Classification with Noisy Labels: "Multiple Account" Cheating Detection in Open Online Courses

by

Curtis George Northcutt

Submitted to
the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Master of Science in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2017

Author . . . .
Department of Electrical Engineering and Computer Science
May 19, 2017

Certified by . . . .
Isaac L. Chuang
Professor of Physics and Professor of Electrical Engineering and
Computer Science, Senior Associate Dean of Digital Learning
Thesis Supervisor

Accepted by . . . .
Leslie A. Kolodziejski
Professor of Electrical Engineering and Computer Science
Chair, Department Committee on Graduate Students

Classification with Noisy Labels: "Multiple Account" Cheating Detection in Open Online Courses

by

Curtis George Northcutt

Submitted to the Department of Electrical Engineering and Computer Science
on May 19, 2017, in partial fulfillment of the
requirements for the degree of
Master of Science in Electrical Engineering and Computer Science

Abstract

Massive Open Online Courses (MOOCs) have the potential to enhance socioeconomic mobility through education. Yet, the viability of this outcome largely depends on the reputation of MOOC certificates as a credible academic credential.

I describe a cheating strategy that threatens this reputation and holds the potential to render the MOOC certificate valueless. The strategy, Copying Answers using Multiple Existences Online (CAMEO), involves a user who gathers solutions to assessment questions using one or more *harvester* accounts and then submits correct answers using one or more separate *master* accounts. To estimate a *lower bound* for CAMEO prevalence among 1.9 million course participants in 115 HarvardX and MITx courses, I introduce a filter-based CAMEO detection algorithm and use a small-scale experiment to verify CAMEO use with certainty. I identify preventive strategies that can decrease CAMEO rates and show evidence of their effectiveness in science courses.

Because the CAMEO algorithm functions as a lower bound estimate, it fails to detect many CAMEO cheaters. As a novelty of this thesis, instead of improving the shortcomings of the CAMEO algorithm directly, I recognize that we can think of the CAMEO algorithm as a method for producing noisy predicted cheating labels. Then a solution to the more general problem of binary classification with noisy labels ($\tilde{P}\tilde{N}$ learning) is a solution to CAMEO cheating detection.

$\tilde{P}\tilde{N}$ learning is the problem of binary classification when training examples may be mislabeled (flipped) uniformly with noise rate ρ_1 for positive examples and ρ_0 for negative examples. I propose Rank Pruning to solve $\tilde{P}\tilde{N}$ learning and the open problem of estimating the noise rates. Unlike prior solutions, Rank Pruning is efficient and general, requiring $\mathcal{O}(T)$ for any unrestricted choice of probabilistic classifier with T fitting time. I prove Rank Pruning achieves consistent noise estimation and equivalent expected risk as learning with uncorrupted labels in ideal conditions, and derive closed-form solutions when conditions are non-ideal. Rank Pruning achieves state-of-the-art noise rate estimation and F1, error, and AUC-PR on the MNIST and CIFAR datasets, regardless of noise rates. To highlight, Rank Pruning with a CNN classifier can predict if a MNIST digit is a *one* or *not one* with only 0.25% error, and 0.46% error across all digits, even when 50% of positive examples are mislabeled and 50% of observed positive labels are mislabeled negative examples. Rank Pruning achieves similarly impressive results when as large as 50% of training examples are actually just noise drawn from a third distribution.

Together, the CAMEO and Rank Pruning algorithms allow for a robust, general, and time-efficient solution to the CAMEO cheating detection problem. By ensuring the validity of MOOC credentials, we enable MOOCs to achieve both openness and value, and thus take one step closer to the greater goal of democratization of education.

Thesis Supervisor: Isaac L. Chuang
Title: Professor of Physics and Professor of Electrical Engineering and Computer Science, Senior Associate Dean of Digital Learning

Acknowledgments

- My advisor, Ike Chuang - for encouraging me to turn my ideas into realities, innovate through exploration, and master the art of scientific communication in all forms. Ike's advice is as beneficial as it is broadly-scoped. Of all his shared wisdom, I appreciated this one the most, "Choose your metrics for accomplishments carefully. It's not research versus award. The causality is that true (and sustained) accolades follow accomplishment. Go for the real thing, and not merely transient public perception... Dig in and make yourself proud of what you've done. Earn the right to know you're one of the top 10 in the world..."

- My family - and in particular my father, Cliff Northcutt, for raising me to respect hard work over privilege, honesty over possession. He laid the groundwork for me to become a good man, mastering the juxtaposition of humor, values, and opportunity, with realism, poverty, and hardship. He helped me to avoid many mistakes in life, by vicariously sharing with me the difficulties of his own life. He raised my sisters and me for our own benefit, not for any expected appreciation. Even more than I know the content of this thesis, I know my father will never know how much I love and respect him. But the rest of my family is a large part of the person I am today. My mother, Jackie, never failed to love me with all her heart. She consistently supports my choices without hesitation. Never once has she judged me for who I am or what I've done. My sister, Grace, in many ways raised me as a surrogate mother in difficult times. Without her laying the rock of my foundation, I could not stand as strong as I do today. And my sister, Virginia, adds a unique, unscathed joy to my world, unmatched by any other. In quiet moments, I often remind myself of her time-withstanding advice, "Don't take yourself too seriously," a wisdom she discovered over a decade before I could.

- Kimberly Leon - for making coming home the best part of my day, every day.

You are a constant motivation, a wonderful partner, and the joy of my tenure at MIT.

- Andrew Ho - For never failing to provide a wonderful and refreshing perspective. Andrew raised the bar of eloquence for me, in all its forms. Without his input, the CAMEO paper would certainly have been an inferior effort. I often hear the comment that someone is "a joy to work with." Andrew defines that standard.

- Tailin Wu - For his hard work and contributions to the theoretical development of Rank Pruning and for long nights running experiments.

- HarvardX and MITx staff - for their timely feedback and continued support. In particular, many thanks to those involved with revoking certificates at MITx. This is how we turn research into impact!

- Jonas Mueller - for never hesitating to let me drill him on all varieties of machine learning and statistics questions. Jonas is a uniquely wonderful friend and brilliant researcher. Although this surely will never be read by any party considering Jonas for employment, if per chance it is, Jonas is your guy. He is as hardworking as he is capable. I wish the world for him.

- Martin Segado - for his incredible kindness and fantastic work on generating cheating labels.

- Cody Coleman - for being a friend, a labmate, a roommate, and someone who I admire, respect, and value.

- My labmates, Rich, Guang Hao, and Ted - for fun times and great discussions.

- And to all the others who shaped and influenced my life. The faculty and lecturers that I've had the opportunity to learn from at MIT. The friends who have made it fun along the way. The folks who I worked with at FAIR (Y-Lan Boureau!) and Microsoft Research in India. And the Computer Science department at Vanderbilt, who beyond teaching, inspired confidence.

Contents

List of Figures

List of Tables

List of Algorithms

Glossary and Feature Descriptions

$P(\mathbf{Beta}(.5{+}\mathbf{NSAB}, .5{+}\mathbf{N}\ \text{-}\ \mathbf{NSAB}) > 0.8)$ (Variable Type: FLOAT). The posterior Beta probability that the true proportion of CH *show answer* clicks that occurred before CM correct answer submissions is greater than 0.9. (i.e. most *show answer* clicks occur before correct answer submissions.". 1

$P(\mathbf{Beta}(.5{+}\mathbf{NSAB}, .5{+}\mathbf{N}\ \text{-}\ \mathbf{NSAB}) > 0.9)$ (Variable Type: FLOAT). The posterior Beta probability that the true proportion of CH *show answer* clicks that occurred before CM correct answer submissions is greater than 0.9. (i.e. most *show answer* clicks occur before correct answer submissions.". 1, 17

Δt $\Delta t_{m,h,c,i} = t_{m,c,i} - t_{h,c,i}$. This is the difference between the time that a master account, m, submits a correct answer and the time that a harvester account, h, acquires the correct solution, for a problem (item) in common, i, in a given MOOC course, c.. 1, 9, 10, 16, 26, 27, 35–42, 44, 54, 92

Δt distribution A $1D$ distribution of all Δts for a CM CH pair.. 1, 10, 54, 91

π The true proportion of a CH's *show answer* clicks that occur before CM's correct answer submissions. 1

25th Percentile of Δt (Variable Type: FLOAT). The 25th percentile of time differences 2.1 between the CH *show answer* clicks that occurred before CM correct answers on the same problem, in seconds.. 1

25th Percentile of CA Δt (Variable Type: FLOAT). The 25th percentile (median) of the inter-arrival time distribution of the CM correct answer submission times that occurred after a CH *show answer* click on the same problem.. 1, 25

50th Percentile of Δt (Variable Type: FLOAT). The 50th percentile (median) of time differences 2.1 between the CH *show answer* clicks that occurred before CM correct answers on the same problem, in seconds.. 1

75th Percentile of Δt (Variable Type: FLOAT). The 75th percentile of time differences 2.1 between the CH *show answer* clicks that occurred before CM correct answers on the same problem, in seconds.. 1

75th Percentile of CA Δt (Variable Type: FLOAT). The 75th percentile (median) of the inter-arrival time distribution of the CM correct answer submission times that occurred after a CH *show answer* click on the same problem.. 1, 25

75th Percentile of Inter-arrival of Δt (Variable Type: FLOAT). The 75th percentile of the absolute values of inter-arrival time distribution of Δt ordered by time of CA submission, i.e. the time differences between the CH *show answer* clicks that occurred before CM correct answers on the same problem.. 1

90th Percentile of Δt (Variable Type: FLOAT). The 90th percentile of time differences 2.1 between the CH *show answer* clicks that occurred before CM correct answers on the same problem, in seconds.. 1, 10, 40–42, 53

90th Percentile of CA Δt (Variable Type: FLOAT). The 90th percentile of the inter-arrival time distribution of the CM correct answer submission times that occurred after a CH *show answer* click on the same problem.. 1

90th Percentile of SA Δt (Variable Type: FLOAT). The 90th percentile of the inter-arrival time distribution of the CH *show answer* click times that occurred before a CM correct submission on the same problem.. 1

90% Confident $\pi > 0.9$ (Variable Type: BOOLEAN). True if P(Beta(.5+NSAB, .5+N - NSAB) > 0.9) > 0.9, otherwise False.. 1

CA Acronym for *Correct Answer*. Refers to the submission of a problem on the edX platform resulting in full credit (completely correct answer).. 1, 17, 26, 92

CAMEO Copying Answers using Multiple Existences Online. *CAMEO* describes the cheating strategy where a single user employs a *harvester* account to obtain test answers via the *show answer* button and a *master* account to submit the correct answers for full credit. A CAMEO users is one who cheats via the CAMEO strategy.. 1, 2, 13, 21, 22, 26, 27, 29, 31, 35, 44–47, 52, 87

CH (Variable Type: STRING). Candidate Harvester. The account used to gather problem answers via the *show answer* button. Answers are recorded by the Candidate Harvester and submitted by the Candidate Master (CM). *Show answer* is usually available after a problem is answered incorrectly.. 1, 10, 12, 16–20, 26, 37, 38, 40–44, 54, 90

CH Modal IP (Variable Type: STRING). Modal IP address of CH. 1

CM (Variable Type: STRING). Candidate Master. The account using the CAMEO cheating strategy to submit correct answers.. 1, 10, 12, 16–20, 26, 37, 38, 40–44, 54, 90

CM Modal IP (Variable Type: STRING). Modal IP address of CM. 1

Fraction of Course Problems Completed (Variable Type: FLOAT). The number of problems correctly or incorrectly answered by the user, divided by the total number of problems available to answer.. 1, 13, 49

Item Response Theory Item Reponse Theory (IRT) is a paradigm for the design, analysis, and scoring of tests, questionnaires, and similar instruments measuring abilities, attitudes, or other variables. In the two-variable case, IRT is used to estimate user proficiency and problem difficulty. IRT may also be used for cheating detection in some contexts.. 1, 56, 88, 89

MOOC Massive Open Online Courses. MOOCs are typically free courses offered by universities that replicate some or all of the content of the traditional course in an online platform. MOOCs are *open*, meaning anyone may create an account, and *massive*, meaning tens of thousands of active users participate in the online course. MOOCs are typically taken through a MOOC platform provider, like edX edX or Coursera Coursera.. 1, 2, 21, 22, 27, 31, 35, 42, 48, 50, 52, 90

N (Variable Type: INTEGER). The number of problems for which CM submitted a correct answer and CH clicked *show answer* on the same problem.. 1, 16, 17, 19

NSAB (Variable Type: INTEGER). Number of Show Answers Before. Sometimes the variable X is also used. The number of CH *show answer* clicks that occurred before the CM correct answer submissions.. 1, 16, 17, 19

Percent Attempts Correct (Variable Type: FLOAT). The percentage of all attempts by the user resulting in a correct answer submission. For example, if a learner answered one question correctly on their second attempt, and a second question correctly on their first attempt, then Percent Correct Attempts $= \frac{2}{3} = 66.67\%$. 1, 13, 49

Positive-Unlabeled (PU) learning A subset of semi-supervised classification requiring training on a (typically small) subset of *true positive labels* without any *true negative labels*, i.e. all other points are *unlabeled*.. 1, 57

PSAB (Variable Type: FLOAT). Percent Show Answers Before. $PSAB = \frac{NSAB}{N} * 100.0$. The percentage of CH *show answer* clicks that occurred before CM correct answers.. 1

SA Acronym for *Show Answer*. Refers to the button available on the edX platform that reveals the correct answer to a problem after it has been answered incorrectly.. 1, 26

SA CA Δt Ordered Correlation (Variable Type: FLOAT). The Pearson Correlation between the CH *show answer* inter-arrival times occuring **before** the CM *correct answer* inter-arrival times. Before the correlation is computed, both inter-arrival time distributions are independently sorted in ascending order.. 1

Chapter 1

Massive Open Online Courses: The Challenge of Openness versus Value

In this chapter, you will learn about Massive Open Online Courses (MOOCs), the affordances MOOCs provide, their implications for cheating, and the contributions of this thesis. In Section 1.1, you will learn about MOOCs and their purpose of open education and in Section 1.2, you will see a survey of the challenges of cheating detection created by the stochasticity of massive learner datasets. In Section 1.3, you will learn about the Copying Answers using Multiple Accounts Online cheating strategy CAMEO and the challenges specific to CAMEO detection. Along the way in Section 1.4, you will learn about the importance of cheating detection in online courses as it relates to credentialing and in Section 1.5, you will learn how this thesis serves to enable a foundation for valuable online course credentials through a statement of contributions.

Throughout the non-technical sections of this and other chapters in this thesis, you will see *big ideas* stylized in italicized bold. These *big ideas* capture forward-thinking concepts and questions that drive the progression of this thesis. *Big ideas* are intended to prepare you for what is to come.

By the end of this chapter, you will have a deeper understanding of MOOCs and how their *show answer* feature and *openness* enables new cheating strategies. You will have seen the implications of the CAMEO cheating strategy and why CAMEO detection and general cheating detection is difficult in MOOCs. Finally, you will know about the contributions of this thesis toward the problem of CAMEO detection and the more general problem of $\tilde{P}\tilde{N}$ learning, i.e. binary classification with noisy labels.

1.1 Massive Open Online Courses and Formative Feedback

Massive Open Online Courses (MOOCs) are freely-available online university offerings that replicate some or all of the content of a traditional course in an online platform. They provide free, open-access to thousands of courses from the world's top universities, such as Harvard and MIT, while simultaneously enabling the analysis of massive learner datasets and the dawn of a new era in educational data-mining and learning analytics. Dubbed a *democratization of education* (Mazoue, 2014), MOOCs are *open*, meaning anyone with an Internet connection may create an account, and *massive*, meaning often tens of thousands of users participate in these online courses (Ho et al., 2014, 2015). MOOCs are typically taken through a MOOC platform provider, such as edX, Udacity, or Coursera, and vary greatly in difficulty, design, content, and intent. MOOCs instantiate a learning environment which may replicate some aspects of university course instruction, such as recorded video lectures, homework assignments, exams, and peer-to-peer and peer-to-instructor interaction through forums.

Beyond providing an online classroom experience for millions of learners, MOOC platforms often support data aggregation and extraction of client-side learner interactions and server-side analysis of problem submissions. These massive learner datasets afford new educational studies and experiments and make possible the educational research conducted in this thesis. I confine our analysis to 115 MITx and HarvardX courses hosted via the edX platform.

A major challenge for open online courses is maintaining global open ac-

cess without negatively impacting the perceived value of course credentials.

For a nominal fee, a learner can earn a certificate of completion in these courses as a representation of their understanding of the material. However, the ability for anyone to create multiple accounts enables new forms of cheating and calls into question the current viability of MOOCs as a credible educational credentialing platform. Are MOOC learners cheating in MOOCs? If so, what strategy are they using and how prevalent is this cheating strategy? Is this strategy detectable, and if so, how can I detect it? Is this method of detection based on human-defined filters and thresholds, and if so, can I instead develop a general solution for detection? Can I prove the correctness of the general solution, and if so, how well does it perform on benchmark datasets? I delve into these questions in Chapters 2, 3, and 4.

The inclusion of a* show answer *button on the edX platform enables immediate, formative feedback, but also new forms of cheating.

The *show answer* feature allows users to obtain instructor-provided solutions to assessment problems after they have depleted all submission attempts. This method of formative feedback is non-evaluative, supportive, timely, and specific and is known to improve learning in online courses (Shute, 2008; Nicol and Macfarlane-Dick, 2006). Unfortunately, the value of *show answer* comes at a price. Because MOOCs are *open* so that anyone can can create one or more accounts, *show answer* enables new copying strategies through the coordination of multiple accounts. Detection of such a strategy is a focus of this thesis.

1.2 Cheating Detection in MOOCs is Challenging Because Learner Data is Stochastic and Noisy

MOOCs vary greatly across many dimensions, and within each dimension, the variation of student behavior produces noisy learner data. In this section, I discuss some of these dimensions and the challenges that each presents for cheating detection.

MOOC courses differ largely in type, duration, content, format, and purpose, making the task of creating a unified cheating detection system more difficult. George Siemens, credited with coining the term MOOC, described two types of MOOCs (McAuley et al., 2010): (1) *xMOOCs*, which focus on scalability of learning and (2) *cMOOCs*, which focus on connectivity (Mackness et al., 2013) and community learning. As most forms of cheating do not apply to cMOOCs, I confine my attention to xMOOCs many of which have ten-thousand or more enrolled learners (Ho et al., 2015). These larger course sizes complicate cheating detection by increasing the possibilities of user behavior patterns and increasing run-times of learning analytic algorithms.

If we are to solve the problem of cheating detection in stochastic environments, then we must solve the problem of interpreting noisy predictions.

Our detection algorithm must produce consistently accurate results regardless of whether a user cheats on 10 or 500 problems, even when MOOCs vary drastically in course duration and content (Ho et al., 2014). If for example, I choose to predict cheating based on the number of videos watched, the result would be confounded by whether the course relied more on video or textual instruction (content) and how many weeks of video lectures occurred in the course (duration). Within a single course, some users may cheat selectively, while others may cheat on nearly every problem. Approaches to cheating detection must make predictions based on noisy behavioral signals without being confounded by the amount of data and how it varies from learner to learner.

Motivated by these challenges, I created the Rank Pruning algorithm. Rank Pruning is a robust and general solution to $\tilde{P}\tilde{N}$ learning, the problem of binary classification with noisy labels. Because cheating prediction is a binary classification task and it is often difficult to acquire true positive and true negative cheating labels with complete certainty, $\tilde{P}\tilde{N}$ learning is a generalization of cheating detection. By solving a more general problem, Rank Pruning is also a robust solution for noisy cheating detection in stochastic online environments. Although a comprehensive development of the Rank Pruning algorithm comprises Chapter 4, Rank Pruning is intrinsically motivated by the challenge of behavioral prediction with stochastic learner datasets and noisy labels illustrated by the examples discussed in this section.

Stochasticity across course subject material (Ho et al., 2014, 2015; King et al., 2013) and user level of expertise exacerbates the difficulty of distinguishing between *experts* and *cheaters*. For example, science courses often begin with prerequisite material and increase problem complexity over the duration of the course. In these cases, the inter-arrival times of correct answers for a typical non-cheating learner may be smaller between simpler questions versus complex questions. This might motivate us to try to detect cheating by measuring the interquartile range of the inter-arrival times of correct answers, 75th Percentile of CA Δt - 25th Percentile of CA Δt, with the expectation that users who cheat consistently are unaffected by problem complexity. However, this method of detection may incorrectly detect non-cheating users with prior knowledge of course content. This simple example illustrates the difficulty of learning a decision boundary that separates cheaters from both non-cheaters and expert users in a complex feature space, a difficulty exacerbated by large variation among learner types from k-12 to post-doctoral (Ho et al., 2014) and learner intentions (Reich, 2014).

In addition to the challenges within and among course content and learners, MOOC platforms are constantly evolving (Henno et al., 2014) and therefore, so must the definitions of cheating. On December 7th, 2015, edX removed the *honor certificate*, a free credentialed recognition of course completion, requiring users who wished to be certified to pay for an *ID-verified certificate*. This change dramatically reduced the number of user's who certify, and required rethinking how to define cheating. Much of the analysis in Chapter 2 took place before this change, and thus, I only considered cheaters who were certified, however, a relaxation of this filter should be considered when analyzing courses ending after December 7th, 2015.

The *massive* population sizes and low cost of MOOCs suggests a need for scalable, inexpensive cheating detection, which precludes the use of human overseers in a proctored setting. This restriction to a non-human-proctored setting may increase cheating risks. For example, in a study across four e-campus courses at Troy University, students scored significantly lower on human-proctored exams versus non-human-proctored exams (Prince et al., 2009). A non-proctored setting may also prompt innovations for acquiring certain cheating labels because often a group of proctors is needed to monitor the student activities to identify cheating (Baker et al., 2004b). Although requiring grading center examinations in MOOCs would alleviate these issues, it would challenge the MOOC model of *free, open* education (Mazoue, 2014), as not every learner can afford or has access to a nearby testing center.

1.3 The Copying Answers using Multiple Existences Online (CAMEO) Cheating Strategy

A broad goal of this thesis is to accurately detect usage of the *Copying Answers using Multiple Existences Online* (CAMEO) cheating strategy. I describe this strategy in this section. A user employing this strategy, whom I refer to as a CAMEO cheater, earns a certificate by creating at least two MOOC accounts: (1) one or more *candidate harvester* (CH) accounts used to acquire correct answers by guessing at test answers and then accessing instructor-provided solutions via a *show answer* (SA) button, and (2) one or more *candidate master* (CM) accounts used to submit the copied solutions for full credit as correct answer (CA) submissions. I define Δt to be the difference of the time when the *master* submits the correct answer minus the time when the *harvester* clicks *show answer*, on the same problem.

Variability of CAMEO-use is vast. CAMEO cheaters can delay submission of copied answers, or submit copied answers immediately. They can vary how much they cheat (all or few problems), how often they cheat (seldom or large consecutive groups), and how consistently they cheat (patterned or randomized).

Additionally, any cheating detection algorithm must detect CAMEO-use irrespective of problem type: *fill-in-the-blank response*, *multiple-choice*, or *multi-select* type. This presents many challenges. For example, *multi-select* problems may contain one or more questions, called *items*. Although the edX platform ensures the consistent behavior of *show answer* buttons across all problem types, there is no *show answer* for each individual *item*. Therefore, the minimum Δt needed to cheat via the CAMEO strategy is proportional to the number of *items* in the *multi-select* problem. Since some *multi-select* problems contain as many as 15 or more *items*, Δt may differ greatly across problem types. Similarly, the minimum Δt needed to cheat via CAMEO *fill-in-the-blank response* is greater than for *multiple-choice* questions, since *fill-in-the-blank response* questions require either copy-and-paste or re-typing of solutions versus the clicking of a single radio button.

CAMEO detection must generalize to all of these behaviors.

1.4 The Vision of CAMEO Cheating Detection

In this section, I motivate the necessity and implications of CAMEO cheating detection within the broader context of MOOC certification.

Although the technical focus of this thesis is to achieve accurate detection of *multiple account* cheating in MOOCs, my broader goal is to establish a foundation of reputable certification in MOOCs. For learners all over the world who hope to improve their socioeconomic mobility, I strive for a future where MOOC certification is recognized as a legitimate credential.

However for such bold claims to be realized, MOOCs must achieve societal acceptance, and course completion certificates must hold significant societal value. Reputable certification is a precursor. Combined with growing evidence that the reputation and usefulness of MOOC certification are predictors of MOOC persistence (Alraimi et al., 2015), I anticipate that widespread awareness of MOOC susceptibility to the CAMEO strategy could depress MOOC popularity and persistence among general users. Although, most MOOC platforms mandate a *one account per person* policy (courseraterms; edxfaq; udacityterms) making the creation of multiple accounts by a single user grounds for revocation of certification, absolute enforcement of this policy poses restrictions on the MOOC initiative to achieve *open-access* quality education.

For many courses, a CAMEO user can easily acquire a course certificate without content knowledge in less than an hour. The CAMEO strategy holds the potential to render the MOOC certificate valueless.

In 2015, MIT announced the creation of the MITx Supply-Chain Management MicroMasters, claiming "The MITx MicroMasters in Supply Chain Management is equivalent to a coursework of one semester at MIT. Upon attainment of the MITx MicroMasters, learners are eligible to apply for an accelerated, residential, one-semester master's degree program in Supply Chain Management at MIT. Performance in the MicroMasters will play a strong role in admissions" (http://scm.mit.edu). Unexpectedly, CAMEO cheating prevalence soared in many of these courses, with MIT course instructors complaining of "rampant cheating". Prevalent cheating in online settings has been corroborated throughout cheating literature. Learner adherence to honor code policies in traditional classroom settings (McCabe and Trevino, 1993; McCabe et al., 1999) has been shown to lessen in online settings (LoSchiavo and Shatz, 2011; Mastin et al., 2009). Additionally, multiple studies have found that participants are less honest when interacting virtually than in person (Rockmann and Northcraft, 2008; Van Zant and Kray, 2014).

Innovations in educational technologies introduce new affordances for cheating. The CAMEO cheating strategy is one example, but users might also plagiarize online resources or other learner responses using peer-graded assessment (McCabe, 2005). Because the CAMEO strategy directly threatens the potential for MOOC platforms to evolve into valued academic credentialing services, I confine my attention solely to the detection of CAMEO cheating.

1.5 Contributions of this Thesis

Much of what is achieved in this thesis is the result of a rich, bi-institutional, collaborative effort between researchers at MIT and Harvard. In particular, Andrew D. Ho, a Professor of Education at the Harvard Graduate School of Education, and Isaac L.

Chuang, a Professor of Physics, a Professor of Electrical Engineering and Computer Science, and Senior Associate Dean of Digital Learning at MIT, co-authored Chapter 2 and contributed substantially to the development of the CAMEO algorithm. Tailin Wu, a graduate student in the Department of Physics at MIT and Isaac Chuang co-authored Chapter 4 and contributed significantly to the theoretical and experimental development of the Rank Pruning algorithm.

In this thesis, with the help of these colleagues, I define and detect *multiple account* cheating and discuss its significance for Massive Open Online Courses. For detection, I develop the CAMEO algorithm for *multiple account* cheating in Massive Open Online Courses and the Rank Pruning algorithm for binary classification with noisy labels, where binary classification is a generalization of cheating detection.

In particular for the CAMEO algorithm, Andrew Ho (Harvard), Isaac Chuang (MIT), and I:

- Define the Copying Answers using Multiple Existences Online (CAMEO) cheating strategy and estimate a *lower bound* for its prevalence among 1.9 million course participants in 115 HarvardX and MITx courses.

- Identify preventive strategies that can decrease CAMEO rates and show evidence of their effectiveness in science courses.

- Establish new educational data-mining methodologies for analysis in Massive Open Online Courses.

- Describe a novel *honeypot* validation technique that verifies cheating by appending unique digit sequences to *show answer* fields.

- Develop the CAMEO detection algorithm using human-defined filters and thresholds as a method for producing noisy CAMEO cheating labels in MOOCs.

In particular for the Rank Pruning algorithm, along with Tailin Wu (MIT) and Isaac Chuang (MIT), and I:

- Develop a robust, time-efficient, general solution for both $\tilde{P}\tilde{N}$ learning, i.e. binary classification with noisy labels, and estimation of the fraction of mislabeling in both the positive and negative training sets.

- Introduce the *learning with confident examples* mantra as a new way to think about robust classification and estimation with mislabeled training data.

- Prove that under assumptions, Rank Pruning achieves perfect noise estimation and equivalent expected risk as learning with correct labels. I provide closed-form solutions when those assumptions are relaxed.

- Demonstrate that Rank Pruning performance generalizes across the number of training examples, feature dimension, fraction of mislabeling, and fraction of added noise examples drawn from a third distribution.

- Improve the state-of-the-art of $\tilde{P}\tilde{N}$ learning across F1 score, AUC-PR, and Error. In many cases, Rank Pruning achieves nearly the same F1 score as learning with correct labels when 50% of positive examples are mislabeled and 50% of observed positive labels are mislabeled negative examples.

An ancillary contribution of this thesis is the provision of a substantive glossary with detailed descriptions of methods and features for MOOC researchers.

Chapter 2

Detecting and Preventing "Multiple-Account" Cheating in Massive Open Online Courses

In this chapter[1], you will learn about the Copying Answers using Multiple Existence Online (CAMEO) strategy, a cheating strategy enabled by the features of massive open online courses (MOOCs) and detectable by virtue of the sophisticated data systems that MOOCs provide. The CAMEO strategy involves a user who gathers solutions to assessment questions using a *harvester* account and then submits correct answers using a separate *master* account. This chapter serves three main purposes, (1) to define the CAMEO strategy, (2) to establish techniques for detection, and (3) to estimate a lower bound of CAMEO prevalence.

In Section 2.1, we frame the CAMEO strategy within the context of MOOCs and other cheating strategies. We then define CAMEO in Section 2.2, and establish an algorithm of five conjunctive filters as a method for detection. In Section 2.3, we use a small-scale experiment to verify CAMEO and estimate a *lower bound* for its prevalence among 1.9 million course participants in 115 MOOCs from two universities.

[1] Andrew Ho, Harvard, and Isaac Chuang, MIT, contributed significantly to the contents of this chapter. I elect the use of the pronoun *we* throughout to emphasize their enlightening contributions.

Using conservative thresholds, we estimate CAMEO prevalence at 1,237 certificates, accounting for 1.3% of the certificates in the 69 MOOCs with CAMEO users. We discuss the impact of the cheating strategy within the broader context of MOOCs in Section 2.4, and find that among earners of 20 or more certificates, 25% have used the CAMEO strategy. We conclude in Section 2.5 with a discussion of the CAMEO strategy's ability to undermine the potential for MOOCs to increase efficiency and spur innovation in higher education.

Once you have finished this chapter, you will know how the CAMEO cheating strategy works and a filter-based algorithm for estimating a lower bound of CAMEO cheating prevalence, as well as new techniques for understanding MOOC learner behaviors and methods for CAMEO prevention.

2.1 Introduction and Motivation

Massive Open Online Courses began receiving significant media coverage in 2012 (McNutt, 2013; Pappano, 2012), coincident with the widespread commitment by established universities to providing free courses online (Christensen et al., 2013; Ho et al., 2014; stanfordonline). These MOOCs distinguished themselves from predecessors like MIT's Open Courseware (Smith, 2009; d'Oliveira et al., 2010) by providing not only free content but a course-like structure, including enrollment, synchronous participation, periodic graded assessments, online discussion forums, interactive simulations, and of greatest relevance for our purposes, certification of successful completion (DeBoer et al., 2014; Linn et al., 2014). One theory of MOOC proliferation holds that free certification of proficiency in college courses can reduce inefficiencies in higher education by replacing high-cost residential courses with low-cost online certification (Hoxby, 2014).

In this chapter, we reveal a particular cheating strategy that is detectable across the 115 MOOCs in our sample and currently presents a serious threat to the trustworthiness of their certifications. We call the strategy, Copying Answers using Multiple Existences Online. A user employing this strategy, whom we refer to as a CAMEO user, earns a certificate by creating at least two MOOC accounts: (1) one or more *harvester* accounts used to acquire correct answers by guessing at test answers and then accessing instructor-provided solutions via a *show answer* button, and (2) one or more *master* accounts used to submit these solutions as correct test answers.

The CAMEO strategy lies at the intersection of a number of other copying techniques and contexts. We distinguish between (1) what is copied, (2) why it is copied, (3) how it is copied, and (4) how copying is detected. The CAMEO strategy occurs in similar contexts as community collaboration in online courses (Yang et al., 2014), and detection of both involves analyzing the interactions of multiple accounts. However, prior efforts have focused on how communities of different users affect learning outcomes (Kumar et al., 2007), in contrast with CAMEO behavior, where a single user exploits multiple accounts, potentially circumventing the learning process entirely. CAMEO is most similar to multiple-account sharing strategies in online games (e.g. (Kafai and Fields, 2009)), where a single user can increase scores or other in-game outcomes by creating multiple accounts and interacting them strategically. However, CAMEO behavior distinguishes itself from online game strategies due to what is copied (correct answers to tests) and why it is copied (to fake or expedite certification of proficiency). As we show, the specificity of these differences enables targeted detection, quantification, and prevention of CAMEO use in these MOOCs.

