# ACTIVE LEARNING WITH RICH FEEDBACK

BY

MANALI SHARMA

Submitted in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Computer Science
in the Graduate College of the
Illinois Institute of Technology

Approved _____
Advisor

Chicago, Illinois
July 2017

## ACKNOWLEDGMENT

My journey during my Ph.D. has been a truly enlightening and transformative experience for me. I would like to express my gratitude to everyone whose support and encouragement has made this remarkable journey possible for me.

I would like to begin by giving my special thanks to my advisor, Dr. Mustafa Bilgic, for his invaluable support and encouragement. He has been a source of great inspiration and has always motivated me to go beyond my potential. He has encouraged me to think creatively when approaching to challenging research problems. He has greatly helped me with my academic and professional endeavors. I have enjoyed having excellent research discussions and stimulating conversations with him. I look up to him for his scientific expertise and knowledge. His invaluable teachings will stay with me lifelong. He kept his belief in me even during tough times. For his tremendous guidance and mentorship, I cannot thank him enough.

I gratefully acknowledge the funding that I received from the National Science Foundation CAREER award no. IIS-1350337 for this research.

My gratitude also extends to my dissertation committee members, Dr. Gady Agam, Dr. Shlomo Argamon, Dr. Boris Glavic, and Dr. Lulu Kang. Their academic support, insightful questions, and helpful feedback are greatly appreciated.

I thank the Machine Learning Lab members, Maria E. Ramirez Loaiza, Caner Komurlu, and Ping Liu, for their friendship and useful discussions. It was a pleasure working alongside them. Their support and friendship will always be cherished.

I would like to thank the Data Sciences Group members, Nikunj C. Oza, Kamalika Das, Bryan Matthews, and David Nielsen, at NASA Ames Research Center. I thank them for their insightful discussions and valuable suggestions during my internship at NASA Ames Research Center and during our collaborative work for the research. Our collabora-

tion helped me to broaden my research perspective and provided an opportunity to apply my research in a practical setting.

Lastly, I would like to thank my family for their love and encouragement. My deepest gratitude goes to my parents, Renu Sharma and Krishna Gopal Sharma, who have always helped me in achieving my dreams. I thank my sister, Sonali, and my brother, Anirudh, for supporting me in all my pursuits and uplifting my spirits. My family's support throughout my Ph.D., during the trials, tribulations, and triumphs, cannot be expressed in words. I thank you all enormously.

TABLE OF CONTENTS

Page

ACKNOWLEDGEMENT . . . . . iii

LIST OF TABLES . . . . . viii

LIST OF FIGURES . . . . . xii

ABSTRACT . . . . . xiii

CHAPTER

1. INTRODUCTION . . . . . 1

   1.1. Contributions of this Dissertation . . . . . 3

2. BACKGROUND AND RELATED WORK . . . . . 7

   2.1. Active Learning . . . . . 7
   2.2. Active Learning Strategies . . . . . 10
   2.3. Incorporating Domain Knowledge into Learning . . . . . 15

3. FRAMEWORK TO MAKE THE ACTIVE LEARNER TRANSPARENT . . . . . 25

   3.1. Introduction . . . . . 25
   3.2. Background and Problem Formulation . . . . . 28
   3.3. Experimental Methodology and Results . . . . . 35
   3.4. User Study . . . . . 46
   3.5. Analytical and Empirical Justifications . . . . . 49
   3.6. Extension to Other Classifiers and Multi-class Classification . . 66
   3.7. Conclusion . . . . . 74

4. RATIONALES FRAMEWORK FOR DOCUMENT CLASSIFICATION . . . . . 75

   4.1. Introduction . . . . . 75
   4.2. Background . . . . . 77
   4.3. Learning with Rationales . . . . . 79
   4.4. Comparison with Baselines . . . . . 92
   4.5. Active Learning with Rationales . . . . . 106
   4.6. Conclusion . . . . . 111

5. RATIONALES FRAMEWORK FOR AVIATION DOMAIN . . . . . 112

   5.1. Introduction . . . . . 113
   5.2. Background . . . . . 115

5.3. Active Learning with Rationales . . . . . 119
5.4. Empirical Evaluation . . . . . 124
5.5. Towards Deployment . . . . . 131
5.6. Conclusion . . . . . 132

6. EXPLANATIONS FRAMEWORK FOR DOCUMENT CLASSIFICATION . . . . . 134

6.1. Introduction . . . . . 134
6.2. Background . . . . . 136
6.3. Learning with Explanations . . . . . 138
6.4. Experimental Methodology and Results . . . . . 140
6.5. Graphical User Interface . . . . . 156
6.6. Conclusion . . . . . 161

7. CONCLUSION AND FUTURE RESEARCH DIRECTIONS . . . . 162

7.1. Summary of Contributions . . . . . 162
7.2. Future Research Directions . . . . . 164
7.3. Conclusion . . . . . 167

APPENDIX . . . . . 169

A. LICENSES AND PERMISSION TO REUSE MATERIAL FROM PUBLICATIONS IN THIS THESIS . . . . . 169

BIBLIOGRAPHY . . . . . 172

LIST OF TABLES


Table Page

3.1 Description of the datasets: the domain, number of instances, number of features, types of features, and the percentage of minority class in the datasets. The datasets are sorted in increasing order of class imbalance. . 37

3.2 UNC-CE and UNC-IE with $t = 5$, $t = 10$, and $t = 20$ versus UNC-1. Number of datasets on which UNC-CE and UNC-IE significantly Win (W), Tie (T), or Lose (L) compared to UNC-1 baseline. . . . . 41

3.3 UNC-CE and UNC-IE versus UNC-u with $t = 5$, $t = 10$, and $t = 20$. Number of datasets on which UNC-CE and UNC-IE significantly Win (W), Tie (T), or Lose (L) compared to UNC-u baseline. . . . . 42

3.4 The mean rank of uncertain instances selected by UNC-CE and UNC-IE for the eight datasets over various iterations of learning and 25 trials. . . 43

3.5 Running times (in seconds) for three datasets for one iteration of active learning, with various $t$ values. We present mean $\pm$ Std. Dev of the running times over 25 trials. . . . . 45

3.6 Annotation time of all users on UNC-CE and UNC-IE movie reviews. The t-test results show that annotation times of UNC-CE and UNC-IE reviews are not significantly different. We report the $p-$values obtained using two-tailed unpaired t-tests. . . . . 48

3.7 Accuracy of all users on UNC-CE and UNC-IE movie reviews. . . . . 49

3.8 Spearman rank correlations between evidence and density, and evidence and prediction variance, with respect to the model trained on $\mathcal{L}$. . . . . 63

3.9 UNC-1, UNC-CE, and UNC-IE versus QBC. Number of datasets on which UNC-1, UNC-CE, and UNC-IE significantly Win (W), Tie (T), or Lose (L) compared to QBC baseline. . . . . 66

4.1 The Lw/oR binary representation (top) and its LwR transformation (bottom) for Documents 1, 2, and 3. Stop words are removed. LwR multiplies the rationales with $r$ and other features with $o$. . . . . 81

4.2 Description of the datasets: the domain, number of instances in training and test datasets, and size of vocabulary. . . . . 83

4.3 Comparison of number of documents needed to be annotated to achieve various target AUC performances using Lw/oR and LwR with multinomial naïve Bayes using binary representation. 'N/A' represents that a target AUC cannot be achieved by the learning strategy. . . . . 90

4.4 Comparison of number of documents needed to be annotated to achieve various target AUC performances using Lw/oR and LwR with multinomial naïve Bayes using tf-idf representation. 'N/A' represents that a target AUC cannot be achieved by the learning strategy. . . . . 91

4.5 Hyper-parameter settings for Lw/oR-SVM, LwR-SVM, and Zaidan et al. [2007] that gave the best learning curves. . . . . 104

4.6 Hyper-parameter settings for Lw/oR-MNB, LwR-MNB, and Melville and Sindhwani [2009] that gave the best learning curves. . . . . 104

4.7 Hyper-parameter settings for Lw/oR-LWLR, LwR-LWLR, and Das et al. [2013] that gave the best learning curves. . . . . 105

4.8 T-test results for UNC-PC versus UNC. UNC-PC improves over UNC significantly for all three classifiers and most of the datasets. . . . . 111

5.1 Comparison of number of labeled flights required by various strategies to achieve a target $precision@5$. 'n/a' represents that the target performance cannot be achieved by a method even with 45 labeled flights. . . . . 127

5.2 Comparison of number of labeled flights required by various strategies to achieve a target $precision@10$. 'n/a' represents that the target performance cannot be achieved by a method even with 45 labeled flights. . . 127

6.1 The binary representation (top) and its LwE transformation (bottom) for Document 2 (D2). Stop words are removed. LwE creates multiple pseudo-documents with various feature weights and class labels. . . . . 140

6.2 Description of the datasets: the domain, number of instances in training and test datasets, and size of vocabulary. . . . . 141

6.3 Comparison of number of documents required to achieve a target AUC by TL, LwE, and LwR using multinomial naïve Bayes. 'n/a' represents that a target AUC cannot be achieved by a method. . . . . 145

6.4 Average number of explanations, average number of words per explanations, accuracy, and average time taken by three users to annotate 200 movie reviews. . . . . 161

LIST OF FIGURES


Figure Page

3.1 Conflicting-evidence vs. insufficient-evidence uncertainty. Conflicting-evidence uncertainty represents a model's uncertainty on an instance due to strong evidence for each class, whereas insufficient-evidence uncertainty represents a model's uncertainty on an instance due to insufficient evidence for each class. Traditional uncertainty sampling does not care about the reasons for uncertainty, and picks the most uncertain instance. 27

3.2 AUC results for all eight datasets. UNC-CE significantly outperforms UNC-1 on seven out of eight datasets ((a), (b), (c), (d), (f), (g), and (h)) and loses on Sick dataset (e). UNC-IE loses to UNC-1 on seven out of eight datasets ((a), (b), (c), (d), (e), (f), and (g), and wins on Hiva dataset (h). . . . . 38

3.3 Accuracy results for four medium-imbalanced datasets (Spambase, Ibn Sina, Calif. Housing and Nova). UNC-CE outperforms UNC-1 on three datasets ((a), (b) and (c)) and loses on Nova (d). UNC-IE loses to UNC-1 on all four datasets. F1 results for four relatively skewed datasets (Sick, Zebra, LetterO and Hiva). UNC-CE outperforms UNC-1 significantly on three datasets ((e), (f) and (h)), and loses on one (g). UNC-IE loses to UNC-1 on all four datasets. . . . . 39

3.4 Histograms showing ranks of uncertain instances selected by UNC-CE and UNC-IE for all eight datasets. . . . . 44

3.5 (a) Average AUC of UNC-CE and UNC-IE over 10 trials on IMDB dataset. (b) Performance of UNC-CE and UNC-IE on the trial used in the user study. . . . . 47

3.6 (a) Performance comparison of UNC-CE and UNC-IE strategies based on the annotation time of Average User using ground-truth labels. (b) Performance comparison of UNC-CE and UNC-IE strategies based on the annotation time of Average User and using majority vote labels. . . . . 50

3.7 Analysis of Gaussian naïve Bayes using two continuous attributes, $X_1$ and $X_2$. The mean of both attributes for class $+1$ is $a$, and the mean of both attributes for class $-1$ is $b$. We consider two instances, $\langle \frac{a+b}{2}, \frac{a+b}{2} \rangle$ and $\langle c, d \rangle$ on the decision boundary and we prove that $\langle \frac{a+b}{2}, \frac{a+b}{2} \rangle$ is insufficient-evidence uncertain instance and we refer to it as $x^{UNC-IE}$ on this graph, and $\langle c, d \rangle$ is conflicting-evidence uncertain instance and we refer to it as $x^{UNC-CE}$ on this graph. . . . . 57

3.8 The histogram of $P(Y = +1|X, \mathcal{L})$ for two instances that are uncertain for two different reasons: conflicting-evidence vs. insufficient-evidence. 64

3.9 AUC results for all eight datasets. UNC-CE outperforms QBC on seven out of eight datasets ((b), (c), (d), (e), (f), (g), and (h)) and loses on Spambase dataset (a). UNC-IE loses to QBC on seven out of eight datasets ((a), (b), (c), (d), (e), (f), and (g), and wins on Hiva dataset (h). . . . . 67

3.10 Accuracy results for four medium-imbalanced datasets (Spambase, Ibn Sina, Calif. Housing and Nova). UNC-CE outperforms QBC on three datasets ((b), (c) and (d)) and loses on Spambase (a). UNC-IE loses to QBC on all four datasets. F1 results for four relatively skewed datasets (Sick, Zebra, LetterO and Hiva). UNC-CE outperforms QBC significantly on three datasets ((e), (f) and (h)), and loses on one (g). UNC-IE loses to QBC on all four datasets. . . . . 68

4.1 Words selected as rationales for positive movie reviews (top) and negative movie reviews (bottom) for IMDB dataset. . . . . 86

4.2 Comparison between LwR and Lw/oR using multinomial naïve Bayes, logistic regression, and support vector machines on four datasets: IMDB ((a), (b), and (c)), NOVA ((d), (e), and (f)), SRAA ((g), (h), and (i)), and WvsH ((j), (k), and (l)). LwR provides drastic improvements over Lw/oR for all datasets with binary and tf-idf representations and using all three classifiers. . . . . 88

4.3 (a) Results showing the effect of setting $C = 0.01$ for Lw/oR using binary and tf-idf representations. (b) Results showing the effect of multiplying the weights for all features by $0.01$, i.e. setting $r = 0.01$ and $o = 0.01$. Using a higher regularization, $C = 0.01$, for Lw/oR or indiscriminately multiplying the weights of all features by $0.01$ does not provide improvement over Lw/oR. . . . . 89

4.4 Comparison of LwR to Lw/oR on user-annotated IMDB dataset with tf-idf representation using (a) multinomial naïve Bayes, (b) logistic regression, and (c) support vector machines. LwR with default weight setting of $r = 1$ and $o = 0.01$ provides improvements over Lw/oR using all three classifiers. Since user-annotated rationales can be rather noisy, LwR with weights $r = 1$ and $o = 0.1$ performs better than LwR with weights $r = 1$ and $o = 0.01$. . . . . 93

4.5 Results comparing our approach to the three baselines using best hyper-parameters. LwR-MNB performs similar to Melville and Sindhwani [2009] on all four datasets ((a), (d), (g), and (j)). LwR-LWLR performs similar to Das et al. [2013] on all four datasets ((b), (c), (h), and (k)). LwR-SVM performs similar to Zaidan et al. [2007] on all four datasets ((c), (f), (i), and (l)). . . . . 102

4.6 Results comparing our approach to the three baselines with hyper-parameters tuned using cross-validation on labeled data. LwR-MNB performs similar to Melville and Sindhwani [2009] on all four datasets ((a), (d), (g), and (j)). LwR-LWLR performs similar to Das et al. [2013] on all four datasets ((b), (c), (h), and (k)). LwR-SVM performs similar to Zaidan et al. [2007] on all four datasets ((c), (f), (i), and (l)). . . . . 107

4.7 Comparison of LwR using UNC and UNC-PC for all datasets with tf-idf representation and using multinomial naïve Bayes classifier. . . . . 110

5.1 System setup: Data collection, processing, and mining. . . . . 118

5.2 Expected flight path and deviation from it for 4 flights. The first three flights are NOS. The last flight is an OS flight. . . . . 122

5.3 MLP vs. RND and MKAD-SAMPLING. MLP significantly outperforms RND and MKAD-SAMPLING for both (a) $precision@5$ and (b) $precision@10$. . . . . 126

5.4 MLP-W/RATIONALES VS. MLP. Incorporating rationales further improves performance over MLP for both (a) $precision@5$ and (b) $precision@10$. . . . . 128

5.5 Comparison of rationale features weights $w_r$ for MLP-W/RATIONALES using (a) $precision@5$ and (b) $precision@10$ . . . . . 129

5.6 Diagrammatic representation of the GUI for deployment of active learning as part of the anomaly detection framework . . . . . 132

6.1 Comparison of LwE to TL and LwR. LwE provides significant improvements over TL. LwE statistically significantly wins over LwR for (a), (b), (c), (d), and (e). LwE ties with LwR on WvsH dataset using support vector machines (f). . . . . 144

6.2 Comparison of LwE with TL and LwR under best parameter settings. . 146

6.3 LwE with incorporating any one explanation vs. all explanations, and LwR with incorporating any one rationale vs. all rationales for all three dataset using multinomial naïve Bayes and support vector machines. . 148

6.4 LwE with noisy explanations, noise level, $k$=2, for IMDB and WvsH datasets using multinomial naïve Bayes and support vector machines. . 150

6.5 LwE with noisy explanations, noise level, $k$=4, for IMDB and WvsH datasets using multinomial naïve Bayes and support vector machines. . 151

6.6 LwR with noisy rationales, noise level, $k$=2, for IMDB and WvsH datasets using multinomial naïve Bayes and support vector machines. . 152

6.7 LwR with noisy rationales, noise level, $k$=4, for IMDB and WvsH datasets using multinomial naïve Bayes and support vector machines. . 153

6.8 LwE with any one noisy explanation (LwE-Any Exp) and noise levels, $k$=2 and $k$=4, with optimal hyper-parameter settings for IMDB and WvsH datasets using multinomial naïve Bayes. . . . . 154

6.9 LwE with any one noisy explanation (LwE-Any Exp) and noise levels, $k$=2 and $k$=4, with optimal hyper-parameter settings for IMDB and WvsH datasets using support vector machines. . . . . 154

6.10 Performances of LwE and LwR with fallible experts. . . . . 157

6.11 Performances of LwE and LwR with reluctant experts. . . . . 158

6.12 Graphical user interface for the Learning with Explanations framework. 159

6.13 LwE with explanations and labels provided by the three users using (a) multinomial naïve Bayes and (b) support vector machines. LwE with real user-annotated explanations provides improvements over traditional learning (TL). . . . . 160

# ABSTRACT

One of the goals of artificial intelligence is to build predictive models that can learn from examples and make predictions. Predictive models are useful in many domains and applications such as predicting fraud in credit card transactions, predicting whether a patient has heart-disease, predicting whether an email is a spam, predicting crime, recognizing images, recognizing speech, and many more. Building predictive models often requires supervision from a human expert. Since there is a human in the loop, the supervision needs to be as resource-efficient as possible to save the human's time, cost, and effort in providing supervision. One solution to make the supervision resource-efficient is active learning, in which the active learner interacts with the human to acquire supervision, usually in the form of labels, for a few selected examples to effectively learn a function that can be used to make predictions. In this thesis, I explore more intuitive and effective use of human supervision through richer interactions between the human expert and the learner, so that the human can understand the learner's reasoning for querying examples, and provide information beyond just the labels for examples.

Traditional active learning approaches select informative examples for labeling, but the human does not get to know why those examples are useful to the learner. While interacting with the learner to annotate examples, humans can provide rich feedback, such as provide their prior knowledge and understanding of the domain, explain certain characteristics of the data, suggest important attributes of the data, give rationales for why an example belongs to a certain category, and provide explanations by pointing out features that are indicative of certain labels. The challenge, however, is that traditional supervised learning algorithms can learn from labeled examples, but they are not equipped to readily absorb the rich feedback. In this thesis, we enable the learner to explain its reasons for selecting instances and devise novel methods to incorporate rich feedback from humans into the training of predictive models. Specifically, I build and evaluate four novel active

learning frameworks to enrich the interactions between the human and learner.

First, I introduce an active learning framework to reveal the learner's perception of informative instances. Specifically, we enable the learner to provide its reasons for uncertainty on examples and utilize the learner's perception of uncertainty to select better examples for training the predictive models. Second, I introduce a framework to enrich the interaction between the human and learner for document classification task. Specifically, we ask the human to annotate documents and provide rationales for their annotation by highlighting phrases that convinced them to choose a particular label for a document. Third, I introduce a framework to enrich the interaction between the human and learner for the aviation domain, where we ask subject matter experts to examine flights and provide rationales for why certain flights have safety concerns. Fourth, I introduce a framework to enrich the interaction between the human and learner for document classification task, where we ask humans to provide explanations for classification by highlighting phrases that reinforce their belief in the document's label and striking-out phrases that weaken their belief in the document's label. We show that enabling richer interactions between the human and learner and incorporating rich feedback into learning lead to more effective training of predictive models and better utilization of human supervision.

# CHAPTER 1

# INTRODUCTION

Artificial intelligence (AI) is the science of making intelligent software and machines. AI consists of many different subfields, such as machine learning, vision, navigation, reasoning, planning, and natural language processing. In the information age of today's, where data is abundant, it is important to learn concepts from data to improve businesses and services, develop new techniques and products, and build tools for knowledge discovery. One of the goals of AI is to make intelligent systems that can make predictions. Predictive models are used in many domains such as image recognition, speech recognition, email classification, recommender systems, credit card fraud detection, crime prediction, and medical diagnosis, to name a few.

Predictive models are built by learning a function that maps the training data to a target variable, and once the model is built, it can be used to make predictions on future examples, which the model has not seen in the training data. These models can be quite complex and are usually developed behind-the-scenes by carefully choosing the training data and evaluating the model's ability to correctly predict a target variable. Very often, these models make use of supervised learning algorithms, such as naïve Bayes, logistic regression, support vector machines, neural networks, and decision trees. Supervised learning algorithms learn from examples, where examples thrive on some supervision, usually in the form of predetermined classification, known as labels.

In domains where examples do not come with labels, supervised learning approaches require *supervision*, usually from a human expert. It is impractical, if not impossible, for a human expert to go over thousands or millions of examples in the data and provide supervision. For example, speech recognition algorithms are trained on large vol-

umes of recorded speech and their transcription, where transcription is manually done by a human. Face detection algorithms are trained on images where the faces are manually marked by humans. Medical diagnosis systems are trained on patient records that are diagnosed by doctors for a certain disease or condition. Credit card fraud detection systems are trained on transactions that have been identified as fraudulent or legitimate. Because human time and expertise is valuable, it is imperative that the example cases are chosen carefully; we cannot simply transcribe all speech, mark faces on all images, or mark all the credit card transactions as fraudulent or legitimate. Hence, the supervision needs to be made as efficient as possible, because supervision usually requires human expertise, time, cost, and effort. A solution to make supervision resource-efficient is *active learning* in which the learning algorithm carefully selects queries, usually examples, from which it wants to learn [57].

Active learning algorithms are supervised learning algorithms that iteratively select informative queries, based on past queries and responses, for annotation by human experts to effectively learn a suitable classifier [94]. Since there is a human in the loop, the active learner aims to select as few as possible, but useful, instances for labeling to save the human expert's time, cost, and effort, and still learn a good classification function. The problem of selecting informative instances optimally is intractable in general, which is typically solved using greedy algorithms that select informative instances according to utility-based heuristics, where high-utility instances are chosen to be labeled by a human labeler. For example, an active learning strategy that iteratively chooses instances about which the learner is uncertain has been shown to improve learning [57].

Much of the research on active learning focused on determining which instances are useful for learning and getting supervision from humans in the form of labels for instances, but the problems of making the active learning sessions viable in practice, identifying the best information for learning, and the best methods for interaction between an active learner

and the human are under-researched. Most active learning systems are opaque; the active learners do not explain why they query a particular instance, and when the humans provide answers to queries, the learner does not get to know the rationale behind the answers provided. Humans are more than just labelers and can provide rich feedback, such as teach the domain knowledge to the learner, point out important features, and provide rationales and explanations for their classification of instances.

## 1.1 Contributions of this Dissertation

In this dissertation, we make the active learning session more transparent for the human so that the human can understand why a particular example is chosen by the learner. We then enable the learner to accept rich feedback so that the human can provide rationales and explanations along with the labels for instances. We develop various active learning frameworks that can (i) facilitate the active learner to explain its queries on instances and (ii) effectively utilize the rich feedback provided by humans to increase the learning efficiency and minimize the time and effort of the human expert. Next, I briefly describe the active learning frameworks that we developed.

**1.1.1 Framework to Make the Active Learner Transparent.** We make the active learner transparent by enabling the learner to provide its reasons for querying an instance. Specifically, we look into uncertainty sampling, an active learning strategy that selects instances on which the learner is uncertain, and dig deeper into why the learner might be uncertain on the instances. We present an evidence-based framework that can uncover the reasons for learner's uncertainty on instances and guide the learner in selecting useful instances for querying to speed-up the training process. I discuss our evidence-based framework in detail in Chapter 3 and provide analytical and empirical justifications for why uncovering the reasons for learner's uncertainty on instances is important for learning.

**1.1.2 Frameworks to Enrich the Interaction between the Expert and Learner.** We

make the interaction between the expert and learner richer for more effective and intuitive use of expert's time, cost, and effort in providing supervision. Specifically, we ask the human expert to provide reasons and explanations behind his/her answers. In traditional active learning setting, the learner asks the human to provide labels for instances that were selected by the learner. However, humans possess much more knowledge than just the labels for instances. For example, the human experts can provide rationales and explanations for classification, which might include simple feature annotations, feature selection, feature rankings, rules, complex domain knowledge, or free-form text entries. However, incorporating supervision in the form of rationales and explanations into the learning process is not trivial, because the underlying models such as naïve Bayes, logistic regression, support vector machines, neural networks, and decision trees, cannot readily handle supervision other than the labels for instances. We enable a richer interaction between the human and learner by asking the human to provide rich feedback and enabling the active learner to incorporate the rich feedback, in the form of rationales and explanations for classification, into the learning process. Next, I describe three frameworks that we developed for incorporating rationales and explanations for classification into the training of predictive models.

**1.1.2.1 Rationales Framework for Document Classification.** In this framework, we ask the human expert to provide his/her rationales for choosing specific labels for instances. I primarily focus on document classification task, where the active learner presents documents to labelers and requests a label and a rationale for choosing labels for documents. The rationales framework can incorporate annotated documents, i.e., the documents with their labels and rationales, into the training of any off-the-shelf classifier. We empirically show that our framework effectively incorporates rationales for document classification into the training of multinomial naïve Bayes, logistic regression, and support vector machines.

I describe the rationales framework for document classification in detail in Chapter 4.

**1.1.2.2 Rationales Framework for Aviation Domain.** In this framework, we further allow experts to provide complex rationales that can be conjunction or disjunction of features in instances. In this case, I discuss a real-world application of active learning in the aviation domain. We worked in collaboration with NASA Ames Research Center to solve the problem of identifying unknown safety events in flight operations data. Unsupervised learning algorithms can identify statistically significant anomalies in flights, however, a very small fraction of statistical anomalies turns out to be operationally significant. A subject matter expert (SME) goes through some of the statistical anomalies to identify a few flights that are operationally significant (e.g., represent a safety concern). The SMEs cannot analyze all the statistical anomalies, since the SME's time and effort is limited. We apply active learning as a practical solution to efficiently build an effective model for predicting flights of operational significance and minimize the time and effort of SMEs in providing supervision. In this case, we ask the SMEs to provide a label and an explanation or a rationale for choosing a label for a flight. I present the rationales framework that can effectively incorporate complex rationales provided by SMEs to learn a suitable classifier that can be used for predicting operationally significant events in unseen flights. The rationales framework for aviation domain is described in detail in Chapter 5.

**1.1.2.3 Explanations Framework for Document Classification.** In this setting, we ask the human expert to provide explanations for classification of documents. We allow the expert to provide explanations in the form of domain-specific features that support and oppose the classification of documents. Supporting features are those whose presence strengthens our belief in the label. Opposing features are those features, which if removed from the instance, would make our belief in the label stronger. The main difference between rationales and explanations is that rationales are features whose existence in an instance convince the expert to choose a particular label, whereas explanations go one step further

and look at features which, if removed from the instance, would make the expert more confident in the chosen label. Our explanations framework can effectively incorporate explanations provided by experts to speed-up the training of any off-the-shelf classifier. I primarily focus on document classification task, where labelers provide explanations by highlighting words within a document that support or oppose the classification. I discuss the explanations framework for document classification in detail in Chapter 6.

The rest of the thesis is organized as follows. In Chapter 2, I provide background on active learning and describe some of the most popular active learning strategies. Then, I discuss related work on incorporating domain knowledge into learning, interactive machine learning, recent work on incorporating domain knowledge into active learning, anomaly detection with active learning, and other areas related to active learning. In Chapter 3, I discuss the evidence-based framework to make the active learner transparent. Then, I discuss the frameworks to enrich the interactions between the human and learner in Chapters 4, 5, and 6. Finally, in Chapter 7, I present a summary of my thesis, present future research directions, and conclude the thesis.

# CHAPTER 2

# BACKGROUND AND RELATED WORK

In this chapter, I describe active learning and the three settings for active learning, in which a learner can pose queries to an expert. Then, I provide detailed description of pool-based sampling for active learning, which is the setting that we use in this dissertation. Then, I describe some of the most common active learning methods from the active learning literature. Then, I provide the related work on incorporating domain knowledge into active learning.

## 2.1 Active Learning

Active learning methods interactively query human experts to obtain annotation for chosen examples to effectively learn the correct classification function [94]. The main idea behind active learning is that a machine learning algorithm can achieve greater performance by carefully selecting informative instances, compared to randomly selecting instances for annotation, if it is allowed to choose the examples from which it learns. An active learner poses queries, usually in the form of instances, and asks a human expert to annotate those instances. The active learner incorporates the annotated instances into its training data, re-trains itself, and then selects more instances for annotation, based on past queries and answers.

There are three settings in which an active learner can select queries to ask a human expert: (i) membership query synthesis, (ii) stream-based selective sampling, and (iii) pool-based sampling. Next, I describe these three active learning settings.

**2.1.1 Membership Query Synthesis.** The key idea behind membership query synthesis [2] is that if the learner can select the *right* examples to label, it can learn the target func-

tion with only a few examples. In membership query synthesis, the learner creates artificial examples, which when annotated would be most beneficial for learning, and presents those artificial examples to the expert. This setting is often applicable to experiments performed in laboratories where various variables need to be set and trying all possible variable settings is infeasible. A problem with creating artificial examples is that the new examples might not represent any real-world scenario. Consider for example the task of text classification, where the learner creates a new document containing an arbitrary list of words. For a human labeler, such a document might be awkward, if not meaningless.

**2.1.2 Stream-Based Selective Sampling.** In stream-based selective sampling [18], [20], the examples are presented to the learner in a stream, and the learner decides whether or not to sample the example and query its label based on some "informativeness measure" or "query strategy". Stream-based selective sampling is applicable to domains in which data is continuously available in stream, such as sensor data, and the data cannot be stored. In stream-based selective sampling, the examples will at least be sensible, since they come from a real underlying data distribution, whereas in membership query synthesis, the examples could be unrealistic.

**2.1.3 Pool-Based Sampling.** In pool-based sampling, the learner has access to a large amount of unlabeled data. The learner measures the "informativeness" of all the unlabeled examples and selects the "best" example based on some ranking of all the examples in the unlabeled data. The main difference between stream-based selective sampling and pool-based sampling is that in the former, the learner must decide on the "informativeness" of an instance individually, whereas in the latter, the learner can rank and compare "informativeness" of all the available unlabeled instances.

Pool-based active learning is extremely important in today's machine learning and data mining applications, because large amounts of unlabeled data are available in many domains. **In this dissertation, we work with the pool-based sampling for active learn-**

**ing**. Next, I describe the pool-based sampling for active learning in detail.

In pool-based active learning, we assume that we are given a dataset $\mathcal{D}$ of instances consisting of attribute vector and label pairs $\{X^{(i)}, Y^{(i)}\}$. Let the uppercase $X$ denote the random variable representing an instance and the lowercase $x$ represent a particular instantiation of $X$. Each instance is described as a vector of $f$ attributes $X \triangleq \langle X_1, X_2, \cdots, X_f \rangle$. Similarly, let the uppercase $Y$ represent the class variable of the instance and let the lowercase $y$ represent a particular instantiation of $Y$. Each $X_i$ can be real-valued or discrete whereas $Y$ is discrete; **in this dissertation, we focus on the binary case, where** $Y \in \{-1, +1\}$. In the pool-based active learning setup, we are given a small set of instances whose labels are known: $\mathcal{L} = \{\langle x^{(i)}, y^{(i)} \rangle\}$, and a much larger collection of instances whose labels are unknown: $\mathcal{U} = \{\langle x^{(i)}, ? \rangle\}$.

The goal of active learning is to learn the correct classification function, $\theta : X \rightarrow Y$, by carefully choosing instances for labeling. Algorithm 1 formally describes the pool-based active learning. A pool-based active learning algorithm iteratively selects an informative instance $\langle x^*, ? \rangle \in \mathcal{U}$ and obtains its label $y^*$ from an expert to learn the classification function $\theta$. Selecting informative instances optimally is an NP-hard problem, which is typically optimized through greedy selection criteria, where informativeness of instances is measured by a *utility* function using the current model, $\theta$. The active learner selects high-utility instances to be labeled by an expert, incorporates the new labeled instance $\langle x^*, y^* \rangle$ into $\mathcal{L}$, and repeats this process until a stopping criterion is met, usually until a given budget, $B$, is exhausted.

**Algorithm 1** Pool-Based Active Learning

**Input:** $\mathcal{U}$ - unlabeled data, $\mathcal{L}$ - labeled data, $\theta$ - classification model, $B$ - budget
**repeat**
  **for all** $\langle x^{(i)}, ? \rangle \in \mathcal{U}$ **do**
    compute *utility*$(x^{(i)}, \theta)$
  **end for**
  pick highest utility $x^*$ and query its label
  $\mathcal{L} \leftarrow \mathcal{L} \cup \{\langle x^*, y^* \rangle\}$
  $\mathcal{U} \leftarrow \mathcal{U} \setminus \{\langle x^*, y^* \rangle\}$
  Train $\theta$ on $\mathcal{L}$
**until** Budget $B$ is exhausted; e.g., $|\mathcal{L}| = B$

## 2.2 Active Learning Strategies

In order to select high-utility or informative instances for labeling, a number of successful active learning methods have been developed in the past two decades. Examples include uncertainty sampling [57], query-by-committee [97], bias reduction [19], variance reduction [21], and expected error reduction [60], [88], to name a few. We refer the reader to [94] for a survey of active learning methods. Next, I describe some of the most popular active learning strategies from the active learning literature.

**2.2.1 Random Sampling.** The most common strategy that is used as a baseline for comparing the performance of an active learning strategy is random sampling, in which instances are picked at random from the unlabeled pool and given to the human expert for labeling, without paying any attention to whether those instances provide any additional information to the classifier. Instances selected randomly are inherently representative examples that are independent and identically distributed (*i.i.d.*), and hence, random sampling often serves as a strong baseline for other active learning strategies.

**2.2.2 Uncertainty Sampling.** Uncertainty sampling selects instances for which the current model is most uncertain how to label [57]. These instances correspond to the ones that lie close to the decision boundary of the model. Uncertainty of an underlying model can be

measured in several ways. One approach is to use conditional entropy:

$$x^* = \underset{x^{(i)} \in \mathcal{U}}{\operatorname{argmax}} - \sum_{y \in Y} P_\theta(y|x^{(i)}) \log \left( P_\theta(y|x^{(i)}) \right) \tag{2.1}$$

where $P_\theta(y|x^{(i)})$ is the probability that instance $x^{(i)}$ has label $y$. Another approach is to use maximum conditional:

$$x^* = \underset{x^{(i)} \in \mathcal{U}}{\operatorname{argmax}} \left( 1 - \max_{y \in Y} P_\theta(y|x^{(i)}) \right) \tag{2.2}$$

The last approach we discuss uses margin of confidence:

$$x^* = \underset{x^{(i)} \in \mathcal{U}}{\operatorname{argmin}} \left( P_\theta(y_m|x^{(i)}) - P_\theta(y_n|x^{(i)}) \right) \tag{2.3}$$

where, $y_m$ is the most likely label and $y_n$ is the next likely label for $x^{(i)}$. More formally,

$$y_m = \underset{y \in \mathcal{Y}}{\operatorname{argmax}} P_\theta(y|x^{(i)}) \tag{2.4}$$

$$y_n = \underset{y \in \mathcal{Y} \setminus \{y_m\}}{\operatorname{argmax}} P_\theta(y|x^{(i)}). \tag{2.5}$$

When the task is binary classification, that is when $Y \in \{+1, -1\}$, the highest utility is achieved when $P_\theta(+1|x^{(i)}) = P_\theta(-1|x^{(i)}) = 0.5$.

Uncertainty sampling is arguably one of the most common active learning methods and is frequently used as a baseline for comparing other active learning methods (e.g., [10], [95], and [114]). It has been shown to work successfully in a variety of domains. Example domains include text classification [10], [47], [57], [120], natural language processing [112], email spam filtering [91], [92], image retrieval [114], medical image classification [48], robotics [14], information retrieval [125], dual supervision [104], and sequence labeling [95], among many others.

Even though uncertainty sampling is frequently utilized, it is known to be susceptible to noise and outliers [88]. A number of approaches have been proposed to make it more robust. For example, Settles and Craven [2008] weighted the uncertainty of an instance by its density to avoid outliers, where density of the instance is defined as average similarity to other instances. Zhu et al. [2008] used a K-Nearest-Neighbor-based density measure to determine whether an unlabeled instance is an outlier. Xu et al. [2003] and Donmez et al. [2007] proposed a hybrid approach to combine representative sampling and uncertainty sampling. Other approaches used the cluster structure of the domain to choose more representative examples [10], [73].

**2.2.3 Most-Likely Positive.** Most-likely positive strategy selects instances for which the underlying model is most confident that the label is positive. For skewed datasets with minority class distribution much less than the majority class distribution, a common and simple approach is to maximize the chances of retrieving minority class instances [3], [7]. Considering minority class as positive, most-likely positive strategy aims to add more positive instances to the labeled set, $\mathcal{L}$. The objective is:

$$x^* = \underset{x^{(i)} \in \mathcal{U}}{\operatorname{argmax}} P_\theta(\hat{y}^+ | x^{(i)})$$

where, $\hat{y}^+$ represents the predicted positive label. Intuitively, most-likely positive strategy is a way of over-sampling the minority class in order to address the issue of imbalanced class distributions.

**2.2.4 Query by Committee.** Query-by-committee (QBC) is another frequently used baseline in active learning. QBC selects instances that once incorporated into learning will reduce the size of the version space [69]. A committee of classifiers is formed by sampling hypotheses from the version space, but since this is not always possible, Abe and Mamitsuka [1998] proposed two approximate versions of QBC, query-by-bagging and query-by-boosting. In query-by-bagging [11], several hypotheses are constructed by train-

ing models on replicates of training data obtained by sampling a number of instances from $\mathcal{L}$. Query-by-boosting uses AdaBoost [37] algorithm that learns a hypothesis using several weak hypotheses, and iteratively selects instances that were misclassified by previous weak hypotheses. The approximate versions of QBC select instances on which the committee disagrees the most. The disagreement between committee members can be measured in a number of ways. For example, Dagan and Engelson [1995] proposed vote entropy as a measure of disagreement between committee members to select informative instances:

$$x^* = \underset{x^{(i)} \in \mathcal{U}}{\operatorname{argmax}} - \sum_{y \in Y} \frac{V(y)}{C} \log \frac{V(y)}{C} \tag{2.6}$$

where, $y$ ranges over all possible labels in $Y$, $V(y)$ is the number of votes that label $y$ receives from the committee members, and $C$ is the committee size. McCallum and Nigam [1998] used Kullback-Leibler divergence to the mean as a measure of disagreement between committee members:

$$x^* = \underset{x^{(i)} \in \mathcal{U}}{\operatorname{argmax}} \frac{1}{C} \sum_{c=1}^{C} D(P_c(Y|x^{(i)})||P_{avg}(Y|x^{(i)})) \tag{2.7}$$

where, $C$ is the size of committee, $D(\cdot||\cdot)$ is the Kullback-Leibler divergence that measures difference between two probability distributions, $P_c(Y|x^{(i)})$ is the probability distribution over instance $x^{(i)}$ according to committee member $c$, and $P_{avg}(Y|x^{(i)})$ is the average probability distribution for instance $x^{(i)}$ over all committee members. Though computationally more demanding than uncertainty sampling, QBC is heavily used as a baseline for active learning, partly because it is less affected by noise and outliers and it is simple to implement.

**2.2.5 Expected Error Reduction.** Expected error reduction, as first defined by Lindenbaum et al. [1999] and then by Roy and McCallum [2001], aims to select instances that once incorporated into learning will produce the lowest expected error on the test set. The

expected error of the learner can be expressed as:

$$E_{\hat{P}_{\mathcal{L}}} = \int_x L(P(y|x), \hat{P}_{\mathcal{L}}(y|x))P(x) \tag{2.8}$$

where, $L$ is a loss function that measures the degree of error, that is, the difference between the true distribution, $P(y|x)$, and the distribution predicted by the model, $\hat{P}_{\mathcal{L}}(y|x)$. Since the true distribution is unknown, $P(y|x)$ is estimated using the current model. Two common loss functions are log loss and 0/1 loss. The expected error for log loss estimates the entropy of the model's posterior distribution:

$$\tilde{E}_{\hat{P}_{\mathcal{L}\cup(x^*,y^*)}} = \frac{1}{|\mathcal{U}|} \sum_{x^{(i)}\in\mathcal{U}} \sum_{y\in Y} \hat{P}_{\mathcal{L}\cup(x^*,y^*)}(y|x^{(i)}) \log(\hat{P}_{\mathcal{L}\cup(x^*,y^*)}(y|x^{(i)})) \tag{2.9}$$

where, $(x^*, y^*)$ is a possible candidate to be queried. The expected error for 0/1 loss is:

$$\tilde{E}_{\hat{P}_{\mathcal{L}\cup(x^*,y^*)}} = \frac{1}{|\mathcal{U}|} \sum_{x^{(i)}\in\mathcal{U}} \left(1 - \max_{y\in Y} \hat{P}_{\mathcal{L}\cup(x^*,y^*)}(y|x^{(i)})\right) \tag{2.10}$$

Expected error reduction strategy selects an instance that is expected to provide most reduction in the model's expected error on all instances in the test set:

$$x^* = \underset{x^{(i)}\in\mathcal{U}}{\operatorname{argmax}} \left(\tilde{E}_{\hat{P}_{\mathcal{L}}} - \sum_{y\in Y} P_{\mathcal{L}}(y|x^{(i)}) \left(\tilde{E}_{\hat{P}_{\mathcal{L}\cup(x^*,y^*)}}\right)\right) \tag{2.11}$$

where, $P_{\mathcal{L}}(y|x^{(i)})$ is the current model's posterior distribution, and $\tilde{E}_{\hat{P}_{\mathcal{L}}}$ represents a loss function, which could be log loss (Equation 2.9) or 0/1 loss (Equation 2.10). $\tilde{E}_{\hat{P}_{\mathcal{L}\cup(x^*,y^*)}}$ represents the expected error of classifier when the candidate instance, $(x^*, y^*)$, is incorporated into its training data, $\mathcal{L} \cup (x^{(i)}, y)$.

Ramirez-Loaiza et al. [2016] provide an extensive empirical evaluation of common active learning strategies, comparing random sampling, uncertainty sampling, and query-by-committee using naïve Bayes and logistic regression classifiers and various performance

measures. Schein and Ungar [2007] evaluate random sampling, uncertainty sampling, query-by-committee, and variance reduction using logistic regression classifier. Settles and Craven [2008] provide an empirical comparison of uncertainty sampling and query-by-committee on sequence labeling task and propose several query strategies for selecting informative instances for sequence labeling task.

Much of the early work on active learning focused on developing strategies to select informative instances for learning and eliciting labels for the selected instances from human experts. However, humans possess knowledge beyond just the labels of instances. For example, humans have domain expertise which helps them in classifying examples. Humans can identify important feature-class correlations, provide logical rules for classification, articulate reasonings for classification, choose the right features for a particular domain, and consider factors outside the feature representation. Next, I describe related work on incorporating domain knowledge into learning.

## 2.3 Incorporating Domain Knowledge into Learning

Transmitting domain knowledge to learning systems has been studied for many years. For example, expert systems relied heavily on eliciting domain knowledge from the experts (e.g., Mycin system [12] was built through eliciting rules from the experts). Several explanation-based learning approaches (e.g., [26] and [70]) were developed to utilize domain knowledge to generalize target concepts using a single training example, and relied on domain experts to provide explanations for generalization. Examples of explanation-based learning systems include GENESIS [71] and SOAR [56]. Ellman [1989] provides a survey on explanation-based learning. Several approaches have been developed for knowledge-based classifiers such as knowledge-based systems such as knowledge-based neural networks (e.g., [40], [116], and [117]), and knowledge-based support vector machines [39].

However, incorporating domain knowledge into the learning process and teaching

the classification reasonings to supervised learning algorithms is not trivial. Many supervised learning systems operate on feature-based representations of instances. For example, in document classification, instances are typically represented as feature vectors in a bag-of-words model. The domain knowledge elicited from the experts, however, often cannot be readily parsed into the representation that the underlying model can understand or operate on. The domain knowledge often refers to features rather than specific instances. Moreover, the domain knowledge is often at a higher level than instances, and sometimes, the domain knowledge is provided as unstructured information, such as free-form text entries.

**2.3.1 Interactive Machine Learning.** Another related area is the work on Interactive Machine Learning (IML) in which the humans interact with machine learning algorithms to observe their behavior, usually in the form of predictions or output, and provide feedback, usually in the form of labels, corrections, or demonstrations. IML aims to provide transparency into the working of a machine learning algorithm to have a better understanding of the model's performance.

Many systems have been built using interactive machine learning. For example, Ware et al. [2001] presented a method that allows users to construct a decision tree classifier interactively by letting the user select the attributes to split the data at a node using data visualizer. Fails and Olsen [2003] presented a powerful interactive machine learning model, Crayons, that allows users to create image classifiers interactively, such that the users can paint small areas of objects in an image that they want to classify. Fogarty et al. [35] presented a system, CueFlik, that allows users to create their own rules for re-ranking images based on their visual characteristics. Kabra et al. [50] presented an interactive machine learning algorithm, JAABA, in which users can build a classifier using active learning. In their system, the users can annotate animal behaviors in the video frames that contain images of animals exhibiting some behavior. The users annotate video frames that

they are most confident in labeling. In addition, they allow users to inspect the classifier's performance on any frame chosen by the user. Their system can find frames for which the classifier has incorrect prediction or has low confidence, so that such frames can be presented to the user for labeling and re-training the classifier.

An area related to interactive machine learning is Human-Computer Interaction (HCI), which studies the design of interfaces for computer systems and aims to improve the usability and effectiveness of the systems [27], [49], [106]. However, human-computer interaction focuses on users' needs for usability, trust, and understanding of the system, whereas IML focuses on building and improving machine learning systems through rich interactions with the users.

**2.3.2 Intelligent User Interfaces.** An area that intersects Interactive Machine Learning and Human Computer Interaction is designing intelligent user interfaces to improve system performance or usability [110]. Stumpf et al. [108] conducted a user study to see how the system can provide rich feedback to the user and what kind of user feedback can be assimilated into the learning algorithms. Following up on their findings, they presented a method to include rich feedback from users to build an email classification system [107]. In their system, the classifier explains its classification of emails by highlighting top ten words in an email, and the users provide feedback by selecting keywords in an email and adjusting the weights of the keywords in an email. Kumar et al. [55] utilized active learning to collect user feedback and build a model to predict errors in health insurance claims. Their system provided explanations for the selected instance of health insurance claim to the auditors by highlighting features with higher influence scores, where influence scores were calculated by multiplying the feature values by feature weights using a support vector machines classifier.

**2.3.3 Feature Annotation.** Recent work on active learning focused on eliciting rich feedback from the experts either instead of or in addition to the labels for instances, to

speed-up the annotation process. Feature annotation work asked the experts to annotate features as relevant/irrelevant or on a likeart scale (e.g., [4], [31], [81], [104], [105], and [107]). Much of this work focused on eliciting feedback on features for text classification task.

Traditional supervised learning algorithms such as naïve Bayes, logistic regression, and support vector machines, are able to handle only labeled instances, that is, $\langle x^{(i)}, y^{(i)} \rangle$ pairs and they cannot readily handle the elicited feature annotations directly into the training of classifiers. In order to incorporate feature annotations into the training of supervised learning algorithms, several classifier-specific approaches were developed. For example, Raghavan et al. [2006] asked users to mark features as relevant/irrelevant and incorporated feature annotation into the training of support vector machines by re-weighting the features, such that the relevant features are weighted 10 times higher than the irrelevant features. Melville and Sindhwani [2009] developed a pooling multinomial naïve Bayes approach, where two multinomial naïve Bayes models are trained, one on labeled features and the other on labeled instances, and the two models are combined using linear pooling [67]. Small et al. [2011] presented an approach for incorporating feature annotations into the training of support vector machines for text classification. In their approach, they asked labelers to provide a ranked list of features, and added additional constraints into support vector machines to exploit the ranked features. Similarly, Stumpf et al. [2008] converted feature annotations into a set of constraints and learned the parameters of a naïve Bayes classifier through a constraint optimization procedure that maximized the likelihood of data given the constraints. Das et al. [2013] utilized locally-weighted logistic regression to incorporate feature annotations into logistic regression classifier by locally fitting a logistic function on instances around a small neighborhood of test instances and taking into account the annotated features.

**2.3.4 Rationale-Based Learning.** Another line of work that is related to feature annota-

tion is the recent work on eliciting rationales for classification, which often corresponds to highlighting a piece of text in text classification or highlighting feature values in feature-valued representations to provide a reason for choosing a particular label, and incorporating them into the learning.

Zaidan et al. [2007] and Zaidan et al. [2008] asked users to provide rationales by highlighting phrases in movie reviews that support the chosen label. They incorporated rationales into learning by specifying additional constraints for support vector machines, creating contrast examples for the rationale features, and incorporating them into learning of support vector machines. Donahue and Grauman [2011] extended the approach in Zaidan et al. [2007] to incorporate rationales for visual recognition task. They proposed eliciting two forms of visual rationales from the labelers. First, they asked labelers to mark spatial regions in an image as rationales for choosing a label for the image. Second, they asked labelers to comment on the nameable visual attributes (based on a predefined vocabulary of visual attributes) that influenced their choices the most. For both forms of rationales, they created contrast examples that lack the rationale and incorporated the contrast examples and pseudo-examples into the training of support vector machines. More recently, Zhang et al. [2016] presented a method to incorporate rationales for text classification into Convolutional Neural Networks (CNN).

In feature annotation work, users are asked to annotate features independent of the instances in which they appear, whereas in rationale elicitation work, users provide reasons for their classification of a particular instance. Since rationales correspond to features in the instances, rationales can be incorporated into learning by utilizing the approaches for feature annotation. The main difference between feature annotation work and rationale elicitation work is that in feature annotation work, the features are weighted globally, whereas in rationale elicitation work, features are tied to particular instances for which they were provided as rationales. However, much of the feature annotation work and rationale

elicitation work are specific to a particular classifier, such as support vector machines.

