# List of Figures

2.1	Cascaded classifier in the form proposed by Viola and Jones (2001 <i>b</i> ). . . .	22
2.2	The structure for parallel cascaded classifiers experimented with by Lienhart, Liang and Kuranov (2003). . . . .	25
2.3	A detector tree of boosted classifiers proposed by Lienhart, Liang and Kuranov (2003). . . . .	26
4.1	Structure illustrating parallel strong classifiers within the same layer (PSL).	42
4.2	The flow of a test sample through the PSL structure during the detection phase illustrating the process of a positive detection and a rejection. . . .	44
4.3	The flow of positive training samples through the PSL structure during the training phase. . . . .	46
4.4	The flow of negative training samples through the PSL structure during the training phase. . . . .	46
4.5	The modified flow of the negative training samples through the variant form of the PSL structure during the training phase. . . . .	48
4.6	The incremental PSL (IPSL) training structure illustrating the creation of additional nodes to the existing <i>base classifier</i> . . . . .	50
5.1	Example of ROC graph curve generalization with three different performances. ( <a href="http://gim.unmc.edu/dxtests/ROC3.htm">http://gim.unmc.edu/dxtests/ROC3.htm</a> ) . . . . .	66
6.1	ROC graph curves for the PenDigit datasets showing the convergence of the standard cascaded and the PSL classifiers on the test dataset. a) cascaded classifiers 0 - 4. b) PSL classifiers 0 - 4. c) cascaded classifiers 5 - 9. d) PSL classifiers 5 - 9. . . . .	73



6.2	ROC graph curves for the OptDigit datasets showing the convergence of the standard cascaded and the PSL classifiers on the test dataset. a) cascaded classifiers 0 - 4. b) PSL classifiers 0 - 4. c) cascaded classifiers 5 - 9. d) PSL classifiers 5 - 9. . . . .	74
6.3	Hit rate results of the monolithic, standard cascaded and PSL classifiers prior to their optimization. a) PenDigit dataset b) OptDigit dataset. . . .	75
6.4	Hit rate results of the monolithic, and optimized standard cascaded and PSL classifiers. a) PenDigit dataset b) OptDigit dataset . . . . .	75
6.5	False detection rate results of the monolithic, standard cascaded and PSL classifiers prior to their optimization. a) PenDigit dataset b) OptDigit dataset	76
6.6	False detection rate results of the monolithic, and optimized standard cascaded and PSL classifiers. a) PenDigit dataset b) OptDigit dataset . . . .	76
6.7	Total error rates of the monolithic, standard cascaded and PSL classifiers prior to their optimization. a) PenDigit dataset b) OptDigit dataset . . .	77
6.8	Total error rates of the monolithic, and optimized standard cascaded and PSL classifiers. a) PenDigit dataset b) OptDigit dataset . . . . .	77
6.9	ROC graph curves for the Letters dataset showing the convergence of the cascaded and the PSL classifiers on the test sets. a) cascaded classifiers A-I. b) PSL classifiers A-I. c) cascaded classifiers J-R. d) PSL classifiers J-R. e) cascaded classifiers S-Z. f) PSL classifiers S-Z. . . . .	79
6.10	Optimized hit rates for the UCI Letter dataset comparing the monolithic, standard cascaded and the PSL classifiers. . . . .	82
6.11	Optimized false detection rates for the UCI Letter dataset comparing the monolithic, standard cascaded and the PSL classifiers. . . . .	82
6.12	Optimized total error rates for the UCI Letter dataset comparing the monolithic, standard cascaded and the PSL classifiers. . . . .	83
6.13	ROC graph curves for the UCI Letter dataset comparing the convergence patterns of the PSL and the variant PSL classifiers on a test set. a) classifiers A-I. b) classifiers J-R. c) classifiers S-Z. . . . .	85
6.14	Optimized hit rates for the UCI Letter dataset comparing the PSL and the variant PSL classifiers. . . . .	87