Cheating by CAMEO shares similarity in purpose with copying in online and conventional courses (Baker et al., 2004a; Kauffman and Young, 2015; McCabe et al., 2012; Palazzo et al., 2010). However, three features of CAMEO make it a unique threat as a cheating strategy in online education. First, it is internally sufficient. Whereas most users copy from other learners or external resources, CAMEO users employ multiple accounts to copy from themselves, making the cheating strategy highly accessible by removing dependence on outside resources. As a result, the strategy is extremely effective. Second, in asynchronous MOOCs, where learners can access course materials and assessments at their own pace, a CAMEO user can employ the CAMEO strategy for every question they attempt, allowing certification for full course completion in a single sitting. Third, it is unrestricted, employable in a nonselective, open admission setting. Degrees from selective institutions assert, at the very least, that users have been pre-screened, but MOOC certificates do not. Because MOOC users, unlike most postsecondary learners, are not selected by any merit-based process or criteria, the considerable accessibility of CAMEO in these MOOCs holds the potential to render their certificates valueless as an academic credential.

The key contributions of this chapter are a detection algorithm for the CAMEO-based cheating that allows for a lower bound estimate of prevalence and a small-scale experiment confirming CAMEO behavior. This latter experiment is an extension of *honey pot* cheating detection (Corrigan-Gibbs et al., 2015b), where copied answers can be confirmed directly. These contributions complement the considerable literature that estimates cheating prevalence through surveys, where survey responses may be influenced by social desirability, interpretation of item prompts, concerns about anonymity, and inflation in self-reported performance (Mastin et al., 2009). This chapter investigates a specific cheating strategy using an algorithm customized to big datasets that contain detailed user interactions with online course content, including activity time-stamps. With 115 courses, this is also the largest analysis of cheating in online courses of which we are aware.

CAMEO also represents an example of a more general tendency for open online learning systems to enable both new strategies for cheating and new strategies for detection (Horton et al., 2011; Li et al., 2015; Raines et al., 2011). Although CAMEO is technically a copying strategy, we argue that its use in MOOCs constitutes *cheating.* At a minimum, employing CAMEO is a violation of policy, because MOOC honor codes forbid the creation of multiple accounts (courseraterms; edxterms; udacityterms). The CAMEO strategy also threatens perceptions of the value of MOOC certification. Any reasonable interpretation of standard MOOC certificates, which refer to *successful completion* (edxterms), includes proven learner proficiency with course content. Yet, the prevalence of the CAMEO strategy justifies a starkly contrasting interpretation of MOOC certification-that a user merely copied answers from a *dummy* harvester account. Combined with growing evidence that the reputation and usefulness of MOOC certification are predictors of MOOC persistence (e.g., (Alraimi et al., 2015)), we anticipate that widespread awareness of MOOC susceptibility to the CAMEO strategy could depress MOOC popularity and persistence among general users.

2.2 Methodology

We begin by describing a CAMEO detection algorithm that relies on the distribution of differences in time between particular user actions across particular user pairs. The CAMEO detection algorithm is comprised of five filters with highly conservative cutoffs intended to reduce false positives, including a Bayesian criterion for the timestamp difference distributions. After we present these filters, we describe a small-scale experiment that confirms CAMEO cases, and we show that the CAMEO algorithm detects these cases as expected.

2.2.1 Indicators of *Copying Answers using Multiple Existences Online* (CAMEO)

This is the difference between the time that a master account, m, submits a correct answer and the time that a harvester account, h, acquires the correct solution, for a problem (item) in common, i, in a given MOOC course, c. It is possible for a single master to have multiple harvesters and a single harvester to have multiple masters. The subscript, c, recognizes that the same master-harvester pair may be employing CAMEO across multiple courses.

Logically, for CAMEO users, these Δt are predominantly or entirely positive in sign. The former time, $t_{m,c,i}$, is recorded in server log files. For the latter time, $t_{m,c,i}$, we take advantage of the fact that instructors of the moocs in our sample generally allow users to click a *show answer* option after submitting answers, to display a staff-prepared answer and/or an explanation of the solution, in order for users to obtain rapid feedback. The timestamp produced by a *show answer* click defines - $t_{m,c,i}$. We introduce a method for probabilistic detection of CAMEO users based on observed distributions of Δt over items i.

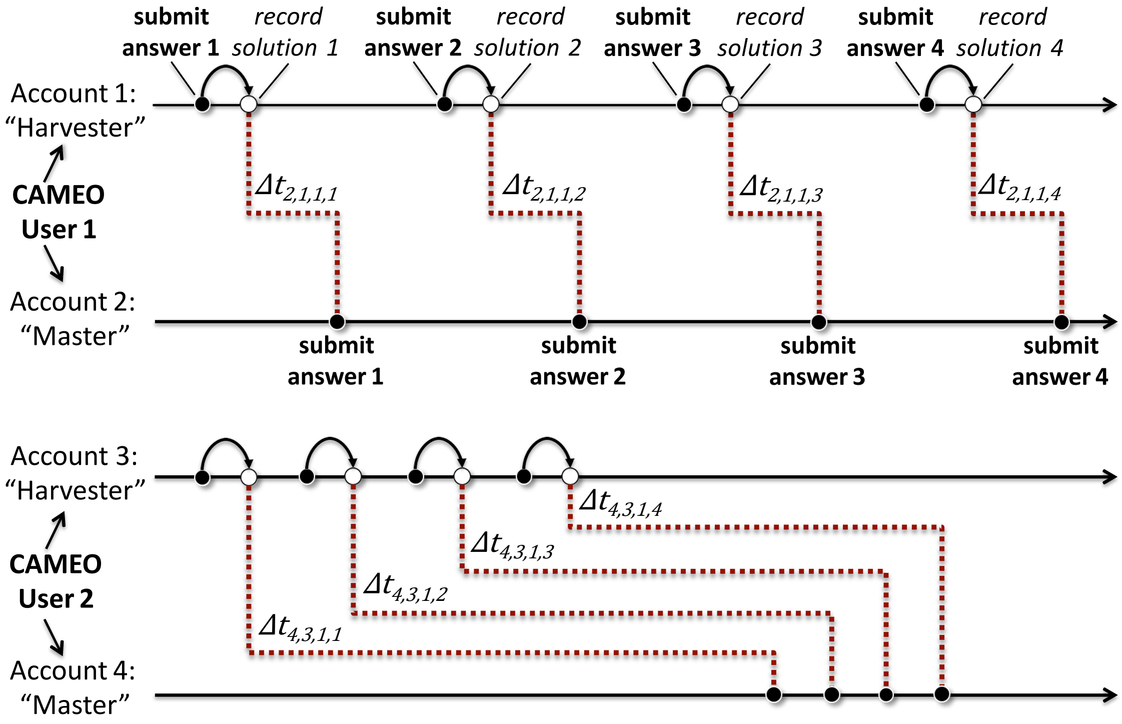


Figure 2-1: Two types of prototypical behavior when Copying Answers using Multiple Existences Online (CAMEO). A *harvester* account h records correct solutions, and a *master* account m submits correct answers. The time between harvesting in account h (white dot) and correct answer submission by account m (black dot) is estimable from the data and defined as Δt for item i in course c. The strategy employed by CAMEO 1 is to alternate harvesting and submission. The strategy of CAMEO 2 is to harvest a batch and then submit a batch.

Fig. 2-1 illustrates two prototypical CAMEO users, each with two accounts, and their timeline of interactions with online assessments. For both CAMEO users in Fig. 2-1, we also illustrate the variable:

$$\Delta t_{m,h,c,i} = t_{m,c,i} - t_{h,c,i} \tag{2.1}$$

2.2.2 Detection of *Copying Answers using Multiple Existences Online* (CAMEO)

The detection strategy begins by considering all possible ordered pairs of accounts, within each course, as candidate CAMEO users. It asks whether the pattern of *show* *answers* from one, the *candidate harvester* (CH), and *correct answers* from the other, the *candidate master* (CM), is ordered and coincident enough to declare the CH-CM pair a CAMEO user. In total, we employ five filters to identify CAMEO users (2.1). These five filters are conjunctive and thus order-independent; we group them conceptually and order them narratively.

The first two filters reflect the logic that a CAMEO user's CH often provides correct answers to the CM fairly quickly; thus, the distribution of Δt over items i should be positive with small magnitudes. Fig. 2-2 shows four contrasting distributions of Δt for four different CH-CM pairs. Distribution A illustrates two unrelated and asynchronous accounts, where one user's *show answer* event is sometimes before and sometimes after another user's correct answer submissions by times that vary widely in magnitude; distributions like this should be common. Distribution B illustrates two users (e.g. siblings, roommates, or learners taking the assessment side-by-side) working in close synchronicity. Due to chance and differences in pacing, one user's *show answers* will sometimes precede but sometimes follow the other's *correct answers*, but times will be in close proximity.

Condition	Explanation	Operationalization
The Δt distribution should be positive	The CH should harvest the correct answer before the CM submits the correct answer.	Bayesian - 90% confident that the proportion of positive Δt values is 90%
The magnitudes of the Δt should be small	The CH should provide answers to the CM quickly.	The 90th percentile of the Δt distribution should be less than 5 min.
The CH should not be certified, and the CM should be certified	The CH should be guessing, uninterested in certification. The goal of the CM is presumably certification	A CM must be certified. A CH must not be certified.
The CM and CH should share an IP address or have shared one at some point in their course-taking history.	This increases the likelihood that the CM and CH are in fact the same person.	The CM and CH must share one of the sets determined by the transitive closure of modal IP address and account name over courses.
There should be few accounts that have shared an IP address with the CM and CH.	This excludes areas, e.g. school networks, where chance coincidence of Δt may lead to false detection	The number of accounts with a shared modal IP address must not exceed 10.

Table 2.1: A detection approach that asserts five necessary filtering conditions for candidate harvester (CH) and candidate master (CM) pairs to be classified as Copying Answers using Multiple Existences Online (CAMEO).
Notes: The filters are chosen to be conservative, and their conjunctive application minimizes the chance of false identification at the cost of conceding missed CAMEO users. In terms of missed identification, Filter 1 excludes small-sample CAMEO users even when their proportions of positive times are 100%. Filter 2 excludes CAMEO users that take more than 5 min to pass solutions between accounts. Filter 3 excludes those who use the CAMEO strategy but do not earn certificates. Filter 4 addresses those who use IP-masking strategies like the Tor browser. Filter 5 excludes CAMEO users within classrooms, cafes, and other scenarios in which IP addresses are shared.

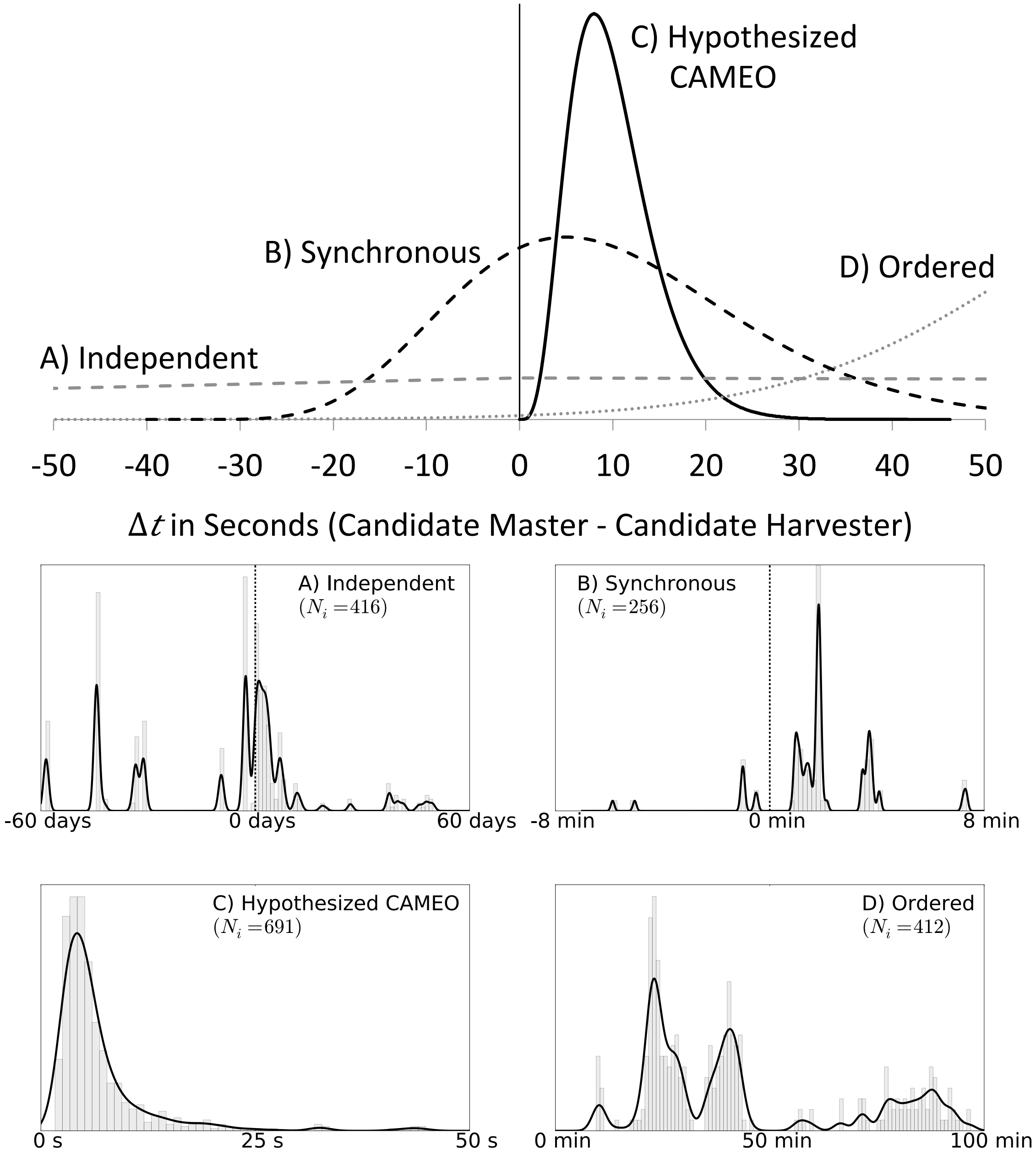


Figure 2-2: Four types of *average* theoretical distributions of Δt (top) with examples of empirical *observed* distributions (below). Distribution A illustrates uniformly distributed *show answer* and *correct submission* times resulting in a shallow triangular distribution symmetrical around 0. Distribution B illustrates synchronous submission with positive and negative Δt values. Distribution C illustrates prototypical *Copying Answers using Multiple Existences Online* (CAMEO) behavior, with candidate harvester accounts passing solutions to candidate master accounts over a short time span. Distribution D illustrates consistently and coincidentally ordered submissions over a longer time span. For the empirical distributions, the number of items shared between a harvester's *show answer* and a master's *correct submission* is displayed as N_i.

Distribution C reflects prototypical CAMEO behavior, corresponding to Fig. 2-1. All Δt are positive, and their magnitudes are extremely small, centered in this illustration at around 10 s. These small Δt magnitudes are typically possible when the CAMEO user is logged in simultaneously to both CH and CM accounts on different internet browsers or computers. Finally, Distribution D is also positive but with Δt magnitudes that are larger and more variable. This is consistent with ordered coincidence, where unrelated pairs of users will be offset from each other due to different enrollment dates or time-of-day preferences.

To identify CAMEO users by distributions of Δt, we considered constraining the population distribution of Δt or $|\Delta t|$ by strong parametric assumptions (e.g., log-normal, exponential), but many observed distributions had extreme skew due to outlying Δt values. We therefore opt for a less parametric approach that targets the percentage of positive observations (Filter 1) and the magnitude of the 90th Percentile of Δt (Filter 2).

2.2.2.1 Filter 1: the Bayesian criterion

For Filter 1, given variation in the quantity of data shared between any CH and CM, we use a Bayesian criterion that is more stringent when data are limited (Lehmann and Casella, 1998). We estimate the parameters of the posterior distribution of a proportion π, our parameter of interest indicating the proportion of positive Δt values, given n, as the number of in-common items for which a CH has a *show answer* and a CM has a correct answer, and x, as the number of times that the CH time precedes the CM time:

$$x_{m,h,c} = \sum_{i=1}^{n} I(\Delta t_{m,h,c,i} > 0) \tag{2.2}$$

Here, I is the indicator function, which is 1 when the argument is true and 0 otherwise. The maximum n for any CH-CM pair is the number of items. The average number of graded items is 141, across courses, allowing considerable data for inference. We assume that x is binomially distributed and that π has a Beta distribution.

Following standard rules of conjugacy:

$$x|n, \pi \sim \text{Binomial}(\pi, n) \tag{2.3}$$

$$\pi|\alpha, \beta \sim \text{Binomial}\pi, n \tag{2.4}$$

$$\pi|x, n, \alpha, \beta \sim \text{Beta}(\alpha + x, \beta + n - x) \tag{2.5}$$

We observe x and n in the data. For the prior distribution, we set $\alpha = \beta = 0.5$, empirically and judgmentally, using full distributions of observed $p = x/n$ when n is large in our data. This is a gentle U-shape, consistent with the fact that many distributions of $t_{m,c}$ are stochastically or entirely offset from other distributions of $t_{h,c}$ in one direction or other, due to the asynchronous nature of moocs.

We operationalize Filter 1 in terms of confidence that π is close to 1, that is, that CH interaction with an item almost always precedes CM interaction. Specifically, Filter 1 selects CH-CM pairs with a 90% probability of $\pi_{m,h,c} > 0.9$. This is a conservative, stringent criterion that requires considerable data before concluding that a distribution is predominantly positive. Even a CH-CM pair with $x = 12$ out of $n = 12$ ($p = 100\%$) positive values is insufficient to meet this criterion.

2.2.2.2 Filter 2: setting the cutoff threshold

Filter 2 addresses the fact that Filter 1 excludes Distributions A and B from CAMEO consideration, but it cannot distinguish between Distributions C and D (Fig. 2-3). To exclude ordered accounts that happen to be offset in time in the positive direction, Filter 2 uses the Δt distribution as a criterion, setting a conservative cutoff at 5 min. In other words, the 90th Percentile of Δt values must be less than 5 min. This cutoff occurs at an *elbow* as shown in Fig. 2-3, where shifting the cutoff between 0 and 5 min changes the number of estimated CAMEO users dramatically, and subsequent shifts past 5 min do not.

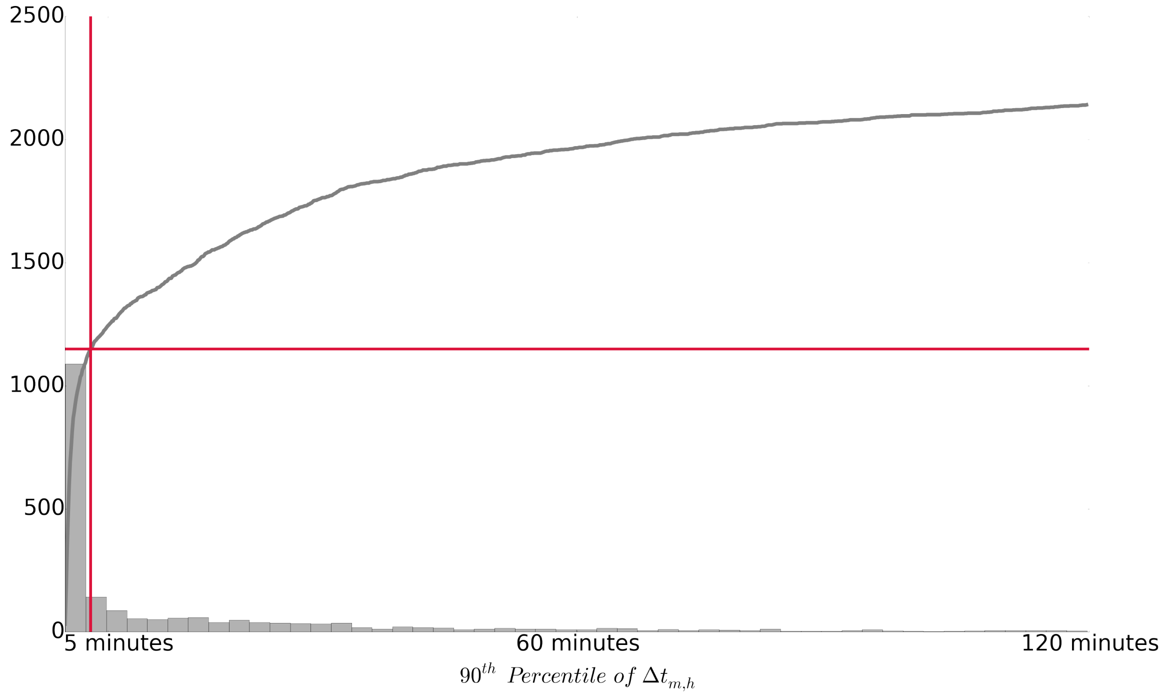


Figure 2-3: Cumulative distribution (line) showing number of CAMEO users identified versus the cutoff value of 90th Percentile of Δt (Filter 2 in Table 1), together with the associated histogram (bars). The vertical red line depicts the cutoff value chosen. The horizontal red line is the corresponding number of CAMEO users identified.

2.2.2.3 Filter 3: certified CM - uncertified CH pairs

The first two filters provide considerable evidence that, for CAMEO users, the distribution of Δt is disproportionately positive and centered at less than 5 min in time. Filters 3 through 5 provide convergent criteria to further minimize the probability of false identification. Filter 3 considers only CH-CM pairs for which the CH is uncertified and the CM is certified. Although this may discard CAMEO users who do not ultimately earn certification, our intention is to address possible threats to MOOC certificate validity as directly as possible, so we include only certified CMs. In addition, a CH that earns a certificate is inconsistent with the interpretation of CAMEO users as a cheating strategy, since it leaves open the possibility that the CH is actually proficient in the course.

2.2.2.4 Filter 4: detecting shared IP address

Filter 4 further reduces the candidate pool to those CH-CM pairs who share an IP address, defined for each account as the modal (most commonly used) IP address across all logged interactions in a given course c. However, considering only users with the same IP address fails to detect users who employ the CAMEO strategy using accounts assigned different modal IP addresses in a given course, either by coincidence or intentional misdirection. To improve detection of these users, we broaden the definition of *sharing an IP address* to CH-CM pairs who have ever shared an IP address in their course-taking history.

To detect CAMEO users with accounts having different modal IP addresses in a given course, we consider every unique (name, IP) tuple across all accounts participating in any of the 115 courses analyzed. We assign each (name, IP) an *IP group*, initially as a unique integer for each pair. Next, we group by modal IP address such that all (name, IP) tuples sharing the same modal IP address are assigned (merged into) the same IP group. Then, we group by username such that all (name, IP) tuples sharing the same username are merged into the same IP group. We repeat both the *merge by IP* and *merge by username* steps until the IP group no longer changes. This can be described as a *transitive closure* of modal IP address and account names for all accounts across courses. It allows us to consider CM-CH pairs whenever the two accounts have shared a common modal IP address within a course, across courses, or across other accounts that have shared the same modal IP address within and across courses.

2.2.2.5 Filter 5: excluding shared routers

Filter 5 excludes all CH-CM pairs who are part of a group that has 10 accounts or more that share a modal IP address. We intend this to exclude shared routers among classrooms or cafes that might increase the likelihood of false positives.

2.2.3 Verification of *Copying Answers using Multiple Existences Online* (CAMEO)

We conducted a small-scale, targeted investigation of registrants in a single, small course to confirm existence of the CAMEO strategy. Through descriptive analyses of usage patterns over time, the instructor identified 3 pairs of users, consisting of 3 candidate master accounts and 3 candidate harvester accounts, whose assessment submissions seemed unusually synchronous. For these three user pairs, we adapted answers to 7 test questions to append a unique random string to the answer displayed to each user. This string took the form of a superfluous symbol (e.g. parentheses), negligible decimal points at the end of a correct answer, or an expression that evaluates to 1. For example, an answer to the question *what is the final momentum of the particle?* could be 3.13, but the answer was displayed as *3.13556* to one user, and *3.13417* to another. For logistical and pedagogical reasons, this targeting was restricted to these three user pairs.

One of these three pairs never viewed these items. For both of the remaining candidate master accounts, we detect direct copying of at least one unique answer from the harvester accounts. This confirms CAMEO behavior, given that the unique combinations of extra digits and symbols had no reason to be submitted and could not have happened by chance. For small-scale validation, among the 3 pairs of users, the CAMEO detection algorithm identified only and exactly the same two master accounts as CAMEO users. The next section builds from this existence proof to estimate the lower bound prevalence for CAMEO behavior in these moocs.

2.3 Results

We investigate the prevalence of CAMEO users in 115 online courses from two institutions, Harvard University and MIT, offered on the MOOC platform, edX.[2] We use

[2] A list of the 115 courses studied, with their classifications into topic areas, and Δt distribution data for CM-CH pairs, are archived in the Harvard Dataverse Network, at http://dx.doi.org/10.7910/DVN/3UKVOR.

data from courses from the fall of 2012 through the spring of 2015, up to an analytic cutoff date of June 2, 2015. About half of these MOOCs are described in detail in other reports (Ho et al., 2015; McNutt, 2013) that emphasize their range of curricular foci and their heterogeneous participant demographics. Our sample consists of 1,893,092 enrollments (1,067,570 from unique accounts) whose users clicked into the course content at least once. A total of 155,301 certificates were ultimately earned from 103,370 unique accounts.

2.3.1 Prevalence of CAMEO

Across these courses, we estimate that a total of 1,237 certificates were earned using the CAMEO strategy, 1% across all 115 courses, by 657 unique users employing 674 harvester accounts. In some courses, CAMEO users account for as many as 5% of the certificates earned. Across the 69 courses in which we identified CAMEO users, they account for 1.3% of certificates. Table 2.2A shows that CAMEO users are more likely to be young, male, less educated, and international than their certified counterparts in the same courses (Ho et al., 2015). Among countries with at least 20 CAMEO users, countries with the highest CAMEO counts per certificates were Albania (12%), Indonesia (4%), Serbia (3%), Colombia (2%), and China (2%). The CAMEO rate in the USA is particularly low, at 0.4% of certificates earned. Table 2.2B shows CAMEO prevalence by broad curricular area. Prevalence of CAMEO users is greatest in the Government, Health, and Social Science category (1.3%) and lowest in the Computer Science category (0.1%).

(A)

Among 69 courses with CAMEO users	Non-CAMEO	CAMEO
N Certified	96,367	1237 (1.3%)
% Female	33%	19%
% Bachelor's	79%	59%
Median Age	32	25
% USA	30%	14%

(B)

Among all 115 courses	N Courses	% CAMEO of certified
Computer Science	12	0.1%
Government, Health, and Social Science	28	1.3%
Humanities, History, Religion, Design, and Education	38	1.1%
Science, Technology, Engineering, and Mathematics	37	0.7%
Overall	115	0.9%

(C)

Among 37 STEM courses	N Courses	N Certified	N CAMEO	% CAMEO (typical user)	% CAMEO (typical course)
No or limited CAMEO prevention	19	19,383	171	0.9%	1.2%
CAMEO prevention	18	11,717	8	0.1%	0.1%
Overall	37	31,100	179	0.6%	0.7%

Table 2.2: Distribution and demographics of those identified as Copying Answers using Multiple Existences Online (CAMEO users) across courses. (A) Prevalence and demographic distribution of CAMEO users versus non-CAMEO certificate earners in the 69 courses with nonzero CAMEO users. (B) Distribution of CAMEO users across four broad curricular areas, for the 115 courses in the dataset. (C) Observed differences in CAMEO percentage for Science, Technology, Engineering, and Math courses that do or do not employ mechanisms that logically prevent CAMEO users, including solutions embargoed until after due dates and algorithmic generation of problems with varying solutions.

Note: Survey methods follow those of other studies: Demographic information collected from edX surveys with response rates >95%; Country is determined by geolocation of the modal IP address; Courses are divided into curricular areas judgmentally.

2.3.2 Prevention of CAMEO

Mechanisms which logically prevent CAMEO use include restricting the "show answer" option until after assignments are due, and using algorithmic generation of assessment items so that participants receive randomly varying items, each with different solutions. Across the 37 Science, Technology, Engineering, and Mathematics (STEM) courses in this sample, 18 employed such prevention mechanisms. Table 2.2C shows that the CAMEO rate in courses that employed these preventive strategies in half or more of the assessment items was substantially lower (0.1%) than the rate in courses that did not employ preventive strategies (1.2%).

2.4 Discussion

As open online courses proliferate, we identify CAMEO as a significant threat to the validity of large-scale certification. Our primary goals are to demonstrate that CAMEO exists and to bound its prevalence in the population. We believe that our method accomplishes this and does so conservatively. Nonetheless, we raise here a central shortcoming of this work and address it briefly while encouraging subsequent research. Like many cheating analyses in real contexts, we have no *true* knowledge of cheating to evaluate whether our detection method is accurate at the individual level. Perhaps a child is guessing haphazardly and clicking *show answer,* while working with a parent who separately submits answers correctly, always a few minutes after the child. This is unlikely but not impossible. However, our aim is not to identify individuals but estimate aggregate prevalence. We believe our filters, combined with the small-scale experiment that provides an existence proof, accomplish this.

We also raise three convergent sources of evidence. First, text-matching of usernames reveals considerable overlap in candidate pairs; many CAMEO users have usernames consistent with the Master-Harvester hypothesis, like *Curtis1* and *Curtis2.* Second, although our CAMEO detection algorithm treats every CM-CH pair independently, we find CAMEO behavior is clustered within users. A total of 43 separate accounts have earned 5 or more certificates by CAMEO. Third, we conducted a limited analysis, in one course, of plagiarism by copying open-response text across users, and we find that these accounts are also identified as CAMEO users. Although we believe our algorithm alone is sufficient to demonstrate the existence and bound the prevalence of CAMEO, we encourage further research to support validation of the detection algorithm.

Another concern is the possibility that some users could be using CAMEO to increase their exposure to assessment items and thereby increase their learning. We argue that this is unlikely given how we operationalize our definition. CAMEO users require nearly all of CH *show answer* clicks to occur *shortly* before CM correct answer submissions. In fact, we found that often the actual time difference was only a few seconds. The extent and timing of this systematic behavior is most consistent with a cynical and blatant attempt to harvest correct answers to rapidly acquire certification, not with a learning strategy.

Finally, although this CAMEO algorithm takes advantage of assessment features in these particular courses on this particular MOOC platform, CAMEO, as a general multiple-account-copying strategy, is possible in any MOOC with open signup policies. Generalization of the approach and its conclusions is certainly possible though arguably less scalable. Many of the courses we analyze use assessment approaches that do not involve or circumvent the *show answer* flag. From this perspective, CAMEO rates in these courses are underestimates of true CAMEO rates, and our algorithm would have to be tuned to the particular environments of these courses. For example, in an independent study tailored to a single course (Alexandron et al., 2015), 9.8% of certificate earners were identified as harvesting at least one answer.

Our estimates of cheating prevalence are arguably consistent with higher estimates from surveys. Such surveys typically ask a variant of the question "Have you cheated?" with allowance for recency and magnitude (McCabe et al., 2012). In contrast, CAMEO is complete in its scope and course-specific, as the introduction notes. The analogous question we address is, "Did you cheat your way through this entire course?" We can establish a basis for comparison through the observation from our data, that those who certify in multiple courses are much more likely to have used the CAMEO strategy at least once, including 25% of those who have earned at least 20 certificates, as depicted in Table 2.2. We consider this commensurate in severity to the reports that two-thirds of college students have engaged in some form of academic dishonesty in the previous year (McCabe et al., 2012), especially considering that the minimum threshold in our analysis is sufficient cheating to earn certification, versus being dishonest in just one or a few problems.

Number of certificates: N (Lower Bound)	Unique certificate earners with $\geq N$ certificates: M	Unique certificate earners, M, with ≥ 1 CAMEO	Percent of unique certificate earners with ≥ 1 CAMEO
1	103,370	657	1%
5	3435	185	5%
10	1262	82	6%
15	200	35	18%
20	73	18	25%
25	35	14	40%
30	15	7	47%
40	3	2	67%

Table 2.3: Rates of CAMEO among unique *high performing* users, where *high performing* defines any user that has Percent Attempts Correct $\geq 99\%$ and Fraction of Course Problems Completed ≥ 0.65.

Our findings are consistent with other observations that MOOC assessment infrastructures rarely support robust inferences about learning (Reich, 2015). All feasible mechanisms that prevent the CAMEO strategy have a downside. If instructors withhold the *show answer* option until after the problems are graded, this would constrain generally desirable asynchronous MOOC usage, and learners will not have the rapid feedback touted as a pedagogical benefit of online learning environments. Algorithmic generation of assessment items and correct answers is challenging and only suitable for some subjects and assessment tasks.

Beyond honor codes (Corrigan-Gibbs et al., 2015b; LoSchiavo and Shatz, 2011), a solution embraced by many MOOC purveyors (Eisenberg, 2013; Kolowich, 2013; Straumsheim, 2015) is to offer certificates earned under controlled assessment conditions, such as in-person assessments taken at secure testing centers for a fee. We observe that the cost and constraints associated with fee-based, in-person testing centers are antithetical to the open, online principles that define MOOCs, as well as their mission of improving worldwide access to not just learning but certification opportunities. Further research on cheating detection and prevention, including experiments that can isolate factors that cause and discourage cheating, is necessary to design spaces and structures that can support open and trustworthy certification at scale.

