Binary Classification from Uncertainty and Triplet Comparison
（不確実性比較と三項比較を用いた二値分類学習）

by

Zhenghang Cui
崔正行

A Doctor Thesis
博士論文

Submitted to
the Graduate School of the University of Tokyo
on December 3rd, 2021
in Partial Fulfillment of the Requirements
for the Degree of Doctor of Information Science and Technology
in Computer Science

Thesis Supervisor: Issei Sato　佐藤一誠
Associate Professor of Computer Science

ABSTRACT

Machine learning has achieved tremendous development and brought innovation to many aspects of the society in the recent decade. Binary classification, one of the core tasks of machine learning, considers learning binary decision functions, usually called *classifiers*, from labeled data. Once learned, classifiers can be deployed in real world applications to accomplish the assigned classification task. However, as the model size and the machine capability has been advancing in a high speed, the training process in the whole classification framework is hunger for a huge amount of *accurate annotation* to be able to provide high performance. Because conducting a real world classification task requires specific knowledge or experience to a certain extent, this is not the case for many applications as recruited annotators lack expertise provide unreliable and inaccurate annotation for a high probability. In this thesis, we focus on the problem of learning well-performing binary classifiers from noisy annotation using alternative forms of feedback.

When conducting decisions, such as an annotator assigning a label to a data point, it has been known for more than three score years that she tends to implicitly conduct some kinds of relative comparison. Moreover, the similar behavior of conducting evaluation based on implicit relative comparison has been observed under various situations. For example, when performing identification tasks such as recognizing the brightness of a color or the tone of a sound, one tends to implicitly compare the current instance with the previously shown instance to give her answer. Motivated by such human actions of carrying out explicit evaluations using comparison information, we focus on the problem of learning binary classifiers from comparison feedback. Although learning from comparison feedback is not new in machine learning, our goal of efficiently learning binary classifications distinguish this thesis from most of existing studies. Specifically, this thesis contributes to the research field in the following two aspects.

Firstly, we extend the possibility of learning binary classifiers from pairwise comparisons alone, without knowing other information of the underlying data distribution. Noise is taken into account due to the innate property of the annotation process. Noisy pairwise comparison feedback has been incorporated to improve the overall query complexity of interactively learning binary classifiers. The *positivity comparison feedback* has been extensively used to provide feedback on which is more likely to be positive in a pair of data points. However, since it is impossible to infer accurate labels using this oracle alone *without knowing the classification threshold*, existing methods still rely on the traditional *explicit labeling feedback*, which explicitly answers the label given a data point. The current method conducts sorting on all data points and use explicit labeling feedback to find the classification threshold. However, it has two drawbacks: (1) it needs unnecessary sorting for label inference; (2) it naively adapts quick sort to noisy feedback. It is desirable to propose an algorithm that can avoid inefficiencies of existing approaches and efficiently acquire information of the classification threshold at the same time. To this end, we propose a new pairwise comparison feedback concerning data uncertainties, which is a common concept in data analysis. To the best of our knowledge, we are the first to propose feedback on comparison uncertainties between data points. This uncertainty comparison feedback answers which one has higher uncertainty given a pair of data points. We then propose an efficient interactive labeling algorithm to take advantage of the proposed comparison feedback. In addition, we also address the situation where the labeling budget is insufficient compared to the dataset size, by using the A2 theoretical active learning framework. Furthermore, we confirm the feasibility of the proposed oracle and the performance of the proposed algorithm theoretically and empirically, using both simulation and user studies.

Secondly, we then move forward to pay attention to a higher meta level of comparing similarities, which leads to the feedback form of triplet comparison. Learning from triplet comparison data has been extensively studied in the context of metric learning, where we want to learn a distance metric between two instances, and ordinal embedding, where we want to learn an embedding in an Euclidean space of the given instances that preserves the comparison order as well as possible. Unlike fully-labeled data, triplet comparison data can be collected in a more accurate and human-friendly way. Although learning from triplet comparison data has been considered in many applications, an important fundamental question of whether we can learn a classifier *only* from triplet comparison

data *without all the labels* has remained untouched. In this thesis, we give a *positive* answer to this important question by proposing an *unbiased estimator* for the *classification risk* under the empirical risk minimization framework, requiring minimum assumptions on the underlying data distribution. Since the proposed method is based on the empirical risk minimization framework, it inherently has the advantage that any surrogate loss function and any model, including highly expressive *neural networks*, can be easily applied. Furthermore, we theoretically establish an estimation error bound for the proposed empirical risk minimizer and provide experimental results to show that our method empirically works well and outperforms various baseline methods using simulation and user studies.

In summary, this thesis focuses on the task of learning binary classifiers when the explicit labeling feedback is unreliable and inaccurate. On incorporating comparison feedback for assistance, we make contribution in two aspects. First, concerning the problem of inefficient learning caused by the information lacking on the underlying data distribution, we propose a new form of comparison feedback and a corresponding interactive algorithm whose feasibility is rigorously evaluated. Second, concerning the impossibility of learning from triplets alone caused by the inadequate modeling, we propose a classification risk rewriting method with freedom on loss functions and model architectures. This thesis makes a concrete step on exploring the possibility of taking advantage of comparison feedback in binary classification.

Acknowledgements

My life of research started from my senior year of bachelor, three years prior to the start of my Ph.D. course. In this total seven years of journey, I stayed in an environment of study and research which did not change too much, so I would like to express my gratitude to all that positively influenced me during these almost $1,500$ days and nights.

First, I would like to thank Prof. Issei Sato, who gives me selfless guidance on research since Practise III lecture, which is even prior to my assignment to the laboratory. I still remember at the beginning I sent Prof. Issei Sato a naively imaginative research plan when I know only few about how to correctly conduct research, he then replied me with patience to tell me every aspect of work I need to do. Along the journey, he always gives warm care on my research life, especially during those time when it went to an dead end and I could not achieve fruitful results. I have also learned the way of reading and digesting academic papers and opening my eyes onto interdisciplinary research from him. Moreover, he shows a striking example on taking a good balance between work and private life. He shows me the importance and the possibility of handling tough and overwhelmed work and keeping good care of families. Additionally, I would like to express my gratitude to Prof. Masashi Sugiyama, who would always be there whenever a student needs any form of help and he can solve any issues with his veteran experience on all aspects. He generously provides me chances of financial supplies during most of my graduate course time. He also teaches me how to think independently and act regarding my own initiatives. Learning from both professors, I am able to know what an ideal researcher should be like. They give me the reason not to be lazy on work using the excuse of being too busy.

Next, I would like to thank laboratory members who spent a lot of time together with me. I would definitely remember the walks and chats with Dr. Nontawat Charoenphakdee, who is always cheering when I have done. I also thank Hongyi Ding and Jongyeong Lee for helping me with many issues. My gratitude also goes to Masahiro Kato, Yivan Zhang, Han Bao and Kento Nozawa. Without the help from the secretary Ms. Yuko Kawashima and Ms. Chie Ogawa, I would not use my budget seamlessly.

In the following, I would like to thank my fiancee, who appears in my life like a magic and supports me along the journey. I also want to say thank you to my parents in Shenyang, who selflessly support me even though we cannot even meet frequently, especially during the pandemic. Moreover, I would like to thank my foster mother and brother in Indiana, who will always have my back when needed. I am also owe to my grandmother, who can barely remember me due to Alzheimer, but without her feeding me I could not grow to be strong and healthy enough to pursue a Ph.D. degree.

Furthermore, I would like to thank members of the breadhouse, a loose circle of friends who meet occasionally to study. They are Hirono Okamoto, Hitoshi Nakanishi, Itto Higuchi, Katsuya Ito and Kaori Hashimoto. I also owe to Hi-

roshi Seno, Rubaiat Hq, Hyungseok Chang, Yoshihiro Kumazawa, Yuki Imai, Yusuke Konno, Yutaka Iso, Takuma Yamashita, Takuma Ishikawa, Saiei Matsubara, Taichi Kiwaki, Kenichiro Nishioka and Sho Tanaka, who I met before my Ph.D. course but received help in various forms during the course.

My research is financially supported by the DC2 program of the Japan Society for the Promotion of Science, the AIP Challenge project and numerous RA programs provided by the department and the university. I also thank the university for supplying dormitories, the owners of apartments I lived during the course and the real estate agent Ms. Nakamura at Hongo 3-chome.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This chapter presents the background and the fundamental motivations of this thesis. We provide a brief summary on our road to machine learning and the more specific problem of learning from more efficient user feedback. Then, we summarize the challenges of existing studies on the focused problem and the contributions of this thesis.

1.1 Historical retrospect

Historical materialism believes that the productivity force is the fundamental driving power of the social and historical development progress throughout the whole human history and also the coming future. Specifically, the productivity force, without mentioning the formal and detailed definition in the literature, simply means how human beings are capable of shaping the nature environment around them based on their own will. The evolution of the productivity force, appearing in the form of groundbreaking technology revolution, brings changes to production relations in the human society and inevitably drives the whole society to move forward. Interestingly, this evolving process appears to keep accelerating and did not show any clue to slow down. Human beings spent millions of years using tools made by stones and living in tribes. After the development and popularization of metal smelting technology, tribe alliance and early kingdoms appeared. Not long after that, better metal tools for cultivation largely increased population for settlement and further promoted the human society to evolve into classical empires.

The most profound change is the industrial revolution, which happened thousands of years after the previous agricultural revolution and deeply changed the progress of history and way of living of almost all humans on this planet. For the first time, human beings can create and build things, such as from skyscrapers and huge dams to missiles and horrible large-scale lethal weapons, with less restrictions by the nature. Consequently, driven by the need of massive computation for such creation process, the modern computer as a powerful tool for achieving a higher level of productivity force is invented after World War II. Following up research on designing better computers and using them more efficiently gives name to the department which this thesis is submitted to.

Then, with the advent of the Internet in 1980s, the productivity force is acceleratively evolving into a new era and almost all aspects of society is facing another shift, usually called the information revolution. With the popularization of the Internet and computers getting smaller and smaller, the remarkable and distinct phenomenon is the accumulation of data, sometimes called the Big Data, that are collected through various on-line and off-line activities. This has the

potential to further change almost every aspect of the society. For example, the way of conducting scientific research can be profoundly changed into a totally data-driven discovery paradigm, usually called the forth paradigm. This exciting and passionate time of history is urging for new theories, new methods and new tools on handling this endlessly emerging large scale data.

There are many aspects to explore and improve on handling such data, such as storage, communication and analysis, etc. This thesis falls into the last aspect, namely the analysis of data to provide insights or tools that achieve a higher level of productivity force and can relieve the burden of human labors.

For broader audience, the methods proposed by this thesis can be usually referred to as artificial intelligence (AI), which is not a new word which first appeared in 1956. However, AI is generally an umbrella word that can have different meanings and can be used in a misleading way. In fact, AI can also point to different technologies in different periods of time. For example, the already matured technology of optical character recognition (OCR) may no longer be called AI in nowadays narrative, which usually refers to fantastic home assistant technologies that can communicate with humans in natural language to some extent. Therefore, we specifically call the field of this thesis as machine learning, which is more narrow than the scope of AI and has a relatively clearer meaning in the context of a scientific discussion.

1.2 Machine learning

Machine learning tries to fulfill the innate curiosity of creating a device that can undertake any task that requires a human being. It roots at the deepest dream of human beings to fully understand ourselves and thus making machines that can copy the intelligence of ourselves. Specifically, as indicated by the name, machine learning focuses on the intriguing and mysterious process of learning, and aims to answer the following questions [108]:

> How can we build computer systems that automatically improve with experience, and what are the fundamental laws that govern all learning processes?

More specifically, machine learning falls into the intersection of statistics and computer science. It also have applications in many other fields of science, such as social science, health care and crime detaining by just naming a few. Disciplines of statistics help researchers in this field to discover and summarize patterns from data, without explicitly writing down every piece of rules for a specific task. Knowledge from computer science helps to actually implements algorithm and to empirically observe how it may perform in real-world situations. Similar to the metaphor of research on physics being a man walking forward by two legs: one leg of experimental physics and the other leg of theoretical physics, statistics and computer science can also be seen as two legs that move machine learning forward. The metaphor of legs is important and intriguing in the sense that one leg may be ahead of the other during some time, and will be lagged behind during another time. However, both legs are important and the field cannot move forward without either one of them.

As it is not a rare phenomenon like planes are invented being inspired from birds but they work in different mechanisms; although being inspired from human learning process, the machine learning approach is different in many aspects:

- Speed: Machine learning is fast and can be further accelerated by improved hardware while human learning is slow and has a nature upper bound for the speed. For example, it would take only few hours for a modern deep learning model to master taxonomy classification at a satisfying accuracy, but may take a human being nearly thirty years to become a specialist in the specific knowledge domain, during which writing a thesis is inevitable to take several months but can also be accomplished by a natural language generative model in a blink.

- Domain Knowledge: Machine learning solves a task without relying on the domain knowledge about it, but only the data collected for the task or results of interactions with the environment. For example, when dealing with classification tasks, machine learning algorithms are not designed to explicitly differentiate the task as it is for dog species or impact craters on the moon. Machine learning algorithms are executed in a totally domain-invariant favor, which is sharply different from the human learning process, which usually takes a coarse-to-fine procedure that first familiarizes one with the background knowledge and then introduces the specific professional knowledge.

- Reproducibility: Machine learning can be easily reproduced as it is nothing but essentially a computation procedure. Although recent methods are criticized for lacking reproducibility, it is mainly due to the bad practise of paper writing, not a fundamental disadvantage of machine learning itself. As long as the corresponding experimental settings are clearly stated, the machine learning procedure as well as its resulting models are totally reproducible, and computer software for reproducibility such as Docker is under activate development and deployment. In contrary, no two human beings will show identical ability for learning and one also has her own preference over different disciplines of knowledge. It is also widely agreed that having their own fitted curriculum design, which shows none reproducibility at all, would benefit children education.

- Adaptation: It would take no additional efforts for machine learning algorithms to adapt to a new problem setting, other than rerun the training procedure. Moreover, this usually does not take too much energy and can even be scheduled as a repeating task. This is due to not taking into account the background domain knowledge into algorithm design. However, it is extremely time consuming for a human being to become an expert in more than two fields. Because one usually needs to start over from the basics before entering a new science field. As the population holding one Ph.D. degree is already few, fewer would choose to obtain a second or even a third Ph.D. degree. We can say population with more than one Ph.D. degree is much smaller than the number of models that can be trained to handle more than one dataset.

- Improvement applicability: One improvement over a machine learning algorithm naturally applies to all tasks it is solving. However, when a human being is working on several different tasks, an improvement made on one of the tasks does not instantly apply for other tasks.

Based on the above advantages, machine learning is indeed an improvement on the productivity force, and will push the society to move forward in many aspects

to a certain extent. Optimists even claim that with enough data, large enough model and powerful enough computational resource, every scientific problem can be solved by machine learning using deep neural networks. Bold names such as *foundation model* are given for such large scale models, with the hope that they can form the new foundation of next generation science. Being cautiously optimistic, we hope to see those algorithms to be applied in real-world to make benefit soon.

In general, we consider machine learning to be categorized into the following two types of algorithms:

- The agent model: This is the model that is going to be deployed for decision making once properly configured. For example, this would be a classifier for classification tasks or a scorer for regression tasks.

- The training / learning algorithm: This is the algorithm that adjusts the configuration of the agent model. Execution of this algorithm adjusts the agent model configurations, usually parameters, to perform better at the designated task, thus performing a process of *training* from the outer algorithm point of view and a process of *learning* from the agent model point of view. Sometimes it runs at the same time with the deployment of the agent model.

Depending on the definition of a specific task, we choose to assign machine learning methods into two large categories, which is slightly different from most of the literature.

- Problem setting with no interaction with the environment: In this simple setting, all information needed for the input to the training algorithm is collected before the start of its execution. At the same time, no further interaction with the environment is allowed during the execution of the training algorithm. Therefore, the training algorithm only needs to concentrate on how to efficiently use the available data at hand.

- Problem setting with interaction with the environment: In this setting, the training algorithm needs to be designed in the way that can not only elicit useful information from data available during its execution, but can also decide when and what to obtain useful information by as few as possible interaction with the outer environment. The interaction usually charges a cost, thus containing its number is usually a part of the goal of the training algorithm in this case.

We believe this categorization offers a more clear view on the overall field and thus provide better guidelines for developing new methods. We briefly introduce methods based on the two categories in the following, as both are related to the contributions of this thesis.

1.2.1 Problem setting with no interaction with the environment

In this section, we categorize learning scenarios depending on the existence of supervision signals.

- Unsupervised learning: In this setting, only data features are given as

$$\{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n\},$$

where $\mathbf{x}_i \in \mathbb{R}^d$ are features in the vector form and n is the size of the data set. The goal is to find interesting or useful patterns, or learn useful latent features of the given data. Typical examples of unsupervised learning tasks are clustering, latent factor analysis, representation learning, etc.

- Supervised learning: In this setting, a supervision signal is given along with the features of a data point, or a group of data points. The goal is to learn a function which takes a data point as input and the desired supervision signal as output. Depending on the type of the supervision signal, the task is divided into classification when working with finite discrete signals, and regression with other signals. With the same type of the supervision signal, different tasks can be defined by different goals, such as ranking or ordinal classification where discrete classes have a latent order. Furthermore, different scales of data points given per one supervision signal can also divide the task into different finer problem settings, such as learning with group labels, etc. In this thesis, we focus on the standard setting that there is one corresponding supervision signal for each data point and the signal comes from a discrete set consisting only two elements. Formally, the input data are given as
$$\{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \cdots, (\mathbf{x}_n, y_n)\},$$
where $y_i \in \{-1, +1\}$. This *binary classification* task plays an important role in many applications as well as theoretical analysis in machine learning.

1.2.2 Problem setting with interaction with the environment

In this section, we categorize learning scenarios depending on the modeling methods of the surrounding environment.

- Online learning: In this setting, the interaction is an one-way communication from the environment to the machine learning algorithm. This means that although the algorithm knows there will be new data appear, it cannot apply specific preference on appearing data but can only passively receive it. The fundamental difference from supervised learning is that the algorithm needs to produce feasible models at every step of the execution, and the overall performance is usually used for evaluation.

- Active learning: In this setting, the algorithm is allowed to tell the environment what kind of information it requires, and receives the desired answer. Based on the way of information query, active learning can be further divided into: pool-based active learning, where a totally unlabeled data pool is given and the algorithm is supposed to pick data points of interest for label querying; stream-based active learning, where data points arrive in an online fashion and the algorithm is supposed to choose whether to ask the label of the current data point or not; membership query synthesis: where the algorithm is supposed to synthesis or generate data points for querying the environment instead of rely on existing data points. The pool-based active learning takes a part in the contributions of this thesis.

- Reinforcement learning: This setting considers a more complex agent model. Different from previous settings which only take data features as input, the agent model in this setting is supposed to be capable to take actions, given the information of the surrounding environment and probably the action history taken so far. This offers a formulation for more complicated application scenes, such as industrial robots or self-driving cars.

1.3 Machine learning in practise

Focusing on the binary classification problem setting previously mentioned, we would like to introduce limitations caused by practical applications in this section, in order to provide motivations for the contributions of this thesis.

The success of machine learning algorithms, especially those on press, are mainly built on the availability of large scale and well curated data collected by large companies or organizations with sufficient financial and human resource power. However, this is not always the case for all application scenes, where data are at least partially deficit and limited. We will elaborate on two possible such cases: limitation on *quantity* and *quality*.

- Limitation on quantity: This happens when collecting raw data features appears to be hard. This can have two reasons: the data are few at the first place, such as computed tomography (CT) images of a special decease that only happens on a small population; or there are plenty of data but the effort to collect is financially intensive. The later scenario appears less frequent with the development of more alternative open-source datasets and the popularization of crowdsourcing platforms.

- Limitation on quality: In this case, the raw data are plenty and not much effort need to make to collect them. For example, collecting image data or articles from the Internet is easy as crawling can be conducted by a simple programming script and a feasible hardware. However, when using collected data for tasks requiring supervision signals, such as binary classification, much effort needs to be contributed to collect *massive correct labels* for raw data. The task itself is not infeasible providing enough time and budget, which is nevertheless better to avoid for saving precious resources and compromise needs to be taken.

This thesis focuses on the latter case of limitation on annotation quality. Depending on where the compromise to take, this limitation can be further categorized into two types: accuracy and the form of supervision signals.

- Compromise on accuracy means assuming the accuracy of collect labels are corrupted and design learning algorithms with this assumption in mind, while maintaining the same form of collecting explicit labels for each data point.

- Compromise on the form of supervision signals means instead of collecting corresponding explicit labels for each data point, we make efforts on designing new forms of supervision signals, or user annotation feedback, that enjoys both advantages of easy to annotate thus cheaper to collect, and consistent to annotation noise. Consequently, the available training algorithm designed for traditional the signal form cannot be directly used. This thesis follows this request and proposes algorithms that can work with alternative forms of feedback and can still return a binary classifier as if it is learned using the traditional form of feedback.

1.3.1 Comparison feedback

In existing studies of using alternative forms of feedback for learning binary classifiers, comparison feedback is a unique option that draws attention. Psychology

studies indicate that people tend to make decisions by making implicit unconsciousness comparisons [142], thus provides innate motivation for this thesis. This can be explained by the hypothesis that people are more capable of comparison than explicit evaluation and the results are more stable and correct in some cases. Note that comparison feedback is also important on its own behalf, such that there are machine learning tasks that purely relies on comparison feedback, such as ranking and similarity learning. Therefore, this thesis focuses on the topic of learning binary classifiers from comparison feedback, and makes contributions to be described in detail in the following section.

Specifically, this thesis is devoted to study how to efficiently use comparison feedback to accomplish binary classification. For a binary classification problem, the typical procedure would be

1. Collect unlabeled data points.

2. Query for labels using *explicit labeling feedback* from sources such as user annotations.

3. Select a classification algorithm and conduct it on collected data points and corresponding labels formulated as $\{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \cdots, (\mathbf{x}_n, y_n)\}$.

In this thesis, we focus on the last two steps, which is illustrated in Figure 1.1[1] and Figure 1.2 for later comparison. Specifically, we work on two forms of comparison feedback: the uncertainty comparison feedback, which is proposed by this thesis; and the triplet comparison feedback, which has mainly been used in machine learning fields other than classification.

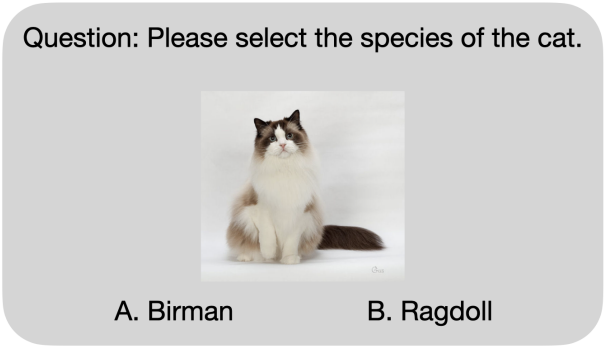


Figure 1.1: Schematic user interface of explicit labeling query.

We simply investigate the accuracies of user feedback itself under the following setting [2]. We asked 100 annotations on 10 questions of explicit labeling and 10 questions on a typical form of comparison feedback [3] on the cat species dataset. From the results shown in Figure 1.3, we can see there is a significant accuracy gap between the two forms of feedback. Thus, we can conclude that there is much space of taking advantage of comparison feedback to assist explicit labeling for learning better classifiers in such cases.

In machine learning, comparison feedback in general has already been used in many methods and it is necessary to note the difference of how this thesis uses them.

[1] Description of the dataset used in queries are deferred to Chapter 3.
[2] Detailed user study settings can be found in the user study part of Chapter 3
[3] We used the positivity comparison which is to be explained in detail in Chapter 3

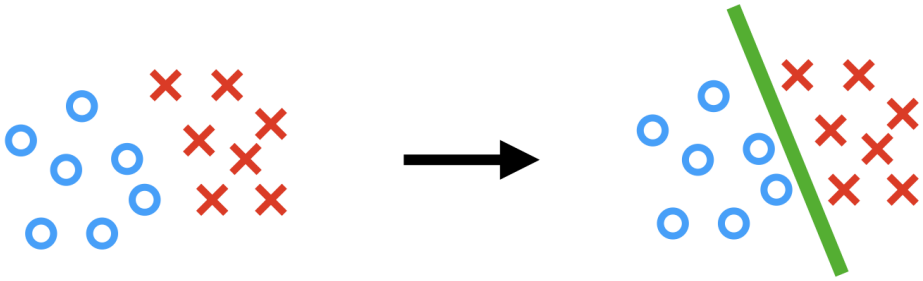


Figure 1.2: Illustration for conducting ordinary binary classification.

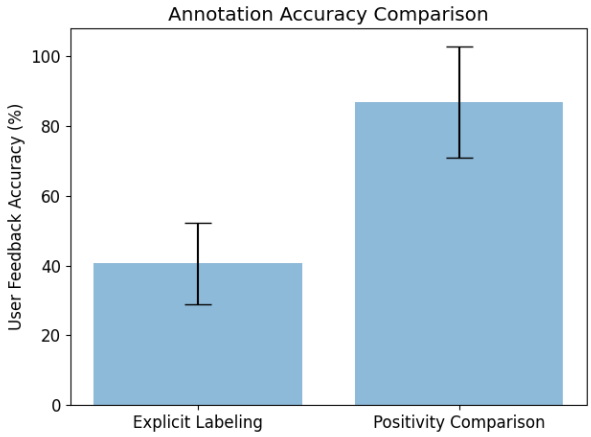


Figure 1.3: Annotation accuracy comparison for explicit labeling and comparison feedback.

