HIERACHICAL ACTIVE LEARNING BIOINFORMATICS APPLICATION

by

James D. Duin

A THESIS

Presented to the Faculty of

The Graduate College at the University of Nebraska

In Partial Fulfilment of Requirements

For the Degree of Master of Science

Major: Computer Science

Under the Supervision of Professor Stephen Scott

Lincoln, Nebraska

February, 2017

HIERACHICAL ACTIVE LEARNING BIOINFORMATICS APPLICATION

James D. Duin, M.S.

University of Nebraska, 2017

Adviser: Stephen Scott

Many classification tasks target high-level concepts that can be decomposed into a hierarchy of finer-grained sub- concepts. For example, some string entities that are Locations are also Attractions, some Attractions are Museums, etc. Such hierarchies are common in named entity recognition (NER), document classification, and biological sequence analysis. We present a new approach for learning hierarchically decomposable concepts. The approach learns a high-level classifier (e.g., location vs. non-location) by seperately learning multiple finer-grained classifiers (e.g., museum vs. non-museum), and then combining the results. Soliciting labels at a finer level of granularity than that of the target concept is a new approach to active learning, which we term active over-labeling. In experiments in NER and document classification tasks, we show that active over- labeling substantially improves area under the precision-recall curve when compared with standard passive or active learning. Finally, because finer-grained labels may be more expensive to obtain, we also present a cost-sensitive active learner that uses a multi-armed bandit approach to dynamically choose the label granularity to target, and show that the bandit-based learner is robust to differences in label cost and labeling budget.

DEDICATION

This thesis is dedicated to my parents Paul and Vicki Duin and fiancee Anna Spady.

ACKNOWLEDGMENTS

I would like to thank my advisor Dr. Stephen Scott, Yugi Mo, and Dr. Douglas Downey.

Contents

Contents **v**

1 Introduction **1**

1.1 Machine Learning 1

1.1.1 Active Learning 1

1.1.1.1 Bio Informatics 1

2 Background and Related Work **2**

2.1 Other Datasets from Yugi 2

2.2 Other Papers cited by Yugi 2

3 Bio HAL **3**

3.1 Passive SVM Rbf kernel vs Logistic Reg 3

3.2 Active vs Passive curves 3

3.2.1 Plots for Logistic Regression Active vs Passive curves 4

3.2.2 Plots for SVM Active vs Passive curves 8

3.3 Plots for FFR experiments 12

4 Conclusions and Future Work **14**

A Tuning the fine grained classes **15**

Bibliography **16**

Chapter 1

Introduction

1.1 Machine Learning

add text'example cite'[4]

1.1.1 Active Learning

add text'example cite'[4]

1.1.1.1 Bio Informatics

add text'example cite'[4]

Chapter 2

Background and Related Work

2.1 Other Datasets from Yugi

add text'example cite'[4]

2.2 Other Papers cited by Yugi

add text'example cite'[4]

Chapter 3

Bio HAL

3.1 Passive SVM Rbf kernel vs Logistic Reg

add text'example cite'[4]

3.2 Active vs Passive curves

The following plots were obtained with a round batch size of 100 and a starter set of 1040 instances out of the total 2098 instances. The plots are the average of 10 folds, for each fold a test set of 2010 containing representatives of each class was held out, out of the remaining 18088, the starter set was selected which again contained representatives of each class. Coarse and fine classifiers share the same starter set. During each round coarse and fine classifiers are trained on their corresponding sets, metrics are outputted on the held out test set, then confidence estimates are ran on the remaining eligible instances. Eligible instances are kept in separate sets for coarse and fine, 100 of the most uncertain instances are removed from each eligible set and added to its corresponding coarse or fine set to be trained on for the next round.