**2.3.5 Tandem Learning.** Another line of related work is active learning with both instance and feature annotations. Raghavan et al. [2006] and Raghavan and Allan [2007] proposed tandem learning, where at each iteration of active learning, the learner presents instances and features for a human to label, and incorporated instance annotations and feature feedback into support vector machines. They asked labelers to provide feedback on features as to whether the features are discriminative or not. They incorporated feature feedback by scaling all the important features by a higher weight, and scaling all the other features by a lower weight.

Attenberg et al. [2010] presented a unified approach to interleave feature annotation and instance annotation and determine which features or instances the classifier will benefit the most by learning its labels and presented a pooling multinomials approach to incorporate labeled instances and labeled features into multinomial naïve Bayes. Parkash and Parikh [2012] proposed a method to incorporate labels and feature feedback for image classification task. They asked users to provide the labels of images, and for each image that was predicted incorrectly by the classifier, they asked users to provide explanations in the form of attribute-based feedback. The attribute feedback was based on relative attributes [75] that are mid-level concepts that can be shared across various class labels. In their approach, the feature feedback provided by the labelers is propagated to other unlabeled images that match the explanation provided by the labelers.

**2.3.6 Active Learning for Anomaly Detection.** Anomaly detection methods aim to find unusual patterns that do not conform to the normal patterns in the data. Unsupervised anomaly detection techniques such as clustering [46], [111] and one-class support vector machines [90] detect anomalies or outliers as instances that are farthest away from the rest of the data. Supervised anomaly detection methods treat anomalies or outliers as instances of one class and the rest of the data as the other class. Since a majority of instances are

usually assumed to be normal and only a few instances correspond to anomalies, anomaly detection tasks have extremely imbalanced class distributions. A solution to address the class-imbalance issue in anomaly detection tasks is to over-sample the instances of minority class, e.g., using SMOTE technique [16], in which synthetic examples are created along the lines joining a minority class instance to other nearest neighboring minority class instances. Zhang et al. [2016] recently proposed a variant of SMOTE to over-sample the minority class instances based on relative change in overall certainty of classifier due to adding an instance of minority class. Chandola et al. [2009] present a survey of common anomaly detection methods.

Active learning has been used in several anomaly detection tasks, such as detecting false health insurance claims [55], detecting intrusion in networks [41], and detecting anomalies in images from astrophysics domain [79]. Pelleg and Moore [2004] presented an active learning framework to discover rare, but *useful*, anomalies as opposed to statistical anomalies with the help (in the form of class labels) from a human expert. They assumed that the classes of rare events are known in advance and used a mixture model approach to surface the most interesting events in each iteration of active learning. Pichara and Soto [80] presented an active learning framework along with subspace clustering for anomaly detection. In their approach, first, a Bayes network is learned that identifies the candidate anomalies. Then, a subspace clustering method is applied to identify relevant subsets of dimensions that describe the anomaly. Finally, a probabilistic active learning scheme, based on properties of Dirichlet distribution, uses the feedback from an expert to efficiently search for relevant anomalies.

While most of the existing methods focused on finding outliers from data, relatively little work has been done on finding the root cause of why a data point is deemed an outlier by the algorithm. One such method is the outlier explanation work by Micenkova et al. [2013], in which they proposed a method to determine possible explanations for an

outlier expressed in the form of subspaces in which the given outlier shows separability from the inliers. This algorithm complements existing outlier detection algorithms by providing additional information about the detected outliers. In a more recent work, Dang et al. [23] addressed the problem of outlier interpretation. Their method is based on subspace identification which allows for discriminating between regular objects and the outliers in a lower dimensional space. Mathematically, the authors proposed a variant of spectral graph embedding which provides an optimal solution for subspace learning. Experimental results on real world datasets show the efficacy of discriminative features for both outlier detection and interpretation. Unlike an active learning approach which combines user feedback with modeling, their method provides a ranked list of outliers, and hence can be time consuming for users to unearth potentially *relevant* outliers.

**2.3.7 Active Inference.** Another line of work that is related to active learning is active inference [87], which assumes that a classification model has been trained offline, but there is an option to acquire labels during the inference phase. Bilgic and Getoor [2009] and Bilgic and Getoor [2010] presented label acquisition strategies for collective classification. More recently, Komurlu et al. [2014] and Komurlu et al. [2016] proposed a method for active inference for tissue engineering experiments to determine the optimal time to stop biological experiments. Similarly, Komurlu and Bilgic [2016] proposed a method for active inference for wireless sensor networks to determine which sensor readings would be useful to improve predictions on all unseen sensor readings.

The key difference between active learning and active inference is that in active learning, the objective is to gather labels during the learning phase and optimize model's performance in the learning phase, whereas in active inference, the objective is to collect data during the inference phase and maximize the model's performance. In active learning, the data is typically assumed to be *i.i.d.*, whereas in active inference, the variables are assumed to be correlated. For example, in social networks data, the users having a common

connection are correlated and knowing information, such as political view, of one user can help in predicting this information for the neighbors. The goal of both active learning and active inference is to collect data to maximize the model's performance, however active learning tries to improve the model's performance in the learning phase, whereas active inference tries to improve the model's performance in the prediction phase.

**2.3.8 Learning Using Privileged Information.** Another advanced learning paradigm is learning using privileged information (LUPI) [118], in which the user provides additional information about training data, which is available during the training phase, but is not available during the testing phase. In LUPI paradigm, instead of $\langle x^{(i)}, y^{(i)} \rangle$ pairs, the training set consists of triplets: $\langle x^{(i)}, x^{(i)*}, y^{(i)} \rangle$, where $x^{(i)*}$ denotes any additional information about an instance $x^{(i)}$. Since the additional information is available at the training time, but not available during the test time, it is called *privileged information*.

The transfer of information from privileged data, $X^* = \left\{x^{(1)^*}, \cdots, x^{(n)^*}\right\}$, to the training data, $X = \left\{x^{(1)}, \cdots, x^{(n)}\right\}$, cannot be done directly, but is done by modifying the classification function, using privileged information as a proxy to an oracle that can provide values for slack variables, $\xi_i$, in support vector machines. Vapnik and Vashist [2009] proposed SVM+ method to incorporate privileged information into support vector machines for regression problems. Pechyony and Vapnik [2010] presented a version of SVM+ for classification problems, and Sharmanska et al. [2013] proposed SVM+ for ranking problems. Sharmanska et al. [2013] incorporated four types of privileged information for classifying objects in images: (i) attributes that describe semantic properties of objects, such as shape, color, etc., (ii) bounding box annotation that captures the exact location of an object in an image, (iii) textual description of images, and (iv) rationales for image classification proposed by Donahue and Grauman [2011].

**2.3.9 Active Learning Assumptions.** Much of the work on active learning made several simplified assumptions. For example, they assumed uniform time and cost for all the

queries and assumed that the users are oracles, i.e., the users always provide correct answers to all the queries. However, in practice, the queries can have different costs and require different annotation times, and the experts can be fallible and reluctant. Settles et al. [2008] showed on several tasks that annotation times are variable for different domains, annotation times of experts are different, and annotation times of the first few instances are longer because the annotators are unfamiliar with the annotation task and interface early on.

Several cost-sensitive active learning approaches [96] were developed to lift these active learning assumptions. For example, Tomanek and Hahn [2010] presented several active learning approaches considering the real annotation times needed by humans (instead of uniform cost for annotating tokens) for named-entity recognition task. Donmez and Carbonell [2008] presented a proactive learning approach considering reluctant and fallible experts. Kapoor et al. [2007] presented a decision-theoretic approach for active learning that selects instances for labeling that have the highest value of information, calculated as the difference between the risk of misclassification and the cost of obtaining a label. Ramirez-Loaiza [2016] presented several anytime active learning algorithms to reduce the annotation cost by showing relevant pieces of information from an instance, rather than showing the entire instance, to the users. Ramirez-Loaiza et al. [2013] proposed a decision-theoretic approach to select subinstances of documents, based on value of information of subinstances, for annotation. Ramirez-Loaiza et al. [2014] extended the work in Ramirez-Loaiza et al. [2013] and presented a dynamic anytime active learning method that takes into account the chances of getting a neutral label from a user and conducted a user study to test their approach.

# CHAPTER 3

# FRAMEWORK TO MAKE THE ACTIVE LEARNER TRANSPARENT

In this chapter, I discuss how we make the active learner transparent to provide its reasons for querying an instance. Specifically, we look into uncertainty sampling, an active learning strategy that selects instances for which the learner is most uncertain about the label, and dig deeper into why it might be uncertain on the instances. I present a framework to make the active learner transparent, which we call the *evidence-based framework*, that can uncover the reasons for model's uncertainty on instances. I show that the learner can be uncertain on an instance because it has strong, but conflicting, evidence for both classes or it can be uncertain on an instance because of insufficient evidence for each class. I show how the active learner can utilize the evidence-based framework to compute the evidence that an instance provides for each class, and use the evidences to select better instances for labeling. Finally, I discuss why distinguishing between different types of uncertainties matters and provide analytical and empirical justifications to explain why selecting the instances with conflicting-evidence is better for learning than selecting instances with insufficient-evidence.

This chapter is based on the work that I did with my advisor, Dr. Mustafa Bilgic. This work was published in the IEEE International Conference on Data Mining, 2013 [99]. An extension of this work as a journal article was published in Data Mining and Knowledge Discovery, 2016 [98]. The material from the article "Evidence-based uncertainty sampling for active learning", Volume 31, Issue 1, 2016, pp 164-202, Manali Sharma and Mustafa Bilgic, In Data Mining and Knowledge Discovery, has been included in this chapter "With permission of Springer".

## 3.1 Introduction

Uncertainty sampling is a popular active learning strategy that selects instances that lie near the decision boundary of the model. In Section 2.2.2, I described uncertainty sampling in detail. Due to its simplicity, ease of implementation, and empirical success in many domains, uncertainty sampling has been widely used as a baseline in the literature.

Traditional uncertainty sampling does not delve into the reasons for model's uncertainty on instances. In this chapter, we use the evidence-based framework to analyze why the model might be uncertain about an instance. Specifically, we focus on two types of uncertainties. In the first case, the model is uncertain due of presence of strong, but conflicting evidence for each class. We call this type of uncertainty as *conflicting-evidence uncertainty*. In the second case, the model is uncertain due to insufficient evidence for either class. We call this type of uncertainty as *insufficient-evidence uncertainty*.

For example, for a heart-disease diagnosis, the model can be uncertain because one lab test result strongly suggests presence of heart-disease, while another lab test result strongly suggests absence of heart-disease. In this case, the model is uncertain because of conflicting evidence for both classes. Another reason that the model can be uncertain is that none of the lab test results provide any conclusive evidence for presence or absence of heart-disease. In this case, the model is uncertain because of insufficient evidence for either class. Similarly, in a bag-of-words document classification task, the model can be uncertain because some terms in a document provide strong evidence for one class, while some other terms provide strong evidence for the other class, which makes the model uncertain due to conflicting evidence. On the other hand, the model can be uncertain because none of the terms provide conclusive evidence for either class, which represents model's uncertainty due to insufficient evidence. Figure 3.1 depicts this phenomenon for binary classification.

We provide a mathematical formalism to make a distinction between these two types of uncertainties. We introduce an evidence-based framework to capture the amount of evidence for each class provided by an instance, which facilitates distinguishing be-

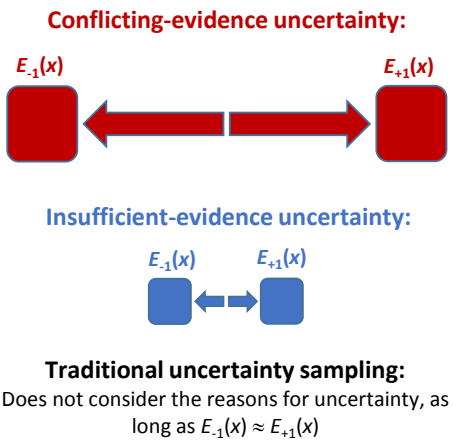



Figure 3.1. Conflicting-evidence vs. insufficient-evidence uncertainty. Conflicting-evidence uncertainty represents a model's uncertainty on an instance due to strong evidence for each class, whereas insufficient-evidence uncertainty represents a model's uncertainty on an instance due to insufficient evidence for each class. Traditional uncertainty sampling does not care about the reasons for uncertainty, and picks the most uncertain instance.

tween these two types of uncertainties. Through empirical evaluations on several real-world datasets, we show that distinguishing between conflicting-evidence uncertainty and insufficient-evidence uncertainty makes a huge difference to the performance of active learning. We show that conflicting-evidence uncertainty provides the most benefit for learning, drastically outperforming both traditional uncertainty sampling and insufficient-evidence uncertainty sampling.

The rest of the chapter is organized as follows. In Section 3.2, we provide background on active learning and uncertainty sampling, and provide formulation for the evidence-based framework. In Section 3.3, we provide experimental details and results comparing conflicting-evidence uncertainty and insufficient-evidence uncertainty to the traditional uncertainty sampling. In Section 3.4, we present results of a user study that examines the users' performance while labeling instances selected by the two types of uncertainties. In Section 3.5, we present empirical and analytical justifications as to why distinguishing between conflicting versus insufficient evidence cases matters. In Section 3.6, we extend the formulation for the evidence-based framework to other classifiers and multi-

class classification. Finally, we conclude in Section 3.7.

## 3.2 Background and Problem Formulation

In this section, we first provide background on active learning and uncertainty sampling. Then we explain active learning and uncertainty sampling in the context of classification. Then we provide the formulation for the evidence-based framework for naïve Bayes.

Many active learning methods have been developed in the past two decades. A number of approaches have been proposed to select informative instances for labeling, e.g. selecting uncertain instances [57], choosing instances for which a committee of learners disagree [97], choosing representative instances [120], selecting more informative data that optimizes expected gain [62], selecting examples that minimize the expected error of the model [42], [43], [88], [122], and selecting instances that minimize the bias of the learner [19] or minimize variance of the learner [21]. We refer the reader to [94] for a survey of active learning methods.

Arguably, the most frequently utilized active learning strategy is uncertainty sampling.[1] It is often used as a baseline for comparing other active learning methods and has been shown to work successfully in a variety of domains. Example domains include text classification [10], [47], [57], [120], natural language processing [112], email spam filtering [91], [92], image retrieval [114], medical image classification [48], robotics [14], information retrieval [125], dual supervision [104], and sequence labeling Settles and Craven [2008], among many others.

Even though uncertainty sampling is frequently utilized, it is known to be susceptible to noise and outliers [88]. A number of approaches have been proposed to make it more robust. For example, Settles and Craven [2008] weighted the uncertainty of an instance by its density to avoid outliers, where density of the instance is defined as average

[1] 1,507 citations on Google Scholar on April $4^{\text{th}}$, 2016

similarity to other instances. Zhu et al. [2008] used a K-Nearest-Neighbor-based density measure to determine whether an unlabeled instance is an outlier. Xu et al. [2003] and Donmez et al. [2007] proposed a hybrid approach to combine representative sampling and uncertainty sampling. Other approaches used the cluster structure of the domain to choose more representative examples [10], [73]. Senge et al. [2014] presented an approach to distinguish between aleatoric and epistemic uncertainties using possibility theory, in which uncertainty is modeled in terms of two measures, namely possibility and necessity. The aleatoric uncertainty results due to variability in the outcome of an experiment due to inherently random effects, and epistemic uncertainty is caused by lack of knowledge. While epistemic uncertainty can be reduced by gathering more information, aleatoric uncertainty cannot be reduced further.

Our work is orthogonal to these approaches. We are not providing yet another alternative approach to improve uncertainty sampling, but instead we are highlighting that distinguishing between the two types of uncertainties (conflicting-evidence vs. insufficient-evidence) has a big impact on active learning. One can imagine combining uncertainty sampling, density weighting, and conflicting-evidence uncertainty methods because they are not mutually exclusive.

Next, we explain active learning and uncertainty sampling and introduce the notations that will be used throughout the chapter.

**3.2.1 Active Learning.** Let the uppercase $X$ denote the random variable representing an instance and the lowercase $x$ represent a particular instantiation of $X$. Each instance is described as a vector of $f$ attributes $X \triangleq \langle X_1, X_2, \cdots, X_f \rangle$. Similarly, let the uppercase $Y$ represent the class variable of the instance and let the lowercase $y$ represent a particular instantiation of $Y$. Each $X_i$ can be real-valued or discrete whereas $Y$ is discrete; in this article, we focus on the binary case, where $Y \in \{-1, +1\}$. In the pool-based active learning setup, we are given a small set of instances whose labels are known: $\mathcal{L} = \{\langle x^{(i)}, y^{(i)} \rangle\}$, and

a much larger collection of unlabeled instances whose labels are unknown: $\mathcal{U} = \{\langle x^{(i)}, ?\rangle\}$.

A pool-based greedy active learning algorithm iteratively selects an informative instance $\langle x^*, ?\rangle \in \mathcal{U}$ to obtain its label $y^*$ from an expert, and incorporates the new labeled instance $\langle x^*, y^*\rangle$ into $\mathcal{L}$. The informative instance, $\langle x^*, ?\rangle$, is selected by computing *utility* of the unlabeled instances in $\mathcal{U}$, where utility can be classifier uncertainty [57], committee disagreement [97], expected reduction in error [88], etc. This process continues until a stopping criterion is met, usually until a given budget, $B$, is exhausted. Algorithm 1 describes this process more formally. The goal of active learning is to learn the correct classification function $\theta : X \rightarrow Y$ by carefully choosing which instances are labeled, subject to budgetary constraints.

**3.2.2 Uncertainty Sampling.** Uncertainty sampling selects instances for which the current model is most uncertain how to label [57]. These instances correspond to the ones that lie close to the decision boundary of the current model. I explained uncertainty sampling in detail in Section 2.2.2, and I briefly describe it here.

Uncertainty of an underlying model can be measured in several ways. One approach is to use conditional entropy:

$$x^* = \underset{x^{(i)} \in \mathcal{U}}{\operatorname{argmax}} - \sum_{y \in Y} P_\theta(y|x^{(i)}) \log \left(P_\theta(y|x^{(i)})\right) \tag{3.1}$$

where $P_\theta(y|x^{(i)})$ is the probability that instance $x^{(i)}$ has label $y$. Another approach is to use maximum conditional:

$$x^* = \underset{x^{(i)} \in \mathcal{U}}{\operatorname{argmax}} \left(1 - \max_{y \in Y} P_\theta(y|x^{(i)})\right) \tag{3.2}$$

The last approach we discuss uses margin of confidence:

$$x^* = \underset{x^{(i)} \in \mathcal{U}}{\operatorname{argmin}} \left( P_\theta(y_m|x^{(i)}) - P_\theta(y_n|x^{(i)}) \right) \tag{3.3}$$

where, $y_m$ is the most likely label and $y_n$ is the next likely label for $x^{(i)}$. More formally, $y_m = \underset{y \in Y}{\operatorname{argmax}}\, P_\theta(y|x^{(i)})$ and $y_n = \underset{y \in Y \setminus y_m}{\operatorname{argmax}}\, P_\theta(y|x^{(i)})$.

When the task is binary classification, that is when $Y \in \{+1, -1\}$, all three uncertainty approaches (Equation 3.1, Equation 3.2 and Equation 3.3) rank instances in the same order and prefer the same uncertain instance, i.e. the instance for which $P_\theta(+1|x^{(i)}) = P_\theta(-1|x^{(i)}) = 0.5$. In this article, we distinguish between the two types of uncertainties that we define next.

**3.2.3 Problem Formulation.** In this section, we define *evidence* that an attribute value provides for a class in the evidence-based framework. The evidence, in its most general form, is the amount of contribution that an attribute value provides to the prediction of belonging to a particular class. Each classifier computes the prediction for a test instance differently, and hence the evidence that an attribute value of an instance provides for a class depends on the classifier. In this section, we provide the formalism of evidence using naïve Bayes classifier. The formalism of evidence for logistic regression and support vector machines is provided later in Section 3.6.

**3.2.3.1 Evidence using naïve Bayes.** A naïve Bayes classifier uses the Bayes rule to compute $P(Y|X)$ and assumes that the attributes $X_j$ are conditionally independent given $Y$:

$$P(Y|x) = P(Y|x_1, x_2, \cdots, x_f) = \frac{P(Y) \prod\limits_{x_j} P(x_j|Y)}{P(x_1, x_2, \cdots, x_f)} \tag{3.4}$$

The instance $x$ can be classified based on the ratio of $\frac{P(+1|x)}{P(-1|x)}$:

$$Y = \begin{cases} +1 & \text{if } \left( \frac{P(+1)}{P(-1)} \prod_{x_j} \frac{P(x_j|+1)}{P(x_j|-1)} \right) > 1 \\ -1 & \text{otherwise} \end{cases} \tag{3.5}$$

From Equation 3.5, it follows that the attribute value $x_j^{(i)}$ of the instance $x^{(i)}$ provides evidence for the positive class if $\frac{P(x_j^{(i)}|+1)}{P(x_j^{(i)}|-1)} > 1$, and it provides evidence for the negative class otherwise.

Note that it does not make sense to talk about the evidence the attribute $X_j$ itself provides. Rather, the particular instantiation $x_j$ provides evidence for one class or the other (or for none of the classes). For example, the cholesterol test itself does not provide evidence for presence or absence of heart-disease; rather, the outcome of the cholesterol test (e.g., high or low) provides the evidence for presence/absence of heart-disease. Hence, we define the evidence at the instance level, rather than the variable level.

Let $\mathcal{P}_{x^{(i)}}$ and $\mathcal{N}_{x^{(i)}}$ be two sets, such that $\mathcal{P}_{x^{(i)}}$ contains the attribute values of the instance $x^{(i)}$ that provide evidence for the positive class and $\mathcal{N}_{x^{(i)}}$ contains the attribute values of the instance $x^{(i)}$ that provide evidence for the negative class:

$$\mathcal{P}_{x^{(i)}} \triangleq \left\{ x_j^{(i)} \mid \frac{P(x_j^{(i)}|+1)}{P(x_j^{(i)}|-1)} > 1 \right\}$$

$$\mathcal{N}_{x^{(i)}} \triangleq \left\{ x_k^{(i)} \mid \frac{P(x_k^{(i)}|-1)}{P(x_k^{(i)}|+1)} > 1 \right\}$$

Note that in these definitions, the numerator for $\mathcal{P}_{x^{(i)}}$ is $P(x_j^{(i)}|+1)$ and numerator for $\mathcal{N}_{x^{(i)}}$ is $P(x_k^{(i)}|-1)$.

The total evidence for the instance $x^{(i)}$ to belong to the positive class is:

$$E_{+1}(x^{(i)}) = \prod_{x_j^{(i)} \in \mathcal{P}_{x^{(i)}}} \frac{P(x_j^{(i)}|+1)}{P(x_j^{(i)}|-1)} \tag{3.6}$$

and, the total evidence for the instance $x^{(i)}$ to belong to the negative class is:

$$E_{-1}(x^{(i)}) = \prod_{x_k^{(i)} \in \mathcal{N}_{x^{(i)}}} \frac{P(x_k^{(i)}|-1)}{P(x_k^{(i)}|+1)} \tag{3.7}$$

With these definitions, we can rewrite the classification rule for naïve Bayes as:

$$Y = \begin{cases} +1 & \text{if } \left(\frac{P(+1)}{P(-1)} \frac{E_{+1}(x^{(i)})}{E_{-1}(x^{(i)})}\right) > 1 \\ -1 & \text{otherwise} \end{cases} \tag{3.8}$$

**3.2.3.2 Conflicting-Evidence vs. Insufficient-Evidence Uncertainty.** In this article, we investigate whether the evidence-based framework provides a useful criteria to distinguish between the uncertain instances and whether such an approach leads to more or less effective active learning.

Traditional uncertainty sampling picks the most uncertain instance, $x^{(i)}$, for which $E_{+1}(x^{(i)}) \approx E_{-1}(x^{(i)})$, regardless of the magnitudes of $E_{+1}(x^{(i)})$ and $E_{-1}(x^{(i)})$. In this article, we analyze if the magnitudes of $E_{+1}(x^{(i)})$ and $E_{-1}(x^{(i)})$ have an impact on learning when $E_{+1}(x^{(i)}) \approx E_{-1}(x^{(i)})$. Specifically, we consider two cases:

- The model is uncertain because of strong, but conflicting evidence for both classes. This represents the case when both $E_{+1}(x^{(i)})$ and $E_{-1}(x^{(i)})$ are equal and large.
- The model is uncertain because of insufficient evidence for either class. This represents the case when both $E_{+1}(x^{(i)})$ and $E_{-1}(x^{(i)})$ are equal and small.

When $E_{+1}(x^{(i)})$ and $E_{-1}(x^{(i)})$ are equal, there are a number of choices to mathematically determine if both $E_{+1}(x^{(i)})$ and $E_{-1}(x^{(i)})$ are small or large by ranking all the uncertain instances according to one of the Equations 3.9, 3.10, 3.11, or 3.12.

$$\underset{x^{(i)} \in \mathcal{U}}{\operatorname{argmax}} E_{+1}(x^{(i)}) \times E_{-1}(x^{(i)}) \tag{3.9}$$

$$\underset{x^{(i)} \in \mathcal{U}}{\operatorname{argmax}} E_{+1}(x^{(i)}) \tag{3.10}$$

$$\underset{x^{(i)} \in \mathcal{U}}{\operatorname{argmax}} E_{-1}(x^{(i)}) \tag{3.11}$$

$$\underset{x^{(i)} \in \mathcal{U}}{\operatorname{argmax}} \min(E_{+1}(x^{(i)}), E_{-1}(x^{(i)})) \tag{3.12}$$

Note that when $E_{+1}(x^{(i)})$ and $E_{-1}(x^{(i)})$ are equal, Equations 3.9, 3.10, 3.11, and 3.12 will all provide the same ranking for uncertain instances, and it does not matter which one of these functions is chosen to rank the uncertain instances based on evidence. [2] In Section 3.3.2, we present the results using multiplication of the evidence for each class, i.e. according to Equation 3.9.

Regardless of whether we want to maximize or minimize $E_{+1}(x) \times E_{-1}(x)$, we want to guarantee that the underlying model is uncertain about the chosen instance. To achieve uncertainty, we first rank the instances $x^{(i)} \in \mathcal{U}$ in decreasing order of their uncertainty score (measured by Equation 2.1), and work with the top $t$ instances, where $t$ is a hyper-

[2] In practice, however, $E_{+1}(x^{(i)})$ and $E_{-1}(x^{(i)})$ might not be exactly equal to each other for all uncertain instances, and hence the ranking of uncertain instances based on evidence according to Equations 3.9, 3.10, 3.11, and 3.12 may be different.

parameter. Formally, let $\mathcal{S}$ be the set of top $t$ uncertain instances. Conflicting-evidence uncertainty will prefer instances where both $E_{+1}(x^{(i)})$ and $E_{-1}(x^{(i)})$ are large:

$$x^* = \underset{x^{(i)} \in \mathcal{S}}{\operatorname{argmax}} E_{+1}(x^{(i)}) \times E_{-1}(x^{(i)}) \tag{3.13}$$

and, insufficient-evidence uncertainty will prefer instances where both $E_{+1}(x^{(i)})$ and $E_{-1}(x^{(i)})$ are small:

$$x^* = \underset{x^{(i)} \in \mathcal{S}}{\operatorname{argmin}} E_{+1}(x^{(i)}) \times E_{-1}(x^{(i)}) \tag{3.14}$$

## 3.3 Experimental Methodology and Results

We designed our experiments to test whether distinguishing between the conflicting-evidence and insufficient-evidence uncertain instances makes a difference to the performance of active learner. We experimented with the following approaches:

1. *Random Sampling* (RND): This is a common baseline for active learning, in which instances are picked at random from the set of candidate unlabeled instances.

2. *Uncertainty Sampling* - $1^{\text{st}}$ (UNC-1): This is the traditional uncertainty sampling method that picks the instance for which the underlying model is most uncertain, as defined in Section 3.2.2.

3. *Conflicting-Evidence Uncertainty* (UNC-CE): Among the top $t$ uncertain instances, this method picks the instance for which the model is uncertain due to conflicting evidence (as defined in Equation 3.13).

4. *Insufficient-Evidence Uncertainty* (UNC-IE): Among the top $t$ uncertain instances, this method picks the instance for which the model is uncertain due to insufficient evidence (as defined in Equation 3.14).

5. *Uncertainty Sampling* - $t^{\text{th}}$ (UNC-u): Among the top $t$ uncertain instances, this

method picks the $t^{\text{th}}$ most uncertain instance. UNC-CE and UNC-IE methods pick one uncertain instance from the top $t$ uncertain instances according to the amount of evidence they provide. If UNC-CE and/or UNC-IE are better than UNC-1, then this result would suggest that different types of uncertainties matter. Similarly, if UNC-CE and/or UNC-IE are worse than UNC-u, then this result would also suggest that different types of uncertainties matter.

We experimented with eight publicly available datasets. We chose four medium-imbalanced (minority class% $> 10\%$) and four highly-imbalanced (minority class% $\leq 10\%$) datasets. The datasets include four active learning challenge datasets [45] (Ibn Sina, Nova, Zebra, and Hiva), and four additional datasets: LetterO [38], Calif. Housing [74], Spambase [36], and a thyroid disease dataset, Sick [36]. The description of these datasets is provided in Table 3.1. We evaluated the five methods using three performance measures: AUC, accuracy, and F1. We computed F1 as a harmonic mean of precision and recall using the minority class as positive labels. We computed AUC for all the datasets, accuracy for only medium-imbalanced datasets (the top four in Table 3.1) and F1 for only highly-imbalanced datasets (bottom four in Table 3.1).

**3.3.1 Parameters and Repeatability.** We performed five-fold cross validation and the train split was treated as the unlabeled set, $\mathcal{U}$. 10 instances (five from each class) were chosen randomly and used as the initially labeled set, $\mathcal{L}$. For each fold, the experiment was repeated five times using different sets of 10 randomly chosen instances at bootstrap. At each iteration of active learning, the methods pick only one instance to be labeled. The budget, $B$, in Algorithm 1 was set to $500$ instances. UNC-CE and UNC-IE operate within top $t$ uncertain instances, as described in Section 3.2.3.2. We experimented with $t = 5$, 10, and 20. We evaluated each method using a naïve Bayes classifier with Laplace smoothing. To speed up the experiments, at each iteration we computed uncertainty over a set of randomly sub-sampled $250$ instances, which is a common practice in active learning.

Table 3.1. Description of the datasets: the domain, number of instances, number of features, types of features, and the percentage of minority class in the datasets. The datasets are sorted in increasing order of class imbalance.

| **Dataset** | **Domain** | **# of Instances** | **# of Features** | **Types of Features** | **Min. %** |
|---|---|---|---|---|---|
| Spambase | Email classification | 4,601 | 57 | Numeric | 39.4% |
| Ibn Sina | Handwriting recognition | 20,722 | 92 | Numeric | 37.8% |
| Calif. Housing | Social | 20,640 | 8 | Numeric | 29% |
| Nova | Text processing | 19466 | 16969 | Binary | 28.4% |
| Sick | Medical | 3772 | 29 | Numeric + Binary | 6.1% |
| Zebra | Embryology | 61,488 | 154 | Numeric | 4.6% |
| LetterO | Letter recognition | 20,000 | 16 | Numeric | 4% |
| Hiva | Chemoinformatics | 42,678 | 1617 | Binary | 3.5% |

The source code for evidence-based framework for naïve Bayes is available at http://www.cs.iit.edu/~ml/code/.

**3.3.2 Results.** In this section, we present the results for the five strategies presented in the beginning of Section 3.3 and show that distinguishing between the two types of uncertainties (conflicting-evidence uncertainty and insufficient-evidence uncertainty) makes a huge difference to the performance of active learning. We compare UNC-CE and UNC-IE strategies with both UNC-1 and UNC-u strategies. We use RND as a reference for the UNC-1 strategy.

We present the learning curves for RND, UNC-1, UNC-CE, UNC-IE, and UNC-u using $t$=10. The learning curves for UNC-CE, UNC-IE, and UNC-u with $t = 5$ and $t = 20$ are similar and are omitted to avoid redundancy. We present the AUC results in Figure 3.2, and accuracy and F1 results in Figure 3.3; these figures show the mean performance and $\pm$ standard error. As these figures show, distinguishing between conflicting-evidence and insufficient-evidence uncertain instances has a huge impact on active learning for all

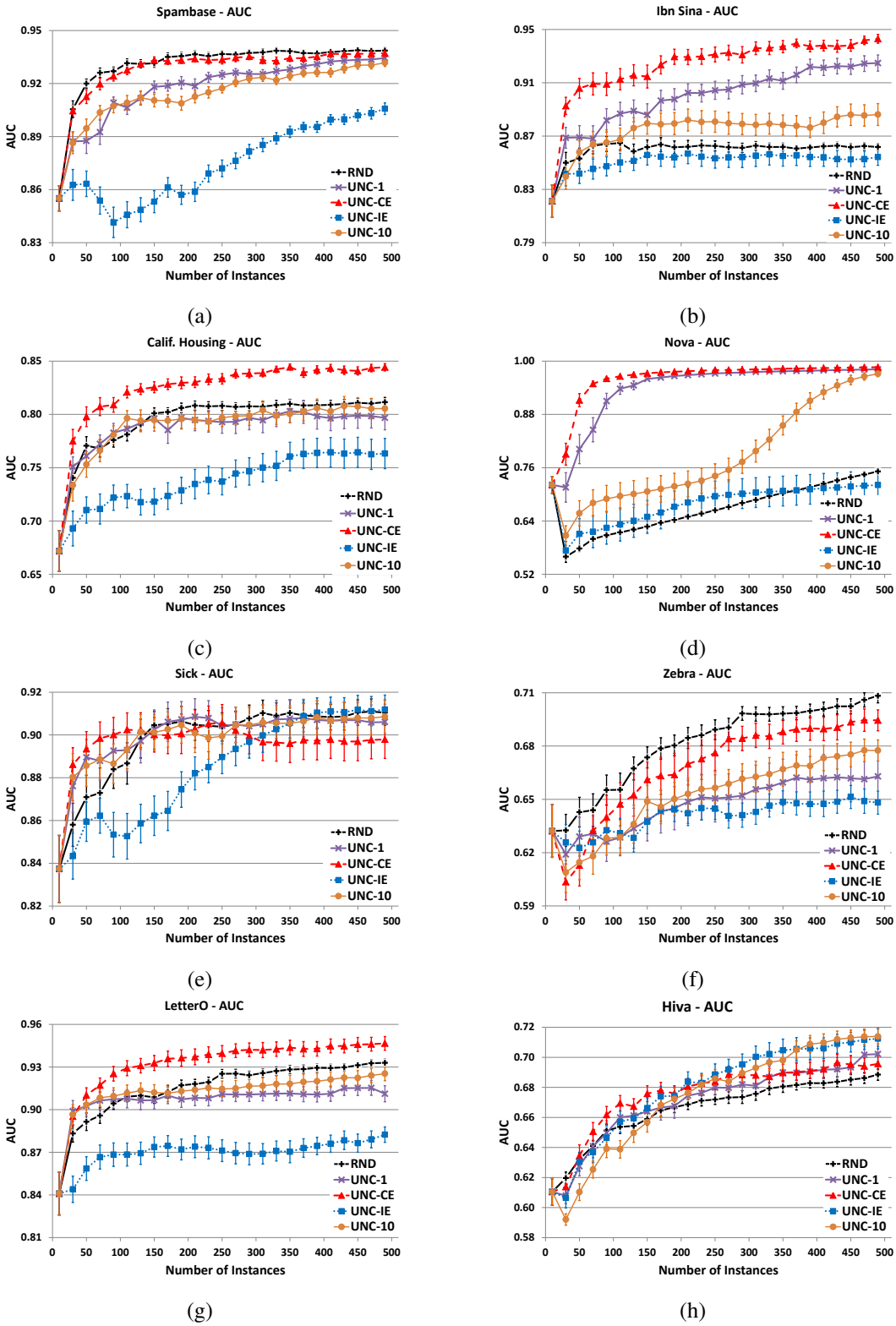



Figure 3.2. AUC results for all eight datasets. UNC-CE significantly outperforms UNC-1 on seven out of eight datasets ((a), (b), (c), (d), (f), (g), and (h)) and loses on Sick dataset (e). UNC-IE loses to UNC-1 on seven out of eight datasets ((a), (b), (c), (d), (e), (f), and (g), and wins on Hiva dataset (h).

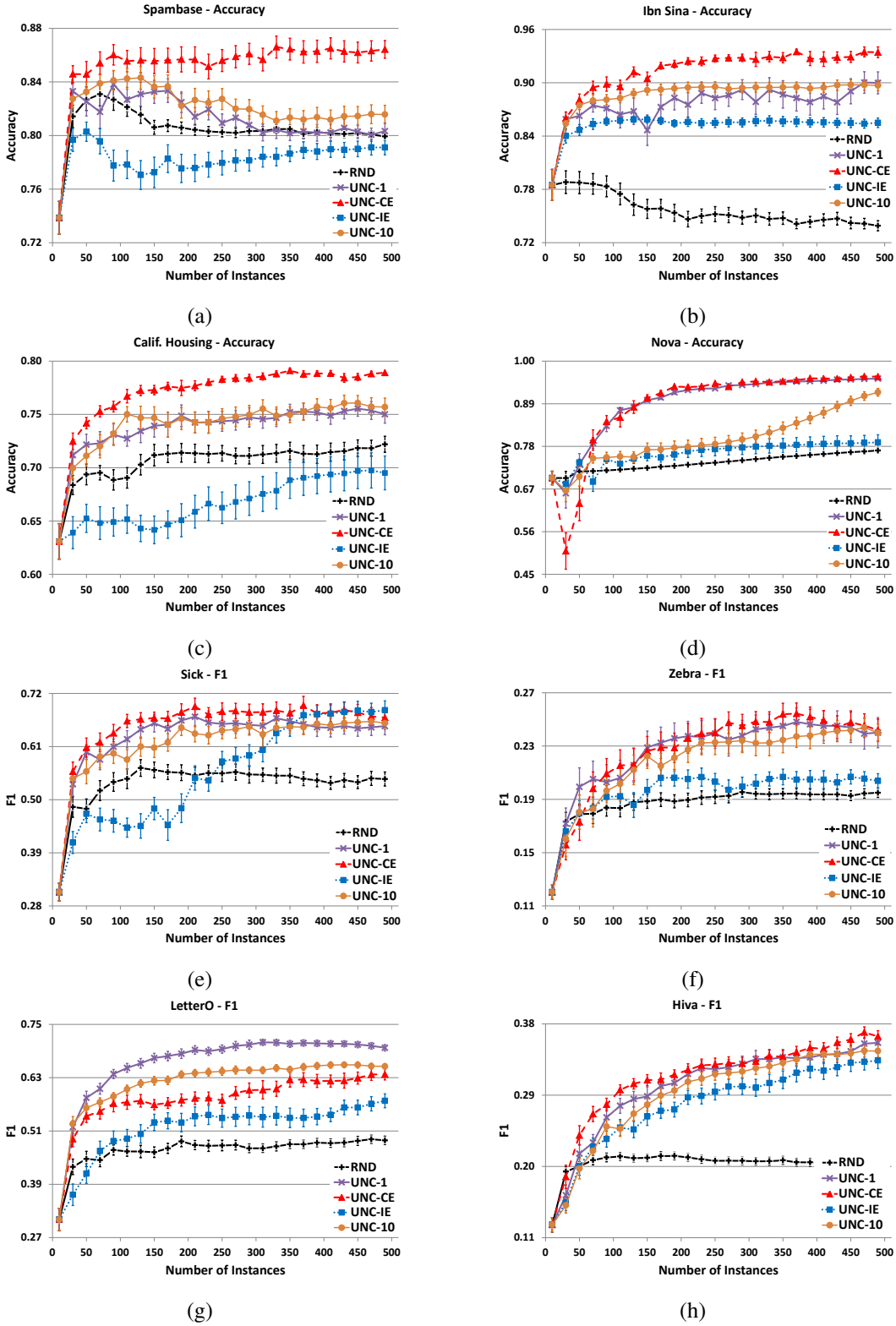



Figure 3.3. Accuracy results for four medium-imbalanced datasets (Spambase, Ibn Sina, Calif. Housing and Nova). UNC-CE outperforms UNC-1 on three datasets ((a), (b) and (c)) and loses on Nova (d). UNC-IE loses to UNC-1 on all four datasets. F1 results for four relatively skewed datasets (Sick, Zebra, LetterO and Hiva). UNC-CE outperforms UNC-1 significantly on three datasets ((e), (f) and (h)), and loses on one (g). UNC-IE loses to UNC-1 on all four datasets.

datasets and performance measures. UNC-CE wins over UNC-1 on most datasets and measures, whereas UNC-IE loses to UNC-1 on most datasets and measures.

Next, we present the results of t-tests comparing UNC-CE and UNC-IE to UNC-1 and UNC-u. Table 3.2 presents a summary of pairwise one-tailed t-tests results under significance level of $0.05$, where the pairs are learning curves of the methods. If a method is statistically significantly better than the baseline, it is a Win (W), if it is statistically significantly worse than the baseline, it is a Loss (L), otherwise it a Tie (T), meaning the differences are not statistically significant. Note that for each method, the total counts of 'W', 'T' and 'L' should add up to 8 for AUC, 4 for accuracy, and 4 for F1.

Table 3.2 presents a summary of 'Win/Tie/Loss' counts of UNC-CE and UNC-IE with $t = 5$, $t = 10$, and $t = 20$ compared to UNC-1 baseline. With respect to UNC-1, there is a clear difference between UNC-CE and UNC-IE. Our results show that UNC-CE statistically significantly wins over UNC-1 on at least 6 out of 8 datasets on AUC and loses on at most two datasets, whereas UNC-IE loses to UNC-1 on 7 out of 8 datasets on AUC. On accuracy, UNC-CE wins over UNC-1 on at least 3 out of 4 datasets, and loses on one dataset (Nova), whereas UNC-IE loses to UNC-1 on all 4 datasets. On F1, UNC-CE wins on at least 2 out of 4 datasets and loses on one dataset (LetterO), whereas UNC-IE loses to UNC-1 on all 4 datasets.

UNC-CE not only wins over UNC-1 for all performance measures, but is also quite efficient in saving the number of labeled instances required to achieve a target performance. For example, in order to achieve a target AUC of 80% for Calif. Housing dataset, UNC-1 required 199 labeled instances, UNC-CE with $t = 10$ required only 59 labeled instances (70.4% savings in the number of labels), and UNC-IE could not achieve this target AUC even with 500 labeled instances. As another example, in order to achieve a target accuracy of 90% on Ibn Sina dataset, UNC-1 required 344 labeled instances, UNC-CE with $t = 10$ required only 71 labeled instances (79.4% savings in the number of labels), and UNC-IE

Table 3.2. UNC-CE and UNC-IE with $t = 5$, $t = 10$, and $t = 20$ versus UNC-1. Number of datasets on which UNC-CE and UNC-IE significantly Win (W), Tie (T), or Lose (L) compared to UNC-1 baseline.

| **UNC-1 Baseline** | **AUC** | **ACCU** | **F1** |
|---|---|---|---|
| **Method** | **W/T/L** | **W/T/L** | **W/T/L** |
| UNC-CE with $t = 5$ | **7**/0/1 | **4**/0/0 | **3**/0/1 |
| UNC-CE with $t = 10$ | **7**/0/1 | **3**/0/1 | **3**/0/1 |
| UNC-CE with $t = 20$ | **6**/0/2 | **3**/0/1 | **2**/1/1 |
| UNC-IE with $t = 5$ | 1/0/**7** | 0/0/**4** | 0/0/**4** |
| UNC-IE with $t = 10$ | 1/0/**7** | 0/0/**4** | 0/0/**4** |
| UNC-IE with $t = 20$ | 1/0/**7** | 0/0/**4** | 0/0/**4** |

with $t = 10$ could not achieve this target accuracy even with 500 labeled instances. On Sick dataset, in order to achieve a target F1 of 65%, UNC-1 required 127 labeled instances, UNC-CE with $t = 10$ required only 100 labeled instances (21.3% savings in the number of labels), and UNC-IE with $t = 10$ required 345 labeled instances to achieve this target F1.

Next, we compared UNC-CE and UNC-IE to UNC-u with $t = 5$, $t = 10$, and $t = 20$. Table 3.3 presents the 'Win/Tie/Loss' results using UNC-u as the baseline. We observe that UNC-CE significantly outperforms UNC-u on almost all datasets, which is not surprising because even a strategy that selects instances randomly from the top $t$ uncertain instances has the potential to outperform UNC-u. However, it is surprising to observe that UNC-IE performs statistically significantly worse than UNC-u for almost all datasets and measures. Selecting uncertain instances that have insufficient evidence often performs worse than selecting the least uncertain instance among the top $t$ uncertain instances. Note that UNC-1, UNC-CE, UNC-IE, and UNC-u do not have much flexibility in choosing between uncertain instances; that is they all work within the top $t$ uncertain instances, and yet UNC-IE performs much worse than both UNC-1 and UNC-u, whereas UNC-CE performs much better than both UNC-1 and UNC-u.

UNC-CE clearly stands out as a winner strategy, whereas UNC-IE is clearly the worst

Table 3.3. UNC-CE and UNC-IE versus UNC-u with $t = 5$, $t = 10$, and $t = 20$. Number of datasets on which UNC-CE and UNC-IE significantly Win (W), Tie (T), or Lose (L) compared to UNC-u baseline.

| **UNC-u Baseline** | **AUC** | **ACCU** | **F1** |
|---|---|---|---|
| **Method** | **W/T/L** | **W/T/L** | **W/T/L** |
| UNC-CE with $t = 5$ | **6**/0/2 | **4**/0/0 | **3**/0/1 |
| UNC-CE with $t = 10$ | **7**/0/1 | **4**/0/0 | **3**/0/1 |
| UNC-CE with $t = 20$ | **6**/1/1 | **4**/0/0 | **3**/0/1 |
| UNC-IE with $t = 5$ | 0/0/**8** | 0/0/**4** | 0/0/**4** |
| UNC-IE with $t = 10$ | 1/0/**7** | 0/0/**4** | 0/0/**4** |
| UNC-IE with $t = 20$ | 1/0/**7** | 0/0/**4** | 0/0/**4** |

performing uncertainty strategy. UNC-CE improves over UNC-1 on almost all datasets and measures, whereas UNC-IE loses to UNC-1 on almost all datasets and measures. This result is surprising because one would not expect such a huge difference between UNC-CE and UNC-IE strategies. After all, UNC-CE strategy picks an uncertain instance that has large evidence for both classes and hence, intuitively, labeling such instances is focused on correcting the mistakes of the learner. On the other hand, UNC-IE strategy picks an uncertain instance that has little evidence for both classes and hence, intuitively, labeling such instances is focused on teaching new things to the learner. Both types of uncertainties are expected to be important for improving the model. We provide analytical and empirical justifications as to why UNC-CE outperforms UNC-IE in Section 3.5.

Next, we present a comparison of the ranks of the uncertain instances selected by UNC-CE and UNC-IE. Note that UNC-1 will always pick the top most uncertain instance, and hence would select rank 1 uncertain instance. UNC-u on the other hand would always select rank $t$ uncertain instance. UNC-CE and UNC-IE work within the top $t$ uncertain instances and select rank $u$ uncertain instance, where $u$ is between 1 and $t$. Table 3.4 presents the mean rank of uncertain instances selected by UNC-CE and UNC-IE with $t = 10$ for all datasets. Figure 3.4 presents histograms for all eight datasets, showing the ranks

of uncertain instances selected by UNC-CE and UNC-IE with $t = 10$. The histograms with $t = 5$ and $t = 20$ have similar trends and are omitted to avoid redundancy. These histograms show that UNC-CE and UNC-IE choose a variety of ranks of uncertain instances for most datasets and hence the differences between UNC-1, UNC-u, UNC-CE, and UNC-IE do not stem from the rank of uncertain instances but rather, they are due to the information content of the different instances chosen by each method.

Table 3.4. The mean rank of uncertain instances selected by UNC-CE and UNC-IE for the eight datasets over various iterations of learning and 25 trials.

| **Dataset** | **UNC-CE** | | **UNC-IE** | |
|---|---|---|---|---|
| | **Mean** | **Std. Dev** | **Mean** | **Std. Dev** |
| **Spambase** | 5.50 | 2.99 | 6.14 | 2.97 |
| **Ibn Sina** | 5.09 | 3.15 | 7.09 | 2.53 |
| **Calif. Housing** | 5.57 | 2.87 | 5.58 | 2.89 |
| **Nova** | 5.42 | 2.86 | 6.02 | 2.76 |
| **Sick** | 7.16 | 2.73 | 6.03 | 2.85 |
| **Zebra** | 5.92 | 2.88 | 6.03 | 2.85 |
| **LetterO** | 7.18 | 2.73 | 5.39 | 2.88 |
| **Hiva** | 5.27 | 2.95 | 6.83 | 2.73 |

**3.3.3 Scalability.** We discuss the comparison of running times of UNC-1, UNC-CE, and UNC-IE methods for naïve Bayes for one iteration of active learning. Given dataset $\mathcal{D} = \left\{x^{(i)}, y^{(i)}\right\}_1^m$ where, $x^{(i)} \in \mathbb{R}^f$, and $y^{(i)} \in \{+1, -1\}$ is discrete valued. UNC-1 calculates uncertainty score (measured through Equations 2.1 or 2.2). The time complexity of calculating the conditional probabilities $P_\theta(Y|X)$ in each of these equations is proportional to the number of attributes, which is $O(f)$. Since we compute uncertainty on $m$ subsampled instances, the time complexity of UNC-1 is $O(m \times f)$.