2.5 Summary and Contributions

The CAMEO detection algorithm uses three strategies that hold general promise for the analysis of clickstream data. First, time difference analysis is a tool to infer relationships among learners. Second, Bayesian criteria allow appropriately conservative classification when data are limited. Third, transitive closure is a technique for robust consideration of possible CAMEO users. Beyond cheating detection in MOOCs, these tools may aid more generally in identification of collaboration and interaction among online users.

There is continued interest in the potential for MOOCs to increase efficiency and spur innovation in higher education. Four features of CAMEO severely undermine this potential. First, unless prevented, this cheating strategy allows students to earn certificates in open online courses without any understanding of the domain material. Second, the strategy is highly convenient, requiring no interactions with external resources, either animate or inanimate. Third, it is unrestricted, employable in a non-selective, open admission setting. Fourth, whereas cheating is traditionally considered with respect to individual assessments or portions thereof, CAMEO is a course-level strategy. It is less cheating than the wholesale falsification of a certificate.

In this chapter, we have demonstrated the prevalence of the CAMEO cheating strategy in a large sample of MOOCs, and we have argued that it poses a serious threat to interpretations of their certifications. Protecting certification requires CAMEO prevention, and we have shown that preventive strategies hold promise. Yet, CAMEO is only one of many possible cheating strategies. Sophisticated detection algorithms should be a part of a general approach to protect the validity of online course certification. We recommend and look forward to future interventions that increase and encourage honest behavior in online learning environments while disallowing and discouraging cheating in all its forms.

Chapter 3

From CAMEO to Rank Pruning: Classification with Noisy Labels

The origins of Massive Open Online Courses (MOOCs) lie in their potential for socioeconomic mobility through education (Hansen and Reich, 2015b; Mazoue, 2014; Pappano, 2012). Yet, the viability of this outcome (Hansen and Reich, 2015a) largely depends on the reputation of the MOOC certificate as a credible academic credential (Alraimi et al., 2015), a credibility threatened by the prevalence of CAMEO cheating among learners in MOOCs (Northcutt et al., 2016).

In this chapter, you will learn about the shortcomings of the CAMEO detection algorithm and how solving the more general problem of binary classification with noisy labels addresses these shortcomings.

Along the way, in Section 3.1 you will learn three specific shortcomings of the CAMEO algorithm and the inherent challenges they pose. In Section 3.2, you will see other approaches and learn about their suitability as general solutions for multiple-account cheating detection. Finally, in Section 3.3 you will learn how Rank Pruning addresses these three shortcomings and how solving the problem of binary classification with noisy labels enables a general solution for multiple-account cheating.

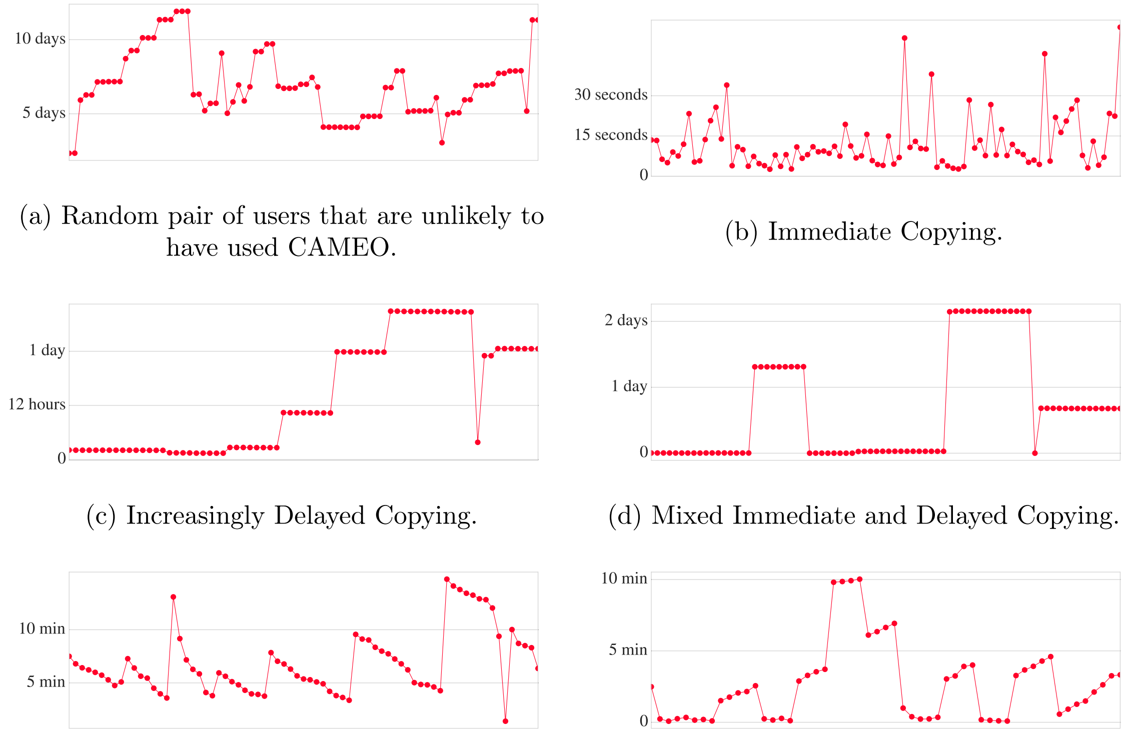
3.1 Where the CAMEO Algorithm Falls Short

In this section, I discuss the shortcomings of the CAMEO algorithm, their implications, and the need to consider other approaches.

The CAMEO cheating detection algorithm (Northcutt et al., 2016) takes a necessary first step toward detection using human-curated filters and human-tuned thresholds that operationalize the expected behavior of users employing the CAMEO strategy. However, the following key questions quickly unveil its deficiencies: Why are there five filters? Why not three, or eighteen? If a machine could infer five disjunctive filters using true CAMEO cheating labels and all interaction data, would it uncover the same five criteria? Would we not prefer to use the machine-learned criteria? Given how drastically MOOCs can vary, does it make sense to use the same thresholds across every course? Is it possible that the "90th Percentile of Δt< 5 minutes" filter is inappropriate for a uniquely fast-paced course? MOOCs and cheating strategies adapt over time, so why are the thresholds static?

These questions highlight three important shortcomings of the CAMEO algorithm. First, threshold parameters are not course-specific or time-specific. Parameters are not relearned even though courses, cheating strategies, and learner interactions vary over time. This problem is known as *domain shift* in semi-supervised learning because the test and training sets have *shifted* apart and can drastically reduce the accuracy of predictions (Jiang, 2008). Second, the filters and thresholds are chosen to operationalize expected CAMEO behavior, but to avoid human biases, they should instead be learned, or at least influenced, by data. Third, the CAMEO algorithm is a lower-bound estimate for cheating because it uses overly conservative thresholds to reduce the risk of false positives, leaving the number of false negatives unbounded. For example, a CAMEO user who delayed submission of just 10% of copied answers by six minutes will not be detected. Alas, even with conservative thresholds the CAMEO algorithm cannot guarantee the absence of false positives, and can only claim to make noisy predictions.

The third shortcoming is a manifestation of the variation of CAMEO usage, as



(a) Random pair of users that are unlikely to have used CAMEO.

(b) Immediate Copying.

(c) Increasingly Delayed Copying.

(d) Mixed Immediate and Delayed Copying.

(e) Delayed Copying in batches of 8-10. User submits faster than harvests.

(f) Delayed Copying in batches of 4. User submits slower than harvests.

Figure 3-1: Six examples of observed CH-CM pair Δt distributions for a suspected non-cheater in (a) and five known CAMEO cheaters in (b-f). Each plot represents one pair of accounts. The y-axis of all graphs depicts the Δt in varying units of time. A red point is plotted for each $\Delta t > 0$. The points are ordered horizontally along the x-axis by order of submission.

shown in Fig. 3.1. Each sub-figure captures the behavior of a unique pair of accounts in an MITx Calculus course by depicting the Δt, or time passed from when the candidate *harvester* account acquires the correct solution by clicking *show answer* to when the candidate *master* account submits a correct answer, for every assessment question where the *master* account clicked *show answer* and the *harvester* account answered correctly, ordered horizontally along the x-axis by order of submission. Except for sub-figure (a), each sub-figure depicts a known cheater, verified using the validation method described in Section 2.2.3. Observe the variations of CAMEO usage and note that apart from sub-figure (b), the CAMEO algorithm fails to detect cheating in every case. Sub-figure (c) depicts a CAMEO user changing strategy over time and sub-figure (d) depicts a CAMEO user switching back and forth between strategies.

Sub-figures (e) and (f) show that even within a single strategy, variation in CAMEO usage exists.

As a solution, I could introduce additional conjunctive and disjunctive filters, each with human-tuned thresholds, adding a new rule for each variation. With some reluctance, I admit to having done this, implementing hundreds of rules by hand, and although the quality of predictions improved, the three shortcomings of the original CAMEO algorithm remained. A better approach is to learn the mapping from learner interactions to cheating labels directly from data, avoiding human supposition.

To address the three shortcomings of the CAMEO algorithm in a supervised learning framework, a natural next step might be to view the five disjunctive filters of the CAMEO algorithm as an ensemble of decision trees (Quinlan, 1986; Dietterich, 2000) to be learned using a random forests approach (Liaw and Wiener, 2002). This would eliminate the need for human-curated filters and thresholds. Unfortunately, a random forests decision boundary cannot be learned in a traditional supervised manner without access to correctly-labeled CAMEO cheating examples. Because the CAMEO algorithm can only produce noisy predicted labels, a different approach is needed.

Instead of thinking about how to improve the CAMEO algorithm directly, can we use it to generate noisy predicted labels? Then it is suffices to solve the problem of binary classification with noisy labels.

The above question addresses the goal of this chapter: to explain the choice to address the problem of binary classification with noisy labels instead of improving the CAMEO algorithm or using another approach entirely.

3.2 A Good Approach is Choosing No Approach

In light of the shortcomings of the CAMEO algorithm, in this section I consider the suitability of other natural approaches for multiple-account cheating detection. Although these approaches may perform well in certain conditions, I reveal that most approaches inherently depend on a human-chosen threshold to determine the cutoff for labeling a learner a cheater and none provide a general solution for all three shortcomings of the CAMEO algorithm. To conclude, I posit a culminating, rhetorical question to suggest *why a good approach is choosing no approach.*

Remote webcam monitoring has successfully aided in cheating detection during online examinations (Kolowich, 2013; Li et al., 2015). However, these techniques often depend on a combination of cameras and body-mounted sensors worn by every learner. Aside from scalability and economical feasibility, these methods require visual detection of external resources, e.g. a *cheat-sheet*, which is not a necessary condition for CAMEO users. Given that CAMEO detection is intended to enable continued *open-access*, I instead consider approaches that do not impose access restrictions due to camera and sensor availability limitations.

Item Response Theory (IRT) is a latent-trait model useful for estimating learner proficiency and problem difficulty, and by consequence, detecting copying on computerized exams. IRT can be used to detect copying via an additional ω statistic (Wollack, 2004), but at least 20%-30% of answers must be copied to detect cheating (Wollack, 1997). IRT is most appropriate in the context of standardized testing (Embretson and Reise, 2013), where examination occurs in a timed setting, with low-variance populations. MOOC populations and courses differ greatly, where users may work asynchronously, and courses may vary in type, duration, content, format, and purpose. More sophisticated variants like the deterministic gated IRT model are well-designed for cheating detection, but like the CAMEO algorithm's third shortcoming, ultimately require a cheating probability threshold cut-off to decide what constitutes collaborative cheating.

SPARse Factor Analysis (SPARFA) provides an alternative to IRT by estimating user proficiency per learning concept (Lan et al., 2014), improving on the Wesolowsky method which is limited to multiple-choice problems (Galambos, 1977; Worsley, 1982; Wesolowsky, 2000). The SPARFA framework takes a Bayesian approach, using MCMC sampling (Gilks, 2005) to infer collaborative communities (Waters et al., 2013). Waters et al. (2013) use SPARFA to identify *parasitic* collaborations in two undergraduate courses that use the Open-Stax Tutor (https://openstaxtutor.org/;

Waters et al., 2014). By introducing model parameters for every concept, SPARFA can detect collaboration more accurately than IRT in certain contexts, but the fully Bayesian approach over all pairwise collaborations imposes limits on scalability, particularly in the context of MOOCs.

SPARFA differs from CAMEO by learning communities directly from data by introducing a hyperparameter, K, for the number of concepts. Additionally, a collaboration is deemed parasitic when two learners in a community perform *worse* on an individual assessment task compared with a third learner in the same community. *How much worse* must they perform to be labeled as a cheater? This becomes another threshold facing the same false positive, false negative trade-off challenges as the third shortcoming of the CAMEO algorithm.

An entirely different approach is the *honeypot*, an enticing online resource created for the purpose of attracting and identifying adversarial accounts. Honeypots have proven effective at detecting system vulnerabilities (Provos et al., 2004), spam (Andreolini et al., 2005; Zook, 2007), and malware (Baecher et al., 2006).

A honeypot can be used to obtain true-positive cheating labels in MOOCs.

You have already seen a honeypot used in a small-scale validation experiment in chapter 2. In this experiment, decimal answers were extended with uniquely identifying digits, pairing harvester and master accounts at the time of correct answer submission. In a different context in the massively empowered classroom MOOC platform, a honeypot was used to obtain the first true-positive cheating labels in MOOCs (Corrigan-Gibbs et al., 2015a,b) by hosting an easily-searchable, imitation solutions-repository website during an online course examination and identifying users who visited the site via browser-cookies. Both of these honeypots yielded true-positive cheating labels, but no true-negative cheating labels.

The problem of binary classification when a subset of true-positive labels, but no true-negative labels are available is referred to as Positive-Unlabeled (PU) learning. Methods for PU learning have successfully tackled challenging problems like fraud detection (Phua et al., 2004), protein identification (Elkan and Noto, 2008), and intrusion detection (Mohamed et al., 2012). By considering the unlabeled examples as a noisy negative class, PU learning is a simpler version of the general problem of binary classification with noisy labels. However, even with true-positive cheating labels, it is possible to have many false-negatives. In practice, obtaining true-positive labels using honeypots requires a technical infrastructure that may be limiting for some course instructors, whereas acquiring noisy CAMEO labels may be simpler. A general solution to binary classification with noisy labels is needed.

3.3 The Rank Pruning Approach

The Rank Pruning algorithm receives a comprehensive treatment in the next chapter. Instead, in this section I discuss the benefits of the Rank Pruning approach and how they address the shortcomings of the CAMEO algorithm.

Rank Pruning is a robust, time-efficient, general solution for $\tilde{P}\tilde{N}$ learning, i.e. binary classification with noisy labels. Its works by fitting a classifier on a subset of the training data that it is confident is labeled correctly, avoiding training with mislabeled data. Rank Pruning also estimates the noise rates, i.e. the fraction of mislabeled positive examples and the fraction of mislabeled negative examples. It turns out that under some assumptions, Rank Pruning achieves equivalent expected risk as learning with correct labels and perfect noise rate estimation, with closed-form solutions when those assumptions are relaxed.

In the context of Rank Pruning, the CAMEO algorithm can be understood as a tool for generating noisy CAMEO cheating labels, where the goal of Rank Pruning is to train a classifier on a subset of examples that it is confident are labeled correctly. But Rank Pruning offers another valuable feature: the noise rates. For every course that CAMEO analyzes, Rank Pruning can estimate the noise rates, producing a quantitative statistic for how well the CAMEO algorithm does. In this way, Rank Pruning provides a measure of the error of CAMEO cheating detection, a fundamental contribution.

Rank Pruning addresses the three of the shortcomings of the CAMEO algorithm.

First, Rank Pruning enables adaptivity and course-specificity. Rank Pruning is applicable to any course for which the CAMEO algorithm generates labels, but unlike CAMEO which has static filters and thresholds, Rank Pruning estimates course-specific noise rates and trains a course-specific classifier. Second, Rank Pruning is not reliant on human-chosen filters and thresholds, but learns a decision boundary from a subset of correctly labeled CAMEO cheating examples. Moreover, Rank Pruning is classifier independent. If in ten years, a revolutionary probabilistic classifier is invented, Rank Pruning can use it just as easily as it can use the simplest logistic regression classifier. Third, no algorithm can claim perfect cheating classification in every problem space, the essence of Rank Pruning is to directly address the issue of noisy labels by uncovering a subset of the correctly labeled training data.

The Rank Pruning algorithm is a good approach for these shortcomings because it works independently of how the labels are generated. It is not restricted to any single approach or classifier, nor tied to any physical or technical constraints. Beyond binary cheating predictions, it produces probabilities of cheating for every learner, and for every course, it provides noise rates to estimate the error of the CAMEO algorithm, or any other approach for generating noisy cheating labels.

Instead of solving the complex problem of choosing an approach for labeling cheaters, Rank Pruning trains a classifier using confidently labeled cheaters for any labeling approach, such as the CAMEO algorithm.

In the next chapter, you will learn the principle of *learning from confident examples.*

Chapter 4

Learning with Confident Examples: Rank Pruning for Robust Classification with Noisy Labels

In this chapter [1], you will learn about Rank Pruning, a solution for robust binary classification when training data may be mislabeled and/or contain noise examples drawn from an unrelated distribution. Along the way, you will learn the concept of *learning from confident examples* and how this principle enables robust classification with noisy labels.

In Section 4.1, you will learn about $\tilde{P}\tilde{N}$ learning, the problem of binary classification when training examples may be mislabeled (flipped) uniformly with noise rate ρ_1 for positive examples and ρ_0 for negative examples, and in Section 4.2 you will see a rigorous framing of the $\tilde{P}\tilde{N}$ learning problem as it relates to Rank Pruning.

You will see each step of the Rank Pruning algorithm explained in Section 4.3, along with proofs showing that that Rank Pruning achieves consistent noise estimation and equivalent expected risk as learning with uncorrupted labels in ideal conditions, with closed-form solutions when conditions are non-ideal. To simplify reading,

[1] Tailin Wu, MIT, and Isaac Chuang, MIT, contributed significantly to the contents of this chapter. I elect the use of the pronoun *we* throughout to emphasize their greatly appreciated contributions.

in Section 4.3.4 you will see the entire Rank Pruning algorithm explained in only a few sentences.

By way of illustration, in Section 4.4 you will see how Rank Pruning works on a simple two-dimensional example and how Rank Pruning achieves state-of-the-art noise rate estimation and F1, error, and AUC-PR on the MNIST and CIFAR datasets, regardless of noise rates. You may be surprised to find that with a CNN classifier, Rank Pruning can predict if a MNIST digit is a *one* or *not* with only 0.25% error, and 0.46% error across all digits, even when 50% of positive examples are mislabeled and 50% of observed positive labels are mislabeled negative examples.

By the end of this chapter, you will know how Rank Pruning solves the $\tilde{P}\tilde{N}$ learning problem and the open problem of estimating the noise rates. You will know more about why Rank Pruning is a time-efficient and general solution, requiring $\mathcal{O}(T)$ for any unrestricted choice of probabilistic classifier with T fitting time. Finally, you will know how *learning with confident examples* enables robust noise rate estimation and classification with noisy labels.

4.1 Introduction

Consider a student with no knowledge of any animal, tasked with learning to classify whether an image contains a dog. A teacher shows the student example images depicting one of ten animals which may overlap in appearance, stating either (1) the image contains a dog or (0) the image does not contain a dog. Unfortunately, the teacher may often make mistakes, asymmetrically, with a significantly large false positive rate, $\rho_1 \in [0, 1]$, and significantly large false negative rate, $\rho_0 \in [0, 1]$. Additionally, the teacher may include “white noise" images with a uniformly random label. This information is hidden from the student, who knows only the examples and corrupted labels. However, the student suspects that the teacher may make mistakes. Can the student (1) estimate ρ_1 and ρ_0, i.e. asymmetrically how likely a mistake is made, (2) still learn to classify images containing a dog with high accuracy, and (3) do so efficiently (e.g. less than an hour for 50 images)? This allegory clarifies the challenges of $\tilde{P}\tilde{N}$ learning for any classifier trained with corrupted labels, perhaps with intermixed noise examples. We elect the notation $\tilde{P}\tilde{N}$ to emphasize that both the positive and negative sets may contain mislabeled examples, reserving P and N for uncorrupted sets.

This example illustrates a fundamental reliance of supervised learning on training labels (Michalski et al., 1986). If training examples may be mislabeled or noisy, traditional learning performance degrades monotonically with noise (Aha et al., 1991; Nettleton et al., 2010), necessitating semi-supervised approaches (Blanchard et al., 2010). Vital examples of noisy datasets are medical (Raviv and Intrator, 1996), human-labeled (Paolacci et al., 2010), and sensor (Lane et al., 2010) datasets. The problem of uncovering the same classifications as if the data was not mislabeled is the essential goal of Rank Pruning.

Towards this goal, we introduce Rank Pruning [2], an algorithm for $\tilde{P}\tilde{N}$ learning composed of two sequential parts: (1) asymmetric noise estimation of ρ_1 and ρ_0 and (2) removal of predicted mislabeled examples prior to training. The fundamental mantra of Rank Pruning is estimation and prediction with *confident examples*, i.e. examples with a predicted probability of being positive *near* 1 when the training label is positive or 0 when the training label is negative. If we imagine non-confident examples as a third, noise class, separate from the confident positive and negative classes, then removal of this third class should unveil a subset of the uncorrupted data.

An ancillary mantra of Rank Pruning is *removal by rank* which elegantly exploits ranking without ever sorting. Instead of pruning non-confident examples based on predicted probability, we estimate the number of mislabeled examples in each class and remove the k^{th}-most or k^{th}-least examples, *ranked* by predicted probability. This reduces to finding the k^{th} largest (or smallest) predicted probability, solved by the BFPRT algorithm (Blum et al., 1973) in $\mathcal{O}(n)$ time, where n is the number of training examples. *Removal by rank* mitigates sensitivity to probability estimation and exploits the reduced complexity of learning to rank over probability estimation (Menon

[2] Rank Pruning is open-source and available at https://github.com/cgnorthcutt/rankpruning

et al., 2012). We use both *learning with confident examples* and *removal by rank* for the purpose of robustness, i.e. invariance to erroneous input deviation.

In essence, confident examples are those having agreement of predicted probability and training label. If an example has a positive label, but a low predicted probability of being positive, we prune out this training example from the dataset. The intuition is that non-confident examples are likely to be mislabeled and their removal may unveil a subset of the true, uncorrupted distribution. If so, then the expected risk of training on the pruned set is equivalent to training on the correctly labeled data.

Beyond prediction, confident examples can be used to estimate ρ_1 and ρ_0. Typical approaches require averaging predicted probabilities on a holdout set (Liu and Tao, 2016; Elkan and Noto, 2008). However, this ties the estimation of ρ_1 and ρ_0 to the accuracy of the predicted probabilities, which in practice may be confounded by added noise or poor model selection. Instead, we estimate ρ_1 and ρ_0 as a fraction of the predicted counts of confident examples in each class, encouraging robustness for variation in probability estimation.

4.1.1 Related Work

Rank Pruning bridges framework, nomenclature, and application across PU and $\tilde{P}\tilde{N}$ learning. In this section, we consider the contributions of Rank Pruning in both.

4.1.1.1 PU Learning

Positive-unlabeled (PU) learning is a related binary classification problem in which a subset of positive training examples are labeled, and the rest are unlabeled. For example, co-training (Blum and Mitchell, 1998; Nigam and Ghani, 2000) with labeled and unlabeled examples can be framed as a PU learning problem by assigning all unlabeled examples the label 0. PU learning methods often assume corrupted negative labels for the unlabeled examples U such that PU learning is $\tilde{P}\tilde{N}$ learning with perfectly labeled P, hence their naming conventions.

Early approaches to PU learning modified the loss functions via weighted logis-

Table 4.1: Variable definitions and descriptions for $\tilde{P}\tilde{N}$ learning and PU learning. Related work contains a prominent author using each variable. ρ_1 is also referred to as *contamination* in PU learning literature.

Variable	Conditional	Description	Domain	Related Work
ρ_0	$P(s=1 \mid y=0)$	Fraction of N examples mislabeled as positive	$\tilde{P}\tilde{N}$	Liu
ρ_1	$P(s=0 \mid y=1)$	Fraction of P examples mislabeled as negative	$\tilde{P}\tilde{N}$, PU	Liu, Claesen
π_0	$P(y=1 \mid s=0)$	Fraction of mislabeled examples in $\tilde{N}$	$\tilde{P}\tilde{N}$	Scott
π_1	$P(y=0 \mid s=1)$	Fraction of mislabeled examples in $\tilde{P}$	$\tilde{P}\tilde{N}$	Scott
$c = 1 - \rho_1$	$P(s=1 \mid y=1)$	Fraction of correctly labeled P if $P(y=1 \mid s=1) = 1$	PU	Elkan

tic regression (Lee and Liu, 2003) and biased SVM (Liu et al., 2003) to penalize more when positive examples are predicted incorrectly. Bagging SVM (Mordelet and Vert, 2014) and RESVM (Claesen et al., 2015) extended biased SVM to instead use an ensemble of classifiers trained by resampling U (and P for RESVM) to improve robustness (Breiman, 1996). RESVM claims state-of-the-art for *PU* learning, but is impractically inefficient for large datasets because it requires optimization of five parameters and suffers from the pitfalls of model selection for SVM (Chapelle and Vapnik, 1999). Elkan and Noto (2008) introduce a formative time-efficient probabilistic approach (denoted *Elk08*) for *PU* learning that directly estimates $1 - \rho_1$ by averaging predicted probabilities of a holdout set and dividing all predicted probabilities by $1 - \rho_1$. On the SwissProt database, *Elk08* was 621 times faster than biased SVM, which only requires two parameter optimization. However, *Elk08* noise rate estimation is highly sensitive to inexact probability estimation. Both RESVM and *Elk08* assume $P = \tilde{P}$ and do not generalize to $\tilde{P}\tilde{N}$ learning. Rank Pruning leverages *Elk08* to initialize ρ_1, but then re-estimates ρ_1 using confident examples for both robustness (RESVM) and efficiency (*Elk08*).

4.1.1.2 $\tilde{P}\tilde{N}$ Learning

Theoretical approaches for $\tilde{P}\tilde{N}$ learning often have two steps: (1) estimation of noise rates ρ_1 and ρ_0 and (2) incorporation of noise rates for prediction. To our knowledge, Rank Pruning is the only time-efficient solution for the open problem (Liu and Tao, 2016; Yang et al., 2012) of noise rate estimation.

We first consider relevant work in noise rate estimation. Scott et al. (2013) established a lower bound method for estimating the *inversed* noise rates π_1 and π_0

(defined in Table 4.1). However, the method can be intractable due to unbounded convergence and assumes that the positive and negative distributions are mutually irreducible. Under additional assumptions, Scott (2015) proposed a time-efficient method for estimation of π_1 and π_0, but when implemented by Liu and Tao (2016), this method resulted in poor performance. Liu and Tao (2016) instead used the minimum predicted probabilities as the noise rates. Although trivially efficient, in practice this often yields futile estimates of min = 0. Natarajan et al. (2013) provide no method for estimation and view the noise rates as parameters optimized with cross-validation, inducing a sacrificial accuracy, efficiency trade-off. In comparison, Rank Pruning noise rate estimation is time-efficient, consistent in ideal conditions, and robust to imperfect probability estimation.

Natarajan et al. (2013) developed two methods for prediction in the $\tilde{P}\tilde{N}$ setting which modify the loss function. The first method constructs an unbiased estimator of the loss function for the true distribution from the noisy distribution, but the estimator may be non-convex even if the original loss function is convex. If the classifier's loss function cannot be modified directly, this method requires splitting each example in two with class-conditional weights and ensuring split examples are in the same batch during optimization. For these reasons, we instead compare Rank Pruning with their second method (*Nat13*), which constructs a label-dependent loss function such that for 0-1 loss, the minimizers of *Nat13*'s risk and the risk for the true distribution are equivalent.

Liu and Tao (2016) generalized *Elk08* to the $\tilde{P}\tilde{N}$ learning setting by modifying the loss function with per-example importance reweighting (*Liu16*), but reweighting terms are derived from predicted probabilities which may be sensitive to inexact estimation. To mitigate sensitivity, Liu and Tao (2016) examines the use of density ratio estimation (Sugiyama et al., 2012). Instead, Rank Pruning mitigates sensitivity by learning from confident examples selected by rank order, not predicted probability. For fairness of comparison across methods, we compare Rank Pruning with their probability-based approach.

Assuming perfect estimation of ρ_1 and ρ_0, we, Natarajan et al. (2013), and Liu

Table 4.2: Summary of state-of-the-art and selected general solutions to $\tilde{P}\tilde{N}$ and PU learning for the following features, from left-to-right: (1) noise rate estimation, (2) a solution to $\tilde{P}\tilde{N}$ learning, (3) a solution to PU learning, (4) works for any probabilistic classifier, (5) robustness to poor probability estimation, (6) time-efficent for some choice of classifier, (7) theoretical justification, and (8) works with added noise drawn from a third distribution mixed into the train set. A checkmark indicates the feature is supported by the method.

Related Work	Noise Estim.	$\tilde{P}\tilde{N}$	PU	Any Prob. Classifier	Prob Estim. Robustness	Time Efficient	Theory Support	Added Noise
Elkan and Noto (2008)	✓		✓	✓		✓	✓	
Claesen et al. (2015)			✓		✓			
Scott et al. (2013)	✓			✓	✓		✓	
Natarajan et al. (2013)		✓	✓	✓	✓	✓	✓	
Liu and Tao (2016)		✓	✓	✓		✓	✓	
Rank Pruning	✓	✓	✓	✓	✓	✓	✓	✓

and Tao (2016) all prove that the expected risk for their modified loss function is equivalent to the expected risk for the perfectly labeled dataset given perfect ρ_1 and ρ_0. However, both Natarajan et al. (2013) and Liu and Tao (2016) effectively "flip" example labels in the construction of their loss function, providing no benefit for added random noise. In comparison, Rank Pruning will also remove added random noise because noise drawn from a third distribution is unlikely to appear confidently positive or negative. A summarizing comparison of $\tilde{P}\tilde{N}$ and PU learning methods is distilled in Table 4.2.

Procedural efforts have improved robustness to mislabeling in the context of machine vision (Xiao et al., 2015), neural networks (Reed et al., 2015), and face recognition (Angelova et al., 2005). Though promising, these methods are restricted in theoretical justification and generality, motivating the need for Rank Pruning.

4.2 Framing the $\tilde{P}\tilde{N}$ Learning Problem

In this section, we formalize the foundational definitions, assumptions, and goals of the $\tilde{P}\tilde{N}$ learning problem illustrated by the student-teacher motivational example.

Given n observed training examples $x \in \mathcal{R}^D$ with associated observed corrupted labels $s \in \{0, 1\}$ and unobserved true labels $y \in \{0, 1\}$, we seek a binary classifier f that estimates the mapping $x \to y$. Unfortunately, if we fit the classifier using observed (x, s) pairs, we estimate the mapping $x \rightarrow s$ and obtain $g(x) = P(\hat{s} = 1|x)$.

We define the observed noisy positive and negative sets as $\tilde{P} = \{x|s = 1\}, \tilde{N} = \{x|s = 0\}$ and the unobserved true positive and negative sets as $P = \{x|y = 1\}, N = \{x|y = 0\}$. Define the hidden training data as $D = \{(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)\}$, drawn i.i.d. from some true distribution $\mathcal{D}$. We assume that a class-conditional Classification Noise Process (CNP) (Angluin and Laird, 1988) maps y true labels to s observed labels such that each label in P is flipped independently with probability ρ_1 and each label in N is flipped independently with probability ρ_0 ($s \leftarrow CNP(y, \rho_1, \rho_0)$). The resulting observed, corrupted dataset is $D_\rho = \{(x_1, s_1), (x_2, s_2), ..., (x_n, s_n)\}$. Therefore, $(s \perp\!\!\!\perp x)|y$ and $P(s = s|y = y, x) = P(s = s|y = y)$. In recent work, CNP is referred to as the random noise classification (RCN) noise model (Liu and Tao, 2016; Natarajan et al., 2013).

The noise rate $\rho_1 = P(s = 0|y = 1)$ is the fraction of P examples mislabeled as negative and the noise rate $\rho_0 = P(s = 1|y = 0)$ is the fraction of N examples mislabeled as positive. Note that $\rho_1 + \rho_0 < 1$ is a necessary condition, otherwise more examples would be mislabeled than labeled correctly. Thus, $\rho_0 < 1 - \rho_1$. We elect a subscript of "0" to refer to the negative set and a subscript of "1" to refer to the positive set. Additionally, let $p_{s1} = P(s = 1)$ be the fraction of corrupted labels that are positive and $p_{y1} = P(y = 1)$ be the fraction of true labels that are positive. It follows that the inversed noise rates are $\pi_1 = P(y = 0|s = 1) = \frac{\rho_0(1-p_{y1})}{p_{s1}}$ and $\pi_0 = P(y = 1|s = 0) = \frac{\rho_1 p_{y1}}{(1-p_{s1})}$. Combining these relations, given any pair in $\{(\rho_0, \rho_1), (\rho_1, \pi_1), (\rho_0, \pi_0), (\pi_0, \pi_1)\}$, the remaining two and p_{y1} are known.

We consider five levels of assumptions for P, N, and g:

Perfect Condition: g is a "perfect" probability estimator iff $g(x) = g^*(x)$ where $g^*(x) = P(s = 1|x)$. Equivalently, let $g(x) = P(s = 1|x) + \Delta g(x)$. Then $g(x)$ is "perfect" when $\Delta g(x) = 0$ and "imperfect" when $\Delta g(x) \neq 0$. g may be imperfect due to the method of estimation or due to added uniformly randomly labeled examples drawn from a third noise distribution.

