

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.

EFFICIENT BOOSTED ENSEMBLE-BASED MACHINE  
LEARNING IN THE CONTEXT OF CASCADED  
FRAMEWORKS

A thesis presented in partial  
fulfilment of the requirements

for the degree of

Doctor of Philosophy in Computer Science

at Massey University,  
Auckland, New Zealand

Teo Sušnjak

2012



## Abstract

The ability to both efficiently train robust classifiers and to design them for fast detection is an important goal of machine learning. With an ever increasing amount of available data being generated, the task of expeditiously producing real-time capable classifiers is becoming more challenging. In the context of the increasing complexity of the task, ensemble-based learning methods have proven themselves to be effective approaches for satisfying these requirements.

Ensemble methods produce a number of weak models that are strategically combined into a single classifier. They have been particularly effective when combined with boosting algorithms and strategies that structure the ensembles into cascades. The strength of cascaded-ensembles lies in the separate-and-conquer approach they employ during the training of each layer. Class decision-boundaries for trivial cases are learned in the early rounds, while more difficult decision boundaries are refined with each succeeding layer. In a two-class problem domain, non-target instances learned in initial layers are removed and replaced by more complex samples, frequently referred to as bootstrapping. With this procedure, efficient coarse-to-fine learning is accomplished.

The contribution of this thesis lies in three main areas that centre around the concept of improving the efficiency in the training and execution process. The first explored ways in which the conventional ensemble-cascades could be combined with an even more aggressive separate-and-conquer strategy that further partitions the ensemble inside each layer. The focus was on the two-class learning problem and used face detection as the medium to observe the trade-offs involved concerning both the accuracy and the efficiency of the resulting classifiers. The algorithm was further developed in a way that enabled the bootstrapping of positive samples within a cascade, alongside the conventional approach that bootstraps only the negative samples. Secondly, the negative effect of dynamic environments on static classifiers on binary class problems was considered. A method was developed which enabled the cascaded classifiers to efficiently adapt to the changing environment on domains with high volume streaming data. This environment was simulated using face detection as well. Lastly, the open problem of creating integrated multiclass cascades was researched and an algorithm was devised.

Overall, the findings have shown that invariably a trade-off is incurred between reduced training runtimes resulting from aggressive separate-and-conquer strategies and the accuracy of the final classifiers. Using the CMU MIT test dataset, the experiments showed that though the proposed positive sample bootstrapping component succeeded in significantly reducing the training runtimes without compromising the accuracy, the general decomposition strategy did lower the accuracy when compared to the benchmark Viola-Jones classifiers. The proposed adaptive cascade learning algorithm for drifting concepts was also evaluated on a face detection problem set. The results demonstrated its ability to effectively adapt to dynamic environments in high speed data streams without requiring explicit re-training of the individual classifiers. The multiclass cascaded algorithm was compared to three existing algorithms on 18 UCI datasets. It was found to be, on average, several times faster to train and to execute, while generating comparable accuracy rates. The algorithm exhibited scalability to large datasets but was found to be susceptible to producing overly complex classifiers on datasets with a large number of class labels.



## Acknowledgements

My thanks to the Institute of Information and Mathematical Sciences at Massey University for enabling me to grow over the years in my academic pursuits and for fostering an environment which allowed this to take place. Thanks in particular to the faculty at the Department of Computer Science. You have inspired, encouraged and taught me not just knowledge, but the value of it. My gratitude goes to all the administration staff at the institute for their help and assistance over the years, as well as to all my colleagues from whom I have learned much.

I am grateful to the Tertiary Education Commission for providing me with a generous doctorate scholarship. I express my thanks also to the Ministry of Science and Innovation for their internship grant that enabled me to implement research and expertise arising from this research into the industrial setting. A special thanks to Compac Sorting Ltd. for their support in the final year and for providing me with an opportunity to apply my research to solving real-world problems.