UNC-CE and UNC-IE methods also calculate uncertainty on $m$ instances, which takes time $O(m \times f)$. Additionally, UNC-CE and UNC-IE methods calculate evidence for each attribute of an instance, which again takes time $O(f)$. This additional step is done

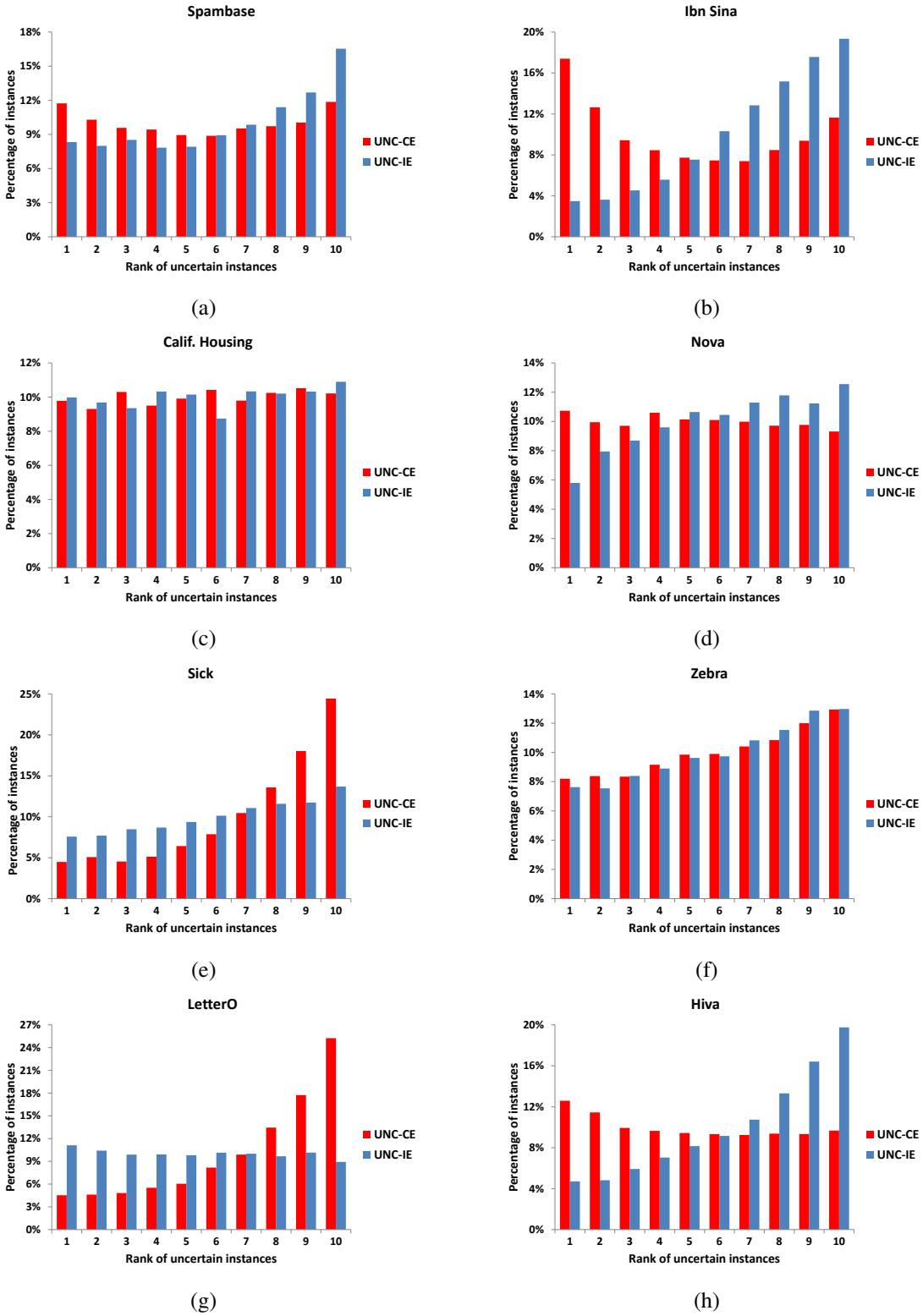



Figure 3.4. Histograms showing ranks of uncertain instances selected by UNC-CE and UNC-IE for all eight datasets.

only for the top $t$ uncertain instances. Hence, the running time of UNC-CE and UNC-IE methods is $O((t + m) \times f)$. Given that $t$ is a small constant ($t << m$), the running times of UNC-CE and UNC-IE are comparable to the running time of UNC-1. Table 3.5 presents the running times of UNC-1, UNC-CE, and UNC-IE for one iteration of active learning with various $t$ values for three datasets, Nova, Zebra and Hiva. We omit the running times for other five datasets, as the running time per iteration for them is less than 1 second. As presented in Table 3.1, these three datasets have the highest number of features and thus it is not surprising that the running times are largest for these three datasets. These experiments were run on a Windows 7 machine with Intel Xeon processor (2.4 GHz). The results show that the running times of UNC-CE and UNC-IE are comparable to UNC-1. Moreover, the running times of UNC-CE and UNC-IE do not vary much with different $t$ values. Interestingly, we observe that sometimes UNC-CE and UNC-IE seem to take less time than UNC-1, but these differences are not statistically significant and hence we attribute these differences to variances in the run times due to other uncontrollable factors such as other processes that might be run by the OS. The overall conclusion is that the run time is dominated by the number of features and the additional time cost that UNC-CE and UNC-IE require on top of UNC-1 is negligible.

Table 3.5. Running times (in seconds) for three datasets for one iteration of active learning, with various $t$ values. We present mean $\pm$ Std. Dev of the running times over 25 trials.

| | | **Dataset** | | |
|---|---|---|---|---|
| | | Nova | Zebra | Hiva |
| **UNC-1** | $t = 1$ | $15.02 \pm 2.25$ | $1.34 \pm 0.05$ | $1.95 \pm 0.05$ |
| **UNC-CE** | $t = 1$ | $14.58 \pm 1.44$ | $1.33 \pm 0.05$ | $1.95 \pm 0.04$ |
| | $t = 5$ | $14.81 \pm 1.25$ | $1.32 \pm 0.05$ | $1.95 \pm 0.05$ |
| | $t = 20$ | $14.95 \pm 1.24$ | $1.32 \pm 0.05$ | $1.94 \pm 0.05$ |
| **UNC-IE** | $t = 1$ | $15.48 \pm 0.98$ | $1.36 \pm 0.05$ | $1.94 \pm 0.05$ |
| | $t = 5$ | $15.90 \pm 0.93$ | $1.37 \pm 0.06$ | $1.95 \pm 0.06$ |
| | $t = 20$ | $15.72 \pm 1.40$ | $1.38 \pm 0.06$ | $1.98 \pm 0.06$ |

## 3.4 User Study

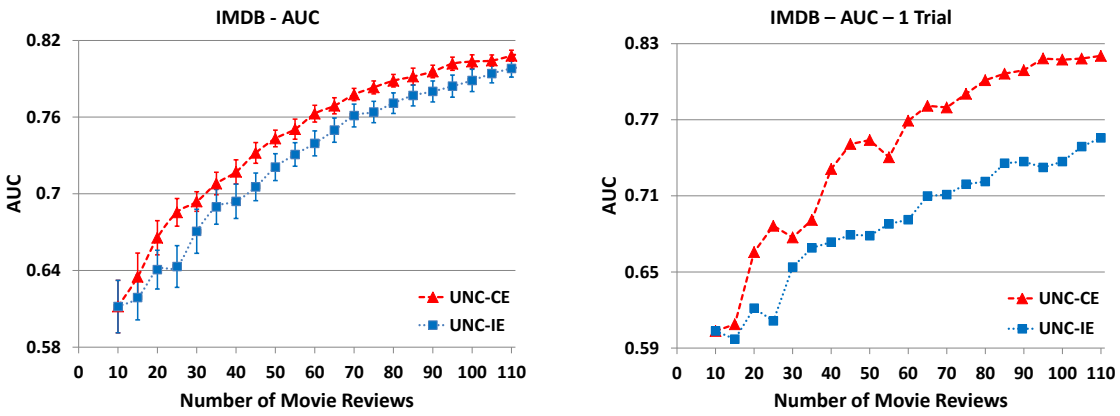
We designed and ran a user study to investigate whether it is easier or harder for humans to label conflicting-evidence cases versus insufficient-evidence cases. Specifically, we were interested in two measures: i) how long does it take humans to label and ii) how accurate are the humans on their labels for conflicting-evidence cases versus insufficient-evidence cases.

It could very well be that conflicting cases can be harder for humans because they contain conflicting information suggesting both classes, which might confuse humans about the class label. It is also possible for insufficient-evidence cases to be difficult for humans because they do not have enough information, e.g. neutral cases. We note that we define conflicting-evidence and insufficient-evidence uncertainties with respect to the underlying model and not with respect to the expert. Thus, it is possible that the model has conflicting evidence or insufficient evidence but it still might be an easy case for the expert. In this section, we investigate these questions through a user study.

We experimented with IMDB dataset consisting of 50K movie reviews [61], as labeling movie reviews does not require much domain expertise and hence it is easier to recruit users for our user study. Moreover, this dataset contains full text of the reviews whereas the other datasets we have used in Section 3.3 simply consist of feature-value pairs. We trained a multinomial naïve Bayes model, as multinomial naïve Bayes is known to outperform Bernoulli naïve Bayes for text classification [64]. The evidences for multinomial naïve Bayes are calculated similar to that of Bernoulli naïve Bayes, which we describe in Section 3.6.

We bootstrapped the multinomial naïve Bayes model with 10 reviews, selecting 5 random reviews from each class and used tf-idf representation of the data. Figure 3.5(a) presents the average AUC results of UNC-CE and UNC-IE strategies over 10 trials simulated

using ground truth. Out of the 10 trials, UNC-CE wins over UNC-IE on 6 of the trials. For the user study, we selected one of the 10 trials, shown in Figure 3.5(b), for which UNC-CE and UNC-IE had the biggest difference in performance because we wanted to test the case where UNC-CE and UNC-IE had the most difference in impact on the learning. The accuracy of UNC-CE after labeling 110 (10 bootstrap + 100 budget) reviews was 73.5% and the accuracy of UNC-IE was 67.24%.




(a) (b)

Figure 3.5. (a) Average AUC of UNC-CE and UNC-IE over 10 trials on IMDB dataset. (b) Performance of UNC-CE and UNC-IE on the trial used in the user study.

We shuffled these 200 movie reviews selected by UNC-CE and UNC-IE to make sure that the users had no way of determining which was a conflicting versus insufficient evidence case with respect to the underlying model. In fact, users were not told that they were part of a study to distinguish between conflicting versus insufficient evidence cases. They were simply asked to label 200 movie reviews as positive or negative. We had five users for our study and each user was shown movie reviews in the same order. For each movie review, we recorded the response time and annotation (positive/negative). We treated the actual labels as gold standard labels and measured accuracy of the users by comparing their annotations with the gold standard labels.

We first compare whether UNC-CE and UNC-IE differ on the length of the docu-

ments chosen. We observe that the average length of reviews selected by UNC-CE was 213.32 and the average length of reviews selected by UNC-IE was 205.04. The two-tailed unpaired t-tests between the lengths of UNC-CE and UNC-IE reviews show that the difference in lengths of UNC-CE and UNC-IE reviews is not significantly different.

Next, we compare the average time taken by users, in seconds, to label UNC-CE and UNC-IE reviews in Table 3.6. We also include the Average User as the mean of all the users in the last row. We observe that even though users took slightly more time (a few more seconds) on UNC-CE instances than UNC-IE instances, the differences are not statistically significant as measured by two-tailed unpaired t-tests and the p-values are reported in the last column of Table 3.6.

Table 3.6. Annotation time of all users on UNC-CE and UNC-IE movie reviews. The t-test results show that annotation times of UNC-CE and UNC-IE reviews are not significantly different. We report the $p-$values obtained using two-tailed unpaired t-tests.

| **Users** | **Annotation time of** | | $p-$**value** |
|---|---|---|---|
| | **UNC-CE reviews** | **UNC-IE reviews** | |
| **User 1** | $14.27 \pm 9.54$ | $13.26 \pm 11.03$ | 0.49 |
| **User 2** | $55.40 \pm 35.01$ | $52.49 \pm 36.64$ | 0.57 |
| **User 3** | $81.03 \pm 71.41$ | $74.62 \pm 61.55$ | 0.50 |
| **User 4** | $21.86 \pm 14.91$ | $20.18 \pm 15.82$ | 0.45 |
| **User 5** | $25.79 \pm 33.54$ | $26.29 \pm 29.68$ | 0.91 |
| **Average User** | $39.57 \pm 25.84$ | $37.39 \pm 24.25$ | 0.54 |

Table 3.7 presents accuracy of the users on the 100 movie reviews selected by UNC-CE and UNC-IE. The accuracy of Average User is the average accuracy of all users. We also present majority vote accuracy which is calculated by taking a majority voting of all users on each movie review. The accuracy of all users, except User 3, was similar for both UNC-CE and UNC-IE reviews.

We plot the same Figure 3.5(b) again, this time the x-axis is not the number of instances but rather the average time it took the 5 users (i.e., the Average User's time).

Table 3.7. Accuracy of all users on UNC-CE and UNC-IE movie reviews.

| Users | Accuracy on | |
|---|---|---|
| | **UNC-CE reviews** | **UNC-IE reviews** |
| **User 1** | 95% | 93% |
| **User 2** | 93% | 94% |
| **User 3** | 90% | 96% |
| **User 4** | 95% | 95% |
| **User 5** | 95% | 97% |
| **Average User** | 93.6% | 95% |
| **Majority vote accuracy** | 96% | 96% |

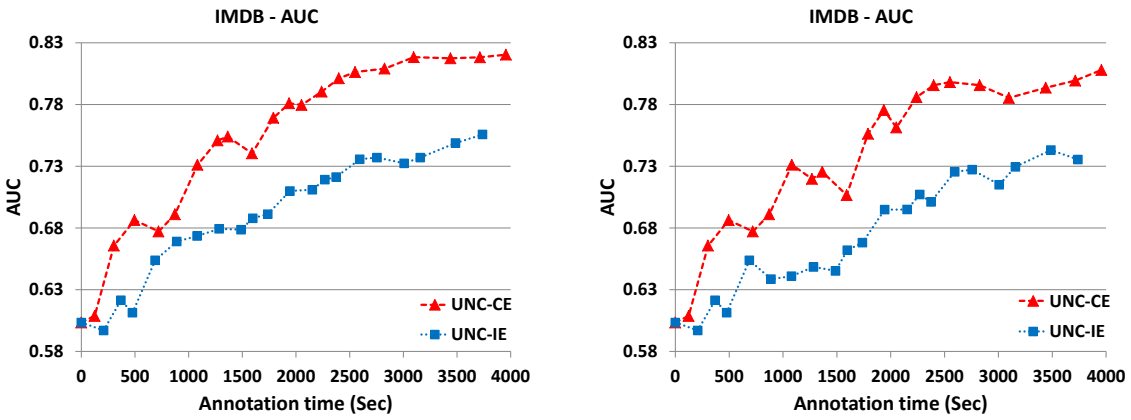
Figure 3.6(a) shows the results using the ground-truth labels and Figure 3.6(b) shows the results using majority vote labels.[3] This result shows that even though labeling UNC-CE reviews takes slightly more time than UNC-IE reviews, it is still worth labeling reviews using UNC-CE strategy.

Overall, our user study focused on sentiment classification. Though we cannot claim that our results carry over to other document classification tasks or other domains, we conclude that we did not observe significant differences between UNC-CE and UNC-IE instances in terms of the annotation time and labeling difficulty for the sentiment classification task.

## 3.5 Analytical and Empirical Justifications

Extensive experiments with real-world datasets presented in Section 3.3.2 clearly show that UNC-CE provides significant improvements over UNC-IE. This is a bit startling because one would expect that the model would benefit from both the UNC-CE cases and

[3]This figure does not correspond to a real-time simulation of active learning with users. When the user-provided labels are used, the underlying active learning strategy, whether it be UNC-CE or UNC-IE, would potentially take a different path per user based on their labels. Then, each user would potentially differ on the documents they label, and therefore meaningful comparisons of time and accuracy across users would not be possible.




(a) (b)

Figure 3.6. (a) Performance comparison of UNC-CE and UNC-IE strategies based on the annotation time of Average User using ground-truth labels. (b) Performance comparison of UNC-CE and UNC-IE strategies based on the annotation time of Average User and using majority vote labels.

UNC-IE cases. When the conflicting, UNC-CE, cases are annotated, the model would have a chance to correct its perceived conflict, and when the inconclusive, UNC-IE, cases are annotated, the model would learn about new feature-value class correlations that it did not know before. In this section, we provide both analytical and empirical results that shed light on why UNC-CE often outperforms UNC-IE. Specifically,

- We show both analytically and empirically that UNC-CE cases have lower density, with respect to the model trained on the labeled data, than the UNC-IE cases. Density of an instance, $x^{(i)}$, is defined as the probability distribution, $P(x^{(i)})$, with respect to the model trained on the current training data.

- We show empirically that the model has higher variance on the UNC-CE cases than on the UNC-IE cases.

These two results suggest that the conflict perceived by the model is supported by less amount of training data than the insufficiency of the evidences. Put another way, there is less labeled data that supports the conflict and there is more labeled data that supports the

inconclusiveness. This is further supported by the finding that UNC-CE cases have higher variance than UNC-IE cases. That is, the parameter values that support conflict have higher variance because they rely on smaller amount of labeled data. Therefore, the model is more likely to be incorrect in its decision that the evidence is conflicting than its decision that the evidence is inconclusive.

This is not to say that the UNC-IE cases are totally useless. Even though UNC-IE cases are supported by more labeled data than the UNC-CE cases, the total amount of labeled data is still fairly small in active learning settings. Therefore, the model is likely to be incorrect in its decision that the case is inconclusive. However, the UNC-CE cases have even less support than the UNC-IE cases and thus the model is often better off labeling more of the UNC-CE cases.

**3.5.1 Analytical Justification.** For simplicity, we first prove the density argument for binary variables using a two-attributes case where out of four possible cases, one is UNC-CE and the other is UNC-IE. We then provide explanation of density argument for continuous attributes.

**3.5.1.1 Binary Attributes.** Assume we have a single attribute, $X_1$, that is binary with $\langle T, F \rangle$. Similarly, the class variable $Y$ is binary with $\langle -1, +1 \rangle$. In this section, we prove that i) $X_1 = T$ and $X_1 = F$ cannot provide evidence for the same class at the same time, ii) if $X_1 = T$ provides evidence for one class, then $X_1 = F$ has to provide evidence for the opposing class, and finally iii) the amount of evidence that $X_1 = T$ provides for one class can be larger/smaller than the evidence $X_1 = F$ provides for the opposing class. These three properties will be needed to prove the density argument for the two-attributes case. Let

$$P(X_1 = T|Y = +1) = p; P(X_1 = F|Y = +1) = 1 - p$$

$$P(X_1 = T|Y = -1) = q; P(X_1 = F|Y = -1) = 1 - q$$

The following propositions hold when both $X_1$ and $Y$ are binary.

**Proposition 1**: If $X_1 = T$ provides evidence for $Y = +1$, then $X_1 = F$ cannot provide evidence for $Y = +1$ at the same time.

*Proof.* Without loss of generality, assume $p > q$. Then, $X_1 = T$ provides evidence for $Y = +1$ and the magnitude of the evidence is $\frac{p}{q}$. Can $X_1 = F$ provide evidence for $Y = +1$ at the same time? That is, when $p > q$, can $\frac{1-p}{1-q}$ be greater than 1? The answer is obviously no and hence two different values of $X_1$ cannot provide evidence for the same class at the same time.

**Proposition 2:** If $X_1 = T$ provides evidence for one class then $X_1 = F$ *has* to provide evidence for the other class.

*Proof.* When $X_1 = T$ provides evidence for one class, is it possible that $X_1 = F$ provides evidence for no class? That is, is it possible to have $\frac{p}{q} \neq 1$ and $\frac{1-p}{1-q} = 1$? This is obviously impossible, and hence if $X_1 = T$ provides evidence for one class then $X_1 = F$ *has* to provide evidence for some class. Given proposition 1, we know that $X_1 = F$ cannot provide evidence for the class that $X_1 = T$ supports. Therefore, if $X_1 = T$ supports one class, then $X_1 = F$ has to support the other class.

**Proposition 3:** One value of an attribute can provide a greater evidence for one class than the evidence the other value of the same attribute provides for the other class.

*Proof.* Without loss of generality, assume $\frac{p}{q} > 1$. Then, $X_1 = T$ provides evidence for $Y = +1$. Hence $\frac{1-q}{1-p} > 1$ and $X_1 = F$ provides evidence for $Y = -1$. The evidence that $X_1 = T$ provides for $Y = +1$ is greater than the evidence $X_1 = F$ provides for $Y = -1$, that is, $\frac{p}{q} > \frac{1-q}{1-p}$, if and only if $p = q + \epsilon \leq 0.5$ for $\epsilon > 0$ or $p = 0.5 + \alpha$ and $q = 0.5 - \beta$ for $0 < \alpha < \beta < 0.5$.

For the two-attributes case, assume we have two binary attributes, $X_1$ and $X_2$. In

this case, there are four possible instances (e.g., $\langle X_1 = T, X_2 = T\rangle$, $\langle X_1 = T, X_2 = F\rangle$, etc.). To compare UNC-CE and UNC-IE methods, we need the model to be uncertain on at least two of these instances and we want one of them to be a conflicting-evidence case and the other one to be an insufficient-evidence case. Assume the following distributions for a naïve Bayes classifier:

$$P(X_1 = T|Y = +1) = p; P(X_1 = F|Y = +1) = 1 - p$$
$$P(X_1 = T|Y = -1) = q; P(X_1 = F|Y = -1) = 1 - q$$
$$P(X_2 = T|Y = +1) = r; P(X_2 = F|Y = +1) = 1 - r$$
$$P(X_2 = T|Y = -1) = s; P(X_2 = F|Y = -1) = 1 - s$$

Assume that the uncertain instances are $\langle X_1 = T, X_2 = T\rangle$ and $\langle X_1 = F, X_2 = F\rangle$. That is:

$$\frac{P(Y = +1)P(X_1 = T|Y = +1)P(X_2 = T|Y = +1)}{P(Y = -1)P(X_1 = T|Y = -1)P(X_2 = T|Y = -1)} \approx 1$$
$$\frac{P(Y = +1)P(X_1 = F|Y = +1)P(X_2 = F|Y = +1)}{P(Y = -1)P(X_1 = F|Y = -1)P(X_2 = F|Y = -1)} \approx 1$$

Without loss of generality, assume $X_1 = T$ provides evidence for $Y = +1$. Then, Propositions 1 and 2 above show that $X_1 = F$ provides evidence for $Y = -1$. Assuming $P(Y)$ is uniform with $0.5$, for the instance $\langle X_1 = T, X_2 = T\rangle$ to be uncertain, $X_2 = T$ must provide evidence for $Y = -1$ and this evidence must be roughly equal to the evidence that $X_1 = T$ provides for $Y = +1$. Invoking propositions 1 and 2 again, $X_2 = F$ then must provide evidence for $Y = +1$ and for $\langle X_1 = F, X_2 = F\rangle$ to be uncertain, the evidence $X_1 = F$ provides for $Y = -1$ must be roughly equal to the evidence $X_2 = F$ provides for $Y = +1$.

Without loss of generality, assume $\langle X_1 = T, X_2 = T\rangle$ is the UNC-CE instance and

$\langle X_1 = F, X_2 = F\rangle$ is the UNC-IE instance. Then, for both instances to be uncertain, and for $\langle X_1 = T, X_2 = T\rangle$ to be the conflicting case as opposed to $\langle X_1 = F, X_2 = F\rangle$, we need

$$\frac{p}{q} \approx \frac{s}{r} > \frac{1-q}{1-p} \approx \frac{1-r}{1-s}$$

**Proposition 4:** The density of UNC-CE instance *with respect to* the naïve Bayes model is less than the density of UNC-IE instance, i.e. $P(X_1 = T, X_2 = T) < P(X_1 = F, X_2 = F)$.

*Proof.* Assume that $P(Y)$ is uniform, $P(Y = +1) = P(Y = -1) = 0.5$. We need to prove that

$$0.5 \times p \times r + 0.5 \times q \times s < 0.5 \times (1-p) \times (1-r) + 0.5 \times (1-q) \times (1-s)$$

$$\begin{aligned}
0.5 \times p \times r + 0.5 \times q \times s &\overset{?}{<} 0.5 \times (1-p) \times (1-r) + 0.5 \times (1-q) \times (1-s) \\
p \times r + q \times s &\overset{?}{<} (1-p) \times (1-r) + (1-q) \times (1-s) \\
p \times r + q \times s &\overset{?}{<} 1 - r - p + p \times r + 1 - s - q + q \times s \\
0 &\overset{?}{<} 2 - r - p - s - q \\
r + p + s + q &\overset{?}{<} 2
\end{aligned}$$

Because $\frac{p}{q} > \frac{1-q}{1-p}$ and as we have shown in Proposition 3, either $p = q + \epsilon \leq 0.5$ for $\epsilon > 0$ or $p = 0.5 + \alpha$ and $q = 0.5 - \beta$ for $0 < \alpha < \beta < 0.5$. Similar arguments apply to $s$ and $r$: either $s = r + \epsilon \leq 0.5$ for $\epsilon > 0$ or $s = 0.5 + \alpha$ and $r = 0.5 - \beta$ for $0 < \alpha < \beta < 0.5$.

Case 1: $p = q + \epsilon \leq 0.5$ for $\epsilon > 0$. Then $p + q < 1$. Similarly, if $s = r + \epsilon \leq 0.5$ for $\epsilon > 0$, then $s + r < 1$.

Case 2: $p = 0.5 + \alpha$ and $q = 0.5 - \beta$ for $0 < \alpha < \beta < 0.5$. Then $p + q = 0.5 + \alpha + 0.5 - \beta = 1 + \alpha - \beta < 1$. Similarly, $s + r = 0.5 + \alpha + 0.5 - \beta = 1 + \alpha - \beta < 1$.

Since in both cases, $p + q < 1$ and $s + r < 1$, we conclude that $p + q + r + s < 2$, proving that the density with respect to the underlying naïve Bayes model is lower for the UNC-CE case than the UNC-IE case. Our proof assumed that $P(Y)$ was uniform; the proposition holds when $P(Y)$ is not uniform and the proof is similar. Moreover, for simplicity, our proof focused on the two-attributes case. The same arguments can be extended to multiple-attributes case by induction.

**3.5.1.2 Continuous Attributes.** In this section we investigate the density hypothesis for continuous attributes. For continuous attributes, Gaussian naïve Bayes assumes that within each class, the continuous attributes are normally distributed:

$$p(x|Y) = \mathcal{N}(x; \mu, \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

For simplicity of exposition, consider a training data with two continuous attributes, $X_1$ and $X_2$, and a binary class variable, $Y$ with $\langle -1, +1 \rangle$. Let the mean of attribute $X_1$ for class $+1$ be $\mu_{1,+1}$ and mean of attribute $X_1$ for class $-1$ be $\mu_{1,-1}$. Similarly, let mean of attribute $X_2$ for class $+1$ be $\mu_{2,+1}$ and mean of attribute $X_2$ for class $-1$ be $\mu_{2,-1}$. Let the standard deviation of attribute $X_1$ for class $+1$ be $\sigma_{1,+1}$ and standard deviation of attribute $X_1$ for class $-1$ be $\sigma_{1,-1}$. Similarly, let standard deviation of attribute $X_2$ for class $+1$ be $\sigma_{2,+1}$ and standard deviation of attribute $X_2$ for class $-1$ be $\sigma_{2,-1}$. For each class and attribute,

Gaussian naïve Bayes estimates the conditional probability of attribute given class as:

$$p(X_1|Y=+1) = \mathcal{N}(\mu_{1,+1}, \sigma_{1,+1})$$
$$p(X_1|Y=-1) = \mathcal{N}(\mu_{1,-1}, \sigma_{1,-1})$$
$$p(X_2|Y=+1) = \mathcal{N}(\mu_{2,+1}, \sigma_{2,+1})$$
$$p(X_2|Y=-1) = \mathcal{N}(\mu_{2,-1}, \sigma_{2,-1})$$

Assume $\mu_{1,+1}$=$\mu_{2,+1}$=$a$ and $\mu_{1,-1}$=$\mu_{2,-1}$=$b$, where $b > a$. This can be easily achieved by rotating and shifting the axes. For simplicity, assume that both attributes have equal variance in both classes, i.e. $\sigma_{1,+1}$=$\sigma_{1,-1}$=$\sigma_{2,+1}$=$\sigma_{2,-1}$=$\sigma$ (the case where each class and feature value pair has unequal variances is similar). Hence the data for class $+1$ is centered around the point $\langle a, a\rangle$ and the data for class $-1$ is centered around the point $\langle b, b\rangle$. Figure 3.7 illustrates these points for the two classes. The decision boundary represents the line where an instance has equal probability, $0.5$, of belonging to each class. We consider two instances on the decision boundary, where one instance is $\langle \frac{a+b}{2}, \frac{a+b}{2}\rangle$ and the other is $\langle c, d\rangle$, assuming $c < \frac{a+b}{2}$ and $d > \frac{a+b}{2}$.

Next we provide analytical justification showing that conflicting cases have higher evidence but lower density in the training data, whereas insufficient-evidence cases have lower evidence and higher density in the training data.

**Proposition 5**: Instance $\langle c, d\rangle$ has higher total evidence than instance $\langle \frac{a+b}{2}, \frac{a+b}{2}\rangle$.

*Proof.* First, we show how the evidences for +1 (or -1) class can be computed. The

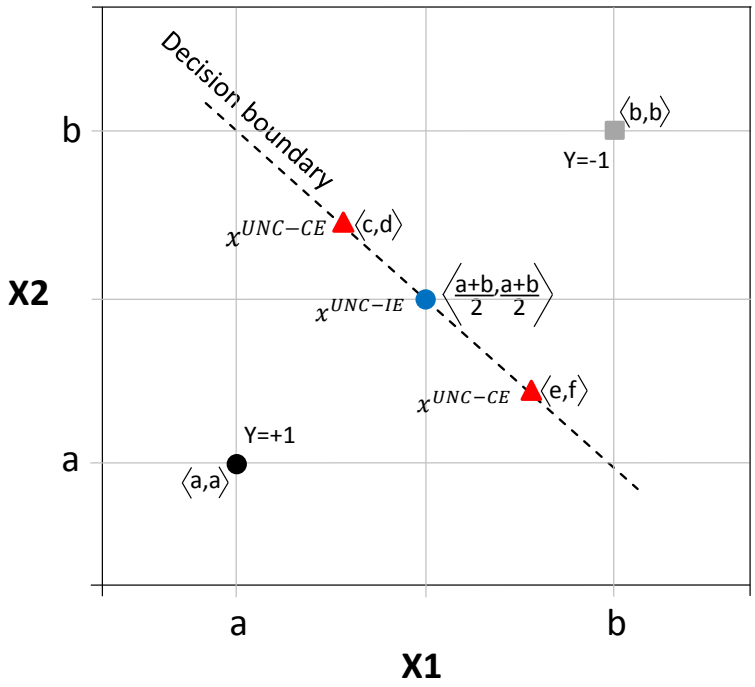



Figure 3.7. Analysis of Gaussian naïve Bayes using two continuous attributes, $X_1$ and $X_2$. The mean of both attributes for class $+1$ is $a$, and the mean of both attributes for class $-1$ is $b$. We consider two instances, $\langle \frac{a+b}{2}, \frac{a+b}{2} \rangle$ and $\langle c, d \rangle$ on the decision boundary and we prove that $\langle \frac{a+b}{2}, \frac{a+b}{2} \rangle$ is insufficient-evidence uncertain instance and we refer to it as $x^{UNC-IE}$ on this graph, and $\langle c, d \rangle$ is conflicting-evidence uncertain instance and we refer to it as $x^{UNC-CE}$ on this graph.

evidence provided by attribute $X_f$ for class +1 using Gaussian naïve Bayes is computed as:

$$\frac{\frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(X_f-\mu_{f,+1})^2}{2\sigma^2}}}{\frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(X_f-\mu_{f,-1})^2}{2\sigma^2}}} = e^{\frac{-(X_f-\mu_{f,+1})^2+(X_f-\mu_{f,-1})^2}{2\sigma^2}}$$

$$= e^{\frac{(X_f-\mu_{f,-1}+X_f-\mu_{f,+1})(X_f-\mu_{f,-1}-X_f+\mu_{f,+1})}{2\sigma^2}}$$

$$= e^{\frac{(2X_f-\mu_{f,-1}-\mu_{f,+1})(\mu_{f,+1}-\mu_{f,-1})}{2\sigma^2}}$$

For class -1, this ratio is reversed, hence the evidence provided by attribute $X_f$ for the class

-1 is:

$$e^{\frac{(2X_f-\mu_{f,-1}-\mu_{f,+1})(\mu_{f,-1}-\mu_{f,+1})}{2\sigma^2}}$$

First, we compute the evidences for instance $\langle\frac{a+b}{2}, \frac{a+b}{2}\rangle$. The evidence that attribute $X_1$ of instance $\langle\frac{a+b}{2}, \frac{a+b}{2}\rangle$ provides for class +1 is:

$$e^{\frac{(2(\frac{a+b}{2})-a-b)(a-b)}{2\sigma^2}} = e^0 = 1$$

That is, $X_1 = \frac{a+b}{2}$ does not provide evidence for either class, because $\frac{P(X_1=\frac{a+b}{2}|+1)}{P(X_1=\frac{a+b}{2}|-1)} = 1$. The same argument applies to $X_2 = \frac{a+b}{2}$. The overall evidence provided by attributes of instance $\langle\frac{a+b}{2}, \frac{a+b}{2}\rangle$ using Equation 3.9 is $1 \times 1 = 1$.

Next we compute the evidences for instance $\langle c, d\rangle$. Note that $c$ is closer to class +1 and $d$ is closer to class -1. The evidence that $X_1 = c$ provides for class +1 is:

$$e^{\frac{(2c-a-b)(a-b)}{2\sigma^2}}$$

Since $c < \frac{a+b}{2}$ and $a < b$, this evidence is greater than 1. The evidence that $X_2 = d$ provides for class -1 is:

$$e^{\frac{(2d-a-b)(b-a)}{2\sigma^2}}$$

Since $d > \frac{a+b}{2}$ and $b > a$, this evidence is greater than 1. The total evidence provided by attributes of instance $\langle c, d\rangle$ using Equation 3.9 is:

$$e^{\frac{(2c-a-b)(a-b)}{2\sigma^2}} \times e^{\frac{(2d-a-b)(b-a)}{2\sigma^2}}$$

which is greater than 1, whereas the total evidence provided by attributes of $\langle \frac{a+b}{2}, \frac{a+b}{2} \rangle$ is equal to 1.

Similar reasoning applies to the instance $\langle e, f \rangle$ in Figure 3.7. We conclude that as we move on the decision boundary away from its center, i.e. move away from $\langle \frac{a+b}{2}, \frac{a+b}{2} \rangle$ in the direction of $\langle c, d \rangle$ (or $\langle e, f \rangle$), the evidences for each class get higher and hence the conflict grows.

Before we prove the density argument that conflicting-evidence cases have lower density compared to the insufficient-evidence cases, we first establish a relationship among $c$, $d$, $a$, and $b$. Note that for instance $\langle c, d \rangle$ to be uncertain, the evidence for class $+1$ must be equal to the evidence for class $-1$. Hence,

$$e^{\frac{(2c-a-b)(a-b)}{2\sigma^2}} = e^{\frac{(2d-a-b)(b-a)}{2\sigma^2}}$$

$$\therefore 2c - a - b = a + b - 2d$$

$$c + d = a + b$$

**Proposition 6**: The density of instance $\langle c, d \rangle$ with respect to the underlying model is lower than the density of instance $\langle \frac{a+b}{2}, \frac{a+b}{2} \rangle$.

*Proof.* Density of instance $\langle X_1, X_2 \rangle$ with respect to the underlying model can be computed as follows:

$$\begin{aligned} P(X_1, X_2) =& P(X_1, X_2, +1) + P(X_1, X_2, -1) \\ =& P(+1)P(X_1, X_2| + 1) + P(-1)P(X_1, X_2| - 1) \end{aligned}$$

Naïve Bayes assumes that attributes are conditionally independent given class, hence,

$$\begin{aligned}P(X_1, X_2) =& P(+1)P(X_1|+1)P(X_2|+1) + P(-1)P(X_1|-1)(X_2|-1)\\ =& P(+1) \times \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(X_1-\mu_{1,+1})^2}{2\sigma^2}} \times \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(X_2-\mu_{2,+1})^2}{2\sigma^2}} +\\ & P(-1) \times \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(X_1-\mu_{1,-1})^2}{2\sigma^2}} \times \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(X_2-\mu_{2,-1})^2}{2\sigma^2}}\end{aligned}$$

Assuming $P(+1) = P(-1) = 0.5$,

$$P(X_1, X_2) = \frac{1}{2} \times \frac{1}{\sqrt{2\pi\sigma^2}} \left( e^{-\frac{(X_1-\mu_{1,+1})^2+(X_2-\mu_{1,+1})^2}{2\sigma^2}} + e^{-\frac{(X_1-\mu_{1,-1})^2+(X_2-\mu_{1,-1})^2}{2\sigma^2}} \right)$$

Density of instance $\langle \frac{a+b}{2}, \frac{a+b}{2} \rangle$ is:

$$\begin{aligned}& \frac{1}{2} \times \frac{1}{\sqrt{2\pi\sigma^2}} \left( e^{-\frac{(\frac{a+b}{2}-a)^2+(\frac{a+b}{2}-a)^2}{2\sigma^2}} + e^{-\frac{(\frac{a+b}{2}-b)^2+(\frac{a+b}{2}-b)^2}{2\sigma^2}} \right)\\ =& \frac{1}{2} \times \frac{1}{\sqrt{2\pi\sigma^2}} \left( e^{-\frac{(\frac{b-a}{2})^2}{2\sigma^2}} + e^{-\frac{(\frac{a-b}{2})^2}{2\sigma^2}} \right)\\ =& \frac{1}{2} \times \frac{1}{\sqrt{2\pi\sigma^2}} \times 2e^{-\frac{(\frac{b-a}{2})^2}{2\sigma^2}}\end{aligned}$$

Density of instance $\langle c, d \rangle$ is:

$$\frac{1}{2} \times \frac{1}{\sqrt{2\pi\sigma^2}} \left( e^{-\frac{(c-a)^2+(d-a)^2}{2\sigma^2}} + e^{-\frac{(c-b)^2+(d-b)^2}{2\sigma^2}} \right)$$

First, note that $(c-a)^2 + (d-a)^2 = (c-b)^2 + (d-b)^2$.

$$(c-a)^2 - (c-b)^2 \stackrel{?}{=} (d-b)^2 - (d-a)^2$$
$$(c-a+c-b)(c-a-c+b) \stackrel{?}{=} (d-b+d-a)(d-b-d+a)$$
$$(2c-a-b)(-a+b) \stackrel{?}{=} (2d-b-a)(-b+a)$$
$$2c-a-b \stackrel{?}{=} b+a-2d$$
$$c+d \stackrel{?}{=} a+b$$

We earlier established relationship among $c$, $d$, $a$, and $b$ and proved that $c+d = a+b$. Therefore, the density of instance $\langle c, d\rangle$ is:

$$\frac{1}{2} \times \frac{1}{\sqrt{2\pi\sigma^2}} 2e^{-\frac{(c-a)^2+(d-a)^2}{2\sigma^2}}$$

Next, we test whether density of instance $\langle \frac{a+b}{2}, \frac{a+b}{2}\rangle$ is higher than density of instance $\langle c, d\rangle$.

$$\frac{1}{2} \times \frac{1}{\sqrt{2\pi\sigma^2}} \times 2e^{-\frac{(\frac{b-a}{2})^2}{2\sigma^2}} \stackrel{?}{>} \frac{1}{2} \times \frac{1}{\sqrt{2\pi\sigma^2}} 2e^{-\frac{(c-a)^2+(d-a)^2}{2\sigma^2}}$$
$$\frac{(b-a)^2}{2} \stackrel{?}{<} (c-a)^2 + (d-a)^2$$
$$(b-a)^2 \stackrel{?}{<} 2(c-a)^2 + 2(d-a)^2$$

Since $c + d = a + b$, assume $c = a + \epsilon$ and $d = b - \epsilon$, where $\epsilon$ is any real number.

$$
\begin{aligned}
(b-a)^2 &\overset{?}{<} 2(a+\epsilon-a)^2 + 2(b-\epsilon-a)^2 \\
(b-a)^2 &\overset{?}{<} 2\epsilon^2 + 2(b-a)^2 + 2\epsilon^2 - 4(b-a\epsilon) \\
0 &\overset{?}{<} 4\epsilon^2 + (b-a)^2 + -4(b-a\epsilon) \\
0 &\overset{?}{<} (2\epsilon - b + a)^2
\end{aligned}
$$

For any real numbers, $a$, $b$, and $c$, $(2\epsilon - b + a)^2$ will always be greater than $0$, except when $\epsilon = \frac{b+a}{2}$, $(2\epsilon - b + a)^2$ will be equal to $0$. When $\epsilon = \frac{b+a}{2}$, $c = a + \frac{b-a}{2}$, i.e. $c = \frac{a+b}{2}$. For any other value of $\epsilon$, instance $\langle \frac{a+b}{2}, \frac{a+b}{2} \rangle$ has a higher density, with respect to the underlying model, than instance $\langle c, d \rangle$.

**3.5.2 Empirical Justifications.** We have shown that the UNC-CE case has lower density than the UNC-IE case, with respect to the underlying naïve Bayes model. Our proof assumed that the instances were nearly perfectly uncertain, i.e. $P(X|Y = +1) = P(X|Y = -1) = 0.5$. In reality, however, it is impractical to assume that the instances lie perfectly on the decision boundary. To analyze such cases, we provide an empirical study to investigate the correlation between density and evidence for instances that are close to decision boundary but not necessarily on the decision boundary of the model.

We created a synthetic dataset using a Bernoulli Naïve Bayes model where the number of features was 10. We assumed that each parameter had a Beta prior, and hence the posterior was also a Beta distribution. Note that even though the joint posterior distribution $P(Y, X|\mathcal{L})$ has a closed-form solution, computing the conditional $P(Y|X, \mathcal{L})$ requires us to resort to sampling. Therefore, rather than plugging in the mean of the posterior distributions for $P(Y|\mathcal{L})$ and $P(X|Y, \mathcal{L})$, we instead sampled their values from their posterior distributions, which gave us a sample over $P(Y|X, \mathcal{L})$, rather than a single point estimate.

Table 3.8. Spearman rank correlations between evidence and density, and evidence and prediction variance, with respect to the model trained on $\mathcal{L}$.

| $\lvert\mathcal{L}\rvert$ | **Evidence's Correlation with** | | | |
|---|---|---|---|---|
| | **Density** | | **Variance** | |
| | Mean | Std. Dev | Mean | Std. Dev |
| 20 | -0.84 | 0.0353 | 0.91 | 0.0139 |
| 40 | -0.93 | 0.0031 | 0.96 | 0.0018 |
| 60 | -0.94 | 0.0025 | 0.97 | 0.0006 |
| 80 | -0.95 | 0.0025 | 0.97 | 0.0006 |
| 100 | -0.92 | 0.0092 | 0.95 | 0.0093 |

Using this sample, we computed the variance of $P(Y|X, \mathcal{L})$.

We tested if, how, and how much the evidence, density, and variance are correlated for the top uncertain instances. We used Equation 2.2 to compute the uncertainty score of all instances $x^{(i)} \in \mathcal{U}$ and considered instances above the threshold of 0.45 uncertainty score to be the uncertain instances. We computed the evidence, which we earlier defined as $E_{+1}(x^{(i)}) \times E_{-1}(x^{(i)})$, for each uncertain instance $x^{(i)}$, and ranked them in increasing order of evidence. Let this ranking be $r_e$. We compared this ranking with the ranking with respect to variance, $r_v$, and with the ranking with respect to density, $r_d$.

We computed the Spearman rank correlation between the evidence-based ranking, $r_e$, and the variance-based ranking, $r_v$. We also computed the Spearman rank correlation between the evidence-based ranking, $r_e$, and the density-based ranking, $r_d$. We computed the correlations for various sizes of labeled data, $\mathcal{L}$. We repeated each experiment 10 times, each time randomly choosing the labeled data $\mathcal{L}$. We report the mean and standard deviation of the correlations over the 10 trials.

Table 3.8 presents the results for Spearman rank correlations between evidence and density, and between evidence and variance of the posterior predictive distribution, of the uncertain instances for various training data sizes, $|\mathcal{L}|$. These results clearly show that

the amount of evidence the model has on uncertain instances and the densities of these uncertain instances with respect to the model are highly negatively correlated (ranging between -0.84 and -0.95), providing empirical evidence that uncertain instances with higher evidence (UNC-CE instances) have lower density in the training data than the uncertain instances with lower evidence (UNC-IE instances). These results further show that the Spearman rank correlation between $r_e$ and $r_v$ is positive and quite high, ranging from 0.91 to 0.97 for various training data sizes, showing that UNC-CE cases have higher variance than the UNC-IE cases.

In Figure 3.8 we plot the histograms of the posterior predictive distributions $P(Y = +1|X, \mathcal{L})$ for two instances for which the model is uncertain for different reasons: conflicting vs. inconclusive evidences. In both cases, the model is equally uncertain on $X$ where the mean of $P(Y = +1|X, \mathcal{L})$ is $0.49$. However, UNC-CE instance (the red histogram) has twice the variance of the UNC-IE instance (the blue histogram), $0.10$ versus $0.05$ respectively. Regular uncertainty sampling for active learning would not make a distinction between these two instances as both have equally high uncertainty of $0.49$, but UNC-CE strategy would prefer the high variance one and the UNC-IE strategy would prefer the low variance one.

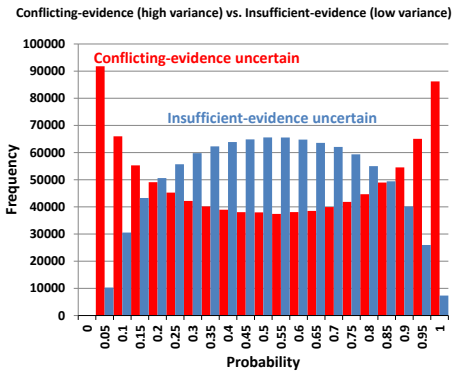



Figure 3.8. The histogram of $P(Y = +1|X, \mathcal{L})$ for two instances that are uncertain for two different reasons: conflicting-evidence vs. insufficient-evidence.

We have seen that the underlying model has higher variance on UNC-CE cases.

Next, we compare UNC-CE and UNC-IE strategies to query-by-committee strategy [97], which chooses instances on which the model has the highest prediction variance.

**3.5.3 Comparison to Query-by-Committee.** Query-by-committee (QBC) [97] is another frequently used baseline in active learning. QBC selects instances that reduce the version space size of the underlying model class [69]. A committee of classifiers is formed by sampling hypotheses from the version space, but since this is not always possible, an approximate version of QBC can be formed by technique known as bagging, as described by Abe and Mamitsuka [1998], and selects instances on which the committee disagrees the most. The two most common approaches to measure the disagreement between committee members are margin of disagreement, i.e. the difference between number of votes for the most popular label and number of votes for the next most popular label [65], and vote entropy [22]. Vote entropy is defined as:

$$x^* = \underset{x^{(i)} \in \mathcal{U}}{\operatorname{argmax}} - \sum_{y \in Y} \frac{V(y)}{C} log \frac{V(y)}{C} \tag{3.15}$$

where $y$ ranges over all possible labels in $Y$, $V(y)$ is the number of votes that a label receives from the committee members, and $C$ is the committee size.

We built a committee of $10$ classifiers using bagging technique described by Abe and Mamitsuka [1998] and used vote entropy [22] as a measure of informativeness of instances. Figs. 3.9 and 3.10 present the learning curves comparing UNC-CE and UNC-IE with $t = 10$ to QBC. These results show that for most datasets and measures, UNC-CE outperforms QBC whereas UNC-IE is worse than QBC. Table 3.9 presents the t-test results comparing UNC-1, UNC-CE, and UNC-IE to QBC. For AUC measure, UNC-CE significantly wins over QBC on seven datasets and loses on one (Spambase), whereas UNC-IE loses to QBC on all datasets except Hiva. For accuracy, UNC-CE significantly outperforms QBC on three datasets and loses on one (Ibn Sina), and for F1, it wins on three datasets and loses

on one (LetterO). UNC-IE loses to QBC for both accuracy and F1 measures for all datasets.

Table 3.9. UNC-1, UNC-CE, and UNC-IE versus QBC. Number of datasets on which UNC-1, UNC-CE, and UNC-IE significantly Win (W), Tie (T), or Lose (L) compared to QBC baseline.

| **QBC baseline** | **AUC** | **ACCU** | **F1** |
|---|---|---|---|
| Method | **W/T/L** | **W/T/L** | **W/T/L** |
| UNC-1 | 3/0/**5** | **2**/0/**2** | **4**/0/0 |
| UNC-CE | **7**/0/1 | **3**/0/1 | **3**/0/1 |
| UNC-IE | 1/0/**7** | 0/0/**4** | 0/0/**4** |

**3.5.4 Discussion.** We presented both analytical and empirical results showing that the conflicting cases have lower density, with respect to the underlying model, than the inconclusive cases. That is, the perceived conflict is supported by a small amount of labeled data whereas the lack of evidence is supported by more labeled data. This suggests that the model is more likely to be incorrect in its reasoning that there is a conflict than its reasoning that there is not enough evidence. Further, we showed that the model has higher variance on the UNC-CE cases than on the UNC-IE cases. Put another way, the model parameters are more "sure" about the uncertainty of the UNC-IE cases (lower variance) and therefore the UNC-IE cases might indeed continue to be inconclusive even if more labeled data is collected. We compared UNC-CE and UNC-IE strategies to QBC and showed that UNC-CE outperforms QBC whereas UNC-IE loses to QBC.

## 3.6 Extension to Other Classifiers and Multi-class Classification

In this section, we describe how the evidence-based framework can be extended to other classifiers. We formally define evidence using multinomial naïve Bayes, logistic regression, linear support vector machines, and non-linear support vector machines. Finally, we discuss how it can be generalized to multi-class classification domains.