- This thesis focuses on the problem of learning classifiers. This draws sharp difference from fields such as metric learning, representation learning and ordinal embedding. In those fields, a distance function or an embedding function is learned from comparison feedback. Classifiers cannot be directly achieved unless further supervision information and learning procedure is conducted in the downstream.
- This thesis considers the interactive / active learning framework where query complexity is expected to be smaller than the passive case. This draws difference from most weakly supervised learning methods where only the passive case is considered.
- This thesis focuses on using comparison feedback alone to learn, without relying on the explicit labeling feedback which is considered to be unreliable and inaccurate in our problem setting. This draws difference from methods even partially relying on explicit labels such as learning from noisy labels.

Before moving on, we would like to note the difference between contributions on problem settings, such as this thesis, and model architectures, such as popular studies proposing a more powerful neural network. Although both types of contributions are important for the field of machine learning to move forward, they are orthogonal to each other. This can be explained by that both designing new off-road vehicles that can run on a situation that no car can do before, and

designing new car engines or new materials for tyres are important for the car industry. For this reason, empirical evaluations in this thesis are not conducted using state-of-the-art model architectures but rather classical neural networks and even the traditional k-nearest neighbour algorithm.

1.4 Summary of contributions

Concretely, this thesis focusing on exploring the possibility of using comparison feedback as assistance in binary classification when the default explicit labeling feedback is found to be unreliable and inaccurate. Towards this direction, we move forward two steps concerning different properties of comparison feedback.

1.4.1 Chapter 3: Learning from uncertainty and positivity comparisons

In this chapter, our objective is to accomplish the task in the two step shown above, assigning labels to unlabeled data points, when only comparison feedback is appropriate instead of the explicit labeling feedback which is assumed to be inaccurate or not robust enough to noise. More specifically, we would like to design label assigning algorithms using comparison feedback to have a query complexity as low as possible.

In order to achieve this goal, we turn our attention to the fundamental properties of a desired form of comparison feedback. It should be weak enough, i.e., easier for a user to answer than explicit labels, and strong enough, i.e., label information can be elicited from the answers. By considering such trade-off, we resort to propose a new form of comparison feedback, which compares the *uncertainties* of data points as shown in Figure 1.4. Uncertainty is an delicate property for data and is difficult to quantify properly, but would be easier for relative comparisons. Combined with another type of pairwise comparison, the positivity comparison, as shown in Figure 1.5, the process using is illustrated in Figure 1.6.

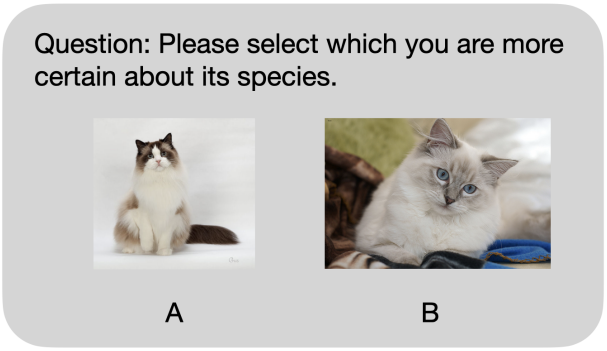


Figure 1.4: Schematic user interface of our proposed uncertainty comparison query.

Consequently, we justify our proposal in two directions. First, we provide a feasible label assigning algorithm that works on the proposed feedback form with empirical justification using simulated data. Then, we conduct extensive user studies to provide justification of robustness and user preference on the proposed feedback.

In summary, for the problem of interactive binary label assigning with access to *pairwise comparison feedback*, our contributions are four-fold in this chapter:

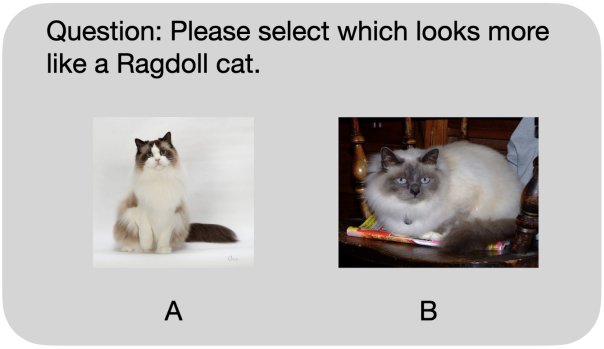


Figure 1.5: Schematic user interface of positivity comparison query.

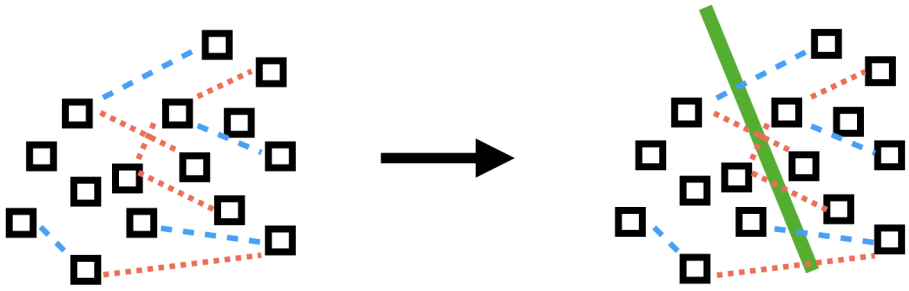


Figure 1.6: Illustration for binary classification from pairwise comparisons. Note that two different types of lines indicate two different types of pairwise comparisons.

- We propose a novel pairwise comparison oracle that compares *uncertainties* of two unlabeled data points.
- We propose an efficient and robust labeling algorithm accessing the aforementioned two kinds of pairwise comparison oracles. We also develop its active version under insufficient query budget.
- We establish the error rate bound for the proposed algorithm and generalization error bounds for its applications, and confirm their empirical performance using both simulation.
- We design and conduct user studies to illustrate the performance superiority of the proposed algorithm against existing methods.

1.4.2 Chapter 4: Learning from triplet comparisons

In this chapter, we focus on the last two steps of the binary classification procedure shown above. First, we would like to question the fundamental reaction when a user is asked to assign a label. When given a test query image, a user implicitly compare it with category prototypes in her mind to achieve category similarities. Then, by comparing these category similarities, the user chooses an answer with the highest one, which is shown in Figure 1.7. Therefore, by substituting implicit category prototypes with actual images, which can be different for each query image, we recover the *triplet comparison feedback*, which is used

in other fields had has not been looked at through such lens. The process is illustrated in Figure 1.8.

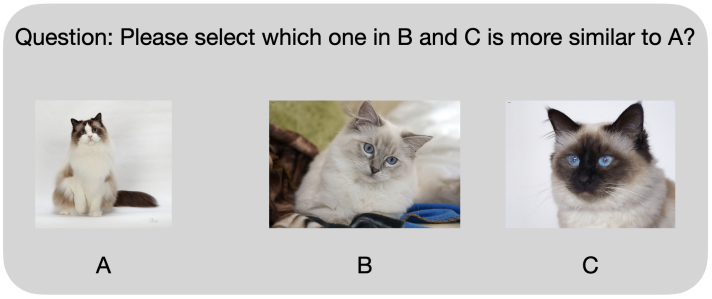


Figure 1.7: Schematic user interface of triplet comparison query.

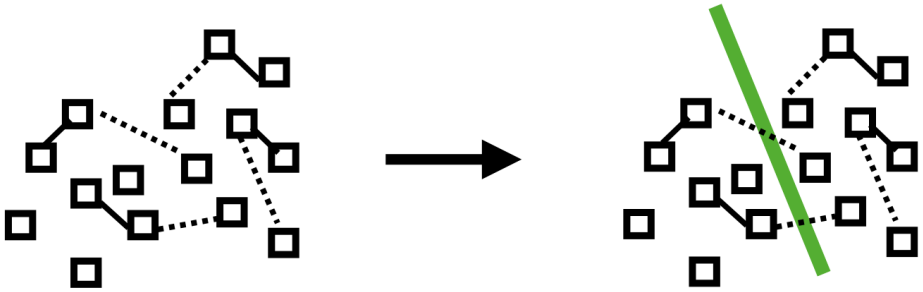


Figure 1.8: Illustration for binary classification from triplet comparisons. Note that a triplet is indicated by three unlabeled data points connected by two different types of lines.

Consequently, we justify using this feedback in binary classification by user studies to examine its robustness. Then, we build a feasible learning algorithm for any differentiable models with theoretical and empirical justification. Note that we consider the passively collected feedback in this chapter, which is different from previous chapter where the algorithm can actively choose the desired pairwise comparison feedback.

In summary, for the problem of binary classification from *triplet comparisons*, our contributions in this chapter are three-fold:

1. We propose an empirical risk minimization method for binary classification using *only* passively obtained triplet comparison data without relying on explicit labels, which gives us an inductive classifier.

2. We theoretically establish an estimation error bound for our method, showing that the learning is consistent.

3. We experimentally demonstrate the practical usefulness of our method.

1.5 Organization

This thesis consists of five chapters. A flow chart is shown in Figure 1.9 for the recommended reading order. Chapter 3 and Chapter 4 are mutually independent contributions and can be read without reading the other chapter in advance.

- In Chapter 2, we presents notations, preliminaries and related work of various machine learning fields that are closely related to the problem studied in this thesis.
- In Chapter 3, we presents the proposal of the uncertainty comparison feedback with justification by user studies and its corresponding application in passive and active binary classification and theoretical analysis.
- In Chapter 4, we presents the application of the triplet comparison feedback in binary classification and corresponding theoretical and empirical justification.
- In Chapter 5, we presents proofs for lemmas, theories and corollaries stated in Chapter 3 and Chapter 4. For self-containment, the main statement of each lemma, theory or corollary is presented again before its corresponding proof.
- In Chapter 6, we conclude this thesis and point out several potential directions for further work extending this thesis.

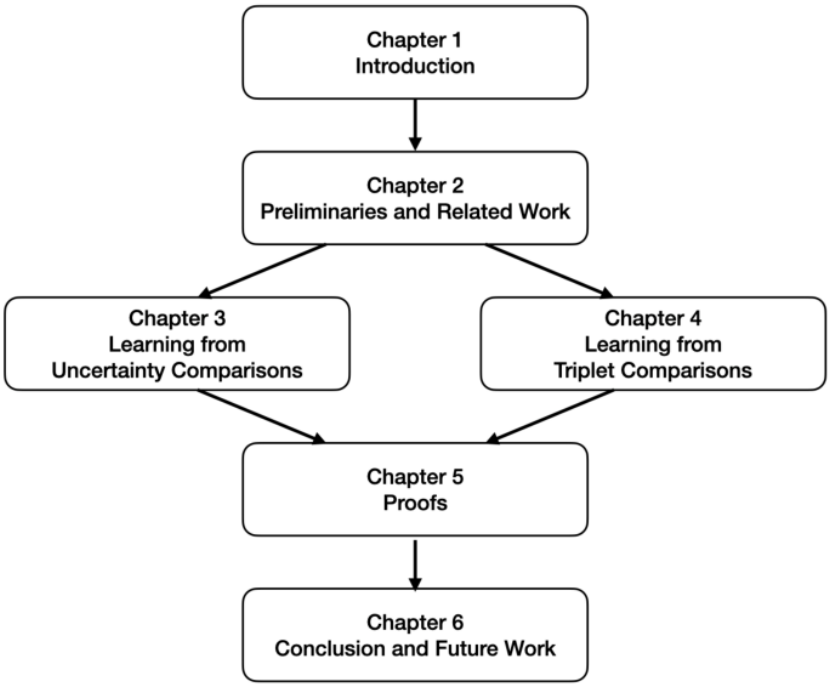


Figure 1.9: Organization of this thesis. Then contributions of this thesis are mainly presented in Chapter 3 and Chapter 4.

Chapter 2

Preliminaries and Related Work

In this chapter, we provide necessary review on related machine learning fields to provide an adequate and clear prospective on the position of our work in the context of existing studies. We first review important studies on social science for a substantial motivation for this thesis, which corresponds to Section 2.2. Then, we elaborate on main topics of interest, namely machine learning methods aiming to learn from comparisons. Specifically, we focus on three sub-fields that are most closely related to our focus: metric learning, contrastive representation learning, weakly-supervised learning, learning to rank and theoretical active learning in Section 2.3, Section 2.4, Section 2.5, Section 2.6 and Section 2.7, respectively. At the end of each section, a brief summary is dedicated to discuss the detailed relation between the section and the thesis proposal. Although covering some research literature on psychology, we would also state concepts from a machine learning perspective, which provides a tighter connection to the rest of the thesis.

2.1 Notations

We list mathematical notations that are used throughout this thesis in the following Table 2.1.

We consider the binary classification problem. Let $\mathcal{X} \subset \mathbb{R}^d$ denote the d-dimensional sample space and $\mathcal{Y} = \{+1, -1\}$ denote the binary label space. Let $\mathcal{P}_{\mathcal{X}\mathcal{Y}}$ denote the underlying data distribution over $\mathcal{X} \times \mathcal{Y}$. Then $h^* \triangleq \text{sign}(\eta(x) - 0.5)$ is the Bayes classifier minimizing the classification risk $R(f) \triangleq \mathbb{E}_{(x,y)\sim\mathcal{P}_{\mathcal{X}\mathcal{Y}}}[\mathbb{1}_{f(x)\neq y}]$ for a classifier $f : \mathcal{X} \to \mathcal{Y}$. In this problem setting, we are given unlabeled data points drawn from $\mathcal{P}_{\mathcal{X}}$, the marginal distribution over $\mathcal{X}$ [1].

2.2 A social psychological perspective on comparison

> *"Walking among three people, I find my teacher among them. I choose that which is good in them and follow it, and that which is bad and change it."*
>
> *Confucius, 551 BC*

In this section, we first review the latent psychological mechanism that drives people to voluntarily compare themselves with others [51], namely the action of *social comparison*. Then, we review the literature that reveals that people are actually good at comparison, rather than conducting direct judgement.

[1] We assume $\mathcal{P}_{\mathcal{X}}$ is a non-degenerate distribution, whose support contains at least two points of $\mathbb{R}^d$. Otherwise, all data points would have the same conditional probability and we can obtain a trivial classifier by a constant function.

Table 2.1: Notations.

Notation	Description
$\mathbb{R}$	The set of real numbers.
$\mathbb{R}^d$	The set of d-dimensional vectors of real numbers.
$\mathcal{X}$	Input space.
$\mathcal{Y}$	Output space.
$\mathcal{S}$	The set of similar pairwise comparison data.
$\mathcal{D}$	The set of dissimilar pairwise comparison data.
$\mathcal{T}$	The set of triplet comparison data.
$\mathbf{x}, \mathbf{y}$	Vectors.
$\mathbf{M}, \mathbf{S}$	Matrices.

2.2.1 Innate motivation and a psychological theory

The seminal work of Festinger [51] studies self-evaluation through comparisons against people in the neighborhood for the first time, and provides a theoretical analysis from the psychological perspective. This theory consists of nine hypothesis and several corollaries, which we will not elaborate on but provide a precise summary as follows.

The theory first states clearly there is an innate drive for one to evaluate her behavior (as opinions or abilities). There are two ways to achieve this: *physical evaluation* which conducts *direct measurement* and *social comparison* which provide only *indirect feedback through comparisons*. Readers with knowledge on machine learning can intuitively draw an analogy to classification with full or weak supervision. When possible, physical evaluation is preferred by people, which is the same as full supervision is essentially the most prioritized paradigm for learning. This psychological corollary is reported to be examined by an experiments on the behaviour of decision making [69]. Those who believes they are capable of making correct decisions are less likely to change their decision when discovering others opinions, especially disagreements.

Next, attention is paid to individuals or groups that are being compared in the process. The theory states that similarity to the decision maker to some extent is essential for the stability of evaluations. Although examined by psychological experiments, this statement raises difficulty on drawing analogy to machine learning. It is known for long that *negative sampling* is necessary for stabilizing representation learning in Natural Language Processing (NLP) [106]. However, representation learning methods that only take advantage of similar data pairs [58] are recently proposed and drawing great attention in both academic and industries. It is an interesting phenomenon to observe distinctly different fields reaching a similar conclusion, suggesting the essence of science.

Other theoretical statements are mainly about group divergence and how it will change under different circumstances. Among these statements, one notable hypothesis assumes that there is always a drive to improve one's ability, where the strength of this drive is dependent on one's cultural background. Consequently, we can conclude that an eternal urge of comparing with the best of others exists for people with strong drives, especially for Ph.D. students usually accompanied with high self-esteem. This suggests one reason for peer pressure and may explains the astonishing phenomenon that almost one in every two Ph.D. students experience psychological distress [96].

2.2.2 Categories of social comparisons

Depending on the difference between the individual and the compared group, social comparisons can be classified into *upward comparisons* and *downward comparisons* [152]. Almost self-evident from the terminologies, the former type corresponds to comparing against those with higher ability on some attributes and the later means the opposite. Ideally and naturally, it is necessary for one to conduct both types of social comparisons to realize a balanced self-evaluation and maintain reasonable and healthy self-esteem. It is also subjectively flexible to choose when and which type of social comparison to conduct actively or passively, regarding the psychological situation at the moment. Interestingly, this indirectly incurs motivation for using triplet comparison data, consists of triplets of an anchor, a upward example and a downward example, to conduct machine learning tasks including our contribution to be elaborated in Chapter 4. Studies on downward comparison also leads to the subfield concerning self-enhancement, which has less connection to machine learning and will not be discussed. Psychological models are also developed and examined in order to further clarify the hidden principle of the mental mechanism of social comparison.

2.2.3 Questionable results on absolute judgements

In the following, we discuss psychological studies on evaluating the ability to conduct direct or comparative judgement. Generally, studies report that people are much more good at conducting comparative / relative judgements then absolute / direct judgements. This forms the other part of the motivation of this thesis.

We would like to first clarify common settings for psychological experiments. In general, subjects of the experiment receive stimuli of sound tones, tastes, smells, visual objects such as lines and areas, colors or other cutaneous stimulation [142]. These types of simple uni-dimensional stimuli is commonly believed to be appropriate to test psychological hypothesis. Subjects are usually asked to compare one or more attributes or the stimulus received. For example, they may be asked to first listen to different sound tone clips with different frequencies, and then conduct judgements on listened sound clips or new ones [107].

It is a commonly agreed hypothesis that one has an upper bound limiting the ability of absolute judgement. In tasks of identifying sound tones [124], sound loudness [55] and taste intensities [17], the experimental results all show consistency with the hypothesis. The same statement holds when experimental settings vary on the range of the stimulus attribute. Although the aforementioned experiments focus on uni-dimensional stimuli, it is shown to have similar observations for cases of multi-dimensional stimuli [86].

Additionally, the bow effect is also worthy to note, suggesting the instability of absolute judgements. This describes when plotting accuracies of absolute judgements, extremities with the smallest or the largest stimuli usually have higher accuracy than other stimuli in the middle [78, 93]. A detailed table of similar experimental results is omitted due to the lack of relevance to the thesis, and can be found in the reference [142].

2.2.4 Relation to this thesis

In summary, we can conclude that people are tend to conduct relative comparison than absolute judgement even for absolute evaluation, while the results are not satisfactory and stable when conducting absolute judgement. Moreover, absolute judgements are sometimes indirectly conducted by actually performing relative

comparisons. This motivates us to study from a machine learning perspective how a absolute judgement model can be learnt from relative comparisons.

2.3 Metric learning

In this section, we review the field of metric learning [89, 19]. Generally, metric learning is used with input data which have various forms of relative comparisons, and the goal is to learn a latent metric that can measure in the sense of relative relationship given the input data.

2.3.1 Preliminaries

The motivation of metric learning is to learn a task-specific distance function, or dissimilarity function, that can provide useful information when used instead of the general Euclidean distance function in a simply defined feature space. Face image recognition is usually considered as a motivated example, which requires different definitions of similarity on different tasks. When the task is to recognize individual faces, one should focus on features such as hair style, hair color or face shape. However, when the task is to recognize facial expressions, one should focus on features such as eyebrow angle, mouth shape and cheek status. Therefore, it is important to have a distance function that is specifically tailored for a given task to achieve optimal performance. Lead by the spirit of machine learning, we would *learn* the desired function from collected comparison data instead of hand-crafting the detailed form of the distance function itself.

In pragmatic situations, metric learning is innately intimate to algorithms that critically rely on such a function, such as nearest neighbor methods, information retrieval methods or human verification and identification methods, which can be applied as a downstream task after metric learning. Notably, the development of metric learning can be seen as a typical example of that of a machine learning subfield, which starts from a strong practical motivation and intuitive methods based on linear transformation, then evolves to nonlinear methods using tools such as kernel functions, and finally arrives at the deep learning stage.

Data formulation for metric learning

Considering people are good at relative comparison as argued by the last section, the following forms of comparison data enjoys high large-scale availability for data collection, thus are extensively used by almost all metric learning algorithms.

- Similar pairs / Must-link constraints

$$\mathcal{S} = \{(x_i, x_j) : x_i \text{ and } x_j \text{ should be similar.}\} \tag{2.1}$$

- Dissimilar pairs / Cannot-link constraints

$$\mathcal{D} = \{(x_i, x_j) : x_i \text{ and } x_j \text{ should be dissimilar.}\} \tag{2.2}$$

- Triplet comparisons / Relative constrains / Training triplets

$$\mathcal{T} = \{(x_a, x_b, x_c) : x_a \text{ should be more similar to } x_b \text{ than to } x_c.\} \tag{2.3}$$

For triplet comparisons, x_a can be called as the anchor data point as it is compared to both of x_b and x_c.

Simple metrics

We first introduce some basic knowledge about metrics. Generally, a *distance function* should satisfy the following four conditions.

Definition 1. *A* distance *over a set $\mathcal{X}$ is a pairwise function $d : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ which satisfies the following* distance axioms $\forall x, x', x'' \in \mathcal{X}$

- *nonnegativity:* $d(x, x') \geq 0$,
- *identity of indiscernible:* $d(x, x') = 0$ *if and only if* $x = x'$,
- *symmetry:* $d(x, x') = d(x', x)$,
- *triangle inequality:* $d(x, x'') \leq d(x, x') + d(x', x'')$.

In a weaker form, a *pseudo-distance* is required to satisfy all but the second condition, where only $d(x, x) = 0$ is needed.

In contrary, a similarity function can be any form of a pair-input function, thus has less consensus on a general definition. To be clear, we adopt the following basic definition.

Definition 2. *A* similarity function *is a pair-input function $S : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$. When $S(x, x') = S(x', x) \forall x, x' \in \mathcal{X}$ holds, we call S a symmetric similarity function.*

Although being ahead of formal application on metric learning, we introduce similar definition of kernel functions which is a refined definition for symmetric similarity functions [130].

Definition 3. *A symmetric similarity function K is called a* kernel function *if there exists a mapping function $\phi : \mathcal{X} \to \mathbb{H}$ from $\mathcal{X}$ to a vector space $\mathbb{H}$ equipped with an inner product $\langle \cdot, \cdot \rangle$ (then $\mathbb{H}$ is called a* Hilbert space*) such that K can be rewritten as*

$$K(x, x') = \langle \phi(x), \phi(x') \rangle. \tag{2.4}$$

Equivalently, it is shown that K is a kernel if it is positive semi-definite (PSD), namely

$$\sum_{i=1}^{n} \sum_{j=1}^{n} c_i c_j K(x_i, x_j) \geq 0 \tag{2.5}$$

for all finite sequences of $c_1, \cdots, c_n \in \mathbb{R}$ and $x_1, \cdots, x_n \in \mathcal{X}$.

Finally, we introduce several widely used metrics. Without loss of generality, we assume all data points are located in an appropriate vector space $\mathcal{X} \subseteq \mathbb{R}^d$.

Minkowski distances are a family of distances defined by L_p norms. Specifically, it is defined as

$$d_p(\mathbf{x}, \mathbf{x}') = \|\mathbf{x} - \mathbf{x}'\|_p = \left(\sum_{i=1}^{d} |x_i - x'_i|^p \right)^{\frac{1}{p}}, \ p \geq 1. \tag{2.6}$$

It is a general family that has many commonly used metrics as its special cases.

- When $p = 1$, it gives the Manhattan distance as

$$d_{\text{Manhattan}}(\mathbf{x}, \mathbf{x}') = \|\mathbf{x} - \mathbf{x}'\| = \sum_{i=1}^{d} |x_i - x'_i|. \tag{2.7}$$

- When $p = 2$, we recover the ordinary Euclidean distance as

$$d_{\text{Euclidean}}(\mathbf{x}, \mathbf{x}') = \|\mathbf{x} - \mathbf{x}'\|_2 = \sqrt{(\mathbf{x} - \mathbf{x}')^\top (\mathbf{x} - \mathbf{x}')}. \tag{2.8}$$

- When $p \to \infty$, we recover the Chebyshev distance as

$$d_{\text{Chebyshev}}(\mathbf{x}, \mathbf{x}') = \|\mathbf{x} - \mathbf{x}'\|_\infty = \max_i |x_i - x'_i|. \tag{2.9}$$

More cases with different p values are illustrated in Figure 2.1.

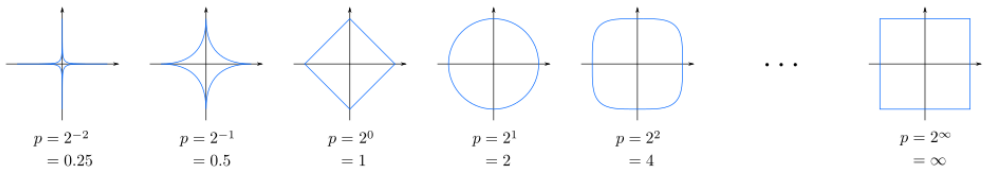


Figure 2.1: Minkowski distances with different parameters.