6.15	Optimized false detection rates for the UCI Letter dataset comparing the PSL and the variant PSL classifiers. . . . .	88
6.16	Optimized total error rates for the UCI Letter dataset comparing the PSL and the variant PSL classifiers. . . . .	89
7.1	ROC graph curves for the Letters dataset showing the generalization of the IPSL classifiers on a test dataset, comparing the base classifier against the 10%, 20% and 30% increment classifiers. a) Class L, showing an example of a IPSL generalization affected by higher false detection rates and slightly improved hit rates. b) Class M displaying a successful example of IPSL classifiers improving the generalization of the base classifier with both false positive and true positive detections. . . . .	92
7.2	Total number of weak classifiers generates per base and IPSL classifier for each incremental dataset on the UCI Letter dataset. . . . .	94
7.3	Optimized hit rates for the UCI Letter dataset comparing the base classifier and the classifiers for dataset increments of 10% - 30%. . . . .	97
7.4	Optimized hit rates for the UCI Letter dataset comparing the standard cascaded and the IPSL classifiers. a) 10% increment dataset b) 20% increment dataset c) 30% increment dataset. . . . .	98
7.5	Optimized false detection rates for the UCI Letter dataset comparing the standard cascaded and the IPSL classifiers. a) 10% increment dataset b) 20% increment dataset c) 30% increment dataset. . . . .	99
7.6	Optimized false detection rates for the UCI Letter dataset comparing the base classifier and the classifiers for dataset increments of 10% - 30%. . .	100
7.7	Optimized total error rates for the UCI Letter dataset comparing the base classifier and the classifiers for dataset increments of 10% - 30%. . . . .	101
7.8	Optimized total error rates for the UCI Letter dataset comparing the standard cascaded and the IPSL classifiers. a) 10% increment dataset b) 20% increment dataset c) 30% increment dataset. . . . .	102
7.9	Total number of weak classifiers per classifier showing comparisons between counts of the standard cascaded and IPSL classifiers. a) 10% increment dataset b) 20% increment dataset c) 30% increment dataset. . . . .	103

A.1	ROC graph curves for the Letters dataset showing convergences of IPSL classifiers representing letters A-B. . . . .	113
A.2	ROC graph curves for the Letters dataset showing convergences of IPSL classifiers representing letters C-J. . . . .	114
A.3	ROC graph curves for the Letters dataset showing convergences of IPSL classifiers representing letters K-R. . . . .	115
A.4	ROC graph curves for the Letters dataset showing convergences of IPSL classifiers representing letters S-Z. . . . .	116

# List of Tables

5.1	General details of UCI OptDigit, UCI PenDigit and UCI Letter datasets used for the experiments. . . . .	59
5.2	Details of the UCI OptDigit and PenDigit datasets featuring characteristics per class label. . . . .	59
5.3	Details of the UCI Letter dataset featuring characteristics per class label.	60
5.4	Size of the training sets for the UCI Letter dataset used in incremental learning. . . . .	61
5.5	The settings of tunable parameters for the training of the four experiments.	63
6.1	Total number of weak classifiers comprising each digit classifier for the PenDigit and OptDigit datasets comparing the single layered monolithic, standard cascaded and the PSL classifiers. The totals include weak classifiers for all the nodes in the PSL classifiers. . . . .	70
6.2	Total number of layers comprising each classifier for every digit in the Opt-Digit dataset. Total layer numbers required to train a classifier are shown with adjacent columns demonstrating the total number of layers per classifier after the optimization process is performed. . . . .	71
6.3	Total number of layers comprising each classifier for every digit in the PenDigit dataset. Total layer numbers required to train a classifier are shown with adjacent columns demonstrating the total number of layers per classifier after the optimization process is performed. . . . .	71
6.4	Training runtimes (seconds) of the monolithic, standard cascaded and the PSL classifiers on the PenDigit and OptDigit datasets. . . . .	72