**3.6.1 Evidence using Multinomial Naïve Bayes.** The probability of a document, $d^{(i)}$,

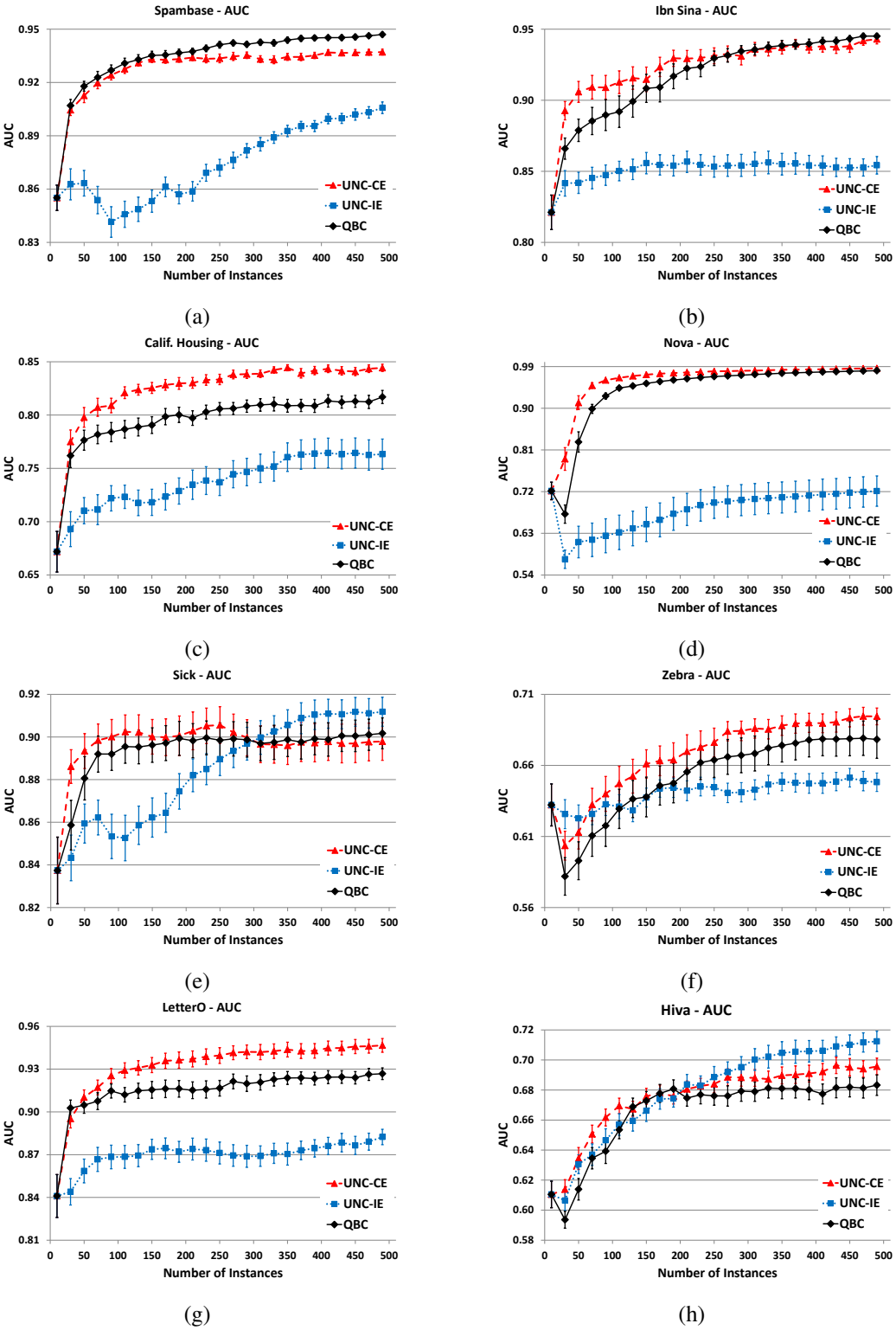



Figure 3.9. AUC results for all eight datasets. UNC-CE outperforms QBC on seven out of eight datasets ((b), (c), (d), (e), (f), (g), and (h)) and loses on Spambase dataset (a). UNC-IE loses to QBC on seven out of eight datasets ((a), (b), (c), (d), (e), (f), and (g), and wins on Hiva dataset (h).

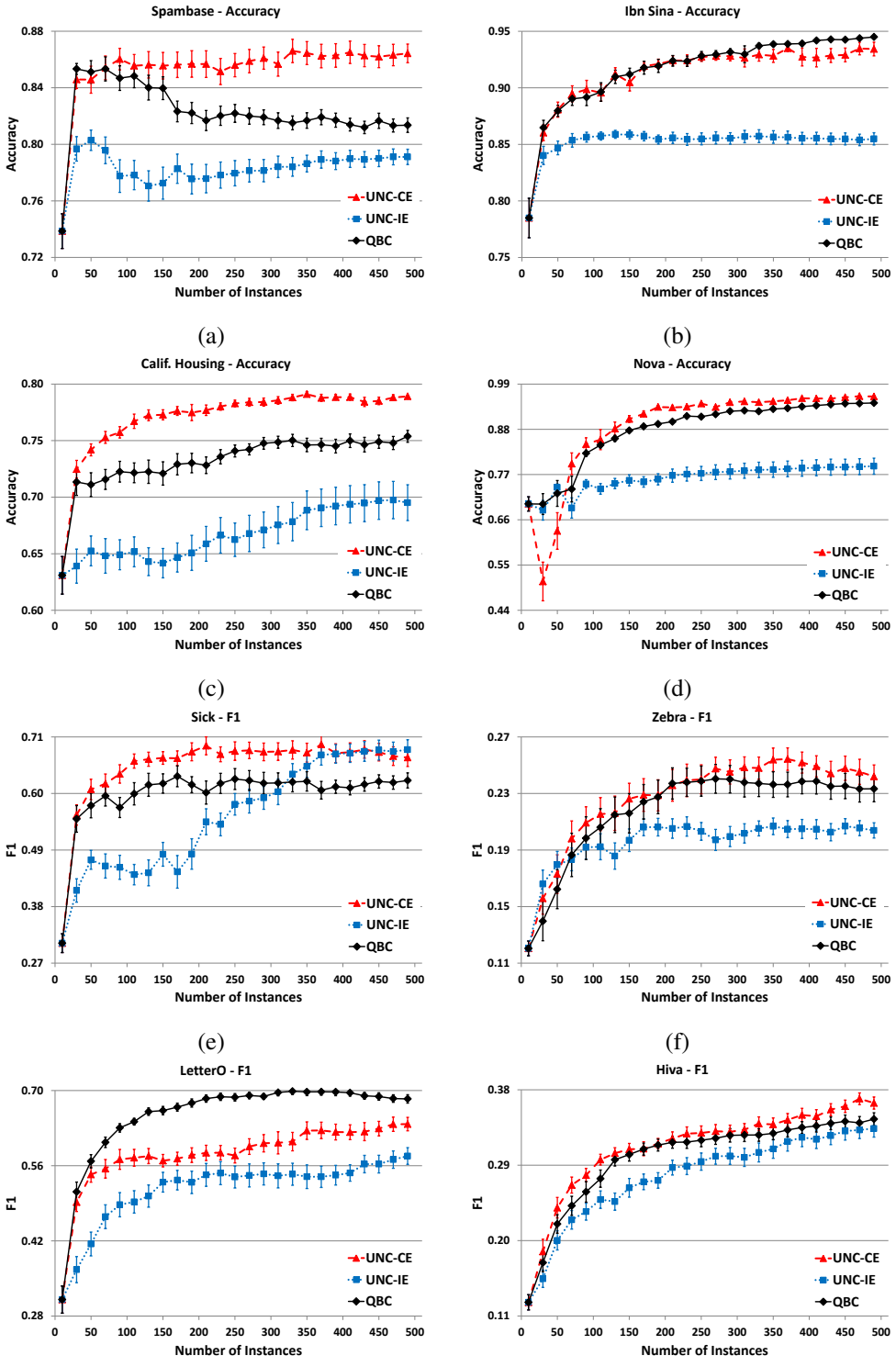



Figure 3.10. Accuracy results for four medium-imbalanced datasets (Spambase, Ibn Sina, Calif. Housing and Nova). UNC-CE outperforms QBC on three datasets ((b), (c) and (d)) and loses on Spambase (a). UNC-IE loses to QBC on all four datasets. F1 results for four relatively skewed datasets (Sick, Zebra, LetterO and Hiva). UNC-CE outperforms QBC significantly on three datasets ((e), (f) and (h)), and loses on one (g). UNC-IE loses to QBC on all four datasets.

belonging to a class +1 is computed using Equation 3.16.

$$P(+1|d^{(i)}) = \frac{P(+1)\prod_{1\leq k^{(i)}\leq n} P(t_k^{(i)}|+1)}{P(d^{(i)})} \tag{3.16}$$

where, $t_k^{(i)}$ is the $k^{th}$ term in a document, $d^{(i)}$, $k^{(i)}$ is the number of terms that appear in document, $d^{(i)}$, and $n$ is the dictionary size. A document $d^{(i)}$ can then be classified based on the ratio of $\frac{P(+1|d^{(i)})}{P(-1|d^{(i)})}$:

$$Y = \begin{cases} +1 & \text{if } \left(\frac{P(+1)}{P(-1)} \prod\limits_{1\leq k^{(i)}\leq n} \frac{P(t_k^{(i)}|+1)}{P(t_k^{(i)}|-1)}\right) > 1 \\ -1 & \text{otherwise} \end{cases} \tag{3.17}$$

From Equation 3.17, it follows that the term $t_k^{(i)}$ of document $d^{(i)}$ provides evidence for the positive class if $\frac{P(t_k^{(i)}|+1)}{P(t_k^{(i)}|-1)} > 1$, and it provides evidence for the negative class otherwise. Let $\mathcal{P}_{d^{(i)}}$ and $\mathcal{N}_{d^{(i)}}$ be two sets, such that $\mathcal{P}_{d^{(i)}}$ contains the terms that provide evidence for the positive class and $\mathcal{N}_{d^{(i)}}$ is the set of terms that provide evidence for the negative class:

$$\mathcal{P}_{d^{(i)}} \triangleq \{t_k^{(i)} \mid \frac{P(t_k^{(i)}|+1)}{P(t_k^{(i)}|-1)} > 1\}$$

$$\mathcal{N}_{d^{(i)}} \triangleq \{t_k^{(i)} \mid \frac{P(t_k^{(i)}|-1)}{P(t_k^{(i)}|+1)} > 1\}$$

Then, the total evidence the document, $d^{(i)}$, provides for the positive class is:

$$E_{+1}(d^{(i)}) = \prod_{t_k^{(i)}\in\mathcal{P}_{d^{(i)}}} \frac{P(t_k^{(i)}|+1)}{P(t_k^{(i)}|-1)} \tag{3.18}$$

and, the total evidence the document provides for the negative class is:

$$E_{-1}(d^{(i)}) = \prod_{t_k^{(i)} \in \mathcal{N}_{d^{(i)}}} \frac{P(t_k^{(i)}|-1)}{P(t_k^{(i)}|+1)} \tag{3.19}$$

**3.6.2 Evidence using Logistic Regression.** The parametric model assumed by logistic regression for binary classification is:

$$P(Y=-1|x) = \frac{1}{1+e^{(w_0+\sum_{j=1}^{f} w_j x_j^{(i)})}} \tag{3.20}$$

$$P(Y=+1|x) = \frac{e^{(w_0+\sum_{j=1}^{f} w_j x_j^{(i)})}}{1+e^{(w_0+\sum_{i=1}^{f} w_j x_j^{(i)})}} \tag{3.21}$$

An instance can then be classified using:

$$Y = \text{sgn}\left(w_0 + \sum_{i=1}^{f} w_j x_j^{(i)}\right) \tag{3.22}$$

From Equation 3.22, it follows that the attribute value $x_j^{(i)}$ of instance $x^{(i)}$ provides evidence for the positive class if $w_j x_j^{(i)} > 0$, and it provides evidence for the negative class otherwise.

Let $\mathcal{P}_{x^{(i)}}$ and $\mathcal{N}_{x^{(i)}}$ be two sets, such that $\mathcal{P}_{x^{(i)}}$ contains the attribute values that provide evidence for the positive class and $\mathcal{N}_{x^{(i)}}$ contains the attribute values that provide evidence for the negative class:

$$\mathcal{P}_{x^{(i)}} \triangleq \{x_j^{(i)} \mid w_j x_j^{(i)} > 0\}$$

$$\mathcal{N}_{x^{(i)}} \triangleq \{x_k^{(i)} \mid w_k x_k^{(i)} < 0\}$$

Then, the total evidence that instance $x^{(i)}$ provides for the positive class is:

$$E_{+1}(x^{(i)}) = \sum_{x_j^{(i)} \in \mathcal{P}_{x^{(i)}}} w_j x_j^{(i)} \tag{3.23}$$

and, the total evidence that instance $x^{(i)}$ provides for the negative class is:

$$E_{-1}(x^{(i)}) = - \sum_{x_k^{(i)} \in \mathcal{N}_{x^{(i)}}} w_k x_k^{(i)} \tag{3.24}$$

**3.6.3 Evidence using Linear Support Vector Machines.** Support Vector Machines (SVM) maximize the margin of classification:

$$w = \underset{w}{\operatorname{argmax}} \left( y \times (w_0 + \sum_{j=1}^{f} w_j x_j^{(i)}) \right) \tag{3.25}$$

and the classification rule is identical to that of logistic regression (Eqn. 3.22):

$$Y = \operatorname{sgn} \left( w_0 + \sum_{i=1}^{f} w_j x_j^{(i)} \right) \tag{3.26}$$

Following the reasoning of evidence using logistic regression, the equations for $E_{+1}(x^{(i)})$ and $E_{-1}(x^{(i)})$ for linear SVM are identical to those for logistic regression.

**3.6.4 Evidence using Non-linear Support Vector Machines.** Non-linear SVM maps the data on to a higher dimensional space and uses a linear classifier in a higher dimensional space. For non-linear SVM, the optimization problem is:

$$w = \underset{w}{\operatorname{argmin}} \lambda \| \ w \ \|^2 + \sum_{l=1}^{m} L(w.\phi(x^{(l)}), y^{(l)}) \tag{3.27}$$

where $w = \sum_{l=1}^{m} \beta_l \phi(x^{(l)})$, $\lambda = \frac{1}{C}$ is the regularization parameter, and $L(y,t) = max(0, 1-yt)^p$ is a loss function. An instance, $x^{(i)}$, is then classified using:

$$Y = \text{sgn} \sum_{l=1}^{m} \beta_l k(x^{(l)}, x^{(i)}) + b \tag{3.28}$$

where $k(x^{(l)}, x^{(i)}) = \phi(x^{(l)})^T.\phi(x^{(i)})$ is a kernel function that defines weighted similarity between $x^{(l)}$ and $x^{(i)}$ and $\beta_l$ is the coefficient which is non-zero for the support vectors and zero for all other instances in the training data.

In case of non-linear SVMs, the evidence that instance $x^{(i)}$ provides for one class or another is it's weighted similarity to the support vectors, $x^{(l)}$, which is defined using a kernel function, $k(x^{(l)}, x^{(i)})$. Let $\mathcal{P}_{x^{(i)}}$ and $\mathcal{N}_{x^{(i)}}$ be two sets for instance $x^{(i)}$, such that $\mathcal{P}_{x^{(i)}}$ contains the support vectors that provide evidence for the positive class for $x^{(i)}$ and $\mathcal{N}_{x^{(i)}}$ contains the support vectors that provide evidence for the negative class for $x^{(i)}$:

$$\mathcal{P}_{x^{(i)}} \triangleq \{x^{(j)} \mid \beta_j k(x^{(j)}, x^{(i)}) > 0\}$$

$$\mathcal{N}_{x^{(i)}} \triangleq \{x^{(k)} \mid \beta_k k(x^{(k)}, x^{(i)}) < 0\}$$

Then, the total evidence that instance $x^{(i)}$ contains for the positive class is:

$$E_{+1}(x^{(i)}) = \sum_{x^{(j)} \in \mathcal{P}_{x^{(i)}}} \beta_j k(x^{(j)}, x^{(i)}) \tag{3.29}$$

and, the total evidence that instance $x^{(i)}$ contains for the negative class is:

$$E_{-1}(x^{(i)}) = \sum_{x^{(k)} \in \mathcal{N}_{x^{(i)}}} \beta_k k(x^{(k)}, x^{(i)}) \tag{3.30}$$

**3.6.5 Evidence for Multi-class Classification.** For binary classification, all three types of

uncertainties (Equations 2.1, 2.2, and 2.3) prefer instances closest to the decision boundary as specified by Equations 3.5, 3.22, and 3.26. However, their preferences differ in multi-class classification. The entropy approach (Equation 2.1), for example, considers overall uncertainty and takes into account all classes, whereas the maximum conditional approach (Equation 2.2) considers how confident the model is about the most likely class. To keep the discussion simple and brief, and as a proof-of-concept, we show how the evidence for multi-class can be extended for naive Bayes (Equation 3.4) when used with the margin uncertainty approach (Equation 2.3).

The margin uncertainty prefers instances for which the difference between the probabilities of most-likely class $y_m$ and next-likely class $y_n$ is minimum. Let $\mathcal{M}_{x^{(i)}}$ and $\mathcal{N}_{x^{(i)}}$ be two sets, such that $\mathcal{M}_{x^{(i)}}$ contains the attribute values that provide evidence for the most-likely class and $\mathcal{N}_{x^{(i)}}$ contains the attribute values that provide evidence for the next likely class:

$$\mathcal{M}_{x^{(i)}} \triangleq \{x_j^{(i)} \mid \frac{P(x_j^{(i)}|y_m)}{P(x_j^{(i)}|y_n)} > 1\}$$

$$\mathcal{N}_{x^{(i)}} \triangleq \{x_k^{(i)} \mid \frac{P(x_k^{(i)}|y_n)}{P(x_k^{(i)}|y_m)} > 1\}$$

Then, the total evidence that instance $x^{(i)}$ provides for the most-likely class (in comparison to the next-likely class) is:

$$E_m(x^{(i)}) = \prod_{x_j^{(i)} \in \mathcal{M}_x^{(i)}} \frac{P(x_j^{(i)}|y_m)}{P(x_j^{(i)}|y_n)} \tag{3.31}$$

and, the total evidence that instance $x^{(i)}$ provides for the next-likely class (in comparison to the most-likely class) is:

$$E_n(x^{(i)}) = \prod_{x_k^{(i)} \in \mathcal{N}_x^{(i)}} \frac{P(x_k^{(i)}|y_n)}{P(x_k^{(i)}|y_m)} \tag{3.32}$$

## 3.7 Conclusion

We introduced an evidence-based framework to uncover the reasons for model's uncertainty on instances, and made the active learner transparent to provide its reasons for selecting instances. We used this framework to distinguish between two types of uncertainties: a model is uncertain about an instance due to strong and conflicting evidence for both classes (conflicting-evidence uncertainty) vs. a model is uncertain because it does not have sufficient evidence for either class (insufficient-evidence uncertainty). The traditional uncertainty sampling does not distinguish between these types of uncertainties, but our empirical evaluations showed that making this distinction had a big impact on the performance of uncertainty sampling. While insufficient-evidence uncertain instances provided the least value to an active learner, actively labeling conflicting-evidence uncertain instances significantly improved the traditional uncertainty sampling. We provided analytical and empirical results showing that the conflicting-evidence instances are underrepresented in the labeled data compared to the insufficient-evidence instances. We further provided empirical results showing that the model has higher variance on the conflicting-evidence instances than on the insufficient-evidence instances. These two results suggest that the model is more likely to be incorrect in its decision that there is a conflict than its decision that the case is inconclusive.

# CHAPTER 4

# RATIONALES FRAMEWORK FOR DOCUMENT CLASSIFICATION

In this chapter, I discuss how we enrich the interaction between the human and active learner and provide a framework to incorporate rich feedback from human expert into the training of predictive models. Specifically, we ask the human expert to read documents and provide labels and rationales for the classification of a document. In this chapter, I focus on text classification task, as labeling documents does not require much domain expertise. In the rationales framework, the learner iteratively selects a document for querying and asks the human to provide a label *and* a rationale for classification by highlighting phrases that convinced him/her to choose a particular label. We provide the rationales framework that uses a simple, and yet effective approach, to incorporate rationales for document classification into the training of any off-the-shelf classifier, such as naïve Bayes, logistic regression, and support vector machines, and show that incorporating rationales into learning can increase the learning efficiency of active learner and minimize the human time and effort in providing supervision.

This chapter is based on the work that I did with my advisor, Dr. Mustafa Bilgic, and Di Zhuang. This work has been published in the North American Chapter of the Association for Computational Linguistics Human Language Technologies, 2015 [101]. An extension of this work as a journal article has been accepted with minor revision in the Machine Learning Journal, 2017.

## 4.1 Introduction

Annotating documents for supervised learning is a tedious, laborious, and time consuming task for humans. Given huge amounts of unlabeled documents, it is impractical for annotators to go over each document and provide a label. To reduce the annotation time and

effort, various approaches such as semi-supervised learning [15] that utilizes both labeled and unlabeled data, and active learning [94] that carefully chooses instances for annotation have been developed. To further minimize the human effort, recent work looked at eliciting domain knowledge, such as rationales and feature annotations, from the annotators instead of just the labels of documents.

Humans can classify instances based on their prior knowledge about feature-class correlations. In order for the classifier to learn similar feature-class correlations from the data, it needs to see many labeled instances. For example, consider the task of sentiment analysis for movie reviews where a classifier is tasked with classifying the reviews as overall positive or overall negative. When the classifier is presented with a negative review that reads "I saw this movie with my friends over the weekend. The movie was terrible.", the classifier does not know which terms in this review are responsible for classifying it as a negative review. Unless the classifier has observed many more negative reviews that have the word "terrible" in them, it would not know that "terrible" is a negative sentiment word, and unless it has seen many positive and negative reviews that have the words "friend" and "weekend" in them, it would not know that these words are potentially neutral sentiment words. In domains where labeled data is scarce, teasing out this kind of information is like searching for a needle in haystack. In learning with rationales framework, in addition to a label, the annotator provides a rationale, pointing out the phrases that are responsible for the assigned label, enabling the classifier to quickly identify the important feature-class correlations and speed up the learning.

A bottleneck in effective utilization of rationales elicited from annotators is that the traditional supervised learning approaches cannot readily handle the elicited rich feedback. To address this issue, many methods have been developed that are classifier-specific. Examples include knowledge-based neural networks [40], [116], [117], knowledge-based support vector machines [39], pooling multinomial naïve Bayes [66], incorporating feature

annotation into locally-weighted logistic regression [25], incorporating constraints into the training of naïve Bayes [108], and converting rationales and feature annotations into constraints for support vector machines [105], [123]. Being classifier-specific limits their applicability when one does not know which classifier is best suited for his/her domain and hence would like to test several classifiers, necessitating a simple and generic approach that can be utilized by several off-the-shelf classifiers.

In this chapter, we present a simple and yet effective approach that can incorporate the elicited rationales in the form of feature annotations into the training of any off-the-shelf classifier. We empirically show that it is effective at incorporating rationales into the learning of naïve Bayes, logistic regression, and support vector machines using four text categorization datasets. We compare our approach to other baselines from the literature. We further discuss a novel active learning strategy specifically geared towards the learning with rationales framework and empirically show that it improves over traditional active learning.

This chapter is organized as follows. In Section 4.2, we provide a brief background on eliciting rationales in the context of active learning. In Section 4.3, we describe our approach for incorporating rationales into the training of classifiers, compare the improvements provided by incorporating rationales into learning to the traditional learning that does not use rationales, and evaluate our approach on a dataset with user-annotated rationales. We compare our method to three baselines, Melville and Sindhwani [2009], Das et al. [2013], and Zaidan et al. [2007] in Section 4.4. In Section 4.5, we present an active learning method using the learning with rationales framework and present relevant results. Finally, we conclude in Section 4.6.

## 4.2 Background

Let $\mathcal{D}$ be a set of document-label pairs $\langle x, y \rangle$, where the label (value of $y$) is known

for only a small subset $\mathcal{L} \subset \mathcal{D}$ of documents: $\mathcal{L} = \{\langle x, y\rangle\}$ and the rest, $\mathcal{U} = \mathcal{D} \setminus \mathcal{L}$, consists of the unlabeled documents: $\mathcal{U} = \{\langle x, ?\rangle\}$. We assume that each document $x^i$ is represented as a vector of features (most commonly as a bag-of-words model with a dictionary of predefined set of phrases, which can be unigrams, bigrams, etc.): $x^i \triangleq \{f_1^i, f_2^i, \cdots, f_n^i\}$. Each feature $f_j^i$ represents the binary presence (or absence), frequency, or tf-idf representation of the word/phrase $j$ in document $x^i$. Each label $y \in \mathcal{Y}$ is a discrete-valued variable: $\mathcal{Y} \triangleq \{y_1, y_2, \cdots, y_l\}$.

Typical greedy active learning algorithms iteratively select an informative document $\langle x^*, ?\rangle \in \mathcal{U}$ according to utility-based heuristics, query a labeler for its label $y^*$, and incorporate the new document $\langle x^*, y^*\rangle$ into the training set, $\mathcal{L}$. This process continues until a stopping criterion is met, usually until a given budget, $B$, is exhausted.

In the learning with rationales framework, in addition to querying for label $y^*$ of document $x^*$, the active learner asks the labeler to provide a rationale, $R(x^*)$, for the chosen label. The rationale in its most general form consists of a subset of the terms that are present in document $x^*$: $R(x^*) = \{f_k^* : k \in x^*\}$. Note that there might be cases where the labeler cannot pinpoint any phrase as a rationale, in which case $R(x^*)$ is allowed to be empty ($\phi$). The labeled set now contains the document-label-rationale triplets $\langle x^*, y^*, R(x^*)\rangle$, instead of the document-label pairs $\langle x^*, y^*\rangle$. Algorithm 2 formally describes the active learning process that elicits rationales from the labeler.

**Algorithm 2** Active Learning with Rationales

1: **Input:** $\mathcal{U}$ - unlabeled documents, $\mathcal{L}$ - labeled documents, $\theta$ - underlying classification model, $B$ - budget
2: **repeat**
3: $\quad x^* = \underset{x \in \mathcal{U}}{\operatorname{argmax}}\, utility(x|\theta)$
4: $\quad$ request label and rationale for this label
5: $\quad \mathcal{L} \leftarrow \mathcal{L} \cup \{\langle x^*, y^*, R(x^*)\rangle\}$
6: $\quad \mathcal{U} \leftarrow \mathcal{U} \setminus \{\langle x^*\rangle\}$
7: $\quad$ Train $\theta$ on $\mathcal{L}$
8: **until** Budget $B$ is exhausted; e.g., $|\mathcal{L}| = B$

The goal of eliciting rationales is to improve the learning efficiency by incorporating domain knowledge. However, it is not trivial to integrate domain knowledge into the state-of-the-art classifiers, such as logistic regression and support vector machines, because the traditional classifiers are able to handle only $\langle x, y \rangle$ pairs and they cannot readily handle $\langle x, y, R(x) \rangle$ triplets. In order to incorporate the additional rationales or feature annotations into learning, a few classifier-specific approaches have been developed, that *modify* the way a classifier is trained. For example, Zaidan et al. [2007] and Raghavan and Allan [2007] introduced constraints for support vector machines to incorporate rationales. Melville and Sindhwani [2009] incorporated feature annotation into multinomial naïve Bayes by training two multinomial naïve Bayes models, one on labeled instances and the other on labeled features, and used linear pooling to combine the two models. Das et al. [2013] utilized locally-weighted logistic regression to incorporate feature labels into logistic regression by locally fitting a logistic function on instances around a small neighborhood of test instances and taking into account the labeled features. We next describe our approach that can readily incorporate rationales into any classifier by modifying the training data, without requiring changes to the training algorithm of a classifier.

## 4.3 Learning with Rationales

In this section we first provide the formulation of our approach to incorporate rationales into learning and then present results comparing *learning with rationales* (LwR) to *learning without rationales* (Lw/oR) on four document classification datasets. We evaluate our approach using multinomial naïve Bayes, logistic regression, and support vector machines classifiers.

**4.3.1 Training a Classifier Using Labels and Rationales.** Like most previous work, we assume that the rationales, i.e. the phrases, returned by the labeler already exist in the dictionary of the vectorizer. Hence, the rationales correspond to features in our vector representation. It is possible that the labeler returns a phrase that is currently not in the dic-

tionary; for example, the labeler might return a phrase that consists of three words whereas the representation has single words and bi-grams only. In that case, the representation can be enriched by creating and adding a new feature that represents the phrase returned by the labeler.

Our simple approach works as follows: we modify the features of the annotated document $\langle x^i, y^i, R(x^i)\rangle$ to emphasize the rationale(s) and de-emphasize the remaining phrases in that document. We simply multiply the features corresponding to phrase(s) that are returned as rationale(s) by weight $r$ and we multiply the remaining features in the document by weight $o$, where $r > o$, and $r$ and $o$ are hyper-parameters. The modified document becomes:

$$x_r^i = \langle r \times f_j^i, \forall f_j^i \in R(x^i); o \times f_j^i, \forall f_j^i \notin R(x^i)\rangle \tag{4.1}$$

Note that the rationales are tied to documents for which they were provided as rationales. One phrase might be a rationale for the label of one document and yet it might not be a rationale for the label of another document. Hence, the feature weightings are done at the document level, rather than globally. To illustrate this concept, we provide an example dataset below with three documents. In these documents, the words that are returned as rationales are underlined.

*Document 1: This is a great movie.*
*Document 2: The plot was great, but the performance of the actors was terrible. Avoid it.*
*Document 3: I've seen this at an outdoor cinema; great atmosphere. The movie was terrific.*

As these examples illustrate, the word "great" appears in all three documents, but it is marked as a rationale only for *Document 1*. Hence, we do not weight the rationales globally; rather, we modify only the labeled document using its particular rationale. Table 6.1 illustrates the Lw/oR and LwR representations for these documents.

Table 4.1. The Lw/oR binary representation (top) and its LwR transformation (bottom) for Documents 1, 2, and 3. Stop words are removed. LwR multiplies the rationales with $r$ and other features with $o$.

| | great | movie | plot | performance | actor | terrible | avoid | outdoor | cinema | atmosphere | terrific |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Lw/oR Representation (binary) | | | | | | | | | | | |
| *Document 1* | 1 | 1 | | | | | | | | | |
| *Document 2* | 1 | | 1 | 1 | 1 | 1 | 1 | | | | |
| *Document 3* | 1 | 1 | | | | | | 1 | 1 | 1 | 1 |
| LwR Transformation of the binary Lw/oR representation | | | | | | | | | | | |
| *Document 1* | $r$ | $o$ | | | | | | | | | |
| *Document 2* | $o$ | | $o$ | $o$ | $o$ | $r$ | $r$ | | | | |
| *Document 3* | $o$ | $o$ | | | | | | $o$ | $o$ | $o$ | $r$ |

Our approach modifies the training data, in which the rationale features are weighted higher than the other features, and hence our approach can incorporate rationales into the training of any off-the-shelf classifier, without requiring changes to the training algorithm of a classifier. In our approach, the training algorithm of a classifier uses the modified training data to estimate the parameters of the model. This approach is simple, intuitive, and classifier-agnostic. As we will show later, it is quite effective empirically as well. To gain a theoretical understanding of this approach, consider the work on regularization: the aim is to build a sparse/simple model that can capture the most important features of the training data and thus have large weights for important features and small/zero weights for irrelevant features. For example, consider the gradient of weight $w_j$ for feature $f_j$ for logistic regression with $l_2$ regularization (assuming $y$ is binary with 0/1):

$$\nabla w_j = C \times \sum_{x^l \in \mathcal{L}} f_j^l \times (y^l - P(y = 1|x^l)) - w_j \tag{4.2}$$

where $C$ is the complexity parameter that balances between fit to the data and the model complexity. With our rationales framework, the gradient for $w_j$ will be:

$$\nabla w_j = C \times \left( \sum_{x^l \in \mathcal{L}: f_j^l \in R(x^l)} r \times f_j^l \times (y^l - P(y^l = 1|x^l)) + \sum_{x^l \in \mathcal{L}: f_j^l \notin R(x^l)} o \times f_j^l \times (y^l - P(y^l = 1|x^l)) \right) - w_j \tag{4.3}$$

In Equation 4.3, feature $f_j$ contributes more to the gradient of weight $w_j$ when a document in which it is marked as a rationale is misclassified. When $f_j$ appears in another document $x^k$, but is not a rationale, its contribution to the gradient is muted by $o$. Hence, when $r > o$, this framework implicitly provides more granular (per instance-feature combination) regularization by placing a higher importance on the contribution of the rationales versus non-rationales in each document.[4]

Note that in our framework, the rationales are tied to their own documents; that is, we do not weight rationales and non-rationales globally. In addition to providing more granular regularization, this approach has the benefit of allowing different rationales to contribute differently to the objective function of the trained classifier. For example, consider the case where the number of documents in which word $f_j$ (e.g., "excellent") is marked as a rationale is much more than the number of documents in which another word $f_k$ (e.g., "good") is marked as a rationale. In this case, the first summation term in Equation 4.3 will range over more documents for the gradient of $w_j$ compared to the gradient of $w_k$, giving more importance to $w_j$ than to $w_k$. In the traditional feature annotation work, this

[4]The justification for our approach is similar for support vector machines. The idea is also similar for multinomial naïve Bayes with Dirichlet priors $\alpha_j$. For a fixed Dirichlet prior with $\langle \alpha_1, \alpha_2, \cdots, \alpha_n \rangle$ setting, when $o < 1$ for a feature $f_j$, its counts are smoothed more.

Table 4.2. Description of the datasets: the domain, number of instances in training and test datasets, and size of vocabulary.

| Dataset | Task | Train | Test | Vocabulary |
|---|---|---|---|---|
| IMDB | Sentiment analysis of movie reviews | 25,000 | 25,000 | 27,272 |
| NOVA | Email classification (politics versus religion) | 12,977 | 6,498 | 16,969 |
| SRAA | Aviation vs. auto document classification | 48,812 | 24,406 | 31,883 |
| WvsH | 20Newsgroups (Windows vs. hardware) | 1,176 | 783 | 4,026 |

can be achieved only if the labeler can rank the features; but then, it is often very difficult, if not impossible, for the labelers to determine how much more important one feature is compared to another.

**4.3.2 Experiments Comparing LwR to Lw/oR.** In this section we first describe the settings, datasets, and classifiers used for our experiments and how we simulated a human labeler to provide rationales. Then, we present results comparing the learning curves achieved with *learning without rationales* (Lw/oR) and *learning with rationales* (LwR).

**4.3.2.1 Methodology.** For this study, we used four document classification datasets. IMDB dataset consists of movie reviews [61]. Nova is a text classification dataset used in active learning challenge [44]. SRAA[5] dataset consists of documents that discuss either auto or aviation. WvsH[6] is a 20 Newsgroups dataset using the Windows vs. hardware categories. We provide the description of these datasets in Table 3.1. IMDB and WvsH had separate train and test datasets. For NOVA and SRAA datasets, we randomly selected two-thirds of the documents for train dataset and the remaining one-third of the documents were used as the test dataset. We treated the train datasets as unlabeled set, $\mathcal{U}$, in Algorithm 2.

We used the bag-of-words representation of documents with a dictionary of predefined vocabulary of phrases, consisting of only unigrams. To test whether our approach

[5]http://people.cs.umass.edu/ mccallum/data.html

[6]http://qwone.com/ jason/20Newsgroups/

works across representations, we experimented with both binary and tf-idf representations for these text datasets. We evaluated our method using multinomial naïve Bayes, logistic regression, and support vector machines, as these are strong classifiers for text classification. We used the scikit-learn [78] implementation of these classifiers with their default parameter settings for the experiments in this section.

To compare various strategies, we used learning curves. The initially labeled dataset was bootstrapped using 10 documents by picking 5 random documents from each class. A budget, $B$, of 200 documents was used in our experiments, because most of the learning curves flattened out after about 200 documents. We evaluated all the strategies using AUC (Area Under an ROC Curve) measure. The code to repeat our experiments is available on Github (http://www.cs.iit.edu/~ml/code/).

While incorporating the rationales into learning, we set the weights for rationales and the remaining features of a document as 1 and 0.01 respectively (i.e., $r = 1$ and $o = 0.01$). That is, we did not overemphasize the features corresponding to rationales but rather de-emphasized the remaining features in the document. These weights worked reasonably well for all four datasets, across all three classifiers, and using both binary and tf-idf data representations.

Obviously, these are not necessarily the best weight settings one can achieve; the optimal settings for $r$ and $o$ depend on many factors, such as the extent of the knowledge of the labeler (i.e., how many words a labeler can recognize), how noisy the labeler is, and how much labeled data there is in the training set. A more practical approach is to tune these parameters (e.g., using cross-validation) at each step of the learning curve. For simplicity, in this section, we present results using fixed weights for $r$ and $o$ as $1$ and $0.01$ respectively. Later, in Section 4.4, we present results by tuning the weights $r$ and $o$ using

cross-validation on labeled data.

**4.3.2.2 Simulating the Human Expert.** Like most literature on feature labeling, we constructed an artificial labeler to simulate a human labeler, to allow for large-scale experimentation on several datasets and parameter configurations. Every time a document is annotated, we asked the artificial labeler to mark a word as a rationale for the chosen label. We allowed the labeler to return any one, and not necessarily the top one, of the positive words as a rationale for a positive document and any one of the negative words as a rationale for a negative document. If the labeler did not recognize any of the words as positive (negative) in a positive (negative) document, we let the labeler return null ($\phi$) as the rationale.

To make this as practical as possible in a real-world setting, we constructed the artificial labeler to recognize only the most apparent words in the documents. For generating rationales, we chose only the positive (negative) features that had the highest $\chi^2$ (chi-squared) statistic in at least 5% of the positive (negative) documents. This resulted in an overly-conservative labeler that recognized only a tiny subset of the words as rationales. For example, the artificial labeler knew about only 49 words out of $27272$ for IMDB, $111$ words out of $16969$ for NOVA, 67 words out of $31883$ for SRAA, and $95$ words out of $4026$ for WvsH dataset.

To determine whether the rationales selected by this artificial labeler are meaningful, we printed the actual words returned as rationales for IMDB dataset in Figure 4.1, and verified that a majority of these words are human-recognizable words that could be naturally provided as rationales for classification. For example, the positive terms for the IMDB dataset included “great”, “excellent”, and “wonderful” and the negative terms included “worst”, “bad”, and “waste”. As Figure 4.1 shows, the rationales returned by the artificial labeler are unigrams.

| |
|---|
| ‘great’, ‘excellent’, ‘wonderful’, ‘perfect’, ‘best’, ‘amazing’, ‘beautiful’, ‘love’, ‘favorite’, ‘loved’, ‘superb’, ‘brilliant’, ‘highly’, ‘fantastic’, ‘today’, ‘performance’, ‘beautifully’, ‘also’, ‘always’, ‘both’, ‘heart’, ‘performances’, ‘touching’, ‘wonderfully’, ‘enjoyed’, ‘well’ |
| ‘worst’, ‘bad’ , ‘waste’, ‘awful’, ‘terrible’, ‘stupid’, ‘worse’, ‘boring’, ‘horrible’, ‘poor’, ‘nothing’, ‘crap’, ‘minutes’, ‘supposed’, ‘poorly’, ‘no’, ‘lame’, ‘ridiculous’, ‘plot’, ‘script’, ‘avoid’, ‘dull’, ‘mess’ |

Figure 4.1. Words selected as rationales for positive movie reviews (top) and negative movie reviews (bottom) for IMDB dataset.

**4.3.2.3 Results.** Figure 4.2 presents the learning curves comparing LwR to Lw/oR on four document classification datasets with binary and tf-idf representations and using multinomial naïve Bayes, logistic regression, and support vector machines. We made sure that both Lw/oR and LwR work with the same set of documents, and the only difference between them is that in Lw/oR, the labeler provides only a label whereas in LwR, the labeler provides both a label and a rationale. Hence, the difference between the learning curves of Lw/oR and LwR stems not from choosing different documents but rather from incorporating rationales into learning. Figure 4.2 shows that even though the artificial labeler knew about only a tiny subset of the vocabulary, and returned any *one* word, rather than the top word or all the words, as a rationale, LwR drastically outperformed Lw/oR across all datasets, classifiers, and representations. These results show that our method for incorporating rationales into the learning process is quite effective.

LwR provides improvements over Lw/oR, especially at the beginning of learning, when the labeled data is limited. LwR improves learning by enabling the classifier to quickly identify important feature-class correlations using the rationales provided by labeler. When the labeled data is large, Lw/oR can surpass LwR when $r >> o$. Ideally, one should have $r >> o$ when the labeled data is small and $r$ should be closer to $o$ when the

labeled data is large. A more practical approach would be to tune these parameters (e.g., using cross-validation, as we later present in Section 4.4.2.2) at each iteration of learning. We empirically found that most settings where $r > o$ in LwR approach performed better than Lw/oR. In this section, for simplicity, we set $r = 1$ and $o = 0.01$.

As discussed in Section 4.3.2.1, we used the default complexity parameters for logistic regression and support vector machines and used Laplace smoothing for multinomial naïve Bayes. Since most features are expected to be non-rationales, in Equation 4.3, most features will appear in the second summation term, with $o = 0.01$. We tested whether the improvements that LwR provides over Lw/oR are simply due to implicit higher regularization for most of the features with $o = 0.01$, and hence experimented with Equation 4.2 (which is Lw/oR) using $C = 0.01$. We observed that setting $C = 0.01$ and indiscriminately regularizing all the terms did not improve Lw/oR on most datasets and classifiers using both binary and tf-idf representations, providing experimental evidence that the improvements provided by LwR are not due to just higher regularization, but they are due to a more fine-grained regularization, as explained in Section 4.3.1. We present one such result for IMDB dataset using logistic regression in Figure 4.3(a).

Similarly, since most features in LwR representation had a weight of $0.01$, and only a handful of features had a weight of $1$, we repeated all the experiments using $r = 0.01$ and $o = 0.01$ to test whether indiscriminately decreasing the weights for all the terms in all the documents provides any improvement in Lw/oR. One would not expect that decreasing the weights for all the terms in all the documents would provide any improvement in learning, however, the LwR representation with $r = 1$ and $o = 0.01$ is quite similar to the representation where $r = 0.01$ and $o = 0.01$, because all the words, except the rationale word, in a document have a weight of $0.01$. As expected, we found that for all datasets and classifiers and using both binary and tf-idf representations, indiscriminately multiplying all the terms by $0.01$, i.e. setting $r = 0.01$ and $o = 0.01$, did not improve Lw/oR, providing

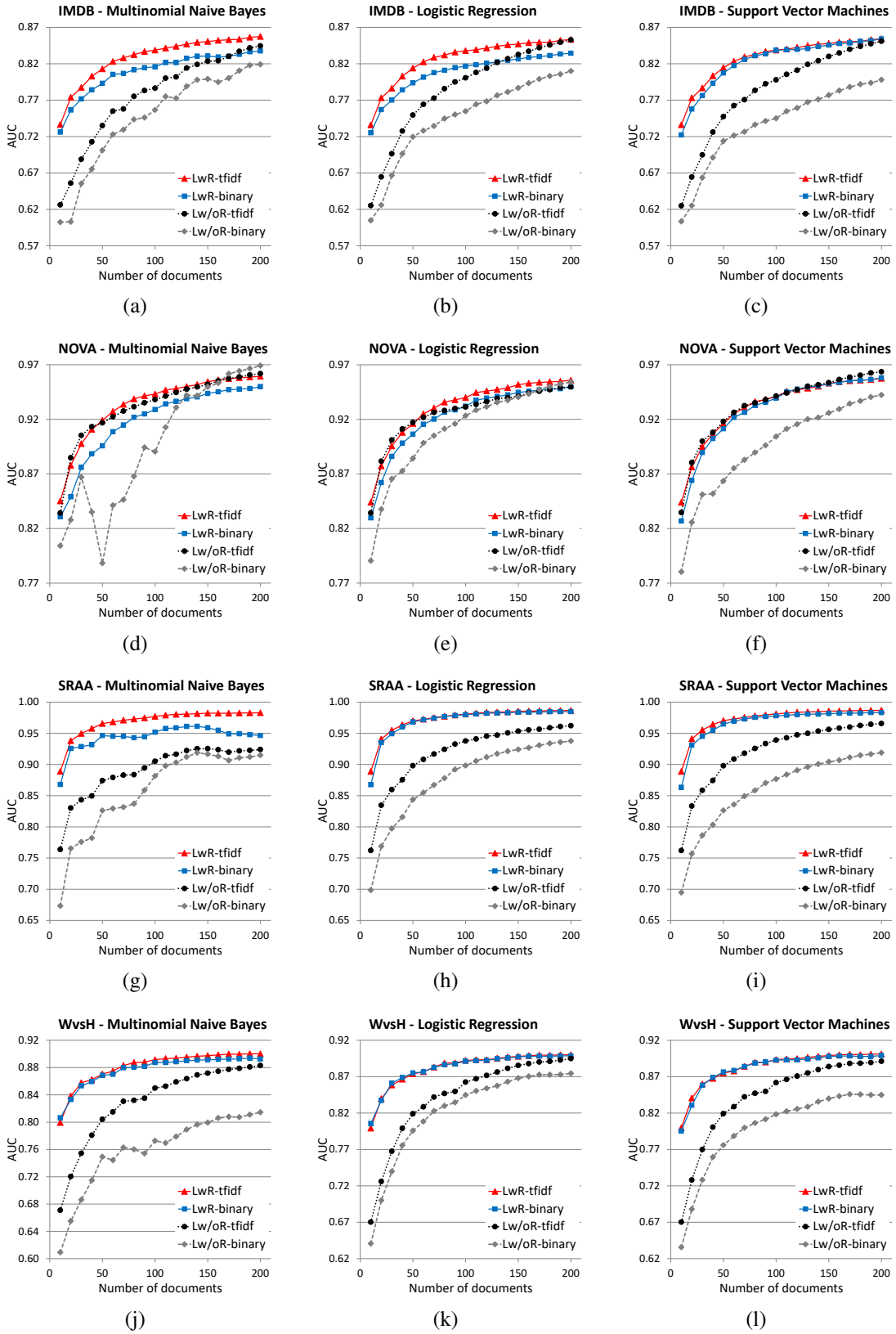
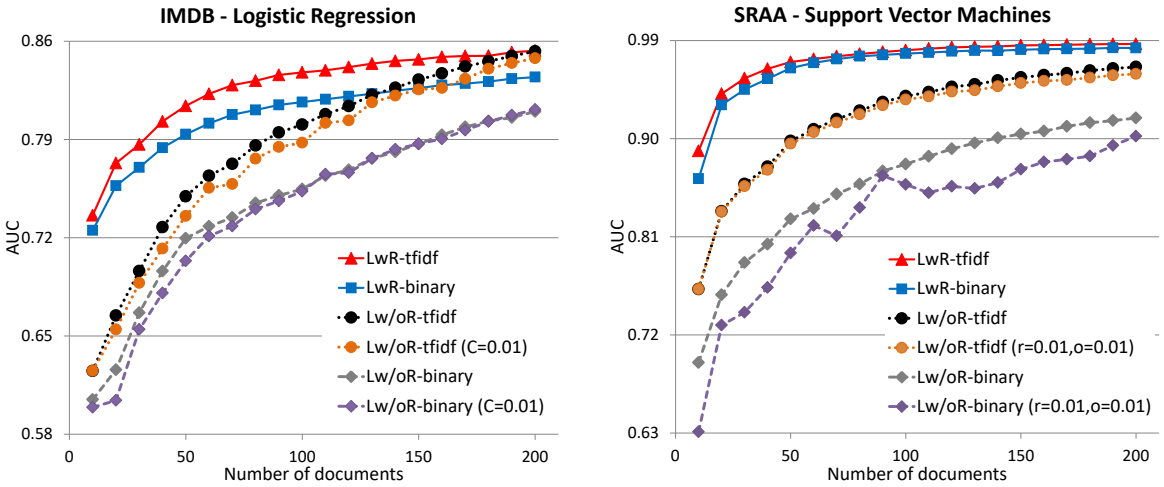



Figure 4.2. Comparison between LwR and Lw/oR using multinomial naïve Bayes, logistic regression, and support vector machines on four datasets: IMDB ((a), (b), and (c)), NOVA ((d), (e), and (f)), SRAA ((g), (h), and (i)), and WvsH ((j), (k), and (l)). LwR provides drastic improvements over Lw/oR for all datasets with binary and tf-idf representations and using all three classifiers.

further experimental evidence that the improvements provided by LwR over Lw/oR are not just due to placing smaller weights on all the terms. We present one such result for SRAA dataset using support vector machines in Figure 4.3(b).




(a) (b)

Figure 4.3. (a) Results showing the effect of setting $C = 0.01$ for Lw/oR using binary and tf-idf representations. (b) Results showing the effect of multiplying the weights for all features by $0.01$, i.e. setting $r = 0.01$ and $o = 0.01$. Using a higher regularization, $C = 0.01$, for Lw/oR or indiscriminately multiplying the weights of all features by $0.01$ does not provide improvement over Lw/oR.

Even though LwR improves performance drastically over Lw/oR, providing both a label and a rationale is expected to take more time of the labeler than simply providing a label. The question then is how to best utilize the labeler's time and effort: is it better to ask for only the labels of documents or should we elicit rationales along with the labels? To test how much a document annotated with a label and a rationale is worth, we computed how many documents a labeler would need to inspect to achieve a target AUC performance, using Lw/oR and LwR. Table 4.3 and Table 4.4 present the number of documents required to achieve a target AUC using Lw/oR and LwR for multimonial naïve Bayes using binary and tf-idf representations.

Tables 4.3 and 4.4 show that LwR drastically accelerates learning compared to Lw/oR, and it requires relatively very few annotated documents for LwR to achieve the

same target AUC as Lw/oR. For example, in order to achieve a target AUC of $0.95$ for SRAA dataset (using tf-idf representation with MNB classifier), Lw/oR required labeling 656 documents, whereas LwR required annotating a mere 29 documents. That is, if the labeler is spending a minute per document to simply provide a label, then it is better to provide a label *and* a rationale as long as providing both a label and a rationale does not take more than $656/29 \approx 22$ minutes of labeler's time. The results for logistic regression and support vector machines using both binary and tf-idf representations are similar, and hence they are omitted to avoid redundancy.