3.2.1 Plots for Logistic Regression Active vs Passive curves

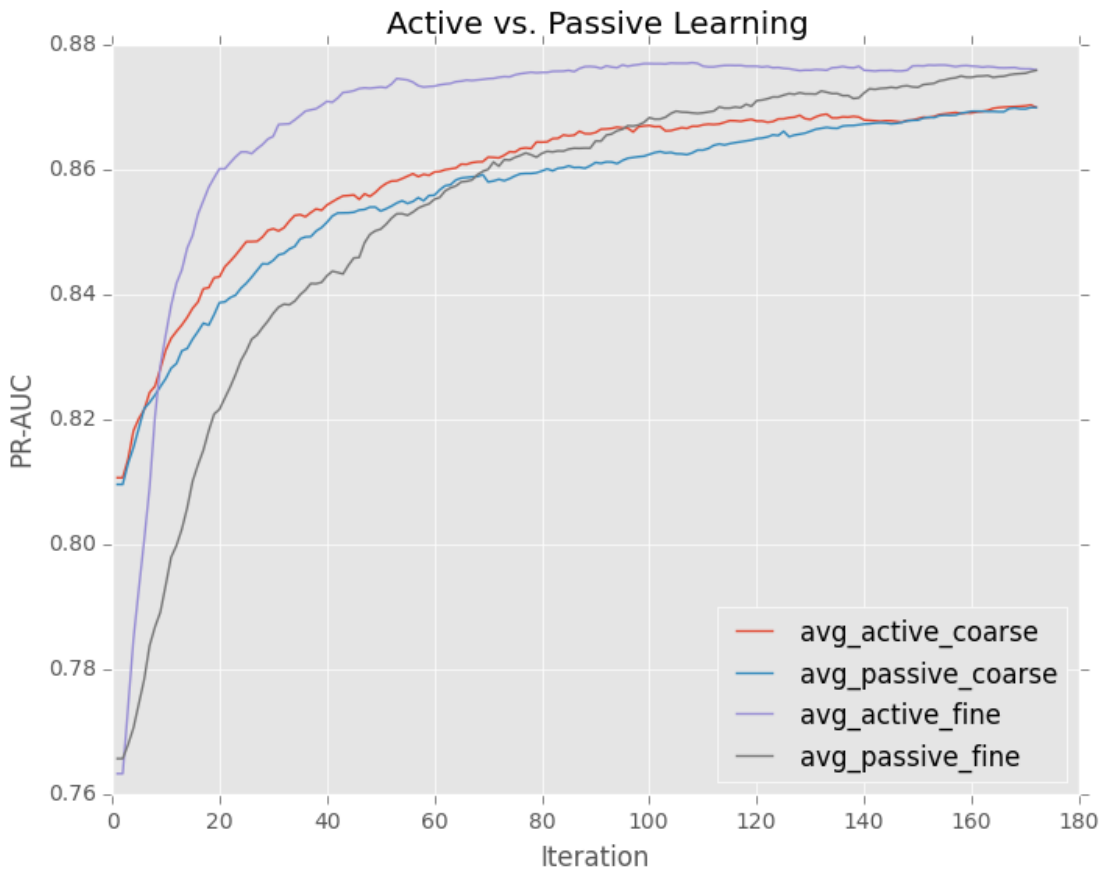

Figure 3.1: The PR AUC curves for rounds with the Logistic Regression classifier conforms to expectations, with active-fine having the highest performance. Active-coarse outperforms passive-coarse. Passive-fine doesn't outperform the coarse classifiers until rnd 100.