Non-overlapping Condition: P and N have "non-overlapping support" if P and N have non-overlapping distributions, or formally, if $P(y = 1|x) = \mathbb{1}[[y = 1]]$, where the indicator function $\mathbb{1}[[\cdot]]$ is 1 if the expression is true, and 0 otherwise.

Ideal Condition[3]: g is "ideal" when both perfect and non-overlapping conditions hold and $(s \perp\!\!\!\perp x)|y$ such that

$$\begin{aligned} g(x) =& g^*(x) = P(s=1|x) \\ &= P(s=1|y=1,x) \cdot P(y=1|x) + P(s=1|y=0,x) \cdot P(y=0|x) \\ &= (1-\rho_1) \cdot \mathbb{1}[[y=1]] + \rho_0 \cdot \mathbb{1}[[y=0]] \end{aligned} \tag{4.1}$$

Range Separability Condition g range separates P and N iff $\forall x_1 \in P$ and $\forall x_2 \in N$, we have $g(x_1) > g(x_2)$.

Unassuming Condition: g is "unassuming" when perfect and/or non-overlapping conditions may not be true.

Their relationship is: **Unassuming** $\supset$ **Range Separability** $\supset$ **Ideal** $=$ **Perfect**$\cap$ **Non-overlapping**.

We can now state the two goals of Rank Pruning for $\tilde{P}\tilde{N}$ learning. **Goal 1** is to perfectly estimate $\hat{\rho}_1 \triangleq \rho_1$ and $\hat{\rho}_0 \triangleq \rho_0$ when g is ideal. When g is not ideal, to our knowledge perfect estimation of ρ_1 and ρ_0 is impossible and at best **Goal 1** is to provide exact expressions for $\hat{\rho}_1$ and $\hat{\rho}_0$ w.r.t. ρ_1 and ρ_0. **Goal 2** is to use $\hat{\rho}_1$ and $\hat{\rho}_0$ to uncover the classifications of f from g. Both tasks must be accomplished given only observed (x, s) pairs. y, ρ_1, ρ_0, π_1, and π_0 are hidden.

4.3 Rank Pruning

We develop the Rank Pruning algorithm to address our two goals. In Section 4.3.1, we propose a method for noise rate estimation and prove consistency when g is ideal. An estimator is *consistent* if it achieves perfect estimation in the expectation of infinite examples. In Section 4.3.2, we derive exact expressions for $\hat{\rho}_1$ and $\hat{\rho}_0$ when g is unassuming. In Section 4.3.3, we propose Rank Pruning, and in Section 4.3.5, prove that the expected risk for the modified loss function induced by Rank Pruning for D_ρ

[3] Eq. (4.1) is first derived in (Elkan and Noto, 2008) .

is equivalent to the expected risk of the loss function for the hidden training data D for both ideal g and non-ideal g with weaker assumptions.

Throughout Section 4.3, we assume $n \to \infty$ so that P and N are the true, hidden distributions, each containing infinite examples. This is a necessary condition for Theorems 2 and 4 and Lemmas 1 and 3.

4.3.1 Deriving Noise Rate Estimators $\hat{\rho}_1^{conf}$ and $\hat{\rho}_0^{conf}$

We propose the *confident counts* estimators $\hat{\rho}_1^{conf}$ and $\hat{\rho}_0^{conf}$ to estimate ρ_1 and ρ_0 as a fraction of the predicted counts of confident examples in each class, encouraging robustness for variation in probability estimation.

Define the *confidence* of an example, x, as $g(x) \cdot \mathbb{1}[[s = 1]] + (1 - g(x)) \cdot \mathbb{1}[[s = 0]]$. The *confident counts* estimators embody the *learning with confident examples* principle. Intuitively, to estimate $\rho_1 = P(s = 0|y = 1)$ we count the number of examples that we are confident belong to $s = 0$ and $y = 1$ and divide it by the number of examples that we are confident belong to $y = 1$. More formally, we propose the following method to obtain $\hat{\rho}_1^{conf}$ and $\hat{\rho}_0^{conf}$.

Fit g to the corrupted training set D_ρ to obtain $g(x) = P(\hat{s} = 1|x)$. Then, use $g(x)$ to obtain $LB_{y=1}$, the predicted probability in $g(x)$ above which we guess that an example x has hidden label $y = 1$, and $UB_{y=0}$, the predicted probability in $g(x)$ below which we guess x has hidden label $y = 0$. $LB_{y=1}$ and $UB_{y=0}$ partition $\tilde{P}$ and $\tilde{N}$ into four sets representing a *best guess* of a *subset* of examples having labels (1) $s = 1, y = 0$, (2) $s = 1, y = 1$, (3) $s = 0, y = 0$, (4) $s = 0, y = 1$. They are defined as

$$\begin{cases} LB_{y=1} := P(\hat{s} = 1 \mid s = 1) = E_{x \in \tilde{P}}[g(x)] \\ UB_{y=0} := P(\hat{s} = 1 \mid s = 0) = E_{x \in \tilde{N}}[g(x)] \end{cases}$$

where $\hat{s}$ is the predicted label from a classifier fit to the observed data. Then the *confident counts* estimators are

$$\hat{\rho}_1^{conf} := \frac{|\tilde{N}_{y=1}|}{|\tilde{N}_{y=1}| + |\tilde{P}_{y=1}|}, \quad \hat{\rho}_0^{conf} := \frac{|\tilde{P}_{y=0}|}{|\tilde{P}_{y=0}| + |\tilde{N}_{y=0}|} \tag{4.2}$$

where

$$\begin{cases} \tilde{P}_{y=1} = \{x \in \tilde{P} \mid g(x) \geq LB_{y=1}\} \\ \tilde{N}_{y=1} = \{x \in \tilde{N} \mid g(x) \geq LB_{y=1}\} \\ \tilde{P}_{y=0} = \{x \in \tilde{P} \mid g(x) \leq UB_{y=0}\} \\ \tilde{N}_{y=0} = \{x \in \tilde{N} \mid g(x) \leq UB_{y=0}\} \end{cases} \tag{4.3}$$

Intuitively, $|\tilde{P}_{y=1}|$ counts the examples with label $s = 1$ that are *most* likely to be correctly labeled since $LB_{y=1} = P(\hat{s} = 1|s = 1)$. The three other terms follow similar reasoning. Importantly, the four terms in Eq. (4.3) do not sum to n, i.e. $|N|+|P|$, but $\hat{\rho}_1^{conf}$ and $\hat{\rho}_0^{conf}$ are valid estimates because mislabeling noise is assumed to be uniformly random. The choice of threshold values relies on the following two important equations:

$$\begin{aligned} LB_{y=1} =& E_{x\in\tilde{P}}[g(x)] = E_{x\in\tilde{P}}[P(s=1|x)] \\ =& E_{x\in\tilde{P}}[P(s=1|x,y=1)P(y=1|x) + P(s=1|x,y=0)P(y=0|x)] \\ =& E_{x\in\tilde{P}}[P(s=1|y=1)P(y=1|x) + P(s=1|y=0)P(y=0|x)] \\ =& (1-\rho_1)(1-\pi_1) + \rho_0\pi_1 \end{aligned} \tag{4.4}$$

Similarly, we have

$$UB_{y=0} = (1-\rho_1)\pi_0 + \rho_0(1-\pi_0) \tag{4.5}$$

To our knowledge, although simple, this is the first time that the relationship in Eq. (4.4) (4.5) has been published, linking the work of Elkan and Noto (2008), Liu and Tao (2016), Scott et al. (2013) and Natarajan et al. (2013). From Eq. (4.4) (4.5), we observe that $LB_{y=1}$ and $UB_{y=0}$ are linear interpolations of $1-\rho_1$ and ρ_0 and since $\rho_0 < 1-\rho_1$, we have that $\rho_0 < LB_{y=1} \leq 1-\rho_1$ and $\rho_0 \leq UB_{y=0} < 1-\rho_1$. When g is ideal we have that $g(x) = (1-\rho_1)$, if $x \in P$ and $g(x) = \rho_0$, if $x \in N$. Thus when g is ideal, the thresholds $LB_{y=1}$ and $UB_{y=0}$ in Eq. (4.3) will perfectly separate P and N examples within each of $\tilde{P}$ and $\tilde{N}$. Lemma 1 immediately follows.

Lemma 1 *When g is ideal,*

$$\begin{aligned}&\tilde{P}_{y=1} = \{x \in P \mid s = 1\}, \tilde{N}_{y=1} = \{x \in P \mid s = 0\},\\ &\tilde{P}_{y=0} = \{x \in N \mid s = 1\}, \tilde{N}_{y=0} = \{x \in N \mid s = 0\}\end{aligned} \tag{4.6}$$

Thus, when g is ideal, the thresholds in Eq. (4.3) partition the training set such that $\tilde{P}_{y=1}$ and $\tilde{N}_{y=0}$ contain the correctly labeled examples and $\tilde{P}_{y=0}$ and $\tilde{N}_{y=1}$ contain the mislabeled examples. Theorem 2 follows (for brevity, proofs of all theorems/lemmas are in Appendix A.1.1-A.1.5).

Theorem 2 *When g is ideal,*

$$\hat{\rho}_1^{conf} = \rho_1, \hat{\rho}_0^{conf} = \rho_0 \tag{4.7}$$

Thus, when g is ideal, the *confident counts* estimators $\hat{\rho}_1^{conf}$ and $\hat{\rho}_0^{conf}$ are consistent estimators for ρ_1 and ρ_0 and we set $\hat{\rho}_1 := \hat{\rho}_1^{conf}, \hat{\rho}_0 := \hat{\rho}_0^{conf}$. These steps comprise Rank Pruning noise rate estimation (see Alg. 1). There are two practical observations. First, for any g with T fitting time, computing $\hat{\rho}_1^{conf}$ and $\hat{\rho}_0^{conf}$ is $\mathcal{O}(T)$. Second, $\hat{\rho}_1$ and $\hat{\rho}_0$ should be estimated out-of-sample to avoid over-fitting, resulting in sample variations. In our experiments, we use 3-fold cross-validated probabilities, requiring at most $2T = \mathcal{O}(T)$.

4.3.2 Noise Estimation for Unassuming Condition

Theorem 2 states that $\hat{\rho}_i^{conf} = \rho_i$, $\forall i \in \{0, 1\}$ when g is ideal. Though theoretically constructive, in practice this is unlikely. Next, we derive expressions for the estimators when g is unassuming, i.e. g may not be perfect and P and N may have overlapping support.

Define $\Delta p_o := \frac{|P \cap N|}{|P \cup N|}$ as the fraction of overlapping examples in $\mathcal{D}$ and remember that $\Delta g(x) := g(x) - g^*(x)$. Denote $LB^*_{y=1} = (1-\rho_1)(1-\pi_1) + \rho_0\pi_1, UB^*_{y=0} = (1-\rho_1)\pi_0 + \rho_0(1-\pi_0)$. We have

Lemma 3 *When g is unassuming, we have*

$$\begin{cases} LB_{y=1} = LB^*_{y=1} + \langle \Delta g(x) \rangle_{s1} - \frac{(1-\rho_1-\rho_0)^2}{p_{s1}} \Delta p_o \\ UB_{y=0} = UB^*_{y=0} + \langle \Delta g(x) \rangle_{s0} + \frac{(1-\rho_1-\rho_0)^2}{1-p_{s1}} \Delta p_o \\ \hat{\rho}_1^{conf} = \rho_1 + \frac{1-\rho_1-\rho_0}{|P|-|\Delta P_1|+|\Delta N_1|} |\Delta N_1| \\ \hat{\rho}_0^{conf} = \rho_0 + \frac{1-\rho_1-\rho_0}{|N|-|\Delta N_0|+|\Delta P_0|} |\Delta P_0| \end{cases} \tag{4.8}$$

where

$$\begin{cases} \Delta P_1 = \{x \in P \mid g(x) < LB_{y=1}\} \\ \Delta N_1 = \{x \in N \mid g(x) \geq LB_{y=1}\} \\ \Delta P_0 = \{x \in P \mid g(x) \leq UB_{y=0}\} \\ \Delta N_0 = \{x \in N \mid g(x) > UB_{y=0}\} \end{cases}$$

The second term on the R.H.S. of the $\hat{\rho}_i^{conf}$ expressions captures the deviation of $\hat{\rho}_i^{conf}$ from ρ_i, $i = 0, 1$. This term results from both imperfect $g(x)$ and overlapping support. Because the term is non-negative, $\hat{\rho}_i^{conf} \geq \rho_i$, $i = 0, 1$ in the limit of infinite examples. In other words, $\hat{\rho}_i^{conf}$ is an *upper bound* for the noise rates ρ_i, $i = 0, 1$. From Lemma 3, it also follows:

Theorem 4 *Given non-overlapping support condition,*

If $\forall x \in N, \Delta g(x) < LB_{y=1} - \rho_0$, then $\hat{\rho}_1^{conf} = \rho_1$.

If $\forall x \in P, \Delta g(x) > -(1 - \rho_1 - UB_{y=0})$, then $\hat{\rho}_0^{conf} = \rho_0$.

Theorem 4 shows that $\hat{\rho}_1^{conf}$ and $\hat{\rho}_0^{conf}$ are robust to imperfect probability estimation. As long as $\Delta g(x)$ does not exceed the distance between the threshold in Eq. (4.3) and the perfect $g^*(x)$ value, $\hat{\rho}_1^{conf}$ and $\hat{\rho}_0^{conf}$ are consistent estimators for ρ_1 and ρ_0. Our numerical experiments in Section 4.4 suggest this is reasonable for

$\Delta g(x)$. The average $|\Delta g(x)|$ for the MNIST training dataset across different (ρ_1, π_1) varies between 0.01 and 0.08 for a logistic regression classifier, 0.01∼0.03 for a CNN classifier, and 0.05∼0.10 for the CIFAR dataset with a CNN classifier. Thus, when $LB_{y=1} - \rho_0$ and $1 - \rho_1 - UB_{y=0}$ are above 0.1 for these datasets, from Theorem 4 we see that $\hat{\rho}_i^{conf}$ still accurately estimates ρ_i.

4.3.3 The Rank Pruning Algorithm

After estimating $\hat{\rho}_1$ and $\hat{\rho}_0$, Rank Pruning must uncover the classifications of f from g. In this section, we describe how Rank Pruning identifies confident examples, removes the rest, and trains on the pruned set using a reweighted loss function.

First, we obtain the inverse noise rates $\hat{\pi}_1$ and $\hat{\pi}_0$ from $\hat{\rho}_1$ and $\hat{\rho}_0$:

$$\hat{\pi}_1 = \frac{\hat{\rho}_0}{p_{s1}} \frac{1 - p_{s1} - \hat{\rho}_1}{1 - \hat{\rho}_1 - \hat{\rho}_0}, \hat{\pi}_0 = \frac{\hat{\rho}_1}{1 - p_{s1}} \frac{p_{s1} - \hat{\rho}_0}{1 - \hat{\rho}_1 - \hat{\rho}_0} \tag{4.9}$$

Next, we prune the $\hat{\pi}_1|\tilde{P}|$ examples in $\tilde{P}$ with smallest $g(x)$ and the $\hat{\pi}_0|\tilde{N}|$ examples in $\tilde{N}$ with highest $g(x)$ and denote the pruned sets $\tilde{P}_{conf}$ and $\tilde{N}_{conf}$. To prune, we define k_1 as the $(\hat{\pi}_1|\tilde{P}|)^{th}$ smallest $g(x)$ for $x \in \tilde{P}$ and k_0 as the $(\hat{\pi}_0|\tilde{N}|)^{th}$ largest $g(x)$ for $x \in \tilde{N}$. BFPRT ($\mathcal{O}(n)$) (Blum et al., 1973) is used to compute k_1 and k_0 and pruning is reduced to the following $\mathcal{O}(n)$ filter:

$$\tilde{P}_{conf} := \{x \in \tilde{P} \mid g(x) \geq k_1\}, \quad \tilde{N}_{conf} := \{x \in \tilde{N} \mid g(x) \leq k_0\} \tag{4.10}$$

Lastly, we refit the classifier to $X_{conf} = \tilde{P}_{conf} \cup \tilde{N}_{conf}$ by class-conditionally reweighting the loss function for examples in $\tilde{P}_{conf}$ with weight $\frac{1}{1-\hat{\rho}_1}$ and examples in $\tilde{N}_{conf}$ with weight $\frac{1}{1-\hat{\rho}_0}$ to recover the estimated balance of positive and negative examples. The full Rank Pruning algorithm is presented in Alg. 1 and illustrated step-by-step on a synthetic dataset in Fig. 4-1.

We conclude this section with a formal discussion of the loss function and efficiency of Rank Pruning. Define $\hat{y}_i$ as the predicted label of example i for the classifier fit to X_{conf}, s_{conf} and let $l(\hat{y}_i, s_i)$ be the original loss function for $x_i \in D_\rho$. Then the loss function for Rank Pruning is simply the original loss function exerted on the pruned

Algorithm 1 Rank Pruning

```
Input: Examples $X$, corrupted labels $s$, classifier clf
Part 1. Estimating Noise Rates:
(1.1) clf.fit($X$,$s$)
        $g(x) \leftarrow$clf.predict_crossval_probability($\hat{s} = 1|x$)
        $p_{s1} = \frac{\text{count}(s=1)}{\text{count}(s=0 \vee s=1)}$
        $LB_{y=1} = E_{x \in \tilde{P}}[g(x)]$, $UB_{y=0} = E_{x \in \tilde{N}}[g(x)]$
(1.2) $\hat{\rho}_1 = \hat{\rho}_1^{conf} = \frac{|\tilde{N}_{y=1}|}{|\tilde{N}_{y=1}|+|\tilde{P}_{y=1}|}$, $\hat{\rho}_0 = \hat{\rho}_0^{conf} = \frac{|\tilde{P}_{y=0}|}{|\tilde{P}_{y=0}|+|\tilde{N}_{y=0}|}$
        $\hat{\pi}_1 = \frac{\hat{\rho}_0}{p_{s1}} \frac{1-p_{s1}-\hat{\rho}_1}{1-\hat{\rho}_1-\hat{\rho}_0}$, $\hat{\pi}_0 = \frac{\hat{\rho}_1}{1-p_{s1}} \frac{p_{s1}-\hat{\rho}_0}{1-\hat{\rho}_1-\hat{\rho}_0}$
Part 2. Prune Inconsistent Examples:
(2.1) Remove $\hat{\pi}_1|\tilde{P}|$ examples in $\tilde{P}$ with least $g(x)$, Remove $\hat{\pi}_0|\tilde{N}|$ examples in $\tilde{N}$ with greatest $g(x)$,
        Denote the remaining training set ($X_{conf}$, $s_{conf}$)
(2.2) clf.fit($X_{conf}$, $s_{conf}$), with sample weight $w(x) = \frac{1}{1-\hat{\rho}_1}\mathbb{1}[[s_{conf} = 1]]+\frac{1}{1-\hat{\rho}_0}\mathbb{1}[[s_{conf} = 0]]$
Output: clf
```

X_{conf}, with class-conditional weighting:

$$\tilde{l}(\hat{y}_i, s_i) = \frac{1}{1-\hat{\rho}_1} l(\hat{y}_i, s_i) \cdot \mathbb{1}[[x_i \in \tilde{P}_{conf}]] + \frac{1}{1-\hat{\rho}_0} l(\hat{y}_i, s_i) \cdot \mathbb{1}[[x_i \in \tilde{N}_{conf}]] \tag{4.11}$$

Effectively this loss function uses a zero-weight for pruned examples. Other than potentially fewer examples, the only difference in the loss function for Rank Pruning and the original loss function is the class-conditional weights. These constant factors do not increase the complexity of the minimization of the original loss function. In other words, we can fairly report the running time of Rank Pruning in terms of the running time ($\mathcal{O}(T)$) of the choice of probabilistic estimator. Combining noise estimation ($\mathcal{O}(T)$), pruning ($\mathcal{O}(n)$), and the final fitting ($\mathcal{O}(T)$), Rank Pruning has a running time of $\mathcal{O}(T) + \mathcal{O}(n)$, which is $\mathcal{O}(T)$ for typical classifiers.

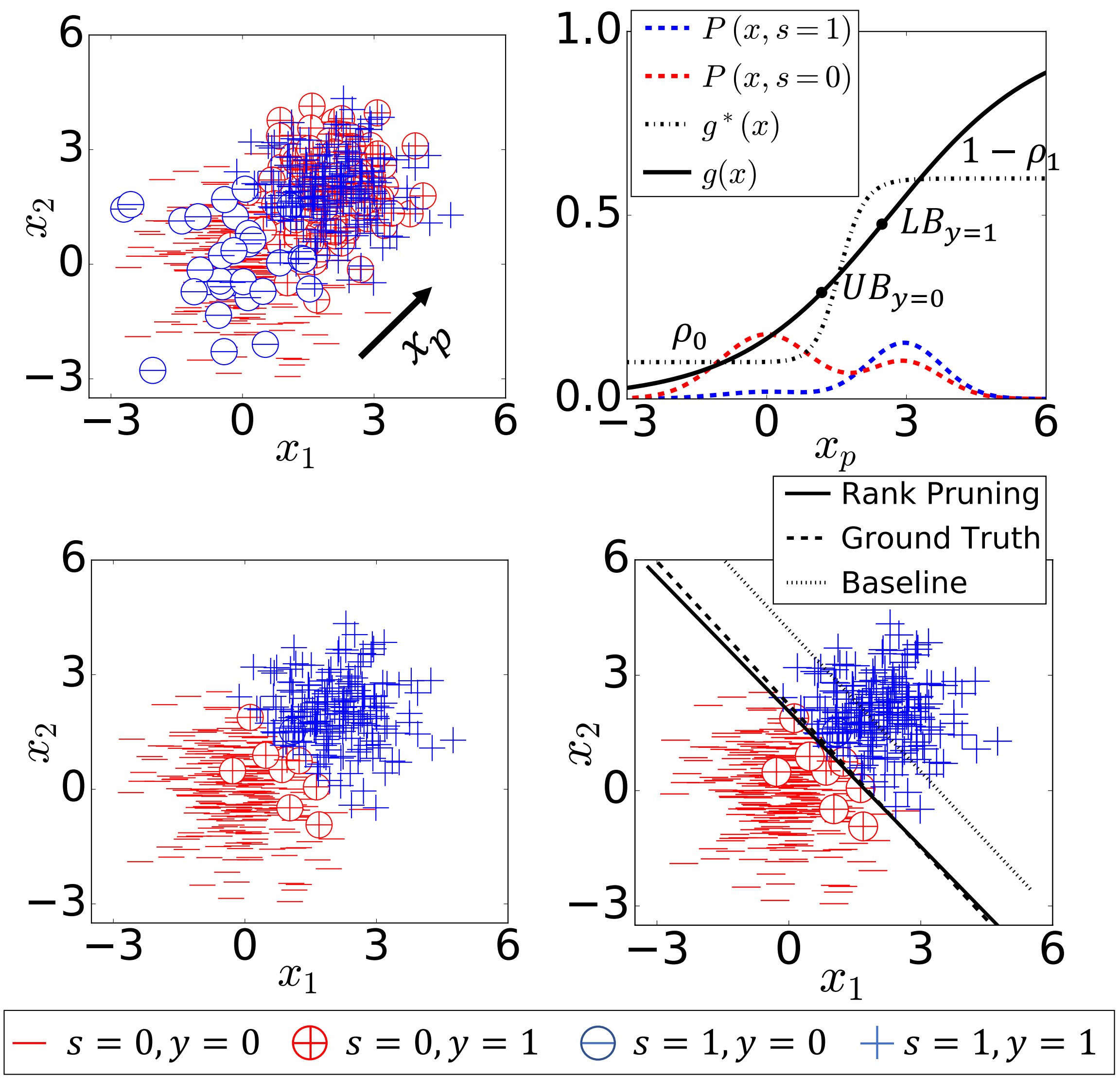


Figure 4-1: Illustration of Rank Pruning with a logistic regression classifier ($\mathcal{LR}_\theta$). (a): The corrupted training set D_ρ with noise rates $\rho_1 = 0.4$ and $\rho_0 = 0.1$. Rank Pruning is given corrupted colored labels red ($s = 0$) and blue ($s = 1$). True y ($+$ or $-$) is hidden. (b): The marginal distribution of D_ρ projected onto the x_p axis (indicated in (a)), and the $\mathcal{LR}_\theta$'s estimated $g(x)$, from which $\hat{\rho}_1^{conf} = 0.4237$, $\hat{\rho}_0^{conf} = 0.1144$ are estimated. (c): The pruned X_{conf}, s_{conf}. (d): The classification result by Rank Pruning ($\hat{f} = \mathcal{LR}_\theta.\text{fit}(X_{conf}, s_{conf})$), ground truth classifier ($f = \mathcal{LR}_\theta.\text{fit}(X, y)$), and baseline classifier ($g = \mathcal{LR}_\theta.\text{fit}(X, s)$), with an accuracy of 94.16%, 94.16% and 78.83%, respectively.

4.3.4 Rank Pruning: A simple summary

Recognizing that formalization can create obfuscation, in this section we describe the entire algorithm in a few sentences. Rank Pruning takes as input training examples X, noisy labels s, and a probabilistic classifier clf and finds a subset of X, s that is likely to be correctly labeled, i.e. a subset of X, y. To do this, we first find two thresholds, $LB_{y=1}$ and $UB_{y=0}$, to *confidently* guess the correctly and incorrectly labeled examples in each of $\tilde{P}$ and $\tilde{N}$, forming four sets, then use the set sizes to estimate the noise rates $\rho_1 = P(s = 0|y = 1)$ and $\rho_0 = P(s = 1|y = 0)$. We then use the noise rates to estimate the number of examples with observed label $s = 1$ and hidden label $y = 0$ and remove that number of examples from $\tilde{P}$ by removing those with lowest predicted probability $g(x)$. We prune $\tilde{N}$ similarly. Finally, the classifier is fit to the pruned set, which is intended to represent a subset of the correctly labeled data.

4.3.5 Expected Risk Evaluation

In this section, we prove Rank Pruning exactly uncovers the classifier f fit to hidden y labels when g range separates P and N and exact estimates of ρ_1 and ρ_0 are given.

Denote $f_\theta \in \mathcal{F} : x \to \hat{y}$ as a classifier's prediction function belonging to some function space $\mathcal{F}$, where θ represents the classifier's parameters. f_θ represents f, but without θ necessarily fit to the training data. $\hat{f}$ is the Rank Pruning estimate of f.

Denote the empirical risk of f_θ w.r.t. the loss function l and distribution $\mathcal{D}$ as $\hat{R}_{l,\mathcal{D}}(f_\theta) = \frac{1}{n}\sum_{i=1}^{n} \tilde{l}(f_\theta(x_i), s_i)$, and the expected risk $R_{l,\mathcal{D}}(f_\theta) = E_{(x,y)\sim\mathcal{D}}[\hat{R}_{l,\mathcal{D}}(f_\theta)]$. We show that using Rank Pruning, a classifier $\hat{f}$ can be learned for the hidden data (X, y), given the corrupted data (X, s), by minimizing the empirical risk $\hat{R}_{\tilde{l},D_\rho}(f_\theta)$:

$$\hat{f} = \underset{f_\theta \in \mathcal{F}}{\operatorname{argmin}} \hat{R}_{\tilde{l},D_\rho}(f_\theta) = \underset{f_\theta \in \mathcal{F}}{\operatorname{argmin}} \frac{1}{n}\sum_{i=1}^{n} \tilde{l}(f_\theta(x_i), s_i) \tag{4.12}$$

Under the *range separability* condition, we have

Theorem 5 *If g range separates P and N and $\hat{\rho}_i = \rho_i$, $i = 0, 1$, then for any classifier f_θ and any bounded loss function $l(\hat{y}_i, y_i)$, we have*

$$R_{\tilde{l},\mathcal{D}_\rho}(f_\theta) = R_{l,\mathcal{D}}(f_\theta) \tag{4.13}$$

where $\tilde{l}(\hat{y}_i, s_i)$ *is Rank Pruning's loss function (Eq. 4.11).*

The proof of Theorem 5 is in Appendix A.1.5. Intuitively, Theorem 5 tells us that if g range separates P and N, then given exact noise rate estimates, Rank Pruning will exactly prune out the positive examples in $\tilde{N}$ and negative examples in $\tilde{P}$, leading to the same expected risk as learning from uncorrupted labels. Thus, Rank Pruning can exactly uncover the classifications of f (with infinite examples), since the expected risk is equivalent for any f_θ. Note Theorem 5 also holds when g is ideal, since *ideal* $\subset$ *range separability.*

In practice, *range separability* encompasses a wide range of imperfect $g(x)$ scenarios, e.g. $g(x)$ can have large fluctuation in both P and N, or have systematic drift w.r.t. to $g^*(x)$. When g does not range separate P and N, Rank Pruning is still invariant to imperfect $g(x)$ within a range separable subset of P and N. Because Rank Pruning uses confident examples for robustness, when noise rate estimates are inexact, Rank Pruning maintains comparatively superior performance as shown in our experiments in the next section.

4.4 Experimental Results

In Section 4.3, we developed a theoretical framework for Rank Pruning, proved exact noise estimation and equivalent expected risk when conditions are ideal, and derived closed-form solutions when conditions are non-ideal. Our theory suggests that, in practice, Rank Pruning should (1) accurately estimate ρ_1 and ρ_0, (2) typically achieve as good or better F1, error and AUC-PR (Davis and Goadrich, 2006) when compared with prior state-of-the-art, regardless of classifier used or input distribution, and (3) be robust to both mislabeling and added random noise.

In this section, we support these claims with an evaluation of the comparative performance of Rank Pruning in non-ideal conditions across thousands of scenarios including different datasets, classifier, values of ρ_1, values of π_1, added random noise, separability of P and N, input dimension, and number of training examples. On the contrary, if we only considered one scenario, Rank Pruning could be "tuned" to perform well. Instead, we consider less complex (MNIST) and more complex (CIFAR) datasets, simple (logistic regression) and complex (CNN) classifiers, the range of noise rates, among other practical variations to ensure that Rank Pruning is a general and agnostic solution for $\tilde{P}\tilde{N}$ learning.

In our experiments, we adjust π_1 instead of ρ_0 because binary noisy classification problems (e.g. detection and recognition tasks) often have that $|P| \ll |N|$. This choice allows us to adjust both noise rates with respect to P, i.e. the fraction of true positive examples that are mislabeled as negative (ρ_1) and the fraction of observed positive labels that are actually mislabeled negative examples (π_1). All $\tilde{P}\tilde{N}$ algorithms are trained with corrupted labels s, and tested on an unseen test set by comparing predictions $\hat{y}$ with the true test labels y using F1 score, error, and AUC-PR metrics. We include all three to emphasize our apathy toward tuning results to any single metric. We provide F1 scores in this section with error and AUC-PR scores in Appendix A.3.

4.4.1 Synthetic Dataset

The synthetic dataset is comprised of a Guassian positive class and a Guassian negative classes such that negative examples ($y = 0$) obey an m-dimensional Gaussian distribution $N(\mathbf{0}, \mathbf{I})$ with unit variance $\mathbf{I} = diag(1, 1, ...1)$, and positive examples obey $N(d\mathbf{1}, 0.8\mathbf{I})$, where $d\mathbf{1} = (d, d, ...d)$ is an m-dimensional vector, and d measures the separability of the positive and negative set.

We test Rank Pruning by varying 4 different settings of the environment: separability d, dimension, number of training examples n, and percent (of n) added random noise drawn from a uniform distribution $U([-10, 10]^m)$. In each scenario, we test 5 different (π_1, ρ_1) pairs: $(\pi_1, \rho_1) \in \{(0, 0), (0, 0.5), (0.25, 0.25), (0.5, 0), (0.5, 0.5)\}$. From Fig. 4-3, we observe that across these settings, the F1 score for Rank Pruning is fairly agnostic to magnitude of mislabeling (noise rates). As a validation step, in Fig.