6.5	Total number of weak classifiers required to train classifiers for each of the UCI Letter character classes for the monolithic, standard cascaded, PSL and the variant PSL classifiers. The totals include weak classifiers for all the nodes in the PSL classifiers. . . . .	80
6.6	a) Training runtimes (seconds) of classifiers for each of the UCI Letter character classes for the monolithic, standard cascaded, PSL and the variant PSL classifiers. . . . .	81
6.7	Total number of layers comprising the classifiers for the UCI Letter dataset. The figures show totals for the optimized PSL and the variant PSL classifiers.	86
7.1	Total number of layers generated per cascaded and IPSL classifiers for the UCI Letter dataset. Figures show layer totals for the unoptimized base classifiers as well as the final layer numbers after the optimization is applied.	95
7.2	Training runtimes for each classifier class on the UCI Letter dataset comparing the standard cascaded and the IPSL classifiers of 10%, 20% and 30% size increments. . . . .	96

# List of Notations

$\alpha_t$	Factor associated to the error of $h_t$ .....	15
$d$	Number of dimensions representing a sample instance ....	2
$D_n$	The training dataset consisting of $n$ samples .....	2
$D_t(i)$	Weight of sample $i$ at round $t$ .....	15
$f(x)$	Classification function applied on a sample $x$ .....	3
$F$	False detection rate .....	42
$H$	Strong hypothesis - strong classifier function .....	15
$h_t$	Weak hypothesis - weak classifier at the $t^{\text{th}}$ boosting round	15
$i^{\text{th}}$	Position of an element in a sequence .....	2
$m$	Number of intervals used to discretize an input domain. .	16
$N$	Negative samples .....	42
$P$	Positive samples .....	42
$\mathbb{R}$	Real numbers .....	2
$T$	Total number of boosting rounds .....	15
$\mathcal{V}(f(h_t), x, y)$	Multi-class, confidence vote resolution function .....	19
$x_i$	Feature array .....	2
$y_i$	Class label array .....	2

## Abstract

This thesis seeks to address current problems encountered when training classifiers within the framework of cascades of boosted ensembles (CoBE). At present, a significant challenge facing this framework are inordinate classifier training runtimes. In some cases, it can take days or weeks (Viola and Jones, 2004; Verschae et al., 2008) to train a classifier. The protracted training runtimes are an obstacle to the wider use of this framework (Brubaker et al., 2006). They also hinder the process of producing effective object detection applications and make the testing of new theories and algorithms, as well as verifications of others research, a considerable challenge (McCane and Novins, 2003).

An additional shortcoming of the CoBE framework is its limited ability to train classifiers incrementally. Presently, the most reliable method of integrating new dataset information into an existing classifier, is to re-train a classifier from beginning using the combined new and old datasets. This process is inefficient. It lacks scalability and discards valuable information learned in previous training.

To deal with these challenges, this thesis extends on the research by Barczak et al. (2008), and presents alternative CoBE frameworks for training classifiers. The alternative frameworks reduce training runtimes by an order of magnitude over common CoBE frameworks and introduce additional tractability to the process. They achieve this, while preserving the generalization ability of their classifiers.

This research also introduces a new framework for incrementally training CoBE classifiers and shows how this can be done without re-training classifiers from beginning. However, the incremental framework for CoBEs has some limitations. Although it is able to improve the positive detection rates of existing classifiers, currently it is unable to lower their false detection rates.





## **Acknowledgements**

I would like to thank my supervisor Dr. Andre Barczak for his invaluable input, encouragement and patience over the last year. Thanks also to my colleagues at the Massey University Computer Science Department who helped me with technical issues and provided a fun and stimulating environment to work in. Thanks to the Computer Science faculty members at Massey University for always being encouraging and generous with their feedback and assistance.

Thanks to my friends for still being there despite not seeing much of me. My gratitude goes to Brian and Beverly for the genuine care and strength they have given me and also to my dear parents who have always been unwavering in their support. Thanks to Sarah for being so understanding and for providing me with much motivation during the last year.