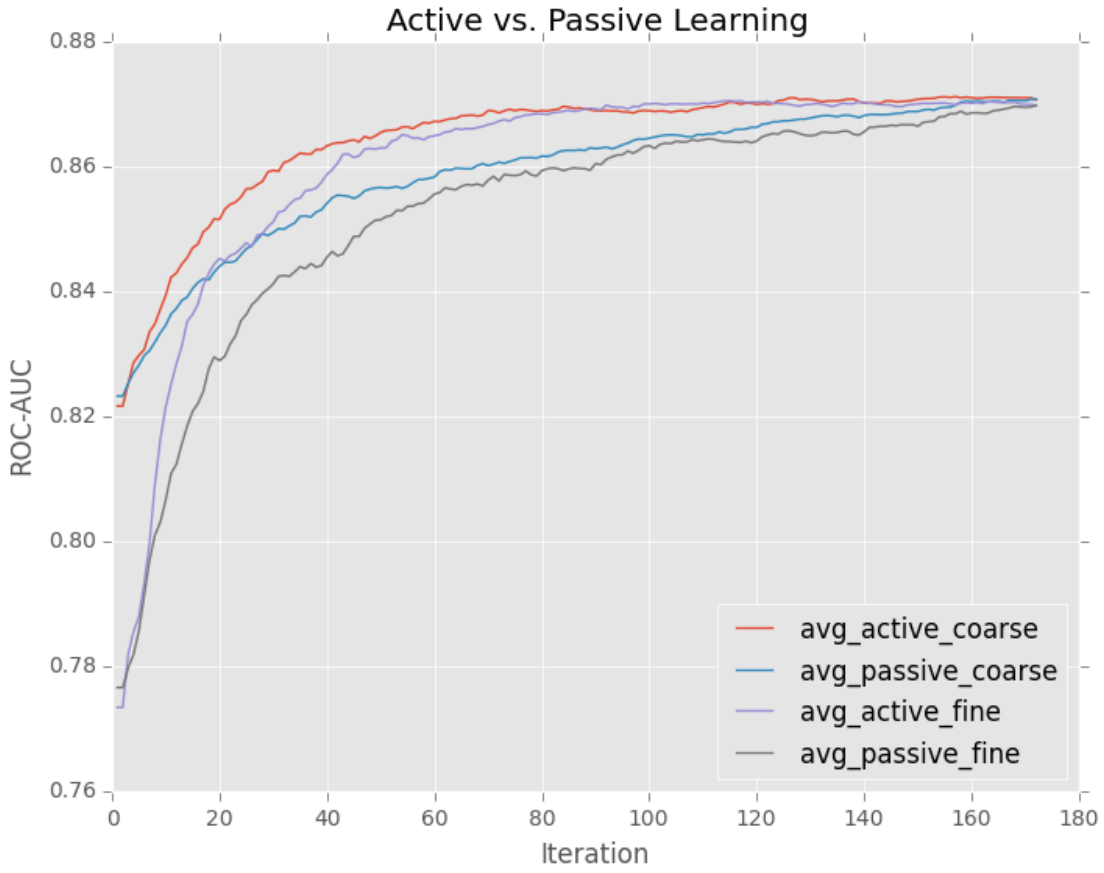

Figure 3.2: The ROC AUC curves for rounds with the Logistic Regression classifier. The active curves beat out the passive curves for both coarse and fine. Coarse roc starts with an advantage over fine as in the PR curves. Both converge to the same rate after roc auc level after 80.

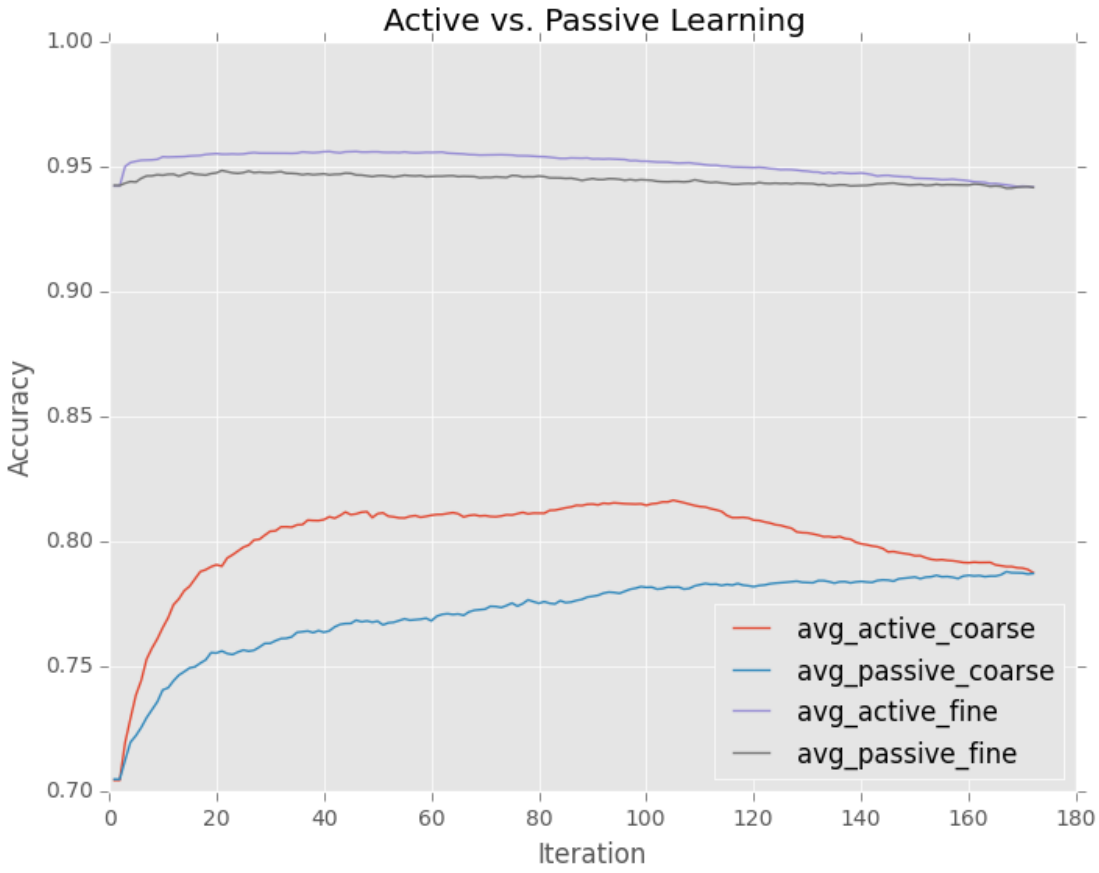

Figure 3.3: The accuracy of the fine classifiers stays at roughly the same rate throughout the rounds, this is due to an effective weighting scheme for the fine grained classifiers. The active coarse accuracy drops towards the end due to an increase in false positives as more negative instances are added in the later rounds.