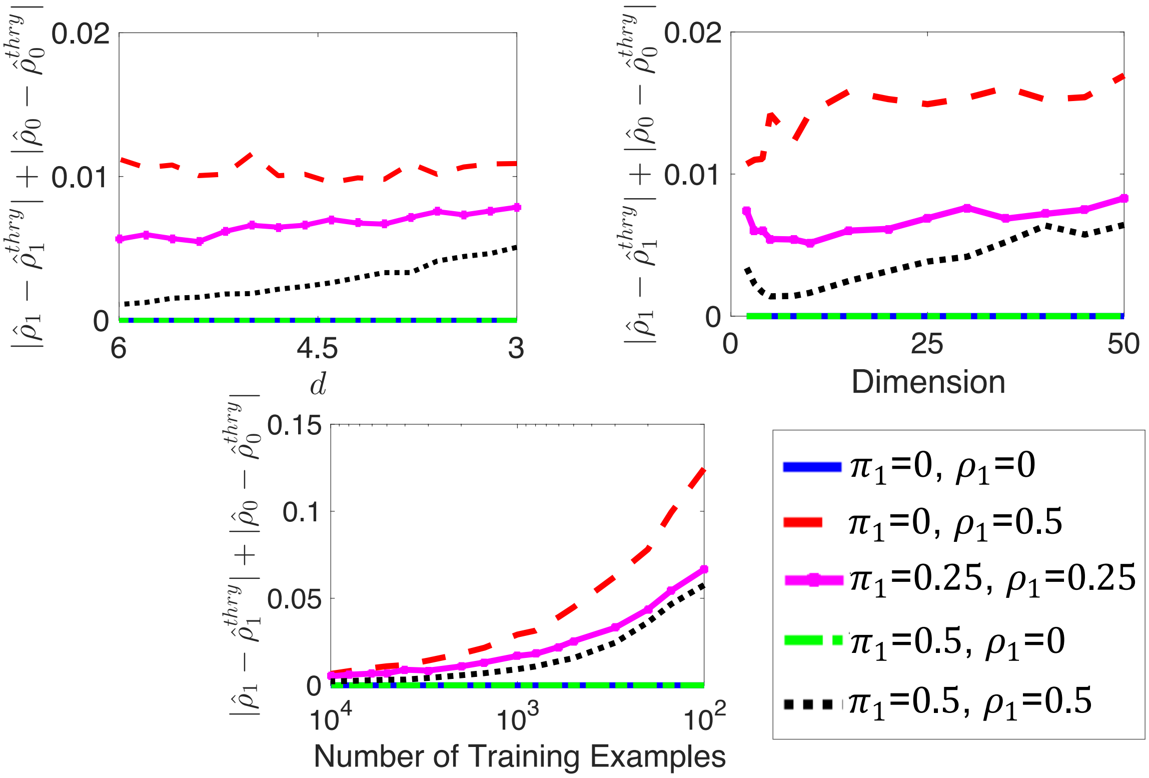


Figure 4-2: Sum of absolute difference between theoretically estimated $\hat{\rho}_i^{thry}$ and empirical $\hat{\rho}_i$, $i = 0, 1$, with five different (π_1, ρ_1), for varying separability d, dimension, and number of training examples. Note that no figure exists for percent random noise because the theoretical estimates in Eq. (4.8) do not address added noise examples.

4-2 we measure how closely our empirical estimates match our theoretical solutions in Eq. (4.8) and find near equivalence except when the number of training examples approaches zero.

For significant mislabeling ($\rho_1 = 0.5$, $\pi_1 = 0.5$), Rank Pruning often outperforms other $\tilde{P}\tilde{N}$ learning methods (Fig. 4-4). In the scenario of different separability d, it achieves nearly the same F1 score as the ground truth classifier. Remarkably, from Fig. 4-3 and Fig. 4-4, we observe that when added random noise comprises 50% of total training examples, Rank Pruning still achieves F1 > 0.85, compared with F1 < 0.5 for all other methods. This emphasizes a unique feature of Rank Pruning, it will also remove added random noise because noise drawn from a third distribution is unlikely to appear confidently positive or negative.

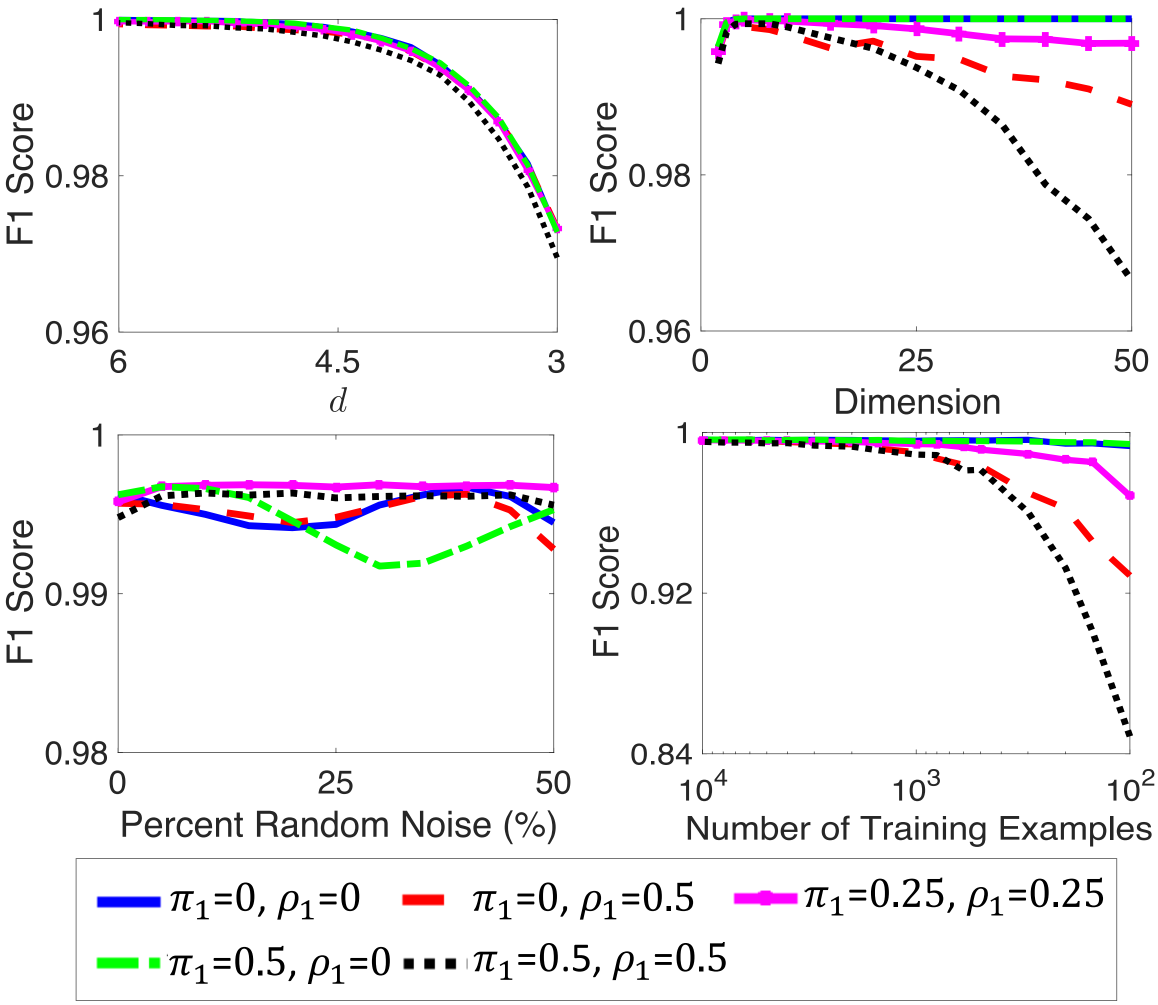


Figure 4-3: Comparison of Rank Pruning with different noise ratios (π_1, ρ_1) on a synthetic dataset for varying separability d, dimension, added random noise and number of training examples. Default settings for Fig. 4-3, 4-2 and 4-4: $d = 4$, 2-dimension, 0% random noise, and 5000 training examples with $p_{y1} = 0.2$. The lines are an average of 200 trials.

4.4.2 MNIST and CIFAR Datasets

We consider the binary classification task of one-vs-rest for the MNIST (LeCun and Cortes, 2010) and CIFAR-10 (Krizhevsky et al. (2017)) datasets. For example, in MNIST, the task is to predict if a digit is a “1" or “not", for all digits, and similarly for CIFAR-10 images. As in the synthetic experiments, ρ_1, π_1 is given to all methods for fair comparison, except for RP_ρ which is the entire Rank Pruning algorithm including noise rate estimation. ρ_1 and π_1 are kept hidden from RP_ρ so that RP_ρ metric scores measure our performance on the unadulterated $\tilde{P}\tilde{N}$ learning problem.

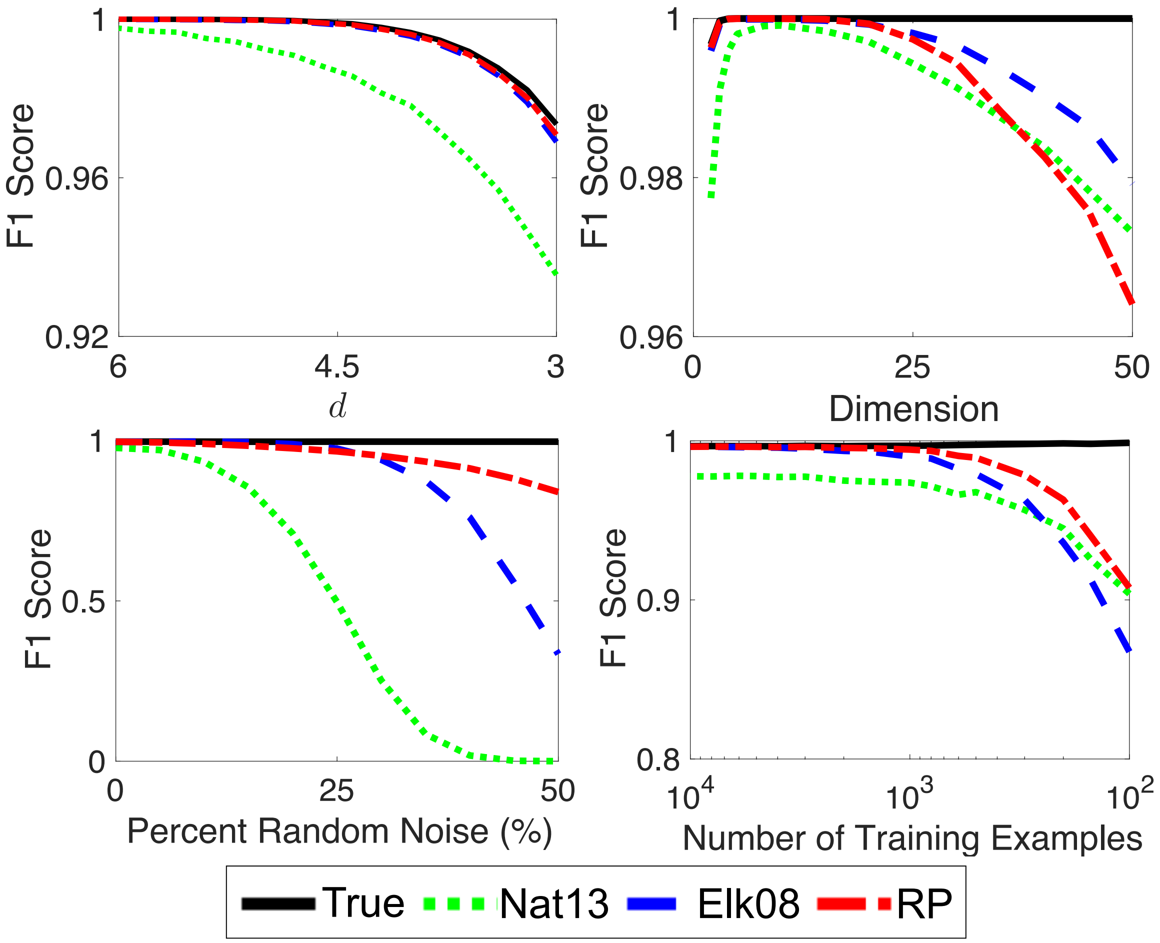


Figure 4-4: Comparison of $\tilde{P}\tilde{N}$ methods for varying separability d, dimension, added random noise, and number of training examples for $\pi_1 = 0.5$, $\rho_1 = 0.5$ (given to all methods).

As evidence that Rank Pruning is both dataset-agnostic and classifier-agnostic, we demonstrate its superiority over three prior state-of-the-art methods on the MNIST and CIFAR datasets for both (1) a linear logistic regression model with unit L2 regularization and (2) an AlexNet CNN variant with max pooling and dropout, modified to have a two-class output. The CNN structure can be found in Chollet (2016b) for MNIST and Chollet (2016a) for CIFAR. A 10% holdout set monitors the CNN loss and ends training when there is no decrease for 10 epochs (max 50 for MNIST and 150 for CIFAR).

We consider noise rates $\pi_1, \rho_1 \in \{(0, 0.5), (0.25, 0.25),\ (0.5, 0), (0.5, 0.5)\}$ for both MNIST and CIFAR, with additional settings for MNIST in Table 4.3 to empha-

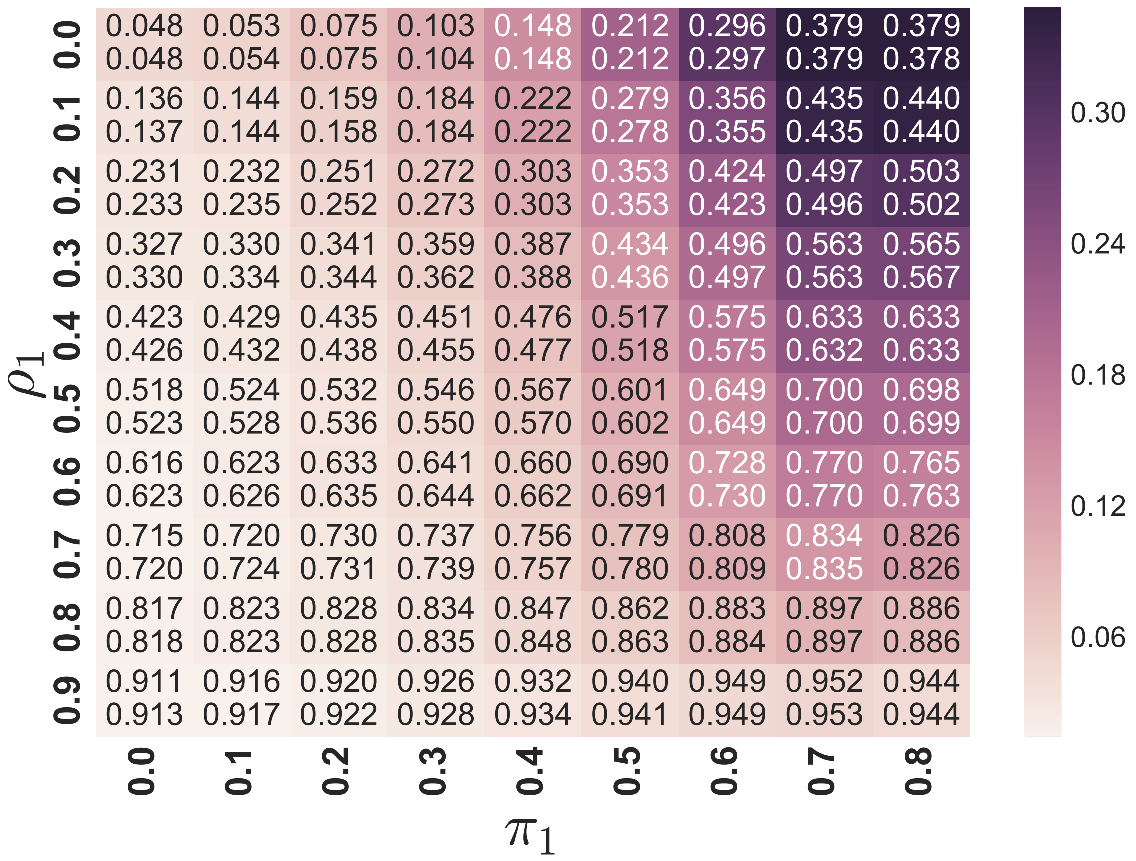
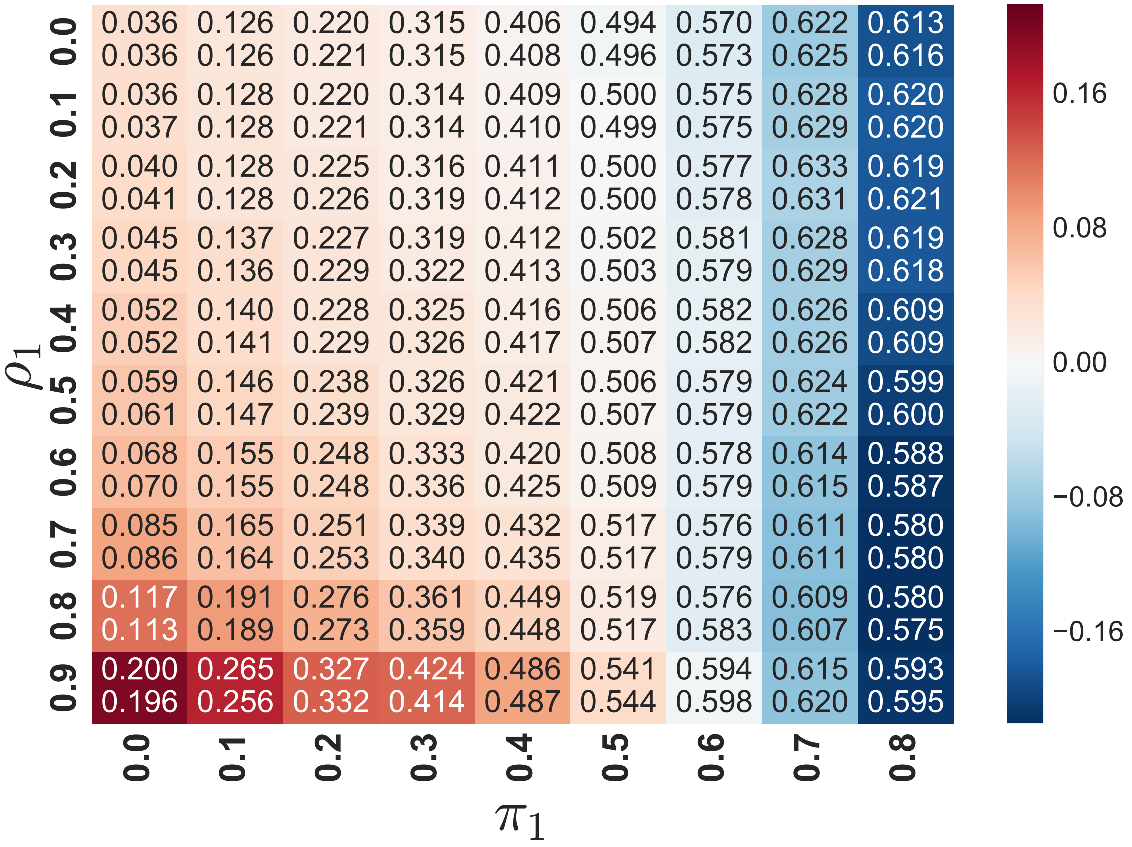


Figure 4-5: Rank Pruning $\hat{\rho}_1$ estimation consistency, averaged over all digits in MNIST. Color depicts $\hat{\rho}_1 - \rho_1$ with $\hat{\rho}_1$ (upper) and theoretical $\hat{\rho}_1^{thry}$ (lower) in each block.

size Rank Pruning performance is noise rate agnostic. The $\rho_1 = 0$, $\pi_1 = 0$ case is omitted because when given ρ_1, π_1, all methods have the same loss function as the ground truth classifier, resulting in nearly identical F1 scores.

For MNIST using logistic regression, we evaluate the consistency of our noise rate estimators with actual noise rates and the theoretical estimates (Eq. 4.8) across $\pi_1 \in [0, 0.8] \times \rho_1 \in [0, 0.9]$. The results for $\hat{\rho}_1$ (Fig. 4-5) and $\hat{\pi}_1$ (Fig. 4-6) are satisfyingly consistent, with mean absolute difference $\text{MD}_{\hat{\rho}_1,\rho_1} = 0.105$ and $\text{MD}_{\hat{\pi}_1,\pi_1} = 0.062$, and validate our theoretical solutions ($\text{MD}_{\hat{\rho}_1,\hat{\rho}_1^{thry}} = 0.0028$, $\text{MD}_{\hat{\pi}_1,\hat{\pi}_1^{thry}} = 0.0058$).

There are two important observations from our analysis of Rank Pruning on CIFAR and MNIST. First, Rank Pruning performs well in nearly every scenario and boasts the most dramatic improvement over prior state-of-the-art in the presence of extreme noise ($\pi_1 = 0.5$, $\rho_1 = 0.5$). This is easily observed in the right-most quadrant



ρ_1 \ π_1	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8
0.0	0.036	0.126	0.220	0.315	0.406	0.494	0.570	0.622	0.613
	0.036	0.126	0.221	0.315	0.408	0.496	0.573	0.625	0.616
0.1	0.036	0.128	0.220	0.314	0.409	0.500	0.575	0.628	0.620
	0.037	0.128	0.221	0.314	0.410	0.499	0.575	0.629	0.620
0.2	0.040	0.128	0.225	0.316	0.411	0.500	0.577	0.633	0.619
	0.041	0.128	0.226	0.319	0.412	0.500	0.578	0.631	0.621
0.3	0.045	0.137	0.227	0.319	0.412	0.502	0.581	0.628	0.619
	0.045	0.136	0.229	0.322	0.413	0.503	0.579	0.629	0.618
0.4	0.052	0.140	0.228	0.325	0.416	0.506	0.582	0.626	0.609
	0.052	0.141	0.229	0.326	0.417	0.507	0.582	0.626	0.609
0.5	0.059	0.146	0.238	0.326	0.421	0.506	0.579	0.624	0.599
	0.061	0.147	0.239	0.329	0.422	0.507	0.579	0.622	0.600
0.6	0.068	0.155	0.248	0.333	0.420	0.508	0.578	0.614	0.588
	0.070	0.155	0.248	0.336	0.425	0.509	0.579	0.615	0.587
0.7	0.085	0.165	0.251	0.339	0.432	0.517	0.576	0.611	0.580
	0.086	0.164	0.253	0.340	0.435	0.517	0.579	0.611	0.580
0.8	0.117	0.191	0.276	0.361	0.449	0.519	0.576	0.609	0.580
	0.113	0.189	0.273	0.359	0.448	0.517	0.583	0.607	0.575
0.9	0.200	0.265	0.327	0.424	0.486	0.541	0.594	0.615	0.593
	0.196	0.256	0.332	0.414	0.487	0.544	0.598	0.620	0.595

Figure 4-6: Rank Pruning $\hat{\pi}_1$ estimation consistency, averaged over all digits in MNIST. Color depicts $\hat{\pi}_1 - \pi_1$ with $\hat{\pi}_1$ (upper) and $\hat{\pi}_1^{thry}$ (lower) in each block.

of Table 4.4. The $\pi_1 = 0.5$, $\rho_1 = 0$ quadrant is nearest to $\pi_1 = 0$, $\rho_1 = 0$ mostly captures CNN prediction variation because $|\tilde{P}| \ll |\tilde{N}|$.

Second, RP$_\rho$ often achieves equivalent (MNIST in Table 4.4) or significantly higher (CIFAR in Tables 4.3 and 4.4) F1 score than Rank Pruning when ρ_1 and π_1 are provided, particularly when noise rates are large. This effect is exacerbated for harder problems (lower F1 score for the ground truth classifier) like the "cat" in CIFAR or the "9" digit in MNIST. The reason is these problems are more complex, resulting in less confident predictions, and therefore more pruning.

Remember that Rank Pruning noise estimation is an upper bound when g is unassuming. Noise rate overestimation accounts for the complexity of harder problems. As a result, Rank Pruning removes correctly labeled examples that "confuse" the classifier, instead fitting only the confident examples in each class. We observe this on

Table 4.3: Comparison of F1 score for one-vs-rest MNIST and CIFAR-10 (averaged over all digits/images) using logistic regression. Except for RP_ρ, ρ_1, ρ_0 are given to all methods. Top model scores are in bold with RP_ρ in red if greater than non-RP models. Due to sensitivity to imperfect $g(x)$, ***Liu16*** often predicts the same label for all examples.

Dataset	CIFAR				MNIST														
$\pi_1 =$	**0.0**	**0.25**	**0.5**	**0.5**	$\pi_1 =$ **0.0**			$\pi_1 =$ **0.25**				$\pi_1 =$ **0.5**				$\pi_1 =$ **0.75**			
Model,$\rho_1 =$	**0.5**	**0.25**	**0.0**	**0.5**	**0.25**	**0.5**	**0.75**	**0.0**	**0.25**	**0.5**	**0.75**	**0.0**	**0.25**	**0.5**	**0.75**	**0.0**	**0.25**	**0.5**	**0.75**
True	0.248	0.248	0.248	0.248	0.894	0.894	0.894	0.894	0.894	0.894	0.894	0.894	0.894	0.894	0.894	0.894	0.894	0.894	0.894
RP$_\rho$	**0.301**	**0.316**	**0.308**	**0.261**	**0.883**	**0.874**	**0.843**	**0.881**	**0.876**	**0.863**	**0.799**	0.823	0.831	**0.819**	**0.762**	0.583	0.603	0.587	0.532
RP	**0.256**	**0.262**	**0.244**	0.209	**0.885**	**0.873**	**0.839**	**0.890**	**0.879**	**0.863**	**0.812**	**0.879**	**0.862**	**0.838**	**0.770**	**0.855**	**0.814**	**0.766**	0.617
Nat13	0.226	0.219	0.194	0.195	0.860	0.830	0.774	0.865	0.836	0.802	0.748	0.839	0.810	0.777	0.721	0.809	0.776	0.736	**0.640**
Elk08	0.221	0.226	0.228	**0.210**	0.862	0.830	0.771	0.864	0.847	0.819	0.762	0.843	0.835	0.814	0.736	0.674	0.669	0.599	0.473
Liu16	0.182	0.182	0.000	0.182	0.021	0.000	0.000	0.000	0.147	0.147	0.073	0.000	0.164	0.163	0.163	0.047	0.158	0.145	0.164

Table 4.4: F1 score comparison on MNIST and CIFAR-10 using a CNN. Except for RP_ρ, ρ_1, ρ_0 are given to all methods.

MNIST/CIFAR		$\pi_1 =$ **0.0**					$\pi_1 =$ **0.25**					$\pi_1 =$ **0.5**									
IMAGE		$\rho_1 =$ **0.5**					$\rho_1 =$ **0.25**					$\rho_1 =$ **0.0**					$\rho_1 =$ **0.5**				
CLASS	True	RP$_\rho$	RP	Nat13	Elk08	Liu16	RP$_\rho$	RP	Nat13	Elk08	Liu16	RP$_\rho$	RP	Nat13	Elk08	Liu16	RP$_\rho$	RP	Nat13	Elk08	Liu16
0	0.993	**0.991**	**0.988**	0.977	0.976	0.179	**0.991**	**0.992**	0.982	0.981	0.179	**0.991**	**0.992**	0.984	0.987	0.985	**0.989**	**0.989**	0.937	0.964	0.179
1	0.993	**0.990**	**0.991**	0.989	0.985	0.204	**0.992**	**0.992**	0.984	0.987	0.204	0.990	0.991	0.992	**0.993**	0.990	**0.989**	**0.989**	0.984	0.988	0.204
2	0.987	**0.973**	**0.976**	0.972	0.969	0.187	**0.984**	**0.983**	0.978	0.975	0.187	0.985	0.986	0.985	0.986	**0.988**	**0.971**	**0.975**	0.968	0.959	0.187
3	0.990	**0.984**	**0.984**	0.972	0.981	0.183	**0.986**	**0.986**	0.978	0.978	0.183	**0.990**	0.987	**0.989**	**0.989**	0.984	**0.981**	**0.979**	0.957	0.971	0.183
4	0.994	**0.981**	0.979	**0.981**	0.977	0.179	**0.985**	**0.987**	0.971	0.964	0.179	0.987	**0.990**	**0.990**	0.989	0.985	**0.977**	**0.982**	0.955	0.961	0.179
5	0.989	**0.982**	**0.980**	0.978	0.979	0.164	**0.985**	**0.982**	0.964	0.965	0.164	**0.988**	**0.987**	**0.987**	0.984	**0.987**	**0.965**	**0.968**	0.962	0.957	0.164
6	0.989	**0.986**	**0.985**	0.972	0.982	0.175	**0.985**	**0.987**	0.978	0.981	0.175	0.985	0.985	**0.988**	0.987	0.985	**0.983**	**0.982**	0.946	0.959	0.175
7	0.987	**0.981**	**0.980**	0.967	0.948	0.186	**0.976**	**0.975**	0.971	0.971	0.186	0.976	0.980	**0.985**	0.982	0.983	**0.973**	**0.968**	0.942	0.958	0.186
8	0.989	**0.975**	**0.978**	0.943	0.967	0.178	**0.982**	**0.981**	0.967	0.951	0.178	0.982	**0.984**	0.982	0.979	0.983	**0.977**	**0.975**	0.864	0.959	0.178
9	0.982	0.966	**0.974**	0.972	0.935	0.183	**0.976**	**0.974**	0.967	0.967	0.183	0.976	0.975	0.974	**0.978**	0.970	**0.959**	0.940	0.931	**0.942**	0.183
AVG$_{MN}$	0.989	**0.981**	**0.981**	0.972	0.970	0.182	**0.984**	**0.984**	0.974	0.972	0.182	0.985	**0.986**	**0.986**	0.985	0.984	**0.976**	**0.975**	0.945	0.962	0.182
PLANE	0.755	**0.689**	**0.634**	0.619	0.585	0.182	**0.695**	**0.702**	0.671	0.640	0.182	**0.757**	**0.746**	0.716	0.735	0.000	**0.628**	**0.635**	0.459	0.598	0.182
AUTO	0.891	**0.791**	**0.785**	0.761	0.768	0.000	**0.832**	**0.824**	0.771	0.783	0.182	0.862	0.866	**0.869**	0.865	0.000	**0.749**	**0.720**	0.582	0.501	0.182
BIRD	0.669	**0.504**	**0.483**	0.445	0.389	0.182	**0.543**	**0.515**	0.469	0.426	0.182	**0.577**	**0.619**	0.543	0.551	0.000	**0.447**	**0.409**	0.366	0.387	0.182
CAT	0.487	**0.350**	0.279	0.310	**0.313**	0.000	**0.426**	0.317	**0.350**	0.345	0.182	**0.489**	**0.433**	0.426	0.347	0.000	**0.394**	0.282	0.240	**0.313**	0.182
DEER	0.726	**0.593**	**0.540**	0.455	0.522	0.182	**0.585**	0.554	0.480	**0.569**	0.182	0.614	0.630	**0.643**	0.633	0.000	**0.458**	0.375	0.310	**0.383**	0.182
DOG	0.569	**0.544**	**0.577**	0.429	0.456	0.000	**0.579**	0.559	0.569	**0.576**	0.182	0.647	0.637	**0.667**	0.630	0.000	**0.516**	0.461	0.412	**0.465**	0.182
FROG	0.815	**0.746**	0.727	**0.733**	0.718	0.000	**0.729**	**0.750**	0.630	0.584	0.182	0.767	**0.782**	0.777	0.770	0.000	**0.635**	**0.615**	0.589	0.524	0.182
HORSE	0.805	**0.690**	0.670	0.624	**0.672**	0.182	**0.710**	0.669	**0.683**	0.627	0.182	0.761	**0.776**	0.769	0.753	0.000	**0.672**	**0.569**	0.551	0.461	0.182
SHIP	0.851	**0.791**	**0.783**	0.719	0.758	0.182	**0.810**	**0.801**	0.758	0.723	0.182	0.816	0.822	0.830	**0.831**	0.000	**0.715**	**0.738**	0.569	0.632	0.182
TRUCK	0.861	**0.744**	**0.722**	0.655	0.665	0.182	**0.814**	**0.826**	0.798	0.774	0.182	0.812	0.830	**0.826**	0.824	0.000	**0.654**	0.543	0.575	**0.584**	0.182
AVG$_{CF}$	0.743	**0.644**	**0.620**	0.575	0.585	0.109	**0.672**	**0.652**	0.618	0.605	0.182	**0.710**	**0.714**	0.707	0.694	0.000	**0.587**	**0.535**	0.465	0.485	0.182

CIFAR in Table 4.3 where logistic regression severely underfits the CIFAR dataset so that RP$_\rho$ has significantly higher recall and slightly lower precision than other models, resulting in higher error, but significantly higher F1 score than the ground truth classifier. In this way, Rank Pruning may outperform (F1 score) the ground truth classifier on hard problems regardless of added noise.

4.5 Discussion

To our knowledge, Rank Pruning is the first time-efficient algorithm for $\tilde{P}\tilde{N}$ learning that achieves similar or better F1, error, and AUC-PR than current state-of-the-art methods across practical scenarios for synthetic, MNIST, and CIFAR datasets, with logistic regression and CNN classifiers, across the range of ρ_1, ρ_0 values, for varying added noise examples, dimension, separability, and number of training examples. By *learning with confident examples*, we discover provably consistent estimators for noise rates ρ_1, ρ_0, derive theoretical solutions when g is unassuming, and exactly uncover the classifications of f fit to hidden labels when g range separates P and N.

An important contribution of Rank Pruning is generality, both in classifier and implementation. We allow any classifier, but use logistic regression and a CNN in our experiments to emphasize that our findings are not dependent on model complexity. We evaluate thousands of scenarios to ensure our findings are not an artifact of problem setup. A key point of Rank Pruning is that we only consider the simplest, non-parametric version. We tried many variants for how to perform pruning, tested across these settings, and achieved significant improvements across all metrics, but to ensure generality, we omit these results and present only the basic model.

At its core, Rank Pruning is a simple, robust, and general solution for noisy binary classification by *learning with confident examples*, but it also challenges how we think about training data. For example, SVM changed the way we think about training examples by showing how a decision boundary can be recovered from only support vectors. Yet, when training data contains significant mislabeling, the confident examples, many of which are far from the boundary, are informative in uncovering the true relationship $P(y = 1|x)$. Although modern affordances of “big data" emphasize the value of *more* examples for training, through Rank Pruning we instead encourage a rethinking of learning with *confident* examples.