Mahalanobis distance [104] is initially defined to incorporates the correlation between $\mathbf{x}$ features as

$$d_{\mathbf{\Sigma}^{-1}}(\mathbf{x}, \mathbf{x}') = \sqrt{(\mathbf{x} - \mathbf{x}')^\top \mathbf{\Sigma}^{-1} (\mathbf{x} - \mathbf{x}')}, \tag{2.10}$$

where $\mathbf{x}, \mathbf{x}'$ are random vectors generated from the same distribution with covariance matrix $\mathbf{\Sigma}$. By generalization of the covariance matrix, denoted by $\mathbf{M}$, we define the Mahalanobis distance as the generalized quadratic distance as

$$d_{\mathbf{M}}(\mathbf{x}, \mathbf{x}') = \sqrt{(\mathbf{x} - \mathbf{x}')^\top \mathbf{M} (\mathbf{x} - \mathbf{x}')}, \tag{2.11}$$

where $\mathbf{M} \in \mathbb{S}_+^d$ and $\mathbb{S}_+^d$ denotes the cone of symmetric PSD $d \times d$ real-valued matrices. This condition ensures $d_{\mathbf{M}}$ is a properly defined pseudo-distance.

Interestingly, the Euclidean distance can also be recovered from this family by setting $\mathbf{M}$ as the identity matrix. Moreover, by expressing matrix decomposition $\mathbf{M} = \mathbf{L}^\top \mathbf{L}$ where $\mathbf{L} \in \mathbb{R}^{k \times d}$ and k is the rank of $\mathbf{M}$, we can rewrite as

$$d_{\mathbf{M}}(\mathbf{x}, \mathbf{x}') = \sqrt{(\mathbf{L}\mathbf{x} - \mathbf{L}\mathbf{x}')^\top (\mathbf{L}\mathbf{x} - \mathbf{L}\mathbf{x}')}. \tag{2.12}$$

The above form indicates that Mahalanobis distance can be interpreted as the Euclidean distance in a projected feature space induced by $\mathbf{L}$. This further indicates that metric learning and representation learning, to be introduced in the next section, implicitly share the same goal of looking for a useful mapping from the current feature space. It is thus not very surprised to see that representation learning also use the same form of comparison data as input.

Last but not least, the *cosine similarity* is also worth noting which measures the cosine of the angle formed by the two data points

$$S_{\cos}(\mathbf{x}, \mathbf{x}') = \frac{\mathbf{x}^\top \mathbf{x}'}{\|\mathbf{x}\|_2 \|\mathbf{x}'\|_2}. \tag{2.13}$$

It is widely used in the field of information retrieval [6, 139].

2.3.2 Linear metric learning

We first review several important studies on linear metrics, which is preferable among applications due to its simple form that can be efficiently learnt and calculated. Recall the definition of Equation 2.11, the goal is to learn the desired $\mathbf{M}$. It is a key to main the PSD constraint property of $\mathbf{M}$ during learning. For convenience, the squared form $d^2_{\mathbf{M}}(\mathbf{x}, \mathbf{x}')$ is usually used as the objective function.

Mahalanobis Metric for Clustering (MMC) [155]

It is unavoidable to introduce this seminal work which is the first Mahalanobis distance learning method, with an application on clustering. MMC uses the input data of similar pairs as defined by Equation 2.1 and dissimilar pairs as defined by Equation 2.2. The intuition for designing the objective function is simple and remains the same as the recent flourish development of deep metric learning that is going to be introduced in the following sections later: maximize the distances between dissimilar data points while keeping the distances between similar data points being small enough. It is formalized as

$$\begin{aligned} \max_{\mathbf{M}\in\mathbb{S}^d_+} & \sum_{(\mathbf{x}_i,\mathbf{x}_j)\in\mathcal{D}} d_{\mathbf{M}}(\mathbf{x}_i, \mathbf{x}_j) \\ \text{s.t.} & \sum_{(\mathbf{x}_i,\mathbf{x}_j)\in\mathcal{S}} d^2_{\mathbf{M}}(\mathbf{x}_i, \mathbf{x}_j) \leq 1. \end{aligned} \tag{2.14}$$

This becomes a standard constraint optimization problem and the authors propose to solve it using a projected gradient descent algorithm. One drawback is its high time complexity due to the projection onto the PSD cone which requires $\mathcal{O}(d^3)$ time to calculate all eigenvalues of $\mathbf{M}$ in order to set negative ones to zero. Recent studies [27, 160] formulate MMC to an eigenvalue optimization problem and successfully reduce the time complexity to $\mathcal{O}(d^2)$. Moreover, this method only looks at the summation of all distances, which permits the existence of extreme distance violations for few pairs.

Large Margin Nearest Neighbors (LMNN)

LMNN [149, 151, 150] can be considered as the most popular metric learning algorithm and a plethora of extensions are proposed based on it.

First, the input data is the same as a supervised learning problem, $\{\mathbf{x}_i, y_i\}_{i=1}^n$. Then, the similarity set and the triplet set are constructed in a similar way to the second part of this thesis, which is to be introduced in Chapter 4. Specifically, LMNN focuses on the *local* scale characterized by k nearest neighbors in the original Euclidean space. Formally, the two comparison sets are defined as follows:

$$\begin{aligned} \mathcal{S}_{\text{LMNN}} &= \{(\mathbf{x}_i, \mathbf{x}_j) : y_i = y_j \text{ and } \mathbf{x}_j \text{ is in the } k \text{ nearest neighbors of } \mathbf{x}_i\} \\ \mathcal{T}_{\text{LMNN}} &= \{(\mathbf{x}_i, \mathbf{x}_j, \mathbf{x}_k) : (\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{S} \text{ and } y_i \neq y_k\}. \end{aligned} \tag{2.15}$$

The learning objective is then defined as

$$\begin{aligned} \min_{\mathbf{M}\in\mathbb{S}^d_+, \xi\geq\mathbf{0}} & (1-\mu) \sum_{(\mathbf{x}_i,\mathbf{x}_j)\in\mathcal{S}_{\text{LMNN}}} d^2_{\mathbf{M}}(\mathbf{x}_i, \mathbf{x}_j) + \mu \sum_{i,j,k} \xi_{ijk} \\ \text{s.t. } & d^2_{\mathbf{M}}(\mathbf{x}_i, \mathbf{x}_k) - d^2_{\mathbf{M}}(\mathbf{x}_i, \mathbf{x}_j) \geq 1 - \xi_{ijk} \quad \forall (\mathbf{x}_i, \mathbf{x}_j, \mathbf{x}_k) \in \mathcal{T}_{\text{LMNN}}, \end{aligned} \tag{2.16}$$

where $\mu \in (0, 1)$ is the weighting coefficient that controls the trade-off between pulling together similar data points and pushing away different data points. The intuition for the objective is illustrated in Figure 2.2.

Note that the number of the constraints is in the order of kn^2, thus grows into an almost infeasible large number when dealing with modern large dataset. Prior to the deep learning era, many methods are then devoted on efficient optimization with a minimized number of constraints through methods such as careful book-keeping or focusing only the closest imposters. As for deep metric learning methods, this is no longer a difficult obstacle thanks to backpropagation and stochastic gradient descent.

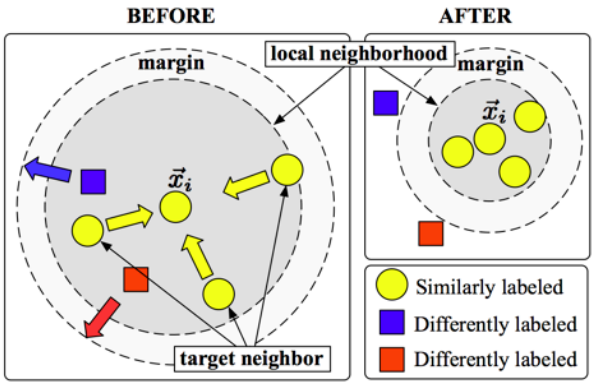


Figure 2.2: Illustration for the formulation of LMNN: pulling together similar data points and pushing away dissimilar ones.

Independent from the two methods introduced above, there are also methods developed that result to a similar formulation, with difference on the objective term or constraints. For example, both the methods proposed by Schultz and Joachims [133] and by Kwok and Tsang [91] chooses to use the squared Frobenius norm of $\mathbf{M}$ as the objective function. The formal one uses only the triplet comparison set $\mathcal{T}$ to generate constraints, and the latter one use both pairwise comparison sets $\mathcal{S}$ and $\mathcal{D}$. Additionally, both methods requires slack variables for feasible solutions.

Finally, we would like to introduce an interesting line of research that formulates the objective function from an alternative perspective.

Information-Theoretic Metric Learning (ITML)

This method proposes to use the LogDet regularization for metric learning. The intuitive motivation is to let $\mathbf{M}$ be close to a predefined matrix $\mathbf{M}_0$, such as the identity matrix inducing the Euclidean distance which can also be interpreted as a prior matrix from the Bayesian perspective.

This method propose to measure the distance between two matrices by a Bregman divergence called LogDet such as

$$D_{\text{LogDet}}(\mathbf{M}, \mathbf{M}_0) = \ \text{trace}(\mathbf{M}\mathbf{M}_0^{-1}) - \log\det(\mathbf{M}\mathbf{M}_0^{-1}) - d \tag{2.17}$$

It can be shown that the above distance is equivalent to minimizing the KL divergence between two multivariate Gaussian distributions parameterized by $\mathbf{M}$ and $\mathbf{M}_0$. Other interesting properties of the LogDet distance, such as scale invariance, translation invariance and range space preservation, can be found in Kulis et al. [90] for a detailed discussion.

Similar to other methods, the constraints of ITML are generated from pairwise comparison sets $\mathcal{S}$ and $\mathcal{D}$. Slack variables are also required for obtaining stable solutions. ITML enjoys the benefit of simple implementation and scalability to

high dimensional data and large datasets, but bears the limitation of the choice of $\mathbf{M}_0$ crucially influence the performance of the learnt distance.

2.3.3 Nonlinear metric learning

Linear methods have their limit as they cannot properly data points with complex high-order relations. It is intuitively motivated for advanced methods that can appropriately model the nonlinearity among data points, and the field of metric learning is not an exception to this trend. In the following, we would mainly introduce nonlinear metric learning methods based on kernelization, sometimes called the *kernel trick*, in which the kernel function (Definition 3) used has a similar form to the oracles used in Chapter 3.

The kernel functions are defined as functions of a pair of data points, thus usually called pairwise functions. Therefore, applying the *kernel trick* generally requires the original linear algorithm to be written in a form that input data points only appear in pairs. For a simple example, the Euclidean distance can be expanded as

$$\|\mathbf{x}-\mathbf{x}'\|_2 = \sqrt{\mathbf{x}^\top\mathbf{x} - 2\mathbf{x}^\top\mathbf{x}' + \mathbf{x}'^\top\mathbf{x}'}, \tag{2.18}$$

which can be easily kernelized by substituting the inner products with a proper kernel function.

More generally, consider a squared Mahalonobis distance in kernel space as

$$d^2_{\mathbf{M}}(\phi,\phi') = (\phi-\phi')^\top\mathbf{M}(\phi-\phi') = (\phi-\phi')^\top\mathbf{L}^\top\mathbf{L}(\phi-\phi'), \tag{2.19}$$

where we denote $\phi = \phi(\mathbf{x})$ and $\phi' = \phi(\mathbf{x}')$ for simplicity. Then, let $\mathbf{\Phi} = [\phi_1, \cdots, \phi_n]$, $\mathbf{k} = \mathbf{\Phi}^\top\phi(\mathbf{x}) = [K(\mathbf{x}_1,\mathbf{x}), \cdots, K(\mathbf{x}_n,\mathbf{x})]^\top$ and the same for $\mathbf{k}'$, using the expansion $\mathbf{L}^\top = \mathbf{\Phi}\mathbf{U}^\top$ where $\mathbf{U} \in \mathbb{R}^{D\times n}$ and D denotes the dimension of the feature space implicitly induced by the kernel function. We can get

$$d^2_{\mathbf{M}}(\phi,\phi') = (\mathbf{k}-\mathbf{k}')^\top\mathbf{U}^\top\mathbf{U}(\mathbf{k}-\mathbf{k}'), \tag{2.20}$$

which is feasible for computation for any pair of input data points without explicitly awareness of the feature space. This enables computation over potentially infinite dimensional feature space induced by such as the Gaussian radial basis function kernel function. Moreover, using kernel functions eases the algorithm to be extended to structural data, such as trees, graphs and time series data, without further careful treatment, as long as using a properly designed kernel function for the data structure. Theoretically, the application of the kernel trick to metric learning is substantially supported by a representation theorem [32].

However, it is sometimes nontrivial for kernelizing a metric learning, as it requires to limit the algorithm accessing data points only through pairs, most time inner products. To this end, general methods based on kernel principle component analysis (KPCA) [32, 163]. As the nonlinear extension of Principle component analysis (PCA), KPCA mapps data points into the feature space induced by the kernel function. Then, the original linear metric learning algorithms can be applied to this mapped space. This KPCA trick is shown to be theoretically sound and can avoid heavy computation [32].

2.3.4 Generalization guarantees for metric learning

Machine learning also pays attention to how well the algorithms would perform when applied to new situations, namely the ability of *generalization*, which forms a wide research topic throughout the field of statistical machine learning. Metric

learning is no exception to this, and we would like to review efforts on theoretical justifications of metric learning.

Specifically, the following two aspects of generalization is considerable in metric learning [18]:

- Consistency (Generalization of the metric): The objective is to investigate how well the learned metric will performance on unseen data pairs or triplets. Considering the data format, this can be further divided into theories on batch methods and online methods, where all data are available for the first case and data are arriving by sequence thus only the data of the current time step is available for the second case.

- Classification performance (Generalization of the downstream classifier learned on the metric): The objective is to investigate how well a classifier will perform when the classifier itself is learned after we learn the metric, thus we specifically use the word *downstream* to emphasize.

The second aspect can be seen as an extension to the first one in a chronological order. This thesis focuses on problem settings that directly learns classifiers from comparison data, bypassing the metric learning stage. Therefore, we would like to focus on the second aspect and introduce its recent developments. Due to the limitation of available theoretical tools, we can only investigate simple cases of linear and binary classifiers.

The development can be summarized in three stages: the proposal of a goodness definition in order to ease analysis [9, 8], its application as metric learning objective function [18], and the following extension using the notion of Rademacher complexity [59].

A criterion for similarity functions

Kernel functions, which are usually used as similarity functions, must be symmetric and PSD to be valid. However, this may be difficult to satisfy in some applications, and the potentially infinite dimension feature space induced by the kernel function is hard to manipulate. Driven by such motivation, the following goodness criterion is proposed to not only widen the definition of similarity functions, but also proved link to linear classifiers. Notation is slightly refined for the context.

Definition 4. *A similarity function $S : \mathcal{X} \times \mathcal{X} \to [-1, 1]$ is an (ϵ, γ, τ)-good similarity function for a binary classification problem considered on a distribution μ if there exists an indicator function $R(\mathbf{x})$ defining a set of "reasonable points" such that the following two conditions hold:*

- *A $1 - \epsilon$ probability mass of labeled data points $(\mathbf{x}, y)$ satisfy*

$$\mathbb{E}_{(\mathbf{x},y)\sim\mu}[yy'S(\mathbf{x}, \mathbf{x}')|R(\mathbf{x}) = 1] \geq \gamma. \tag{2.21}$$

- $\mathbb{E}_{(\mathbf{x},y)\sim\mu}[R(x)] \geq \tau$.

The first condition can be interpreted as that a significant $1 - \epsilon$ proportion of data points are more similar to reasonable points of the same class, namely $y = y'$ thus $yy' = 1$, than to reasonable points of different class. The second condition simply lower bounded the proportion of reasonable points with respect to the whole distribution. Then, it is feasible to construct a low-error linear binary

classifier in the space of similarities to reasonable points as $\sum_{(\mathbf{x},y)\sim\mu|R(\mathbf{x})} yS(\cdot,\mathbf{x})$ with a little abuse of notation on the data generation distribution of reasonable points. Balcan et al. [8] formally derived a theory to address the existence of such a classifier, which is omitted here. Essentially, if we are given an (ϵ,γ,τ)-good similarity function and enough number of data points, thus enough number of reasonable points, the theory states there exists a low-error linear binary classifier with substantially high probability in the space of similarities to reasonable points. Consequently, this linear binary classifier can then be calculated by solving a linear problem, which resembles an L_1-regularized linear support vector machine, thus enjoying computation efficiency.

As the objective function

As the favorable properties of the (ϵ,γ,τ)-goodness definition introduced in the last section, it is then well-motivated to use it as the objective of metric learning. Bellet et al. [18] formulate the following similarity learning for linear classification (SLLC) problem for bilinear similarities $S_{\mathbf{M}} : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ defined as

$$S_{\mathbf{M}}(\mathbf{x},\mathbf{x}') = \mathbf{x}^\top \mathbf{M}\mathbf{x}', \tag{2.22}$$

which is linear to either of two inputs. In order to learn $\mathbf{M}$ that satisfies Definition 4, the objective function of SLLC is defined as

$$\min_{\mathbf{M}\in\mathbb{R}^{d\times d}} \frac{1}{n}\sum_{i=1}^{n} \ell(\mathbf{M},\mathbf{z}_i,R) + \lambda\|\mathbf{M}\|_{\mathcal{F}}^2, \tag{2.23}$$

where $\mathbf{z}_i = (\mathbf{x}_i, y_i)$ denotes n labeled data points, R denotes the set of reasonable points, $|R| < n$, and the loss function following Definition 4 is defined as

$$\ell(\mathbf{M},\mathbf{z}_i,R) = \left[1 - \frac{y_i}{\gamma^n R}\sum_{k=1}^{|R|} y_k S_{\mathbf{M}}(\mathbf{x}_i,\mathbf{x}_k)\right]_+. \tag{2.24}$$

Consequently, the consistency of SLLC is investigated. Because the reasonable points are later drawn from the training data, it may not follow exactly the same distribution from which the training data are generated. Bellet et al. [18] then propose to adapt the framework of uniform stability [23].

Using the Rademacher complexity

As the uniform stability used in the previous section depends on a specific learning algorithm, Guo and Ying [59] propose a two steps method to derive generalization guarantees for linear classifiers based on the learned metric, using the notion of Rademacher complexity [16] which is usually used to express the richness of a hypothesis set whose size could be infinite. In the theoretical guarantees of Chapter 4, we also take advantage of Rademacher complexity to justify our proposed algorithm, as well as obtaining the convergence rate at the same time. Same as the previous section, the theorems are derived for the bilinear similarity functions.

We would like to note that in order to make $S_{\mathbf{M}}$ a valid kernel, additional constraint is added to $\mathbf{M}$ to be symmetric PSD. This is based on the motivation that the learned $\mathbf{M}$ can be used in many downstream algorithms without further consideration, such as support vector machines. Then, the same objective function as in the previous section is used, with the reasonable point set being enlarged to

be the whole training dataset. Within this setting, a generalization bound over the learned bilinear similarity is obtained, followed by a generalization bound for linear classifiers based on the learned bilinear similarity.

2.3.5 Deep metric learning

In additional to the attention to kernel methods mentioned in Section 2.3.3, Chopra et al. [37] starte on investigating the possibility of applying neural networks, especially convolutional neural networks (CNN) tailored for image processing, on dimension reduction and metric learning. For a function $f : \mathcal{X} \to \mathbb{R}^D$ parameterized as a CNN, the motivation is the same as to pull together similar pairs and push away dissimilar ones. With the powerful nonlinear expressiveness of CNNs, the objective can be simply formulated by the Euclidean distance in the projected feature space. Specifically, authors propose to minimize

$$(1-y)\|f(\mathbf{x}) - f(\mathbf{x}')\|_2^2 + y\exp(-\|f(\mathbf{x}) - f(\mathbf{x}')\|_2^2), \tag{2.25}$$

where $y = 0$ if $(\mathbf{x}, \mathbf{x}') \in \mathcal{S}$ and $y = 1$ if $(\mathbf{x}, \mathbf{x}') \in \mathcal{D}$.

Starting here is a travel to the empirical wasteland without theoretical supports. For a brief walkthrough, we will first introduce two seminal papers, followed by their extensions, and conclude this section by a reality check on the progress of deep metric learning.

Siamese network and triplet network

Siamese network [36] is an intuitive deep metric learning method. It introduces a margin hyper-parameter m for pushing away dissimilar data points and the loss function is defined as

$$L_{\text{siamese}}(\mathbf{x}, \mathbf{x}') = (1-y)d(\mathbf{x}, \mathbf{x}') + y(m - d(\mathbf{x}, \mathbf{x}')), \tag{2.26}$$

where we simplified notation using $d(\mathbf{x}, \mathbf{x}') = \|f(\mathbf{x}) - f(\mathbf{x}')\|_2^2$.

Using this loss function, similar pairs are learned to have representations extremely close to each other. Furthermore, optimization for representations of dissimilar pairs stop when their distance is more than the margin m. These issues cause the problem of learning imbalanced representations for similar and dissimilar pairs.

To this end, the triplet network [131] is proposed. As stated before, a triplet consists of three data points: an anchor point $\mathbf{x}_a$, a positive point $\mathbf{x}_b$ and a negative point $\mathbf{x}_c$. It is interpreted that $\mathbf{x}_b$ is more similar to $\mathbf{x}_a$, than $\mathbf{x}_c$ to $\mathbf{x}_a$. This is also the same formulation of comparisons as those used in Chapter 4.

Then, instead of forcing representation distances to be zero, the triplet loss just learns to have representation distances of $\mathbf{x}_b$ and $\mathbf{x}_a$ to be smaller than that of $\mathbf{x}_c$ and $\mathbf{x}_a$, and a similar margin hyper-parameter is used. The loss function illustrated by Figure is formally defined as

$$L_{\text{triplet}} = \max(0, d(\mathbf{x}_b, \mathbf{x}_a) - d(\mathbf{x}_c, \mathbf{x}_a) + m). \tag{2.27}$$

This forms *the* mainstream workhorse for deep metric learning and most of the recent methods are developed as extensions to this loss function.

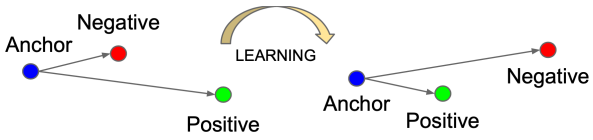


Figure 2.3: Illustration for triplet loss [131].

Triplet selection

Triplet loss also suffers the problem that learning process stops soon. This is because the number of all possible combinations of triplets are extremely large, which is roughly $\mathcal{O}(n^3)$ where n denotes the number of training data. When most of the possible combinations have negligible influence on parameters of the model f, it appears like the learning process stops. Empirically, this happens not very long after the learning starts [131]. Therefore, it is necessary to conduct selection for useful triplets before feeding them into the loss function.

Considering triplet selection given a fixed anchor point, it seems like there are two directions we can work on: the positive point and the negative point. However, deep metric learning is usually used in applications such as face re-identification, where the number of classes k are considerably large. Therefore, given an anchor point, the number of available positive point, namely data points belong to the same class, is roughly $\frac{1}{k}$ of negative points, namely data points belong to other classes. Consequently, extension methods usually conduct simple random selection for positive points, but carefully design selection process for suitable negative points.

Given an anchor point and a positive point, a useful negative point should cause the loss to be nonzero. That is to say, the negative point is closer to the anchor point in the representation space. Formally, the following three situations can be considered given a margin hyper-parameter as shown in Figure 2.4:

- Easy negative: $d(\mathbf{x}_a, \mathbf{x}_b) + m \leq d(\mathbf{x}_a, \mathbf{x}_c)$,
- Semi-hard negative: $d(\mathbf{x}_a, \mathbf{x}_b) \leq\leq d(\mathbf{x}_a, \mathbf{x}_c) \leq d(\mathbf{x}_a, \mathbf{x}_b) + m$,
- Hard negative: $d(\mathbf{x}_a, \mathbf{x}_c) \leq d(\mathbf{x}_a, \mathbf{x}_b)$.

Considering the existence of label noise, authors choose the strategy to mine semi-hard negative points for efficient triplet loss learning.

Reality check

Following the spirit of triplet loss, a plethora of extensions are proposed under the name of various motivations [77]. Lifted structured loss [118] is proposed to use all the pairwise edges among data points within one training batch for better computational efficiency. Multi-Class N-pair loss [140] generalizes triplet loss to include comparison with multiple negative samples. Contrastive Predictive Coding (CPC) [119] is proposed to use the information noise contrastive estimation (InfoNCE) loss to take advantage of categorical cross-entropy loss to identify the positive point amongst a set of unrelated noise points. This is inspired by noise contrastive estimation (NCE) [60] which is originally proposed to estimate parameters of statistical models. Finally, soft-nearest neighbors loss [128, 52] extends the triplet loss to include multiple positive points. Performance gain seems to be overwhelmingly claimed, and Musgrave et al. [112] decides to conduct a sanity

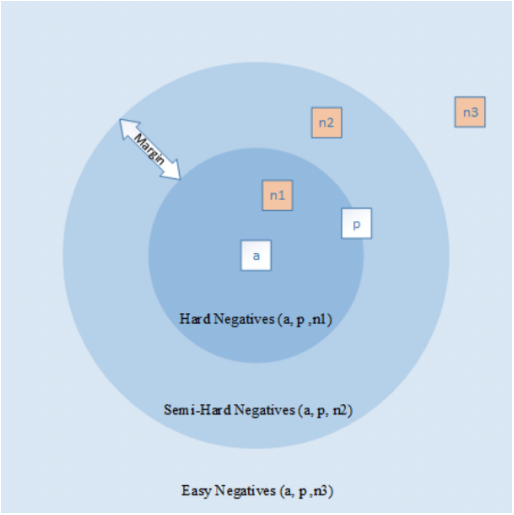


Figure 2.4: Three types of negative points.

check to have a sense of the actual situation. It is found that in most papers, the comparison against baselines is unfair in the sense of model architecture, data augmentation methods, optimizer and other options. Moreover, the popular metrics commonly used by many papers are found to be vulnerable in the sense of showing identical performance for different embeddings. More seriously, training using the test set is also found among literature.