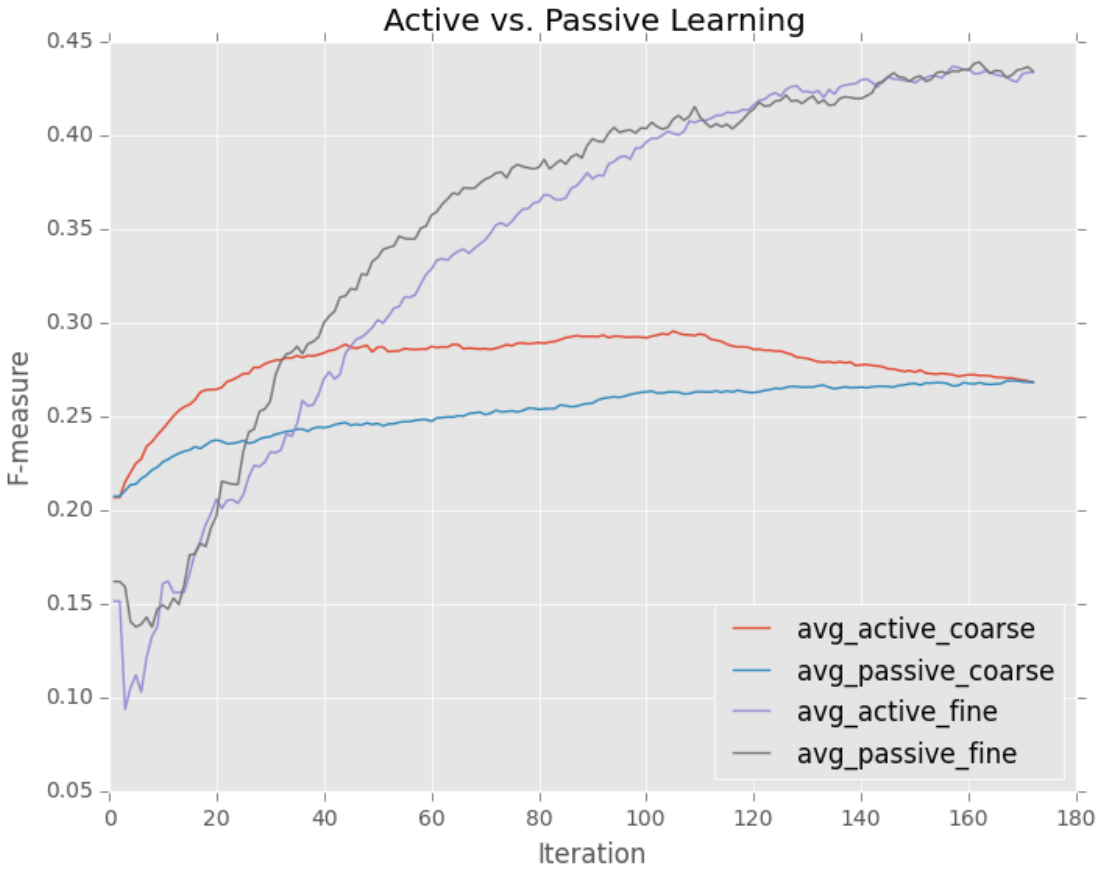

Figure 3.4: The F-measure of the the fine classifiers increases throughout the rounds as more true positives are predicted. The active coarse again decreases at later rounds due to increased false positives.