Chapter 5

Conclusion

In Chapter 1, you learned that if MOOC platforms are to remain open-access, and thus continue to allow the creation of multiple accounts, the CAMEO strategy challenges their viability as credible academic credentialing services. In Chapter 2, you learned about the CAMEO algorithm and how it was used to estimate a *lower bound* of CAMEO prevalence among 1.9 million course participants in 115 HarvardX and MITx courses. In Chapter 3, you saw the shortcomings of CAMEO and other approaches for multiple-account cheating detection, and in response in Chapter 4 you saw the inner-workings of the Rank Pruning algorithm for binary classification with noisy labels.

In this chapter, you will learn about the choice to separate the development of the CAMEO and Rank Pruning algorithms as well as the generalizabilty and impact of this thesis. I begin with a restatement of the contributions of both algorithms in Section 5.1 followed by a discussion of the choice to present the CAMEO and Rank Pruning algorithms independently and their joint application in Section 5.2. In Section 5.3, I clarify the choice to focus this thesis on cheating detection instead of cheating prevention and in Section 5.4, I illustrate how the techniques used in the CAMEO algorithm can be re-purposed to solve new and different problems. I conclude this chapter in Section 5.5 with a refocusing of the purpose of this thesis and an encouraging message for the future of massive open online courses.

The CAMEO algorithm and Rank Pruning algorithm can be used together to undertake the challenge of openness versus value in Massive Open Online Courses. It is my hope that this final chapter inspires us all to pursue solutions that encourage honest behavior in online academic settings and to remain steadfast in our endeavor to reify the intended promise of accessible education for all aspiring learners.

5.1 Contributions

Now that you have seen the CAMEO and Rank Pruning algorithms, I restate the contributions of this thesis as they relate to each algorithm. In particular for the CAMEO algorithm, with the support of co-authors Andrew Ho (Harvard) and Isaac Chuang (MIT), I:

- Defined the Copying Answers using Multiple Existences Online (CAMEO) cheating strategy and estimated a *lower bound* for its prevalence among 1.9 million course participants in 115 HarvardX and MITx courses.
- Identified preventive strategies that can decrease CAMEO rates and showed evidence of their effectiveness in science courses.
- Established new educational data-mining methodologies for analysis in Massive Open Online Courses.
- Described a novel *honeypot* validation technique that verifies cheating by appending unique digit sequences to *show answer* fields.
- Developed the CAMEO detection algorithm using human-defined filters and thresholds as a method for producing noisy CAMEO cheating labels in MOOCs.

In particular for the Rank Pruning algorithm, along with co-authors Tailin Wu (MIT) and Isaac Chuang (MIT), I:

- Developed a robust, time-efficient, general solution for both $\tilde{P}\tilde{N}$ learning, i.e. binary classification with noisy labels, and estimation of the fraction of mislabeling in both the positive and negative training sets.

- Introduced the *learning with confident examples* principle as a new way to think about robust classification and estimation with mislabeled training data.

- Proved under the ideal condition assumption, Rank Pruning achieves perfect noise estimation and equivalent expected risk as learning with the uncorrupted, hidden labels. I provided closed-form solutions when those assumptions are relaxed.

- Demonstrated that Rank Pruning performance generalizes across the number of training examples, feature dimension, fraction of mislabeling, and fraction of added noise examples drawn from a third distribution.

- Improved the state-of-the-art of $\tilde{P}\tilde{N}$ learning across F1 score, AUC-PR, and Error. In many cases, Rank Pruning achieves nearly the same F1 score as learning with correct labels when 50% of positive examples are mislabeled and 50% of observed positive labels are mislabeled negative examples.

5.2 CAMEO and Rank Pruning

The CAMEO and Rank Pruning algorithms are developed as a joint solution to *multiple-account* cheating detection, yet their contributions are developed and presented independently. This is by design. In this section, I justify this choice along with a glimpse into the application of Rank Pruning to noisy CAMEO labels.

To clarify why the CAMEO and Rank Pruning algorithms are developed independently, I explain what this thesis is and what it is not. The purpose of this thesis is not to portray the combination of the CAMEO and Rank Pruning algorithms as a unified and exclusively-appropriate solution to CAMEO detection. Nor is the purpose to exclude other approaches like Item Response Theory or SPARFA. Both of these approaches are substantial contributions to the fields of measurement theory and community detection and in certain contexts, may be potential candidates for generating noisy CAMEO cheating labels.

The purpose of this thesis is to present two separate, but complimentary algorithms that can jointly serve as a robust, general solution to *multiple-account* cheating detection, but separately serve as independent solutions to other problems. Kept apart from Rank Pruning, the CAMEO algorithm provides a lower-bound estimate of CAMEO prevalence and fundamental insights into how one can operationalize multiple-account coordination. Kept apart from the CAMEO algorithm, Rank Pruning provides a general, time-efficient, robust solution for binary classification with noisy labels. This separation-by-design choice emphasizes that Rank Pruning is not tied to CAMEO labels and could take noisy Item Response Theory or SPARFA labels as input instead, while also ensuring that Rank Pruning does not overshadow the important insights that the CAMEO algorithm enables for operationalizing CAMEO behavior.

However, because this thesis is partly entitled *multiple-account* cheating detection in open online, I would be remiss not to include a brief discussion of their joint application. As mentioned in Section 3.1, I augmented CAMEO with hundreds of additional rules, each operationalizing a particular variation of CAMEO behavior. This *extended-CAMEO* algorithm was applied to 204 MITx and HarvardX courses and Rank Pruning was applied transductively (Gammerman et al., 1998) to the generated noisy labels. Among the 184 courses having at least one noisily detected CAMEO user, Rank Pruning estimated that for every 100 cheaters, the extended-CAMEO algorithm missed 13 on average, and for every 1000 non-cheaters the extended-CAMEO algorithm missed two on average ($\rho_1 = 0.13$, and $\rho_0 = 0.002$).

A subset of these 184 courses contained only one noisy extended-CAMEO cheating label, making Rank Pruning underdetermined and thus producing noise rates of zero ($\rho_1 = 0$, and $\rho_0 = 0$). Across the 135 courses with $\rho_1 \neq 0$, Rank Pruning estimated that for every 100 cheaters, the extended-CAMEO algorithm missed 18 on average, and across the 75 courses with $\rho_0 \neq 0$, for every 1000 non-cheaters the extended-CAMEO algorithm missed seven on average ($\rho_1 = 0.18$, and $\rho_0 = 0.007$). A medium-scale validation experiment like the one described in Section 2.2.3 was used to verify that the combination of extended-CAMEO and Rank Pruning achieved over

99% accuracy across three courses. A gradient-boosted logistic regression classifier with transductive hyper-parameter optimization was used by Rank Pruning for this analysis.

5.3 Cheating Detection versus Prevention

In this section, I motivate the benefits of cheating detection over cheating prevention and discusses how this relates to learning.

Methods for CAMEO prevention exist, but they are time-intensive to implement or may inhibit learning. Four such methods are (1) eliminate the *show answer* option, (2) restrict visibility of *show answer* formative feedback until after assignments are due, (3) randomly sample problems from a larger *test bank* of problems, where the size of the larger *test bank* is inversely proportion to the probability of the CM and CH accounts receiving the same question for the same problem, and (4) algorithmically generate assessment items so that users receive randomly varying items, each with different solutions.

It is immediately obvious that there are downsides associated with each preventative mechanism. In response to rampant CAMEO cheating in MITx MicroMasters MOOCs in Fall of 2015, course teams implemented method (1) the following spring. Learners vehemently complained that this decision negatively impacted their learning experience. This decision to enable *show answer* is known as the *Assistance Dilemma* (Koedinger and Aleven, 2007) in Cognitive Tutoring learning, and although *show answer* is used for assessment (versus learning) problems in MOOCs, *immediate formative feedback* has been linked with more efficient, improved learning (Anderson et al., 1995). Method (2) improves on method (1) by providing the *show answer* option after assignments are due, but because submission windows are often 1-3 weeks, the delay prevents early feedback and still inhibits learning. In method (3), in order for a CM account and CH account to receive the same question $\sim 25\%$ of the time, the MOOC course staff would need to create 400% more content. This is unreasonably exhausting (Kellogg, 2013). Finally, method (4) only works for problems with numerical components that can be randomly generated, failing for most liberal arts and humanities questions.

It remains undiscovered if CAMEO cheating may be used strategically to promote learning outcomes. What can we learn about CAMEO users who outperform non-CAMEO users on assessment problems without using the *show answer* button? Can CAMEO users learn more efficiently than non-CAMEO users? These questions suggest that detection of cheating may pose advantages not afforded by strict prevention. Given that a foundational goal of online courses is to promote learning, best practices for cheating detection and prevention should not negatively impact learning outcomes.

By addressing cheating in online courses via detection (versus prevention), we avoid unnecessary restriction for course teams without compromising the learning experience for all users.

5.4 Repurposability

In this section, I illustrate how the CAMEO algorithm can be re-purposed with two examples.

Although the focus of this analysis is confined to MITx and HarvardX courses, I argue that the (1) techniques for analysis and features presented are applicable to many domains, particularly where users can *game the system* (Baker et al., 2004b) by coordinating multiple accounts. For example in economic theory, shill bidding (Chakraborty and Kosmopoulou, 2004) is a multiple-account strategy used to raise auction prices (Kauffman and Wood, 2005) by bidding on one's own auction. Although the problem domains of shill bidding and cheating detection appear drastically different, analysis of the Δt distribution for pairs of accounts may prove to be effective as a detection mechanism for shill bidding because both shill bidding and CAMEO cheating require coordinating multiple accounts (Marshall and Marx, 2007) within a system that tracks user activity.

Beyond domain, minor modifications of the techniques used by multiple-account cheating detection allow tackling of new problems. For example, by re-defining the definition of Δt to measure the time difference between one account's CA time-stamps and another account's CA time-stamps, ignoring *show answer* click all together, we can detect collaboration instead of copying. The ability to re-purpose CAMEO detection with a single modification is indicative of its generalizability.

These examples advocate the repurposability of the technical approaches of the CAMEO algorithm and their applicability to a broader context of problems.

5.5 One Small Step for Science, One Giant Leap for Education

Massive open online courses hold the promise of a future where learners all over the world can learn from the best institutions for free. But this *democratization of education* is limited to the legitimacy of the certification process. CAMEO sets a precedent for this legitimacy by providing a foundation of consistent value for MOOC certificates as a reliable academic credential.

As a member of the MOOC research community, I continue to look forward to the unprecedented opportunities made available by massive learner interaction datasets. Yet of far greater importance is the questions we ask about these datasets, and the techniques we use to answer those questions. I hope that this thesis serves to spark new questions, while providing foundational tools to answer those questions. I recommend and look forward to future interventions that encourage honest behavior in online learning environments while disallowing and discouraging cheating in all its forms.

Beyond cheating detection, this work helps establish a foundation for reputable certification for hardworking learners all over the world, particularly learners in underdeveloped and developing nations. I genuinely hope that these learners can use these certificates to demonstrate what they've accomplished, and perhaps even improve their lives. I strongly encourage future endeavors toward sustainable enhancement of the perceived value of open-access academic credentials. I hope that the work in this thesis brings us one step closer to the intended promise of accessible education for everyone.

Appendix A

Appendix

A.1 Proofs

In this section, we provide proofs for all the lemmas and theorems in the main paper. For all the theorems and lemmas, we assume that a class-conditional extension of the Classification Noise Process (CNP) (Angluin and Laird, 1988) maps true labels y to observed labels s such that each label in P is flipped independently with probability ρ_1 and each label in N is flipped independently with probability ρ_0 ($s \leftarrow CNP(y, \rho_1, \rho_0)$), so that $P(s = s|y = y, x) = P(s = s|y = y)$. Remember that $\rho_1 + \rho_0 < 1$ is a necessary condition of minimal information, other we may learn opposite labels.

In Lemma 1, Theorem 2, Lemma 3 and Theorem 4, we assume that P and N have infinite number of examples so that they are the true, hidden distributions.

A very important equation we will use in the proofs is the following lemma:

Lemma A1 *When g is ideal, i.e. $g(x) = g^*(x)$ and P and N have non-overlapping support, we have*

$$g(x) = (1 - \rho_1) \cdot \mathbb{1}[[y = 1]] + \rho_0 \cdot \mathbb{1}[[y = 0]] \tag{A.1}$$

Proof: Since $g(x) = g^*(x)$ and P and N have non-overlapping support, we have

$$\begin{aligned} g(x) =& g^*(x) = P(s=1|x) \\ =& P(s=1|y=1,x) \cdot P(y=1|x) + P(s=1|y=0,x) \cdot P(y=0|x) \\ =& P(s=1|y=1) \cdot P(y=1|x) + P(s=1|y=0) \cdot P(y=0|x) \\ =& (1-\rho_1) \cdot \mathbb{1}[[y=1]] + \rho_0 \cdot \mathbb{1}[[y=0]] \end{aligned}$$

A.1.1 Proof of Lemma 1

Lemma 1 *When g is ideal, i.e. $g(x) = g^*(x)$ and P and N have non-overlapping support, we have*

$$\begin{cases} \tilde{P}_{y=1} = \{x \in P | s=1\}, \tilde{N}_{y=1} = \{x \in P | s=0\} \\ \tilde{P}_{y=0} = \{x \in N | s=1\}, \tilde{N}_{y=0} = \{x \in N | s=0\} \end{cases} \tag{A.2}$$

Proof: Firstly, we compute the threshold $LB_{y=1}$ and $UB_{y=0}$ used by $\tilde{P}_{y=1}$, $\tilde{N}_{y=1}$, $\tilde{P}_{y=0}$ and $\tilde{N}_{y=0}$. Since P and N have non-overlapping support, we have $P(y=1|x) = \mathbb{1}[[y=1]]$. Also using $g(x) = g^*(x)$, we have

$$\begin{aligned} LB_{y=1} =& E_{x \in \tilde{P}}[g(x)] = E_{x \in \tilde{P}}[P(s=1|x)] \\ =& E_{x \in \tilde{P}}[P(s=1|x,y=1)P(y=1|x) + P(s=1|x,y=0)P(y=0|x)] \\ =& E_{x \in \tilde{P}}[P(s=1|y=1)P(y=1|x) + P(s=1|y=0)P(y=0|x)] \\ =& (1-\rho_1)(1-\pi_1) + \rho_0\pi_1 \end{aligned} \tag{A.3}$$

Similarly, we have

$$UB_{y=0} = (1-\rho_1)\pi_0 + \rho_0(1-\pi_0)$$

Since $\pi_1 = P(y=0|s=1)$, we have $\pi_1 \in [0,1]$. Furthermore, we have the requirement that $\rho_1 + \rho_0 < 1$, then $\pi_1 = 1$ will lead to $\rho_1 = P(s=0|y=1) = 1 - P(s=1|y=1) = 1 - \frac{P(y=1|s=1)P(s=1)}{P(y=1)} = 1 - 0 = 1$ which violates the requirement of $\rho_1 + \rho_0 < 1$. Therefore, $\pi_1 \in [0, 1)$. Similarly, we can prove $\pi_0 \in [0, 1)$. Therefore, we see that both $LB_{y=1}$ and $UB_{y=0}$ are interpolations of $(1 - \rho_1)$ and ρ_0:

$$\rho_0 < LB_{y=1} \leq 1 - \rho_1$$
$$\rho_0 \leq UB_{y=0} < 1 - \rho_1$$

The first equality holds iff $\pi_1 = 0$ and the second equality holds iff $\pi_0 = 0$.

Using Lemma A1, we know that under the condition of $g(x) = g^*(x)$ and non-overlapping support, $g(x) = (1 - \rho_1) \cdot \mathbb{1}[[y = 1]] + \rho_0 \cdot \mathbb{1}[[y = 0]]$. In other words,

$$g(x) \geq LB_{y=1} \Leftrightarrow x \in P$$
$$g(x) \leq UB_{y=0} \Leftrightarrow x \in N$$

Since

$$\begin{cases} \tilde{P}_{y=1} = \{x \in \tilde{P} | g(x) \geq LB_{y=1}\} \\ \tilde{N}_{y=1} = \{x \in \tilde{N} | g(x) \geq LB_{y=1}\} \\ \tilde{P}_{y=0} = \{x \in \tilde{P} | g(x) \leq UB_{y=0}\} \\ \tilde{N}_{y=0} = \{x \in \tilde{N} | g(x) \leq UB_{y=0}\} \end{cases}$$

where $\tilde{P} = \{x | s = 1\}$ and $\tilde{N} = \{x | s = 0\}$, we have

$$\begin{cases} \tilde{P}_{y=1} = \{x \in P | s = 1\}, \tilde{N}_{y=1} = \{x \in P | s = 0\} \\ \tilde{P}_{y=0} = \{x \in N | s = 1\}, \tilde{N}_{y=0} = \{x \in N | s = 0\} \end{cases}$$

A.1.2 Proof of Theorem 2

We restate Theorem 2 here:

Theorem 2 *When g is ideal, i.e. $g(x) = g^*(x)$ and P and N have non-overlapping support, we have*

$$\hat{\rho}_1^{conf} = \rho_1, \hat{\rho}_0^{conf} = \rho_0$$

Proof: Using the definition of $\hat{\rho}_1^{conf}$ in the main paper:

$$\hat{\rho}_1^{conf} = \frac{|\tilde{N}_{y=1}|}{|\tilde{N}_{y=1}| + |\tilde{P}_{y=1}|},\ \hat{\rho}_0^{conf} = \frac{|\tilde{P}_{y=0}|}{|\tilde{P}_{y=0}| + |\tilde{N}_{y=0}|}$$

Since $g(x) = g^*(x)$ and P and N have non-overlapping support, using Lemma 1, we know

$$\begin{cases} \tilde{P}_{y=1} = \{x \in P | s = 1\}, \tilde{N}_{y=1} = \{x \in P | s = 0\} \\ \tilde{P}_{y=0} = \{x \in N | s = 1\}, \tilde{N}_{y=0} = \{x \in N | s = 0\} \end{cases}$$

Since $\rho_1 = P(s = 0|y = 1)$ and $\rho_0 = P(s = 1|y = 0)$, we immediately have

$$\hat{\rho}_1^{conf} = \frac{|\{x \in P | s = 0\}|}{|P|} = \rho_1,\ \hat{\rho}_0^{conf} = \frac{|\{x \in N | s = 1\}|}{|N|} = \rho_0$$

A.1.3 Proof of Lemma 3

We rewrite Lemma 3 below:

Lemma 3 *When g is unassuming, i.e., $\Delta g(x) := g(x) - g^*(x)$ can be nonzero, and P and N can have overlapping support, we have*

$$\begin{cases} LB_{y=1} = LB^*_{y=1} + \langle \Delta g(x) \rangle_{s1} - \frac{(1-\rho_1-\rho_0)^2}{p_{s1}} \Delta p_o \\ UB_{y=0} = UB^*_{y=0} + \langle \Delta g(x) \rangle_{s0} + \frac{(1-\rho_1-\rho_0)^2}{1-p_{s1}} \Delta p_o \\ \hat{\rho}_1^{conf} = \rho_1 + \frac{1-\rho_1-\rho_0}{|P|-|\Delta P_1|+|\Delta N_1|} |\Delta N_1| \\ \hat{\rho}_0^{conf} = \rho_0 + \frac{1-\rho_1-\rho_0}{|N|-|\Delta N_0|+|\Delta P_0|} |\Delta P_0| \end{cases} \tag{A.4}$$

where

$$\begin{cases} LB^*_{y=1} = (1-\rho_1)(1-\pi_1) + \rho_0\pi_1 \\ UB^*_{y=0} = (1-\rho_1)\pi_0 + \rho_0(1-\pi_0) \\ \Delta p_o := \frac{|P\cap N|}{|P\cup N|} \\ \Delta P_1 = \{x \in P | g(x) < LB_{y=1}\} \\ \Delta N_1 = \{x \in N | g(x) \geq LB_{y=1}\} \\ \Delta P_0 = \{x \in P | g(x) \leq UB_{y=0}\} \\ \Delta N_0 = \{x \in N | g(x) > UB_{y=0}\} \end{cases} \tag{A.5}$$

Proof: We first calculate $LB_{y=1}$ and $UB_{y=0}$ under unassuming condition, then calculate $\hat{\rho}_i^{conf}$, $i = 0, 1$ under unassuming condition.

Note that Δp_o can also be expressed as

$$\Delta p_o := \frac{|P \cap N|}{|P \cup N|} = P(\hat{y} = 1, y = 0) = P(\hat{y} = 0, y = 1)$$

Here $P(\hat{y} = 1, y = 0) \equiv P(\hat{y} = 1|y = 0)P(y = 0)$, where $P(\hat{y} = 1|y = 0)$ means for a perfect classifier $f^*(x) = P(y = 1|x)$, the expected probability that it will label a $y = 0$ example as positive ($\hat{y} = 1$).

(1) $LB_{y=1}$ and $UB_{y=0}$ under unassuming condition

Firstly, we calculate $LB_{y=1}$ and $UB_{y=0}$ with perfect probability estimation $g^*(x)$, but the support may overlap. Secondly, we allow the probability estimation to be imperfect, superimposed onto the overlapping support condition, and calculate $LB_{y=1}$ and $UB_{y=0}$.

I. Calculating $LB_{y=1}$ and $UB_{y=0}$ when $g(x) = g^*(x)$ and support may overlap

With overlapping support, we no longer have $P(y = 1|x) = \mathbb{1}[[y = 1]]$. Instead, we have

$$
\begin{aligned}
LB_{y=1} =& E_{x\in\tilde{P}}[g^*(x)] = E_{x\in\tilde{P}}[P(s=1|x)]\\
=& E_{x\in\tilde{P}}[P(s=1|x,y=1)P(y=1|x) + P(s=1|x,y=0)P(y=0|x)]\\
=& E_{x\in\tilde{P}}[P(s=1|y=1)P(y=1|x) + P(s=1|y=0)P(y=0|x)]\\
=& (1-\rho_1)\cdot E_{x\in\tilde{P}}[P(y=1|x)] + \rho_0\cdot E_{x\in\tilde{P}}[P(y=0|x)]\\
=& (1-\rho_1)\cdot P(\hat{y}=1|s=1) + \rho_0\cdot P(\hat{y}=0|s=1)
\end{aligned}
$$

Here $P(\hat{y}=1|s=1)$ can be calculated using Δp_o:

$$
\begin{aligned}
P(\hat{y}=1|s=1) &= \frac{P(\hat{y}=1,s=1)}{P(s=1)}\\
&= \frac{P(\hat{y}=1,y=1,s=1) + P(\hat{y}=1,y=0,s=1)}{P(s=1)}\\
&= \frac{P(s=1|y=1)P(\hat{y}=1,y=1) + P(s=1|y=0)P(\hat{y}=1,y=0)}{P(s=1)}\\
&= \frac{(1-\rho_1)(p_{y1}-\Delta p_o) + \rho_0\Delta p_o}{p_{s1}}\\
&= (1-\pi_1) - \frac{1-\rho_1-\rho_0}{p_{s1}}\Delta p_o
\end{aligned}
$$

Hence,

$$
P(\hat{y}=0|s=1) = 1 - P(\hat{y}=1|s=1) = \pi_1 + \frac{1-\rho_1-\rho_0}{p_{s1}}\Delta p_o
$$

Therefore,

$$
\begin{aligned}
LB_{y=1} &= (1-\rho_1)\cdot P(\hat{y}=1|s=1) + \rho_0\cdot P(\hat{y}=0|s=1)\\
&= (1-\rho_1)\cdot\left((1-\pi_1) - \frac{1-\rho_1-\rho_0}{p_{s1}}\Delta p_o\right) + \rho_0\cdot\left(\pi_1 + \frac{1-\rho_1-\rho_0}{p_{s1}}\Delta p_o\right)\\
&= LB^*_{y=1} - \frac{(1-\rho_1-\rho_0)^2}{p_{s1}}\Delta p_o \qquad \text{(A.6)}
\end{aligned}
$$

where $LB^*_{y=1}$ is the $LB_{y=1}$ value when $g(x)$ is ideal. We see in Eq. (A.6) that the overlapping support introduces a non-positive correction to $LB^*_{y=1}$ compared with the ideal condition.

Similarly, we have

$$UB_{y=0} = UB^*_{y=0} + \frac{(1-\rho_1-\rho_0)^2}{1-p_{s1}}\Delta p_o \tag{A.7}$$

II. Calculating $LB_{y=1}$ and $UB_{y=0}$ when g is unassuming

Define $\Delta g(x) = g(x) - g^*(x)$. Also define $\langle\Delta g(x)\rangle_{s1} := E_{x\in\tilde{P}}[\Delta g(x)]$, $\langle\Delta g(x)\rangle_{s0} := E_{x\in\tilde{N}}[\Delta g(x)]$. When the support may overlap, we have

$$\begin{aligned} LB_{y=1} &= E_{x\in\tilde{P}}[g(x)] \\ &= E_{x\in\tilde{P}}[g^*(x)] + E_{x\in\tilde{P}}[\Delta g(x)] \\ &= LB^*_{y=1} - \frac{(1-\rho_1-\rho_0)^2}{p_{s1}}\Delta p_o + \langle\Delta g(x)\rangle_{s1} \end{aligned} \tag{A.8}$$

Similarly, we have

$$\begin{aligned} UB_{y=0} &= E_{x\in\tilde{N}}[g(x)] \\ &= E_{x\in\tilde{N}}[g^*(x)] + E_{x\in\tilde{N}}[\Delta g(x)] \\ &= UB^*_{y=0} + \frac{(1-\rho_1-\rho_0)^2}{1-p_{s1}}\Delta p_o + \langle\Delta g(x)\rangle_{s0} \end{aligned} \tag{A.9}$$

In summary, Eq. (A.8) (A.9) give the expressions for $LB_{y=1}$ and $UB_{y=0}$, respectively, when g is unassuming.

(2) $\hat{\rho}_i^{conf}$ under unassuming condition

Now let's calculate $\hat{\rho}_i^{conf}$, $i = 0, 1$. For simplicity, define

$$\begin{cases} PP = \{x \in P | s = 1\} \\ PN = \{x \in P | s = 0\} \\ NP = \{x \in N | s = 1\} \\ NN = \{x \in N | s = 0\} \\ \Delta_{PP_1} = \{x \in PP | g(x) < LB_{y=1}\} \\ \Delta_{NP_1} = \{x \in NP | g(x) \geq LB_{y=1}\} \\ \Delta_{PN_1} = \{x \in PN | g(x) < LB_{y=1}\} \\ \Delta_{NN_1} = \{x \in NN | g(x) \geq LB_{y=1}\} \end{cases} \tag{A.10}$$

For $\hat{\rho}_1^{conf}$, we have:

$$\hat{\rho}_1^{conf} = \frac{|\tilde{N}_{y=1}|}{|\tilde{P}_{y=1}| + |\tilde{N}_{y=1}|}$$

Here

$$\begin{aligned} \tilde{P}_{y=1} &= \{x \in \tilde{P} | g(x) \geq LB_{y=1}\} \\ &= \{x \in PP | g(x) \geq LB_{y=1}\} \cup \{x \in NP | g(x) \geq LB_{y=1}\} \\ &= (PP \setminus \Delta_{PP_1}) \cup \Delta_{NP_1} \end{aligned}$$

Similarly, we have

$$\tilde{N}_{y=1} = (PN \setminus \Delta_{PN_1}) \cup \Delta_{NN_1}$$

Therefore

$$\begin{aligned}\hat{\rho}_1^{conf} &= \frac{|PN| - |\Delta_{PN_1}| + |\Delta_{NN_1}|}{[(|PP| - |\Delta_{PP_1}|) + (|PN| - |\Delta_{PN_1}|)] + (|\Delta_{NN_1}| + |\Delta_{NP_1}|)} \\ &= \frac{|PN| - |\Delta_{PN_1}| + |\Delta_{NN_1}|}{|P| - |\Delta P_1| + |\Delta N_1|}\end{aligned} \tag{A.11}$$

where in the second equality we have used the definition of ΔP_1 and ΔN_1 in Eq. (A.5).

Using the definition of ρ_1, we have

$$\begin{aligned}\frac{|PN| - |\Delta_{PN_1}|}{|P| - |\Delta P_1|} &= \frac{|\{x \in PN | g(x) \geq LB_{y=1}\}|}{|\{x \in P | g(x) \geq LB_{y=1}\}|} \\ &= \frac{P(x \in PN, g(x) \geq LB_{y=1})}{P(x \in P, g(x) \geq LB_{y=1})} \\ &= \frac{P(x \in PN | x \in P, g(x) \geq LB_{y=1}) \cdot P(x \in P, g(x) \geq LB_{y=1})}{P(x \in P, g(x) \geq LB_{y=1})} \\ &= \frac{P(x \in PN | x \in P) \cdot P(x \in P, g(x) \geq LB_{y=1})}{P(x \in P, g(x) \geq LB_{y=1})} \\ &= \rho_1\end{aligned}$$

Here we have used the property of CNP that $(s \perp\!\!\!\perp x)|y$, leading to $P(x \in PN | x \in P, g(x) \geq LB_{y=1}) = P(x \in PN | x \in P) = \rho_1$.

Similarly, we have

$$\frac{|\Delta_{NN_1}|}{|\Delta N_1|} = 1 - \rho_0$$

Combining with Eq. (A.11), we have

$$\hat{\rho}_1^{conf} = \rho_1 + \frac{1 - \rho_1 - \rho_0}{|P| - |\Delta P_1| + |\Delta N_1|} |\Delta N_1| \tag{A.12}$$

Similarly, we have

$$\hat{\rho}_0^{conf} = \rho_0 + \frac{1 - \rho_1 - \rho_0}{|N| - |\Delta N_0| + |\Delta P_0|}|\Delta P_0| \tag{A.13}$$

From the two equations above, we see that

$$\hat{\rho}_1^{conf} \geq \rho_1, \hat{\rho}_0^{conf} \geq \rho_0 \tag{A.14}$$

In other words, $\hat{\rho}_i^{conf}$ is an ***upper bound*** of ρ_i, $i = 0, 1$. The equality for $\hat{\rho}_1^{conf}$ holds if $|\Delta N_1| = 0$. The equality for $\hat{\rho}_0^{conf}$ holds if $|\Delta P_0| = 0$.

A.1.4 Proof of Theorem 4

Let's restate Theorem 4 below:

Theorem 4 *Given non-overlapping support condition,*

If $\forall x \in N, \Delta g(x) < LB_{y=1} - \rho_0$, then $\hat{\rho}_1^{conf} = \rho_1$.

If $\forall x \in P, \Delta g(x) > -(1 - \rho_1 - UB_{y=0})$, then $\hat{\rho}_0^{conf} = \rho_0$.

Theorem 4 directly follows from Eq. (A.12) and (A.13). Assuming non-overlapping support, we have $g^*(x) = P(s = 1|x) = (1 - \rho_1) \cdot \mathbb{1}[[y = 1]] + \rho_0 \cdot \mathbb{1}[[y = 0]]$. In other words, the contribution of overlapping support to $|\Delta N_1|$ and $|\Delta P_0|$ is 0. The only source of deviation comes from imperfect $g(x)$.

For the first half of the theorem, since $\forall x \in N, \Delta g(x) < LB_{y=1} - \rho_0$, we have $\forall x \in N, g(x) = \Delta g(x) + g^*(x) < (LB_{y=1} - \rho_0) + \rho_0 = LB_{y=1}$, then $|\Delta N_1| = |\{x \in N|g(x) \geq LB_{y=1}\}| = 0$, so we have $\hat{\rho}_1^{conf} = \rho_1$.

Similarly, for the second half of the theorem, since $\forall x \in P, \Delta g(x) > -(1 - \rho_1 - UB_{y=0})$, then $|\Delta P_0| = |\{x \in P|g(x) \leq UB_{y=0}\}| = 0$, so we have $\hat{\rho}_0^{conf} = \rho_0$.

A.1.5 Proof of Theorem 5

Theorem 5 reads as follows:

Theorem 5 *If g range separates P and N and $\hat{\rho}_i = \rho_i$, $i = 0, 1$, then for any classifier f_θ and any bounded loss function $l(\hat{y}_i, y_i)$, we have*

$$R_{\tilde{l},\mathcal{D}_\rho}(f_\theta) = R_{l,\mathcal{D}}(f_\theta) \tag{A.15}$$

where $\tilde{l}(\hat{y}_i, s_i)$ is Rank Pruning's loss function given by

$$\tilde{l}(\hat{y}_i, s_i) = \frac{1}{1-\hat{\rho}_1} l(\hat{y}_i, s_i) \cdot \mathbb{1}[[x_i \in \tilde{P}_{conf}]] + \frac{1}{1-\hat{\rho}_0} l(\hat{y}_i, s_i) \cdot \mathbb{1}[[x_i \in \tilde{N}_{conf}]] \tag{A.16}$$

and $\tilde{P}_{conf}$ and $\tilde{N}_{conf}$ are given by

$$\tilde{P}_{conf} := \{x \in \tilde{P} \mid g(x) \geq k_1\}, \tilde{N}_{conf} := \{x \in \tilde{N} \mid g(x) \leq k_0\} \tag{A.17}$$

where k_1 is the $(\hat{\pi}_1|\tilde{P}|)^{th}$ smallest $g(x)$ for $x \in \tilde{P}$ and k_0 is the $(\hat{\pi}_0|\tilde{N}|)^{th}$ largest $g(x)$ for $x \in \tilde{N}$

Proof:

Since $\tilde{P}$ and $\tilde{N}$ are constructed from P and N with noise rates π_1 and π_0 using the class-conditional extension of the Classification Noise Process (Angluin and Laird, 1988), we have

$$\begin{cases} P = PP \cup PN \\ N = NP \cup NN \\ \tilde{P} = PP \cup NP \\ \tilde{N} = PN \cup NN \end{cases} \tag{A.18}$$

where

$$\begin{cases} PP = \{x \in P | s = 1\} \\ PN = \{x \in P | s = 0\} \\ NP = \{x \in N | s = 1\} \\ NN = \{x \in N | s = 0\} \end{cases} \tag{A.19}$$

satisfying

$$\begin{cases} PP \sim PN \sim P \\ NP \sim NN \sim N \\ \frac{|NP|}{|\tilde{P}|} = \pi_1, \frac{|PP|}{|\tilde{P}|} = 1 - \pi_1 \\ \frac{|PN|}{|\tilde{N}|} = \pi_0, \frac{|NN|}{|\tilde{N}|} = 1 - \pi_0 \\ \frac{|PN|}{|P|} = \rho_1, \frac{|PP|}{|P|} = 1 - \rho_1 \\ \frac{|NP|}{|N|} = \rho_0, \frac{|NN|}{|N|} = 1 - \rho_0 \end{cases} \tag{A.20}$$

Here the $\sim$ means obeying the same distribution.