To this end, authors set up a fair and reproducible comparison environment and conduct extensive experiments. There main results are shown in Figure 2.5. The sad message it tries to convey is that despite the claimed performance gain shown in the left, actually the deep metric learning field does not achieve considerable development in recent years if you cool down without listening to promotion of paper authors and compare each method in a fair way, as being shown by a flat wave in the right. Once again the crisis of reproducibility and how it can harm the development of a practically important field draw significant attention. Another point we would like to stress is that healthy and sustainable development of a science field can not be achieved in the absence of rigorous theoretical guarantees.

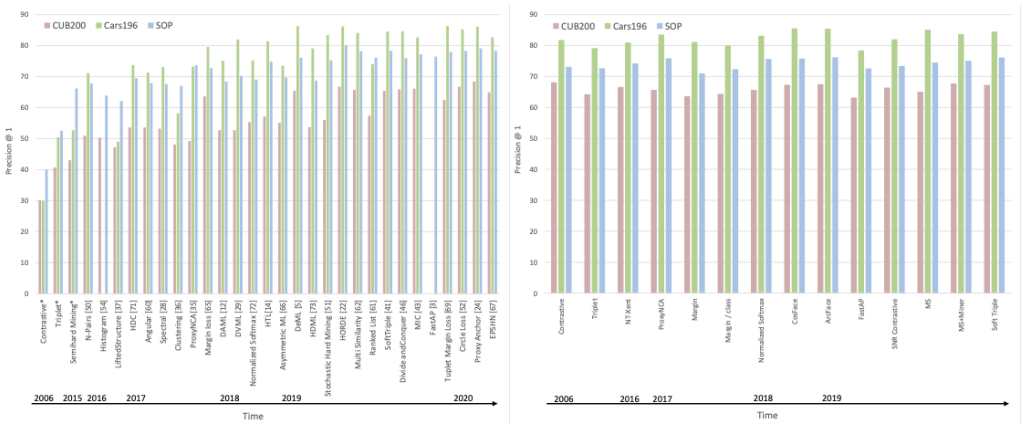


Figure 2.5: Left shows the trend according to papers and right shows the trend according to fair comparisons.

2.3.6 Relation to this thesis

In this section, we introduce the development of metric learning methods using pairs of data points as input. Other metric learning methods that either aims more than metric learning, such as achieving sparsity, boosting, or learning multiple local metrics at the same time, or do not directly or indirectly use comparison data, such as methods that solely use fully labeled data, are also omitted due to their irrelevance to this thesis.

2.4 Contrastive Representation learning

Contrastive representation learning [94], or self-supervised learning (SSL) is a recently emerged and rapidly growing field of machine learning, drawing heavy attention from both academic and industry communities. This is partially owing to the fast development of computation devices such as general purpose graphics processing unit (GPGPU) and the collection of large datasets [42]. It is also called contrastive unsupervised representation learning and being abbreviated as CURL. As the terminology is not unified, we can see how this field is under rapid development.

One main difference from deep metric learning is that representation learning does not require labels at all. As pairwise or triplet comparisons are constructed according to data point labels in deep metric learning, representation learning constructs such comparisons totally using unlabeled data by taking advantage of various data augmentation techniques.

Conceptually, the core idea is to design specific self-supervised tasks, also known as *pretext* tasks, to learn models that are supposed to capture the underlying structural representation of *unlabeled* data. Recent prosperous trend can be seen as driven by the success of two fields: natural language processing (NLP) and computer vision (CV).

Especially in the field of NLP, it is empirically observed that a huge model, usually consists of hundreds of billions of parameters, is able to show considerably high performance on downstream few-shot or zero-shot tasks. In article generation tasks, given the averagely high quality of generated contents, it is not surprising to observe unhealthy attention on these models. Although being criticised by some of the wide community including machine learning itself, the Stanford University insists to create a new research institution for research on this specific topic[2]. A long report has been put online to summarize the status of current work and show the direction of future research, as well as promote the new research institution [22].

In the following, we will briefly cover progress in each field. NLP pretext tasks are more domain specific and do not take advantage of data comparisons. As this thesis is about learning from comparisons, we will put more weight on the CV section of which pretext tasks prefer more on data comparisons. Note that the summary is emphasized on the problem setting, thus discussion on the model structure development is omitted, although it also serves an important role in achieving performance gain.

2.4.1 Pretext tasks in NLP

In NLP, similar idea of learning word representation, or *word embeddings* as usually called in NLP, using the information from context is not new. The seminal

[2] https://crfm.stanford.edu/

work of Mikolov et al. [106] propose an efficient method for this purpose which becomes the workhorse of most word embedding methods being used today. Another early work worth to mention is embeddings from language model (ELMO) [123] that learns contextualized word embeddings by pre-training the model in an unsupervised way. Taking advantage of the natural sequential property of sentences, the pretext task here is to predict the next word, or token, given a sequence of multiple prior tokens.

Recently, bidirectional encoder representations from transformers (BERT) [43] is shown to revolutionize many aspects of the field of NLP. Focusing on the pretext tasks it proposes, we can find the intuition and simplicity that can generalize when designing similar pretext tasks on other fields. The two tasks are listed as follows:

- Masked Language Model (MLM): The input sentence is randomly masked at a given percentage, and the model is pre-trained to predict the words that are masked. This task let the model learn the local information. The same format of test is also known as the Cloze test in the field of language education and psychology [145]. The word cloze is derived from the word closure, originating the law of closure in Gestalt psychology, indicating the hypothesis that humans tend to perceive objects, sentences in this case, as a whole. This pretext task is illustrated in Figure 2.6.

- Next Sentence Prediction: Instead of predicting the next word, this pretext task uses a pair of sentences as input. With half probability to be true, the model is pre-trained to answer whether the second sentence is the next sentence of the first one. This task let the model learn information beyond the local scope inside a sentence. This pretext task is illustrated in Figure 2.7.

Figure 2.6: Illustration for the MLM task.

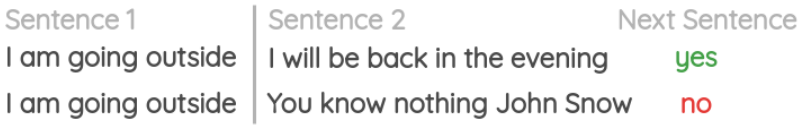


Figure 2.7: Illustration for the next sentence prediction task.

2.4.2 Pretext tasks in CV

Prior to the idea of learning from contrastive comparisons, various methods of pre-training has been proposed for better representation learning.

Examplar-CNN [45] is proposed create *surrogate classes* from totally unlabeled data. The idea is to sample fixed size patches from different images as different classes, and then distort them by applying various data augmentation methods to increase the number of data per surrogate classes. Cares are taken

such that patch cropping only takes place at where gradients are considerably high so that meaningful objectives will be covered. Note that the gradients here do not come from models, but simply mean the variations of image pixels. One sample of a generated surrogate class is shown in Figure 2.8.

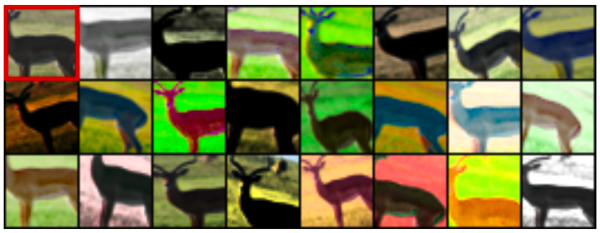


Figure 2.8: Illustration for Exampler-CNN. Top left is the orignal patch and others are generated by applying random transformations.

Using rotation angle as the signal for the pretext task is proposed as an alternative method for representation learning [56]. Considering the square shape of input images, authors propose to simply use four degrees $[0°, 90°, 180°, 270°]$ which do not need further treatment for image processing. Then, the pretext task becomes a four-class classification problem, which will help the model to learn and recognize high level object features that are commonly sensitive to rotation.

Similar to the manipulation of words in a sentence, it is intuitive to draw analogy in images by sampling a bag of patches. In this direction, Doersch et al. [44] propose a pretext task that trains the model to predict relative positions of a pair of patches. As shown in Figure 2.9, the center blue patch is first sampled. Different from Exampler-CNN, this step is conducted without considering the structural of the target image. Then, the second patch is sampled from one of eight possible positions. Authors propose various methods to avoid the model to learn trivial solutions driven by simple features such as continuous lines crossing through both patches. For example, sample the second patch with gap from the first patch, assign random jitters to patches, or randomly downsample image qualities to enhance the robustness of the model. An intriguing trivial solution called chromatic aberration that causes offsets between color channels can be avoided by shifting colors or just simply drop several color channels. It is an interesting observation that this methodology shows satisfying performance by accepting only *two* patches as input.

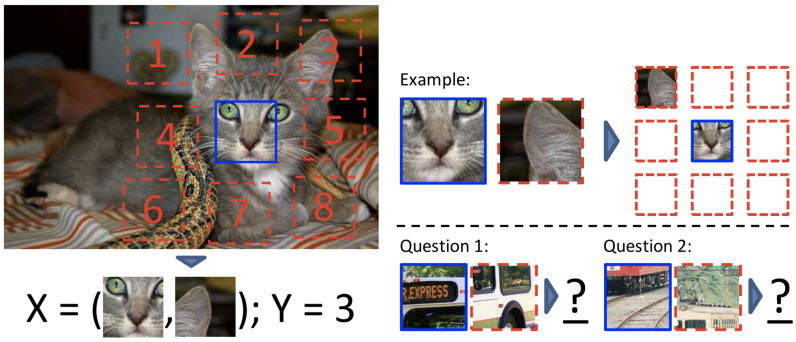


Figure 2.9: Illustration of self-supervised learning by predicting the relative position of two random patches.

Proceeding further in the direction of using patches, Noroozi and Favaro [115] propose to use the jigsaw puzzle composed of all nine patches as the pretext task. As shown in Figure 2.10, the model is pre-trained to be capable to answer the original places of nine shuffled patches. In order to make the problem easy for models to learn, authors propose to pre-define a fixed size of shuffling order set, and let the model to predict as a multi-class classification problem.

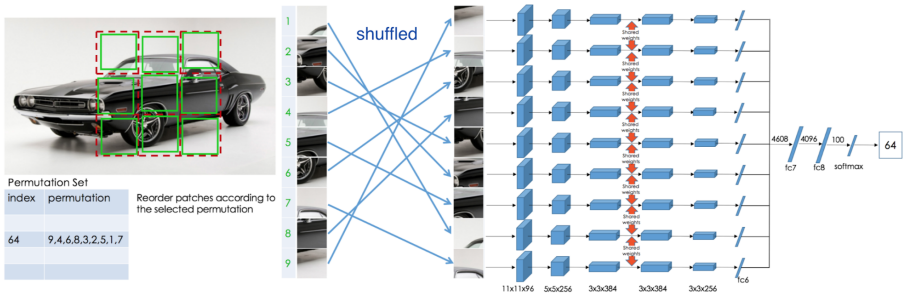


Figure 2.10: Illustration of self-supervised learning by solving jigsaw puzzle.

A follow-up method [116] simplifies the problem to learn a scalar that counts the number of features, or *visual primitives*, of an image. The pretext task here is to learn transformation invariant such scalar functions. Specifically, two kinds of transformation is considered.

- Scaling: If an image is being scaled, its number of features should stay the same.
- Tiling: If an image is tiled up into a $n \times n$ grid, the number of features should become n^2 times the original number.

Care is taken for avoiding trivial solution of the constant zero function by adding a regularization term to the mean squared error loss to encourage different image having different numbers. It is interesting to see learning carefully designed pretext task in a *scalar space* results well representation.

Lastly before introducing the recent trend, we would like to remark on generative modeling that learns from reconstruction. Although can be seen as a kind of pretext task, generative modeling methods such as variational autoencoders [81] or generative adversarial networks [57] aim to learn a model that can *generate* new data, rather than mapping a given input to a learned representation. This field itself is out of the scope of this thesis and worth another tens of theses.

Recent advances in contrastive representation learning

This line of research focus on learning from a pair of input images, which are usually generated from a single image by different data augmentation methods. The pseudo label for the pair is based on the assumption that the object identity does not change when data augmentation methods are applied.

SimCLR [33], short for a simple framework for contrastive learning of visual representations, is proposed in shed of the simple idea of encouraging agreement between differently augmented outputs from a single image. Notably, the representation space $\mathbf{h}$ is different from the space $\mathbf{z}$ where learning actually happens, which resembles the intermediate representation extraction in supervised learning and is shown in Figure 2.11.

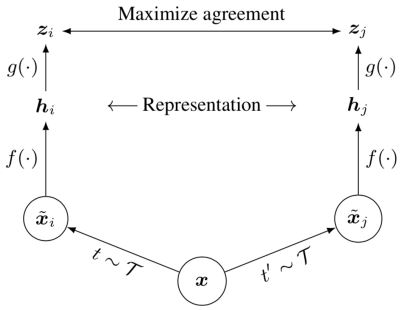


Figure 2.11: Illustration of the SimCLR framework.

In the learning procedure, first a batch of n images are sampled. Each of the sampled image are applied two augmentation operations from a predefined operation set to produce totally n pair of images. For an image in a pair, the positive sample is the other image in the pair, and the negative sample can be chosen from all $2(n-1)$ images from other pairs. Then, the model is learned using a contrastive loss with the cosine similarity $\text{CosSim}(\cdot,\cdot)$ as

$$L_{\text{SimCLR}} = -\log \frac{\exp(\text{CosSim}(\mathbf{z}_i, \mathbf{z}_j)/\tau)}{\sum_{k=1}^{2n} \mathbb{1}_{k \neq i} \exp(\text{CosSim}(\mathbf{z}_i, \mathbf{z}_k)/\tau)}, \tag{2.28}$$

where τ is a controlling hyper-parameter. Not surprisingly, SimCLR empirically needs to run on a large batch size to support enough negative samples, which resembles the phenomenon in deep metric learning using the triplet loss.

Following similar philosophy, Barlow twins [162] is proposed to maintain the *cross-correlation matrix* of different distortions of a single image to be close to the identity matrix, as shown in Figure 2.12. In this way, representations for different distortions from the same image will be similar to each other, in the sense that the redundancy is being minimized. This method naturally avoids trivial solutions, and more importantly it is robust to different batch sizes. The naming of this method is from the concept of *redundancy reduction* promoted by the neuroscientist Horace Barlow [14].

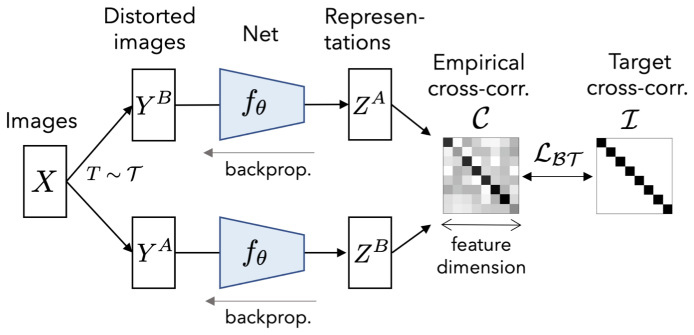


Figure 2.12: Illustration of the Barlow twins framework.

Bootstrap your own latent (BYOL) [58] is proposed to show the surprising unnecessariness of negative samples, indicating the name of the method. The framework itself resembles that of SimCLR, and as shown in Figure 2.13. The difference is at abolishing using the same network for both augmented images, but two different networks: the online network parameterized by θ that contains the

desired encoder, and the target network parameterized by ξ that aims learning. Both networks share the same model architecture, and more importantly the weights of the target network is updated by $\xi \leftarrow \tau\xi + (1-\tau)\theta$ where τ is a weighting hyper-parameter.

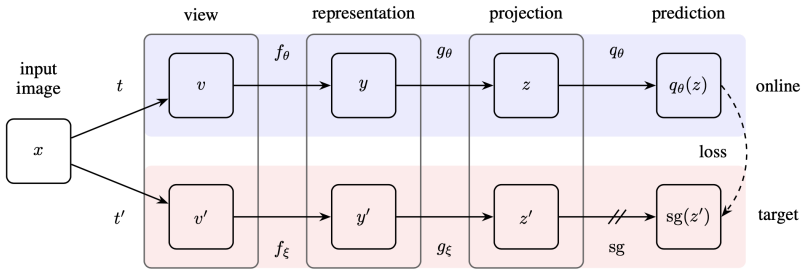


Figure 2.13: Illustration of the BYOL framework; sg means stop gradients as ξ is updated using θ.

Later, there are reported investigations on reproducing the results reported by the paper[3]. It is found that batch normalization [71], a technique for stabling neural network training, plays an important role in the learning process of BYOL. Without having batch normalization layers in the network structure, the BYOL framework averagely *performs no better than random*. The intuitive assumption is thus that the presence of batch normalization *implicitly* causes dependencies on negative samples, as values are normalized and re-distributed within a batch and the batch size is usually large in the framework.

Lastly, we would mention the momentum contrast (MoCo) frameworks [65, 34] which in contrast do not use batch normalization but propose an alternative efficient way for constructing negative samples. Similar to BYOL, MoCo maintains a *momentum network* that are updated as the same way of the target network in BYOL, as shown in Figure 2.14. This allows MoCo to take advantage of prjections / representations of the past batch as a first in first out (FIFO) queue to be used in the contrastive loss of the current batch of data. It is interesting to see that performance gain can be achieved by focusing on the efficiency of using negative samples, without care taken on their sampling scheme.

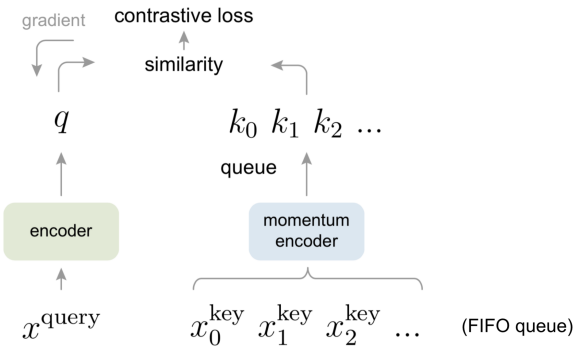


Figure 2.14: Illustration of the MoCo framework.

Additionally, because the assumption that being able to access only the unlabeled data pool may not be realistic in some situations, constrastive represen-

[3]https://generallyintelligent.ai/blog/2020-08-24-understanding-self-supervised-contrastive-learning

tation learning has also been explored in the online setting [30] where data point arrives one by one as a stream, or even with supervision [125, 79]. This is still a relatively new field and much more research is happening when this line is being read.

Theoretical analysis for contrastive representation learning

After witnessing the terrific empirical success of modern contrastive representation learning, such as competitive performance against fully supervised learning using deep neural networks by a simple downstream *linear* classifier, it is an intuitive motivation to conduct theoretical investigation in order to know and design algorithms better.

Investigations are mainly conducted on the contradiction around the number of negative samples for learning, denoted by K. Empirically, as shown in the previous section, it is important to have a large K for achieving satisfying performance. However, theoretical analysis so far has been struggling on matching this behaviour while providing probability bounds of the downstream classification risk. Essentially, the seminal work by Saunshi et al. [129] first proposed a lower bound for the contrastive loss, which turns to unfavorably increase when K becomes larger. Later, Nozawa and Sato [117] tried to fix the inconsistency between theory and practise by inspiration from Coupon collector's problem, but the bound is valid when $K+1 \geq C$ where C denotes the number of latent classes for supervision. Ash et al. [4] further relieved the restriction of K by proposing an alternative bound, but still requires of dependence on C. By using the notion of the mean supervised loss, which is empirically inaccessible in experiments, Bao et al. [11] presents sharp upper and lower bounds that matches empirical results much more tightly. However, the bound still have a distance from supervised loss, and the mysterious roll of negative samples in frameworks such as BYOL is not clearly understood and need future investigation.

2.4.3 Relation to this thesis

Contrastive representation learning is an unavoidable field when considering relate work for this field, although the problem setting is different depending on the existence of supervision. As the theoretical analysis on powerful practical methods are being investigated, we are looking forward to see new methods going to emerge and take advantage of weak supervision.

2.5 Weakly-supervised learning

Learning from weak supervision may sound vague and not specific as a title. Actually, this is true as every situation where the supervision signal is not perfect can be reasonably considered as learning from weak supervision. For example, there are cases of learning from full but noisy labels being called as weakly-supervised learning. However, we would like to focus on the classification problem from imperfect labels, which also mostly being referred by the umbrella term "weakly-supervised learning".

Apparently, this problem setting is well-motivated in practical applications. Although machine learning achieves astonishing success from *fully labeled* big data, where the fully labeled part is often ignored by narratives, there exist various applications where massive labeled data is not available, such as in medicine,

infrastructure or robotics, which heavily holding up the popularization of machine learning methods in the real world. Therefore, learning from many, but weak supervision is a promising direction for research. It tries to achieve both high classification accuracy as well as low labeling cost at the same time.

All methods to be covered in this section use the empirical risk minimization (ERM) framework, which is intuitive for training classifiers. Essentially, for a classifier model f parameterized by $\theta \in \mathbb{R}^d$, given a labeled dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$, we simply tries to find the configuration of θ that minimizes the empirical risk

$$\hat{\mathcal{R}}(f) = \frac{1}{n} \sum_{i=1}^{n} \ell(f(x_i), y_i), \tag{2.29}$$

where ℓ denotes a loss function. It is shown that the above empirical risk can be bounded by the population risk

$$R(f) = \mathop{\mathbb{E}}_{p(x,y)} [\ell(f(x), y)] \tag{2.30}$$

plus an additional error term that decreases in the order of $\mathcal{O}\left(\frac{1}{\sqrt{n}}\right)$, where $p(x, y)$ denotes the unknown underlying data generation distribution. Since the perfect full labels $\{y_i\}_{i=1}^n$ are considered unknown in weak supervision, efforts are made to find an alternative equivalent expression of the empirical risk using the form of imperfect label information at hand.

Note that besides classification risk stated above, there are also other objectives one can optimize, such as the area under the receiver operating characteristic (AUROC) for ranking or classification from imbalanced data. However, literature review are omitted for work on these directions as this thesis is mainly on binary classification.

2.5.1 Positive and unlabeled (PU) classification

In this problem setting, we assume only part of labels belong to the positive data is revealed, and the rest positive part as well as the whole negative part are left as unlabeled data. This problem settings has many favorable applications, such as classification on user logs where only happened interactions are recorded.

Unbiased risk estimators have been proposed [47, 46] by only using the PU data, instead of the fully unlabeled data. This risk reconstruction technique can be considered as the core of this series of methods. Formally, denoting the positive marginal distribution by $p_+(x) = p(x|y = +1)$, the negative marginal distribution by $p_-(x) = p(x|y = -1)$ and the data distribution by $p(x) = \pi p_+ + (1 - \pi)p_-$, where $\pi = p(y = +1)$ is called the *class prior* which is plays an important role in the risk rewriting process and is assumed to be known or can be precisely inferred. Then, it is shown [47] that the population risk can be equivalently rewritten as

$$\mathcal{R}(f) = \pi \mathop{\mathbb{E}}_{p_+(x)} [\ell(f(x), +1)] - \pi \mathop{\mathbb{E}}_{p_+(x)} [\ell(f(x), -1)] + \mathop{\mathbb{E}}_{p(x)} [\ell(f(x), -1)]. \tag{2.31}$$

The empirical version of the above estimator is known to enjoy favorable properties such as it is consistent with respect to all common loss functions. The consistency here means as the number of data points approaching infinity, the empirical risk estimator will approach the true population estimator. Moreover, it is shown [46] that if the loss function satisfies the so called symmetric condition which is not rare among popular loss functions:

$$\ell(t, +1) + \ell(t, -1) = 1, \tag{2.32}$$

the risk estimator can be further written as

$$\mathcal{R}(f) = 2\pi \mathop{\mathbb{E}}_{p_+(x)}[\ell(f(x), +1)] + \mathop{\mathbb{E}}_{p(x)}[\ell(f(x), -1)] - \pi, \tag{2.33}$$

which can be minimized by separating positive and unlabeled data with simple cost-sensitive learning. Conditions for the risk to be a convex function with respect to classifier parameters is further investigated.

Although being theoretically justified, it is often observed in experiments that overfitting is inevitable when using a highly flexible model, such as a neural network. This is because the term $\mathbb{E}_{p(x)}[\ell(f(x), -1)] - \pi\, \mathbb{E}_{p_+(x)}[\ell(f(x), -1)]$ can decreases too fast due to the flexibility of the classifier model. Therefore, a simple and intuitive extension is proposed [82] to forcefully set this term to be non-negative. The method is thus naturally coined as non-negative PU, and its risk estimator is

$$\mathcal{R}(f) = \pi \mathop{\mathbb{E}}_{p_+(x)}[\ell(f(x), +1)] + \max\left(0, \mathop{\mathbb{E}}_{p(x)}[\ell(f(x), -1)] - \pi \mathop{\mathbb{E}}_{p_+(x)}[\ell(f(x), -1)]\right). \tag{2.34}$$

Although this risk estimator is trivially biased because of the max operation, it is shown to be able to achieve an optimal convergence rate towards the true risk. Empirical results are also observed to be supportive.

Methods so far simply assume the positive data are independently and identically distributed (i.i.d.), which is natural for machine learning problems. This is because it is impossible to learn a classifier without casting assumptions on the generation process of positive data [49], and i.i.d. is the most simple and common assumption to be used. However, this may not be realistic in many applications of PU as the positive data may be collected in a way with ineluctable *selection bias*, such as patient health record classification or a recommendation system. This is to say, the distribution of the positive data collected may differ from that of the unlabeled data. Kato et al. [76] tackles this problem by a mild assumption, that $p(o = +1|\mathbf{x})$ and $p(y = +1|\mathbf{x})$ induces the same order on $\mathbf{x} \in \mathcal{X}$, where $o = +1$ denotes the event that the data points being observed and vice versa. Formally, it requires for any $\mathbf{x}_i, \mathbf{x}_j \in \mathcal{X}$ that

$$p(y = +1|\mathbf{x}_i) \leq p(y = +1|\mathbf{x}_j) \leftrightarrow p(o = +1|\mathbf{x}_i) \leq p(o = +1|\mathbf{x}_j). \tag{2.35}$$

This assumption can be intuitively understood as high positivity indicates high probability to be observed.