Table 4.3. Comparison of number of documents needed to be annotated to achieve various target AUC performances using Lw/oR and LwR with multinomial naïve Bayes using binary representation. 'N/A' represents that a target AUC cannot be achieved by the learning strategy.

| **Dataset** | **Target AUC** | **0.60** | **0.65** | **0.70** | **0.75** | **0.80** | **0.85** | **0.90** | **0.95** |
|---|---|---|---|---|---|---|---|---|---|
| IMDB | Lw/oR-binary | 23 | 63 | 79 | 102 | 152 | 339 | N/A | N/A |
| | LwR-binary | 2 | 5 | 11 | 22 | 62 | 257 | N/A | N/A |
| | **Ratio** | **11.5** | **12.6** | **7.2** | **4.6** | **2.5** | **1.3** | **N/A** | **N/A** |
| NOVA | Lw/oR-binary | 2 | 5 | 98 | 134 | 160 | 201 | 304 | 584 |
| | LwR-binary | 2 | 2 | 5 | 6 | 11 | 24 | 51 | N/A |
| | **Ratio** | **1** | **2.5** | **19.6** | **22.3** | **14.5** | **8.4** | **5.9** | **N/A** |
| SRAA | Lw/oR-binary | 6 | 9 | 25 | 76 | 100 | 188 | 294 | 723 |
| | LwR-binary | 2 | 2 | 3 | 5 | 7 | 9 | 20 | N/A |
| | **Ratio** | **3** | **4.5** | **8.3** | **15.2** | **14.3** | **20.9** | **14.7** | **N/A** |
| WvsH | Lw/oR-binary | 6 | 17 | 28 | 38 | 139 | 693 | N/A | N/A |
| | LwR-binary | 2 | 3 | 4 | 6 | 12 | 32 | 200 | N/A |
| | **Ratio** | **3** | **5.7** | **7** | **6.3** | **11.6** | **21.7** | **N/A** | **N/A** |

Zaidan et al. [2007] conducted user studies and showed that providing 5 to 11 rationales and a class label per document takes roughly twice the time of providing only the label for the document. In our experiments, the labeler was asked to provide *any* one ra-

Table 4.4. Comparison of number of documents needed to be annotated to achieve various target AUC performances using Lw/oR and LwR with multinomial naïve Bayes using tf-idf representation. 'N/A' represents that a target AUC cannot be achieved by the learning strategy.

| Dataset | Target AUC | 0.60 | 0.65 | 0.70 | 0.75 | 0.80 | 0.85 | 0.90 | 0.95 |
|---|---|---|---|---|---|---|---|---|---|
| IMDB | Lw/oR-tfidf | 7 | 14 | 37 | 65 | 106 | 233 | 841 | N/A |
| | LwR-tfidf | 2 | 4 | 10 | 16 | 37 | 164 | N/A | N/A |
| | **Ratio** | **3.5** | **3.5** | **3.7** | **4.1** | **2.9** | **1.4** | **N/A** | **N/A** |
| NOVA | Lw/oR-tfidf | 2 | 2 | 3 | 3 | 5 | 12 | 28 | 126 |
| | LwR-tfidf | 2 | 2 | 2 | 3 | 4 | 11 | 31 | 110 |
| | **Ratio** | **1** | **1** | **1.5** | **1** | **1.2** | **1.1** | **0.9** | **1.1** |
| SRAA | Lw/oR-tfidf | 2 | 4 | 7 | 12 | 21 | 58 | 109 | 656 |
| | LwR-tfidf | 2 | 2 | 3 | 4 | 6 | 8 | 13 | 29 |
| | **Ratio** | **1** | **2** | **2.3** | **3** | **3.5** | **7.3** | **8.4** | **22.6** |
| WvsH | Lw/oR-tfidf | 5 | 9 | 17 | 33 | 57 | 127 | 380 | N/A |
| | LwR-tfidf | 2 | 3 | 4 | 6 | 12 | 33 | 188 | N/A |
| | **Ratio** | **2.5** | **3** | **4.3** | **5.5** | **4.8** | **3.8** | **2** | **N/A** |

tionale instead of all the rationales. Hence, even though we do not know for sure whether labelers would take more/less time in providing one rationale as opposed to all the rationales, Tables 4.3 and 4.4 show that documents annotated with rationales are often worth at least as two and sometimes more than even 20 documents that are simply annotated with labels.

**4.3.2.4 Results with User-Annotated Rationales.** We evaluated our approach on user-annotated IMDB dataset provided by Zaidan et al. [2008]. The dataset consists of 1800 IMDB movie reviews for which a user provided rationales for labeled documents. The main difference between the simulated expert and the user-annotated dataset is that the simulated expert selected only one word as a rationale, whereas the human highlighted many words, and sometimes even phrases, as rationales. Simulated rationales can also be

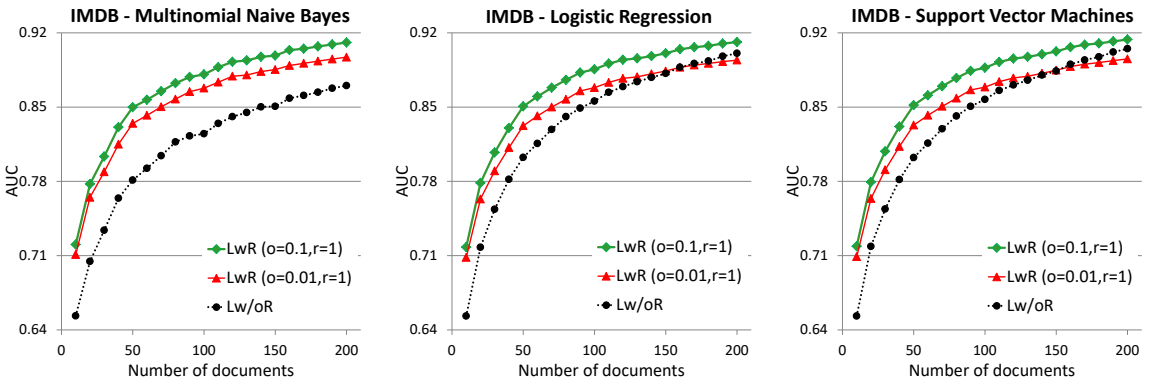
noisy; in our study, the simulated labeler returns *any* one word as a rationale, but in real life, it might not be the rationale.

We performed 5-fold cross validation and repeated each experiment 5 times for each fold and present average results. We used tf-idf representation of the dataset. Figure 4.4 presents the results on user-annotated IMDB dataset comparing LwR to Lw/oR using multinomial naïve Bayes, logistic regression, and support vector machines. We found that LwR performed better than Lw/oR using the default weight settings ($r = 1$ and $o = 0.01$). However, user-annotated rationales can be really noisy, where users do not necessarily pinpoint just the important words, but rather highlight phrases (or even sentences) that span several words. When the expert is noisy, the trust in the expert should be reflected in the weights $r$ and $o$. If the user is trustworthy and precise in pin-pointing the rationales, then $r$ should be much greater than $o$, but if the user is noisy, then $r$ should be relatively closer to $o$.

To test the effect of weights $r$ and $o$ on noisy rationales, we experimented with various settings for $r$ and $o$ between 0.001 and 1000. For user-annotated IMDB dataset, we found that weight settings where $r$ was closer to $o$ worked better than weight settings where $r$ was much greater than $o$. In general, the default setting of $r$=1 and $o$=0.01 worked well for the simulated labeler case and the setting $r = 1$ and $o = 0.1$ worked well for the user-annotated case.

## 4.4 Comparison with Baselines

In this section, we empirically compare our approach to incorporate rationales with other classifier-specific approaches from the literature. Our experiments were based on three classifiers: multinomial naïve Bayes, logistic regression, and support vector machines. Hence, we looked for classifier-specific approaches in the literature that focused on these three classifiers.




(a) (b) (c)

Figure 4.4. Comparison of LwR to Lw/oR on user-annotated IMDB dataset with tf-idf representation using (a) multinomial naïve Bayes, (b) logistic regression, and (c) support vector machines. LwR with default weight setting of $r = 1$ and $o = 0.01$ provides improvements over Lw/oR using all three classifiers. Since user-annotated rationales can be rather noisy, LwR with weights $r = 1$ and $o = 0.1$ performs better than LwR with weights $r = 1$ and $o = 0.01$.

When the underlying classifier is support vector machines, the closest work to ours is that of Zaidan et al. [2007], in which they incorporated rationales into the training of support vector machines, so we chose this as a baseline for our approach using support vector machines. When the underlying classifier is multinomial naïve Bayes, we are not aware of any approach specifically developed to incorporate rationales into learning. The closest work to learning with rationales is feature annotation (e.g., [66], [81], and [109]), in which labelers annotate features independent of the documents. Even though learning with rationales is not the same as feature annotation, learning with rationales can be treated as feature annotation if the underlying rationales correspond to features. Melville and Sindhwani [2009] presented pooling multinomials to incorporate feature annotations into the training of multinomial naïve Bayes, hence we chose this as a baseline for our approach using multinomial naïve Bayes. We are not aware of any approach specifically developed to incorporate rationales into the training of logistic regression classifier, and the closest work is that of Das et al. [2013], which was specifically designed to incorporate feature annotation into the training of locally-weighted logistic regression, and hence we chose it

as a baseline for our approach using logistic regression.

**4.4.1 Description of the Baselines.** In this section, we describe the three baselines, Zaidan et al. [2007], Melville and Sindhwani [2009], and Das et al. [2013], to which we compare our approach.

**4.4.1.1 Description of Zaidan et al. [2007].** Zaidan et al. [2007] presented a method to incorporate rationales into the training of support vector machines. They asked labelers to highlight the most important words and phrases as rationales to justify why a movie review is labeled as positive or negative. For each document, $x^i$, annotated with a label and one or more rationales, one or more contrast examples, $v^{ij}$ (where $j$ is the number of rationales for document $x^i$), is created that resembles $x^i$, but lacks the evidence (rationale) that the annotator found significant, and new examples $x^{ij} \stackrel{\text{def}}{=} \frac{x^i - v^{ij}}{\mu}$ along with their class labels, $\langle x^{ij}, y^i \rangle$, are added to the training set, where $\mu$ controls the desired margin between the original and contrast examples. The soft-margin SVM chooses $w$ and $\xi_i$ to minimize:

$$\min_w \frac{1}{2} \|w\|^2 + C(\sum_{i=1}^{n} \xi_i) \tag{4.4}$$

subject to the constraints:

$$(\forall i) w \cdot x^i \cdot y^i \geq 1 - \xi_i \tag{4.5}$$

$$(\forall i) \xi_i \geq 0 \tag{4.6}$$

where $x^i$ is a training document, $y^i \in \{-1, +1\}$ is the class, and $\xi_i$ is the slack variable. The parameter $C > 0$ controls the relative importance of minimizing $w$ and cost of the

slack. In their approach, they add the contrast constraints:

$$(\forall i, j) w \cdot (x^i - v^{ij}) \cdot y^i \geq \mu(1 - \xi_{ij}) \tag{4.7}$$

where $\xi_{ij} > 0$ is the associated slack variable. The contrast constraints have their own margin, $\mu$, and the slack variables have their own cost, so their objective function for support vector machines becomes:

$$\min_w \frac{1}{2} \|w\|^2 + C(\sum_{i=1}^{n} \xi_i) + C_{contrast}(\sum_{i,j} \xi_{ij}) \tag{4.8}$$

In Zaidan et al. [2007], for each document, one contrast example, $v^{ij}$, and several pseudoexamples, $x^{ij}$, for the rationales are created. Hence, according to Equation 4.8, the hyperplane is determined by whether the contrast examples or the pseudoexamples add to the loss function or participate in the optimization as a support vector. Analytically, our approach is equivalent to Zaidan et al. [2007] when all of the following three conditions hold: (i) $C = C_{contrast}$, (ii) $o = 1$ and $r = \frac{1}{\mu}$, and (iii) in our approach, if a document $x^i$ becomes a support vector, then in Zaidan et al. [2007] approach, both the contrast example, $v^{ij}$, and pseudoexamples, $x^{ij}$, for the document $x^i$ also become support vectors.

**4.4.1.2 Description of Melville and Sindhwani [2009].** Melville and Sindhwani [2009] presented an approach to incorporate feature labels and instance labels into the training of a multinomial naïve Bayes classifier. They build two multinomial naïve Bayes models: one trained on labeled instances and the other trained on labeled features. The two models are then combined using linear pooling [67] to aggregate the conditional probabilities, $P(f_j|y_k)$ using:

$$P(f_j|y_k) = \beta P_e(f_j|y_k) + (1 - \beta) P_f(f_j|y_k) \tag{4.9}$$

where $y_k$ is the class, $P_e(f_j|y_k)$ and $P_f(f_j|y_k)$ represent the probabilities assigned by the

model trained on labeled instances and the model trained on labeled features respectively, and $\beta$ is the weight for combining these two conditional probability distributions.

In order to build a model trained on labeled features, Melville et al. [2009] assumed that a positive term, $f+$, is more likely to appear in a positive document than in a negative document and a negative term, $f-$, is more likely to appear in a negative document than in a positive document. To build a model trained on labeled features, they specified a parameter for polarity level, $\gamma$, to measure the likeliness of positive (negative) term to occur in a positive (negative) document compared to a negative (positive) document. Equation 4.10 computes the conditional probabilities of the unknown terms, $f_u$, given class labels, '$+$' and '$-$'.

$$\begin{aligned} P(f_u|+) &= \frac{n(1-1/\gamma)}{(p+n)(m-p-n)}, \\ \text{and } P(f_u|-) &= \frac{n(1-1/\gamma)}{(p+n)(m-p-n)} \end{aligned} \tag{4.10}$$

where $P(f_u|+)$ and $P(f_u|-)$ are the conditional probabilities of the unknown terms given class, $m$ is the number of terms in the dictionary, $p$ is the number of positive terms labeled by the labeler, and $n$ is the number of negative terms labeled by the labeler.

The main difference between our approach and Melville and Sindhwani [2009] is that in our approach, rationales are tied to the documents in which they appear as rationales, whereas in Melville and Sindhwani [2009], the feature labels are weighted globally, and all positive words are equally positive, and all negative words are equally negative. Our approach provides more granular (per instance-feature combination) regularization as described in Section 4.3.1. Hence, there is no parameter setting where our approach is equivalent to Melville and Sindhwani [2009], however, as we show in Section 4.4.2, empirically, our approach performs quite similar to Melville and Sindhwani [2009].

**4.4.1.3 Description of Das et al. [2013].** Das et al. [2013] proposed an approach for incor-

porating feature labels into the training of a locally-weighted logistic regression classifier [17]. In feature annotation, each feature (for example, the term) is labeled by the human. For example, for a binary sentiment classification task, the terms are labeled as positive or negative. Locally-weighted logistic regression fits one logistic function per test instance, where the objective function for the logistic regression model is modified so that the training instances that are closer to the test instance are given higher weights compared to the training instances that are farther away from the test instance. When computing similarity between the test instances and training instances, in addition to regular document similarity, Das et al. [2013] takes labeled features into account: when a test document shares labeled features with a training document, it computes similarity between the test document and the training document based on the labeled features and the label of the training instance.

Logistic regression maximizes the conditional log likelihood of data as:

$$l_w(\theta) = \sum_{i=1}^{N} \log(P_\theta(y^i|x^i)) \tag{4.11}$$

Locally-Weighted Logistic Regression (LWLR) fits a logistic function around a small neighborhood of test instance, $x^t$, where the training instances, $x^i$, that are closer to $x^t$ are given higher weights compared to the training instances that are farther away from $x^t$. LWLR maximizes the conditional log likelihood of data as:

$$l_w(\theta) = \sum_{i=1}^{m} w(x^t, x^i) \log(P_\theta(y^i|x^i)) \tag{4.12}$$

where, the weight $w(x^t, x^i)$ is a kernel function:

$$w(x^t, x^i) = \exp\left(-\frac{f(x^t, x^i)^2}{k^2}\right) \tag{4.13}$$

where $f(x^t, x^i)$ is a distance function and $k$ is the kernel width.

Das et al. [2013] used LWLR for its ability to weight training instances differently, rather than for its ability to learn a non-linear decision boundary. LWLR assigns higher weights to documents that are more similar to $x^t$, and lower weights to documents that are less similar to $x^t$. They used $\text{cosim}(x^t, x^i) = 1 - \cos(x^t, x^i)$ as the baseline distance function to measure similarity between documents. To incorporate feature labeling into LWLR, they changed the baseline distance function to include two components: (i) distance between documents $x^t$ and $x^i$ based on all the words present in $x^t$ and $x^i$, i.e. $\text{cosim}(x^t, x^i)$ and (ii) distance between documents $x^t$ and $x^i$ based on all the features that have been labeled by user.

The second component of the distance function is computed as the difference between contributions of class-relevant and class-irrelevant features in $x^t$, where $x^t$ is $l_2$-normalized tf-idf feature vector. Considering binary classification, $y \in \{+, -\}$, if the label of $x^i$ is '$+$', the class-relevant features in $x^t$ will be all features that have been labeled as '$+$', and the class-irrelevant features in $x^t$ will be all features that have been labeled as '$-$'. Similarly, if the label of $x^i$ is '$-$', the class-relevant features in $x^t$ will be all features that have been labeled as '$-$', and the class-irrelevant features in $x^t$ will be all features that have been labeled as '$+$'. Let $\mathcal{R}$ be a set of class-relevant features in $x^t$ and let $\mathcal{I}$ be a set of class-irrelevant features in $x^t$. Their modified distance function for incorporating feature labels into LWLR becomes:

$$f(x^t, x^i) = \text{cosim}(x^t, x^i) \left( \sum_{j \in \mathcal{R}} x_j^t - \sum_{j \in \mathcal{I}} x_j^t \right) \tag{4.14}$$

Since the above distance function can sometimes become negative, the weight $w(x^t, x^i)$ is computed as:

$$w(x^t, x^i) = \exp\left( -\frac{\max(0, f(x^t, x^i))^2}{k^2} \right) \tag{4.15}$$

For simplicity, in Equation 4.14, we present formulation of their approach for binary classification. We refer the reader to Das et al. [2013] for a general formulation of their approach for multi-class classification.

Next, we present the results to empirically compare our classifier-agnostic approach with the three classifier-specific approaches: Melville and Sindhwani [2009], Zaidan et al. [2007], and Das et al. [2013].

**4.4.2 Results.** In this section, we first describe the experimental settings used to compare our approach to three baselines, Zaidan et al. [2007], Melville and Sindhwani [2009], and Das et al. [2013], and then present the results for empirical comparison. Note that the results for our approach and the baselines depend on hyper-parameters used in the experiments, hence, in order to have a fair comparison between our approach and the baselines, we compared them under two settings. First, we compared them using the best possible hyper-parameter settings. We ran several experiments using a wide range of values for all hyper-parameters and report the best possible performance, measured as the highest area under the learning curve, for each method. This is essentially equivalent to tuning parameters using the test data itself. We performed this test to observe how different methods would behave at their best. Second, we compared them using hyper-parameters that were optimized at each iteration of learning using cross validation on the labeled set, $\mathcal{L}$ obtained including and up to that iteration of active learning. We also provide results for learning without rationales (Lw/oR) using best parameters and using hyper-parameters optimized using cross validation on training data.

We used the same four document classification datasets described in Section 4.3.2.1. Since the results in Section 4.3.2 showed that tf-idf representation gave better results than the binary representation, in this section, we present results using only the tf-idf representation of the datasets. We repeated each experiment 10 times, starting with a different bootstrap, and report average results on 10 different trials.

Our method using multinomial naïve Bayes classifier (LwR-MNB) needs to tune the following hyper-parameters: (i) the Dirichlet prior, $\alpha$, for the features (ii) weight for the rationale features, $r$, and (iii) weight for the other features, $o$. The method in Melville and Sindhwani [2009] needs to tune the following hyper-parameters: (i) smoothing parameter for the instance model, $\alpha$, (ii) polarity level for the feature model, ($\gamma$), and (iii) weights for combining the instance model and feature model ($\beta$ and $1 - \beta$ respectively).

Our method using support vector machines (LwR-SVM) needs to tune the following parameters: (i) regularization parameter, $C$, (ii) weight for the rationale features, $r$, and (iii) weight for the other features, $o$. Zaidan et al. [2007] approach needs to tune the following hyper-parameters: (i) regularization parameter, $C$, for the pseudoexamples, $x_{ij}$, (ii) regularization parameter, $C_{contrast}$, for the contrast examples, $v_{ij}$, and (iii) margin between the original and contrast examples, $\mu$.

Das et al. [2013] used locally-weighted logistic regression specifically to incorporate feature labels into learning. Our method to incorporate rationales is independent of the classifier, hence we compared our approach to Das et al. [2013] using both logistic regression and locally-weighted logistic regression to see whether the improvements provided by incorporating rationales stem from using locally-weighted logistic regression. Our method using locally-weighted logistic regression classifier (LwR-LWLR) needs to tune the following parameters: (i) regularization parameter, $C$, (ii) kernel width, $k$, (iii) weight for the rationale features, $r$, and (iv) weight for the other features, $o$. Our method using vanilla logistic regression classifier (LwR-LR) needs to tune the following parameters: (i) regularization parameter, $C$, (ii) weight for the rationale features, $r$, and (iii) weight for the other features, $o$. Das et al. [2013] approach needs to tune the following parameters: (i) regularization parameter, $C$ and (ii) kernel width, $k$.

For each instance, $x^t$, in the test data, LWLR builds a model around a small neighborhood of $x^t$, based on distances between the test instance and training instances, $x^i$. This

method requires learning a logistic function for each test instance, and is therefore computationally very expensive. In this study, we compare our approach to the baselines using best hyper-parameters, which requires repeating each experiment several times with all possible hyper-parameter combinations. Moreover, our cross validation experiments require tuning hyper-parameters at each step of learning. To reduce the running time of LWLR experiments, we reduced the test data by randomly subsampling 500 test instances. To further reduce the running time, we searched for one parameter at a time, fixing others; that is, we did not perform a joint search over all the hyper-parameters for LWLR experiments.

**4.4.2.1 Comparison to Baselines under Best Parameter Settings.** In this section, we present results comparing the best learning curves obtained using our approach and the baselines. We bootstrapped the initial model using $10$ instances chosen randomly, picking $5$ documents from each class. At each iteration of learning, we selected 10 documents randomly from the unlabeled pool, $\mathcal{U}$. We repeated the experiments using a wide range of hyper-parameters for our approach and the baselines and plotted the best learning curve for each method.

For our approach using multinomial naïve Bayes, we searched for $\alpha$ between $10^{-6}$ and $10^2$. For our approach using support vector machines, we searched for $C$ between $10^{-2}$ and $10^2$. For our approach using locally-weighted logistic regression, we searched for $C$ between $10^{-3}$ and $10^3$ and $k$ between $0.1$ and $1$. For our approach using multinomial naïve Bayes and support vector machines, we searched for weights $r$ and $o$ between $10^{-4}$ and $10^7$. For our approach using locally-weighted logistic regression, we searched for weights $r$ and $o$ between $10^{-3}$ and $10^3$. In Zaidan et al. [2007], for $C$ and $C_{contrast}$, we searched for values between $10^{-3}$ and $10^3$, and $\mu$ between $10^{-2}$ and $10^2$. In Melville and Sindhwani [2009], we searched for $\alpha$ between $10^{-6}$ and $10^2$, $\gamma$ between $1$ and $10^5$, and $\beta$ between $0$ and $1$. In Das et al. [2013], we searched for $C$ between $10^{-3}$ and $10^3$ and $k$ between $0.1$ and $1$.

Figure 4.5 presents the learning curves comparing LwR-SVM to Zaidan

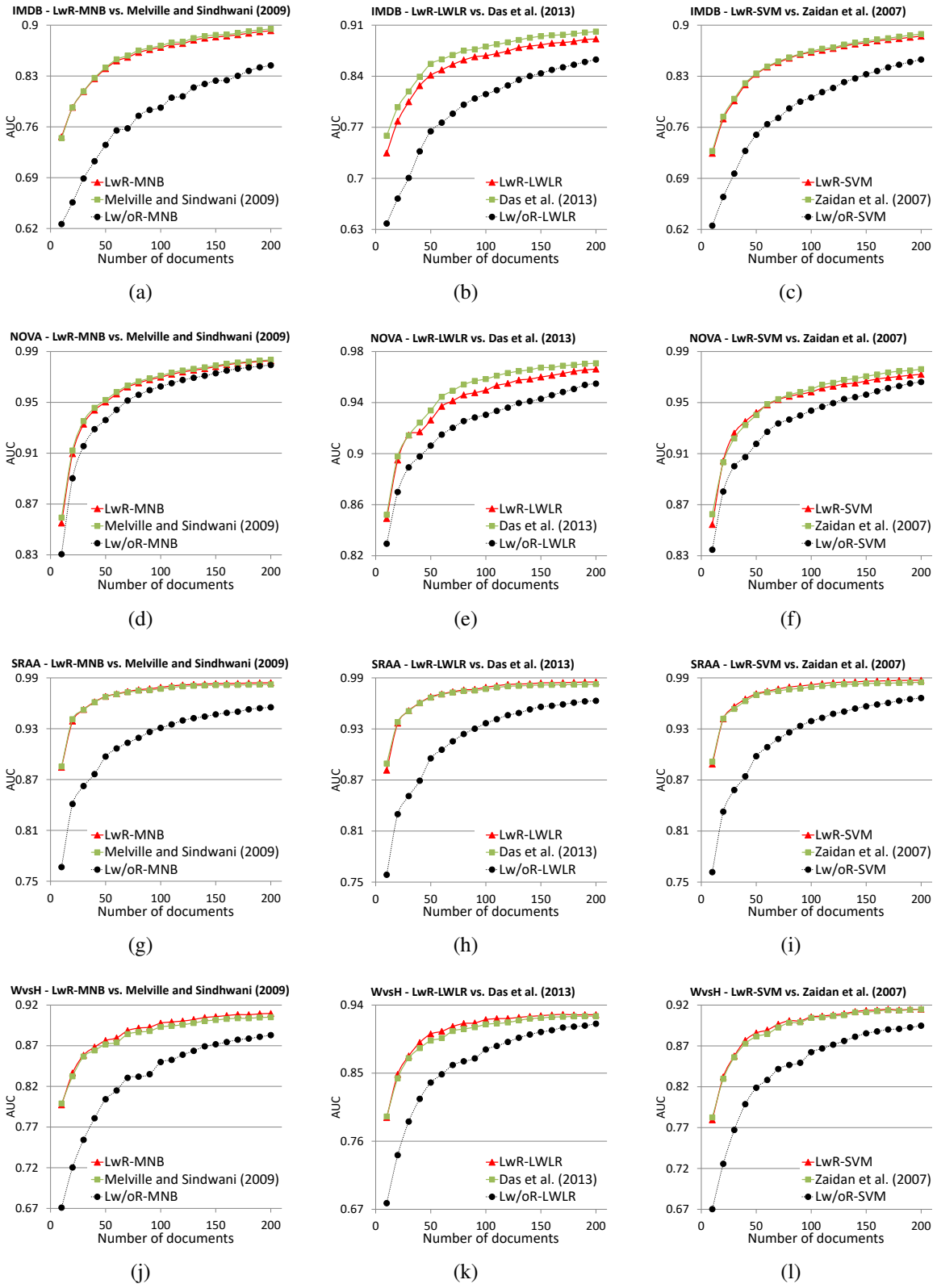



Figure 4.5. Results comparing our approach to the three baselines using best hyper-parameters. LwR-MNB performs similar to Melville and Sindhwani [2009] on all four datasets ((a), (d), (g), and (j)). LwR-LWLR performs similar to Das et al. [2013] on all four datasets ((b), (c), (h), and (k)). LwR-SVM performs similar to Zaidan et al. [2007] on all four datasets ((c), (f), (i), and (l)).

et al. [2007], LwR-MNB to Melville and Sindhwani [2009], and LwR-LWLR to Das et al. [2013]. These results show that under best parameter settings, our classifier-agnostic approach performs as good as other classifier-specific approaches. The results for our approach using logistic regression and locally-weighted logistic regression are very similar under best parameter settings, however, LWLR is computationally very expensive. We omit the learning curves for LwR-LR in Figure 4.5, as it is very similar to LwR-LWLR.

We report the hyper-parameter values that gave us the best possible learning curves (learning curves with the highest area under the AUC curve) for our approach and the baselines in Tables 4.5, 4.6, and 4.7. For our approaches, LwR-SVM, LwR-MNB, and LwR-LWLR, as expected, $r > o$ gave the best results. For Zaidan et al. [2007], we found that $\mu = 0.1$ and setting $C <= C_{contrast}$ gave the best results. Melville and Sindhwani [2009] used the weights for combining the instance model ($\beta$) and feature model ($1-\beta$) as $0.5$ and $0.5$ respectively. However, we found that for the four text datasets we used in this study, placing a much higher weight (e.g. $0.9$ or $0.99$) on the instance model gave better results than using their default weights for combining the two models. Note that if we place a weight of $1$ for the instance model (i.e. $\beta = 1$), the weight for the feature model will be zero, and this will give the same results as Lw/oR-MNB. Das et al. [2013] reported that setting $k = \sqrt{0.5}$ for LWLR-FL gave reasonably good macro-F1 scores, however, for the four text datasets, we found that $k > 0.4$ gave good results for AUC measure.

**4.4.2.2 Comparison to Baselines by Tuning Parameters using Cross Validation.** In this section, we present the results comparing our approach with the baselines under the setting where we search for optimal hyper-parameters using cross validation on labeled data, $\mathcal{L}$, at each iteration of learning. We performed $5$ fold cross validation on $\mathcal{L}$ and optimized all the hyper-parameters for the AUC measure, since AUC is the target performance measure in our experiments.

AUC of a classifier is equivalent to the probability that the classifier will rank a

Table 4.5. Hyper-parameter settings for Lw/oR-SVM, LwR-SVM, and Zaidan et al. [2007] that gave the best learning curves.

| Dataset | Lw/oR-SVM | LwR-SVM | | | Zaidan et al. [2007] | | |
|---|---|---|---|---|---|---|---|
| | $C$ | $C$ | $r$ | $o$ | $C$ | $C_{contrast}$ | $\mu$ |
| IMDB | 0.1 | 0.1 | 10 | 1 | 0.5 | 0.5 | 0.1 |
| NOVA | 10 | 0.1 | 10 | 1 | 1 | 1 | 0.1 |
| SRAA | 10 | 10 | 1 | 0.01 | 0.1 | 10 | 0.1 |
| WvsH | 0.1 | 10 | 1 | 0.1 | 0.2 | 0.2 | 0.1 |

Table 4.6. Hyper-parameter settings for Lw/oR-MNB, LwR-MNB, and Melville and Sindhwani [2009] that gave the best learning curves.

| Dataset | Lw/oR-MNB | LwR-MNB | | | Melville and Sindhwani [2009] | | |
|---|---|---|---|---|---|---|---|
| | $\alpha$ | $\alpha$ | $r$ | $o$ | $\alpha$ | $\gamma$ | $\beta$ |
| IMDB | 1 | 1 | 100 | 1 | 1 | $100,000$ | 0.99 |
| NOVA | 0.1 | 1 | 250 | 10 | 0.1 | $100,000$ | 0.9 |
| SRAA | 0.01 | 1 | 125 | 0.1 | 10 | $100,000$ | 0.99 |
| WvsH | 1 | 1 | 75 | 1 | 0.9 | $100,000$ | 0.9 |

randomly chosen positive instance higher than a randomly chosen negative instance. In an active learning setting, the labeled data ($\mathcal{L}$) is severely limited, consisting of only a few instances. When we use $5$ fold cross validation, each fold containing only $20\%$ instances are evaluated to produce an AUC score, which does not give an accurate measure of ranking. Hence, in order to fully utilize the scores assigned by the classifier for instances in all

Table 4.7. Hyper-parameter settings for Lw/oR-LWLR, LwR-LWLR, and Das et al. [2013] that gave the best learning curves.

| Dataset | Lw/oR-LWLR | | LwR-LWLR | | | | Das et al. [2013] | |
|---|---|---|---|---|---|---|---|---|
| | $C$ | $k$ | $C$ | $k$ | $r$ | $o$ | $C$ | $k$ |
| IMDB | 1 | 0.7 | 1 | 0.7 | 10 | 1 | 1000 | 1 |
| NOVA | 1000 | 1 | 1000 | 1 | 1 | 0.1 | 1000 | 1 |
| SRAA | 1000 | 1 | 1000 | 1 | 1 | 0.01 | 100 | 0.4 |
| WvsH | 10 | 0.5 | 10 | 0.5 | 1 | 0.1 | 1000 | 1 |

the folds, we merge-sorted the instances in all the folds using their assigned scores, and computed AUC score based on instances in all the folds. This is similar to the approach described in Fawcett [2006].

Figure 4.6 presents the learning curves comparing LwR-SVM to Zaidan et al. [2007], LwR-MNB to Melville and Sindhwani [2009], and LwR-LWLR to Das et al. [2013]. As these results show, when we optimize the hyper-parameters using cross validation on training data, LwR-SVM performs very similar to Zaidan et al. [2007], LwR-MNB performs very similar to Melville and Sindhwani [2009], and LwR-LWLR performs very similar to Das et al. [2013]. We performed t-tests comparing the learning curves obtained using our method and the baselines and found that the differences are not statistically significant in most cases.

The results for our approach using logistic regression (LwR-LR) and using locally-weighted logistic regression (LwR-LWLR) have some differences, when the hyper-parameters are optimized using cross validation on training set. For experiments using LWLR, we did not perform a grid search for the parameters, and optimized only one param-

eter at a time, which could result in sub-optimal hyper-parameters. Moreover, our approach using LWLR needs to tune four hyper-parameters ($C$, $k$, $r$, and $o$) and Das et al. [2013] needs to tune two hyper-parameters ($C$ and $k$).

These results show that our approach to incorporate rationales is as effective as three other approaches from the literature, Zaidan et al. [2007], Melville and Sindhwani [2009], and Das et al. [2013], that were designed specifically for incorporating rationales and feature annotations into support vector machines, multinomial naïve Bayes, and locally-weighted logistic regression respectively. Our approach has the additional benefit of being independent of the underlying classifier.

## 4.5 Active Learning with Rationales

So far we have seen that LwR provides drastic improvements over Lw/oR and our approach performs as well as other classifier-specific approaches in the literature. In previous sections, we made sure that both LwR and Lw/oR saw the same documents and we chose those documents randomly from the unlabeled set of documents. Active learning [94] aims to carefully choose instances for labeling to improve over random sampling. Many successful active learning approaches have been developed for annotating instances [57], [88], [97]. Ramirez-Loaiza et al. [2016] provide an empirical evaluation of common active learning strategies. Several approaches have been developed for annotating features [25], [31], and rotating between annotating instances and annotating features [4], [31], [66], [81]. In this section, we introduce an active learning strategy that is specifically tailored for the learning with rationales framework.

**4.5.1 Active Learning to Select Documents based on Rationales.** Arguably, one of the most successful active learning strategies for text categorization is uncertainty sampling, which was first introduced by Lewis and Catlett [1994] for probabilistic classifiers and later formalized for support vector machines by Tong and Koller [2001]. The idea is to label

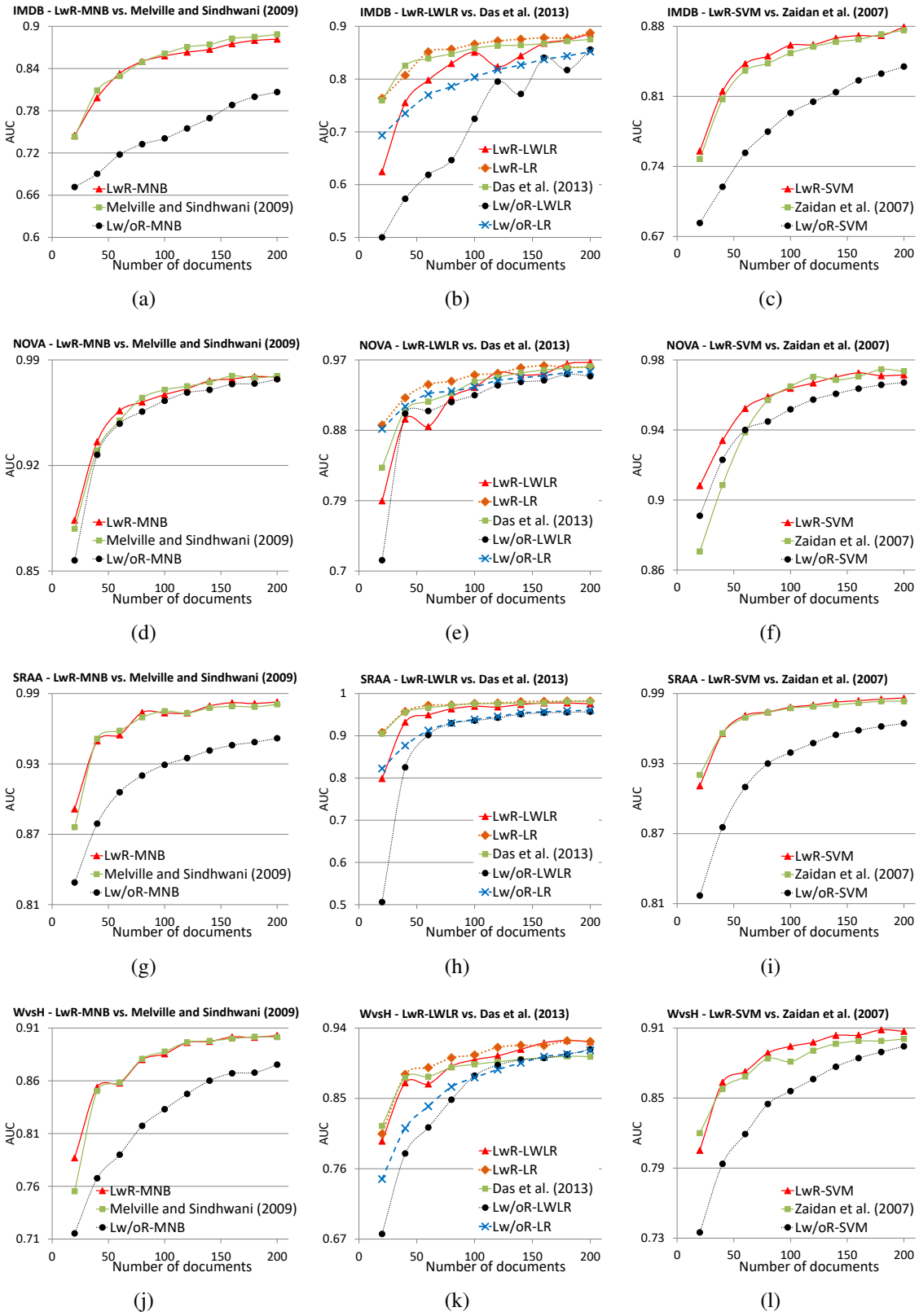



Figure 4.6. Results comparing our approach to the three baselines with hyper-parameters tuned using cross-validation on labeled data. LwR-MNB performs similar to Melville and Sindhwani [2009] on all four datasets ((a), (d), (g), and (j)). LwR-LWLR performs similar to Das et al. [2013] on all four datasets ((b), (c), (h), and (k)). LwR-SVM performs similar to Zaidan et al. [2007] on all four datasets ((c), (f), (i), and (l)).

instances for which the underlying classifier is uncertain, i.e., the instances that are close to the decision boundary of the model. It has been successfully applied to text classification tasks in numerous publications, including [92], [104], and [128].

We adapt uncertainty sampling for the learning with rationales framework. To put simply, when the underlying model is uncertain about an unlabeled document, we examine whether the unlabeled document contains words/phrases that were returned as rationales for any of the existing labeled documents. More formally, let $R^+$ denote the union of all the rationales returned for the positive documents so far. Similarly, let $R^-$ denote the union of all the rationales returned for the negative documents so far. An unlabeled document can be one of these three types:

- Category 1: has no words in common with $R^+$ and $R^-$.
- Category 2: has word(s) in common with either $R^+$ or $R^-$ but not both.
- Category 3: has at least one word in common with $R^+$ and at least one word in common with $R^-$.

One would imagine that annotating each of the Category 1, Category 2, and Category 3 documents has its own advantage. Annotating Category 1 documents has the potential to elicit new domain knowledge, i.e., terms that were not provided as a rationale for any of the existing labeled documents. It also carries the risk of containing little to no useful information for the classifier (e.g., a neutral review). For Category 2 documents, even though the document shares a word that was returned as a rationale for another document, the classifier is still uncertain about the document either because that word is not weighted high enough by the classifier and/or there are other words that pull the classification decision in the other direction, making the classifier uncertain. Category 3 documents contain conflicting words/phrases and are potentially harder cases, and annotating Category 3 documents has the potential to resolve conflicts for the classifier.

Building on our previous work on uncertainty sampling [99] in Chapter 3, we devised an active learning approach, where given uncertain documents, the active learner prefers documents of Category 3 over Categories 1 and 2. We call this strategy as *uncertain-prefer-conflict* (UNC-PC) because Category 3 documents carry conflicting words (with respect to rationales) whereas Category 1 and Category 2 documents do not. The difference between this approach and our work in Chapter 3 is that in Chapter 3, we selected uncertain instances based on model's perceived conflict whereas in this work, we are selecting documents based on conflict caused by the domain knowledge provided by the labeler. Next, we compare the vanilla uncertainty sampling (UNC) and UNC-PC strategies using LwR to see if using uncertain Category 3 documents could improve active learning.

**4.5.2 Active Learning with Rationales Experiments.** We used the same four text datasets and evaluated our method UNC-PC using multinomial naïve Bayes, logistic regression, and support vector machines. For the active learning strategies, we used a bootstrap of 10 random documents, and labeled five documents at each round of active learning. We used a budget of 200 documents for all methods. UNC simply picks the top five uncertain documents, whereas UNC-PC looks at top 20 uncertain documents and picks five uncertain documents giving preference to the conflicting cases (Category 3) over the non-conflicting cases (Category 1 and Category 2). We repeated each experiment 10 times starting with a different bootstrap at each trial and report the average results.

Figure 4.7 presents the learning curves comparing UNC-PC with UNC for multinomial naïve Bayes. Since the performances of both LwR and Lw/oR using tf-idf representation are better than the performance using binary representation, we compared UNC-PC to UNC for LwR using only the tf-idf representation. We see that for multinomial naïve Bayes, UNC-PC improves over traditional uncertainty sampling, UNC, on two datasets, and hurts performance on one dataset. The trends are similar for other classifiers and hence we omit them for simplicity.

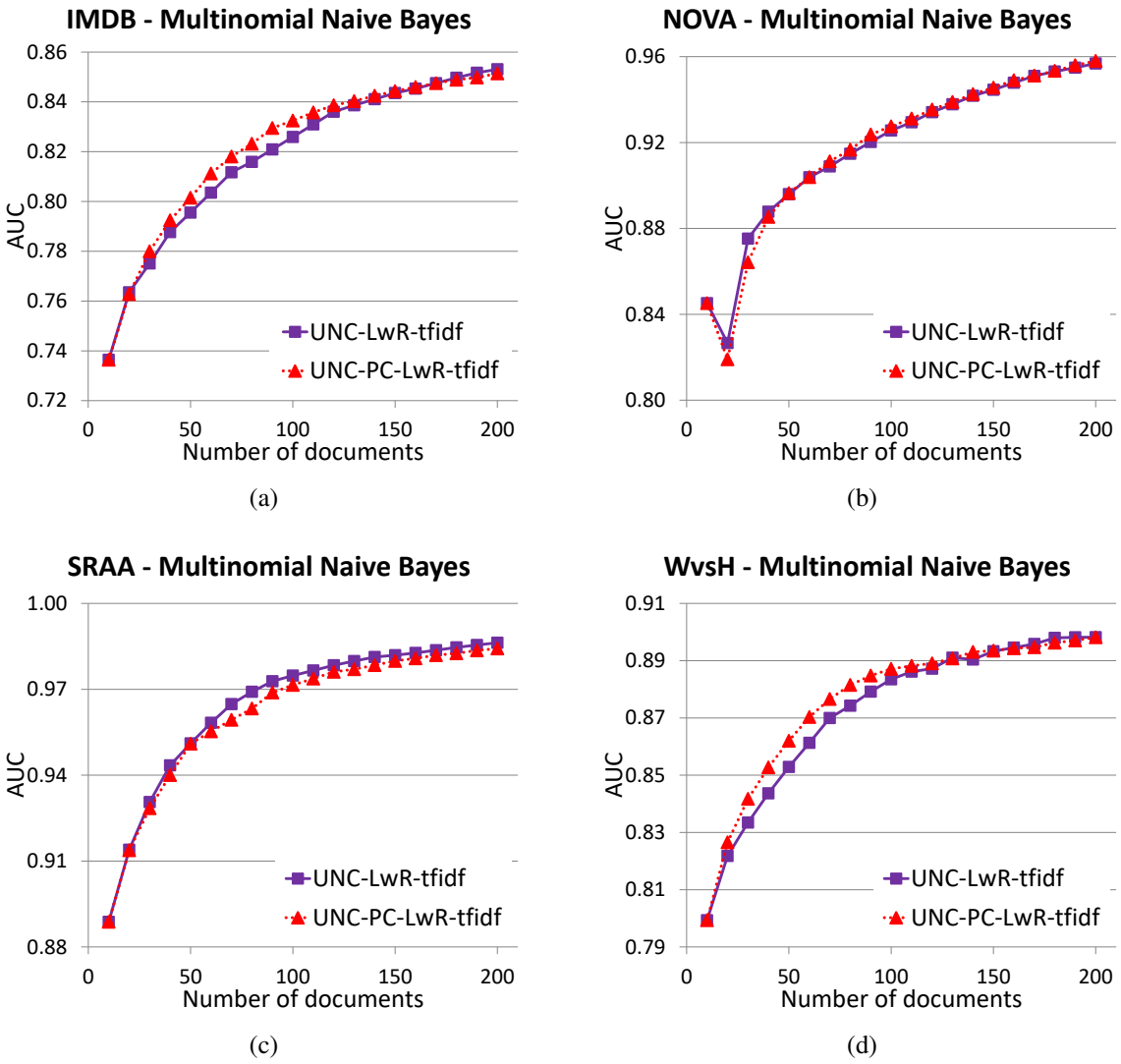



Figure 4.7. Comparison of LwR using UNC and UNC-PC for all datasets with tf-idf representation and using multinomial naïve Bayes classifier.

We performed paired t-tests to compare the learning curves of UNC-PC with the learning curves of UNC, to test whether the average of one learning curve is significantly better or worse than the average of the other learning curve). If UNC-PC has a higher average AUC than UNC with a t-test significance level of 0.05 or better, it is a Win, if it has significantly lower performance, it is a Loss, and if the difference is not statistically significant, the result is a Tie.

Table 4.8 shows the datasets for which UNC-PC wins, ties, or loses compared to

UNC. The t-test results show that UNC-PC wins on two out of four datasets for MNB and LR, and wins on three datasets for SVM. However, as these results and Figure 4.7 show, even though UNC-PC has potential, it is far from perfect, leaving room for improvement.

Table 4.8. T-test results for UNC-PC versus UNC. UNC-PC improves over UNC significantly for all three classifiers and most of the datasets.

| **UNC-PC versus UNC** | **MNB** | **LR** | **SVM** |
|---|---|---|---|
| Win | IMDB, WvsH | SRAA, NOVA | SRAA, NOVA, WvsH |
| Tie | NOVA | WvsH | - |
| Loss | SRAA | IMDB | IMDB |

## 4.6 Conclusion

We introduced a novel framework to enable richer interaction between the human and active learner by (i) asking the human experts to label the documents and provide rationales by highlighting phrases that convinced them to choose labels for documents, and (ii) presenting a simple approach to incorporate rationales into the training of any off-the-shelf classifier. The empirical evaluations on four text datasets with binary and tf-idf representations and three classifiers showed that our proposed framework utilizes rationales effectively. We evaluated our approach on user-provided rationales, which can be noisy, and showed that our framework can effectively incorporate user-provided rationales. We compared our classifier-agnostic approach to three classifier-specific approaches from the literature and showed that our method performs at least as well as the classifier-specific approaches. Additionally, we presented an active learning strategy that is tailored specifically for the learning with rationales framework and empirically showed that it improved over traditional active learning on at least two out of four datasets using multinomial naive Bayes, logistic regression, and support vector machines.

# CHAPTER 5

# RATIONALES FRAMEWORK FOR AVIATION DOMAIN

In this chapter, I describe how we enrich the interaction between a human expert and active learner in the aviation domain to identify a few anomalous flights that are of operational significance, e.g., represent a safety concern. In this case, I discuss a real-world application of active learning in the aviation domain. In Chapter 4, I presented the rationales framework for document classification, where labelers provided rationales by highlighting words in documents. In this chapter, we further allow the expert to provide more complex rationales, that are expressed in the form of conjunction of multiple features, for flights that are of operational significance. I present our rationales framework for aviation domain that can effectively incorporate the complex rationales into the training of support vector machines.

This chapter is based on our collaborative work with NASA Ames Research Center. My advisor, Dr. Mustafa Bilgic, and I collaborated with Nikunj Oza, Kamalika Das, Bryan Matthews, and David Nielsen. This work was published in the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery Knowledge Discovery, 2016 [102]. The material from the paper "Active Learning with Rationales for Identifying Operationally Significant Anomalies in Aviation", 2016, pp 209-225, Manali Sharma, Kamalika Das, Mustafa Bilgic, Bryan Matthews, David Nielsen, and Nikunj Oza, In Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2016, Riva del Garda, Italy, September 19-23, 2016, Proceedings, Part III, has been included in this chapter "With permission of Springer". The data used in this study is NASA's proprietary data, which is not publicly available. The data and domain expertise (flights data, flights labels, and rationales for classification of flights) were provided to us

by NASA Ames Research Center.

## 5.1 Introduction

As new technologies are developed to handle complexities of the Next Generation Air Transportation System (NextGen), it is increasingly important to address both current and future safety concerns along with the operational, environmental, and efficiency issues within the National Airspace System (NAS). NASA, in partnership with the Federal Aviation Administration (FAA) and industry is continuing to develop new technologies to identify previously undiscovered safety events through data mining of large heterogeneous aviation data sets that are collected on a regular basis. These techniques have the potential to discover new safety risks in the existing system or risks that did not exist previously but are a result of the implementation of the NextGen concepts. Combined with more traditional monitoring of safety, the Aviation Safety program at NASA has invested significant resources for development and use of data mining methods for identification of unknown safety and other events in Flight Operations Quality Assurance (FOQA) data [72].

Several unsupervised anomaly detection methods have been developed to identify anomalies in commercial flight-recorded data. In the absence of knowledge regarding the types of safety events that are present in the data, and absence of labels, unsupervised techniques are the only ones that have the unique ability to find previously unknown anomalies; however, they do so only in the statistical sense—the anomalies found are not always operationally significant (e.g., represent a safety concern). After an algorithm produces a list of statistical anomalies, a Subject Matter Expert (SME) must go through that list to identify those that are operationally relevant for further investigation. A very small fraction of statistical anomalies (less than 1%) turns out to be operationally relevant, so substantial time and effort is spent by SMEs in examining anomalies that are not of interest.