Since g range separates P and N, there exists a real number z such that $\forall x_1 \in P$ and $\forall x_0 \in N$, we have $g(x_1) > z > g(x_0)$. Since $P = PP \cup PN$, $N = NP \cup NN$, we have

$$\begin{aligned} &\forall x \in PP, g(x) > z;\ \forall x \in PN, g(x) > z; \\ &\forall x \in NP, g(x) < z;\ \forall x \in NN, g(x) < z \end{aligned} \tag{A.21}$$

Since $\hat{\rho}_1 = \rho_1$ and $\hat{\rho}_0 = \rho_0$, we have

$$\begin{cases} \hat{\pi}_1 = \frac{\hat{\rho}_0}{p_{s1}} \frac{1-p_{s1}-\hat{\rho}_1}{1-\hat{\rho}_1-\hat{\rho}_0} = \frac{\rho_0}{p_{s1}} \frac{1-p_{s1}-\rho_1}{1-\rho_1-\rho_0} = \pi_1 \equiv \frac{\rho_0|N|}{|\tilde{P}|} \\ \hat{\pi}_0 = \frac{\hat{\rho}_1}{1-p_{s1}} \frac{p_{s1}-\hat{\rho}_0}{1-\hat{\rho}_1-\hat{\rho}_0} = \frac{\rho_1}{1-p_{s1}} \frac{p_{s1}-\rho_0}{1-\rho_1-\rho_0} = \pi_0 \equiv \frac{\rho_1|P|}{|\tilde{N}|} \end{cases} \tag{A.22}$$

Therefore, $\hat{\pi}_1|\tilde{P}| = \pi_1|\tilde{P}| = \rho_0|N|$, $\hat{\pi}_0|\tilde{N}| = \pi_0|\tilde{N}| = \rho_1|P|$. Using $\tilde{P}_{conf}$ and $\tilde{N}_{conf}$'s definition in Eq. (A.17), and $g(x)$'s property in Eq. (A.21), we have

$$\tilde{P}_{conf} = PP \sim P, \tilde{N}_{conf} = NN \sim N \tag{A.23}$$

Hence P_{conf} and N_{conf} can be seen as a uniform downsampling of P and N, with a downsampling ratio of $(1 - \rho_1)$ for P and $(1 - \rho_0)$ for N. Then according to Eq. (A.16), the loss function $\tilde{l}(\hat{y}_i, s_i)$ essentially sees a fraction of $(1 - \rho_1)$ examples in P and a fraction of $(1 - \rho_0)$ examples in N, with a final reweighting to restore the class balance. Then for any classifier f_θ that maps $x \to \hat{y}$ and any bounded loss function

$l(\hat{y}_i, y_i)$, we have

$$\begin{aligned}
R_{\tilde{l},\mathcal{D}_\rho}(f_\theta) &= E_{(x,s)\sim\mathcal{D}_\rho}[\tilde{l}(f_\theta(x), s)] \\
&= \frac{1}{1-\hat{\rho}_1} \cdot E_{(x,s)\sim\mathcal{D}_\rho} \left[l(f_\theta(x), s) \cdot \mathbb{1}[[x \in \tilde{P}_{conf}]] \right] + \frac{1}{1-\hat{\rho}_0} \cdot E_{(x,s)\sim\mathcal{D}_\rho} \left[l(f_\theta(x), s) \cdot \mathbb{1}[[x \in \tilde{N}_{conf}]] \right. \\
&= \frac{1}{1-\rho_1} \cdot E_{(x,s)\sim\mathcal{D}_\rho} \left[l(f_\theta(x), s) \cdot \mathbb{1}[[x \in \tilde{P}_{conf}]] \right] + \frac{1}{1-\rho_0} \cdot E_{(x,s)\sim\mathcal{D}_\rho} \left[l(f_\theta(x), s) \cdot \mathbb{1}[[x \in \tilde{N}_{conf}]] \right. \\
&= \frac{1}{1-\rho_1} \cdot E_{(x,s)\sim\mathcal{D}_\rho} \left[l(f_\theta(x), s) \cdot \mathbb{1}[[x \in PP]] \right] + \frac{1}{1-\rho_0} \cdot E_{(x,s)\sim\mathcal{D}_\rho} \left[l(f_\theta(x), s) \cdot \mathbb{1}[[x \in NN]] \right] \\
&= \frac{1}{1-\rho_1} \cdot (1-\rho_1) \cdot E_{(x,y)\sim\mathcal{D}} \left[l(f_\theta(x), y) \cdot \mathbb{1}[[x \in P]] \right] + \frac{1}{1-\rho_0} \cdot (1-\rho_0) \cdot E_{(x,y)\sim\mathcal{D}} \left[l(f_\theta(x), y) \cdot \right. \\
&= E_{(x,y)\sim\mathcal{D}} \left[l(f_\theta(x), y) \cdot \mathbb{1}[[x \in P]] + l(f_\theta(x), y) \cdot \mathbb{1}[[x \in N]] \right] \\
&= E_{(x,y)\sim\mathcal{D}} \left[l(f_\theta(x), y) \right] \\
&= R_{l,\mathcal{D}}(f_\theta)
\end{aligned}$$

Therefore, we see that the expected risk for Rank Pruning with corrupted labels, is exactly the same as the expected risk for the true labels, for any bounded loss function l and classifier f_θ. The reweighting ensures that after pruning, the two sets still remain unbiased w.r.t. to the true dataset.

Since the ideal condition is more strict than the range separability condition, we immediately have that when g is ideal and $\hat{\rho}_i = \rho_i$, $i = 0, 1$, $R_{\tilde{l},\mathcal{D}_\rho}(f_\theta) = R_{l,\mathcal{D}}(f_\theta)$ for any f_θ and bounded loss function l.

A.2 Additional Figures

Figure BA.1 shows the average image for each digit for the problem “1" or “not 1" in MNIST with logistic regression and high noise ($\rho_1 = 0.5, \pi_1 = 0.5$). The number on the bottom and on the right counts the total number of examples (images). From the figure we see that RP makes few mistakes, and when it does, the mistakes vary greatly in image from the typical digit.

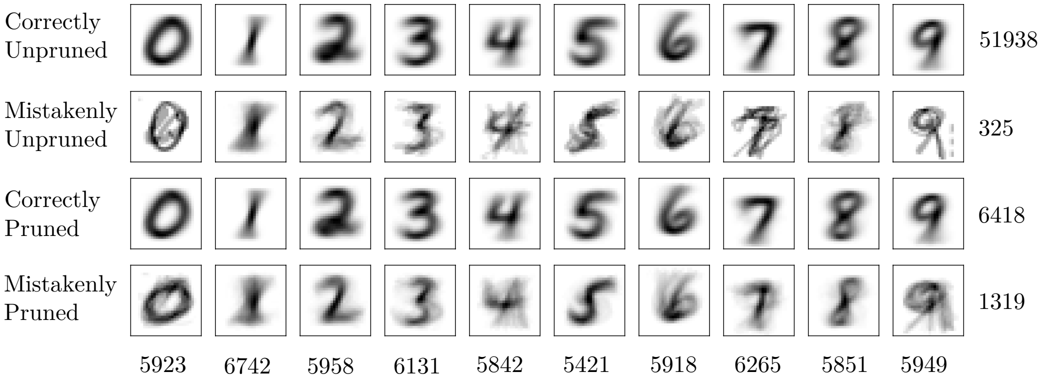


Figure B A.1: Average image for each digit for the problem "1" or "not 1" in MNIST with logistic regression and high noise ($\rho_1 = 0.5, \pi_1 = 0.5$). The number on the bottom and on the right counts the total number of examples in the corresponding column or row.

A.3 Additional Tables

Here we provide additional tables for the comparison of error, Precision-Recall AUC (AUC-PR, Davis and Goadrich (2006)), and F1 score for the algorithms *RP*, *Nat13*, *Elk08*, *Liu16* with ρ_1, ρ_0 given to all methods for fair comparison. Additionally, we provide the performance of the ground truth classifier (*true*), and RP_ρ with Rank Pruning's own estimated noise rates. The top model scores are in bold with RP_ρ in red if its performance is better than non-RP models. The $\pi_1 = 0$ quadrant in each table represents the "PU learning" case of $\tilde{P}\tilde{N}$ learning.

Whenever the $g(x) = P(\hat{s} = 1|x)$ is to be estimated for any algorithm, we use a 3-fold cross-validation to estimate the probability $g(x)$.

For the logistic regression classifier, we use scikit-learn's LogisticRegression class (scikit learn (2016)) with default setting (L2 regularization with inverse strength $C = 1$).

For the convolutional neural network (CNN), for MNIST we use the structure in Chollet (2016b) and for CIFAR-10, we use the structure in Chollet (2016a). A 10% holdout set monitors the weighted validation loss (using the sample weight given by each algorithm) and ends training when there is no decrease for 10 epochs, with a maximum of 50 epochs for MNIST and 150 epochs for CIFAR-10.

The following list comprises the MNIST and CIFAR experimental result tables for error, AUC-PR and F1 score metrics:

Table CA.1: Error for MNIST with logisitic regression as classifier.

Table CA.2: AUC-PR for MNIST with logisitic regression as classifier.

Table CA.3: Error for MNIST with CNN as classifier.

Table CA.4: AUC-PR for MNIST with CNN as classifier.

Table CA.5: F1 score for CIFAR-10 with logistic regression as classifier.

Table CA.6: Error for CIFAR-10 with logistic regression as classifier.

Table CA.7: AUC-PR for CIFAR-10 with logistic regression as classifier.

Table CA.8: Error for CIFAR-10 with CNN as classifier.

Table CA.9: AUC-PR for CIFAR-10 with CNN as classifier.

Due to its sensitivity to imperfect probability estimation, here *Liu16* always predicts all labels to be positive or negative, resulting in the same metric score for every digit/image in each scenario. Since $p_{y1} \simeq 0.1$, when predicting all labels as positive, *Liu16* has an F1 score of 0.182, error of 0.90, and AUC-PR of 0.55; when predicting all labels as negative, *Liu16* has an F1 score of 0.0, error of 0.1, and AUC-PR of 0.55.

A.4 Additional Related Work

In this section we include tangentially related work which was unable to make it into the final manuscript.

A.4.1 One-class classification

One-class classification (Moya et al., 1993) is distinguished from binary classification by a training set containing examples from only one class, making it useful for outlier and novelty detection (Hempstalk et al., 2008). This can be framed as $\tilde{P}\tilde{N}$ learning when outliers take the form of mislabeled examples. The predominant approach, one-class SVM, fits a hyper-boundary around the training class (Platt et al., 1999), but often performs poorly due to boundary over-sensitivity (Manevitz and Yousef, 2002) and fails when the training class contains mislabeled examples.

A.4.2 $\tilde{P}\tilde{N}$ learning for Image Recognition and Deep Learning

Variations of $\tilde{P}\tilde{N}$ learning have been used in the context of machine vision to improve robustness to mislabeling (Xiao et al., 2015). In a face recognition task with 90% of non-faces mislabeled as faces, a bagging model combined with consistency voting was used to remove images with poor voting consistency (Angelova et al., 2005). However, no theoretical justification was provided. In the context of deep learning, consistency of predictions for inputs with mislabeling enforces can be enforced by combining a typical cross-entropy loss with an auto-encoder loss (Reed et al., 2015). This method enforces label consistency by constraining the network to uncover the input examples given the output prediction, but is restricted in architecture and generality.

Table C A.1: Comparison of error for one-vs-rest MNIST (averaged over all digits) using logistic regression as classifier. Except for RP_ρ, ρ_1, ρ_0 are given to all methods. Top model scores are in bold with RP_ρ in red if better (smaller) than non-RP models.

	π_1 = 0			π_1 = 0.25				π_1 = 0.5				π_1 = 0.75			
MODEL,ρ_1 =	**0.25**	**0.50**	**0.75**	**0.00**	**0.25**	**0.50**	**0.75**	**0.00**	**0.25**	**0.50**	**0.75**	**0.00**	**0.25**	**0.50**	**0.75**
TRUE	0.020	0.020	0.020	0.020	0.020	0.020	0.020	0.020	0.020	0.020	0.020	0.020	0.020	0.020	0.020
RP$_\rho$	**0.023**	**0.025**	**0.031**	**0.024**	**0.025**	**0.027**	**0.038**	0.040	0.037	0.039	0.049	0.140	0.128	0.133	0.151
RP	**0.022**	**0.025**	**0.031**	**0.021**	**0.024**	**0.027**	**0.035**	**0.023**	**0.027**	**0.031**	**0.043**	**0.028**	**0.036**	**0.045**	0.069
NAT13	0.025	0.030	0.038	0.025	0.029	0.034	0.042	0.030	0.033	0.038	0.047	0.035	0.039	0.046	**0.067**
ELK08	0.025	0.030	0.038	0.026	0.028	0.032	0.042	0.030	0.031	0.035	0.051	0.092	0.093	0.123	0.189
LIU16	0.187	0.098	0.100	0.100	0.738	0.738	0.419	0.100	0.820	0.821	0.821	0.098	0.760	0.741	0.820

Table C A.2: Comparison of AUC-PR for one-vs-rest MNIST (averaged over all digits) using logistic regression as classifier. Except for RP_ρ, ρ_1, ρ_0 are given to all methods. Top model scores are in bold with RP_ρ in red if greater than non-RP models.

	π_1 = 0			π_1 = 0.25				π_1 = 0.5				π_1 = 0.75			
MODEL,ρ_1 =	**0.25**	**0.50**	**0.75**	**0.00**	**0.25**	**0.50**	**0.75**	**0.00**	**0.25**	**0.50**	**0.75**	**0.00**	**0.25**	**0.50**	**0.75**
TRUE	0.935	0.935	0.935	0.935	0.935	0.935	0.935	0.935	0.935	0.935	0.935	0.935	0.935	0.935	0.935
RP$_\rho$	0.921	**0.913**	**0.882**	**0.928**	**0.920**	**0.906**	**0.853**	**0.903**	**0.902**	**0.879**	**0.803**	0.851	0.835	0.788	0.640
RP	**0.922**	**0.913**	**0.882**	**0.930**	**0.921**	**0.906**	**0.858**	**0.922**	**0.903**	**0.883**	**0.811**	**0.893**	**0.841**	**0.799**	0.621
NAT13	**0.922**	0.908	0.878	0.918	0.909	0.890	0.839	0.899	0.892	0.862	0.794	0.863	0.837	0.784	**0.645**
ELK08	0.921	0.903	0.864	0.917	0.908	0.884	0.821	0.898	0.892	0.861	0.763	0.852	0.837	0.772	0.579
LIU16	0.498	0.549	0.550	0.550	0.500	0.550	0.505	0.550	0.550	0.550	0.549	0.503	0.512	0.550	0.550

Table C A.3: Comparison of error for one-vs-rest MNIST (averaged over all digits) using CNN as classifier. Except for RP_ρ, ρ_1, ρ_0 are given to all methods. Top model scores are in bold with RP_ρ in red if better (smaller) than non-RP models.

		π_1 = 0					π_1 = 0.25					π_1 = 0.5									
		ρ_1 = 0.5					ρ_1 = 0.25					ρ_1 = 0					ρ_1 = 0.5				
IMAGE	TRUE	RP$_\rho$	RP	NAT13	ELK08	LIU16	RP$_\rho$	RP	NAT13	ELK08	LIU16	RP$_\rho$	RP	NAT13	ELK08	LIU16	RP$_\rho$	RP	NAT13	ELK08	LIU16
0	0.0013	**0.0018**	**0.0023**	0.0045	0.0047	0.9020	**0.0017**	**0.0016**	0.0034	0.0036	0.9020	**0.0017**	**0.0016**	0.0031	0.0026	0.0029	**0.0021**	**0.0022**	0.0116	0.0069	0.9020
1	0.0015	**0.0022**	**0.0020**	0.0025	0.0034	0.8865	**0.0019**	**0.0019**	0.0035	0.0030	0.8865	0.0023	0.0020	0.0018	**0.0016**	0.0023	**0.0025**	**0.0025**	0.0036	0.0027	0.8865
2	0.0027	**0.0054**	**0.0049**	0.0057	0.0062	0.8968	**0.0032**	**0.0035**	0.0045	0.0051	0.8968	0.0030	0.0029	0.0031	0.0029	**0.0024**	**0.0059**	**0.0050**	0.0066	0.0083	0.8968
3	0.0020	**0.0032**	**0.0032**	0.0055	0.0038	0.8990	**0.0029**	**0.0029**	0.0043	0.0043	0.8990	**0.0021**	0.0027	**0.0023**	**0.0023**	0.0032	**0.0038**	**0.0042**	0.0084	0.0057	0.8990
4	0.0012	**0.0037**	0.0040	**0.0038**	0.0044	0.9018	**0.0029**	**0.0025**	0.0055	0.0069	0.9018	0.0026	0.0020	**0.0019**	0.0021	0.0030	**0.0044**	**0.0035**	0.0086	0.0077	0.9018
5	0.0019	**0.0032**	**0.0035**	0.0039	0.0038	0.9108	**0.0027**	**0.0031**	0.0062	0.0060	0.9108	**0.0021**	0.0024	0.0024	0.0028	**0.0023**	**0.0061**	**0.0056**	0.0066	0.0074	0.9108
6	0.0021	**0.0027**	**0.0028**	0.0053	0.0035	0.9042	**0.0028**	**0.0025**	0.0042	0.0036	0.9042	0.0029	0.0029	**0.0022**	0.0024	0.0028	**0.0032**	**0.0035**	0.0098	0.0075	0.9042
7	0.0026	**0.0039**	**0.0041**	0.0066	0.0103	0.8972	**0.0050**	**0.0052**	0.0058	0.0058	0.8972	0.0049	0.0040	**0.0030**	0.0037	0.0035	**0.0054**	**0.0064**	0.0113	0.0085	0.8972
8	0.0022	**0.0047**	**0.0043**	0.0106	0.0063	0.9026	**0.0034**	**0.0036**	0.0062	0.0091	0.9026	0.0036	**0.0030**	0.0035	0.0041	0.0032	**0.0044**	**0.0048**	0.0234	0.0077	0.9026
9	0.0036	0.0067	**0.0052**	0.0056	0.0124	0.8991	**0.0048**	**0.0051**	0.0065	0.0064	0.8991	0.0048	0.0050	0.0051	**0.0043**	0.0059	**0.0081**	0.0114	0.0131	**0.0112**	0.8991
AVG	0.0021	**0.0038**	**0.0036**	0.0054	0.0059	0.9000	**0.0031**	**0.0032**	0.0050	0.0054	0.9000	0.0030	**0.0028**	**0.0028**	0.0029	0.0032	**0.0046**	**0.0049**	0.0103	0.0074	0.9000

Table C A.4: Comparison of AUC-PR for one-vs-rest MNIST (averaged over all digits) using CNN as classifier. Except for RP_ρ, ρ_1, ρ_0 are given to all methods. Top model scores are in bold with RP_ρ in red if greater than non-RP models.

		π_1 = 0					π_1 = 0.25					π_1 = 0.5									
		ρ_1 = 0.5					ρ_1 = 0.25					ρ_1 = 0					ρ_1 = 0.5				
IMAGE	TRUE	RP$_\rho$	RP	NAT13	ELK08	LIU16	RP$_\rho$	RP	NAT13	ELK08	LIU16	RP$_\rho$	RP	NAT13	ELK08	LIU16	RP$_\rho$	RP	NAT13	ELK08	LIU16
0	0.9998	**0.9992**	**0.9990**	0.9986	0.9982	0.5490	**0.9996**	**0.9996**	0.9986	0.9979	0.5490	**0.9989**	**0.9995**	0.9976	0.9979	0.9956	**0.9984**	**0.9982**	0.9963	0.9928	0.5490
1	0.9999	**0.9995**	**0.9995**	0.9976	0.9974	0.5568	**0.9996**	0.9993	**0.9995**	**0.9995**	0.5568	**0.9995**	**0.9998**	0.9982	0.9972	0.9965	**0.9995**	**0.9994**	0.9978	0.9985	0.5568
2	0.9994	**0.9971**	**0.9969**	0.9917	0.9942	0.5516	**0.9980**	**0.9977**	0.9971	0.9945	0.5516	**0.9988**	**0.9992**	0.9958	0.9934	0.9940	**0.9938**	**0.9947**	0.9847	0.9873	0.5516
3	0.9996	**0.9986**	**0.9987**	0.9983	0.9984	0.5505	**0.9991**	**0.9989**	0.9982	0.9980	0.5505	**0.9993**	**0.9994**	0.9991	0.9971	0.9974	**0.9969**	**0.9959**	0.9951	**0.9959**	0.5505
4	0.9997	0.9982	**0.9989**	0.9939	0.9988	0.0891	**0.9992**	**0.9991**	0.9976	0.9965	0.5491	**0.9994**	**0.9996**	0.9985	0.9978	0.9986	**0.9983**	**0.9977**	0.9961	0.9919	0.5491
5	0.9993	**0.9982**	**0.9976**	0.9969	0.9956	0.5446	**0.9986**	**0.9987**	0.9983	0.9979	0.5446	**0.9984**	**0.9982**	0.9971	0.9963	0.9929	**0.9958**	**0.9965**	0.9946	0.9934	0.5446
6	0.9987	**0.9976**	**0.9970**	0.9928	0.9931	0.5479	**0.9974**	**0.9980**	0.9956	0.9959	0.5479	**0.9968**	**0.9983**	0.9933	0.9950	0.9905	**0.9964**	0.9957	0.9942	**0.9961**	0.5479
7	0.9989	**0.9973**	**0.9972**	0.9965	0.9944	0.0721	0.9968	0.9973	0.9966	**0.9979**	0.5514	0.9969	**0.9983**	0.9961	0.9958	0.9974	**0.9933**	**0.9937**	0.9896	0.9886	0.5514
8	0.9996	**0.9974**	**0.9964**	**0.9964**	0.9946	0.5487	**0.9981**	**0.9981**	0.9973	0.9971	0.5487	**0.9983**	0.9988	0.9984	0.9976	**0.9989**	**0.9976**	**0.9975**	0.9873	0.9893	0.5487
9	0.9979	**0.9931**	**0.9951**	0.9901	0.9922	0.5504	**0.9935**	**0.9951**	0.9933	0.9920	0.5504	**0.9961**	**0.9951**	0.9924	0.9922	0.9912	**0.9877**	**0.9876**	0.9819	0.9828	0.5504
AVG	0.9993	**0.9976**	**0.9976**	0.9953	0.9957	0.4561	**0.9980**	**0.9982**	0.9972	0.9967	0.5500	**0.9983**	**0.9986**	0.9966	0.9960	0.9953	**0.9958**	**0.9957**	0.9918	0.9917	0.5500

Table C A.5: Comparison of F1 score for one-vs-rest CIFAR-10 (averaged over all images) using logistic regression as classifier. Except for RP_ρ, ρ_1, ρ_0 are given to all methods. Top model scores are in bold with RP_ρ in red if greater than non-RP models.

		π_1 = 0					π_1 = 0.25					π_1 = 0.5									
		ρ_1 = 0.5					ρ_1 = 0.25					ρ_1 = 0					ρ_1 = 0.5				
IMAGE	TRUE	RP$_\rho$	RP	NAT13	ELK08	LIU16	RP$_\rho$	RP	NAT13	ELK08	LIU16	RP$_\rho$	RP	NAT13	ELK08	LIU16	RP$_\rho$	RP	NAT13	ELK08	LIU16
PLANE	0.272	**0.311**	**0.252**	0.217	0.220	0.182	**0.329**	**0.275**	0.222	0.224	0.182	**0.330**	**0.265**	0.231	0.259	0.0	**0.266**	**0.188**	0.183	0.187	0.182
AUTO	0.374	**0.389**	**0.355**	0.318	0.320	0.182	**0.388**	**0.368**	0.321	0.328	0.182	**0.372**	**0.355**	0.308	0.341	0.0	**0.307**	0.287	0.287	**0.297**	0.182
BIRD	0.136	**0.241**	0.167	0.143	0.136	**0.182**	**0.248**	**0.185**	0.137	0.137	0.182	**0.258**	**0.147**	0.100	0.126	0.0	**0.206**	0.153	0.132	0.150	**0.182**
CAT	0.122	**0.246**	0.170	0.141	0.150	**0.182**	**0.232**	0.163	0.112	0.127	**0.182**	**0.241**	**0.125**	0.068	0.103	0.0	**0.209**	0.148	0.119	0.157	**0.182**
DEER	0.166	**0.250**	**0.184**	0.153	0.164	0.182	**0.259**	0.175	0.146	0.163	**0.182**	**0.259**	**0.177**	0.126	0.164	0.0	**0.222**	0.162	0.132	0.164	**0.182**
DOG	0.139	**0.245**	0.174	0.146	0.148	**0.182**	**0.262**	0.171	0.115	0.126	**0.182**	**0.254**	**0.152**	0.075	0.120	0.0	**0.203**	0.151	0.128	0.137	**0.182**
FROG	0.317	**0.322**	**0.315**	0.289	0.281	0.182	**0.350**	**0.319**	0.283	0.299	0.182	**0.346**	**0.305**	0.239	0.279	0.0	**0.308**	0.252	0.244	**0.269**	0.182
HORSE	0.300	**0.300**	**0.299**	0.283	0.263	0.182	**0.334**	**0.313**	0.272	0.281	0.182	**0.322**	**0.310**	0.260	0.292	0.0	**0.275**	**0.258**	0.240	0.245	0.182
SHIP	0.322	**0.343**	**0.322**	0.297	0.272	0.182	**0.385**	**0.319**	0.287	0.289	0.182	**0.350**	**0.303**	0.250	0.293	0.0	**0.304**	**0.248**	0.230	0.237	0.182
TRUCK	0.330	**0.359**	**0.323**	0.273	0.261	0.182	**0.369**	**0.327**	0.293	0.290	0.182	**0.343**	**0.302**	0.278	0.299	0.0	**0.313**	0.246	0.252	**0.262**	0.182
AVG	0.248	**0.301**	**0.256**	0.226	0.221	0.182	**0.316**	**0.262**	0.219	0.226	0.182	**0.308**	**0.244**	0.194	0.228	0.000	**0.261**	0.209	0.195	**0.210**	0.182

Table C A.6: Comparison of error for one-vs-rest CIFAR-10 (averaged over all images) using logistic regression as classifier. Except for RP_ρ, ρ_1, ρ_0 are given to all methods. Top model scores are in bold with RP_ρ in red if better (smaller) than non-RP models. Here the logistic regression classifier severely underfits CIFAR, resulting in Rank Pruning pruning out some correctly labeled examples that "confuse" the classifier, hence in this scenario, RP and RP$_\rho$ generally have slightly smaller precision, much higher recall, and hence larger F1 scores than other models and even the ground truth classifier (Table CA.5). Due to the class inbalance ($p_{y1} = 0.1$) and their larger recall, RP and RP$_\rho$ here have larger error than the other models.

		$\pi_1 = \mathbf{0}$					$\pi_1 = \mathbf{0.25}$					$\pi_1 = \mathbf{0.5}$									
		$\rho_1 = \mathbf{0.5}$					$\rho_1 = \mathbf{0.25}$					$\rho_1 = \mathbf{0}$					$\rho_1 = \mathbf{0.5}$				
IMAGE	TRUE	RP$_\rho$	RP	NAT13	ELK08	LIU16	RP$_\rho$	RP	NAT13	ELK08	LIU16	RP$_\rho$	RP	NAT13	ELK08	LIU16	RP$_\rho$	RP	NAT13	ELK08	LIU16
PLANE	0.107	0.287	0.133	0.123	**0.122**	0.900	0.177	0.128	**0.119**	0.123	0.900	0.248	0.124	0.110	0.118	**0.100**	0.202	0.147	**0.142**	0.160	0.900
AUTO	0.099	0.184	0.120	**0.110**	**0.110**	0.900	0.132	0.114	**0.105**	0.109	0.900	0.189	0.110	0.105	0.110	**0.100**	0.159	0.129	**0.125**	0.139	0.900
BIRD	0.117	0.354	0.148	0.133	**0.131**	0.900	0.217	0.135	**0.120**	0.125	0.900	0.277	0.135	0.115	0.123	**0.100**	0.226	0.147	**0.139**	0.158	0.900
CAT	0.114	0.351	0.138	**0.129**	**0.129**	0.900	0.208	0.139	**0.122**	0.125	0.900	0.303	0.132	0.114	0.122	**0.100**	0.225	0.151	**0.141**	0.158	0.900
DEER	0.112	0.336	0.143	**0.128**	0.130	0.900	0.194	0.135	**0.120**	0.122	0.900	0.271	0.133	0.118	0.126	**0.100**	0.209	0.150	**0.147**	0.161	0.900
DOG	0.119	0.370	0.150	**0.136**	0.138	0.900	0.205	0.142	**0.129**	0.132	0.900	0.288	0.135	0.120	0.128	**0.100**	0.229	0.154	**0.147**	0.168	0.900
FROG	0.107	0.228	0.128	**0.117**	**0.117**	0.900	0.155	0.124	**0.113**	0.115	0.900	0.228	0.118	0.110	0.116	**0.100**	0.167	0.137	**0.130**	0.142	0.900
HORSE	0.104	0.251	0.127	**0.114**	0.116	0.900	0.153	0.123	**0.110**	0.112	0.900	0.224	0.116	0.108	0.113	**0.100**	0.178	0.134	**0.129**	0.144	0.900
SHIP	0.112	0.239	0.134	**0.121**	0.126	0.900	0.160	0.131	**0.119**	0.123	0.900	0.236	0.122	0.113	0.120	**0.100**	0.193	0.145	**0.139**	0.159	0.900
TRUCK	0.106	0.210	0.130	**0.121**	0.122	0.900	0.145	0.125	**0.113**	0.117	0.900	0.213	0.121	0.108	0.117	**0.100**	0.165	0.142	**0.134**	0.150	0.900
AVG	0.110	0.281	0.135	**0.123**	0.124	0.900	0.175	0.130	**0.117**	0.120	0.900	0.248	0.125	0.112	0.119	**0.100**	0.195	0.144	**0.137**	0.154	0.900

Table C A.7: Comparison of AUC-PR for one-vs-rest CIFAR-10 (averaged over all images) using logistic regression as classifier. Except for RP_ρ, ρ_1, ρ_0 are given to all methods. Top model scores are in bold with RP_ρ in red if greater than non-RP models. Since $p_{y1} = 0.1$, here *Liu16* always predicts all labels as positive or negative, resulting in a constant AUC-PR of 0.550.

		$\pi_1 = \mathbf{0}$					$\pi_1 = \mathbf{0.25}$					$\pi_1 = \mathbf{0.5}$									
		$\rho_1 = \mathbf{0.5}$					$\rho_1 = \mathbf{0.25}$					$\rho_1 = \mathbf{0}$					$\rho_1 = \mathbf{0.5}$				
IMAGE	TRUE	RP$_\rho$	RP	NAT13	ELK08	LIU16	RP$_\rho$	RP	NAT13	ELK08	LIU16	RP$_\rho$	RP	NAT13	ELK08	LIU16	RP$_\rho$	RP	NAT13	ELK08	LIU16
PLANE	0.288	0.225	0.224	0.225	0.207	**0.550**	0.261	0.235	0.225	0.217	**0.550**	0.285	0.251	0.245	0.248	**0.550**	0.196	0.171	0.171	0.159	**0.550**
AUTO	0.384	0.350	0.317	0.312	0.316	**0.550**	0.342	0.335	0.331	0.331	**0.550**	0.328	0.348	0.334	0.333	**0.550**	0.256	0.257	0.259	0.261	**0.550**
BIRD	0.198	0.160	0.169	0.166	0.161	**0.550**	0.188	0.185	0.179	0.177	**0.550**	0.186	0.173	0.174	0.175	**0.550**	0.150	0.154	0.150	0.147	**0.550**
CAT	0.188	0.164	0.175	0.174	0.175	**0.550**	0.163	0.169	0.168	0.170	**0.550**	0.148	0.156	0.154	0.152	**0.550**	0.145	0.143	0.140	0.145	**0.550**
DEER	0.215	0.161	0.177	0.180	0.183	**0.550**	0.194	0.180	0.180	0.182	**0.550**	0.174	0.175	0.176	0.175	**0.550**	0.151	0.152	0.146	0.151	**0.550**
DOG	0.188	0.162	0.161	0.165	0.155	**0.550**	0.175	0.160	0.161	0.158	**0.550**	0.173	0.169	0.162	0.164	**0.550**	0.145	0.142	0.139	0.133	**0.550**
FROG	0.318	0.246	0.264	0.262	0.258	**0.550**	0.292	0.277	0.272	0.273	**0.550**	0.276	0.274	0.277	0.277	**0.550**	0.239	0.212	0.206	0.212	**0.550**
HORSE	0.319	0.242	0.267	0.269	0.260	**0.550**	0.283	0.264	0.264	0.263	**0.550**	0.288	0.282	0.279	0.278	**0.550**	0.223	0.218	0.208	0.207	**0.550**
SHIP	0.317	0.257	0.267	0.271	0.248	**0.550**	0.296	0.266	0.267	0.259	**0.550**	0.279	0.268	0.259	0.262	**0.550**	0.220	0.212	0.207	0.191	**0.550**
TRUCK	0.329	0.288	0.261	0.271	0.263	**0.550**	0.298	0.275	0.286	0.284	**0.550**	0.289	0.272	0.276	0.277	**0.550**	0.241	0.213	0.208	0.204	**0.550**
AVG	0.274	0.226	0.228	0.229	0.223	**0.550**	0.249	0.235	0.233	0.231	**0.550**	0.243	0.237	0.234	0.234	**0.550**	0.197	0.187	0.183	0.181	**0.550**

Table C A.8: Comparison of error for one-vs-rest CIFAR-10 (averaged over all images) using CNN as classifier. Except for RP_ρ, ρ_1, ρ_0 are given to all methods. Top model scores are in bold with RP_ρ in red if better (smaller) than non-RP models.