Consequently, the positive data is assumed to be generated from the biased distribution $p(\mathbf{x}|y = +1, o = +1)$. Then, it is shown that

$$p(y = +1|\mathbf{x}_i) \leq p(y = +1|\mathbf{x}_j) \leftrightarrow r(\mathbf{x}_i) \leq r(\mathbf{x}_j), \tag{2.36}$$

where $r(\mathbf{x}) = \frac{p(\mathbf{x}|y=+1,o=+1)}{p(\mathbf{x})}$ denotes the density ratio that is to be estimated from biased positive and unlabeled data. However, due to the weak assumption, this method still needs a hyper-parameter threshold to turn the learned scoring function of density ratio into a valid classifier, taking advantage of the order preserving property of Equation 2.36. This is similar to the approach taken in Chapter 3, but the method proposed by this thesis provides a concrete way to estimate the threshold with firm theoretical guarantees.

Although superior performance has been achieved theoretically and empirically so far, existing methods may still not be suitable enough for industry deployment. To this end, Kwon et al. [92] propose to optimize a modified version of

integral probability metric (IPM) to find classfiers, extending the similar method proposed by Sriperumbudur et al. [141] on fully supervised binary classification, where IPM itself is an extensively studied topic in statistics for its own goods.

Given the general IPM being defined with two probability measures P, Q on $\mathcal{X}$ and a function class $\mathcal{F}$ as

$$\mathrm{IPM}(P, Q; \mathcal{F}) = \sup_{f \in \mathcal{F}} \left| \int_{\mathcal{X}} f(x) dP(x) - \int_{\mathcal{X}} f(x) dQ(x) \right|, \tag{2.37}$$

it is shown that calculating IPM betwen $p(x|y = +1)$ and $p(x|y = -1)$ is *negatively related* to the minimization of a loss function as

$$\mathrm{IPM}(p(x|y = +1), p(x|y = -1); \mathcal{F}) = -\inf R_\ell(f), \tag{2.38}$$

where the loss function here follows specifically $\ell(+1, t) = -\frac{t}{\pi}$ and $\ell(-1, t) = \frac{t}{1-\pi}$. Thus, this can been seen as providing an alternative solution for binary classification. Drawing analogy to this process, Kwon et al. [92] first proposed a simplified weighted IPM (WIPM) as

$$\mathrm{WIPM}(P, Q; w, \mathcal{F}) = \sup_{f \in \mathcal{F}} \left| \int_{\mathcal{X}} f(x) dP(x) - w \int_{\mathcal{X}} f(x) dQ(x) \right|, \tag{2.39}$$

where w is the weighting hyper-parameter. Extending the constant w to a function on x as $w(x)$ will produced a generalized WIPM. Then, authors shows the critial theorem that it holds for the hinge loss $\ell_h(y, t) = \max(0, 1 - yt)$ that

$$\inf_f R_{\ell_h}(f) = 1 - \mathrm{WIPM}(p(x), p(x|y = +1); 2\pi, \mathcal{F}). \tag{2.40}$$

Loss functions that uses the input (y, t) only through its *margin* yt, such as the hinge loss, is usually called margin losses in the literature. In addition to theoretically investigate important properties, such as estimation error bound and excess risk bound for general function spaces , of the empirical estimator for WIPM, authors also provide a closed-form analytic solution assuming the function space to be a closed ball in a kind of vector space called reproducing kernel Hilbert space that can be characterized by a valid kernel function.

We would like to note that in Equation 2.32, all unlabeled data are actually treated as negative samples for loss computation, indicated by -1 being used for unlabeled data points, Inspired by the memorization effect of deep neural networks [164], Xu et al. [157] propose to treat unlabeled samples that causes large loss with the negative label to be pseudo positive samples. Being built upon the non-negative PU framework [82], it is shown that although being biased to the real risk to be optimized, carefully choosing when and how to select pseudo positive samples can significantly outperforming existing methods, indicating new insights on further evolution of PU learning.

2.5.2 Positive-negative-unlabeled (PNU) classification

Thanks to the symmetric property of binary classification, there is no need to create methods for the negative and unlabeled data (NU) case, as we can simply flip the labels to treat the original negative class as positive. Then, when we combine the classification risks for PU and NU, we can have a new risk that can use all positive, negative and unlabeled data, recovering the semi-supervised learning setting.

After constructing PU and NU risk estimators separately, Sakai et al. [127] proposed an intuitive method to combine these estimators by a weighting hyper-parameter. This enjoys favorable theoretically guarantees inherent from PU framework and can be easily extended by PU framework extensions such as non-negative PU. It is also shown that the variance of the combined risk estimator is always reduced under mild assumptions.

Following the aforementioned work, Hsieh et al. [70] pay attention to more realistic settings where only a small portion of all possible negative samples can be collected thus they may not be able to represent the underlying distribution of $p(x|y=-1)$. This is reasonable to happen in applications such as remote-sensing or text classification, where there are many diverse classes and it is hard to collect data that cover all classes.

Similar to the technique alleviating bias in the simple PU setting, authors introduce a latent variable s indicating the status of being observed. Formally, negative data points are assumed to be generated from the distribution $p(\mathbf{x}|y=-1, s=+1)$. Therefore, it is natural to assume $p(s=+1|\mathbf{x}, y=+1)=1$. Similarly, a risk estimator being defined by available data is derived under additional assumptions on the values of π and $p(y=-1, s=+1)$, which is argued to be easily estimated from data.

2.5.3 Unlabeled and unlabeled (UU) classification

Another direction of extending PU classification is totally getting rid of the label requirement, but instead assuming more knowledge about the underlying distribution. Specifically, we consider learning from two groups of unlabeled data with different underlying data generation distributions in UU classification.

Note that although being highly similar, classification is an inductive problem setting that requires the resulted classifier to be able to inference on newly unseen data points in the future, while clustering is an innately transductive problem setting that only requires manipulation on only existing data points. Moreover, clustering methods requires certain assumptions on data structure to be feasible, thus hindering the performance of the resulted classifier. The work by du Plessis et al. [48] pioneered the line of research of directly learning classifiers in such settings. However, a critical limitation exists that the performance measure can only be the *balanced error* [26], which is the classification accuracy when positive and negative data have same probability to be sampled, that is $\pi=0.5$. Then, Lu et al. [100] extend the work and first apply the risk rewriting technique to derive a valid risk estimator based on only two groups of unlabeled dataset with different class priors π_1 and π_2. After tedious algebra derivation, the risk estimator is defined as

$$\begin{aligned} R_{\mathrm{UU}} = & \mathop{\mathbb{E}}_{\mathbf{x}\sim p_1(x)}\left[\frac{(1-\pi_2)\pi}{\pi_1-\pi_2}\ell(f(\mathbf{x}),+1)-\frac{\pi_2(1-\pi)}{\pi_1-\pi_2}\ell(f(\mathbf{x}),-1)\right] \\ & + \mathop{\mathbb{E}}_{\mathbf{x}\sim p_2(x)}\left[-\frac{(1-\pi_1)\pi}{\pi_1-\pi_2}\ell(f(\mathbf{x}),+1)+\frac{\pi_1(1-\pi)}{\pi_1-\pi_2}\ell(f(\mathbf{x}),-1)\right], \end{aligned} \tag{2.41}$$

where $p_1(\mathbf{x})$ and $p_2(\mathbf{x})$ denote the two distribution where the two group of data are respectively drawn from, and π denotes the class prior for test distribution. Observing the operations on class priors, we can intuitive know the high sensitivity to class prior estimations. This is to say, a small error on the estimation of either on of π, π_1, π_2 may heavily harm the accuracy of the risk estimator. This risk estimator then enjoys the same accompanying favorable theoretical guaran-

tees, such as consistency to the true classification risk and high convergence rate, as those of other weakly supervised settings.

However, it is empirically found that severe overfitting occurs when learning using the above risk estimator. To this end, Lu et al. [101] propose to constrain certain terms to non-negative, similar to the non-negative PU learning method.

Lastly, concerning existing method on UU classifcation can only work with *two* groups of unlabeled data, Lu et al. [99] further propose a method for handling multiple groups of unlabeled data. This method first formulates an multiclass classification on predicting from which unlabeled group a data point comes from. Then, the learned multiclass classifier can be used to construct the final binary classifier.

2.5.4 Positive-confidence (Pconf) classification

Another direction to find weaker supervision is to relieve the dependence on unlabeled data, but requires a little more information on the only available positive data. This can be used in applications such that data from rival companies cannot be obtained or only positive results are reported due to the publication bias. Note that this problem setting is closely related to one-class classification depending the extra information required. This setting also resembles anomaly / novelty detection when only normal / ordinary data are available.

Ishida et al. [74] explore this direction by requiring the confidence information of positive data under the name *Pconf classification*. Formally, it requires to know $r(\mathbf{x}) = p(y = +1|\mathbf{x})$ to be given in addition to other basic information. Then, a risk estimator can be derived in a similar rewriting way as

$$R_{\mathrm{Pconf}} = \pi \mathop{\mathbb{E}}_{\mathbf{x}\sim p(\mathbf{x}|y=+1)} \left[\ell(f(\mathbf{x}), +1) + \frac{1 - r(\mathbf{x})}{r(\mathbf{x})} \ell(f(\mathbf{x}), -1) \right], \tag{2.42}$$

which can then be estimated by available data and corresponding information. This is a very intriguing problem setting that opens the possibility for various forms of weak supervision information and has the potential of many interesting extensions. One natural extension is to break the normality in assumptions and consider biased confidence data [137]. The confidence data can be prone to bias when there data annotators lack domain knowledge and experience or they have imbalanced knowledge on different classes. In this case, the skewed confidence is shown be possible for correction when requiring addition information of the misclassification rate of positive data points. We will cover some of other relevant Pconf extensions in the next section.

2.5.5 Weakly-supervised classification from pairwise data

In this section, we will cover weakly supervised learning that using *comparison data* as input, which coincides the focus of this thesis. Weak supervision so far is assumed on data points of which the generation process can be simply written using $p(x|y = +1)$ or $p(x|y = -1)$. As we will see, care needs to be taken on assuming a proper generation process when dealing with pairwise data.

Bao et al. [12] pioneer this line of research by considering pairs sharing similarity. Specifically, we only know the two data points in a similar pair share the same underlying class label, but have no information of the label being positive or negative. This is claimed to be particularly useful in applications concerning people's sensitive matters, where explicitly answering questions is mentally more difficult than just point out a person who shares the same status. When such

similar pairs and unlabeled data is available, a rewritten classifier risk can derived under a suitable assumption on the data generation process. Formally, similar pairs are supposed to be generated from the distribution

$$\begin{aligned} p_{\mathrm{S}}(\mathbf{x}, \mathbf{x}') &= p(\mathbf{x}, \mathbf{x}' | y = y' = +1 \vee y = y' = -1) \\ &= \frac{\pi_+^2 p_+(\mathbf{x}) p_+(\mathbf{x}') + \pi_-^2 p_-(\mathbf{x}) p_-(\mathbf{x}')}{\pi_+^2 + \pi_-^2}, \end{aligned} \tag{2.43}$$

where $\pi_+ = p(y = +1)$ denotes the class prior so far, $\pi_- = 1 - \pi_+$, $p_+(\mathbf{x}) = p(\mathbf{x}|y = +1)$, $p_-(\mathbf{x}) = p(\mathbf{x}|y = -1)$.

In contrast to similar pairs, it would be not difficult to collect dissimilar pairs, where collected opposite answers can be fully used without being discarded. To this end, Shimada et al. [136] propose to learn binary classifiers from three groups of data: similar pairs, dissimilar pairs, and unlabeled data. For easier notation, a variable indicating similarity s is introduced. Formally, the generation process of pairwise data is assumed as

$$\begin{aligned} p(\mathbf{x}, \mathbf{x}', s = +1) &= p(s = +1) p(\mathbf{x}, \mathbf{x}' | s = +1) = p(y = y') p(\mathbf{x}, \mathbf{x}' | y = y') \\ p(\mathbf{x}, \mathbf{x}', s = -1) &= p(s = -1) p(\mathbf{x}, \mathbf{x}' | s = -1) = p(y \neq y') p(\mathbf{x}, \mathbf{x}' | y \neq y'). \end{aligned} \tag{2.44}$$

Then, a risk estimator that only takes these three groups of data is derived and theoretically investigated.

Bao et al. [13] further investigate how to use similar and dissimilar pairs in a more principled way. Although lack of pointwise label, the similarities naturally form a class when considering a pair of data points as a whole. Denoting the risk as R_{point} and R_{pair} for ponitwise and pairwise risk respectively, it is shown that

$$\min\{R_{\mathrm{point}}(f), R_{\mathrm{point}}(-f)\} = \frac{1}{2} - \frac{\sqrt{1 - 2R_{\mathrm{pair}}}}{2}. \tag{2.45}$$

The left hand side denotes how good a pointwise classifier can achieve *up to label flipping*. This label flipping-unaware risk is thus expressed by the pairwise risk. This is to say, we can construct a valid pointwise binary classifier using a pairwise classifier, which can be trained using only similar and dissimilar pairs, and the class prior of the underlying distribution, telling us which class takes more portion.

In another direction, Cao et al. [29] considered to require full details of similarity: the similarity confidence score of each pair. Formally, it assumes to be able to access

$$\begin{aligned} s(\mathbf{x}, \mathbf{x}') &= p(y = y' | \mathbf{x}, \mathbf{x}') \\ &= \frac{\pi_+^2 p_+(\mathbf{x}) p_+(\mathbf{x}') + \pi_-^2 p_-(\mathbf{x}) p_-(\mathbf{x}')}{p(\mathbf{x}) p(\mathbf{x}')}. \end{aligned} \tag{2.46}$$

Taking advantage of the confidence information, a risk estimator can be derived without accessing unlabeled data.

Other than the similarity information of a pair, Feng et al. [50] turns to use the positivity comparison information. That is to say, the information of whether $p(y = +1|\mathbf{x}) > p(y = +1|\mathbf{x}')$ holds is available. This has wide applications when explicit label is hard to assign but relative comparison is easier to conduct. Formally, they assume all possible pairs are first sampled and then filtered by annotators, leaving only possible pairs of $\{(+1, -1), (+1, +1), (-1, -1)\}$. Then, the generation process for pairs are defined as

$$p(\mathbf{x}, \mathbf{x}') = \frac{\pi_+^2 p_+(\mathbf{x}) p_+(\mathbf{x}') + \pi_-^2 p_-(\mathbf{x}) p_-(\mathbf{x}') + \pi_+ \pi_- p_+(\mathbf{x}) p_-(\mathbf{x}')}{\pi_+^2 + \pi_-^2 + \pi_+ \pi_-}. \tag{2.47}$$

Then a valid risk estimator can be algebraically derived.

2.5.6 Weakly-supervised multiclass classification

We have covered weakly-supervised learning in the binary classification setting up until now, and will introduce similar settings in the multi-class classification, where richer forms of imperfect information can be explored.

The number of classes is usually very large in modern datasets. This causes mental difficulty for annotations to select *the correct class* from probably hundreds of classes. In order to relieve this burden, Ishida et al. [72] propose the form of *complementary labels*, that indicate a class that the data point does *not* belong to. Although this will require roughly k times more annotations where k is the number of classes, it is an interesting direction to explore more practically feasible and low difficulty annotation forms of supervision. Formally, it is assumed the data are drawn from the distribution

$$\bar{p}(\mathbf{x}, \bar{y}) = \frac{1}{k-1} \sum_{y \neq \bar{y}} p(\mathbf{x}, y), \tag{2.48}$$

where $\bar{y}$ denotes the complementary label. Then, a multiclass risk estimator is derived for complementarily labeled data by two famous multiclass losses: the one-versus-all loss and the pairwise-comparison loss. Although not restricting model selection, One clear flaw is the popular cross-entropy loss, which is a common choice in modern deep learning, cannot be adapted into the framework. Ishida et al. [73] further generalize the framework by proposing a complementary loss that can be coped with arbitrary losses.

On the other line of research concerning the confidence information, Cao et al. [28] discovered that fully labeled single class is enough for multiclass classification if all class confidences are availabel. That is to say, it is possible to rewrite the multiclass risk only by data from a single class y_s as $\{\mathbf{x}_i, r_i\}_{i=1}^n$, where $r_i = \{p(y|\mathbf{x}_i)\}_{y=1}^k$ denotes confidences for all k classes. Although being less practical as accurately correcting all confidences is difficult, this problem setting opens new mind to think about what kind of supervision is essentially needed in multiclass classification.

Lastly, we would like mention the method that cope with the same weakly supervised learning from a totally different Bayesian point of view. Probabilistic models with latent variables are designed for weak supervision and model parameters are learned by maximum likelihood estimation [165, 166]. In addition, regression from pairwise comparison data can also be addressed by the empirical risk estimation framework [156].

2.5.7 Surrogate loss functions

After defining various forms of classification risk, we discuss about how to actually conduct optimization on model parameters according to these risks. To this end, concerns are drawn on the selection of loss functions [15].

Formally, a loss function for binary classification $\ell : \{+1, -1\} \times \{+1, -1\} \to \mathbb{R}$ is a function that measures how wrong a prediction is according to the groundtruth label. The value returned by the loss function $\ell(\hat{y}, y)$ can be interpreted as when the true label is y, how much damage or cost is incurred by predicting it to be $\hat{y}$. When defining the goal as the population risk defined in Equation 2.30, we use the ideal *zero-one* loss function. In this loss function, $\ell(\hat{y}, y)$ will be 0 only when $\hat{y} = y$ and will be 1 otherwise.

When considering find a classifier that minimizes a classification risk defined by the zero-one loss, it would directly obtain the optimal classifier, also called the

Bayes classifier or *Bayes predictor*. However, when thinking from a pragmatic perspective, two problems arise:

- We do not have direct access to the underlying data distribution which the classification risk is defined on. This can be dealt with using samples to form empirical estimation, just as defined by the empirical risk in Equation 2.29.
- The other obstacle is that the zero-one loss is an obviously discontinuous function. Therefore, the empirical risk defined using the zero-one loss is very difficult to be optimized directly.

For the second issue on the zero-one loss function, a proxy loss function that is continuous and easy for optimization is considered under the name *surrogate loss functions*: $\Phi(t) : \mathbb{R} \to \mathbb{R}_+$, which are usually convex functions for computational reasons. A popular example of these is the *hinge loss function*:

$$\Phi(t) = \max(1 - t, 0), \tag{2.49}$$

which is the loss function used in the famous classification method *support vector machines* [148]. The name comes from that its shape is similar to the hinge part of a door. Another example is the logistic loss function:

$$\Phi(t) = \frac{1}{(1 + \exp(-t))}, \tag{2.50}$$

which is used by the logistic regression model. Several famous loss functions are shown in Figure 2.15.

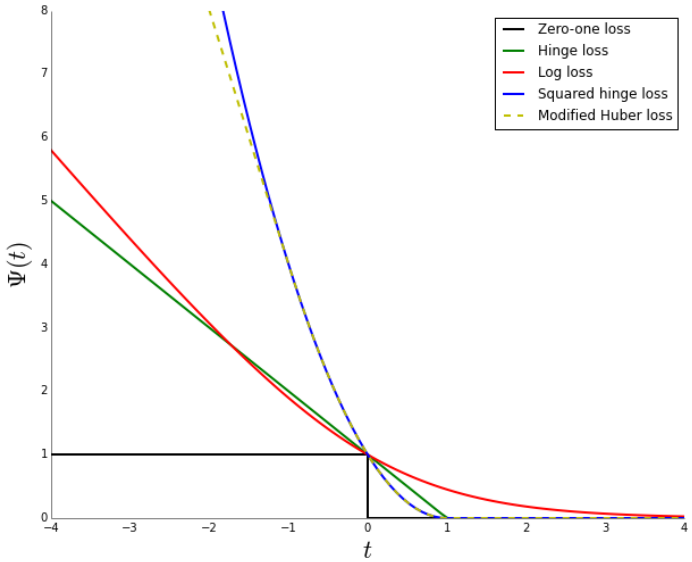


Figure 2.15: Illustration of several surrogate loss functions.

Then, a natural question rises that how much the optimization problem changes by replacing the ideal zero-one loss with surrogate losses. It is important to know how the obtained classifier would change when using different loss functions, and further how their generalization ability would change, which is the main concern of machine learning. This problem is then formulated as whether

the classifier that minimized the surrogate loss risk also minimizes the zero-one loss risk, which is defined as a property for surrogate losses called *consistency* or *classification calibration* [15]. This property is obviously dependent on how the surrogate loss is defined. One important finding on this property is that

Theorem. *If a surrogate loss function Φ is a convex function, then it is consistent or classification calibrated if and only if it is differentiable at zero and is derivative at zero is negative.*

This theorem indicates that most of the common surrogate loss functions, such as hinge logistic and huber, enjoys the good property. Thus, it is safe and theoretically justified to substitute the zero-one loss with above surrogate loss functions without worrying about the change of the outcome of optimization problem.

2.5.8 Relation to this thesis

Considering from the perspective of different forms of imperfect information used in weakly-supervised learning, the contributions of this thesis can be partly seen as an instance of this field. However, it should be noted that while most of the forms of imperfect information can be recovered when given full labels, uncertainty comparison information that are used in Chapter 3 is not interchangeably recoverable with a single set of full labels. However, when being accessible to a sufficiently large set of annotators, quantities similar to uncertainty can be aggregated from disagreement measures of collected annotations. It would be interesting to further investigate how the disagreement would reflect data uncertainties, such as feeding the calculated quantities in the algorithm proposed in Chapter 3, or proposing new methods on taking advantage of data uncertainties to assist learning binary classifiers.

2.6 Learning to rank

Pairwise comparisons are naturally amenable to the task of ranking. Although ranking in machine learning can also consider inductive generalization on unseen data, we focus on the transductive setting in this section which is closely related to this thesis. Besides recovering the full order over the set of all n items, usually only the top-k items where $k \ll n$ is enough for some applications, such as information retrieval where only the top results are actually needed and the rest are ignored.

2.6.1 Relation to this thesis

Our contribution in Chapter 3 uses a top-k selection algorithm as a subroutine. Thus, we would briefly cover the literature around such algorithms in this section.

In general, the Bradley-Terry-Luce (BTL) model [24, 102] is assumed for its simplicity and flexibility. It assumes the latent scalar variable $\{w_i\}_{i=1}^n$ for n items. The probability for the results of comparing a pair of items i and j follows

$$y_{ij} = \begin{cases} 1, & \text{with probability } \frac{w_i}{w_i + w_j} \\ 0, & \text{otherwise,} \end{cases} \tag{2.51}$$

where $y_{ij} = 1$ indicates that item i is larger than item j.

In the presence of random and passive sampling, where the algorithm cannot choose which pair of items to compare, Chen and Suh [35] propose a practical

algorithm that is shown to have guarantees on successfully identify top-k items. It also provides a lower bound for l to successful identification of top-k items when each pair is allowed to be repeated for l times, which is an important theoretical result for identifiability. In the theorems, the term $\Delta_k = \frac{w_k - w_{k+1}}{w_{\max}}$ is shown to play a crucial role, which resembles the corresponding content of Chapter 3.

Under the same problem setting, Shah and Wainwright [135] relieve the BTL assumption on the data generation process. Instead, a simple Copeland method, or the Borda count method, that counts each item i the quality $N_i = \sum_{j=1}^{n} \sum_{\ell} \mathbb{1}_{Y_{ij}^{\ell}=1}$ where

$$Y_{ij}^{\ell} = \begin{cases} 0, & \text{no comparison between } (i,j) \text{ in trial } \ell, \\ +1, & \text{if comparison is made and item } i \text{ beats } j, \\ -1, & \text{if comparison is made and item } j \text{ beats } i. \end{cases} \tag{2.52}$$

Then, a simple algorithm that return the k items with highest scores is shown to have practically satisfying performance. In theoretically investigating the bounds on the Hamming error of the results, a similar term $\Delta_k = \frac{1}{n}\sum_{i=1}^{n} M_{ki} - \frac{1}{n}\sum_{i=1}^{n} M_{k+1,i}$ where M_{ij} denotes the probability that item i beats item j also plays an important role.

Next, we introduce methods where the algorithm can *actively select* pairs for comparison, instead of running on passively collected comparison results.

Szorenyi et al. [144] focus on retrieving the single best item, which is top-1 selection in other words. Not suprisingly, the BTL model for data generation is assumed. They show that by applying the QuickSort algorithm [68], an ϵ-optimal item can be retrieved under a reasonable query complexity. However, although being theoretically justified, the algorithm is vulnerable to noises as the comparison for each pair is only conducted at most once.

Motivated by the applications of peer grading or crowdsourcing, Braverman et al. [25] propose an algorithm that can work with repetition for better accuracy. For the theoretical bounds on error to be valid, the uniform noise model is assumed instead of the BTL model, which needs all comparisons are universally correct with probability $\frac{1}{2}+\frac{\gamma}{2}$ where γ is a hyper-parameter. The high level design of the proposed algorithm is to select a subset of size $\sqrt{n}$, and then query almost all pairs within the subset as well as across the subset. The idea of working on a selected delegation subset is interesting and also inspires the method in Chapter 3, where the selected subset nevertheless serves a distinctly different role in the overall algorithm.

2.7 Theoretical active learning

When thinking of relieving the burden of annotating a huge amount of data, active learning [134] provides an alternative point of view. It designs algorithms that can *actively* select which unlabeled data point to query, in order to achieve a more efficient annotation budget overall. Among possible problem settings, we consider the most simple case called *pool-based* active learning, where the unlabeled data points are given initially, as a pool, and the algorithm would iteratively select unlabeled data points, receive there annotations, and loop these actions until meeting the stop criterion, such as running out of annotation budget.