The goal of this work is to semi-automate the process of distinguishing between

operationally significant anomalies and uninteresting statistical anomalies through use of supervised learning approaches, which require labeled instances. We propose to use active learning for training a classifier, so that SME time and effort is spent on only the most informative and critical anomaly instances. In this process, first an unsupervised anomaly detection algorithm is run on all the flight data to generate a ranked list of statistically significant anomalies. A very small percentage of these are presented to SMEs to bootstrap the active learning process. The SME provides labels for each of these instances along with an explanation about the label. A positive label indicates an operationally significant safety event whereas a negative label indicates otherwise. Based on these few labels we build an active learning system that (i) utilizes the SME's time in the most effective manner by iteratively asking for labels for few informative instances, (ii) elicits rationales/explanations from the SME for why s/he assigns a certain label to an instance, and (iii) constructs new features, based on rationales, that are incorporated in future iterations of active learning and classifier training.

Active learning for anomaly detection has been studied in the past with the goal of finding *useful* anomalies as opposed to statistical anomalies [79] where a priori knowledge of the number of rare event classes is assumed. In our application the number of types of anomalies encountered is unknown and therefore, the assumption does not hold true. Recent work in active learning has focused on eliciting richer feedback from the experts in addition to labels, to speed up the annotation process. For example, experts are asked to annotate features as relevant/irrelevant for a specific task [4], [104]. Similarly, several researchers have investigated eliciting rationales, which often correspond to highlighting a piece of text in text classification or highlighting feature values in feature-valued representations, and incorporated them into the training of classifier [101], [124]. In this work, we build on the rationale framework by allowing the domain experts to provide rationales for their classification. The main difference between our work and existing work is that in this work we enrich the representation by creating additional features that are combinations of

existing features rather than focusing on feature value distribution.

The advantages of this method are twofold: (i) it dramatically minimizes the time an SME needs to spend to find operationally significant anomalies from the long list of statistical anomalies output by any unsupervised anomaly detection method, and (ii) at the end of training, we have a classifier that can be run on the original flight operations data set to uncover many more operationally significant safety events that might have been missed in the original anomaly detection process due to the presence of overwhelming number of statistically significant, but uninteresting, anomalies. Our experiments with real aviation data show that using active learning with rationales improves $precision@5$ (defined as number of positive instances in top 5 instances ranked according to their distance from the decision boundary) results by as much as 75% compared to the state-of-the-art.

The rest of the chapter is organized as follows. Section 5.2 discusses the data setup and the existing unsupervised anomaly detection framework. Section 5.3 discusses our proposed active learning algorithm and its performance is analyzed in Section 5.4. Section 5.5 discusses deployment plans. Section 5.6 concludes the chapter.

## 5.2 Background

In this section we describe the state-of-the-art unsupervised anomaly detection method used for identifying statistical anomalies in flight operations data, followed by description of the data used in this study.

**5.2.1 Multiple Kernel Anomaly Detection.** The unsupervised anomaly detection algorithm that is currently used in the aviation safety community most frequently is Multiple Kernel Anomaly Detection (MKAD)[7] [24]. The MKAD algorithm is designed to run on heterogeneous data sets consisting of multiple attribute types including discrete and continuous. MKAD is a "multiple kernel" [5] based approach where the major advantage is the

[7]http://ti.arc.nasa.gov/opensource/projects/mkad/

method's ability to combine information from multiple heterogeneous data sources. The heart of MKAD is a one-class SVM model that constructs an optimal hyperplane in the high dimensional feature space to separate the abnormal (or unseen) patterns from the normal (or frequently seen) ones. This is done by solving the following optimization problem [90]:

$$\begin{aligned} \min \quad & Q = \tfrac{1}{2}\textstyle\sum_{i,j} \alpha_i \alpha_j K\left(\mathbf{x}_i, \mathbf{x}_j\right) \\ \text{subject to} \quad & 0 \leq \alpha_i \leq \tfrac{1}{\ell\nu}, \textstyle\sum_i \alpha_i = 1, \rho \geq 0, \quad \nu \in [0,1] \end{aligned} \tag{5.1}$$

where $\alpha_i$'s are Lagrange multipliers, $\ell$ is the number of data tuples in the training set, $\nu$ is a user-specified parameter that defines the upper bound on the training error, and also the lower bound on the fraction of training points that are support vectors, $\rho$ is a bias term, and $K$ is the kernel matrix. Once this optimization problem is solved, at least $\nu\ell$ training points with non-zero Lagrangian multipliers $(\alpha)$ are obtained and the points for which $\{\mathbf{x}_i : i \in [\ell], \alpha_i > 0\}$ are called the support vectors. The decision function is:

$$f(\mathbf{z}) = sign\left(\sum_i \alpha_i \sum_p \eta_p K_p(\mathbf{x}_i, \mathbf{z}) - \rho\right)$$

which predicts positive or negative label for a given test vector $\mathbf{z}$. Instances with negative labels are categorized as outliers.

The classifier that we learn using active learning for differentiating between operationally significant and uninteresting anomalies is a two-class support vector machine using multiple kernels. Therefore, it differs from MKAD in the fact that it is not based on a one-class SVM like MKAD, but has the same kernel structure as MKAD. The dual objective function for the two-class problem is:

$$\max_{\alpha} \sum_{i=1}^{\ell} \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j)$$

where $(\mathbf{x}_i, y_i)$'s are the data tuples for $i = 1, \ldots, \ell$. Here $\mathbf{x}_i$ and $y_i$ are the input data points and class labels respectively. In the supervised classification case, the $\mathbf{x}_i$'s correspond to the anomalies found by the MKAD algorithm as discussed above and $y_i$'s correspond to the labels provided by the SMEs. For identifying operationally significant anomalies, this classifier is used to rank the test instances based on their distance from the hyperplane.

**5.2.2 Data Preparation.** The surveillance data used in this study comes from combining two Air Traffic Control (ATC) facilities — Denver Terminal Radar Approach Control (D01) and the Denver Air Route Traffic Control Center (ZDV). The objective of this work is to develop a process that automatically discovers previously unmonitored, operationally significant, flight trajectories representing a safety risk to the airspace. The end goal is to produce a tool that can rank these anomalous flights for controllers to review and help make mitigating decisions about the safety of the airspace. The types of anomalies that are being targeted in this study are unusual trajectories from 30 nautical miles (NM) on approach to landing. These can include strange vectoring that do not conform to standard operating procedures, significant overshooting of the final approach fix, or high altitude and speed profiles that can lead to unstable approaches. Figure 5.1 illustrates the data processing flow from data collection through merging, filtering, unsupervised anomaly detection, and SME feedback incorporation for classification of anomalies into operationally significant and uninteresting categories. Data collection refers to the process of recording the relevant data that is used in this study (done by the PDARS program responsible for collection, processing, and reporting of aviation data from multiple sources). NASA was given access to PDARS data for the 2014 and 2015 calendar years. Approximately 25,000 flights are available to us from 2014, of which approximately 2400 flights for a particular month are being analyzed as part of our safety study for Denver for 2014. The 2015 flights are only used for validation of results. For each trajectory, from 30 NM out from the destination airport, the minimum separation is found and used to create four-dimensional trajectories: latitude, longitude, altitude and distance to nearest flight. These four features are then

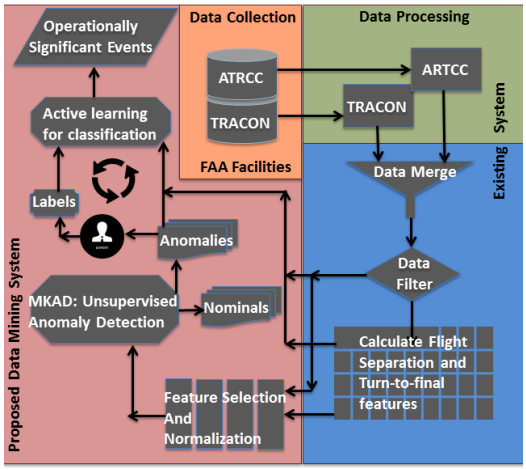



Figure 5.1. System setup: Data collection, processing, and mining.

averaged over half NM intervals from 30 NM to the runway threshold based on distance traveled and are partitioned by runway and destination airport sets on each day. This results in trajectories with fixed vector lengths because of the half-mile binning and the fixed 30 NM distance traveled, which are then used to create similarity kernels. We also use the PDARS turn-to-final (TTF) reports that provide specific characteristics of how the aircraft performed the turn on to the final approach within 20 NM of a runway. All deviations are calculated with respect to the intercept, which is the point at which the flight trajectory crosses the extended runway centerline before making its final approach. These deviations include intercept distance, angle of intercept, altitude deviation, distance deviation, and speed. Maximum overshoot and aircraft size (categorical feature indicating one of four weight categories) are two additional features from this source. In addition, three binary parameters are derived based on the characteristics of the flight identified as the nearest neighbor for each time step. These features are designed to provide domain context since flights on parallel runways or flights in the same flow are allowed to encroach within the standard separation threshold, whereas flights on the same runway should not fall below the separation threshold. These parameters indicate whether two nearest neighboring flights are on the same runway, parallel runway, or are part of the same flow. An additional derived feature called separation is constructed as the 3-d separation between two flights based on

the $l_2$ norm of the horizontal and vertical separation. It should be noted here that all of these (raw and derived) features together constitute the original feature set for our study. The data is heterogeneous in the sense that some of these features are time-series data while others are a single-point feature and some are continuous whereas others are discrete, nominal, or binary.

The data mining block in Figure 5.1 consists of the next steps of unsupervised anomaly detection followed by SME review and labeling, and finally, classifier learning for distinguishing between operationally significant anomalies and uninteresting anomalies. Depending on the size of the input data set, MKAD algorithm may discover hundreds to thousands of ranked anomalies, making it difficult for domain experts to validate all of them. Therefore, we use active learning to learn a classifier using very few labeled instances for this purpose. Each time an SME is provided an instance to be classified, the SME provides the label, along with an explanation/rationale for his/her decision. This rationale, whenever possible, is converted into a new additional feature, which is then incorporated into the classifier training through the creation of a new kernel. The details of this process and approach are described in the next section.

## 5.3 Active Learning with Rationales

Active learning algorithms iteratively select informative instances for labeling to save annotation time, cost, and effort [94]. For skewed data sets with minority class distribution much less than the majority class, a common and simple approach for selecting informative instances is to maximize the chances of retrieving positive instances [7]. Most-likely positive (MLP) strategy aims to add more positive instances into the labeled training set. The objective is:

$$\mathbf{x}^* = \underset{\mathbf{x}\in\mathcal{U}}{\operatorname{argmax}}\, P_\theta(\hat{\mathbf{y}}^+|\mathbf{x})$$

where $\hat{\mathbf{y}}^+$ represents the predicted positive label. I provided a detailed description of MLP

strategy in Section 2.2.3.

In learning with rationales approach [101], [123], SMEs provide rationales in the form of features that they think are responsible for classifying an instance into a particular class. In this chapter, we elicit the rationales from SMEs and incorporate them into the learning process. The main difference between previous work on incorporating rationales in Chapter 4 and our work here is that we create new features based on the rationales provided by the SMEs.

For training our classifier using active learning, we work with the list of anomalies produced by running the unsupervised anomaly detection algorithm, MKAD, on the data described in Section 5.2.2. For each flight, MKAD returns an anomaly score, which is the flight's distance from the hyperplane of a one-class SVM model. Flights with a negative score are considered as anomalous and flights with a positive score are considered as not anomalous. The SMEs are asked to provide labels for top 5% anomalous flights based on whether they think the anomaly is operationally significant (OS/positive labels) or not (NOS/negative labels). They are also asked to provide a rationale for the chosen label. Since labels and rationales are subjective opinions of each SME, we consolidate the labels and rationales from two SMEs by resolving conflicts (by reviewing each others' labels and rationales) whenever there is one, to get gold standard labels and rationales for our study.

**5.3.1 Creating Rationales.** When the SMEs identify a flight as an OS flight, they provide rationales in the form of either domain knowledge or using existing features and thresholds. However, when the SMEs identify a flight as NOS, they only provide acknowledgment of certain characteristics of the flight (e.g., a little overshoot, speed not a factor, small deviations on final). In anomaly detection tasks, it is easy to provide a rationale for why a particular instance is anomalous, but it is often difficult, if not impossible, to provide a rationale for why an instance is not anomalous. Therefore, we use the rationales for only the OS flights to create new features and use them to extend the feature representation.

Note that the rationales provided by SMEs are often in terms of the original features that are already captured by PDARS. Some rationales talk about two or more features whereas some highlight only one feature.

In our training set, most OS anomalies could be explained by one or more of three different rationales. The first rationale provided for operational significance is loss-of-separation, which the domain experts define as 'horizontal separation is less than 3 miles and vertical separation is less than a 1000 feet, and the nearest neighboring flight is not on parallel runways and not part of the same flow'. When a loss-of-separation rationale is provided, we create a new feature that checks whether the criteria 'horizontal separation less than 3 miles and vertical separation less than 1000 feet' and 'the nearest neighboring flight is not on parallel runway and not in the same flow' hold and incorporate it as a new binary feature in our training set.

The second rationale provided by the SMEs is for large overshoots where an overshoot is defined as going past a certain point in the landing trajectory against standard operational procedures. For rationales such as 'maximum overshoot is too large', we create a new feature that checks whether the overshoot is greater than a threshold. The threshold can be either chosen manually based on domain knowledge or based on the values of the overshoot feature for the labeled OS flights with overshoot rationale observed until that point, and updated iteratively.

The third rationale provided by the SMEs is for unusual flight path. Since this rationale is more qualitative than quantitative, and none of the original features represent an 'unusual flight path', we compute a new feature as follows. For each runway, using latitude and longitude features, we compute expected flight trajectory as the average trajectory of all flights that land on a runway. Then we create a new feature that captures the overall deviation of each flight from its expected flight trajectory over the last 10 points in the trajectory. Figure 5.2 shows the plots for a few trajectories. It can be seen that for the first

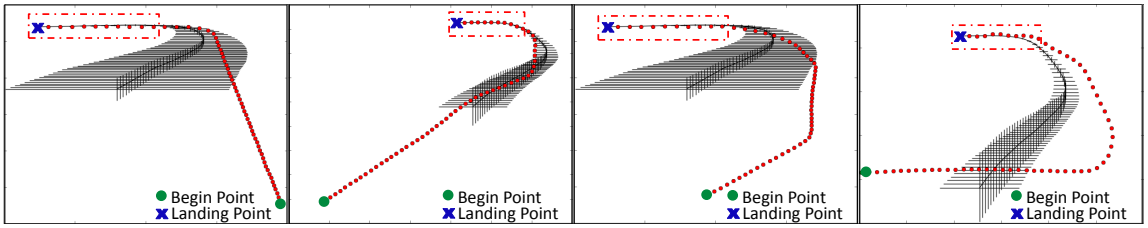



Figure 5.2. Expected flight path and deviation from it for 4 flights. The first three flights are NOS. The last flight is an OS flight.

three flights in Figure 5.2, the red dots align well with the expected trajectory (highlighted using the red box), whereas for the last flight there is significant deviation from the expected trajectory. This can have severe safety implications and is therefore considered an operationally significant safety event.

**5.3.2 Active Learning with Rationales Algorithm.** Algorithm 3 describes our approach for incorporating rationales into active learning. Active learning algorithm starts with a small set of labeled flights, $\mathcal{L}$, and finds the most informative flight, $\mathbf{x}^\star$, from the unlabeled set, $\mathcal{U}$. The most informative flight is the one that provides the classifier maximum information in terms of the decision boundary, or, in other words, one that has the maximum $utility$. The flight $\mathbf{x}^\star$ is then presented to the SME, who provides its label $\mathbf{y}^\star$. For every flight we present to the SME, in addition to a label, we also request for a rationale $R(\mathbf{x}^\star)$ describing why s/he labeled the flight as OS or NOS. If the label is OS, we create a new feature, $f_r^\star$, if possible, for the rationale $R(\mathbf{x}^\star)$ and add it into our existing feature representation: $\mathbf{f} = \langle f_1, f_2, \cdots f_n \rangle \bigcup \langle f_r \rangle$. We assign weight $w_o$ for the original feature kernels and weight $w_r$ for the rationale feature kernels, where $w_r \geq w_o$, since intuitively the rationale features are the ones that have the highest power to separate the OS flights from the NOS ones. However, to satisfy Mercer's condition, we need to ensure that it is a convex combination of the kernels. Therefore, we normalize each weight by the sum of the weights $w = w_o \times n + w_r \times p$, where $n$ and $p$ denote the number of original and rationale features respectively. Let $\eta$ denote the normalized kernel weights for the enhanced feature set. Note that the kernel weights for original features $\langle \eta_1, \eta_2, \cdots \eta_n \rangle$ are uniform and hence the kernel

weight for each original feature will be $\eta_o$, which is computed in Step 10 of Algorithm 3. Similarly, the kernel weight for the rationale feature set $\langle \eta_{n+1}, \eta_{n+2}, \cdots \eta_{n+p} \rangle$ is $\eta_r$ and is computed in Step 11 of Algorithm 3. The final kernel is computed using the updated set of kernel weights $\eta$ containing normalized weights $\eta_o$ for the original feature kernels and the normalized weights $\eta_r$ for the rationale feature kernels for the enhanced feature set $\mathbf{f}$.

**Algorithm 3** Active Learning with Rationales for Identifying Operationally Significant Anomalies in Aviation

1: **Input:** $\mathcal{U}$ - unlabeled flights, $\mathcal{L}$ - labeled flights, $\mathcal{T}$ - test flights, $\mathbf{f} = \langle f_1, f_2, \cdots f_n \rangle$ - current set of features, $\eta = \langle \eta_1, \eta_2, \cdots \eta_n, \eta_{n+1}, \eta_{n+2}, \cdots \eta_{n+p} \rangle$ - normalized kernel weights for enhanced feature set, $\theta$ - underlying classification model, $B$ - budget
2: **repeat**
3: $\mathbf{x}^\star = \underset{\mathbf{x}^i \in \mathcal{U}}{\operatorname{argmax}}\, utility(\mathbf{x}^i|\theta)$
4: request label $\mathbf{y}^\star$ for the flight $\mathbf{x}^\star$
5: **if** $\mathbf{y}^\star$ == OS **then**
6: request SME to provide a rationale $R(\mathbf{x}^\star)$ for why the flight is operationally significant
7: **if** rationale $\neq \phi$ **then**
8: create feature $f_r^\star$ for $R(\mathbf{x}^\star)$
9: add $f_r^\star$ to $\mathcal{U}, \mathcal{L}$, and $\mathcal{T}$
10: $\eta_o = \frac{w_o}{\sum_{i=1}^{n} \eta_o + \sum_{j=1}^{p} \eta_r}$
11: $\eta_r = \frac{w_r}{\sum_{i=1}^{n} \eta_o + \sum_{j=1}^{p} \eta_r}$
12: $\eta = \langle \eta_1, \eta_2, \cdots \eta_n \rangle \bigcup \langle \eta_r \rangle$
13: $\mathbf{f} = \langle f_1, f_2, \cdots f_n \rangle \bigcup \langle f_r \rangle$
14: **end if**
15: **end if**
16: $\mathcal{L} \leftarrow \mathcal{L} \cup \{\langle \mathbf{x}^\star, \mathbf{y}^\star, R(\mathbf{x}^\star) \rangle\}$
17: $\mathcal{U} \leftarrow \mathcal{U} \setminus \{\langle \mathbf{x}^\star \rangle\}$
18: Train $\theta$ on $\mathcal{L}$
19: **until** Budget $B$ is exhausted; e.g., $|\mathcal{L}| = B$

**5.3.3 Possible Enhancements.** Based on the training data and the rationales provided by the SMEs, in this chapter, we created three features that encompass a significant number of OS safety scenarios. However, this set is far from complete as there can be a huge variety of other explanations that can come from SMEs. So the set of rationale features is always expanding. As the set of features grows based on rationales, there might be a need to consolidate features into conjunctions and disjunctions depending on redundancy.

For example, two common rationales in our study are loss-of-separation and large overshoot. However, not all OS flights have both reasons for being labeled OS. Some flights are OS because of loss-of-separation, but they might have perfectly acceptable overshoot values, whereas other OS flights might not have a loss-of-separation but might have large overshoot values. Current framework creates one feature per rationale. An alternative approach is to create one indicator feature and keep revising it by adding the new rationales as disjunctions. Also, once a classifier is trained using this framework, our goal is to find operationally significant events in the original flight data. However, since the classifier is trained on only the anomalies, the feature distribution does not necessarily match that of the overall data set. This unaccounted bias can be handled by sub-sampling some of the flights that are not signaled by MKAD and adding them to the training with NOS (negative) labels. Selecting flights that are ranked lowest by MKAD, for this purpose, can ensure with a high probability that the flights which are most certainly nominal are being used as NOS samples.

## 5.4 Empirical Evaluation

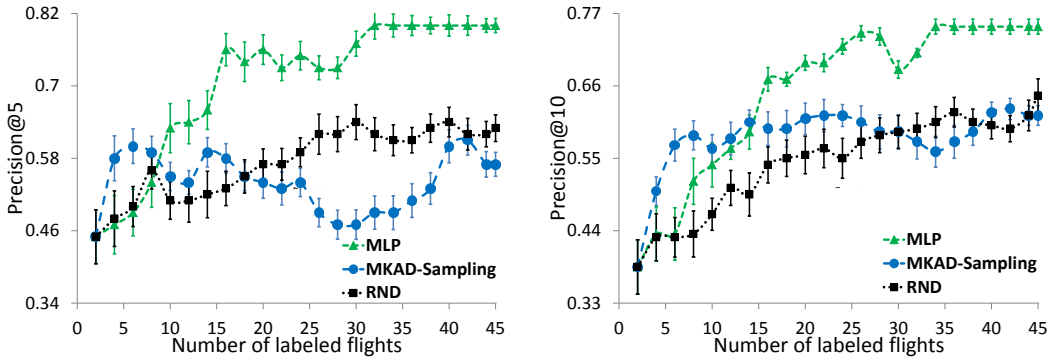
**5.4.1 Experimental Setup.** The data set used for training the classifier using active learning corresponds to PDARS data from the Denver Airport for August 2014, containing approximately 2400 flights out of which 153 flights are marked anomalous by MKAD. These 153 flights are reviewed by two SMEs independently (with conflict resolutions as needed) to provide labels and explanations. In these 153 flights, 26 are marked OS (positive) and the remaining 127 are marked NOS. The original data set contains 16 features as described in in Section 5.2.2. Additionally, we construct 3 rationale features supporting the explanations for the OS flights during the active learning iterations, when OS flights with one or more rationales provided in Section 5.3.1 are encountered.

Our proposed active learning strategy, MLP-W/RATIONALES, selects most-likely positive (MLP) instances for labeling at each iteration of training and creates (or updates)

rationale features whenever an appropriate new instance is encountered. We compare our algorithm's performance with three baselines: (i) random strategy (RND) where random instances are picked from the unlabeled pool and given to the SME for labeling, (ii) most-likely positive strategy (MLP) that selects more of the positive instances for labeling at each iteration, but does not add new features (or rationales), and (iii) MKAD-SAMPLING strategy where flights are given to the SME for labeling in the order of their MKAD anomaly ranking (higher the anomaly rank, the more informative it is for labeling).

We evaluate all strategies using $precision@k$ measure which can be defined as the number of positive instances in top $k$ instances ranked by the classifier. This measure is most suitable for our application because the SMEs go through a list of anomalies to identify those that are operationally significant for further investigation, and improving $precision@k$ means that the SMEs would analyze more of the OS flights compared to the NOS flights. We chose $precision@5$ and $precision@10$ for evaluation since they are the most frequently used in the literature measures to use [6], [121]. We bootstrap the classifier using an initially labeled set containing one OS flight and one NOS flight, and at each round of active learning the learner picks a new flight for labeling. We evaluate all strategies using 2-fold cross validation and repeat each experiment 10 times per fold starting with a different bootstrap, and present average results over 20 different runs. We set the budget ($B$) in our experiments to 45 flights, as most learning curves flatten out after about 35 flights. Since each learning curve is an average over 20 runs, for each learning curve, we report error bars for standard error of the mean (SEM), which is computed as standard deviation divided by the square root of sample size ($SEM = \frac{s}{\sqrt{n}}$).

**5.4.2 Results.** Figure 5.3 presents the learning curves comparing RND, MKAD-SAMPLING, and MLP strategies for $precision@5$ and $precision@10$. MKAD-SAMPLING performs worse than RND for $precision@5$ and it outperforms RND for $precision@10$. However, MLP outperforms both RND and MKAD-SAMPLING for $precision@5$ and




(a) (b)

Figure 5.3. MLP vs. RND and MKAD-SAMPLING. MLP significantly outperforms RND and MKAD-SAMPLING for both (a) $precision@5$ and (b) $precision@10$.

$precision@10$. We performed pairwise one-tailed t-tests under significance level of $0.05$, where pairs are area under the learning curves for 20 runs of each method. If a method has higher average performance than a baseline with a significance level of $0.05$ or better, it is a win, if it has significantly lower performance, it is a loss, and if the difference is not statistically significant, the result is a tie. The t-test results show that MKAD-SAMPLING statistically significantly loses to RND for $precision@5$ and significantly wins over RND for $precision@10$. MKAD-SAMPLING performs better than MLP at the very beginning of the learning curves, but t-test results show that overall, MLP statistically significantly wins over MKAD-SAMPLING for both $precision@5$ and $precision@10$. This justifies our choice of using MLP as the active learning strategy for training our classifier for a highly skewed distribution of class labels.

Tables 5.1 and 5.2 present comparison of the number of labeled flights required by these methods to achieve a target value of $precision@5$ and $precision@10$. The maximum target for each metric is chosen based on the best performance observed in the learning curves for each of the strategies. The results show that MLP often requires fewer labeled flights compared to RND and MKAD-SAMPLING. Moreover, MLP achieves a $precision@5$ of $0.7$ and $precision@10$ of $0.65$ with just 16 labeled flights, whereas RND and MKAD-

SAMPLING could not achieve these targets even with 45 labeled flights.

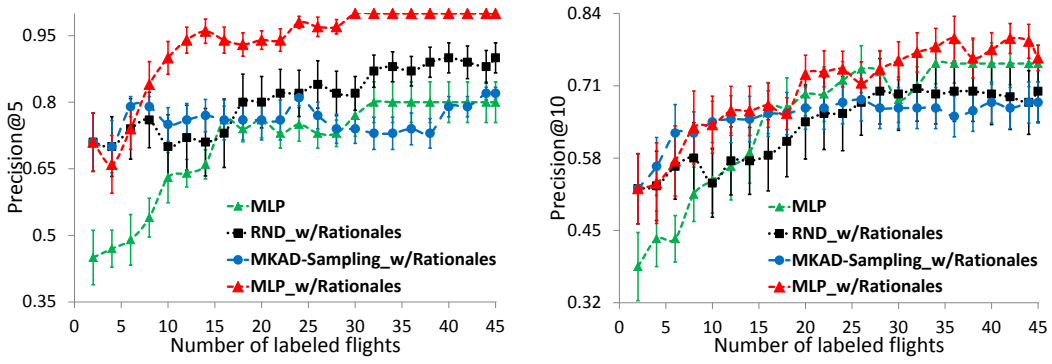
Next, we present the results that demonstrate the effect of incorporating rationales into active learning. Figure 5.4 presents the learning curves comparing MLP strategy for active learning without rationales (MLP) and MLP with rationales strategy (MLP-w/RATIONALES) that utilizes MLP to select instances and incorporates rationales iteratively during active learning (refer to Algorithm 3). We set the rationale feature weight $w_r = 100$ and the original feature weight, $w_o = 1$. The results show that MLP-w/RATIONALES statistically significantly wins over MLP for both $precision@5$ and $precision@10$ performance measures. Moreover, MLP-w/RATIONALES requires even fewer labeled flights compared to MLP to achieve the same target performance measure, as shown in Tables 5.1 and 5.2. For example, MLP achieves a target $precision@5$ of $0.8$ with 32 labeled flights, whereas

Table 5.1. Comparison of number of labeled flights required by various strategies to achieve a target $precision@5$. 'n/a' represents that the target performance cannot be achieved by a method even with 45 labeled flights.

| **Method** | **Target** $precision@5$ | | | | | |
|---|---|---|---|---|---|---|
| | **0.5** | **0.6** | **0.7** | **0.8** | **0.9** | **1.0** |
| RND | 6 | 25 | n/a | n/a | n/a | n/a |
| MKAD-SAMPLING | 4 | 6 | n/a | n/a | n/a | n/a |
| MLP | 5 | 10 | 16 | 32 | n/a | n/a |
| MLP-w/RATIONALES | 2 | 2 | 2 | 8 | 10 | 29 |

Table 5.2. Comparison of number of labeled flights required by various strategies to achieve a target $precision@10$. 'n/a' represents that the target performance cannot be achieved by a method even with 45 labeled flights.

| **Method** | **Target** $precision@10$ | | | | | |
|---|---|---|---|---|---|---|
| | **0.50** | **0.55** | **0.60** | **0.65** | **0.70** | **0.75** |
| RND | 12 | 18 | 33 | n/a | n/a | n/a |
| MKAD-SAMPLING | 4 | 6 | 13 | n/a | n/a | n/a |
| MLP | 8 | 12 | 15 | 16 | 23 | 34 |
| MLP-w/RATIONALES | 2 | 5 | 7 | 11 | 19 | 29 |




(a) (b)

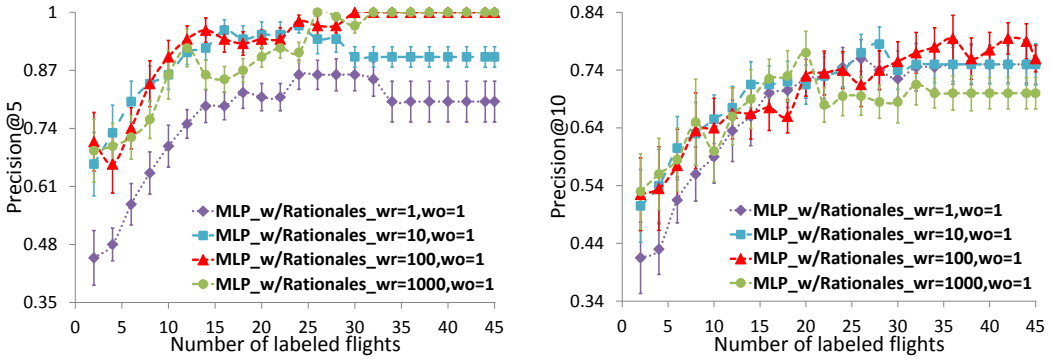
Figure 5.4. MLP-W/RATIONALES VS. MLP. Incorporating rationales further improves performance over MLP for both (a) $precision@5$ and (b) $precision@10$.

MLP-W/RATIONALES achieves this target with only 8 labeled flights, which is 75% savings in the labeling effort over MLP.

Figure 5.4 also compares MLP-W/RATIONALES to RND-W/RATIONALES and MKAD-SAMPLING-W/RATIONALES. MKAD-SAMPLING-W/RATIONALES performs better than MLP-W/RATIONALES at the beginning for both $precision@5$ and $precision@10$, but after seeing approximately 10 labeled instances, MLP-W/RATIONALES outperforms MKAD-SAMPLING-W/RATIONALES. T-tests show that MLP-W/RATIONALES statistically significantly outperforms both MKAD-SAMPLING-W/RATIONALES and RND-W/RATIONALES for both $precision@5$ and $precision@10$.

**5.4.2.1 Choice of Rationale Weights.** We ran experiments to study the effect of weights $w_r$ and $w_o$ on the performance of our algorithm. We chose uniform weighting for the original feature kernels since all 16 of those were suggested by domain experts and were supposed to be important for this safety study. We fixed $w_o$=1 and experimented with four weight settings for $w_r$ (1, 10, 100, or 1000). Figure 5.5 presents the learning curves for these four weight settings for MLP-W/RATIONALES. The results confirm our intuition that weighting rationale features higher than original features provides benefit to the active learner. The $precision@5$ results are significantly better with $w_r$=100 than other weights

for $w_r$. For $precision@10$, setting higher weights for rationale features improves performance at the beginning of active learning, however, t-test results show that weights $w_r$=1, 10, and 100 statistically significantly tie with each other. In general, weighting rationale features higher than original features improves learning. The kernel weights for optimal performance can be obtained through multiple kernel learning.




(a) (b)

Figure 5.5. Comparison of rationale features weights $w_r$ for MLP-W/RATIONALES using (a) $precision@5$ and (b) $precision@10$

Ideally, one would want to search for the best weights setting using cross validation, but given the limited number of anomalous instances that domain experts could review, it was not possible for us to perform cross validation over the training set. Based on the performance observed for these four weight settings, we chose $w_o$=1 and $w_r$=100 for all our experiments.

**5.4.2.2 Scalability.** Active learning methods are typically computationally expensive, since they need to build a classifier at each iteration of learning and evaluate the utility score for every instance in the unlabeled pool. However, in our setting, when active learning is used on the output of an unsupervised anomaly detection algorithm, the unlabeled pool is much smaller in size compared to the entire set of raw instances. Therefore, utilizing this framework in a practical setting is easily viable, without the iterative nature of active

learning being a performance bottleneck.

**5.4.3 Performance Benefits.** In the absence of active learning framework, our SMEs took approximately 33 hours to review the entire set of 153 anomalies produced by MKAD. These 33 hours were spread over multiple weeks due to limited availability of SME time for such tasks, which is a standard problem in the industry. As Figure 5.4 shows, most of the learning curves flatten out after labeling 35 flights. This would reduce the SME review time to less than one-third of the original time. This has implications on both man-hours and monetary savings. Moreover, active learning with state-of-the-art (MKAD-SAMPLING) achieves $precision@5$ of 0.57 and $precision@10$ of 0.61. Active learning with rationales (MLP-W/RATIONALES) achieves $precision@5$ of 1 ($75.4\%$ improvement over MKAD-SAMPLING) and $precision@10$ of 0.76 ($24.6\%$ improvement over MKAD-SAMPLING).

**5.4.4 Validation Set Results.** Currently, MKAD is being used as an unsupervised anomaly detection method to find statistically significant anomalies in the data. We compare performance benefits that active learning with rationales framework (MLP-W/RATIONALES) provides over the MKAD-based classifier for finding OS anomalies in two external validation data sets, July 2014 and July 2015 data sets for the Denver airport. The July 2014 data set has 149 labeled flights with 24 OS anomalies and July 2015 data set has 257 labeled flights with 84 OS anomalies, as determined by the SMEs. Both $precision@5$ and $precision@10$ values for MKAD are 0.4 for the July 2014 data set, and 0.2 for the July 2015 data set. Using our (MLP-W/RATIONALES) framework, $precision@5$ improves by 15% for July 2014 data set and by 50% for July 2015 data set. On the other hand, $precision@10$ improves by 25% and 110% for the July 2014 and July 2015 data sets, respectively.

It should be noted that MKAD performs very poorly for the July 2015 data set. This is because the data set is expected to evolve significantly over the years (due to change in landing procedures and other regulation changes) and the MKAD classifier does not cap-

ture the signatures of the OS flights, but rather focuses on finding statistically different data points which can vary over time due to a change in the underlying distribution. However, the nature of the operationally significant anomalies still remains consistent and therefore MLP-W/RATIONALES can identify those types of anomalies much better than MKAD. These results show how active learning with rationales framework can help in building a classifier that is robust to changing distribution of statistically significant anomalies and can, therefore, be used on new data sets without further labeling needs.

## 5.5 Towards Deployment

The active learning framework improves over traditional learning, and incorporating rationales further improves learning, utilizing the SME's time much more efficiently. The classifier that is trained through this framework is focused on finding operationally significant anomalies, rather than simply statistically significant anomalies, and hence the flights that are signaled by the two-class classifier approach are of higher relevance to FAA.

This active learning framework has been developed as an extension to the anomaly detection framework that is currently used for detecting safety events. We expect this framework to easily fit into the existing anomaly detection framework because the classifier training is part of the same data flow pipeline that can take the output of MKAD as input and can seamlessly plug-in new data sources as needed. Given that the new classifier reduces SME review time significantly while improving coverage and reducing false alarm rate, it seems to be the perfect addition to bolster the existing anomaly detection framework, especially since these safety studies are conducted on a regular basis on data that gets collected every month. We expect that this enhanced data processing pipeline with the active learning framework incorporated into it will make the review and detection system significantly more efficient. In our current setup, we provide our SMEs an excel sheet containing the list of anomalies returned by MKAD and the SMEs note down the annotations and rationales textually. This process is repeated iteratively for each round of labeling. The

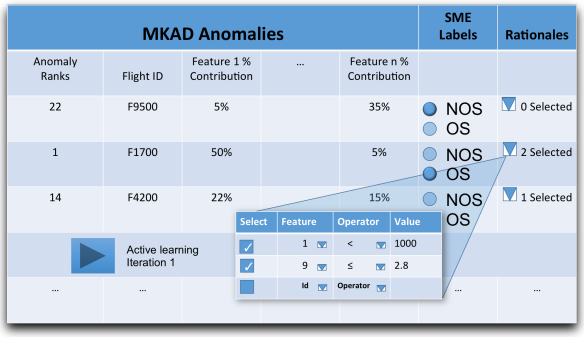



Figure 5.6. Diagrammatic representation of the GUI for deployment of active learning as part of the anomaly detection framework

textual information is then converted into features in batches. The next step towards the deployment of our active learning with rationales framework is to fully automate this process where the SMEs can select appropriate rationales using a drop-down list of features by choosing the criteria that were satisfied or violated by the flight in question. The SMEs can choose multiple features for each flight and, therefore, create complex rationale conditions that can be used to create new complex discriminative features on the fly and those features can be immediately utilized for the next iteration of active learning. Figure 5.6 shows a diagrammatic representation of the software that we are currently developing for deploying as part of the existing framework. It shows the SME initial bootstrap instances for labeling by randomly selecting from the list of anomalies found by MKAD, along with the feature contributions and asks for labels and rationales using drop-down menus. As soon as the classifier has enough number of bootstrap samples, training begins for the classifier. After every iteration the most informative instance is populated in the table for the SME to label and rationalize and classifier training begins again. This iterative process is repeated until the budget $B$ is exhausted or there is no further improvement in the classifier performance on a held-out set.

## 5.6 Conclusion

We presented the rationales framework for the aviation domain to incorporate rich

feedback, in the form of rationales for operationally significant flights, into the training of a classifier that can identify a few operationally significant anomalies from the uninteresting anomalies. Our proposed framework is novel in the sense that it incorporates SME feedback into the learning process by constructing new features to support the labels. Experimental evaluation on real aviation data shows that our approach improves detection of operationally significant events by as much as 75% compared to the state-of-the-art. The learnt classifier also generalizes well when tested on additional validation data sets. We also observe that our approach provides significant reduction in SME review time and labeling effort in order to achieve the same target performance using other baselines.

We are working toward deploying our framework as a daily reporting system that can reveal operationally significant anomalies to safety analysts with the goal of developing mitigation opportunities by changing standard operating procedures. The reduced false alarm rate of our framework compared to the unsupervised anomaly detection method is critical for domain experts to accept our reporting system and not just ignore the alarms, as has happened with other warning systems. Future work also includes developing richer rationales and ability to integrate multiple data sources for supporting those rationales for increased coverage of a wider range of operationally significant anomalies.

# CHAPTER 6

# EXPLANATIONS FRAMEWORK FOR DOCUMENT CLASSIFICATION

In this chapter, I discuss how we enrich the interaction between the human and active learner and provide a framework to elicit rich feedback, in the form of feature-based explanations, for document classification from the human experts. In Chapter 4, I presented the rationales framework for document classification, where labelers provided rationales by highlighting words in documents that convinced them to choose the label for a document. In this chapter, we extend this approach further by allowing the human to provide explanations, in the form of domain-specific features that support and oppose the classification of instances. Specifically, we ask labelers to highlight supporting phrases whose presence strengthens their belief in the label. We also ask labelers to provide opposing phrases, which, if removed from the document, would make their belief in the label stronger. I present the explanations framework for document classification that uses a simple approach to incorporate explanations into the training of any off-the-shelf classifier to speed-up the learning process.

This chapter is based on the work that I did with my advisor, Dr. Mustafa Bilgic. Part of the work in this chapter was published in the International Joint Conference on Artificial Intelligence (IJCAI) Workshop on BeyondLabeler - Human is More Than a Labeler, 2016 [100].

## 6.1 Introduction

Supervised learning approaches learn the class concepts using instances that are annotated with labels. When the labels for instances are not available, traditional active learning approaches [86], [94] ask humans to curate datasets by providing labels for selected instances to learn an effective classifier.

While examining instances, human labelers can provide information beyond just label annotations. Humans can provide domain knowledge, point out important features, provide feature annotations, rationales, and rules for classification. Many studies have shown that, unsurprisingly, supervised learning can benefit if the domain knowledge or reasonings for classification are imparted to the models. However, the main challenge has been to effectively incorporate this domain knowledge, which is often noisy and uncertain, into the training of the machine learning system.

Transmitting domain knowledge to learning systems has been studied for many years. For example, expert systems relied heavily on eliciting domain knowledge from the experts (e.g., Mycin system [12] was built through eliciting rules from the experts). Several explanation-based learning approaches [26], [70]) were developed to utilize domain knowledge to generalize target concepts using a single training example, and relied on domain experts to provide explanations for generalization. Examples of explanation-based learning systems include GENESIS [71] and SOAR [56]. Ellman [32] provides a survey on explanation-based learning. However, incorporating domain knowledge into the learning process and teaching the classification reasonings to supervised models is not trivial. Many supervised learning systems operate on feature-based representations of instances. For example, in document classification, instances are typically represented as feature vectors in a bag-of-words model. The domain knowledge elicited from the experts, however, often cannot be readily parsed into the representation that the underlying model can understand or operate on. The domain knowledge often refers to features rather than specific instances. Moreover, the domain knowledge is often at a higher level than instances, and sometimes, the domain knowledge is provided as unstructured information, such as free-form text entries.

Several approaches have been developed for knowledge-based classifiers such as knowledge-based neural networks [40], [116], [117] and knowledge-based support vector

machines [39]. Recent approaches for document classification have explored incorporating feature annotations [4], [31], [66], [81], [105], [107], and eliciting rationales for text classification [28], [76], [123]. These approaches were specific to classifiers, and hence, in Chapter 4, we proposed an approach to incorporate rationales for classification into the training of any off-the-shelf classifier.

In this chapter, we ask the labeler to provide explanations for their classification. Specifically, we ask the labeler to highlight the phrases in a document that *support* its label (i.e., the phrases whose presence reinforces the belief in the provided label) and phrases that *oppose* its label (i.e., the phrases whose presence weakens the belief in the provided label). For example, in a movie review "The actors were great but the plot was terrible. Avoid it", that is labeled as a 'negative' review, the phrases 'terrible' and 'avoid' support the 'negative' classification, whereas the word 'great' opposes the 'negative' classification. We present a simple and effective approach to incorporate these two types of explanations along with the labeled documents into the training of any off-the-shelf classifier. We evaluate our approach on three document classification datasets using multinomial naïve Bayes and support vector machines.

The rest of the chapter is organized as follows. In Section 6.2, we provide a brief background on eliciting labels and explanations during the curation of datasets. In Section 6.3, we describe our approach for incorporating explanations into the training of classifiers. In Section 6.4, we discuss experimental methodology and results. Finally, we conclude in Section 6.6.

## 6.2 Background

Let $\mathcal{D}$ be a set of document-label pairs $\langle x, y \rangle$, where the label (value of $y$) is known only for a small subset $\mathcal{L} \subset \mathcal{D}$ of the documents: $\mathcal{L} = \{\langle x, y \rangle\}$ and the rest $\mathcal{U} = \mathcal{D} \setminus \mathcal{L}$ consists of the unlabeled documents: $\mathcal{U} = \{\langle x, ? \rangle\}$. We assume that each document $x^i$ is

represented as a vector of features: $x^i \triangleq \{f_1^i, f_2^i, \cdots, f_n^i\}$. Each feature $f_j^i$ represents the binary presence (or absence), frequency, or tf-idf representation of the phrase $j$ in document $x^i$. Each label $y \in \mathcal{Y}$ is discrete-valued variable $\mathcal{Y} \triangleq \{y_1, y_2, \cdots, y_l\}$. Typical supervised learning approaches for data curation select a document $\langle x, ? \rangle \in \mathcal{U}$, query a labeler for its label $y$, and incorporate the new document $\langle x, y \rangle$ into the training set $\mathcal{L}$.

Several approaches looked at eliciting more than just labels from annotators. For example, feature annotation work looked at annotating features in tandem with labeling of documents [4], [31], [66], [81], [105], [107]. More recently, Zaidan et al. [2007] looked at eliciting rationales for the chosen label. In Chapter 4, we presented a classifier-agnostic approach to incorporate rationales into learning. In this chapter, we go one step further, and instead of asking simply the rationales, we ask for an explanation for the chosen label.

Explanations can be pretty broad, such as free-form text entries, rules, feature annotations, and rationales for classification. In this chapter, we focus on explanations for document classification where the human annotator highlights phrases in the document as explanations. Specifically, we ask the labeler to provide two kinds of highlighting. In the first kind, the human highlights the phrases that *support* the underlying label. For example, in sentiment classification, the human would highlight the positive sentiments in a positive review. In the second kind of highlighting, the human highlights the kind of phrases which, if were not present, would make the provided label even more correct. For example, these would be the negative sentiments in a generally-positive review.

Formally, in the learning with explanations framework, when a document is chosen for annotation, the labeler provides label $y^i$ for a document $x^i$, and explanations, which correspond to supporting features $SF(x^i)$ and opposing features $OF(x^i)$ for the label of $x^i$: $SF(x^i) = \{f_k^i : k \in x^i\}$ and $OF(x^i) = \{f_j^i : j \in x^i\}$. It is possible that the labeler cannot pinpoint any supporting or opposing phrases, in which case $SF(x^i)$ and $OF(x^i)$ are allowed to be empty sets. Next, we describe our approach for incorporating explanations

into the learning process.

### 6.3 Learning with Explanations

In this section, we describe our approach to incorporate feature-based explanations into the training of any off-the-shelf feature-based classifier. We assume that the explanations, i.e. the supporting and opposing features, returned by the labeler already exist within the dictionary of the underlying model.[8] For each labeled document, $\langle x^i, y^i, SF(x^i), OF(x^i)\rangle$, we create four types of pseudo-documents as follows:

- For each supporting feature in $SF(x^i)$, we create one pseudo-document containing only one phrase corresponding to a supporting feature, weight the supporting feature by $w_s$, and assign this pseudo-document the label $y^i$.
- For each opposing feature in $OF(x^i)$, we create one pseudo-document containing only one phrase corresponding to an opposing feature, weight the opposing feature by $w_o$ and assign this pseudo-document the label $\neg y^i$, where $\neg y^i$ is the opposite class label.
- We create one pseudo-document, $d'$ which is same as the original document, except the supporting and opposing features are removed, the remaining features are weighted by $w_{d'}^{y}$, and label $y^i$ is assigned to this pseudo-document.
- We create another pseudo-document, $d'$ which is same as the original document, except the supporting and opposing features are removed, the remaining features are weighted by $w_{d'}^{\neg y}$, and label $\neg y^i$ is assigned to this pseudo-document.

We incorporate these pseudo-documents into $\mathcal{L}$, on which the classifier is trained. We call

[8] If the features corresponding to the explanations do not exist in the dictionary, the dictionary can be expanded to include the new phrases, e.g. by creating and adding the corresponding n-grams to the dictionary.

this approach to incorporate explanations as *learning with explanations* (LwE).

We present a sample dataset with two documents, a positive movie review and a negative movie review, below. In these documents, the words that are returned as supporting features are underlined and the words that are provided as opposing features are in strikethrough.

*Document 1: This is a* ~~***weird***~~ *low-budget movie. It is* ~~***awful***~~ *but it pulls off somehow, that is why I* ***love*** *it.*
*Document 2: This movie had* ~~***great***~~ *acting,* ~~***good***~~ *photography, but the plot was* ***terrible****. Ultimately it was a* ***failure****.*

As this example illustrates, there are supporting positive (negative) words and opposing negative (positive) words in a positive (negative) document. Table 6.1 shows the traditional binary representation and LwE representation for *Document 2*.

One would expect that the weights for documents that contain only the explanations ($w_o$ and $w_s$) would be higher than the ones that exclude explanations ($w_{d'}^{y}$ and $w_{d'}^{\neg y}$), to emphasize the supporting features for the chosen label and opposing features for the opposite label, and de-emphasize the remaining phrases in both classes. Moreover, the documents that exclude explanations would be weighted higher for the chosen label, $w_{d'}^{y}$, than for the opposite label, $w_{d'}^{\neg y}$, since even without the supporting features, the document would more likely belong to class $y$ than to class $\neg y$. This is because the document is overall labeled as $y$ and the labeler is not necessarily asked to provide all the explanations for classification.

This approach to incorporate explanations is not tied to any classifier. Any off-the-shelf classifier that can work with numerical features, such as multinomial naïve Bayes (for which all the feature weights, $w_s$, $w_o$, $w_{d'}^{y}$, and $w_{d'}^{\neg y}$, must be non-negative), logistic

Table 6.1. The binary representation (top) and its LwE transformation (bottom) for Document 2 (D2). Stop words are removed. LwE creates multiple pseudo-documents with various feature weights and class labels.

| | movie | great | acting | good | photography | terrible | plot | ultimately | failure | **label** |
|---|---|---|---|---|---|---|---|---|---|---|
| Binary representation | | | | | | | | | | |
| D2 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | $-$ |
| LwE transformation of the binary representation | | | | | | | | | | |
| $D2_1$ | | | | | | $w_s$ | | | | $-$ |
| $D2_2$ | | | | | | | | | $w_s$ | $-$ |
| $D2_3$ | | $w_o$ | | | | | | | | $+$ |
| $D2_4$ | | | | $w_o$ | | | | | | $+$ |
| $D2_5 w_{d'}^{y}$ | | | $w_{d'}^{y}$ | | $w_{d'}^{y}$ | | $w_{d'}^{y}$ | $w_{d'}^{y}$ | | $-$ |
| $D2_6 w_{d'}^{\neg y}$ | | | $w_{d'}^{\neg y}$ | | $w_{d'}^{\neg y}$ | | $w_{d'}^{\neg y}$ | $w_{d'}^{\neg y}$ | | $+$ |

regression, and support vector machines, can be used as the underlying model.

## 6.4 Experimental Methodology and Results

In this section we first describe the settings, datasets, and classifiers used for our experiments and how we simulated a human labeler to provide explanations for document classification. Then, we present the results comparing *traditional learning* (TL), *learning with rationales* (LwR), and *learning with explanations* (LwE). We use LwR strategy presented in Chapter 4 as a baseline for our LwE approach.