3.2.2 Plots for SVM Active vs Passive curves

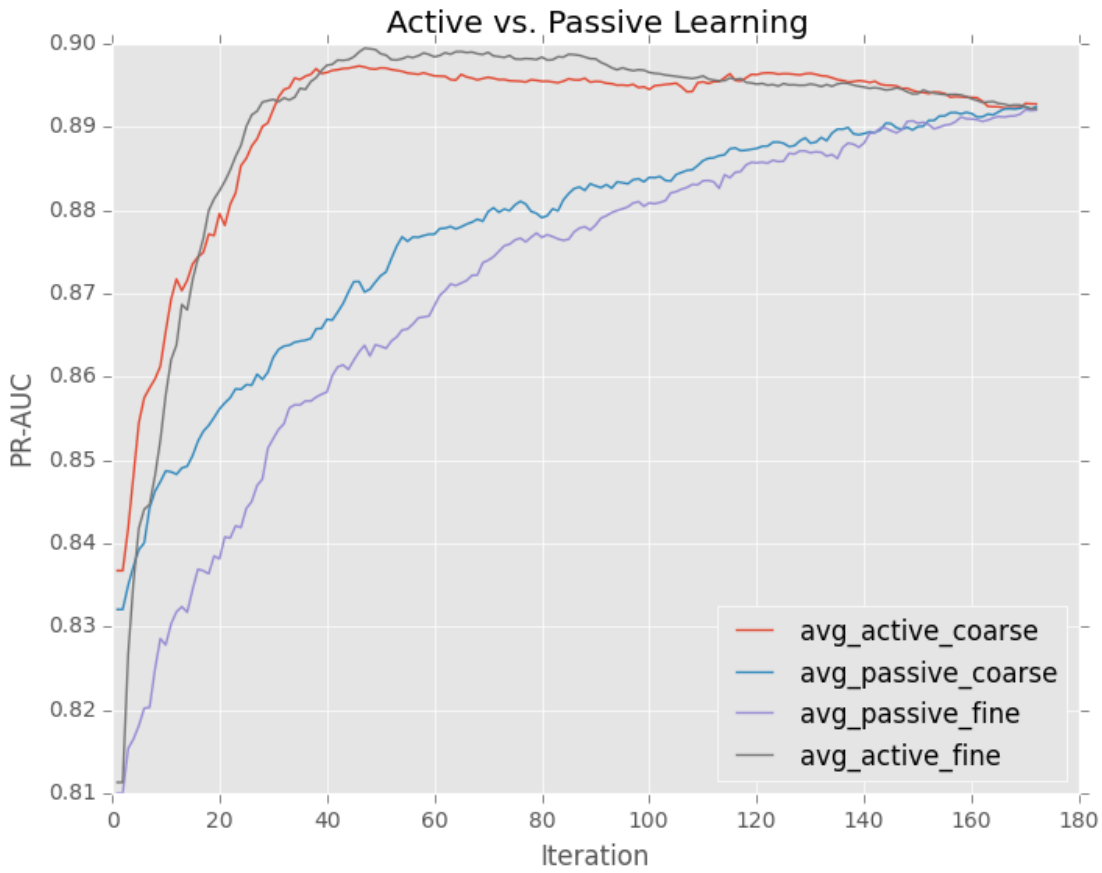

Figure 3.5: The PR AUC curves for rounds with SVM show little advantage for fine. The results are slightly different than the ones shown on 2/14 due to fixing a bug with the code that wasn't performing the preprocessing scaling for the SVM case at the same stage as it was being done for the logistic regression classifier.

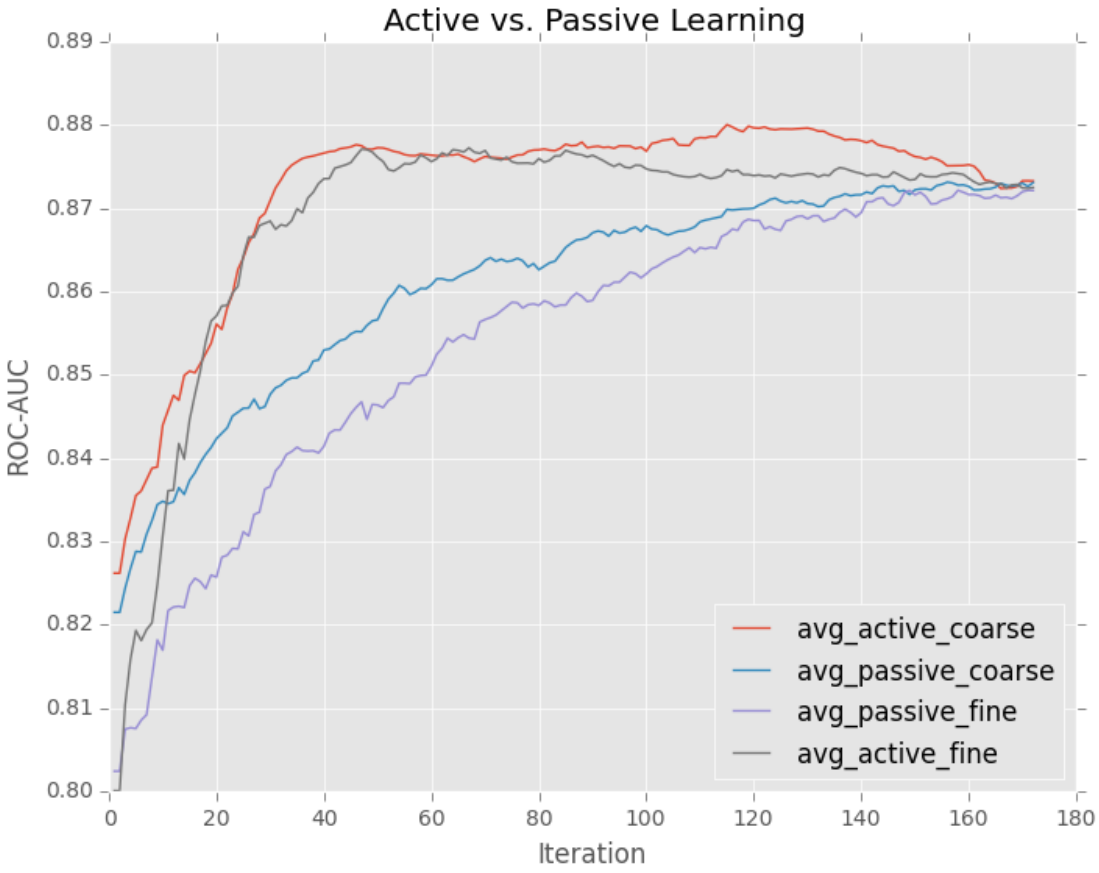

Figure 3.6: The ROC curves show more of an advantage for coarse classifiers.