2.7.1 Relation to this thesis

In Chapter 3, we adopt an instance of active learning algorithm to handle cases when the size of unlabeled data overwhelms the annotation budget. Thus, we would briefly introduce essentials about our selection, followed by active learning methods under weak supervision.

2.7.2 Disagreement-based active learning (DAI)

In this section, we will mainly cover a subset of active learning algorithms that come with rigid theoretical guarantees, the *disagreement-based* active learning (DAI) framework [64].

The active learning process is formulated as locating an optimal function within an initially given perhaps infinite-size function set, or hypothesis set $\mathcal{H}$. DAI focues on the disagreement region of $\mathcal{H}$ at each step

$$\mathrm{DIS}(\mathcal{H}) = \{x \in \mathcal{X} : \exists f, f' \in \mathcal{H}, f(\mathbf{x}) \neq f'(\mathbf{x})\}, \tag{2.53}$$

which is believed to provide useful information once annotated for reduction on the hypothesis set.

The seminal work of Balcan et al. [7] provide a concrete DAI algorithm with rigid theoretical guarantees on both consistency, that the optimal hypothesis will be left in the shrinking set, and query complexity. They consider to reduce the hypothesis set according to a term this is derived directly from a generalization bound.

On the other hand, many new methods mainly using deep neural networks have been recently proposed, mostly lacking rigid theoretical guarantees on its performance. Almost identical to the development of metric learning using deep neural networks, after many papers claiming significant performance gain, it is shown that actually the use of unlabeled data turns to play a more important role than query strategies, which are the main contributions of most of the papers [138]. Once again this shows the importance of theoretical understanding on algorithms, as well as a unified experiment protocol for reproducibility.

2.7.3 Active learning with weak supervision

Beygelzimer et al. [21] used a search oracle that receives a function set as input and outputs a data point *with its explicit class label*. Other two methods by Xu et al. [159] and Kane et al. [75] use the same oracle as positivity comparison. However, they all need to access the explicit labeling oracle. On the other hand, Balcan et al. [10] use only the class conditional queries (CCQ) without accessing the explicit labeling oracle. However, labels can be inferred from a single CCQ query. Although we cannot directly compare both the CCQ query with the query to be introduced in Chapter 3, we claim that ours is weaker than CCQ because explicit labels cannot be inferred from the query results unlike CCQ.

Moreover, feedback in the form of pairwise comparisons has also been considered in the literature of multi-armed bandits [20], which are instances of sequential decision making problems and can be considered as parts of the broad active machine learning. Comparing two arms and get the relative feedback, *dueling* two arms, are used as a cost-effective alternative feedback form. Under the similar motivation, Xu et al. [158] proposed an algorithm to combine comparisons and explicit feedback for the problem of thresholding bandits [98], which aims to find arms better than a given threshold.

Chapter 3

Learning from Uncertainty and Positivity Comparisons

In this chapter, we introduce our entire conducted on a new form of comparison feedback: the uncertainty comparison. After stating the motivation for proposing a new form of comparison feedback, we then address its feasibility by designing an accompanying algorithm that uses the proposed uncertainty comparison feedback. We also provide rigorous theoretical guarantees on its performance under various conditions. Then, we conduct extensive *user studies* to further verify the feasibility of the overall proposal. We not only design useful and instructive user study from scratch, but also carefully investigate what the observation indicate about the feedback and the algorithm. Finally, we conduct simulation studies on the algorithm to examine various concerning properties, as well as comparison against existing similar methods. At last, we summarize the chapter with discussion on directions of future research.

3.1 Introduction

On simple learning settings, it is known that active learning can achieve exponential improvement over passive learning under certain conditions [64]. However, this improvement does not always hold for more general situations. Consequently, active learning methods have been developed by casting assumptions on the underlying data distribution and the target concept, or designing different forms of oracles that can better take advantage of the knowledge of annotators.

This study focuses on the latter approach, specifically on methods incorporating the *positivity comparison oracle* into active learning. As already shown in Chapter 2, this form of oracle has very high practicality in applications and has already been extensively used in many fields of machine learning. It is obvious that without the knowledge of the classification threshold, we can at most sort all unlabeled data points according to their class-posterior probabilities using feedback from only this oracle. Weakly-supervised learning methods introduced in Chapter 2 all assumes to know the class prior in advance. Intuitively speaking, the class prior value tells how much the positive class takes over the whole data distribution. After sorting data points along class-posterior probabilities, we can simply place the classification threshold at the place indicated by the class prior, as we assume data points are generated in an i.i.d. manner thus should be spread over the line with the same trend as the underlying distribution. Figure 3.1 shows this procedure with an illustrative distribution when the class prior is given as $\pi_+ = 0.4$. Therefore, existing methods [75, 159] still need to access the *explicit labeling oracle* to infer labels when the class prior is not given.

Among the existing methods, Kane et al. [75] takes a geometric approach,

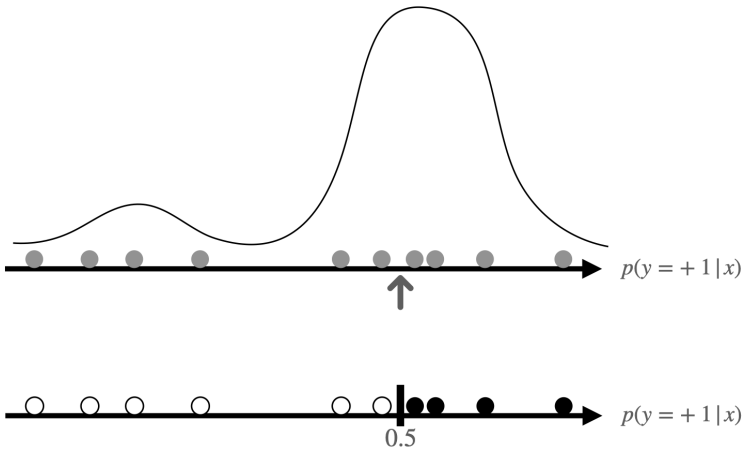


Figure 3.1: Illustration of allocating the classification threshold on sorted data points.

thus results a dimension-dependent query complexity. Moreover, it only considers the noise-free setting of oracles, which limits its practicality. As real-world user feedback is always noise-prone, it is necessary to reckon and design robust algorithms that can work even when the feedback is noisy.

Thus, we focus on the other method by Xu et al. [159] throughout this chapter. Initially given n unlabeled data points, the main idea of the algorithm proposed by Xu et al. [159], Active Data Generation with Adversarial Comparison (ADGAC), can be summarized as

- Conduct *quick sort* on the all data points. This causes $\mathcal{O}(n \log n)$ queries to the positivity comparison oracle. All feedback from the oracle is treated as if it is noise-free. As we will show later in experimental results, this treatment seriously affect the overall performance of ADGAC.

- Conduct *binary search* to locate the classification threshold. This causes $\mathcal{O}(\log n)$ queries to the explicit labeling oracle. However, this efficient approach of binary search will miserably fail when the sorted list is not correct, due to the treatment in the first step.

- Infer all data points that has higher class posterior possibility than the classification threshold as positive, and vice versa. Then, any off-the-shelf classification algorithms can be applied to data with inferred pseudo-labels.

On the first sight, it would be reasonable to improve the sort performance, because its accuracy is crucial for the following binary search. Without a highly accurate sorting results, the binary search will not return useful pseudo-labels, then let along the downstream classification. This is indeed the trend of research on ranking from noisy comparisons, when only concerning this subroutine.

However, we would like to step back a little to rethink about it and pay attention to the fact that even knowing the exact positivity comparison order of all data points is *not necessary* for the goal of label inference at all. More specifically, when given an unlabeled data point $\mathbf{x}$, we are only interested in the relationship between $p(y = +1|\mathbf{x})$ and the classification threshold where $p(y = +1) = 0.5$, not the relationship between $p(y = +1|\mathbf{x})$ and class-posterior probabilities of any other data points $p(y = +1|\mathbf{x}')$. On the other hand, knowing all class labels is also

not a sufficient condition for reconstruction of the sorted list of class-posterior probabilities. Therefore, we can conclude that sorting over all data points is not well motivated for label inference. Moreover, we will show the existing method empirically degrades almost linearly with increasing noise rate by an illustrative experiment. This further restricts the its feasibility and motivates for a better solution.

However, can we totally get rid of comparison information since sorting is not necessary? We would like to argue that the answer is no, if we lack the information of explicit labels. When designing the oracle and the corresponding algorithm, we recognize comparison over at least a subset of data points is inevitable due to collaborating with the explicit labeling oracle and the lack of information of the classification threshold. Intuitively speaking, without the absolute position of a data point on the data distribution, we have to reply on its relative position to find the best place for it.

In summary, although it is reasonable to improve the naive quick sort approach of the existing method, we choose to pose a question at a higher level: *"What form of feedback can efficiently and robustly provide information of the classification threshold?"*. With an ideal form of feedback, we should be capable to save positivity comparison queries from unnecessary sorting. We would like to note that this question is fundamentally motivated by the Vapnick principle [148], which guided many algorithm design in machine learning and beyond.

> If you possess a restricted amount of information for solving some problem, try to solve the problem directly and never solve a more general problem as an intermediate step.

In our case, sorting all data points is apparently a more general problem as it subsumes label information as part of its solution.

In this chapter, our problem is the lack of information of the classification threshold. To this end, we propose a new form of oracle, the *uncertainty comparison oracle*, which asks annotators to compare *uncertainties* of a pair of data points. Existing oracles demand the information of distances between class posteriors $p(y = +1|\mathbf{x})$ and $p(y = +1) = 1$. This requires the annotators to have the domain knowledge of what data with $p(y = +1|\mathbf{x}) = 1$ and $p(y = -1|\mathbf{x})$ should look like and also on how to compare data points along the axis of $p(y|\mathbf{x})$. In contrast, the proposed uncertainty comparison demands an alternative form of information concerning *distances between class posteriors and the classification threshold*. For annotators lacking domain knowledge, which would happen in many real-world situations, we believe it would be easier and more accurate to conceptualize what data with $p(y = +1|\mathbf{x})$ equals the classification threshold would look like, which represents data that have no salient features for judgement and we later examine this assumption using properly designed user experiments. Formally, we assume that higher uncertainty indicates being closer to the classification threshold $p(y = +1) = 0.5$. Using this new oracle with a corresponding algorithm, we can efficiently and robustly select the subset of data points with high uncertainties, which appear the closest to the classification threshold. Then, using this selected subset as a delegation of the unknown classification threshold, we can further infer labels of the majority of unlabeled data points with accuracy guarantees. Not surprisingly, the expensive and sometimes noise-prone explicit labeling oracle is no longer needed due to its inferior compatibility with pairwise comparisons.

3.2 Interactive label inference with pairwise comparisons

In this section, we introduce the two pairwise comparison oracles and the proposed label inference algorithm.

3.2.1 Two pairwise comparison oracles

Positivity comparison oracle This oracle receives two data points as input and answers whether the first data point has a higher probability of being positive. The answer "+1" means "yes" and "−1" means "no". This is a popular oracle that has been used in many different fields such as interactive classification [75, 159] and preference learning [38, 53]. Let $\eta(\mathbf{x}) \triangleq p(Y = +1|X = \mathbf{x})$ denote the underlying conditional probability for a data point $\mathbf{x}$ being positive. We describe this oracle by $\mathrm{O}_{\mathrm{pos}} : \mathcal{X} \times \mathcal{X} \to \{+1, -1\}$ and define it as follows.

Condition 1. *Distribution $\mathcal{P}_{\mathcal{X}\mathcal{Y}}$ and oracle O_{pos} satisfy this condition with a noise parameter $\epsilon_{pos} \geq 0$ if for every pair $(\mathbf{x}_1, \mathbf{x}_2) \sim \mathcal{P}_{\mathcal{X}\times\mathcal{X}}$, it holds that $(\eta(\mathbf{x}_1) - \eta(\mathbf{x}_2))O_{pos}(\mathbf{x}_1, \mathbf{x}_2) < 0$ with probability at least ϵ_{pos}.*

Intuitively, the oracle will return the correct answer with probability at least ϵ_{pos}, which is a hyper-parameter indicating the noise level.

Uncertainty comparison oracle This is our proposed oracle that receives two data points as input and answers whether the first one has higher uncertainty. The answer "+1" means "yes" and "−1" means "no". We define the uncertainty of a data point $\mathbf{x} \in \mathcal{X}$ as the difference between $\eta(\mathbf{x})$ and the classification threshold 0.5. This difference $|\eta(\mathbf{x}) - 0.5|$ being small means $\mathbf{x}$ has high uncertainty. We denote this oracle by $\mathrm{O}_{\mathrm{unc}} : \mathcal{X} \times \mathcal{X} \to \{+1, -1\}$ and define it as follows.

Condition 2. *Distribution $\mathcal{P}_{\mathcal{X}\mathcal{Y}}$ and oracle O_{unc} satisfy this condition with a noise parameter $\epsilon_{unc} \geq 0$ if for every pair $(\mathbf{x}_1, \mathbf{x}_2) \sim \mathcal{P}_{\mathcal{X}\times\mathcal{X}}$, it holds that $(|\eta(\mathbf{x}_2) - 0.5| - |\eta(\mathbf{x}_1) - 0.5|)O_{unc}(\mathbf{x}_1, \mathbf{x}_2) < 0$ with probability at least ϵ_{unc}.*

Similar to Condition 1, we also parameterize the oracle condition with a noise level ϵ_{unc}. Note that the above conditions only hold a weak assumption on error rates and collected answers need not hold for a proper order. This is to say, according to the oracle, even if $\mathrm{O}(\mathbf{x}_1, \mathbf{x}_2) = +1$ and $\mathrm{O}(\mathbf{x}_2, \mathbf{x}_3) = +1$, it need not to indicate that $\mathrm{O}(\mathbf{x}_1, \mathbf{x}_3) = +1$, where O denotes either of the two oracles. Moreover, it is totally possible to collect positive answers from both $\mathrm{O}(\mathbf{x}_1, \mathbf{x}_2)$ and $\mathrm{O}(\mathbf{x}_2, \mathbf{x}_1)$, which means the asymmetricity of the oracle conditions, or from $\mathrm{O}(\mathbf{x}_1, \mathbf{x}_2)$, $\mathrm{O}(\mathbf{x}_2, \mathbf{x}_3)$ and $\mathrm{O}(\mathbf{x}_3, \mathbf{x}_1)$, which means the intransitivity of the oracle conditions. In summary, our assumptions are relatively weaker compared to parametric models, such as the Bradley-Terry-Luce model [24, 102].

We would to also note that the above conditions consider the Massart's noise for both inter-class and inner-class comparisons, which is different from the more general noise condition used by Xu et al. [159]. We argue that considering under noises is important, but developing a similar algorithm under more challenging noise conditions would not be significantly different, thus is not explored in this chapter.

3.2.2 Proposed labeling algorithm

In the following, we propose an efficient and robust labeling algorithm taking advantage of the uncertainty comparison oracle. Given a set of unlabeled data

points D sampled from $\mathcal{P}_{\mathcal{X}}$ with size n, the idea is to first select a subset of t data points $D' \subset D$ as a delegation for the classification threshold where $t \ll n$. Note that we do not need to know the ranking order based on class-posterior probabilities of either D' or $D \setminus D'$, as discussed in the previous section. We would like to find this subset by actively accessing the uncertainty comparison oracle as few times as possible. Technically, this can be formulated as a top-t selection problem from noisy (uncertainty) comparisons and has been well studied in their own goods. We would like to stress that because we want to select the most uncertain data points, thus only the uncertainty comparison oracle $\mathrm{O}_{\mathrm{unc}}$ will be queried in this step. We choose the theoretical-guaranteed and practically promising algorithm proposed by Mohajer et al. [109] as the first step of subroutine for our algorithm. As it being an important part of the proposed algorithm and also for the self-containment of this thesis, we briefly introduce this top selection algorithm and its theoretical properties.

Top selection algorithm from noisy comparisons [109]

The goal here is select a subset D' consists of t data points from D consists of n data points based on a noisy comparison oracle. The algorithm can be described in following steps with illustrations shown in Figure 3.2.

1. Randomly separate the dataset into t subsets with equal size of $\frac{n}{t}$.
2. On each subset, conduct a randomly formulated single-elimination tournament to select a single data point with the highest uncertainty. Because the comparison results are noisy, each comparison can be repeated m times, where m is a hyper-parameter.
3. Build a heap structure on the t data points each of which is selected from t subsets.
4. Move the data point at the top of the size t heap structure to D'.
5. On the subset where the first element belongs to, conduct a randomly formulated single-elimination tournament with m repetitions to select a new data point, and insert it to the heap structure.
6. Repeat step four and five for the rest $t - 1$ times.

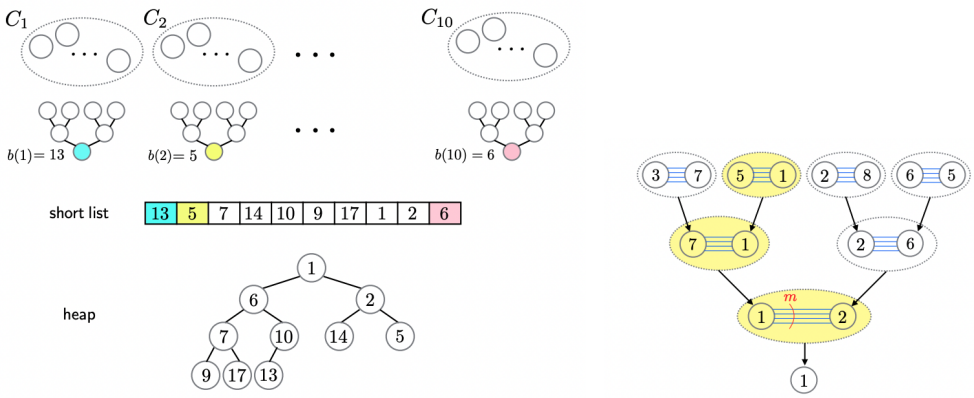


Figure 3.2: Illustration for the overall top selection algorithm on the left and single-elimination tournament with m repetitions on the right [109].

Although the above algorithm is a simple combination of single-elimination tournament and a heap structure, it is shown to enjoy the following favorable query complexity bound.

Theorem 1 (Complexity bound for top selection [109])**.** *With probability exceeding* $1-(\log n)^{-C_0}$*, the subset of top-*t *instances can be identified by the above algorithm with the query complexity upper bounded by* $C_1(n+t\log t)\frac{\max(\log\log n,\log t)}{(\epsilon_{unc}-0.5)^2}$*. Here,* C_0, C_1 *are some universal positive constants.*

After selecting the subset D', which consists of t most uncertain data points, we then use the positivity comparison oracle $\mathrm{O}_{\mathrm{pos}}$ to infer labels of the rest of data points, namely the other subset $D \setminus D'$. To conclude, the whole procedure of the proposed algorithm can be summarized in following three steps.

1. Use $\mathrm{O}_{\mathrm{unc}}$ and the top selection algorithm to find D', the t most uncertain data points.

2. For each $\mathbf{x} \in D \setminus D'$, use $\mathrm{O}_{\mathrm{pos}}$ to compare it with data points in a subset of D' to infer its label by majority votes. Note that the total number of comparisons at this step is controlled by the size of subset and at most the size of D', which is a hyper-parameter.

3. With no further information on D', we can choose to either assign random labels to data points in D', or repeat the algorithm using D' as the initial input. Because we do not assume the original D is i.i.d. sampled from the underlying data distribution, the true label distribution of D' could be skewed and no algorithm will perform better than chance. We show in Figure 3.3 a set of somehow uniformly distributed data points where splitting from the half will work and in Figure 3.4 a set of skewly distributed data points where no proper way of label assigning would give accurate answers without the prior knowledge of the classification threshold, or of the underlying data distribution.

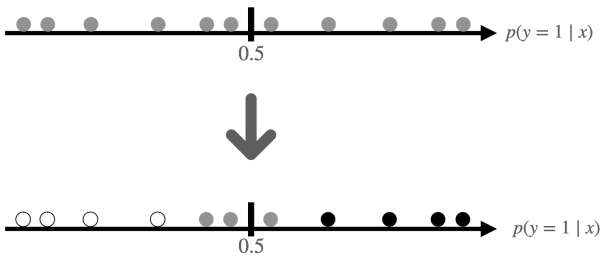


Figure 3.3: Conceptual illustration of well distributed data points.

Figure 3.5 illustrates the concept of the proposed algorithm under the simple condition when the size of D' is $t = 1$.

This algorithm can efficiently and accurately infer labels without requiring unnecessary ranking order according to class-posterior probabilities. For further

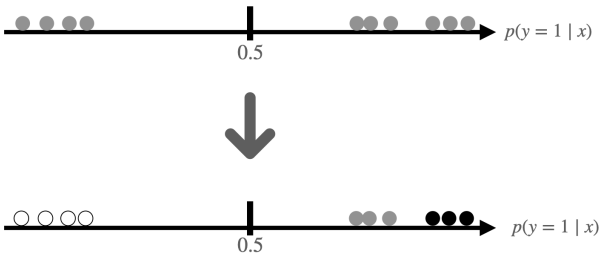


Figure 3.4: Conceptual illustration of skewly distributed data points.

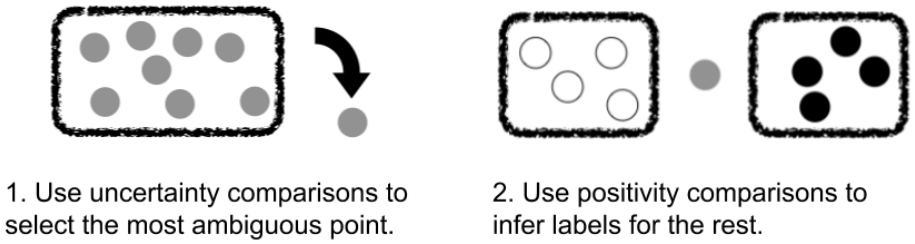


Figure 3.5: Conceptual illustration of the proposed algorithm.

clarification, the algorithm is formally described in Algorithm 1. On guaranteeing its performance, an error rate bound for inferred labels under noise conditions is established in Section 3.3.1.

Algorithm 1 Proposed Labeling Algorithm

Require: Positive integer t, dataset D with size n.

1: Select t most uncertain data points from D using the top selection algorithm [109] and $\mathrm{O}_{\mathrm{unc}}$. Denote the selected subset as D'.
2: **for** $x_i \in D \setminus D'$ **do**
3: **If** $\frac{1}{|D'|} \sum_{x_j \in D'} \mathrm{O}_{\mathrm{pos}}(x_i, x_j) \geq \frac{1}{2}$
4: Let $\hat{y}_i \leftarrow +1$.
5: **Else**
6: Let $\hat{y}_i \leftarrow -1$.
7: **end for**
8: Randomly label $x_i \in D'$ or repeat the algorithm using D' as the initial set.

Ensure: Inferred labels $\hat{Y} \triangleq \{\hat{y}_i\}_{i=1}^{n}$.

3.2.3 Learning classifiers under different budgets

For downstream tasks, we can feed the initial unlabeled set D and inferred pseudo labels $\hat{Y}$ into any algorithms that rely on samples from $\mathcal{P}_{\mathcal{X}\mathcal{Y}}$. In this chapter, we consider the task of learning binary classifiers under the following two different situations.

Passive case In this case, we assume there is enough budget for running Algorithm 1 passively on the whole dataset D. Then, we can obtain the inferred labels and feed them into any classification algorithms. In this paper, we consider the simplest non-parametric k-nearest neighbour (k-NN) algorithm [2] among numerous available supervised learning algorithms from the following two perspectives:

- The k-NN algorithm is almost the simplest algorithm for implementation that achieves favorable performance. It can achieve almost 98% classification accuracy on the MNIST dataset [95], which is a standard machine learning dataset. It can be used as a feasibility check tool.
- It is theoretically well studied and justified algorithm. Building on existing investigations, generalization bound for k-NN classifiers learned from labels inferred by Algorithm 1 can be derived in a concrete way.

Active case In this case, we consider a more practical situation where the dataset is too large compared to the budget. As a result, we cannot afford to run Algorithm 1 passively on the whole dataset D. To this end, we resort to using active learning with Algorithm 1 as a subroutine for the selected batch at each step. The same as Xu et al. [159], we consider a disagreement-based active learning algorithm calling the proposed Algorithm 1 at each step.

We would like to clarify that there are two levels of loop existing in this case. First, in the outer loop of (pool-based) active learning, the following three steps are repeated given initially unlabeled data points:

1. Select a subset of from unlabeled data points, according to the active learning algorithm.
2. Query the oracle, usually the explicit labeling oracle, for the selected subset.
3. Move the selected subset and its labels out of the unlabeled dataset.

Then, Algorithm 1 is used for the subroutine at Step 2. That is, instead of querying the explicit labeling oracle, we run Algorithm 1 to infer pseudo labels for the select subset of unlabeled data points, and used as if they are the true labels in the following steps. Note carefully that the above Step 1 is totally decided by the outer active learning algorithm. In this case, we use one typical disagreement-based active learning algorithm shown below in Algorithm 2.

Algorithm 2 Disagreement-based active learning algorithm [7]

Require: Desired error ϵ, a sequence of n_i, a hypothesis set H.

$H_1 \leftarrow H$
for $i = 1, 2, \cdots, \lceil \log(\frac{1}{\epsilon}) \rceil$ **do**
  $S_i \leftarrow$ i.i.d. sample from $\mathcal{P}_{\mathcal{X}}$ with size n_i.
  $D_i \leftarrow \mathrm{DIS}(S_i, H_i)$.
  Run Algorithm 1 with $\epsilon_i = \frac{1}{2^{i+2}}$ and D_i, obtain $\{\hat{y_j}\}_{j=1}^{|D_i|}$.
  $H_{i+1} \leftarrow \{h \in H_i : \sum_{j=1}^{n_i} \mathbb{1}_{h(x_j) \neq \hat{y}_j} \leq \epsilon_i n_i\}$
end for

Ensure: Any classifier in H_{i+1}.