**6.4.1 Methodology.** We experimented with three document classification datasets, which are described in Table 6.2. We evaluated our strategy using multinomial naïve Bayes and support vector machines, as these are strong classifiers for text classification. We used the scikit-learn [78] implementation of these classifiers.

Table 6.2. Description of the datasets: the domain, number of instances in training and test datasets, and size of vocabulary.

| Dataset | Task | Train | Test | Vocab. |
|---|---|---|---|---|
| IMDB | Sentiment analysis of movie reviews [61] | 25,000 | 25,000 | 27,272 |
| NOVA | Email classification (politics vs. religion) [44] | 12,977 | 6,498 | 16,969 |
| WvsH | 20Newsgroups (Windows vs. Hardware) | 1,176 | 783 | 4,026 |

To compare various strategies, we used learning curves. The initially labeled dataset was bootstrapped using 10 documents by picking 5 random documents from each class. Iteratively, 10 documents were chosen at random and were annotated by TL, LwR, and LwE approaches. This process was repeated 10 times, and average learning curves over 10 different runs are presented. We evaluated all the strategies using AUC (Area Under an ROC Curve) measure. For this study we selected documents randomly, as opposed to using active learning approaches such as uncertainty sampling [57], to run a controlled experiment where TL, LwR, and LwE, all operated on the same set of documents.

The LwR approach [101] elicits rationales for classification, and modifies the document to weight rationale features higher than other features within that document. On the other hand, LwE approach elicits explanations, where supporting features are rationales for classification, but it goes one step further than LwR, and elicits opposing features. For each document, the rationales provided for LwR by the simulated labeler are the same as supporting features provided for LwE, so compared to LwR, LwE has the additional advantage of receiving opposing features from the labeler. However, we cannot argue that the difference between LwR and LwE strategies is only due to eliciting opposing features, since LwE creates several pseudo-documents for explanations, whereas LwR re-weights features within a document.

For LwR baseline, we used the same weights that were used in Chapter 4. That is, we set the weights for rationales and the remaining features of a document to 1 and 0.01

respectively (i.e. $r = 1$ and $o = 0.01$). For LwE using multinomial naïve Bayes, we set the weights $w_s = 100$, $w_o = 100$, $w_{d'}^{y} = 1$, and $w_{d'}^{\neg y} = 0.01$. For LwE using support vector machines, we set the weights $w_s = 1$, $w_o = 1$, $w_{d'}^{y} = 0.1$, and $w_{d'}^{\neg y} = 0.001$. These weights worked reasonably well for all three datasets. We experimented with fixed weight settings in this section to show that LwE can do well across datasets even without parameter tuning. In Section 6.4.3, we also present results using the best possible parameter settings for all approaches.

We simulated the human labeler in the same way as in Chapter 4. The simulated labeler recognized phrases as positive (negative) features that had the highest $\chi^2$ (chi-squared) statistic in at least 5% of the positive (negative) documents. To make the labeler's effort as small as possible, we ask the labeler to highlight any one feature as supporting feature and any one feature as opposing feature, as opposed to asking the labeler to highlight all supporting and opposing features. We also allowed the labeler to skip highlighting any phrase as supporting or opposing, if the answer is not obvious, i.e. if the labeler cannot pinpoint any phrase as a supporting/opposing feature.

**6.4.2 Results.** Figure 6.1 presents the learning curves on three document classification datasets using multinomial naïve Bayes and support vector machines. The results show that LwE provides huge improvements over TL for all datasets and classifiers. We performed pairwise one-tailed t-tests under significance level of 0.05, where pairs are area under the learning curves for 10 runs of each method. If a method has higher average performance than a baseline with a significance level of 0.05 or better, it is a win, if it has significantly lower performance, it is a loss, and if the difference is not statistically significant, the result is a tie. For all three datasets and two classifiers, LwE statistically significantly outperforms TL. For NOVA dataset, LwE outperforms TL on the first half of the learning curve, but later loses to TL under fixed-parameter settings. As we show later in Section 6.4.3, under best parameter settings, LwE outperforms TL for NOVA at all budget levels. LwR also

performs much better than TL, and is therefore a strong baseline for LwE, however, LwE provides further improvements over LwR. The t-test results show that for IMDB and NOVA datasets, LwE statistically significantly wins over LwR using both classifiers. For WvsH dataset, LwE wins over LwR using multinomial naïve Bayes and LwE ties with LwR using support vector machines.

It is not a surprise that LwE is able to outperform TL and LwR. LwE is asking the human to provide further information than just the labels. What we are arguing, however, is that LwE is able to integrate this extra information into the learning process effectively. In Table 6.3, we compare the number of annotated documents required by TL, LwR, and LwE to achieve a target AUC performance using multinomial naïve Bayes. The results using support vector machines are similar and are omitted to avoid redundancy. The ratios of number of documents required by TL and LwE (**TL/LwE**) in this table show that LwE drastically accelerates learning. For example, for IMDB dataset, in order to achieve AUC of $0.85$ using multinomial naïve Bayes, TL requires labeling 233 documents, whereas LwE achieves the same AUC with just 51 labeled documents. We note that providing explanations might take more time than providing just the labels, however, for this case, if the labeler does not take more than 4.5 times the amount time in providing explanations, it is better to ask labeler to provide explanations along with labels for documents. Moreover, LwE often requires fewer labeled documents compared to LwR to achieve the same target AUC. As Table 6.3 shows, the ratio of number of documents required by the LwR and LwE (**LwR/LwE**) is often greater than 1 and sometimes as large as $3.2$. That is, if the labeler is already providing a rationale, then if the labeler does not spend more than 3 times the amount of time in providing an opposing feature, labeling documents with explanations is worth the expert's time.

**6.4.3 LwE vs. TL and LwR under Best Parameter Settings.** So far, we have seen that LwE provides improvements over TL and LwR. All three strategies used a fixed weight

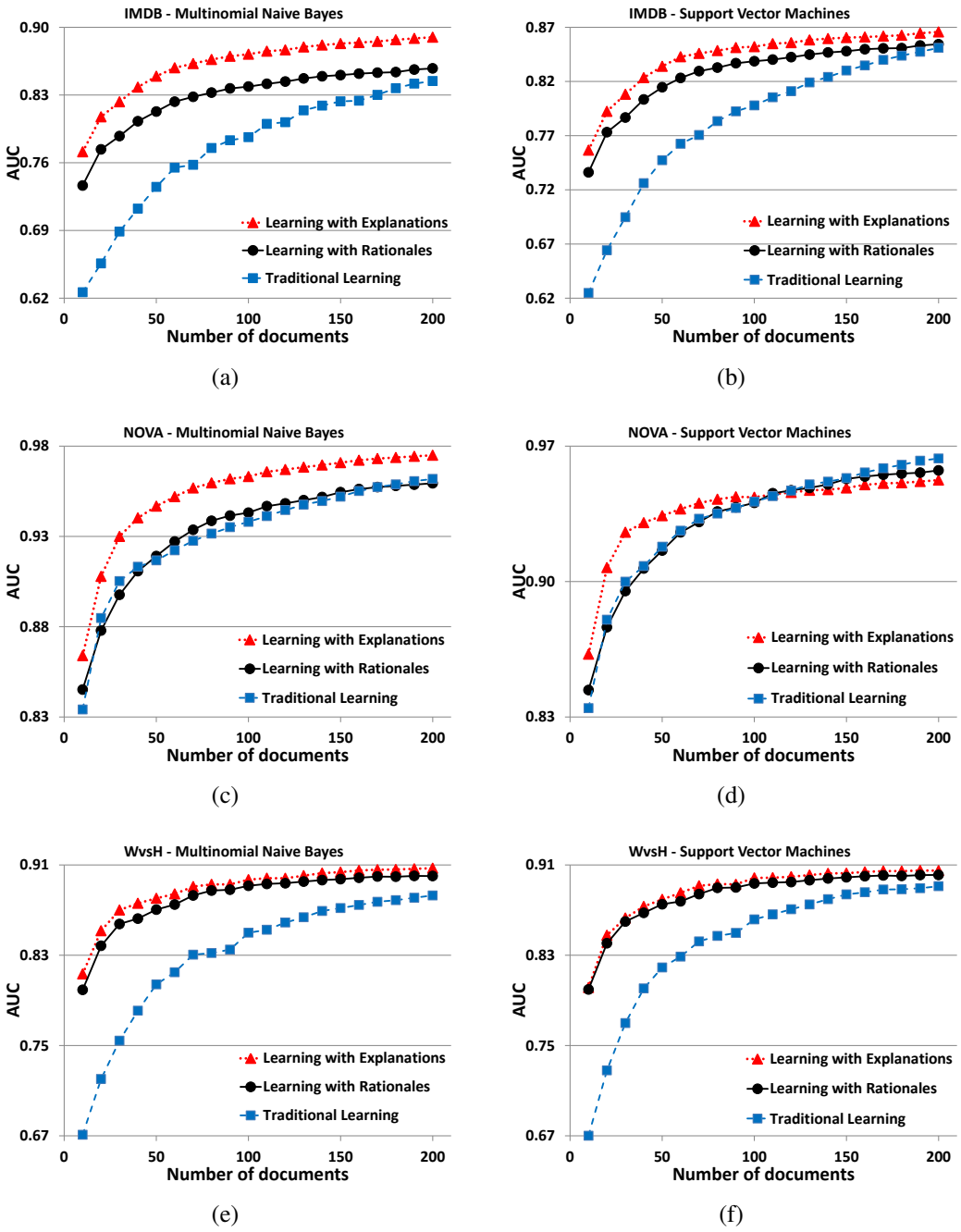



Figure 6.1. Comparison of LwE to TL and LwR. LwE provides significant improvements over TL. LwE statistically significantly wins over LwR for (a), (b), (c), (d), and (e). LwE ties with LwR on WvsH dataset using support vector machines (f).

setting for hyper-parameters. In this section, we examine how TL, LwE, and LwR methods would behave when they are tuned using the best parameter settings. To find out, we searched over several parameters, optimizing on the test data. Note that, normally, one

Table 6.3. Comparison of number of documents required to achieve a target AUC by TL, LwE, and LwR using multinomial naïve Bayes. 'n/a' represents that a target AUC cannot be achieved by a method.

| Dataset | Method | Target AUC | | | | | |
|---|---|---|---|---|---|---|---|
| | | **0.70** | **0.75** | **0.80** | **0.85** | **0.90** | **0.95** |
| IMDB | TL | 37 | 65 | 106 | 233 | 841 | n/a |
| | LwR | 10 | 16 | 37 | 164 | n/a | n/a |
| | LwE | 5 | 9 | 18 | 51 | 379 | n/a |
| | **TL/LwE** | **7.4** | **7.22** | **5.89** | **4.57** | **2.22** | **n/a** |
| | **LwR/LwE** | **2** | **1.78** | **2.06** | **3.22** | **n/a** | **n/a** |
| NOVA | TL | 3 | 3 | 5 | 12 | 28 | 126 |
| | LwR | 2 | 3 | 4 | 11 | 31 | 110 |
| | LwE | 2 | 3 | 4 | 9 | 16 | 51 |
| | **TL/LwE** | **1.5** | **1** | **1.25** | **1.33** | **1.75** | **2.47** |
| | **LwR/LwE** | **1** | **1** | **1** | **1.22** | **1.94** | **2.16** |
| WvsH | TL | 17 | 33 | 57 | 127 | 380 | n/a |
| | LwR | 4 | 6 | 12 | 33 | 188 | n/a |
| | LwE | 4 | 6 | 12 | 30 | 146 | n/a |
| | **TL/LwE** | **4.25** | **5.5** | **4.75** | **4.23** | **2.6** | **n/a** |
| | **LwR/LwE** | **1** | **1** | **1** | **1.1** | **1.29** | **n/a** |

would never optimize over the test data in practical settings. This is a hypothetical setting, and the purpose is to conduct a controlled experiment to tease out whether the LwE framework is different, better, or worse than the LwR framework, when both are tuned using best possible parameter settings.

For LwR, we searched for weights $r$ and $o$, and for LwE, we searched for weights $w_s$, $w_o$, $w_{d'}^{y}$, and $w_{d'}^{\neg y}$. In addition to these parameters, for multinomial naïve Bayes, we searched for the smoothing parameter, $\alpha$, and for support vector machines, we searched for the regularization parameter, $C$. For TL, we searched for $\alpha$ for multinomial naïve Bayes, and $C$ for support vector machines. For all hyper-parameters, we performed a grid search for values between $10^{-3}$ and $10^{3}$.

Figure 6.2 presents learning curves comparing LwE to TL and LwR under the best parameter settings. The t-tests results show that LwE statistically significantly wins over TL for all three datasets. For IMDB and NOVA datasets, LwE wins over LwR using both classifiers. For WvsH dataset, LwE ties with LwR using both classifiers.

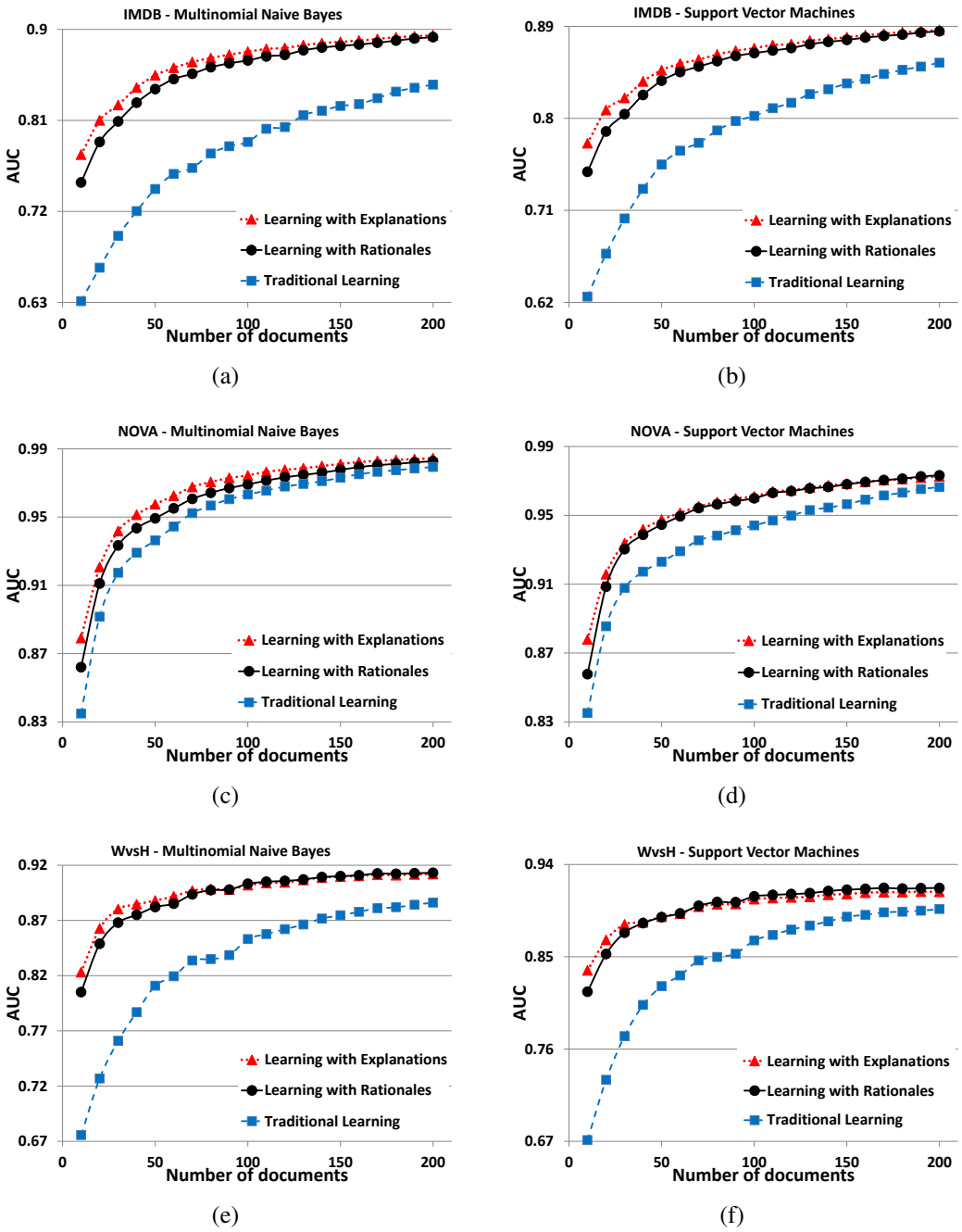



Figure 6.2. Comparison of LwE with TL and LwR under best parameter settings.

**6.4.4 Any One Explanation vs. All Explanations.** So far, we have shown that incorporating explanations consisting of *any* one supporting feature and *any* one opposing feature provides improvements over LwR and TL. In this section, we compare LwE to LwR when *all* the explanations are incorporated using LwE and *all* the rationales are incorporated using LwR.

Figure 6.3 shows the learning curves achieved with LwE using any one explanation and all the explanations, and LwR using any one rationale and all the rationales. Incorporating all the explanations into learning provides improvements over incorporating any one explanation, and incorporating all the rationales into learning provides improvements over incorporating any one rationale for all three datasets using both multinomial naïve Bayes and support vector machines.

**6.4.5 Learning with Noisy Explanations.** So far, we assumed that the simulated labeler is perfect and highlights only the most apparent words as explanations. In reality, however, user-annotated explanations can be noisy, where users do not pinpoint just the important words, but rather highlight phrases (or even sentences) that span several words. In this section, we investigate the effect of noisy explanations on the performance of LwE framework, and the effect of noisy rationales on the performance of LwR framework.

In order to simulate an expert that provides noisy explanations, we allowed the simulated labeler to highlight $k/2$ words before and $k/2$ words after the word that is recognized as a supporting or opposing feature by the simulated labeler. That is, for each word that is recognized as an explanation, the simulated labeler returns $k$ additional words around the explanation as noise. Similarly, for the LwR approach, we allowed the simulated labeler to select $k/2$ words before and $k/2$ words after the word that is recognized as a rationale by the simulated labeler. The parameter $k$ controls the level of noise in explanations and rationales. We experimented with $k = 2$ and $k = 4$. Getting the noise, i.e. the words, around explanations or rationales words requires parsing the text within the document, however,

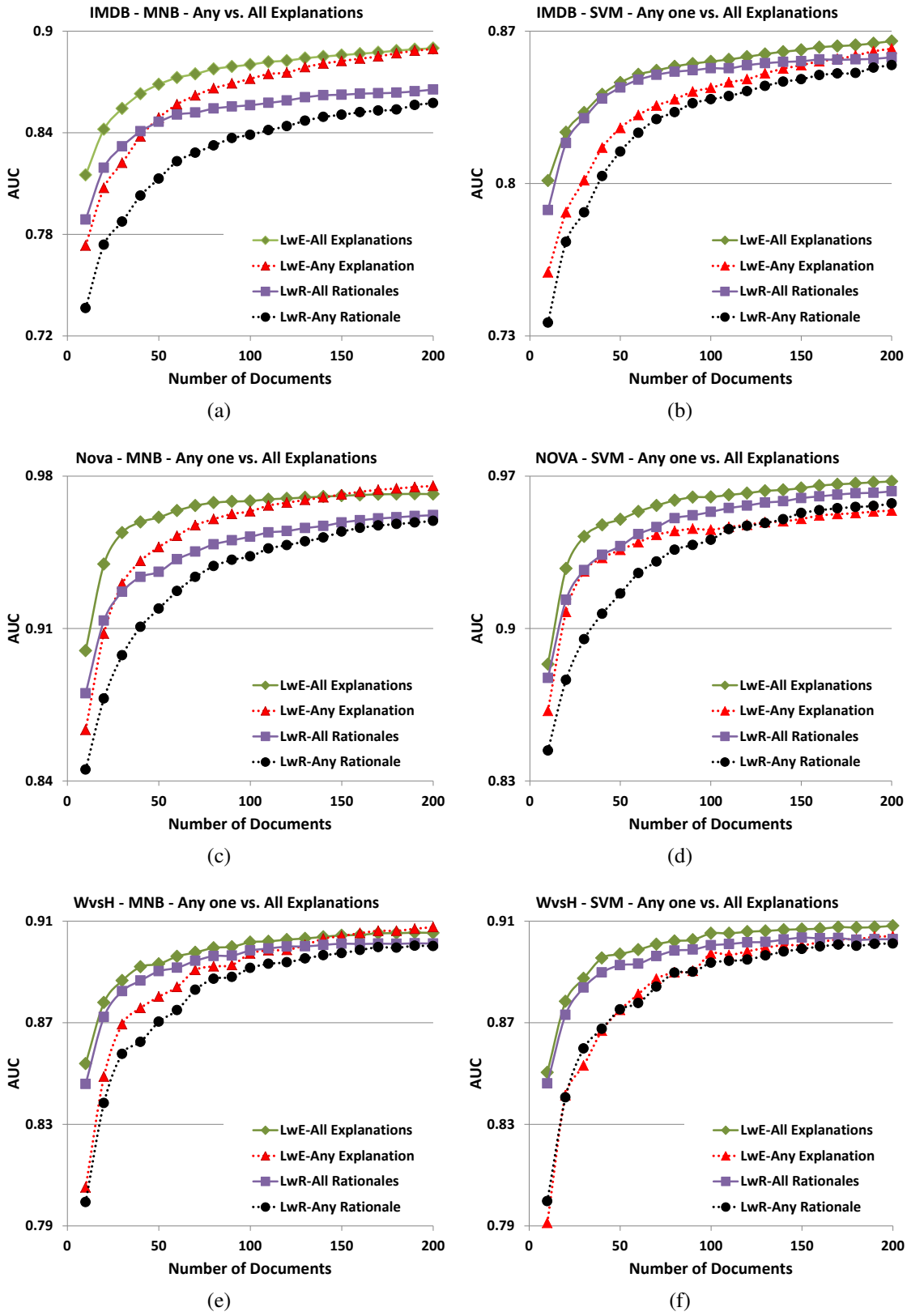



Figure 6.3. LwE with incorporating any one explanation vs. all explanations, and LwR with incorporating any one rationale vs. all rationales for all three dataset using multinomial naïve Bayes and support vector machines.

since the original text for NOVA dataset is not available, we excluded this dataset from the study in this section. We experimented with noisy explanations and noisy rationales for only IMDB and WvsH datasets.

Figures 6.4 and 6.5 show the learning curves achieved by LwE under two noise levels, where $k = 2$ and $k = 4$. Similarly, Figures 6.6 and 6.7 show the learning curves achieved by LwR under two noise levels, where $k = 2$ and $k = 4$. For both LwE and LwR frameworks, we experimented with fixed weight settings, as described in Section 6.4.1, for the hyper-parameters to show the effect of noise on each of the two frameworks. That is, for LwE using multinomial naïve Bayes, we set the weights $w_s = 100$, $w_o = 100$, $w_{d'}^{y} = 1$, and $w_{d'}^{\neg y} = 0.01$, and for LwE using support vector machines, we set the weights $w_s = 1$, $w_o = 1$, $w_{d'}^{y} = 0.1$, and $w_{d'}^{\neg y} = 0.001$. For LwR, we set the weights $r = 1$ and $o = 0.01$.

As Figures 6.4 and 6.5 show, the performance of LwE decreases as the noise level increases, which is not surprising, however LwE is affected more by noisy explanations than LwR by noisy rationales (Figures 6.6 and 6.7). Note that LwE asks the labeler to provide supporting and opposing phrases, whereas LwR asks the labeler to highlight only the rationales for classification. Not all documents have supporting and opposing phrases in them, however, it is more likely for a document to have a supporting phrase than an opposing phrase. For IMDB dataset, the simulated labeler returned at least one supporting phrase for 91.24% of the documents and at least one opposing phrase for 64.42% of the documents. The same simulated labeler for LwR returned at least one rationale for 91.24% of the documents. Hence, the number of phrases returned as explanations by the simulated labeler is ≈1.7 times more than the number of phrases returned as rationales for the IMDB dataset. Similarly, for WvsH dataset, the simulated labeler returned at least one supporting phrase for 94.81% of the documents and at least one opposing phrase for 42.6% of the documents. The same simulated labeler for LwR returned at least one rationale for 94.81% of the documents. Hence, the number of phrases returned as explanations by the simulated

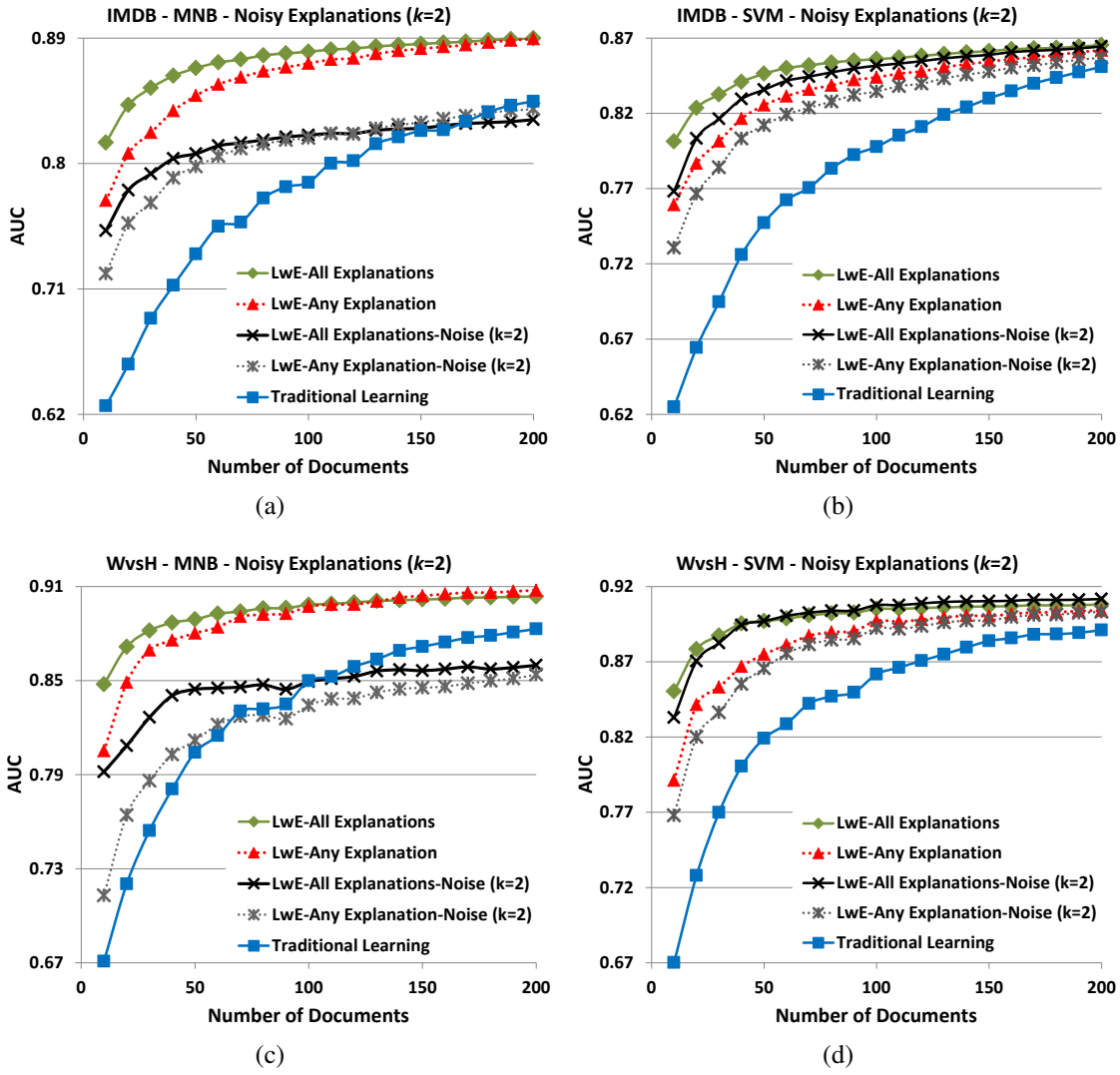



Figure 6.4. LwE with noisy explanations, noise level, $k$=2, for IMDB and WvsH datasets using multinomial naïve Bayes and support vector machines.

labeler is ≈1.45 times more than the number of phrases returned as rationales for the WvsH dataset. With the same noise level, $k$, LwE framework learns with more noise than the LwR framework.

In Figures 6.4, 6.5, 6.6, and 6.7, we used fixed parameter settings, as described in Section 6.4.1, for the hyper-parameters of LwE and LwR to investigate the effect of noise in explanations and rationales. In practice, however, when the explanations are noisy, the confidence in explanations should be reflected in the weights, $w_s$, $w_o$, $w_{d'}^{y}$, and $w_{d'}^{\neg y}$. When

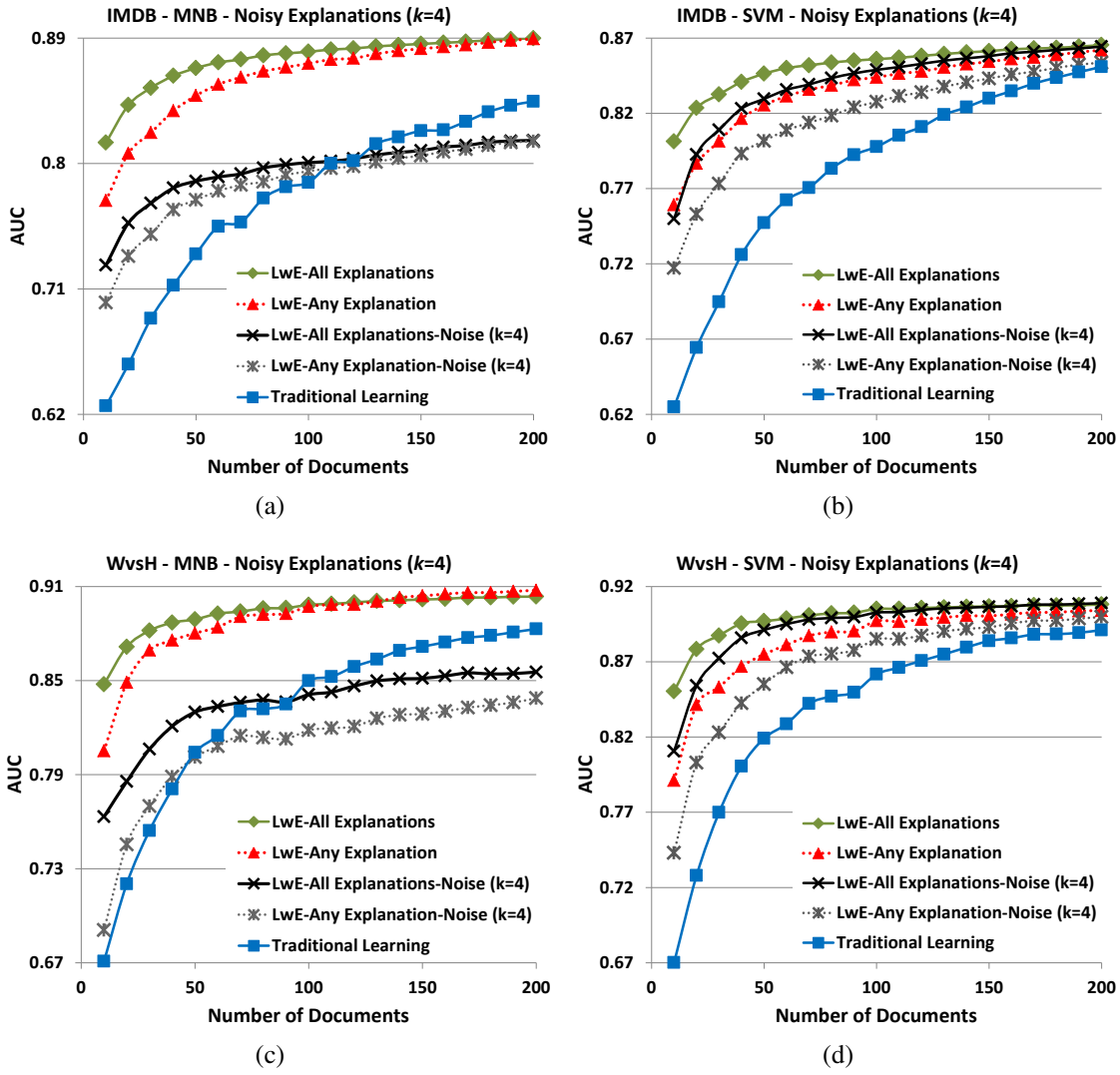



Figure 6.5. LwE with noisy explanations, noise level, $k$=4, for IMDB and WvsH datasets using multinomial naïve Bayes and support vector machines.

the expert provides precise and accurate explanations, the weights $w_s$ and $w_o$ should be much higher than $w_{d'}^{y}$ and $w_{d'}^{\neg y}$. And, when the expert provides noisy explanations, the weights $w_s$ and $w_o$ should become closer to $w_{d'}^{y}$ and $w_{d'}^{\neg y}$, as the noise level increases. That is, the confidence in the expert's explanations should be reflected in these weights.

We observed that when the explanations are noisy, setting the weights $w_s$ and $w_o$ closer to $w_{d'}^{y}$ and $w_{d'}^{\neg y}$ improves the performance of LwE. A more practical approach is to tune these parameters (e.g., using cross validation) at each step of the learning curve. We

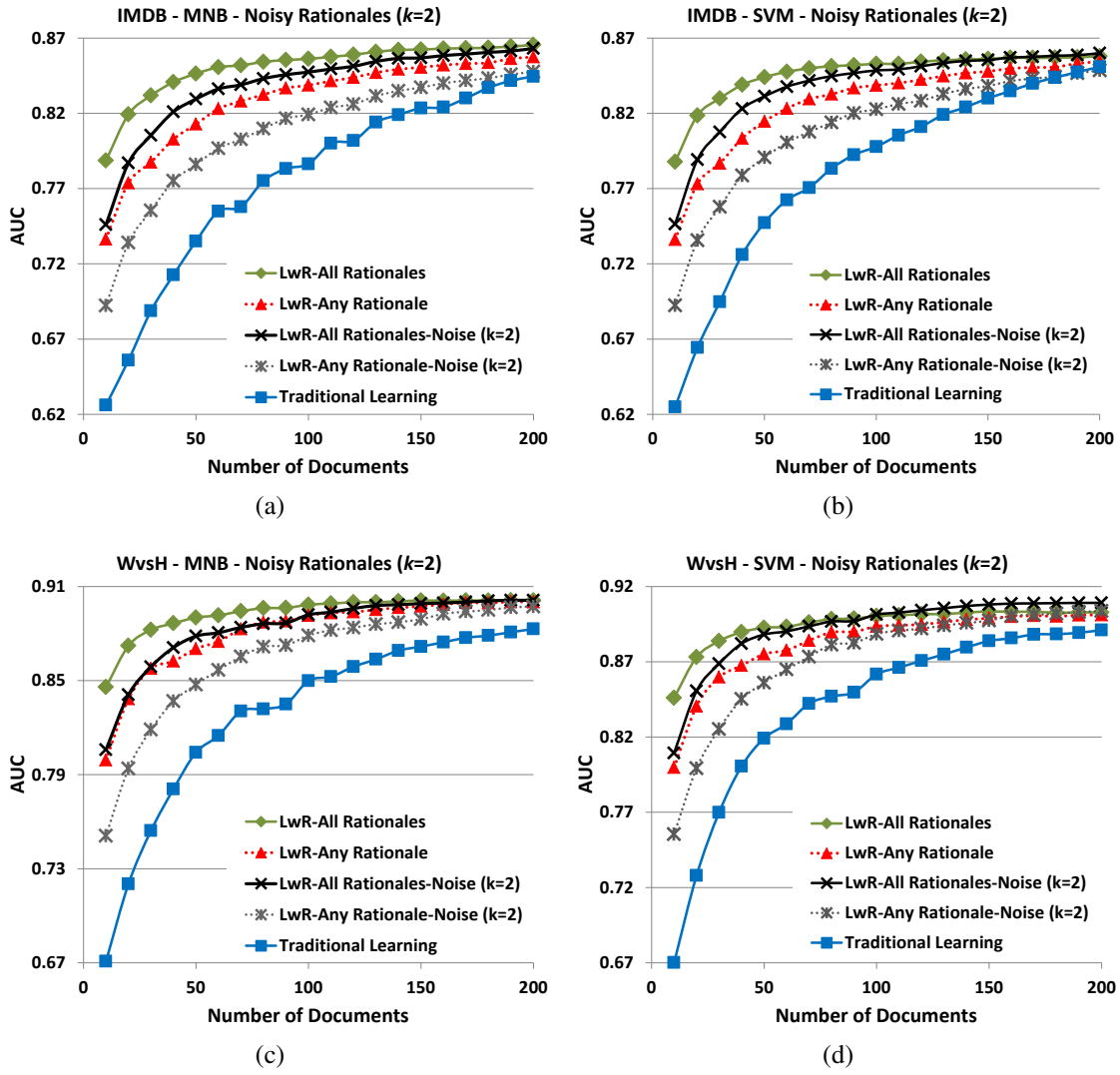



Figure 6.6. LwR with noisy rationales, noise level, $k$=2, for IMDB and WvsH datasets using multinomial naïve Bayes and support vector machines.

investigated the performance of LwE with noisy explanations when the hyper-parameters are optimized at each iteration of learning using the training data. We searched for the optimal parameter settings for $w_s$, $w_o$, $w_{d'}^{y}$, $w_{d'}^{\neg y}$ for LwE. We also searched for the smoothing parameter, $\alpha$, for multinomial naïve Bayes and regularization parameter, $C$, for support vector machines. At each step of the learning curve, we performed a grid search to search for values between $10^{-2}$ and $10^{2}$ for all the hyper-parameters using five-fold cross validation on training set. We optimized all the parameters for AUC measure, since AUC is the

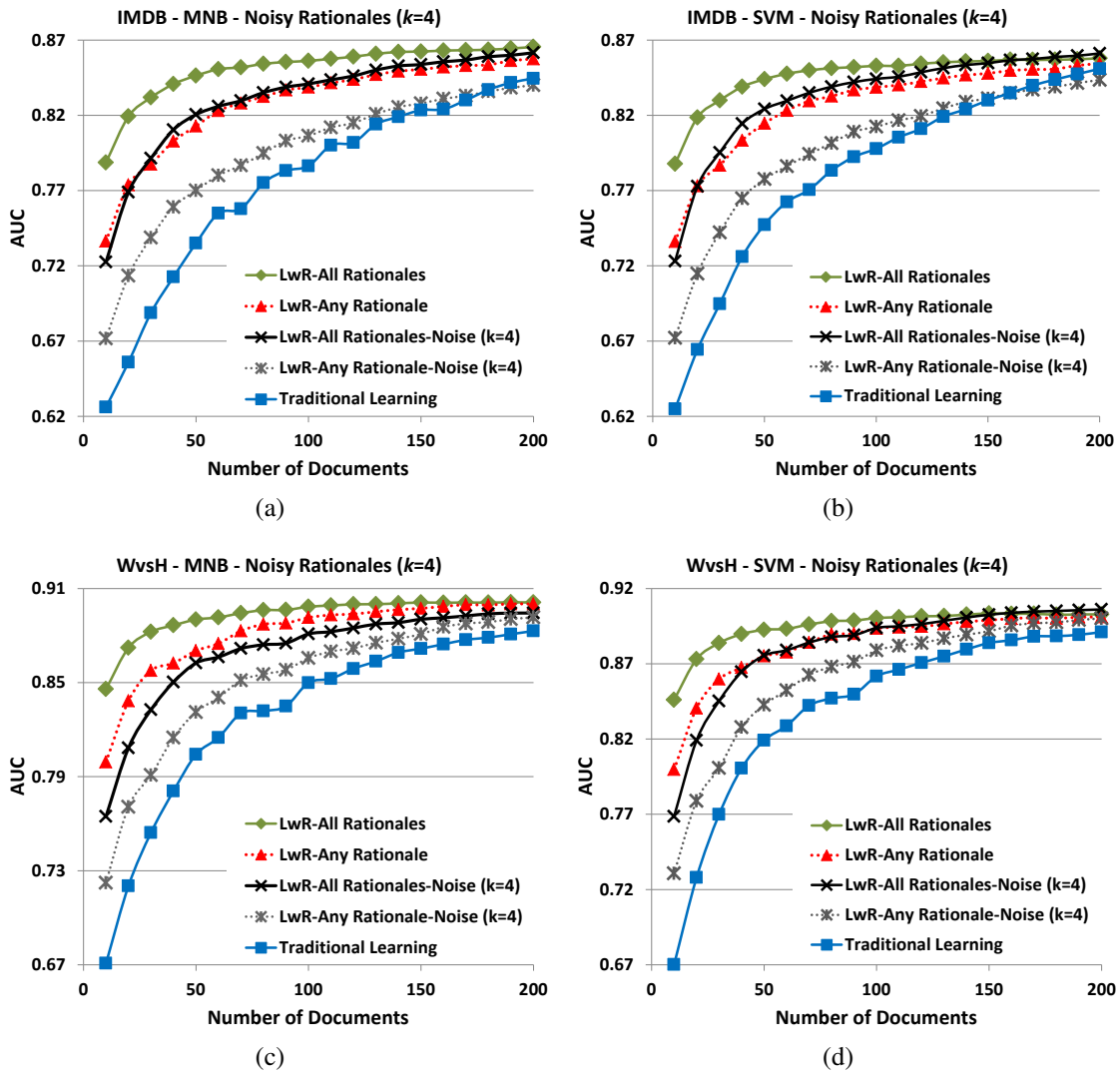
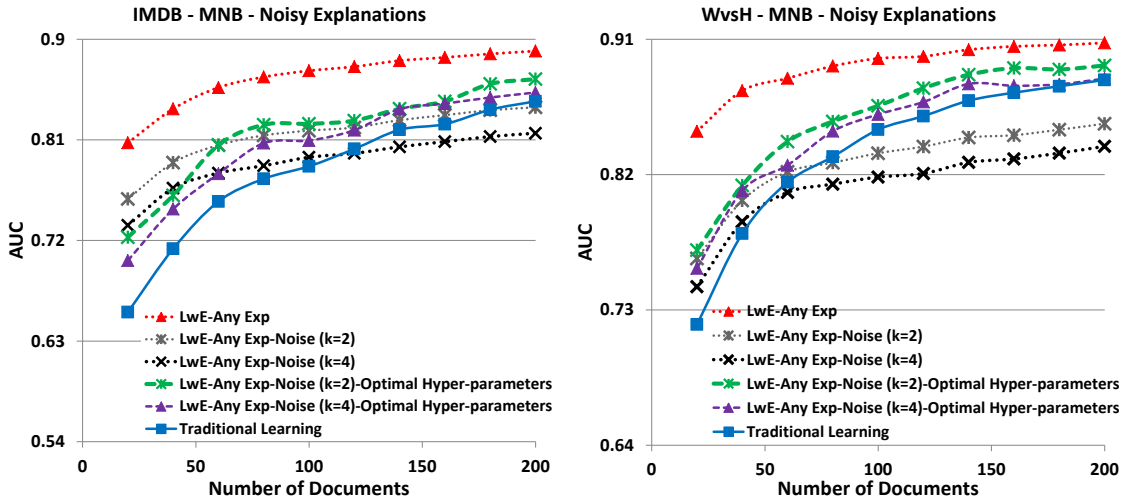



Figure 6.7. LwR with noisy rationales, noise level, $k$=4, for IMDB and WvsH datasets using multinomial naïve Bayes and support vector machines.

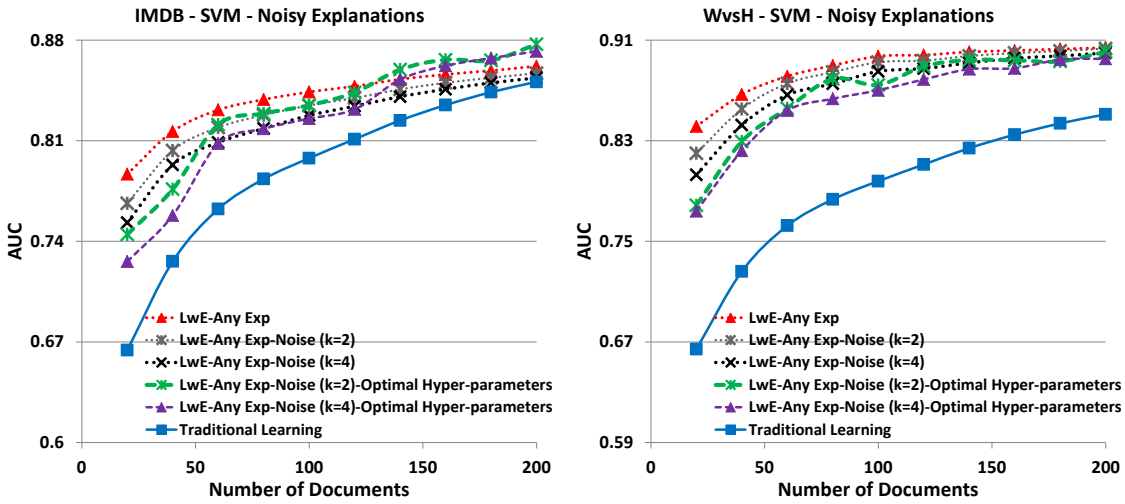
target performance measure in our experiments. Figures 6.8 and 6.9 present the learning curves for LwE with noise levels, $k = 2$ and $k = 4$, for multinomial naïve Bayes and support vector machines when any one explanation for the document is incorporated using LwE.

Figures 6.8 and 6.9 present the learning curves for LwE when the hyper-parameters are optimized at each iteration of learning. As these figures show, the performance of LwE with noisy rationales improves when optimal parameters are searched at each iteration of




(a) (b)

Figure 6.8. LwE with any one noisy explanation (LwE-Any Exp) and noise levels, $k$=2 and $k$=4, with optimal hyper-parameter settings for IMDB and WvsH datasets using multinomial naïve Bayes.




(a) (b)

Figure 6.9. LwE with any one noisy explanation (LwE-Any Exp) and noise levels, $k$=2 and $k$=4, with optimal hyper-parameter settings for IMDB and WvsH datasets using support vector machines.

learning.

**6.4.6 Learning with Explanations with Fallible and Reluctant Experts.** So far, we

assumed that experts provide any one explanation (or rationale) for a document, however, some experts can be reluctant, but precise, in providing explanations (or rationales). We also assumed that the explanations and rationales provided by experts are always correct, however, some experts can be fallible and imprecise in providing explanations (or rationales). In this section, we evaluate the performances of LwE and LwR when the experts are fallible or reluctant in providing explanations or rationales.

We assume that reluctant experts do not provide all the explanations (or rationales), however, when they do provide an explanation (or a rationale), they highlight the most important phrases as explanations (or rationales). In order to simulate a reluctant expert, we allowed the labeler to identify only a few, but most important, phrases as supporting or opposing features. The simulated reluctant expert recognized supporting and opposing phrases as words that have the highest $\chi^2$ (chi-squared) statistic in at least 10% of the documents. This resulted in a simulated labeler that identified very few words that were most apparent as explanations (or rationales) and hence, would not provide an explanation (or rationale) for all the documents. The simulated reluctant expert recognized only 34 words, 54 words, and 51 words as supporting/opposing phrases for IMDB, NOVA, and WvsH datasets respectively.

We assume that a fallible expert has a higher chance of providing an explanation (or a rationale), however, the explanation (or rationale) might not be good enough, that is, the credibility of the explanations provided by the labeler is questionable. To simulate a fallible expert, we allowed the simulated labeler to identify phrases as supporting or opposing features that have the highest $\chi^2$ (chi-squared) statistic in at least 1% of the documents. This resulted in the simulated labeler that identified more words as explanations (or rationales), and thus has a higher chance of providing explanations (or rationales), however some of the words might not be good explanations (or rationales). The simulated fallible expert recognized 108 words, 437 words, and 147 words as supporting/opposing phrases for IMDB,

NOVA, and WvsH datasets respectively.

Figure 6.10 shows the learning curves for LwE and LwR with fallible experts for all three datasets using multinomial naïve Bayes and support vector machines. When the expert is fallible, the performances of both LwE and LwR decrease, which is not surprising, because although the fallible expert recognizes more words as explanations, the explanations could be of inferior quality. Figure 6.11 shows the learning curves achieved with reluctant experts. When the expert is reluctant, the performances of both LwE and LwR decreases, especially at the beginning of the learning, but later on, the performances of both LwE and LwR increase. A reluctant expert does not provide much feedback, but when s/he does provide feedback, it is of superior quality compared to the feedback from a fallible expert. Thus, at the beginning of learning, the performances of both LwE and LwR decrease because the reluctant expert does not provide much feedback, however, providing a superior quality feedback could improve learning in the long-term, e.g., in the case of IMDB dataset (Figures 6.11(a) and 6.11(b)).