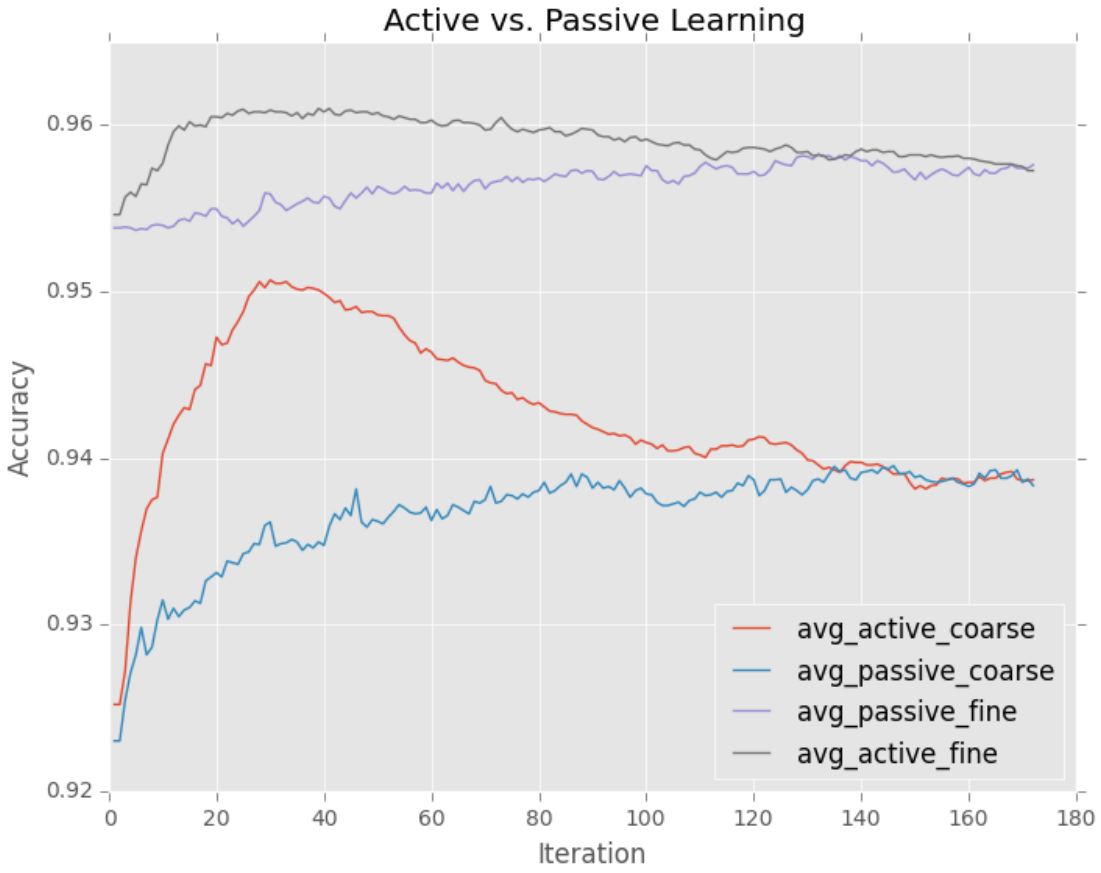

Figure 3.7: The accuracy for the coarse decreases sharply due to coarse predicting steadily more false positives, behaving similar to the Log Reg case. Fine accuracy is higher due to predicting less false positives than coarse. Fine also predicts less true positives, compare apx. 37 to apx. 60 t.p. for coarse at round 60.