3.3 Theoretical analysis

In this section, we provide theoretical justification for the algorithms proposed. Specifically, we establish the error rate bound for Algorithm 1 and generalization error bounds for classifiers learned by the downstream k-NN algorithm and Algorithm 2.

3.3.1 Analysis of the proposed labeling algorithm

Theorem 2 (Error rate bound)**.** *Suppose the following situations hold:*

1. *Condition 1 and Condition 2 hold for* $\epsilon_{pos}, \epsilon_{unc} \in [0, 0.5)$.
2. *There exist* $t = \Omega\left(\frac{\log 2}{2(0.5-\epsilon_{pos})^2}\right)$, $\epsilon > 0$, $D \subset \mathcal{X}$ *and* $n > \frac{t}{\epsilon}$, *where* $n = |D|$ *denotes the size of the initial unlabeled dataset.*

Then, there exist constants C_1 *and* C_2 *such that running Algorithm 1 on* D *with hyper-parameters* t *and* $m \geq \frac{C_1 \max(\log\log n, \log t)}{(0.5-\epsilon_{unc})^2}$, *with probability at least* $1-\delta$ *the following will hold:*

- *The error rate of inferred labels is bounded as* $|\{i \in [n] | \hat{y}_i \neq h^*(x_i)\}| \leq \epsilon n$.
- *The query complexity for* O_{pos} *is* $\mathcal{O}\left(\frac{n}{(0.5-\epsilon_{pos})^2}\right)$.
- *The query complexity for* O_{unc} *is* $\mathcal{O}\left(\frac{n \log\log n}{(0.5-\epsilon_{unc})^2}\right)$.

For simplicity, we denote $\delta \triangleq \delta(C_2, n, t, \epsilon_{pos})$.

Proof can be found in Section 5.1.

We then explain some observations that can be drawn from the theorem. First, for a desired error rate ϵ of pseudo labels, the existing method requires oracle noises falling into favorable ranges for a promised performance. For example, denoting adversarial noises of two oracles as ν_{exp} and ν_{pos}, indicating the slight difference in the definitions from ϵ_{pos}, Theorem 5 in Xu et al. [159] states they should satisfy $\nu_{\text{exp}} = \mathcal{O}(\epsilon)$ and $\nu_{\text{pos}} = \mathcal{O}(\epsilon^2)$. However, the above theory does not impose such dependencies between noise rates $\epsilon_{\text{pos}}, \epsilon_{\text{unc}}$ and the error rate ϵ. This means we can achieve *any desired error rate* with enough query budgets, regardless of underlying noise rates, allowing for broader real-world application scenes.

Also note that the above theory shows a principled way to select the hyper-parameter t, which is the size of the delegation subset. The ideal value of t appears to only depend on ϵ_{pos}, which is also a reasonable result as the subset if only used to be compared with other data points using O_{pos}. For a reasonable range of $\epsilon_{\text{pos}} \leq 0.4$, Algorithm 1 only requires t to be at most 35, which is relatively small compared to the size of a modern dataset. For the other hyper-parameter m, which is the repetition number for each comparison lacking a theoretical guidance on its value, we later empirically observe that a surprisingly small value, even 1, shows promising performance.

Moreover, because the computational complexity of the top-k selection subroutine is $\mathcal{O}(n + t\log t)$ and $t \ll n$, the proposed algorithm has the computational complexity of $\mathcal{O}(n + t\log t + (n-t)t) = \mathcal{O}(nt + t\log t)$.

3.3.2 Analysis of nearest neighbors classifiers

We then establish a generalization error bound for classifiers obtained by combining Algorithm 1 and k-NN. Formally, we want to estimate the function $\eta(\mathbf{x})$ from the inferred labels by Algorithm 1. For $\mathbf{x} \in \mathcal{X}$, we denote indices of other points in a descending distance order by $\{\tau_q(\mathbf{x})\}_{q=1}^{n-1}$. This means that for a metric ρ, it holds $\rho(\mathbf{x}, \mathbf{x}_{\tau_q}) \leq \rho(\mathbf{x}, \mathbf{x}_{\tau_{q+1}})$ for $q \in [1, n-2]$. Thus, we can denote the resulting k-NN classifier as $\hat{f}(\mathbf{x}; k) = \frac{1}{k} \sum_{q=1}^{k} \hat{y}_{\tau_q(\mathbf{x})}$.

We then introduce two essential assumptions, which are shared in many existing theoretical investigations on various k-NN-based algorithms. First, we need a general assumption for achieving fast convergence rates for k-NN classifiers.

Assumption 1 (Measure smoothness)**.** *With $\lambda > 0$ and $\omega > 0$, for all $\mathbf{x}_1, \mathbf{x}_2 \in \mathcal{X}$, it satisfies*

$$|\eta(\mathbf{x}_1) - \eta(\mathbf{x}_2)| \leq \omega \mu \left(B_{\rho(\mathbf{x}_1, \mathbf{x}_2)}(\mathbf{x}_1)\right)^{\lambda}, \tag{3.1}$$

where $B_{\rho(\mathbf{x}_1, \mathbf{x}_2)}(\mathbf{x}_0)$ denotes a ball with center $\mathbf{x}_0$ and radius $\rho(\mathbf{x}_1, \mathbf{x}_2)$.

This intuitively assumes there exists a well behaving function η in the sense that the difference between its results on two inputs will not fall apart from each other at a very long distance (achieving smoothness) considering the underlying metric / measure.

Then, we need the following Tsybakov's margin condition [105], which is a common assumption for establishing fast convergence rates.

Assumption 2 (Tsybakov's margin condition)**.** *There exist $\alpha \geq 0$ and $C_\alpha \geq 1$ such that for all $\xi > 0$ it holds that*

$$P\left(\left\{\mathbf{x} \in \mathcal{X} : 0 < \left|\eta(\mathbf{x}) - \frac{1}{2}\right| < \xi\right\}\right) \leq C_\alpha \xi^\alpha. \tag{3.2}$$

Note that the case for $\alpha = 0$ is trivial which is included for only notation convenience. The above assumption regularize the behaviour of η when being close to $\frac{1}{2}$ to some extent, which turns out to play a crucial rule on investigating the convergence of the resulting classifier. Detailed discussions on the margin condition can be found in Tsybakov et al. [146].

Finally, we can establish the generalization error bound.

Theorem 3 (Generalization error bound for classifiers learned from downstream k-NN)**.** *Let the input and the output of Algorithm 1 be $D = \{\mathbf{x}_i\}_{i=1}^n$ and $\hat{Y} = \{\hat{y}_i\}_{i=1}^n$. Let $\hat{f}(\mathbf{x}; k)$ be the k-NN classifier obtained and $f^*(\mathbf{x}) \triangleq \mathbb{1}_{\eta(\mathbf{x}) \geq \frac{1}{2}}$ be the Bayes classifier. Suppose the following situations hold:*

1. *The conditions for Theorem 2 hold.*
2. *Assumption 1 holds with $\lambda > 0$ and $\omega > 0$.*
3. *Assumption 2 holds with $\alpha \geq 0$ and $C_\alpha \geq 1$.*

Then, using the same notations as Theorem 2, for $\delta' \in (0, 1)$, $4\log(\frac{1}{\delta'}) + 1 \leq k \leq \frac{n}{2}$, with probability at least $(1-\delta)(1-\delta')$, it holds that

$$R(\hat{f}) \leq R(f^*) + C_\alpha \left(\frac{2\epsilon}{k} + \omega \left(\frac{2k}{n}\right)^{\lambda}\right)^{\alpha+1}. \tag{3.3}$$

Proof can be found in Appendix 5.2.

The difference between the above bound and other generalization bounds under unknown asymmetric noise [54, 109] is that Theorem 3 does not require the labels to be an i.i.d. sample from an underlying distribution. This is because they are instead inferred by Algorithm 1 and need not to satisfy the common i.i.d. assumption.

3.3.3 Analysis of disagreement-based active learning

We establish the generalization error bound by the following corollary to justify Algorithm 2. Its proof can be found in Appendix 5.3.

Corollary 4 (Generalization error bound for classifiers learned with disagreement-based active learning)**.** *Suppose conditions for Theorem 2 hold. Then, when running Algorithm 2 with* $\epsilon \in (0,1)$ *and* $\epsilon_i = \frac{1}{2^{i+2}}$*, with probability at least* $1-\delta$*, the output* $\hat{h}$ *satisfies*

$$P_{\mathbf{x}\sim\mathcal{P}_{\mathcal{X}}}[\hat{h}(\mathbf{x}) \neq h^*(\mathbf{x})] \leq \epsilon. \tag{3.4}$$

Note that ϵ, the final generalization error of the output classifier $\hat{h}$, can be set as a hyper-parameter for Algorithm 2, as long as the given budget is enough for running all rounds. Thus, according to the above theorem, we can predict the range of ϵ given the available budget in a principled way.

3.4 Simulation study

We start our empirical evaluation from this section. First, we confirmed the feasibility and performance of the proposed algorithm using simulated data. In order to provide a whole view on the performance, we repeated with 10 different seeds (ranging from 0 to 9) and report their mean values and standard deviations for each experimental setting. All experiments are conducted on a server with an Intel(R) Xeon(R) CPU E5-2698 v4 @ 2.20GHz CPU and a Tesla V100 GPU.

3.4.1 Vulnerability of the existing method

First, we will empirically how the existing method needs to improve to strengthen our motivation. Simply speaking, the method proposed by Xu et al. [159] first sorts all data points by quick sort, then uses binary search to locate the classification threshold. We show that this method is vulnerable to oracle noises and fails even in simple settings through the following illustrative toy experiments.

For data generation process, we considered the following two cases for $\mathcal{X} = [0,1]$:

- The simplest case that data are drawn uniformly over $[0,1]$. Formally, the underlying distribution is Uniform$(0,1)$.
- A more realistic case that data points mainly concentrate near extreme values, namely 0 and 1. We use the Beta distribution with both parameters set as 0.1 to formalize the underlying distribution, as shown in Figure 3.6.

For both cases, after sampling 100 data points, we use 0.5 as the threshold of likelihood to decide the latent binary label. Formally, $y_i = \mathbb{1}_{p(\mathbf{x}_i)\geq 0.5}$ according to the underlying distribution.

Using similar query budgets, we compared averaged performance of the existing method and the proposed method under various noise rates ranging from 0 (absolutely clean feedback) to 0.4 (almost half of the feedback is wrong).

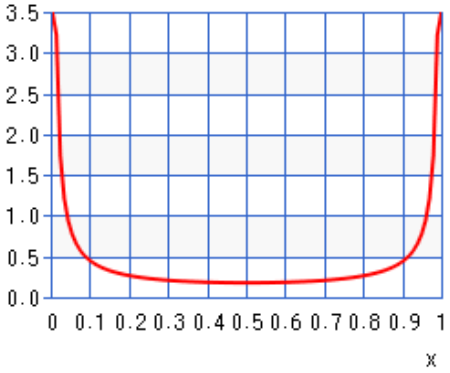


Figure 3.6: Illustration of Beta distribution with both parameters set as 0.1.

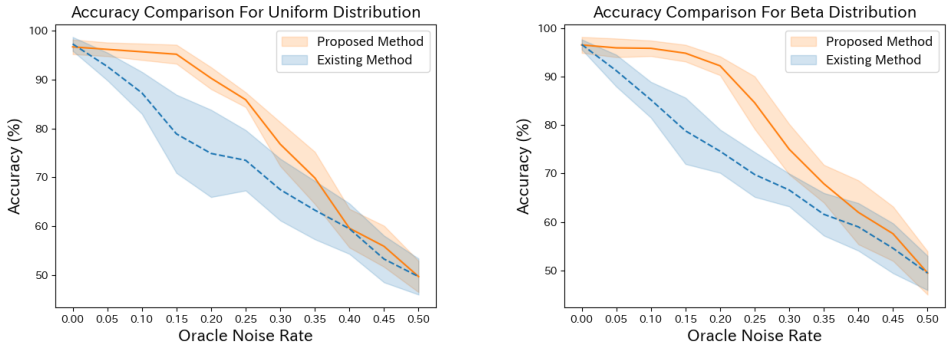


Figure 3.7: Illustrative comparison experiments.

Figure 3.7 clearly shows that the existing method is vulnerable to oracle noises and its performance degrades almost linearly with increasing noise rates. Moreover, the variance is consistently larger than that of the proposed method. We argue that this is mainly because the unnecessary sorting naively treats noisy feedback as if they were clean, which also provide motivation for us to propose the uncertainty comparison oracle to bypass the sorting procedure. Note that the solid lines and the dashed lines represent the mean accuracy for each method respectively. They have no difference in meanings and only serve the purpose for visual distinguishing.

3.4.2 Passive case

We continue to examine using larger datasets and assuming enough annotaion budget in this section.

Dataset selection

We use the following datasets which are common choices from the literature:

- MNIST (Modified National Institute of Standards and Technology database) [95]: A collection of handwritten images. It has ten classes for digits from 0 to 9. It has a training set of $60,000$ images and a test set of $10,000$ images, equally distributed for each class, while each image has 28×28 grayscale pixels. It is a subset of a larger set available from NIST. Due to its well balanced trade-off between data structure complexity and usage

simplicity, it overwhelmingly serves as the starting points for empirical investigation of many machine learning algorithms, and our method is one of them. Although being criticized recently for being too simple thus does not act as sufficient condition for a well-performing algorithm, as a simple nearest neighbour algorithm will achieve over 98% accuracy, it still serves for an initial necessity check step for a valid algorithm.

- Fashion-MNIST [153]: This is a dataset having exactly the same structure as MNIST, namely the number of classes, images per class and image size, but the contents are fashion items, which are T-shirt/top, trouser, pullover, dress, coat, sandal, shirt, sneaker, bag and ankle boot, instead of digits as indicated by its name. It is supposed to sever the role as a drop-in replacement for MNIST for benchmarking machine learning algorithms.

- Kuzushiji-MNIST [39]: This is another drop-in replacement created for MNIST, consisting of handwritten cursive Japanese characters, namely Hiragana, that are collected from historical artefacts. It also shares the exact same data structure as MNIST, with ten classes to be "o", "ki", "su", "tsu", "na", "ha", "ma", "ya", "re" and "wo". This dataset is created by ROIS-DS Center for Open Data in the Humanities (CODH), based on Kuzushiji Dataset created by National Institute of Japanese Literature.

- CIFAR-10 [88]: Provided by CIFAR (Canadian-based global research organization), it consists of 10 classes of totally $60,000$ images each of which has 32×32 RGB pixels. The classes are natural objects or animals, namely airplane, automobile, bird, cat, deer, dog, frog, horse, ship and truck. They are designed to be completely mutually exclusive.

Dataset construction for binary classification

Note that all above datasets are initially designed for multiclass classification. Existing studies on binary classification usually split the whole dataset into two parts according to a heuristic standard, such as separating odd numbers from even numbers for hand-written digits, to construct a binary dataset. However, as we are focusing on uncertainty, it may not be appropriate to simply follow the tradition and it is important to simulate experiments that is capable to raise concern on uncertainty. We note that uncertainty can be expressed as visual similarity for image datasets. Therefore, from each of the above datasets, we constructed two binary classification datasets which we believe share visual similarities and are denoted by the suffix 'a' and 'b':

- **MNIST-a** denotes MNIST images that have the label '1' (7877 images) and '7' (7293 images).

- **MNIST-b** denotes MNIST images that have the label '3' (7141 images) and '5' (6313 images).

- **FMNIST-a** denotes Fashion-MNIST images that have the label 'T-shirt/top' and 'shirt' (each 7000 images).

- **FMNIST-b** denotes Fashion-MNIST images that have the label 'pullover' and 'coat' (each 7000 images).

- **KMNIST-a** denotes Kuzushiji-MNIST images that have the label "ki" and "ma" (each 7000 images).

- **KMNIST-b** denotes Kuzushiji-MNIST images that have the label "na" and "wo" (each 7000 images).
- **CIFAR10-a** denotes CIFAR-10 images that have the label 'automobile' and 'truck' (each 5000 images).
- **CIFAR10-b** denotes CIFAR-10 images that have the label 'deer' and 'horse' (each 5000 images).

Oracle preparation

Before conducting experiments, one issue needs to be addressed is how to properly simulate the comparison oracles in a justified and efficient way. To this end, we learn class-posterior probability regressors for each dataset.

For simple datasets of MNIST, Fashion-MNIST and Kuzushiji-MNIST, we use all 28×28 pixels as a vector input and trained a logistic regression classifier with one hundred thousand maximum iterations. The oracles were then simulated using the output conditional probabilities of this logistic regression classifier. For CIFAR-10, it would be difficult to directly use all image pixels as input. Thus, a ResNet-152 [66] classifier was first trained on the whole dataset ($60,000$ images from 10 classes) for 100 epochs. Then, we extracted the 2048-dimension features before the last fully-connected layer as low-dimension representations, and used them to train a logistic regression classifier. The logistic regression classifier and the k-NN classifiers for the CIFAR-10 case are trained on these 2048-dimension features instead of the original input.

For downstream classification, we set $k = 5$ for k-NN classifiers throughout all experiments. We randomly split training and test set according to the $4 : 1$ ratio for every repetition of the algorithm. Because we do not have sensitive hyper-parameters to tune, we did not prepare a validation set.

Experimental results

We first considered the conservative case where the noise rates are high and the repetition number m is small. If the algorithms perform well in this case, it would perform at least the same in easier settings. Theorem 2 indicates that the size of the delegation subset t can be reasonably restrained to be smaller than 35. Thus, we set t to be two values: an empirical number 10 from rules of thumb and the theoretical maximum 35.

Table 3.1 shows the accuracy mean and standard deviation among 10 trials for each setting. We can observe from it that a larger set of delegation set (corresponding to a higher t) contributes to a better label accuracy, thus accordingly higher generalization capability. This behavior matches the expectation as the inferred label for each non-delegation data point becomes more accurate with a larger t. We also observe that even with a small t, k-NN classifiers can show promising generalization performance.

Then, we consider an easier setting when noise rates are low and more annotation budget is available thus m can be set as a larger value. Table 3.2 shows the results of this optimism situation. We can observe that in this low-noise setting, almost perfect label inference and classification can be achieved even with a small t. Combing with the results above, we can conclude that the proposed algorithm is able to show both high resistance against label noise, but can also achieve high accuracy with moderate noises.

Table 3.1: Performance when the repetition number $m = 1$ and noise rates $\epsilon_{\text{pos}} = \epsilon_{\text{unc}} = 0.4$.

Dataset	Label Accuracy (t=10)	k-NN Test Accuracy (t=10)	Label Accuracy (t=35)	k-NN Test Accuracy (t=35)
MNIST-a	67.89 (0.37)	77.63 (0.83)	80.94 (0.47)	92.36 (0.60)
MNIST-b	67.10 (0.52)	76.11 (0.79)	80.46 (0.37)	92.93 (0.37)
FMNIST-a	65.78 (0.26)	70.96 (0.45)	76.38 (0.20)	81.40 (0.19)
FMNIST-b	66.25 (0.34)	72.28 (0.50)	77.25 (0.24)	83.36 (0.20)
KMNIST-a	68.69 (0.56)	78.90 (1.07)	81.64 (0.62)	94.30 (0.58)
KMNIST-b	67.99 (0.26)	77.45 (0.45)	78.88 (0.36)	90.16 (0.33)
CIFAR10-a	69.34 (0.44)	80.09 (0.82)	82.07 (0.41)	94.28 (0.31)
CIFAR10-b	68.67 (0.20)	78.47 (0.59)	81.83 (0.50)	93.95 (0.42)

Table 3.2: Performance when the repetition number $m = 10$ and noise rate $\epsilon_{\text{pos}} = \epsilon_{\text{unc}} = 0.1$.

Dataset	Label Accuracy (t=10)	k-NN Test Accuracy (t=10)	Label Accuracy (t=35)	k-NN Test Accuracy (t=35)
MNIST-a	99.74 (0.01)	99.39 (0.03)	99.84 (0.01)	99.35 (0.03)
MNIST-b	97.12 (0.03)	98.36 (0.09)	97.22 (0.02)	98.36 (0.06)
FMNIST-a	87.19 (0.06)	83.95 (0.18)	87.38 (0.06)	84.14 (0.16)
FMNIST-b	88.84 (0.04)	86.26 (0.20)	88.86 (0.04)	86.67 (0.18)
KMNIST-a	98.78 (0.01)	99.12 (0.05)	98.90 (0.01)	99.00 (0.02)
KMNIST-b	92.33 (0.03)	94.53 (0.14)	92.36 (0.03)	94.85 (0.09)
CIFAR10-a	99.87 (0.02)	99.92 (0.02)	99.97 (0.01)	99.95 (0.01)
CIFAR10-b	99.86 (0.01)	99.98 (0.01)	99.94 (0.01)	99.98 (0.01)

We further visualize detailed results when conducting more grained ablation on noise levels as well as the size of the delegation set. We conducted experiments on noise rates of $\{0.1, 0.2, 0.3, 0.4\}$ and $t \in \{10, 20, 35\}$. Figure 3.8 shows the detailed investigation on the Fashion MNIST dataset. We show settings with same t number using the same color, and settings with same m number using the same line style. It can be clearly observed that in the simulated environment, the proposed method has a very strong resistance to extremely noisy feedback. Although in the real world, it is difficult to evaluate objective uncertainties, this figure shows the potential of the proposed method on handling unknown comparison noises regarding uncertainties. Note that we set the same noise level for both oracles for simplicity. Thus there is still potential for performance gain in real-world when the noise of positivity comparison is lower.

Results using advanced downstream method

We also confirmed the quality of inferred labels using a more powerful model other than decays old nearest neighbor methods. Co-teaching [63] is a recently proposed training method for noisy labels. It holds two classifiers and each classifier feeds its small loss data points to the other one for training, which is formally shown in Algorithm 3. The percentage of small loss data points can be adjusted according to the current number of epoch. Note that the object model class is not formally restricted to be neural networks, although it may not work very well without the

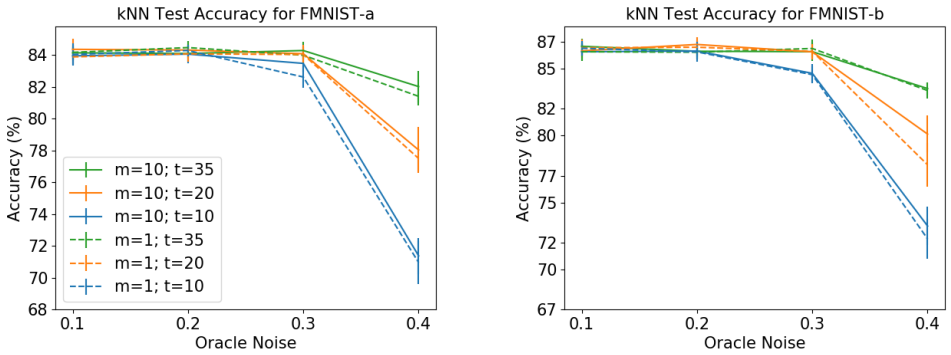


Figure 3.8: Generalization performance of k-NN classifiers for Fashion-MNIST datasets.

significant memorization effect.

Algorithm 3 Co-teaching algorithm [63].

Require: A noisy-labeled training dataset D, two deep neural networks f and g, epoch number T_k and inner-iteration number N.

for $T = 1, 2, \cdots, T_k$ **do**
  Shuffle the training dataset D.
  for $N = 1, 2, \cdots, N$ **do**
    A mini-batch $\tilde{D}$ from D.
    Obtain $\tilde{D}_f$ from $\tilde{D}$ as small loss data points when forwarding using f.
    Obtain $\tilde{D}_g$ from $\tilde{D}$ as small loss data points when forwarding using g.
    Feed $\tilde{D}_g$ to f for parameter update.
    Feed $\tilde{D}_f$ to g for parameter update.
  end for
end for

Although lacking theoretically guarantees, the co-teaching algorithm showed promising performance [63] on benchmark datasets. For experiments using this as the downstream algorithm, we set batchsize as 1024 and epoch number as 100. We adopted the public codes provided by the authors, thus followed all other default settings for other hyper-parameters therein, such as the number of inner-iterations and the learning rate scheduler. In our experiments, we used the relatively small ResNet-18 [66] model and restrained from tuning any hyper-parameters.

Figure 3.9 shows results with same size of delegation set in the same color, and uses dot lines to show results with fewer repetition numbers. We can observe that setting $m = 1$ already shows promising accuracy, when t is set to be the theoretical maximum 35. For the same value of t, increasing m from 1 to 10 can offer only little improvement on the accuracy. Setting m to 1 means we only query each pair once and proceed the algorithm believing the oracle is noiseless thus the answer is correct. This indicates the proposed algorithm is highly robust to oracle noises, as it shows promising performance using the single noisy result without repeating the same query many times. Moreover, the low noise rate regime shows comparable performance under different settings, which means the proposed algorithm can generally achieve high performance with low budget.

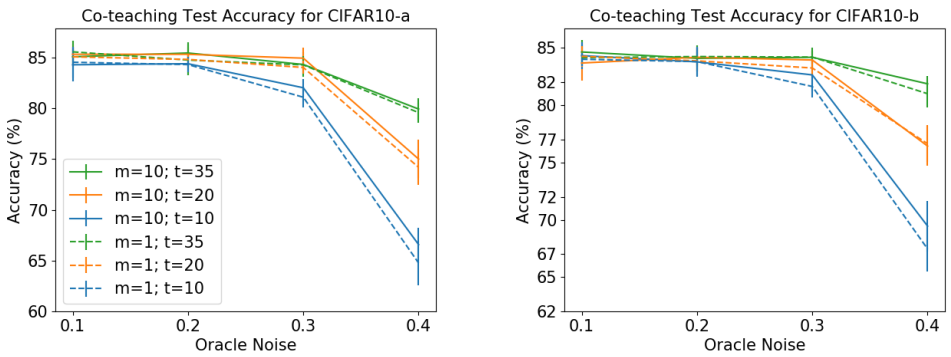


Figure 3.9: Generalization performance of Co-teaching classifiers.