I would like to extend my thanks to the members of the doctoral defence committee, Prof. Pitoyo Hartono, Prof. Alvis Fong and Assoc. Prof. Chris Scogings for their time as well as valuable input and engagement with me in a fruitful discussion. Also a special thanks to Prof. Anne de Bruin for her role as convenor of the examination.

Thanks to my friends and family for your support over the years and a special thanks to a few of you who have reviewed the initial manuscript. I look forward to now seeing and spending more time with all of you. Christoph Schumacher, without your encouragement I do not think I would have embarked on this path. Thank you for your friendship and continuous support.

Brian and Beverley for never doubting, but always believing. Expressing my gratitude to you both seems so inadequate. To my mum and dad who have unceasingly encouraged and urged me on from distant shores, thank you; ovo je i vaš uspjeh, hvala vam za sve.

Ken Hawick, my heartfelt gratitude to you for being my co-supervisor. You have been a source of wisdom, advice and encouragement during many phases of this research. To my supervisor Andre Barczak, I am so indebted to you. You have been my teacher, my mentor and have also become a close friend. I look forward to our conversations over coffee continuing for many years to come.

To my beautiful wife Sarah; I love you. You have been my rock every step of the way.

**Teo Sušnjak**

Auckland, New Zealand

October, 2012



# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Objectives . . . . .	3
1.2	Contributions . . . . .	4
<b>2</b>	<b>Ensemble-Based Machine Learning Theory</b>	<b>7</b>
2.1	Theory . . . . .	7
2.1.1	Ensemble Diversity . . . . .	9
2.1.2	Combining Ensemble Outputs . . . . .	10
2.1.3	Optimal Ensemble Training Procedures . . . . .	12
2.2	Boosting . . . . .	14
2.2.1	Binary Class Problems . . . . .	14
2.2.2	Multiclass Problems . . . . .	16
2.3	Emerging Challenges and Coarse-to-fine Learning . . . . .	21
2.3.1	Cascades of Ensembles . . . . .	22
2.3.2	Ensembles of Nested Dichotomies . . . . .	27
2.4	Summary . . . . .	28
<b>3</b>	<b>Methodology</b>	<b>29</b>
3.1	Face Detection Datasets and Feature Types . . . . .	29
3.2	Multiclass Classification Datasets . . . . .	33
3.3	Weak Learner . . . . .	33
3.4	Training Procedures . . . . .	34
3.5	Classifier Evaluation Methods . . . . .	35
<b>4</b>	<b>Applications of PSL to Face Detection</b>	<b>39</b>
4.1	Motivation . . . . .	39
4.2	Related Work . . . . .	40
4.3	The PSL Algorithm . . . . .	43
4.4	Experimental Setup . . . . .	45
4.5	Results . . . . .	46
4.6	Discussion . . . . .	52
4.7	Summary . . . . .	54
<b>5</b>	<b>PSL with Positive Sample Bootstrapping</b>	<b>57</b>
5.1	Motivation . . . . .	57
5.2	Related Work . . . . .	58