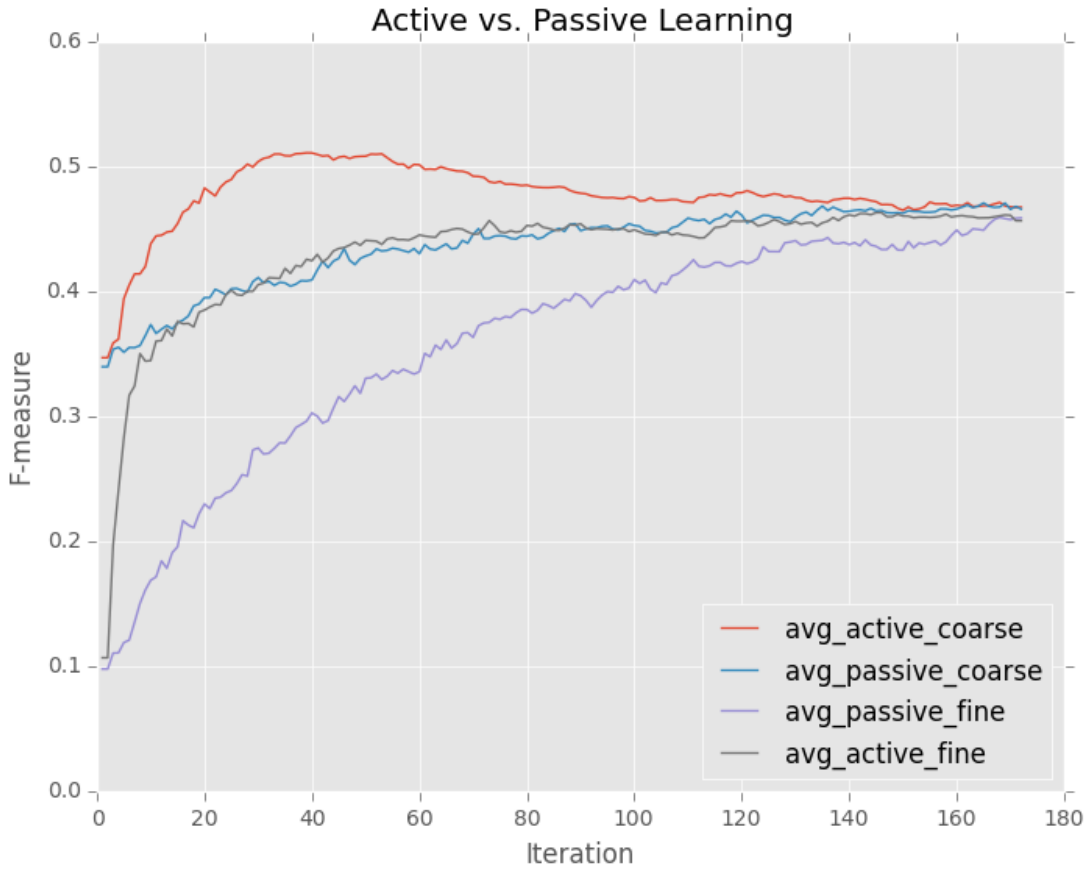

Figure 3.8: The F-measure favors coarse, and trends to the same level for both coarse and fine.

3.3 Plots for FFR experiments

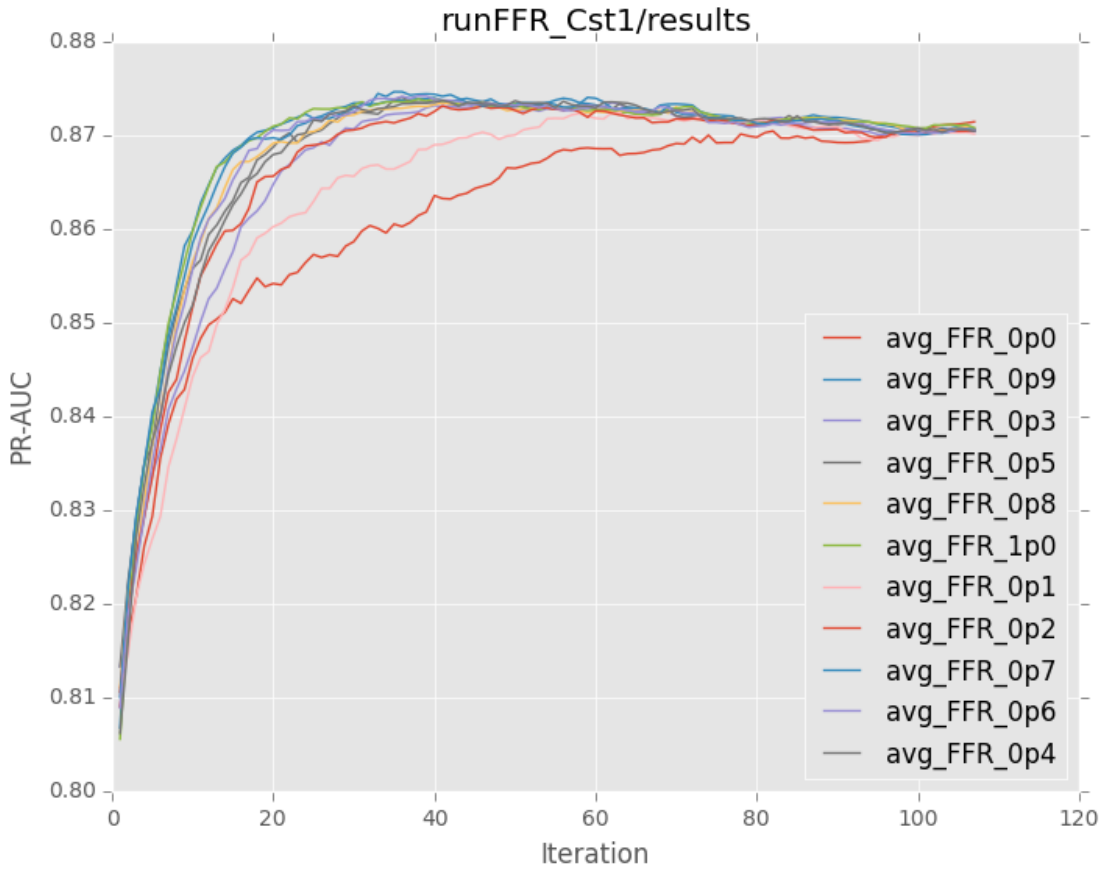

Figure 3.9: The round size is changed to 160 and a fine has a cost of 1. The 0p5 round for instance, corresponds to 0.5 of the total budget being used on fine, so 80 goes to fine and 80 goes to coarse. For the 0p0 round, none of the budget is used for fine and it has the worst performance. After around 0p3 the gains in performance are marginal. The performance increases with the green 1p0 curve outperforming the rest. The results are an average of 10 folds.