		$\pi_1 = \mathbf{0}$					$\pi_1 = \mathbf{0.25}$					$\pi_1 = \mathbf{0.5}$									
		$\rho_1 = \mathbf{0.5}$					$\rho_1 = \mathbf{0.25}$					$\rho_1 = \mathbf{0}$					$\rho_1 = \mathbf{0.5}$				
IMAGE	TRUE	RP$_\rho$	RP	NAT13	ELK08	LIU16	RP$_\rho$	RP	NAT13	ELK08	LIU16	RP$_\rho$	RP	NAT13	ELK08	LIU16	RP$_\rho$	RP	NAT13	ELK08	LIU16
PLANE	0.044	**0.054**	**0.057**	0.059	0.063	0.900	**0.050**	**0.051**	0.054	0.057	0.900	**0.048**	**0.045**	0.049	0.048	0.100	**0.063**	**0.061**	0.074	0.065	0.900
AUTO	0.021	**0.040**	**0.037**	0.041	0.043	0.100	**0.032**	**0.034**	0.040	0.039	0.900	0.028	**0.026**	**0.026**	**0.026**	0.100	**0.047**	**0.049**	0.062	0.070	0.900
BIRD	0.055	0.083	**0.078**	0.080	0.082	0.900	**0.074**	**0.074**	0.077	0.078	0.900	0.072	**0.066**	0.072	0.070	0.100	0.124	**0.084**	0.089	0.093	0.900
CAT	0.077	0.108	**0.091**	0.092	0.095	0.100	0.111	0.090	**0.086**	0.089	0.900	0.113	**0.084**	0.086	0.088	0.100	0.117	0.098	**0.094**	0.100	0.900
DEER	0.049	0.081	**0.078**	**0.078**	0.079	0.900	0.080	**0.069**	0.075	0.070	0.900	0.076	0.062	**0.061**	0.062	0.100	0.106	**0.086**	0.091	0.093	0.900
DOG	0.062	**0.075**	**0.071**	0.079	0.080	0.100	0.071	0.069	0.070	**0.067**	0.900	0.069	0.061	**0.057**	0.076	0.100	0.103	**0.081**	0.084	0.086	0.900
FROG	0.038	0.050	**0.048**	**0.048**	0.054	0.100	**0.047**	**0.052**	0.056	0.062	0.900	0.045	**0.040**	0.042	0.043	0.100	**0.058**	**0.062**	0.066	0.071	0.900
HORSE	0.035	**0.050**	**0.052**	0.057	0.054	0.900	**0.048**	**0.051**	0.052	0.057	0.900	0.045	**0.040**	0.042	0.046	0.100	**0.065**	**0.063**	0.066	0.075	0.900
SHIP	0.028	**0.042**	**0.042**	0.046	**0.042**	0.900	**0.037**	**0.036**	0.042	0.047	0.900	0.035	0.033	**0.031**	0.033	0.100	**0.051**	**0.049**	0.064	0.058	0.900
TRUCK	0.027	**0.044**	**0.046**	0.054	0.056	0.900	**0.034**	**0.032**	0.038	0.043	0.900	**0.034**	**0.031**	0.034	0.034	0.100	**0.060**	0.066	0.067	**0.065**	0.900
AVG	0.043	**0.063**	**0.060**	0.064	0.065	0.580	**0.059**	**0.056**	0.059	0.061	0.900	0.056	**0.049**	0.050	0.053	0.100	0.080	**0.070**	0.076	0.077	0.900

Table C A.9: Comparison of AUC-PR for one-vs-rest CIFAR-10 (averaged over all images) using CNN as classifier. Except for RP_ρ, ρ_1, ρ_0 are given to all methods. Top model scores are in bold with RP_ρ in red if greater than non-RP models.

		$\pi_1 = \mathbf{0}$					$\pi_1 = \mathbf{0.25}$					$\pi_1 = \mathbf{0.5}$									
		$\rho_1 = \mathbf{0.5}$					$\rho_1 = \mathbf{0.25}$					$\rho_1 = \mathbf{0}$					$\rho_1 = \mathbf{0.5}$				
IMAGE	TRUE	RP$_\rho$	RP	NAT13	ELK08	LIU16	RP$_\rho$	RP	NAT13	ELK08	LIU16	RP$_\rho$	RP	NAT13	ELK08	LIU16	RP$_\rho$	RP	NAT13	ELK08	LIU16
PLANE	0.856	0.779	0.780	**0.784**	0.756	0.550	**0.808**	**0.797**	0.770	0.742	0.550	**0.813**	**0.824**	0.792	0.794	0.550	**0.710**	**0.722**	0.662	0.682	0.550
AUTO	0.954	0.874	**0.889**	0.878	0.833	0.550	**0.905**	**0.900**	0.871	0.866	0.550	**0.931**	**0.927**	0.924	0.910	0.550	**0.824**	**0.814**	0.756	0.702	0.550
BIRD	0.761	0.559	0.566	**0.569**	0.568	0.550	**0.619**	**0.618**	0.584	0.597	0.550	**0.623**	**0.679**	0.613	0.619	0.115	0.465	0.492	0.436	0.434	**0.550**
CAT	0.601	0.387	0.447	**0.463**	0.433	0.550	0.423	0.454	0.487	0.480	**0.550**	0.483	**0.512**	0.493	0.473	0.050	0.373	0.375	0.382	0.371	**0.550**
DEER	0.820	**0.620**	0.600	**0.615**	0.573	0.550	0.646	**0.660**	0.610	0.657	0.550	0.658	**0.707**	0.700	0.703	0.550	0.434	0.487	0.414	0.435	**0.550**
DOG	0.758	**0.629**	**0.662**	0.617	0.573	0.550	**0.673**	**0.667**	0.658	0.660	0.550	0.705	0.722	**0.741**	0.705	0.550	0.541	0.545	0.496	0.519	**0.550**
FROG	0.891	0.812	**0.815**	0.812	0.776	0.550	**0.821**	**0.827**	0.808	0.749	0.550	**0.841**	**0.851**	0.828	0.831	0.550	**0.753**	**0.710**	0.691	0.620	0.550
HORSE	0.897	**0.810**	**0.817**	0.799	0.779	0.550	**0.824**	**0.809**	0.801	0.772	0.550	**0.826**	**0.844**	0.818	0.819	0.550	**0.736**	**0.699**	**0.699**	0.600	0.550
SHIP	0.922	**0.870**	0.862	**0.864**	0.853	0.550	**0.889**	**0.885**	0.843	0.848	0.550	0.889	**0.897**	0.891	0.887	0.550	**0.800**	**0.808**	0.767	0.741	0.550
TRUCK	0.929	**0.845**	**0.848**	0.824	0.787	0.550	**0.887**	**0.894**	0.873	0.853	0.550	**0.904**	**0.902**	0.898	0.883	0.550	**0.740**	**0.709**	0.695	0.690	0.550
AVG	0.839	0.719	**0.729**	0.722	0.693	0.550	**0.750**	**0.751**	0.730	0.722	0.550	0.767	**0.787**	0.770	0.762	0.457	**0.637**	**0.636**	0.600	0.579	0.550

Bibliography

David W. Aha, Dennis Kibler, and Marc K. Albert. Instance-based learning algorithms. *Mach. Learn.*, 6(1):37–66, 1991. doi: 10.1007/BF00153759. URL http://dx.doi.org/10.1007/BF00153759.

Giora Alexandron, José A Ruipérez-Valiente, and David E Pritchard. Evidence of MOOC students using multiple accounts to harvest correct answers. *Learning with MOOCs II: A Workshop for Practitioners: New Approaches to Teaching & Learning*, 2015.

Khaled M Alraimi, Hangjung Zo, and Andrew P Ciganek. Understanding the MOOCs continuance: The role of openness and reputation. *Computers & Education*, 80: 28–38, 2015.

John R Anderson, Albert T Corbett, Kenneth R Koedinger, and Ray Pelletier. Cognitive tutors: Lessons learned. *The journal of the learning sciences*, 4(2):167–207, 1995.

Mauro Andreolini, Alessandro Bulgarelli, Michele Colajanni, and Francesca Mazzoni. Honeyspam: Honeypots fighting spam at the source. *SRUTI*, 5:11–11, 2005.

Anelia Angelova, Yaser Abu-Mostafam, and Pietro Perona. Pruning training sets for learning of object categories. In *CVPR*, volume 1, pages 494–501. IEEE, 2005.

Dana Angluin and Philip Laird. Learning from noisy examples. *Machine Learning*, 2 (4):343–370, 1988.

Paul Baecher, Markus Koetter, Thorsten Holz, Maximillian Dornseif, and Felix Freiling. The nepenthes platform: An efficient approach to collect malware. In *Recent Advances in Intrusion Detection*, pages 165–184. Springer, 2006.

Ryan Shaun Baker, Albert T Corbett, and Kenneth R Koedinger. Detecting student misuse of intelligent tutoring systems. In *Intelligent tutoring systems*, pages 531–540. Springer, 2004a.

Ryan Shaun Baker, Albert T Corbett, Kenneth R Koedinger, and Angela Z Wagner. Off-task behavior in the cognitive tutor classroom: when students game the system. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 383–390. ACM, 2004b.

Gilles Blanchard, Gyemin Lee, and Clayton Scott. Semi-supervised novelty detection. *J. Mach. Learn. Res.*, 11:2973–3009, December 2010. ISSN 1532-4435. URL http://dl.acm.org/citation.cfm?id=1756006.1953028.

Avrim Blum and Tom Mitchell. Combining labeled and unlabeled data with co-training. In *11th Conf. on COLT*, pages 92–100, New York, NY, USA, 1998. ACM. doi: 10.1145/279943.279962. URL http://doi.acm.org/10.1145/279943.279962.

Manuel Blum, Robert W. Floyd, Vaughan Pratt, Ronald L. Rivest, and Robert E. Tarjan. Time bounds for selection. *J. Comput. Syst. Sci.*, 7(4):448–461, August 1973. ISSN 0022-0000. doi: 10.1016/S0022-0000(73)80033-9. URL http://dx.doi.org/10.1016/S0022-0000(73)80033-9.

Leo Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, August 1996. ISSN 0885-6125. doi: 10.1023/A:1018054314350. URL http://dx.doi.org/10.1023/A:1018054314350.

Indranil Chakraborty and Georgia Kosmopoulou. Auctions with shill bidding. *Economic Theory*, 24(2):271–287, 2004.

Olivier Chapelle and Vladimir Vapnik. Model selection for support vector machines. In *Proc. of 12th NIPS*, pages 230–236, Cambridge, MA, USA, 1999. URL http://dl.acm.org/citation.cfm?id=3009657.3009690.

Francois Chollet. *Keras CIFAR CNN*, 2016a. URL http://bit.ly/2mVKR3d. bit.ly/2mVKR3d.

Francois Chollet. *Keras MNIST CNN*, 2016b. URL http://bit.ly/2nKiqJv. bit.ly/2nKiqJv.

Gayle Christensen, Andrew Steinmetz, Brandon Alcorn, Amy Bennett, Deirdre Woods, and Ezekiel J Emanuel. The MOOC phenomenon: who takes massive open online courses and why? *Available at SSRN 2350964*, 2013.

Marc Claesen, Frank De Smet, Johan A.K. Suykens, and Bart De Moor. A robust ensemble approach to learn from positive and unlabeled data using SVM base models. *Neurocomputing*, 160:73 – 84, 2015. ISSN 0925-2312. doi: http://dx.doi.org/10.1016/j.neucom.2014.10.081. URL http://www.sciencedirect.com/science/article/pii/S0925231215001174.

Henry Corrigan-Gibbs, Nakull Gupta, Curtis Northcutt, Edward Cutrell, and William Thies. Deterring cheating in online environments. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 22(6):28, 2015a.

Henry Corrigan-Gibbs, Nakull Gupta, Curtis Northcutt, Edward Cutrell, and William Thies. Measuring and maximizing the effectiveness of honor codes in online courses. In *Proceedings of the Second (2015) ACM Conference on Learning@ Scale*, pages 223–228. ACM, 2015b.

Coursera. Coursera. https://www.edx.org/, 2017. Accessed: May 18, 2017.

courseraterms. Coursera terms of use, privacy policy and honor code. https://authentication.coursera.org/auth/auth/normal/tos.php, 2017. Accessed: May 18, 2017.

Jesse Davis and Mark Goadrich. The relationship between precision-recall and ROC curves. In *Proc. of 23rd ICML*, pages 233–240, NYC, NY, USA, 2006. ACM. doi: 10.1145/1143844.1143874. URL http://doi.acm.org/10.1145/1143844.1143874.

Jennifer DeBoer, Andrew D Ho, Glenda S Stump, and Lori Breslow. Changing "course" reconceptualizing educational variables for massive open online courses. *Educational Researcher*, page 0013189X14523038, 2014.

Thomas G Dietterich. Ensemble methods in machine learning. In *Multiple classifier systems*, pages 1–15. Springer, 2000.

Cecilia d'Oliveira, Stephen Carson, Kate James, and Jeff Lazarus. MIT OpenCourseWare: Unlocking knowledge, empowering minds. *Science*, 329(5991):525–526, 2010.

edX. edX. https://www.edx.org/, 2017. Accessed: May 18, 2017.

edxfaq. edX student faq. https://www.edx.org/about/student-faq, 2017. Accessed: May 18, 2017.

edxterms. edX terms of service. https://www.edx.org/edx-terms-service, 2017. Accessed: May 18, 2017.

Anne Eisenberg. Keeping an eye on online test-takers. *New York Times*, 2, 2013.

Charles Elkan and Keith Noto. Learning classifiers from only positive and unlabeled data. In *Proc. of 14th KDD*, pages 213–220, NYC, NY, USA, 2008. ACM. doi: 10.1145/1401890.1401920. URL http://doi.acm.org/10.1145/1401890.1401920.

Susan E Embretson and Steven P Reise. *Item response theory*. Psychology Press, 2013.

Janos Galambos. Bonferroni inequalities. *The Annals of Probability*, pages 577–581, 1977.

A. Gammerman, V. Vovk, and V. Vapnik. Learning by transduction. In *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence*, UAI'98, pages 148–155, San Francisco, CA, USA, 1998. Morgan Kaufmann Publishers Inc. ISBN 1-55860-555-X. URL http://dl.acm.org/citation.cfm?id=2074094.2074112.

Walter R Gilks. *Markov chain monte carlo*. Wiley Online Library, 2005.

John D. Hansen and Justin Reich. Democratizing education? examining access and usage patterns in massive open online courses. *Science*, 350(6265):1245–1248, 2015a. ISSN 0036-8075. doi: 10.1126/science.aab3782. URL http://science.sciencemag.org/content/350/6265/1245.

John D Hansen and Justin Reich. Socioeconomic status and MOOC enrollment: enriching demographic information with external datasets. In *Proceedings of the Fifth International Conference on Learning Analytics and Knowledge*, pages 59–63. ACM, 2015b.

Kathryn Hempstalk, Eibe Frank, and Ian H. Witten. One-class classification by combining density and class probability estimation. In *Proc. of ECML-PKDD*, pages 505–519, Berlin, Heidelberg, 2008. Springer-Verlag. doi: 10.1007/978-3-540-87479-9_51. URL http://dx.doi.org/10.1007/978-3-540-87479-9_51.

Jaak Henno, Hannu Jaakkola, and Jyrki Makela. From learning to e-learning to m-learning to c-learning to...? In *Information and Communication Technology, Electronics and Microelectronics (MIPRO), 2014 37th International Convention on*, pages 616–622. IEEE, 2014.

Andrew Dean Ho, Justin Reich, Sergiy O Nesterko, Daniel Thomas Seaton, Tommy Mullaney, Jim Waldo, and Isaac Chuang. HarvardX and MITx: The first year of open online courses, fall 2012-summer 2013. *Ho, AD, Reich, J., Nesterko, S., Seaton, DT, Mullaney, T., Waldo, J., & Chuang, I.(2014). HarvardX and MITx: The first year of open online courses (HarvardX and MITx Working Paper No. 1)*, 2014.

Andrew Dean Ho, Isaac Chuang, Justin Reich, Cody Austun Coleman, Jacob Whitehill, Curtis G Northcutt, Joseph Jay Williams, John D Hansen, Glenn Lopez, and Rebecca Petersen. Harvardx and MITx: Two years of open online courses fall 2012-summer 2014. *Available at SSRN 2586847*, 2015.

John J Horton, David G Rand, and Richard J Zeckhauser. The online laboratory: Conducting experiments in a real labor market. *Experimental Economics*, 14(3): 399–425, 2011.

Caroline M Hoxby. The economics of online postsecondary education: MOOCs, nonselective education, and highly selective education. Technical report, National Bureau of Economic Research, 2014.

http://scm.mit.edu. MIT MicroMasters in Supply Chain Management. http://scm.mit.edu/program/blended-masters-degree-supply-chain-management, 2017. Accessed: May 18, 2017.

https://openstaxtutor.org/. Open-Stax Tutor. https://openstaxtutor.org/, 2017. Accessed: May 18, 2017.

Jing Jiang. A literature survey on domain adaptation of statistical classifiers. Technical report, University of Illionis Urbana Champaign, 2008. URL http://sifaka.cs.uiuc.edu/jiang4/domainadaptation/survey.

Yasmin B Kafai and Deborah A Fields. Cheating in virtual worlds: Transgressive designs for learning. *On the horizon*, 17(1):12–20, 2009.

Robert J Kauffman and Charles A Wood. The effects of shilling on final bid prices in online auctions. *Electronic commerce research and Applications*, 4(1):21–34, 2005.

Yashu Kauffman and Michael F Young. Digital plagiarism: An experimental study of the effect of instructional goals and copy-and-paste affordance. *Computers & Education*, 83:44–56, 2015.

Sarah Kellogg. Online learning: How to make a MOOC. *Nature*, 499(7458):369–371, 2013.

Carolyn King, Jo-Anne Kelder, Rob Phillips, Fran McInerney, Kathleen Doherty, Justin Walls, Andrew Robinson, and James Vickers. Something for everyone: MOOC design for informing dementia education and research. In *ECEL2013-Proceedings for the 12th European Conference on eLearning: ECEL 2013*, page 191. Academic Conferences Limited, 2013.

Kenneth R Koedinger and Vincent Aleven. Exploring the assistance dilemma in experiments with cognitive tutors. *Educational Psychology Review*, 19(3):239–264, 2007.

Steve Kolowich. Behind the webcam's watchful eye, online proctoring takes hold. *Chronicle of Higher Education*, 2013.

Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. CIFAR-10 (Canadian institute for advanced research), 2017. URL `http://www.cs.toronto.edu/~kriz/cifar.html`.

Rohit Kumar, Carolyn Penstein Rosé, Yi-Chia Wang, Mahesh Joshi, and Allen Robinson. Tutorial dialogue as adaptive collaborative learning support. *Frontiers in artificial intelligence and applications*, 158:383, 2007.

Andrew S Lan, Andrew E Waters, Christoph Studer, and Richard G Baraniuk. Sparse factor analysis for learning and content analytics. *The Journal of Machine Learning Research*, 15(1):1959–2008, 2014.

Nicholas D Lane, Emiliano Miluzzo, Hong Lu, Daniel Peebles, Tanzeem Choudhury, and Andrew T Campbell. A survey of mobile phone sensing. *IEEE Communications*, 48(9), 2010.

Yann LeCun and Corinna Cortes. MNIST handwritten digit database. http://yann.lecun.com/exdb/mnist/, 2010. URL `http://yann.lecun.com/exdb/mnist/`.

Wee Sun Lee and Bing Liu. Learning with positive and unlabeled examples using weighted logistic regression. In *Proc. of 20th ICML*, volume 1, pages 448–455, 12 2003.

Erich Leo Lehmann and George Casella. *Theory of Point Estimation (Springer Texts in Statistics)*. Springer, 1998.

Xuanchong Li, Kai-min Chang, Yueran Yuan, and Alexander Hauptmann. Massive open online proctor: Protecting the credibility of MOOCs certificates. In *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing*, pages 1129–1137. ACM, 2015.

Andy Liaw and Matthew Wiener. Classification and regression by randomForest. *R news*, 2(3):18–22, 2002.

Marcia C Linn, Libby Gerard, Kihyun Ryoo, Kevin McElhaney, Ou Lydia Liu, Anna N Rafferty, et al. Computer-guided inquiry to improve science learning. *Science*, 344(6180):155–156, 2014.

Bing Liu, Yang Dai, Xiaoli Li, Wee Sun Lee, and Philip S. Yu. Building text classifiers using positive and unlabeled examples. In *Proc. of 3rd ICDM*, pages 179–, Washington, DC, USA, 2003. IEEE Computer Society. URL http://dl.acm.org/citation.cfm?id=951949.952139.

Tongliang Liu and Dacheng Tao. Classification with noisy labels by importance reweighting. *IEEE Trans. Pattern Anal. Mach. Intell.*, 38(3):447–461, March 2016. doi: 10.1109/TPAMI.2015.2456899. URL http://dx.doi.org/10.1109/TPAMI.2015.2456899.

Frank M LoSchiavo and Mark A Shatz. The impact of an honor code on cheating in online courses. *MERLOT Journal of Online Learning and Teaching*, 7(2), 2011.

Jenny Mackness, Marion Waite, George Roberts, and Elizabeth Lovegrove. Learning in a small, task–oriented, connectivist MOOC: Pedagogical issues and implications for higher education. *The International Review Of Research In Open And Distributed Learning*, 14(4), 2013.

Larry M. Manevitz and Malik Yousef. One-class SVMs for document classification. *JMLR*, 2:139–154, March 2002. URL http://dl.acm.org/citation.cfm?id=944790.944808.

Robert C Marshall and Leslie M Marx. Bidder collusion. *Journal of Economic Theory*, 133(1):374–402, 2007.

David F Mastin, Jennifer Peszka, and Deborah R Lilly. Online academic integrity. *Teaching of Psychology*, 36(3):174–178, 2009.

James G Mazoue. The MOOC model: Challenging traditional education. *EDUCAUSE*, 2014.

Alexander McAuley, Bonnie Stewart, George Siemens, and Dave Cormier. *The MOOC model for digital practice.* Free Distribution, 2010.

Donald McCabe. Cheating: Why students do it and how we can help them stop. *Guiding students from cheating and plagiarism to honesty and integrity: Strategies for change*, pages 237–246, 2005.

Donald L McCabe and Linda Klebe Trevino. Academic dishonesty: Honor codes and other contextual influences. *Journal of higher education*, pages 522–538, 1993.

Donald L McCabe, Linda Klebe Trevino, and Kenneth D Butterfield. Academic integrity in honor code and non-honor code environments: A qualitative investigation. *Journal of Higher Education*, pages 211–234, 1999.

Donald L McCabe, Kenneth D Butterfield, and Linda K Trevino. *Cheating in college: Why students do it and what educators can do about it.* JHU Press, 2012.

Marcia McNutt. Bricks and MOOCs. *Science*, 342(6157):402–402, 2013.

Aditya Krishna Menon, Xiaoqian Jiang, Shankar Vembu, Charles Elkan, and Lucila Ohno-Machado. Predicting accurate probabilities with a ranking loss. *CoRR*, abs/1206.4661, 2012.

S Ryszard Michalski, G Jaime Carbonell, and M Tom Mitchell. *ML an AI Approach.* Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1986.

Ashara Banu Mohamed, Norbik Bashah Idris, and Bharanidharan Shanmugum. *Trends in Intelligent Robotics, Automation, and Manufacturing*, chapter A Brief Introduction to Intrusion Detection System, pages 263–271. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012. ISBN 978-3-642-35197-6. doi: 10.1007/978-3-642-35197-6_29. URL http://dx.doi.org/10.1007/978-3-642-35197-6_29.

F. Mordelet and J. P. Vert. A bagging SVM to learn from positive and unlabeled examples. *Pattern Recogn. Lett.*, 37:201–209, February 2014. ISSN 0167-8655. doi: 10.1016/j.patrec.2013.06.010. URL http://dx.doi.org/10.1016/j.patrec.2013.06.010.

M. M. Moya, M. W. Koch, and L. D. Hostetler. One-class classifier networks for target recognition applications. *NASA STI/Recon Technical Report N*, 93, 1993.

Nagarajan Natarajan, Inderjit S Dhillon, Pradeep K Ravikumar, and Ambuj Tewari. Learning with noisy labels. In *Adv. in NIPS 26*, pages 1196–1204. Curran Associates, Inc., 2013. URL http://papers.nips.cc/paper/5073-learning-with-noisy-labels.pdf.

David F. Nettleton, Albert Orriols-Puig, and Albert Fornells. A study of the effect of different types of noise on the precision of supervised learning techniques. *Artificial Intelligence Review*, 33(4):275–306, 2010. doi: 10.1007/s10462-010-9156-z. URL http://dx.doi.org/10.1007/s10462-010-9156-z.

David J. Nicol and Debra Macfarlane-Dick. Formative assessment and self-regulated learning: a model and seven principles of good feedback practice. *Studies in Higher Education*, 31(2):199–218, 2006. doi: 10.1080/03075070600572090. URL http://dx.doi.org/10.1080/03075070600572090.

Kamal Nigam and Rayid Ghani. Understanding the behavior of co-training. In *KDD Workshop*, 2000.

Curtis G. Northcutt, Andrew D. Ho, and Isaac L. Chuang. Detecting and preventing 'multiple-account' cheating in massive open online courses. *Computers & Education*, 100:71 – 80, 2016. ISSN 0360-1315. doi: http://dx.doi.org/10.1016/j.compedu.2016.04.008. URL http://www.sciencedirect.com/science/article/pii/S0360131516300896.

David J Palazzo, Young-Jin Lee, Rasil Warnakulasooriya, and David E Pritchard. Patterns, correlates, and reduction of homework copying. *Physical Review Special Topics-Physics Education Research*, 6(1):010104, 2010.

Gabriele Paolacci, Jesse Chandler, and Panagiotis G. Ipeirotis. Running experiments on Amazon Mechanical Turk. *Judgment and Decision Making*, 5(5):411–419, 2010. URL http://EconPapers.repec.org/RePEc:jdm:journl:v:5:y:2010:i:5:p:411-419.

Laura Pappano. The year of the MOOC. *The New York Times*, 2(12):2012, 2012.

Clifton Phua, Damminda Alahakoon, and Vincent Lee. Minority report in fraud detection: classification of skewed data. *Acm sigkdd explorations newsletter*, 6(1):50–59, 2004.

John Platt, Bernhard SchÃűlkopf, John Shawe-Taylor, Alex J. Smola, and Robert C. Williamson. Estimating support of a high dimensional distribution. Technical report, MSR, 1999. URL https://www.microsoft.com/en-us/research/publication/estimating-the-support-of-a-high-dimensional-distribution/.

Diane J Prince, Richard A Fulton, and Thomas W Garsombke. Comparisons of proctored versus non-proctored testing strategies in graduate distance education curriculum. *Journal of College Teaching and Learning*, 6(7):51, 2009.

Niels Provos et al. A virtual honeypot framework. In *USENIX Security Symposium*, volume 173, pages 1–14, 2004.

J. Ross Quinlan. Induction of decision trees. *Machine learning*, 1(1):81–106, 1986.

Deborah A Raines, Peter Ricci, Susan L Brown, Terry Eggenberger, Tobin Hindle, and Mara Schiff. Cheating in online courses: The student definition. *The Journal of Effective Teaching*, 11(2):80–89, 2011.

Yuval Raviv and Nathan Intrator. Bootstrapping with noise: An effective regularization technique. *Connection Science*, 8(3-4):355–372, 1996. doi: 10.1080/095400996116811. URL http://dx.doi.org/10.1080/095400996116811.

Scott E. Reed, Honglak Lee, Dragomir Anguelov, Christian Szegedy, Dumitru Erhan, and Andrew Rabinovich. Training deep neural networks on noisy labels with bootstrapping. In *ICLR*, 2015. URL http://arxiv.org/abs/1412.6596.

Justin Reich. MOOC completion and retention in the context of student intent. *EDUCAUSE Review Online*, 2014.

Kevin W Rockmann and Gregory B Northcraft. To be or not to be trusted: The influence of media richness on defection and deception. *Organizational Behavior and Human Decision Processes*, 107(2):106–122, 2008.

scikit learn. *LogisticRegression Class at scikit-learn*, 2016. URL http://bit.ly/2o3y6r5.

Clayton Scott. A rate of convergence for mixture proportion estimation, with application to learning from noisy labels. *JMLR*, 38:838–846, 2015. ISSN 1532-4435.

Clayton Scott, Gilles Blanchard, and Gregory Handy. Classification with asymmetric label noise: Consistency and maximal denoising. In *COLT*, pages 489–511, 2013. URL http://dblp.uni-trier.de/db/conf/colt/colt2013.html#ScottBH13.

Valerie J. Shute. Focus on formative feedback. *Review of Educational Research*, 78 (1):153–189, 2008. doi: 10.3102/0034654307313795. URL http://dx.doi.org/10.3102/0034654307313795.

Marshall S Smith. Opening education. *Science*, 323(5910):89–93, 2009.

stanfordonline. Harnessing new technologies and methods to advance teaching and learning at Stanford and beyond. http://web.stanford.edu/dept/vpol/vpol-files/2013_Report/Stanford_Online_2013_In_Review.pdf, 2017. Accessed: May 18, 2017.

Carl Straumsheim. MOOCs for (a year's) credit. *Inside Higher Ed*, 2015.

Masashi Sugiyama, Taiji Suzuki, and Takafumi Kanamori. *Density Ratio Estimation in ML*. Cambridge University Press, New York, NY, USA, 1st edition, 2012.

Udacity. Udacity. https://www.edx.org/, 2017. Accessed: May 18, 2017.

udacityterms. Udacity terms of service. https://www.udacity.com/legal/tos, 2017. Accessed: May 18, 2017.

Alex B Van Zant and Laura J Kray. "i can't lie to your face": Minimal face-to-face interaction promotes honesty. *Journal of Experimental Social Psychology*, 55: 234–238, 2014.

A. E. Waters, C. Studer, and R. G. Baraniuk. Bayesian pairwise collaboration detection in educational datasets. In *Global Conference on Signal and Information Processing (GlobalSIP), 2013 IEEE*, pages 989–992, Dec 2013. doi: 10.1109/GlobalSIP.2013.6737059.

Andrew Waters, Christoph Studer, and Richard Baraniuk. Collaboration-type identification in educational datasets. *JEDM-Journal of Educational Data Mining*, 6 (1):28–52, 2014.

George O Wesolowsky. Detecting excessive similarity in answers on multiple choice exams. *Journal of Applied Statistics*, 27(7):909–921, 2000.

James A Wollack. A nominal response model approach for detecting answer copying. *Applied Psychological Measurement*, 21(4):307–320, 1997.

James A Wollack. Detecting answer copying on high-stakes tests. *The Bar Examiner*, 73:35–45, 2004.

KJ Worsley. An improved Bonferroni inequality and applications. *Biometrika*, 69(2): 297–302, 1982.

Tong Xiao, Tian Xia, Yi Yang, Chang Huang, and Xiaogang Wang. Learning from massive noisy labeled data for image classification. In *CVPR*, 2015.

Diyi Yang, Miaomiao Wen, Abhimanu Kumar, Eric P Xing, and Carolyn Penstein Rose. Towards an integration of text and graph clustering methods as a lens for studying social interaction in MOOCs. *The International Review of Research in Open and Distributed Learning*, 15(5), 2014.

Tianbao Yang, Mehrdad Mahdavi, Rong Jin, Lijun Zhang, and Yang Zhou. Multiple kernel learning from noisy labels by stochastic programming. In *Proc. of 29th ICML*, pages 233–240, New York, NY, USA, 2012. ACM. URL http://icml.cc/2012/papers/127.pdf.

Matthew Zook. Your urgent assistance is requested: The intersection of 419 spam and new networks of imagination. *Ethics Place and Environment*, 10(1):65–88, 2007.