5.3	PSL with Positive Sample Bootstrapping . . . . .	58
5.3.1	Experimental Setup . . . . .	60
5.3.2	Results . . . . .	62
5.4	Methods for Improving the Accuracy . . . . .	69
5.5	Discussion . . . . .	79
5.6	Summary . . . . .	80
<b>6</b>	<b>Adaptive Cascade Classifiers</b>	<b>83</b>
6.0.1	Motivation . . . . .	84
6.0.2	Past Research . . . . .	85
6.1	Implementation Details . . . . .	86
6.1.1	Assigning Competence Values to the Static PSL-classifier . . . . .	86
6.1.2	Concept-Drift Learning Algorithm . . . . .	89
6.2	Experiment Design . . . . .	90
6.3	Results . . . . .	92
6.3.1	Analysis of Multi-Frame Concept-Drift Learning . . . . .	93
6.3.2	Results of Single-Frame Concept-Drift Learning . . . . .	94
6.3.3	Response Patterns to Gradual and Abrupt Drifts . . . . .	96
6.3.4	Learning and Detection Runtimes . . . . .	96
6.4	Discussion . . . . .	97
6.5	Summary . . . . .	99
<b>7</b>	<b>Multiclass PSL</b>	<b>101</b>
7.1	Motivation . . . . .	102
7.2	PSL Multiclass Learning Framework . . . . .	103
7.2.1	Multiclass Cascade . . . . .	103
7.2.2	Multiclass Weak Learner . . . . .	107
7.3	Experiment Design . . . . .	109
7.3.1	Multiclass Cascade Implementation . . . . .	112
7.4	Results . . . . .	113
7.4.1	Training Phase . . . . .	113
7.4.2	Generalization . . . . .	121
7.4.3	Runtime Phase . . . . .	128
7.5	Discussion and General Evaluation . . . . .	132
7.6	Summary . . . . .	136
<b>8</b>	<b>Converting Monolithic Multiclass Methods to Cascades</b>	<b>139</b>
8.1	Motivation . . . . .	139
8.2	Converting Single-Layered Multiclass Algorithms to Cascades . . . . .	140
8.3	Experiment design . . . . .	141
8.4	Results . . . . .	141
8.5	Discussion . . . . .	150
8.6	Summary . . . . .	152

<b>9 Conclusion</b>	<b>153</b>
9.1 Recommendations . . . . .	156
9.2 Future Work . . . . .	157
<b>Appendices</b>	<b>161</b>
<b>A Preliminary Experimental Results</b>	<b>161</b>
A.1 Supplementary Results . . . . .	161
<b>B Additional BPSL Graphs and Detection-Runtime Experiments</b>	<b>163</b>
B.1 ROC Graphs for BPSL Classifiers on 5000 Sample Dataset . . . . .	163
B.2 Execution Runtimes with Variable Values of $\Phi$ . . . . .	163
<b>C Supplementary Multiclass Algorithm Results</b>	<b>167</b>
<b>D Complete Graphs and Tables from the Results of Cascadizing OC, ECC and M2</b>	<b>173</b>
<b>Bibliography</b>	<b>179</b>



# List of Figures

2.1	Viola-Jones cascade. . . . .	24
2.2	Parallel cascades. . . . .	25
2.3	A detector tree of boosted classifiers. . . . .	27
3.1	Haar-like feature set. . . . .	31
4.1	The PSL cascade structure . . . . .	42
4.2	Training runtimes in seconds for all classifiers on the four CMU MIT datasets. . . . .	47
4.3	Rate of learning positive samples by PSL classifiers ( $\Phi = 10$ ). . . . .	48
4.4	Typical PSL convergence patterns for FPR where $\Phi = 10$ . . . . .	49
4.5	Weak classifiers totals for each classifier. . . . .	50
4.6	ROC graph on the CMU MIT datasets . . . . .	51
4.7	Total error rate on the CMU MIT test set, as a function of the training runtime. . . . .	52
4.8	Execution runtimes for PSL and Viola-Jones classifiers on the CMU MIT test datasets. . . . .	53
5.1	The propagation of the positive samples in the BPSL bootstrapping method. . . . .	60
5.2	Example of the positive dataset instances. . . . .	61
5.3	Training runtimes for PSL, BPSL and VJ classifiers . . . . .	62
5.4	Convergence of positive samples at training. . . . .	63
5.5	Mean ROC curve with the standard deviations for a BPSL classifier . . . . .	64
5.6	Classifier ROC graph curves on the CMU MIT dataset. . . . .	65
5.7	Cost/benefit trade-off between test error rates and training runtimes of VJ and PSL . . . . .	66
5.8	PSL node analysis. . . . .	67
5.9	Examples of images learned at various nodes. . . . .	68
5.10	ROC curves for PSL and BPSL classifiers with thinning. . . . .	72
5.11	ROC curves for BPSL.r, BPSL and PSL classifiers where $\Phi = 10$ . . . . .	75
5.12	ROC curves for BPSL.r, BPSL and PSL classifiers where $\Phi = 20$ . . . . .	76
5.13	ROC curves for BPSL.r, BPSL and PSL classifiers where $\Phi = 15$ . . . . .	77
5.14	Averaged training runtimes for each of the BPSL.r, BPSL and PSL classifiers . . . . .	78
6.1	Diagram of the concept-drift learning algorithm. . . . .	88
6.2	Example of test images in a dynamic environment . . . . .	91
6.3	Total false positive detections per layer . . . . .	93