## 6.5 Graphical User Interface

In this section, we evaluate our approach to incorporate explanations into learning using real user-annotated explanations. We designed and ran a user study to investigate the performance of LwE framework with real user-annotated explanations. We provide a graphical user interface that facilitates users to easily highlight supporting and opposing phrases using different highlighting colors. To make the highlighting task easier for the user, instead of asking the user to highlight supporting and opposing phrases, we simply asked the user to highlight all the positive phrases using 'green' color and highlight all the negative phrases using 'red' color. The explanations framework determines which phrases are supporting and which phrases are opposing, based on the highlighting color of the phrases and the label chosen by the user for a document. We had three users for our study, and we refer to them as User1, User2, and User3 in this study. Each user was shown 200

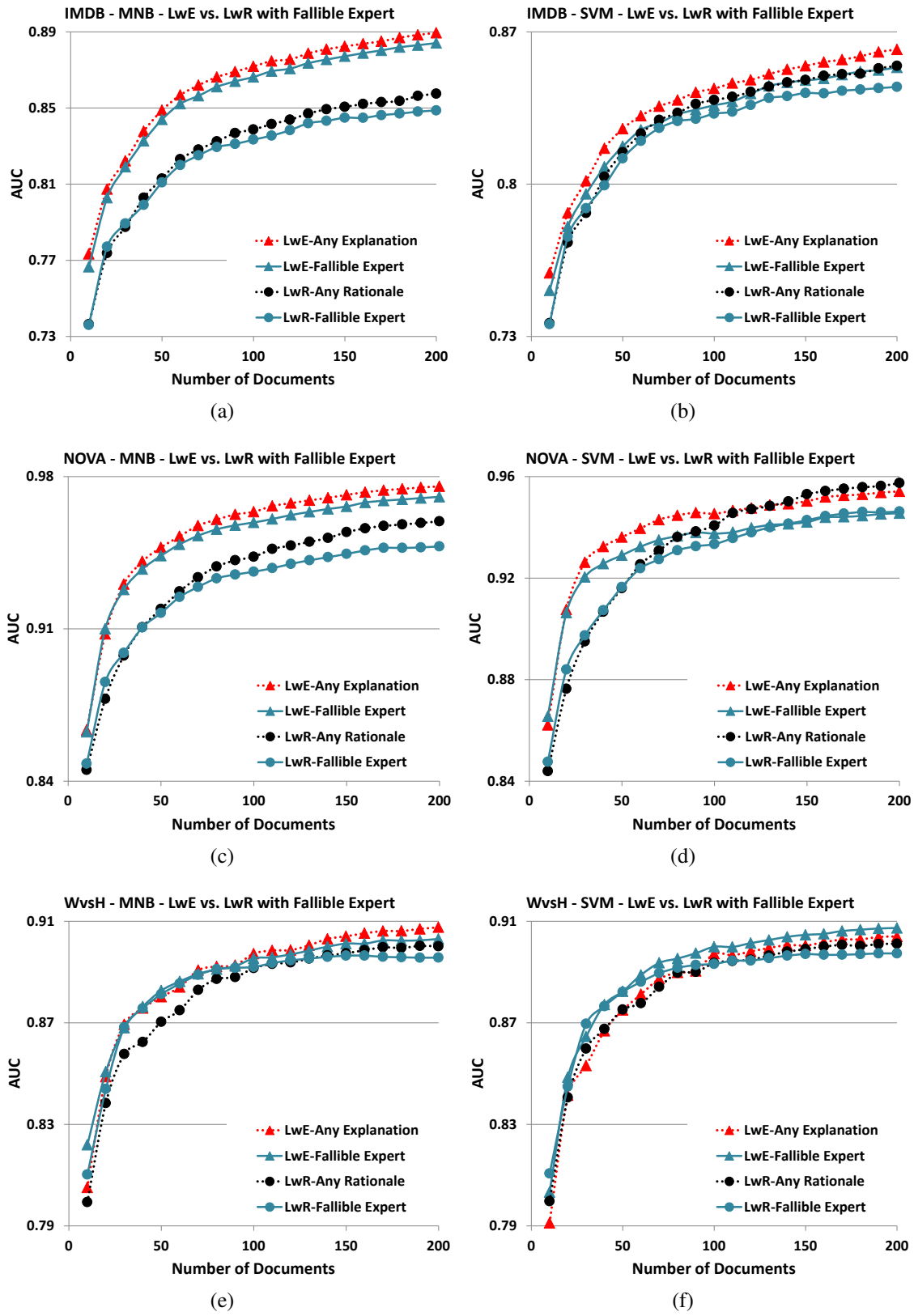



Figure 6.10. Performances of LwE and LwR with fallible experts.

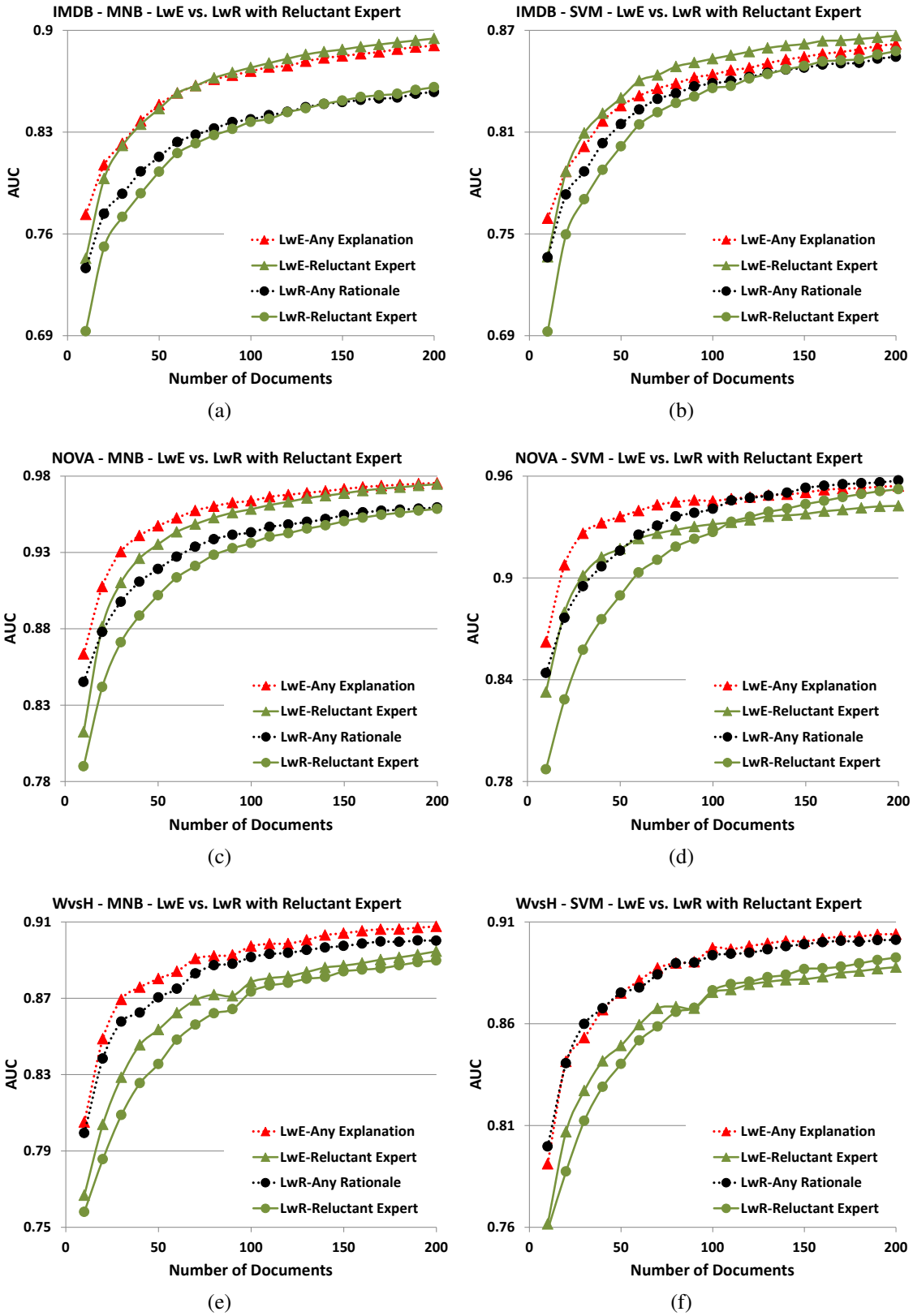



Figure 6.11. Performances of LwE and LwR with reluctant experts.

movie reviews in the same order. For each movie review, we recorded the response time, label (positive/negative), and the supporting and opposing phrases provided for the movie review.

Figure 6.12 shows a screenshot of the graphical user interface for the learning with explanations framework for document classification. For the user study, we tasked the user to annotate IMDB movie reviews as positive or negative sentiment reviews. We displayed the movie review in the graphical user interface and asked the user to highlight positive phrases with 'green' color highlighter and negative phrases with 'red' color highlighter, and provide a label, 'positive' or 'negative', for the movie review. We asked the user to highlight as many positive sentiment phrases and negative sentiment phrases as s/he could identify in a document.

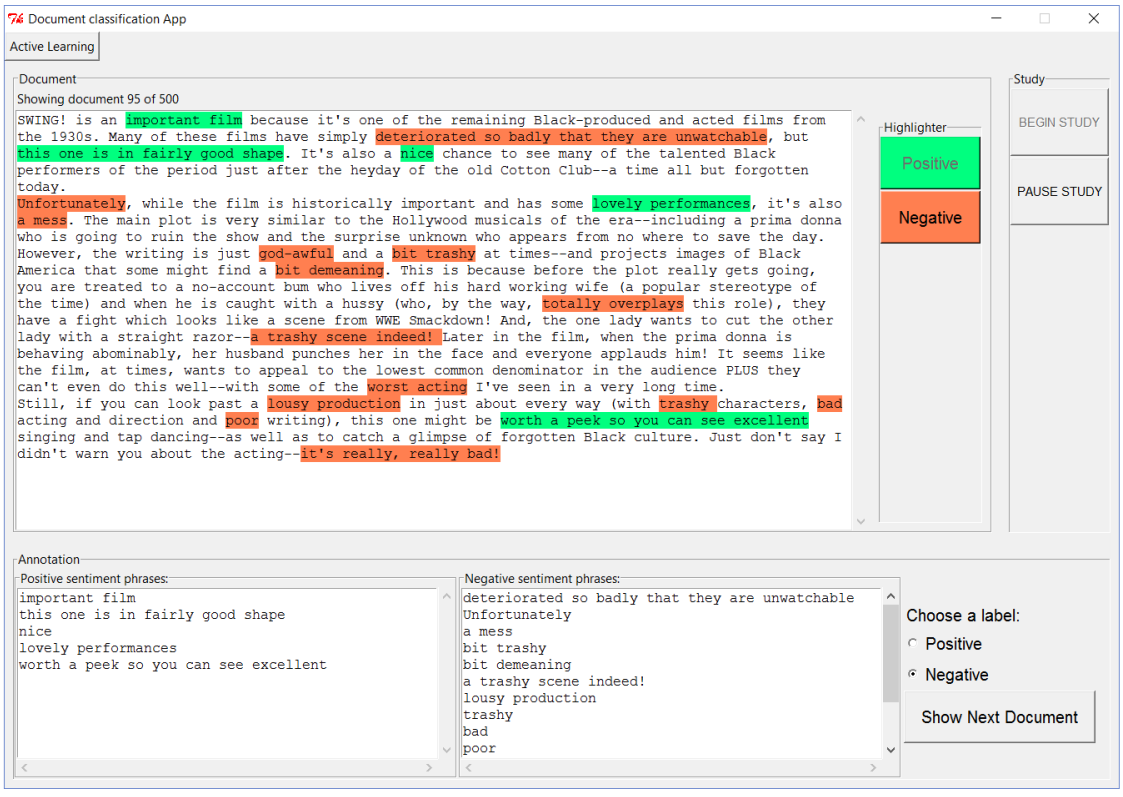
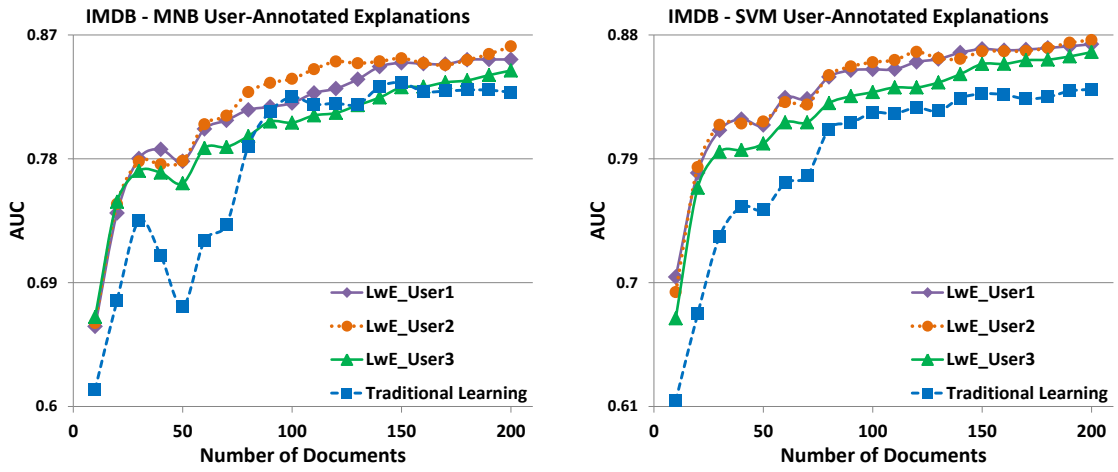



Figure 6.12. Graphical user interface for the Learning with Explanations framework.

Figure 6.13 shows the results comparing LwE with explanations provided by the

three users to traditional learning (TL) using multinomial naïve Bayes and support vector machines. LwE with default weight settings for the hyper-parameters (i.e., $w_s = 100$, $w_o = 100$, $w_{d'}^{y} = 1$, and $w_{d'}^{\neg y} = 0.01$ using multinomial naïve Bayes, and $w_s = 1$, $w_o = 1$, $w_{d'}^{y} = 0.1$, and $w_{d'}^{\neg y} = 0.001$ using support vector machines) performed better than TL. However, real user can be fallible or reluctant, and can provide noisy explanations. Hence the weights, $w_s$, $w_o$, $w_{d'}^{y}$, and $w_{d'}^{\neg y}$, need to reflect the confidence in the expert. If the expert is perfect, placing higher weights on $w_s$ and $w_o$ compared to $w_{d'}^{y}$ and $w_{d'}^{\neg y}$ works better, and if the expert is noisy, the weights $w_s$ and $w_o$ should be closer to $w_{d'}^{y}$ and $w_{d'}^{\neg y}$. Ideally, these parameters should be tuned (e.g., using cross-validation) at each iteration of learning.




(a) (b)

Figure 6.13. LwE with explanations and labels provided by the three users using (a) multinomial naïve Bayes and (b) support vector machines. LwE with real user-annotated explanations provides improvements over traditional learning (TL).

These results show that our framework can effectively incorporate user-annotated explanations to improve learning. Table 6.4 presents the average number of explanations, the average number of words per explanations, the average accuracy of users based on the actual labels of the movie reviews, and the average time that each user took to annotate 200 movie reviews. The average number of explanations provided by the users ranged between ≈2.5 and ≈11 explanations per document. Moreover, each phrase that the user highlighted

Table 6.4. Average number of explanations, average number of words per explanations, accuracy, and average time taken by three users to annotate 200 movie reviews.

| User | Average # of explanations per document | Average # of words per explanation | Accuracy of users on labels | Average time (Sec) |
|---|---|---|---|---|
| **User1** | 10.54 | 3.85 | 96% | 121.9 |
| **User2** | 5.76 | 1.56 | 92.50% | 88.31 |
| **User3** | 2.5 | 2.21 | 92% | 84.48 |

as an explanation often consisted of more than one word. We note that incorporating explanations improves learning, but providing explanations and the label would take more time than providing just the label for a document. We do not know exactly how long the users would have taken to provide just the labels for these 200 documents, but annotating these 200 documents with the labels *and* providing explanations took the users more than 1 minute per document.

## 6.6 Conclusion

We introduced a novel framework to enrich the interaction between the human and active learner by asking the human experts to annotate documents and provide explanations for classification, by highlighting features that support or oppose their classification of documents. Our explanations framework can effectively incorporate the rich feedback, in the form of feature-based explanations, into any off-the-shelf classifier. The empirical evaluations on three text datasets and two classifiers showed that our proposed method can effectively incorporate simple explanations for document classification. We showed that our framework performs well, even when the explanations are noisy, and when the expert is fallible or reluctant in providing explanations. We presented a graphical user interface to elicit explanations from real users and showed that our framework is effective for incorporating explanations from user-annotated explanations, which could be noisy.

# CHAPTER 7

# CONCLUSION AND FUTURE RESEARCH DIRECTIONS

In this dissertation, I discussed the problem of making the supervision resource-efficient through use of active learning and enabling richer interactions between the human expert and learner to make more effective and intuitive use of human expert's time, cost, and effort in providing supervision. In this section, I first summarize our contributions, present future research directions, and then conclude the thesis.

## 7.1 Summary of Contributions

In this thesis, I introduced four novel active learning frameworks to enrich the interactions between the expert and the learner. We addressed two main challenges in this thesis. The first challenge is that the active learners do not provide their reasons for selecting certain instances for labeling. To address this challenge, we enabled the active learner to reveal its perception of uncertainty on instances. The second challenge is that the traditional supervised learning approaches can incorporate labeled examples, but they cannot readily handle the rich feedback, such as domain knowledge, feature annotation, and rationales and explanations for classification. This is a fundamental limitation of the machine learning algorithms, and to overcome this challenge, we devised several approaches that can effectively incorporate rich feedback into the training of supervised learning algorithms.

I described an evidence-based framework to provide transparency into uncertainty sampling, an active learning strategy that selects instances about which the model is uncertain. We discovered two reasons for model's perception of uncertainty: a model can be uncertain about an instance due to strong and conflicting evidence for both classes (conflicting-evidence uncertainty) versus a model can be uncertain because it does not have sufficient evidence for either class (insufficient-evidence uncertainty). Through empirical

evaluations on several real-word datasets, I showed that annotating conflicting cases provides huge improvements to the model's performance. I provided analytical and empirical justifications which show that annotating conflicting-evidence cases is beneficial to the learning, because conflicting-evidence instances are underrepresented in the labeled data compared to the insufficient-evidence cases, and that the model has higher variance on the conflicting-evidence cases than on the insufficient-evidence cases.

I described three frameworks to enable richer interactions between the human expert and learner and incorporate the rich feedback into the training of supervised learning algorithms. The first framework to incorporate rich feedback aimed at incorporating rationales for document classification task, where rationales are phrases in a document that convinced the human expert to choose a particular label for the document. I presented a classifier-free approach that modifies the training data by weighting the rationale features higher than other features in the documents. I empirically showed on four text classification datasets and using three classifiers that our approach to incorporate rationales into learning significantly outperformed traditional learning that uses only labeled examples. I showed that our approach was on par with several other classifier-specific approaches, but our approach has the advantage of being independent of the classifier.

The second framework to incorporate rich feedback was developed for the aviation domain, where subject matter experts looked at flights data and identified a few flights that are of operational significance, i.e., represent a safety concern, and provided rationales for why a flight was operationally significant. The flights data is heterogeneous, where some features are time series data, while others are binary, discrete or continuous, and we utilized multiple kernel learning approach, which builds a kernel for each feature, to combine the data from heterogeneous data sources. Here, the rationales provided by subject matter experts were either based on a single feature or based on conjunction of two or more features. I presented our approach that creates new features based on the rationales

provided by subject matter experts, and weights the kernels for rationale features higher than the kernels for other features. Through empirical evaluations, we showed that our approach improves the detection of operationally significant flights by as much as 75% compared to the state-of-the-art, and provides significant reduction in time for supervision.

The third framework to incorporate rich feedback aimed at eliciting explanations for classification of documents, where experts provided explanations by highlighting phrases that reinforce their belief in the document's label and striking-out phrases that weaken their belief in the document's label. For this framework, I presented a graphical user interface to facilitate humans to read documents and highlight phrases that strengthen (or weaken) their belief in the chosen label. I presented our framework that effectively utilizes the explanations to improve learning by creating various pseudo-documents for the highlighted phrases and the remaining phrases, and weighting the highlighted phrases higher than the other phrases that were not selected by the expert. Through empirical evaluations on three document classification tasks and a user study, I showed that our approach to incorporate explanations into learning provided significant improvements in learning compared to traditional learning and the learning with rationales framework.

## 7.2 Future Research Directions

There are several interesting avenues for future research based on the work presented in this thesis. We discuss some of the possible future research directions in this section.

**7.2.1 Frameworks to Provide Transparency into Other Active Learning Strategies and Using Other Classifiers.** Most active learning strategies are opaque and do not provide their reasons for selecting instances for labeling. Most active learning strategies use greedy algorithms to select high-utility instances based on some heuristics, that is, instances which the active learner "thinks" are important for labeling, and hence, it is not

trivial to make the active learners transparent to provide useful explanations. In Chapter 3, we showed that making the uncertainty sampling strategy transparent to explain its reasons for uncertainty on instances provides huge improvements to the learning efficiency. This result suggests that exploring methods to provide transparency into other active learning strategies, such as query-by-committee [69], expected error reduction [59], [88], and density-weighted methods [95], could be useful to both researchers and labelers for (i) understanding the reasons for why certain instances are queried by the active learning strategies, and (ii) devising better active learning strategies to select more useful instances for labeling.

In Chapter 3, we presented the evidence-based framework and provided formulations of evidence for several off-the-shelf classifiers such as naïve Bayes, logistic regression, and linear and non-linear support vector machines, however, formulating evidences for other classifiers such as neural networks and perceptrons, and for regression methods, is not trivial, but is an interesting direction for future research.

**7.2.2 Evidence-Based Framework and Outliers.** In Chapter 3, we showed that instances with conflicting-evidence have less density in the training data and instances with insufficient-evidence have higher density in the training data. This result suggests an interesting idea of utilizing the evidence-based framework for identifying outliers. It is yet to be determined whether the conflicting cases can be deemed as outliers with respect to the overall data.

Uncertainty sampling is known to be susceptible to outliers [88]. It is not clear at this point whether combining uncertainty sampling with the evidence-based framework makes it more or less susceptible to noise and outliers. We experimented with real-world datasets, which are expected to be noisy, and showed that most-surely uncertain significantly outperforms uncertainty sampling while least-surely uncertain performed significantly worse on many measures and datasets. The effect of noise and outliers on UNC-CE

and UNC-IE needs to be verified through carefully designed experiments with synthetic datasets. A possible idea is to combine density-weighted methods [95] with the evidence-based framework, where instances with less density in the training data *and* more density in the overall data are selected for labeling.

Another interesting idea is to utilize the formalism of Senge et al. [2014] to investigate whether any parallels and similarities can be drawn between conflicting versus insufficient-evidence and aleatoric versus epistemic uncertainty cases.

**7.2.3 Eliciting Richer Feedback from the Labelers.** An exciting future research direction is to allow the labelers to provide richer feedback. This is especially useful for resolving conflicts that stem from seemingly conflicting words and phrases. For example, for the movie review "The plot was great, but the performance of the actors was terrible. Avoid it." the positive word "great" is at odds with the negative words "terrible" and "avoid". If the labeler is allowed to provide richer feedback, stating that the word "great" refers to the plot, "terrible" refers to the performance, and "avoid" refers to the movie, then the learner might be able to learn to resolve similar conflicts in other documents. However, this requires a conflict resolution mechanism in which the labeler can provide rich feedback *and* a learner that can utilize such rich feedback.

We showed that our strategy to incorporate rationales works well for text classification. The proposed framework can potentially be used for non-text domains where the domain experts can provide rationales for their decisions, such as medical domain where the doctor can provide a rationale for his/her diagnosis and treatment decisions. In our framework, we place higher weights on rationales and lower weights on other features, thus our approach can be applied to domains where features represent presence/frequencies of characteristics, such as whether a patient is infant/young/old, whether the cholesterol level is low/medium/high, etc. Each domain is expected to have its own unique research challenges and working with other domains is another interesting future research direction.

Another line of future work is to allow the labelers to provide other types of explanations, where explanations can be complex conjunction or disjunction of domain-specific features, or free-form text entries. Incorporating unstructured domain knowledge, such as free-form text entries, into learning would require parsing and converting the domain knowledge into the representation that the underlying model can understand or operate on.

Future work for the rationales framework presented in Chapter 6 includes allowing the experts to provide richer rationales and the enabling the system to integrate multiple data sources for supporting those rationales for increased coverage of a wider range of operationally significant anomalies.

**7.2.4 Explanations Framework for Multilabel and Multiclass classification.** In this thesis, we presented the explanations framework for binary classification, where users highlight supporting and opposing phrases as explanations for labeling documents. It is easy to incorporate explanations for binary classification using our explanations framework, however, it is not straightforward to incorporate explanations for multiclass and multilabel classification using our framework. Multiclass classification makes the assumption that each instance belongs to only one class, whereas, in multilabel classification, an instance can be assigned to multiple classes. Another interesting future research direction is to generalize the explanations framework for incorporating explanations for multiclass and multilabel classification.

## 7.3 Conclusion

In this thesis, I introduced four novel active learning frameworks that enable rich interactions between the human expert and learner and make the supervision resource-efficient. I described how we make the active learner transparent to explain its perception of uncertainty on instances and how we used it for selecting better instances for labeling. I introduced three frameworks to elicit rich feedback from human experts and described how

we incorporated rich feedback into traditional machine learning algorithms. I showed that enabling rich interactions between the human and learner and incorporating rich feedback into learning makes more intuitive and effective use of human's time and effort in providing supervision.

With the rise in the use of predictive analytics to enhance businesses, provide decision support, and enhance user experience, more and more organizations are turning to predictive modeling for solutions to unlock the power of data for a variety of uses. However, building predictive models often requires supervision, and hence there is a need to make more intelligent use of human's time and effort. Even more important is to devise capabilities to interact with the machine learning systems to (i) understand why the system makes certain predictions, (ii) teach the system when it makes an incorrect prediction, and (iii) make the users' interactions with machine learning systems more enjoyable. Hence, it is becoming ever increasingly important to enrich the interactions between the humans and machine learning systems.

# APPENDIX A

# LICENSES AND PERMISSION TO REUSE MATERIAL FROM PUBLICATIONS IN THIS THESIS

LICENSES TO REUSE MATERIAL FROM PUBLICATIONS IN THIS THESIS

- The material from the following article has been included in this thesis "With permission of Springer": "Evidence-based uncertainty sampling for active learning", Volume 31, Issue 1, 2016, pp 164202, Manali Sharma and Mustafa Bilgic, In Data Mining and Knowledge Discovery

- The material from the following paper has been included in this thesis "With permission of Springer": "Active Learning with Rationales for Identifying Operationally Significant Anomalies in Aviation", 2016, pp 209-225, Manali Sharma, Kamalika Das, Mustafa Bilgic, Bryan Matthews, David Nielsen, and Nikunj Oza, In Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2016, Riva del Garda, Italy, September 19-23, 2016, Proceedings, Part III

# PERMISSION TO REUSE MATERIAL FROM PUBLICATION IN THIS THESIS

National Aeronautics and

Space Administration

**Ames Research Center**

Moffett Field, California 94035-1000

Date: April 21, 2017

From: Nikunj C. Oza, NASA Ames Research Center

To: Manali Sharma, Illinois Institute of Technology

This letter is to confirm that Manali Sharma has permission to include material from the following paper in her Ph.D. Thesis/Dissertation: Manali Sharma, Kamalika Das, Mustafa Bilgic, Bryan Matthews, David Nielsen, Nikunj Oza. "Active Learning with Rationales for Identifying Operationally Significant Anomalies in Aviation" In *The European Conference on Machine Learning and Principles and Practice of Knowledge Discovery*, 2016, vol. 9853, pp. 209–225.

This paper has undergone the required export control review that is required for the public release of this paper.

Sincerely,

Nikunj C. Oza
Leader, Data Sciences Group
NASA Ames Research Center

# BIBLIOGRAPHY


[1] N. Abe and H. Mamitsuka, "Query learning strategies using boosting and bagging," in *Proceedings of the Fifteenth International Conference on Machine Learning*, 1998, pp. 1–9.

[2] D. Angluin, "Queries and concept learning," *Machine Learning*, vol. 2, no. 4, pp. 319–342, 1988.

[3] J. A. Aslam, E. Kanoulas, V. Pavlu, S. Savev, and E. Yilmaz, "Document selection methodologies for efficient and effective learning-to-rank," in *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, ser. SIGIR '09, 2009, pp. 468–475.

[4] J. Attenberg, P. Melville, and F. Provost, "A unified approach to active dual supervision for labeling features and examples," in *European conference on Machine learning and knowledge discovery in databases*, 2010, pp. 40–55.

[5] F. Bach, G. Lanckriet, and M. Jordan, "Multiple kernel learning, conic duality, and the smo algorithm," in *International Conference on Machine Learning*, 2004.

[6] K. Bharat and M. R. Henzinger, "Improved algorithms for topic distillation in a hyperlinked environment," in *ACM SIGIR*. ACM, 1998, pp. 104–111.

[7] M. Bilgic and P. N. Bennett, "Active query selection for learning rankers," in *ACM SIGIR*, August 2012.

[8] M. Bilgic and L. Getoor, "Active inference for collective classification," in *Twenty-Fourth Conference on Artificial Intelligence (AAAI NECTAR Track)*, 2010.

[9] ——, "Reflect and correct: A misclassification prediction approach to active inference," *ACM Transactions on Knowledge Discovery from Data*, vol. 3, no. 4, pp. 1–32, 2009.

[10] M. Bilgic, L. Mihalkova, and L. Getoor, "Active learning for networked data," in *Proceedings of the 27th International Conference on Machine Learning*, 2010, pp. 79–86.

[11] L. Breiman, "Bagging predictors," *Machine Learning*, vol. 24, no. 2, pp. 123–140, 1996.

[12] B. G. Buchanan, E. H. Shortliffe *et al.*, *Rule-based expert systems*. Addison-Wesley Reading, MA, 1984, vol. 3.

[13] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM computing surveys (CSUR)*, vol. 41, no. 3, p. 15, 2009.

[14] C. Chao, M. Cakmak, and A. L. Thomaz, "Transparent active learning for robots," in *5th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. IEEE, 2010, pp. 317–324.

[15] O. Chapelle, B. Schölkopf, and A. Zien, Eds., *Semi-Supervised Learning*. Cambridge, MA: MIT Press, 2006.

[16] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: Synthetic minority over-sampling technique," *Journal of Artificial Intelligence Research*, pp. 321–357, 2002.

[17] W. S. Cleveland and S. J. Devlin, "Locally weighted regression: an approach to regression analysis by local fitting," *Journal of the American Statistical Association*, vol. 83, no. 403, pp. 596–610, 1988.

[18] D. Cohn, L. Atlas, and R. Ladner, "Improving generalization with active learning," *Machine Learning*, vol. 15, no. 2, pp. 201–221, 1994.

[19] D. A. Cohn, "Minimizing statistical bias with queries," in *Advances in Neural Information Processing Systems*, 1997, pp. 417–423.

[20] ——, "Neural network exploration using optimal experiment design," *Neural Networks*, vol. 9, no. 6, pp. 1071–1083, 1996.

[21] D. A. Cohn, Z. Ghahramani, and M. I. Jordan, "Active learning with statistical models," *Journal of Artificial Intelligence Research*, vol. 4, pp. 129–145, 1996.

[22] I. Dagan and S. P. Engelson, "Committee-based sampling for training probabilistic classifiers," in *Proceedings of the Twelfth International Conference on Machine Learning*, 1995, pp. 150–157.

[23] X. H. Dang, I. Assent, R. Ng, A. Zimek, and E. Schubert, "Discriminative features for identifying and interpreting outliers," in *ICDE*, 2014, pp. 88–99.

[24] S. Das, B. L. Matthews, A. N. Srivastava, and N. C. Oza, "Multiple kernel learning for heterogeneous anomaly detection: algorithm and aviation safety case study," in *Proceedings of KDD*, 2010, pp. 47–56.

[25] S. Das, T. Moore, W.-K. Wong, S. Stumpf, I. Oberst, K. McIntosh, and M. Burnett, "End-user feature labeling: Supervised and semi-supervised approaches based on locally-weighted logistic regression," *Artificial Intelligence*, vol. 204, pp. 56–74, 2013.

[26] G. DeJong and R. Mooney, "Explanation-based learning: An alternative view," *Machine learning*, vol. 1, no. 2, pp. 145–176, 1986.

[27] A. Dix, *Human-Computer Interaction*. Boston, MA: Springer US, 2009, pp. 1327–1331. [Online]. Available: http://dx.doi.org/10.1007/978-0-387-39940-9_192

[28] J. Donahue and K. Grauman, "Annotator rationales for visual recognition," in *ICCV*. IEEE, 2011, pp. 1395–1402.

[29] P. Donmez and J. G. Carbonell, "Proactive learning: cost-sensitive active learning with multiple imperfect oracles," in *Proceedings of the 17th ACM conference on Information and knowledge management*. ACM, 2008, pp. 619–628.

[30] P. Donmez, J. G. Carbonell, and P. N. Bennett, "Dual strategy active learning," in *Machine Learning: ECML 2007*. Springer, 2007, pp. 116–127.

[31] G. Druck, B. Settles, and A. McCallum, "Active learning by labeling features," in *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1-Volume 1*. Association for Computational Linguistics, 2009, pp. 81–90.

[32] T. Ellman, "Explanation-based learning: A survey of programs and perspectives," *ACM Computing Surveys (CSUR)*, vol. 21, no. 2, pp. 163–221, 1989.

[33] J. A. Fails and D. R. Olsen, Jr., "Interactive machine learning," in *Proceedings of the 8th International Conference on Intelligent User Interfaces*, ser. IUI '03. New York, NY, USA: ACM, 2003, pp. 39–45. [Online]. Available: http://doi.acm.org/10.1145/604045.604056

[34] T. Fawcett, "An introduction to roc analysis," *Pattern recognition letters*, vol. 27, no. 8, pp. 861–874, 2006.

[35] J. Fogarty, D. Tan, A. Kapoor, and S. Winder, "Cueflik: interactive concept learning in image search," in *Annual SIGCHI conference on Human factors in computing systems*, New York, NY, USA, 2008, pp. 29–38.

[36] A. Frank and A. Asuncion, "UCI machine learning repository," 2010. [Online]. Available: http://archive.ics.uci.edu/ml

[37] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *Journal of Computer and System Sciences*, vol. 55, no. 1, pp. 119–139, 1997.

[38] P. W. Frey and D. J. Slate, "Letter recognition using holland-style adaptive classifiers," *Machine Learning*, vol. 6, no. 2, pp. 161–182, 1991.

[39] G. M. Fung, O. L. Mangasarian, and J. W. Shavlik, "Knowledge-based support vector machine classifiers," in *Advances in neural information processing systems*, 2002, pp. 521–528.

[40] F. Girosi and N. T. Chan, "Prior knowledge and the creation of virtual examples for rbf networks," in *Neural Networks for Signal Processing [1995] V. Proceedings of the 1995 IEEE Workshop*. IEEE, 1995, pp. 201–210.

[41] N. Görnitz, M. Kloft, K. Rieck, and U. Brefeld, "Active learning for network intrusion detection," in *Proceedings of the 2nd ACM workshop on Security and artificial intelligence*. ACM, 2009, pp. 47–54.

[42] Q. Gu, T. Zhang, J. Han, and C. H. Ding, "Selective labeling via error bound minimization," in *Advances in Neural Information Processing Systems*, 2012, pp. 323–331.

[43] Q. Gu, T. Zhang, and J. Han, "Batch-mode active learning via error bound minimization," in *Proceedings of the Thirtieth Conference Annual Conference on Uncertainty in Artificial Intelligence (UAI-14)*. Corvallis, Oregon: AUAI Press, 2014, pp. 300–309.

[44] I. Guyon, "Results of active learning challenge," 2011.

[45] I. Guyon *et al.*, "Datasets of the active learning challenge," *Journal of Machine Learning Research*, 2011.

[46] Z. He, X. Xu, and S. Deng, "Discovering cluster-based local outliers," *Pattern Recognition Letters*, vol. 24, no. 9, pp. 1641–1650, 2003.

[47] S. C. Hoi, R. Jin, and M. R. Lyu, "Large-scale text categorization by batch mode active learning," in *Proceedings of the 15th International Conference on World Wide Web*. ACM, 2006, pp. 633–642.

[48] S. C. Hoi, R. Jin, J. Zhu, and M. R. Lyu, "Batch mode active learning and its application to medical image classification," in *Proceedings of the 23rd International Conference on Machine learning*. ACM, 2006, pp. 417–424.

[49] J. A. Jacko, *Human computer interaction handbook: Fundamentals, evolving technologies, and emerging applications*. CRC press, 2012.

[50] M. Kabra, A. A. Robie, M. Rivera-Alba, S. Branson, and K. Branson, "Jaaba: interactive machine learning for automatic annotation of animal behavior," *nature methods*, vol. 10, no. 1, pp. 64–67, 2013.

[51] A. Kapoor, E. Horvitz, and S. Basu, "Selective supervision: Guiding supervised learning with decision-theoretic active learning." in *IJCAI*, vol. 7, 2007, pp. 877–882.

[52] C. Komurlu and M. Bilgic, "Active inference and dynamic gaussian bayesian networks for battery optimization in wireless sensor networks," in *Proceedings of AAAI Workshop on Artificial Intelligence for Smart Grids and Smart Buildings*, 2016. [Online]. Available: http://www.cs.iit.edu/~ml/pdfs/komurlu-aisgsb16.pdf

[53] C. Komurlu, J. Shao, and M. Bilgic, "Dynamic bayesian network modeling of vascularization in engineered tissues," in *Proceedings of the Eleventh UAI Bayesian Modeling Applications Workshop*, 2014. [Online]. Available: http://www.cs.iit.edu/~ml/pdfs/komurlu-bmaw14.pdf

[54] C. Komurlu, J. Shao, B. Akar, E. S. Bayrak, E. M. Brey, A. Cinar, and M. Bilgic, "Active inference for dynamic bayesian networks with an application to tissue engineering," *Knowledge and Information Systems*, pp. 1–27, 2016. [Online]. Available: http://www.cs.iit.edu/~ml/pdfs/komurlu-kais16.pdf

[55] M. Kumar, R. Ghani, and Z.-S. Mei, "Data mining to predict and prevent errors in health insurance claims processing," in *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, New York, NY, USA, 2010, pp. 65–74.

[56] J. E. Laird, P. S. Rosenbloom, and A. Newell, "Chunking in soar: The anatomy of a general learning mechanism," *Machine learning*, vol. 1, no. 1, pp. 11–46, 1986.

[57] D. D. Lewis and W. A. Gale, "A sequential algorithm for training text classifiers," in *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. Springer-Verlag New York, Inc., 1994, pp. 3–12.

[58] D. Lewis and J. Catlett, "Heterogeneous uncertainty sampling for supervised learning," in *Proceedings of the eleventh international conference on machine learning*, 1994, pp. 148–156.

[59] M. Lindenbaum, S. Markovich, D. Rusakov *et al.*, "Selective sampling for nearest neighbor classifiers," in *Proceedings of The National Conference on Artificial Intelligence*, 1999, pp. 366–371.

[60] M. Lindenbaum, S. Markovitch, and D. Rusakov, "Selective sampling for nearest neighbor classifiers," *Machine Learning*, vol. 54, no. 2, pp. 125–152, 2004.

[61] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts, "Learning word vectors for sentiment analysis," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*. Association for Computational Linguistics, 2011, pp. 142–150.

[62] D. J. MacKay, "Information-based objective functions for active data selection," *Neural Computation*, vol. 4, no. 4, pp. 590–604, 1992.

[63] A. McCallum and K. Nigam, "Employing EM and pool-based active learning for text classification," in *International Conference on Machine Learning*, 1998, pp. 350–358.

[64] A. McCallum, K. Nigam *et al.*, "A comparison of event models for naive bayes text classification," in *AAAI-98 Workshop on Learning for Text Categorization*, vol. 752. Citeseer, 1998, pp. 41–48.

[65] P. Melville and R. J. Mooney, "Diverse ensembles for active learning," in *Proceedings of the Twenty-first International Conference on Machine Learning*, 2004, pp. 74–.

[66] P. Melville and V. Sindhwani, "Active dual supervision: Reducing the cost of annotating examples and features," in *Proceedings of the NAACL HLT 2009 Workshop on Active Learning for Natural Language Processing*. Association for Computational Linguistics, 2009, pp. 49–57.

[67] P. Melville, W. Gryc, and R. D. Lawrence, "Sentiment analysis of blogs by combining lexical knowledge with text classification," in *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2009, pp. 1275–1284.

[68] B. Micenkova, X.-H. Dang, I. Assent, and R. Ng, "Explaining outliers by subspace separability," in *Data Mining (ICDM), 2013 IEEE 13th International Conference on*, 2013, pp. 518–527.

[69] T. M. Mitchell, "Generalization as search," *Artificial Intelligence*, vol. 18, no. 2, pp. 203–226, 1982.

[70] T. M. Mitchell, R. M. Keller, and S. T. Kedar-Cabelli, "Explanation-based generalization: A unifying view," *Machine learning*, vol. 1, no. 1, pp. 47–80, 1986.

[71] R. J. Mooney, "Generalizing explanations of narratives into schemata," in *Machine Learning*. Springer, 1986, pp. 207–212.

[72] National Research Council, *Advancing Aeronautical Safety: A Review of NASA's Aviation Safety-Related Research Programs*. Washington DC: The National Academies Press, 2010.

[73] H. T. Nguyen and A. Smeulders, "Active learning using pre-clustering," in *Proceedings of the Twenty-first International Conference on Machine learning*. ACM, 2004, p. 79.

[74] R. K. Pace and R. Barry, "Sparse spatial autoregressions," *Statistics & Probability Letters*, vol. 33, no. 3, pp. 291–297, 1997.

[75] D. Parikh and K. Grauman, "Relative attributes," in *Computer Vision (ICCV), 2011 IEEE International Conference on*. IEEE, 2011, pp. 503–510.

[76] A. Parkash and D. Parikh, "Attributes for classifier feedback," in *Computer Vision–ECCV 2012*. Springer, 2012, pp. 354–368.

[77] D. Pechyony and V. Vapnik, "On the theory of learnining with privileged information," in *Advances in neural information processing systems*, 2010, pp. 1894–1902.

[78] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[79] D. Pelleg and A. Moore, "Active learning for anomaly and rare-category detection," in *NIPS*, December 2004.

[80] K. Pichara and A. Soto, "Active learning and subspace clustering for anomaly detection," *Intell. Data Anal.*, vol. 15, no. 2, pp. 151–171, Apr. 2011.

[81] H. Raghavan and J. Allan, "An interactive algorithm for asking and incorporating feature feedback into support vector machines," in *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 2007, pp. 79–86.

[82] H. Raghavan, O. Madani, and R. Jones, "Active learning with feedback on features and instances," *Journal of Machine Learning Research*, vol. 7, pp. 1655–1686, 2006.

[83] M. E. Ramirez-Loaiza, "Anytime active learning," Ph.D. dissertation, Illinois Institute of Technology, 2016.

[84] M. E. Ramirez-Loaiza, A. Culotta, and M. Bilgic, "Towards Anytime Active Learning: Interrupting Experts to Reduce Annotation Costs," in *ACM SIGKDD Workshop on Interactive Data Exploration and Analytics (IDEA'13)*, 2013.

[85] ——, "Anytime Active Learning," in *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, 2014, pp. 2048–2054.

[86] M. E. Ramirez-Loaiza, M. Sharma, G. Kumar, and M. Bilgic, "Active learning: an empirical study of common baselines," *Data Mining and Knowledge Discovery*, pp. 1–27, 2016.

[87] M. Rattigan, M. Maier, and D. Jensen, "Exploiting network structure for active inference in collective classification," in *ICDM Workshop on Mining Graphs and Complex Structures*, 2007, pp. 429–434.

[88] N. Roy and A. McCallum, "Toward optimal active learning through sampling estimation of error reduction," in *Proceedings of the Eighteenth International Conference on Machine Learning*, ser. ICML '01. Morgan Kaufmann Publishers Inc., 2001, pp. 441–448.

[89] A. I. Schein and L. H. Ungar, "Active learning for logistic regression: an evaluation," *Machine Learning*, vol. 68, no. 3, pp. 235–265, 2007.

[90] B. Schölkopf, J. C. Platt, J. C. Shawe-Taylor, A. J. Smola, and R. C. Williamson, "Estimating the support of a high-dimensional distribution," *Neural Computation*, vol. 13, no. 7, pp. 1443–1471, 2001.

[91] D. Sculley, "Online active learning methods for fast label-efficient spam filtering," in *Fourth Conference on Email and Anti-Spam (CEAS)*, 2007.

[92] R. Segal, T. Markowitz, and W. Arnold, "Fast uncertainty sampling for labeling large e-mail corpora," in *Third Conference on Email and Anti-Spam (CEAS)*, 2006.

[93] R. Senge, S. Bösner, K. Dembczyński, J. Haasenritter, O. Hirsch, N. Donner-Banzhoff, and E. Hüllermeier, "Reliable classification: Learning classifiers that distinguish aleatoric and epistemic uncertainty," *Information Sciences*, vol. 255, pp. 16–29, 2014.

[94] B. Settles, "Active learning," *Synthesis Lectures on Artificial Intelligence and Machine Learning*, vol. 6, no. 1, pp. 1–114, 2012.

[95] B. Settles and M. Craven, "An analysis of active learning strategies for sequence labeling tasks," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2008, pp. 1070–1079.

[96] B. Settles, M. Craven, and L. Friedland, "Active learning with real annotation costs," in *Proceedings of the NIPS workshop on cost-sensitive learning*, 2008, pp. 1–10.

[97] H. S. Seung, M. Opper, and H. Sompolinsky, "Query by committee," in *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*. ACM, 1992, pp. 287–294.

[98] M. Sharma and M. Bilgic, "Evidence-based uncertainty sampling for active learning," *Data Mining and Knowledge Discovery*, pp. 1–39, 2016.

[99] ——, "Most-surely vs. least-surely uncertain," in *IEEE 13th International Conference on Data Mining (ICDM)*, 2013, pp. 667–676.

[100] ——, "Towards learning with feature-based explanations for document classification," in *IJCAI Workshop on BeyondLabeler - Human is More than a Labeler*, 2016, pp. 1–7. [Online]. Available: http://users.sussex.ac.uk/~nq28/beyondlabeler/ShaBil16.pdf

[101] M. Sharma, D. Zhuang, and M. Bilgic, "Active learning with rationales for text classification," in *North American Chapter of the Association for Computational Linguistics Human Language Technologies*, 2015, pp. 441–451.

[102] M. Sharma, K. Das, M. Bilgic, B. Matthews, D. Nielsen, and N. Oza, "Active learning with rationales for identifying operationally significant anomalies in aviation," in *The European Conference on Machine Learning and Principles and Practice of Knowledge Discovery (ECMLPKDD*. Springer, 2016, pp. 1–17.

[103] V. Sharmanska, N. Quadrianto, and C. H. Lampert, "Learning to rank using privileged information," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 825–832.

[104] V. Sindhwani, P. Melville, and R. D. Lawrence, "Uncertainty sampling and transductive experimental design for active dual supervision," in *Proceedings of the 26th Annual International Conference on Machine Learning*. ACM, 2009, pp. 953–960.

[105] K. Small, B. Wallace, T. Trikalinos, and C. E. Brodley, "The constrained weight space svm: learning with ranked features," in *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, 2011, pp. 865–872.

[106] S. Smith-Atakan, *Human-computer interaction*. Cengage Learning EMEA, 2006.

[107] S. Stumpf, E. Sullivan, E. Fitzhenry, I. Oberst, W. Wong, and M. Burnett, "Integrating rich user feedback into intelligent user interfaces," in *Proceedings of the 13th international conference on Intelligent user interfaces*. ACM, 2008, pp. 50–59.

[108] S. Stumpf, V. Rajaram, L. Li, M. Burnett, T. Dietterich, E. Sullivan, R. Drummond, and J. Herlocker, "Toward harnessing user feedback for machine learning," in *Proceedings of the 12th international conference on Intelligent user interfaces*. ACM, 2007, pp. 82–91.

[109] S. Stumpf, V. Rajaram, L. Li, W.-K. Wong, M. Burnett, T. Dietterich, E. Sullivan, and J. Herlocker, "Interacting meaningfully with machine learning systems: Three experiments," *International Journal of Human-Computer Studies*, vol. 67, no. 8, pp. 639–662, Aug. 2009.

[110] J. Sullivan, , J. W. Sullivan, S. W. T. (editors, P. Luff, R. Xerox, and C. Europarc, "Intelligent user interfaces," 1994.

[111] P.-N. Tan *et al.*, *Introduction to data mining*. Pearson Education India, 2006.

[112] C. A. Thompson, M. E. Califf, and R. J. Mooney, "Active learning for natural language parsing and information extraction," in *Proceedings of the Sixteenth International Conference on Machine Learning*, 1999, pp. 406–414.

[113] K. Tomanek and U. Hahn, "A comparison of models for cost-sensitive active learning," in *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*. Association for Computational Linguistics, 2010, pp. 1247–1255.

[114] S. Tong and E. Chang, "Support vector machine active learning for image retrieval," in *Proceedings of the Ninth ACM International Conference on Multimedia*. ACM, 2001, pp. 107–118.

[115] S. Tong and D. Koller, "Support vector machine active learning with applications to text classification," *Journal of Machine Learning Research*, vol. 2, pp. 45–66, 2001.

[116] G. G. Towell and J. W. Shavlik, "Knowledge-based artificial neural networks," *Artificial intelligence*, vol. 70, no. 1, pp. 119–165, 1994.

[117] G. G. Towell, J. W. Shavlik, and M. Noordewier, "Refinement of approximate domain theories by knowledge-based neural networks," in *Proceedings of the eighth National conference on Artificial intelligence*. Boston, MA, 1990, pp. 861–866.

[118] V. Vapnik and A. Vashist, "A new learning paradigm: Learning using privileged information," *Neural Networks*, vol. 22, no. 5, pp. 544–557, 2009.

[119] M. Ware, E. Frank, G. Holmes, M. Hall, and I. H. Witten, "Interactive machine learning: letting users build classifiers," *International Journal of Human-Computer Studies*, vol. 55, no. 3, pp. 281–292, 2001.

[120] Z. Xu, K. Yu, V. Tresp, X. Xu, and J. Wang, "Representative sampling for text classification using support vector machines," in *Advances in Information Retrieval, Lecture Notes in Computer Science - Volume 2633, 2003*, 2003, pp. 393–407.

[121] C.-N. J. Yu and T. Joachims, "Learning structural svms with latent variables," in *Proceedings of the 26th annual international conference on machine learning*. ACM, 2009, pp. 1169–1176.

[122] K. Yu, J. Bi, and V. Tresp, "Active learning via transductive experimental design," in *Proceedings of the 23rd International Conference on Machine Learning*. ACM, 2006, pp. 1081–1088.

[123] O. Zaidan, J. Eisner, and C. D. Piatko, "Using" annotator rationales" to improve machine learning for text categorization." in *HLT-NAACL*, 2007, pp. 260–267.

[124] O. F. Zaidan, J. Eisner, and C. Piatko, "Machine learning with annotator rationales to reduce annotation cost," in *Proceedings of the NIPS* 2008 Workshop on Cost Sensitive Learning*, 2008.

[125] C. Zhang and T. Chen, "An active learning framework for content-based information retrieval," *IEEE Transactions on Multimedia*, vol. 4, no. 2, pp. 260–268, 2002.

[126] X. Zhang, D. Ma, L. Gan, S. Jiang, and G. Agam, "Cgmos: Certainty guided minority oversampling," in *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*. ACM, 2016, pp. –.

[127] Y. Zhang, I. Marshall, and B. C. Wallace, "Rationale-augmented convolutional neural networks for text classification," *arXiv preprint arXiv:1605.04469*, 2016.

[128] J. Zhu and E. Hovy, "Active learning for word sense disambiguation with methods for addressing the class imbalance problem," in *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, 2007, pp. 783–790.

[129] J. Zhu, H. Wang, T. Yao, and B. K. Tsou, "Active learning with sampling by uncertainty and density for word sense disambiguation and text classification," in *Proceedings of the 22nd International Conference on Computational Linguistics - Volume 1*, 2008, pp. 1137–1144.