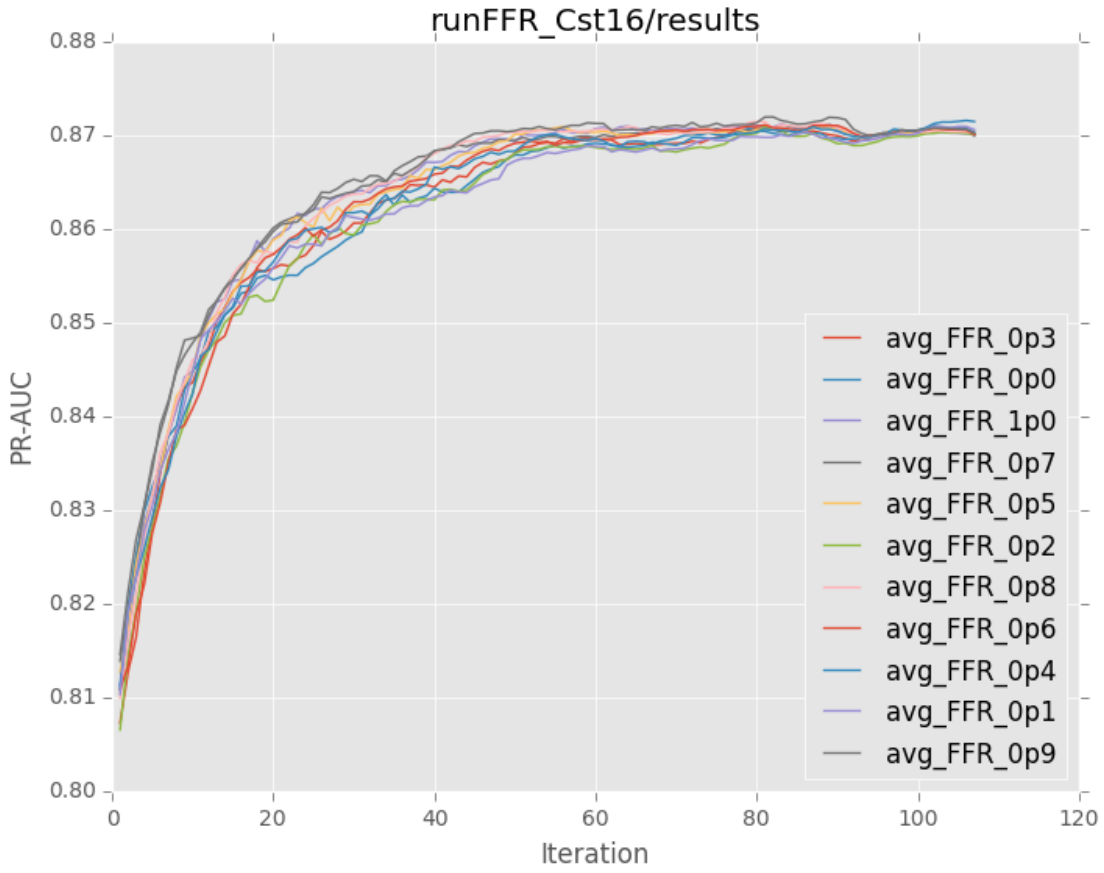

Figure 3.10: The round size is again 160 and fine has a cost of 16. The worst performing curve is again 0p0 with no fine instances, but the benefit of fine instances is marginal. For the 0p5 case 5 instances are purchased for fine and 80 are purchased for coarse. For the 1p0 case 16 instances are purchase for fine and 144 instances are purchased for coarse.

Chapter 4

Conclusions and Future Work

add text'example cite'[4]

Appendix A

Tuning the fine grained classes

add text'example cite'[4]

Bibliography

[1] Sanjoy Dasgupta and Daniel Hsu. Hierarchical sampling for active learning. *Proceedings of the 25th international conference on Machine learning - ICML '08*, pages 208–215, 2008.

[2] Xiao Ling and DS Weld. Fine-grained entity recognition. *Proceedings of the 26th Conference on Artificial . . .* , 2012.

[3] W. Luo, A. Schwing, and R. Urtasun. Latent structured active learning. In *Advances in Neural Information Processing Systems (NIPS)*, 2013.

[4] AKB Merialdo. Improving Collaborative Filtering For New-Users By Smart Object Selection. *In Proceedings of International Conference on Media Features (ICMF)*, May 2001. 1.1, 1.1.1, 1.1.1.1, 2.1, 2.2, 3.1, 4, A