6.4	Execution runtimes for all classifiers . . . . .	94
6.5	Total number of false positive detections per frame . . . . .	95
6.6	Example of an image sequence that triggered a concept-drift learning phase . . . . .	96
6.7	Comparison of adaptation-phase durations . . . . .	97
7.1	Cascaded multiclass framework. . . . .	104
7.2	An example of the proposed multiclass weak learner selecting the best features on the first 8 boosting iterations of the Pendigit dataset . . . . .	110
7.3	Training error convergence graphs for the Pendigit, Letter, Optdigits and Robot Navigation datasets. . . . .	114
7.4	Training error convergence graphs for the Segmentation, Vehicle, Iris and Fourier datasets. . . . .	115
7.5	The effect on training runtimes with the increase in $\Phi$ on the cascade multiclass algorithm. . . . .	118
7.6	Classifier accuracy on test data as a function of training runtime. . . . .	119
7.7	Test error convergence graphs. . . . .	121
7.8	Multiclass cascade classifier execution runtimes from a selection of datasets. . . . .	131
8.1	Training error convergence graphs. . . . .	143
8.2	Classifier accuracy on test data as a function of training runtime. . . . .	144
A.1	Node accuracy analysis post re-sampling procedure . . . . .	161
A.2	Typical convergence patterns of the false alarm rates for PSL classifiers . . . . .	162
B.1	Classifier ROC graph curves comparing PSL, BPSL and VJ classifiers on the CMU MIT dataset. . . . .	164
B.2	ROC curves for BPSL.r, BPSL and PSL classifiers. . . . .	165
C.1	Training error convergence graphs. . . . .	168
C.2	Classifier accuracy on test data as a function of training runtime. . . . .	169
C.3	Test error convergence graphs. . . . .	170
D.1	Training error convergence graphs for cascaded classifiers. . . . .	174
D.2	Classifier accuracy on test data as a function of training runtime for cascaded classifiers. . . . .	175
D.3	Test error convergence graphs for cascaded classifiers. . . . .	176

# List of Tables

2.1	Example of a coding matrix for a four class problem. . . . .	17
3.1	Training dataset details with properties for describing the <i>static</i> classifier. . . . .	32
3.2	Properties of the multiclass datasets. . . . .	33
4.1	Training settings and dataset details. . . . .	46
5.1	Training settings and dataset details. . . . .	61
5.2	Comparison between the training runtimes of VJ, PSL, BPSL . . . . .	63
5.3	Cost associated with the re-sampling procedure . . . . .	79
5.4	Total weak classifiers generated by each of the training structures . . . . .	79
6.1	Dataset and <i>static</i> classifier training properties. . . . .	90
6.2	Characteristics of the concept-drift learning dataset and the method. . . . .	91
7.1	Classifier training runtime results. . . . .	117
	(a) Cascaded.DP $\Phi = 5$ training runtime comparison. . . . .	117
	(b) Cascaded.DP $\Phi = 10$ training runtime comparison. . . . .	117
	(c) Cascaded.DP $\Phi = 25$ training runtime comparison. . . . .	117
	(d) Cascaded.DP $\Phi = 50$ training runtime comparison. . . . .	117
7.2	Classifier accuracy results on datasets with uniform class distributions . . . . .	123
7.3	Statistical results of the Friedman and Iman-Davenport tests . . . . .	124
7.4	Classifier accuracy results on datasets with skewed class-distributions . . . . .	126
7.5	F-value results for each class on the Yeast dataset. . . . .	127
7.6	F-value results for each class on the Glass dataset. . . . .	127
7.7	F-value results for each class on the Page Blocks dataset. . . . .	127
7.8	Statistical test results for classifier accuracies on skewed-distribution datasets. . . . .	128
7.9	Multiclass execution runtimes. . . . .	129
	(a) Cascaded.DP $\Phi = 5$ execution runtime comparison. . . . .	129
	(b) Cascaded.DP $\Phi = 10$ execution runtime comparison. . . . .	129
	(c) Cascaded.DP $\Phi = 25$ execution runtime comparison. . . . .	129
	(d) Cascaded.DP $\Phi = 50$ execution runtime comparison. . . . .	129
7.10	The total numbers of weak classifiers per classifier across all datasets . . . . .	130
	(a) Datasets: Letter - Iris . . . . .	130
	(b) Datasets: Factors - Shuttle . . . . .	130
8.1	Results of training runtimes for all classifiers . . . . .	142
8.2	Classifier accuracy results on datasets with uniform class distributions . . . . .	145

8.3	Statistical analysis of the Friedman, Iman-Davenport and Nemenyi tests . . . . .	146
8.4	Wilcoxon signed-ranks test for results on datasets with balanced class-distributions . . . . .	147
8.5	Classifier accuracy on datasets with biased class-distributions. . . . .	148
8.6	F-Values for the Yeast dataset . . . . .	149
8.7	F-Values for the Satimage dataset . . . . .	149
8.8	Statistical results of the Friedman and Iman-Davenport and Nemenyi tests	149
8.9	Wilcoxon signed-ranks test for results from datasets with biased class-distributions . . . . .	150
8.10	Classifier detection runtime results on all datasets . . . . .	151
B.1	The configuration of the $\Phi$ value per layer of the cascade for the BPSL.r classifier. . . . .	163
B.2	Detection runtime comparison between flexible and fixed $\Phi$ BPSL.r classifiers	164
C.1	F-value accuracy results for the shuttle dataset. . . . .	172
C.2	F-value accuracy results for the robot navigation dataset. . . . .	172
C.3	F-value accuracy results for the satimage dataset. . . . .	172
D.1	F-Values for the Page Blocks dataset for cascaded classifiers . . . . .	177
D.2	F-Values for the Shuttle dataset for cascaded classifiers . . . . .	177
D.3	F-Values for the Glass dataset for cascaded classifiers . . . . .	177
D.4	F-Values for the Robot Navigation dataset for cascaded classifiers . . . . .	178

# List of Algorithms

1	AdaBoost . . . . .	15
2	AdaBoost.M2 . . . . .	18
3	AdaBoost.OC . . . . .	19
4	AdaBoost.ECC . . . . .	20
5	Viola-Jones Cascade . . . . .	23
6	PSL Algorithm . . . . .	44
7	BPSL . . . . .	59
8	BPSL with Re-sampling (BPSL.r) . . . . .	74
9	Concept Drift Learning . . . . .	87
10	PSL Multiclass Cascade . . . . .	105
11	Domain-partitioning Weak Learner . . . . .	108





# List of Acronyms

BPSL	Bootstrapped PSL
BPSL.r	Bootstrapped PSL with re-sampling
CMU MIT	Carnegie Mellon University - Massachusetts Institute of Technology (Face detection dataset)
CTF	Coarse-To-Fine learning
DP	Domain Partitioning (Multiclass weak learner)
ECC	AdaBoost.ECC using Error Correcting Codes
ECOC	Error Correcting Output Codes
M2	Adaboost.M2
MCS	Multiple Classifier Systems
OAA	One-Against-All
OAo	One-Against-One
OC	AdaBoost.OC using Output Coding
PSL	Parallel Strong classifier within the same Layer algorithm
RIPPER	Repeated Incremental Pruning to Produce Error Reduction algorithm
ROC	Receiver Operating Curve
UCI	University of California Irvine (Machine learning dataset repository)
VJ	Viola-Jones (Ensemble-cascade algorithm)