3.4.3 Active case

In this section, we need to take an alternative experimental setting to evaluate Algorithm 2. Specifically, because it needs to loop over every available hypothesis left at the candidate set at each step, it is infeasible to start with a large, let along infinite size, hypotheses set. Note that even for the simplest MNIST dataset with 19×19 features and the simplest linear models, using a discrete exploring space of size 10 for the parameter corresponding to each feature creates a huge hypotheses set of size 10^{784}.

To this end, in order to illustrate the feasibility of Algorithm 2, we resorted to use two Gaussian distributions with mean value of $(2, 2)$ and $(-2, -2)$ and the identity matrix as covariances for both distributions. Then, we choose the first distribution to be $p(\mathbf{x}|y = +1)$ and the second one to be $p(\mathbf{x}|y = -1)$. From the constructed balanced mixture of Gaussian distributions being the underlying data distribution $p(\mathbf{x}) = \frac{1}{2}p(\mathbf{x}|y = +1) + \frac{1}{2}p(\mathbf{x}|y = -1)$, we drew $10,000$ data points in total to compose the dataset. Then, a logistic regression classifier is sufficiently trained to simulate the oracles, in a similar way to the previous section. For the initial hypothesis set, we used $1,000$ equally separated linear classifiers passing through the origin point. Setting the desiring precision $\epsilon = 0.1$ resulted three steps based on Algorithm 2.

Table 3.3 shows the number of left candidate hypotheses and their test accuracy at each step. We can observe that the size of the hypothesis shrinks and the test accuracy of the left hypotheses becomes better with increasing average and decreasing variance.

Table 3.3: Insufficient budget experiment results.

	Step 1	Step 2	Step 3
Number of Left Hypotheses $\lvert H_i \rvert$	674.10 ($\pm$4.97)	525.60 ($\pm$7.34)	196.90 ($\pm$71.85)
Test Accuracy of Left Hypotheses H_i	96.98% ($\pm$0.44%)	99.29% ($\pm$0.19%)	99.78% ($\pm$0.11%)

3.5 User study

The previous section investigated the proposed algorithm using artificial oracles, and the feasibility in real-world situations remains untouched. In this section, we conducted user study using crowdsourcing.

The goal of user study is two-fold:

- To justify the proposed uncertainty comparison oracle and explore the difficulty of conducting uncertainty comparisons for users. Whether users have subjective preference over the proposed method?
- To confirm the performance of algorithms on actual feedback collected through crowdsourcing. Even if users prefer the proposed oracle, their feedback could be noisy, similar to existing oracles. Then, how potentially noisy feedback of the proposed oracle affect the algorithm performance?

Through out this thesis, we used the Lancers platform [1] to conduct user studies, which is the largest crowdsourcing platform in Japan. Depend on the number of questions to ask, we split the questionnaire for each experimental setting into separate crowdsourcing tasks with proper sizes ranging from 35 to 55 so that it is feasible to be posted on the platform. This is possible because we recruited more than 50 annotators for each task, thus the feedback can be considered to be homogeneous among separations. In general, the payment is set as 10 yen per feedback, such as a selection made by the annotator. Since it is a platform mainly based in Japan, all instructions and questions of our user studies are in Japanese and only annotators fluent in Japanese are recruited.

In the following, we introduce the two datasets and the general interface we used, followed by detailed description of the user study setting.

3.5.1 User study using the Kuzushiji-MNSIT dataset

We consider to continue using the cursive Japanese dataset, which would be important for advocating research on historical Japanese books and documents.

From the Kuzushiji-MNIST dataset [40], we selected the 5-th and the 10-th characters to form the binary classification task. The reading alphabet is 'NA' for the 5-th character and 'WO' for the 10-th character. Figure 3.10 shows them in a standard font. Albeit the visual similarity, these two characters are important auxiliary words with distinct meanings. Thus, wrongly recognizing the two characters can harm the understanding of the sentence. This recognition task has a natural affinity with ambiguity comparison, as in daily writing, the difficulty of recognizing a hand written character is easier to interpret, rather than recognizing the exact character.

Figure 3.10: Sample images for 'NA' in the left and 'WO' in the right.

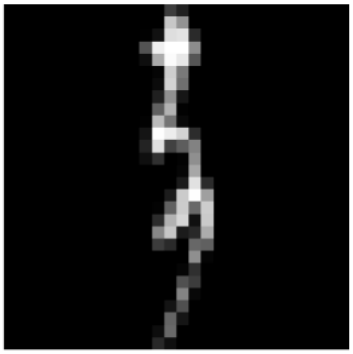
Methods and query interface We prepared three types of questions: explicit labeling, pairwise positivity comparison, and pairwise uncertainty comparison. We also asked annotators for the difficulty of each question when necessary. The detailed interfaces for explicit labeling, positivity comparisons and uncertainty comparisons are shown in Figure 3.11, Figure 3.12, and Figure 3.13 respectively.

[1] https://www.lancers.jp/

Task Description

Hand written character images of 'NA' or 'WO' are to be shown.
Please answer questions on their recoginition.

Question1: Please answer the exact character.



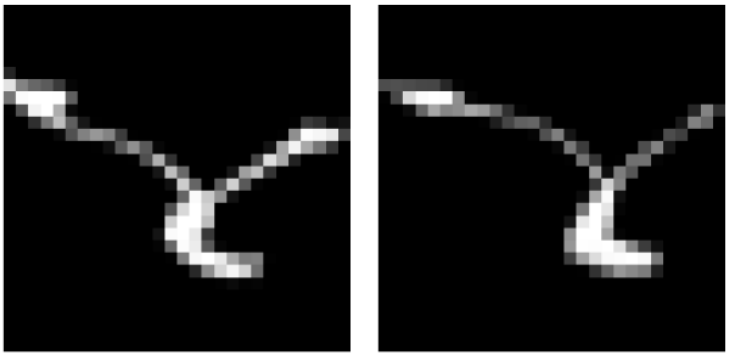
○ 'NA' ○ 'WO'

Figure 3.11: Interface for explicit labeling.

Task Description

Hand written character images of 'NA' or 'WO' are to be shown.
Please answer questions on their recoginition.
Please annotate the one that looks more like 'NA',
or the one that looks like more like 'WO'.
Please also answer the criterion of the selection.

Question1:



The one looks more like ○ 'NA' ○ 'WO' is

○ image A ○ image B.

Figure 3.12: Interface for positivity comparison.

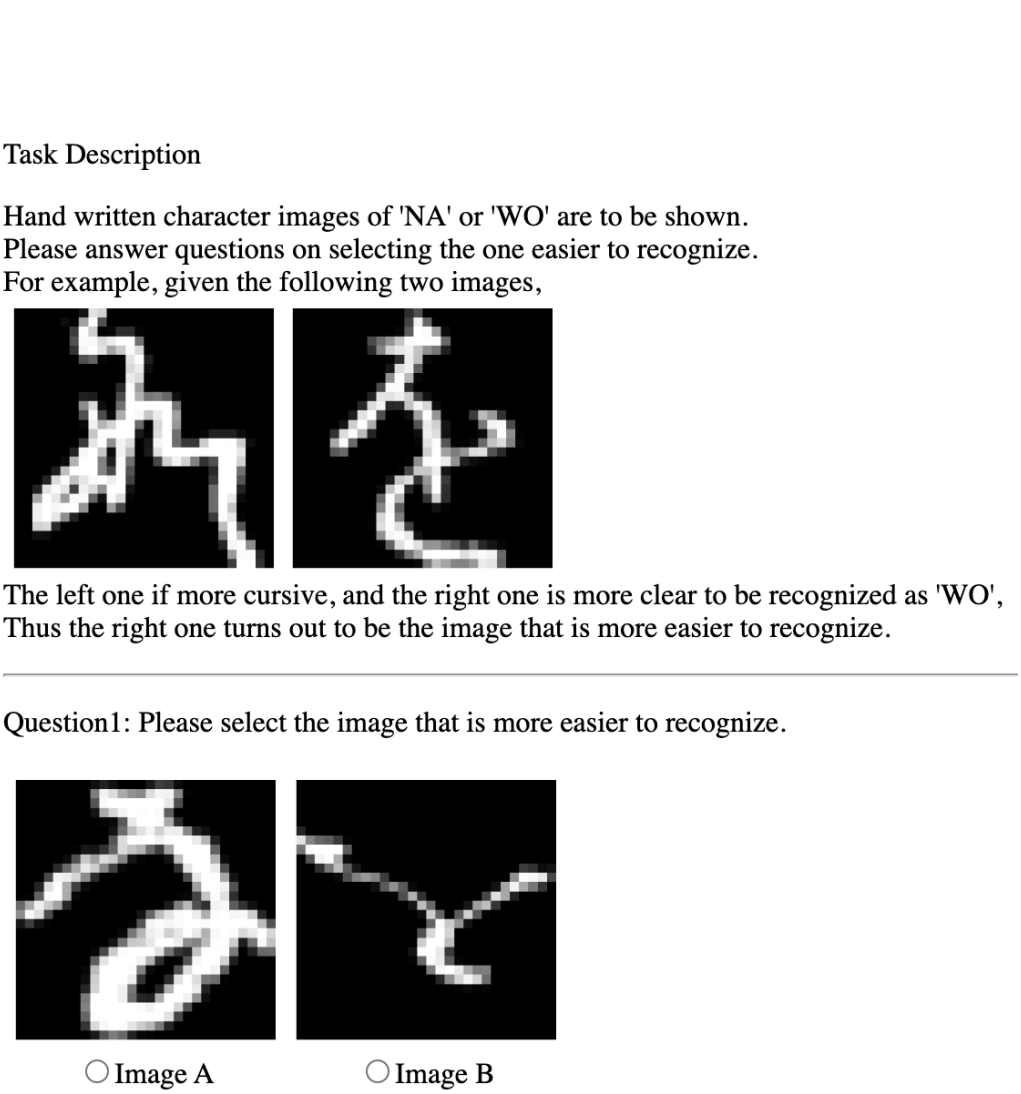


Figure 3.13: Interface for uncertainty comparison.

On the Lancers platform, our user profile is set to have *mass media* as the type of industry, which is public to all annotators. This is to say, for annotators who are looking for questionnaire answering tasks on the Lancers platform, if they are interested in mass media, such as image, video, music, etc., they can search by filtering the type of industry of users who are posting tasks and end up discovering our tasks. In this sense, our user profile can be seen as a kind of advertisement aiming at annotators who are looking for tasks related to mass media.

The English translation of description of the task is:

> Please answer the following questions to judge the kuzushiji of hiragana. Please also judge the difficulty level of each question, and finally to complete a questionnaire regarding the comparison of each question types.

For positivity comparisons, if we fix one label such as 'NA' and ask which one is more likely to be 'NA', there are cases that both images in a pair look similar to 'WO', thus it's difficult to answer. Therefore, we also ask annotators to choose either 'NA' or 'WO' that is used as the criterion of positivity.

For uncertainty comparisons, as this is a newly proposed comparison question and annotators may be not used to answer it, we give an explanatory example on how to select.

Justification for uncertainty comparisons From the results of Section 3.4.2, we know the proposed algorithm is robust to feedback noise. Thus, we want to confirm whether user feedback shows high noise on this binary classification task, thus meaning this task is suitable for testing the proposed algorithm. In this user study, we first uniformly selected 15 data points. Then, we ran the proposed algorithm on these 15 data points using artificial oracles, and collected the 21 pairs that were selected for uncertainty comparison. Finally, we conducted user study on explicit labeling and uncertainty comparison on these 21 pairs. At the end of the questionnaire, we also asked the difficulty of each annotation type using scores from one to five, with a *smaller* score indicating an *easier* question.

We collected answers from 10 annotators and calculated the mean score with standard deviation. The difficulty was $3.85\,(\pm 0.93)$ for explicit labeling and $3.10\,(\pm 1.21)$ for uncertainty comparison. There were 5 annotators who answered uncertainty comparison is easier, and 4 annotators who answered two types of query have same difficulty. From these results, we conclude that for the pairs selected by the proposed algorithm, uncertainty comparison turned out to be a easier query from than explicit labeling both data points.

Algorithm feasibility In this user study, we confirmed the performance of each algorithm on feedback collected through crowdsourcing. We first greedily selected 25 medoids [132], then collected answers for all possible combinations among these medoids (300 distinct pairs) from 10 annotators, and used aggregated majority as input to the proposed algorithm as well as the existing algorithm [159] using both positivity comparisons and explicit labeling. Considering the limited number of available training data, we adopted a pre-trained neural network and fine-tuned its last layer. When using all 25 medoids for training, the test accuracies were 78% for the proposed algorithm and 68% for the existing algorithm [159]. Compared to fully supervised learning using *all explicit class labels* which achieved a test accuracy of 81%, we conclude that the proposed algorithm can show competitive performance to the best possible accuracy.

Figure 3.14 shows test accuracies of running the algorithms on selected medoids. The test accuracy measures the performance of each classifier learnt from inferred labels on a holdout test set of size 100, which is uniformly selected without replacement after the selection of training data points. The mean value from results of 10 annotators are shown in dashed lines and the standard deviation are shown by the shadow. The value from aggregated results are shown in solid lines. The proposed algorithm shows much better performance than the existing algorithm. We note the existing algorithm shows a peak performance when using 10 medoids as training data. This could be explained that 10 happens to be a proper number for selecting medoids which are visually different, thus easier to sort accurately. However, it shows higher overall variances and is unstable for other number of training data points.

When increasing the number of training data points, we observed the proposed algorithm could also show stable and promising generalization ability competitive to full supervision. However, the performance of the existing algorithm [159] was not stable, because it separated data points into small bags for binary search of the classification threshold, and queried a random subset of each bag for explicit class labels. With fewer training data, the size of each bag was small and it could query most of a bag for explicit class labels, thus achieved high labeling accuracy. However, with more training data, a reasonable budget restrained the size of the subset from each bag for querying explicit class labels, thus resulting

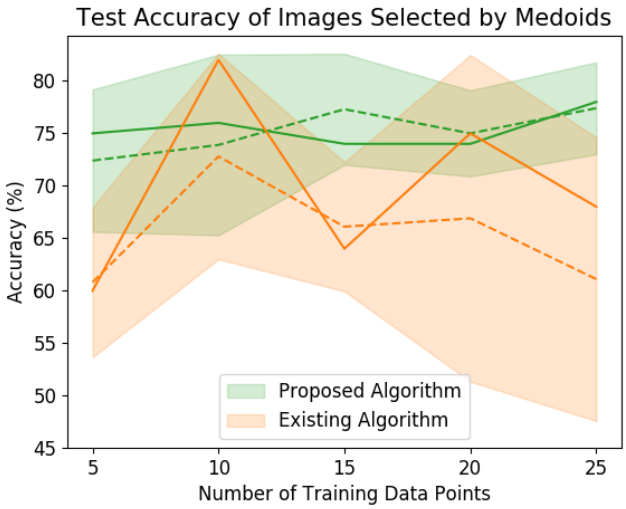


Figure 3.14: Test Accuracy with respect to the number of training data points.

performance drop.

Furthermore, in order to discriminate different types of pairs, we introduce two types of difficulty:

1. *Individual difficulty* indicating the difficulty on assigning the explicit label for a single data point.

2. *Pair difficulty* indicating the difficulty on answering the comparison result for a pair of data points.

Based on the user evaluation of *individual difficulties*, we can classify data pairs into three types:

1. The 'E' type containing two easy data points.

2. The '&' type containing one easy and one difficult data point.

3. The 'D' type containing two difficult data points.

Then, we investigate the relationship between pair types and pair preferences. Figure 3.15 shows the histograms of actually queried uncertainty comparisons, indicating *users preferring uncertainty comparisons* by blue and *users preferring explicit labeling* by orange. We observe that for the 'E' type and especially the '&' type of pairs, uncertainty comparison is overwhelmingly preferred. These types of pairs are important for separating highly uncertain data points from others. On the other hand, it may draw concerns as explicit labeling being majorly preferred for the 'D' type of pairs, and uncertainty comparison for these pairs will result low accuracy. However, as the important step of the proposed algorithm is to separate data points with different uncertainties, the result is robust to noisy annotations for pairs with similar uncertainties.

3.5.2 User study using the Clickbait dataset

In this study, we consider the clickbait recognition task. Specifically, we focuse on the classification of clickbait titles [31], which is important for improving user experience on various web-based service by saving their attention and time. This classification task has a natural affinity with uncertainty comparison. In

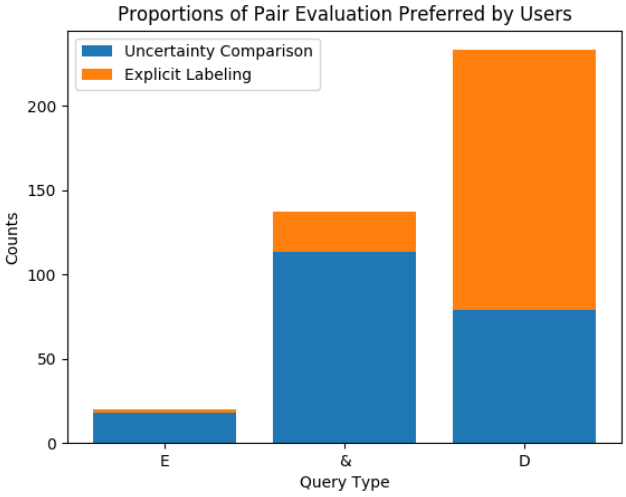


Figure 3.15: Histogram of queried data pairs.

daily interaction with the Internet, it is usually difficult for normal people to distinguish, thus many are often alluded to click on clickbait titles. However, as we found through crowdsourcing feedback, it is easier to answer the suspiciousness level or to compare two titles.

We used the Clickbait dataset [31] which is a binary classification dataset that contains around $60,000$ titles collected from online news media. We uniformly selected 150 titles from each class for this user study. We used 200 titles for crowdsourcing and 100 titles for evaluation. We used a pretrained BERT model[2] [43] to extract 768-dimension features.

Offline evaluation method As the total number of pairwise comparisons among 200 data points would be as large as $10,000$, we used the following way to reduce the crowdsourcing cost down to a reasonable scale. Instead of collecting all necessary pairwise feedback, we collect explicit labels for all titles, as long as their difficulty, or ambiguity, for the labeling. The 2-stage difficulty evaluation takes the form of a binary selection: *confident* or *ambiguous*. We collected 50 feedback for each title from 50 users and used their majority votes. We then use difficulty evaluations to simulate necessary pairwise feedback for both forms of feedback. Note that comparison feedback is proposed for situations where individual feedback is in reliable and this experiment simulates comparison feedback from individual feedback, thus looks like violating the motivation. However, the goal of this experiment is to show the feasibility of the proposed algorithm to improve classification performance at a large margin. It can be said that for some tasks in the real world where individual feedback is indeed hard to collect, the proposed algorithm can serve a perfect job of improving classification performance.

On the Lancers platform, our user profile is still set to have the same type of industry of mass media. The English translation of description of the task is:

> You will be asked to answer questions regarding the identification of fishing article titles. At the end of the questionnaire, you will be

[2]The BERT-Base uncased model with 12 layers, 768 hidden dimensions, 12 head and 110M parameters.

asked to answer your approach in the judgment in a simple, free-text format.

Figure 3.16 shows the English translation of the detailed user interface:

- 'Question 1' is used for collecting explicit labeling and difficulty evaluation.
- 'Question 2' is used for collecting positivity comparison feedback.
- 'Question 3' is used for collecting uncertainty comparison feedback.
- 'Question 4' is used for collecting difficulty evaluation of a pair of titles.

All titles are shown together with a translation in annotators' native language[3].

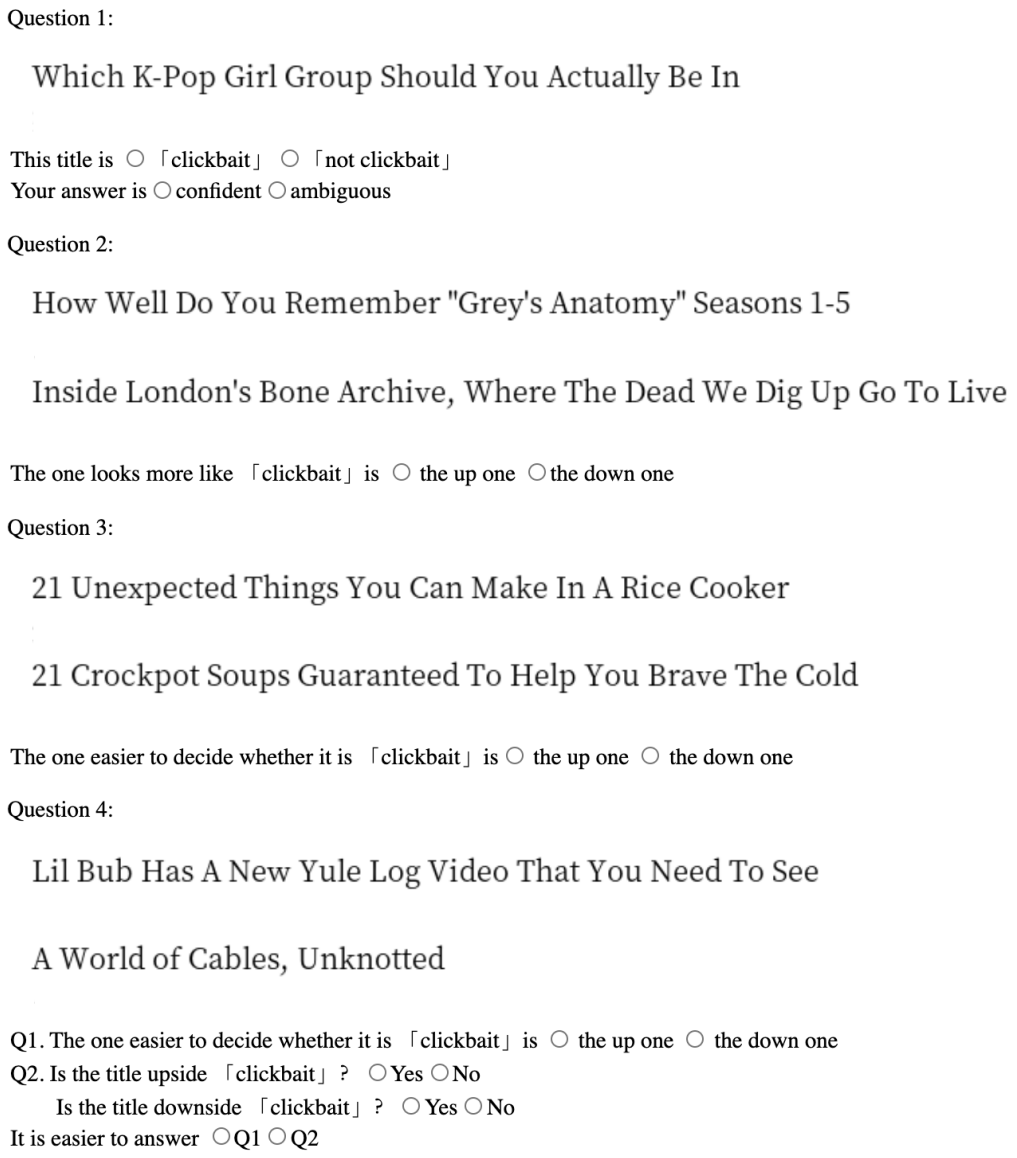


Figure 3.16: Screenshots of sample questions.

[3]The translation is based on the results of DeepL (https://www.deepl.com/translator).

Justification for offline evaluation We evaluate the accuracy of simulated pairwise comparison feedback by comparing with actually queried feedback of a selected part. From $10,000$ available pairs of data points, we sampled 100 pairs. We collected actual pairwise comparison feedback on these pairs: positivity comparisons of 50 pairs and uncertainty comparisons for the other 50 pairs. We collected from 50 users aggregated using majority votes. We used 2-stage difficulty evaluation to simulate pairwise comparison feedback and accuracies with respect to collected comparison feedback were 91.36% ($\pm$12.51%) and 81.52% ($\pm$16.13%), respectively. Considering the performance in Section 3.4.2, 2-stage difficulty evaluation is feasible for simulation on unobserved pairwise comparison.

Remark on the number of confident levels This step also played a crucial role on deciding the user interface. At first, a more grained four-stage confidence query interface is tested. That is to say, the options for answering confidence is *confident*, *somehow confident*, *somehow ambiguous* and *ambiguous*. When we used the information of all 4 stages to conduct simulation and compared with actual feedback, the simulation accuracies for positivity comparison and uncertainty comparison were 89.12%($\pm$13.36%) and 68.28%($\pm$17.22%), respectively. The low consistency may result from noisy difficulty evaluations and it is not ideal for simulation. Thus, we decided to simply the query interface to aggregate feedback of *confident* and *somehow confident* to be one stage, and other two to be another stage.

Algorithm feasibility We confirmed the performance of each algorithm on crowdsourcing feedback. We compared test accuracy which measures generalization ability of a classifier trained on crowdsourced or inferred labels. Table 3.4 indicates the proposed algorithm shows better generalization ability than others. The individual annotation accuracy is 58.38%($\pm$8.12%) and the aggregated majority vote annotation accuracy is 61%.

Table 3.4: User study results.

	Explicit Labeling	Existing Method [159]	Proposed Method
Test Accuracy	60%	60%	67%

Preference of the proposed oracle on queried pairs We examine the motivation and potential of the proposed uncertainty comparison oracle. In order to compare its mental cost with the explicit labeling oracle, we collected preference from 50 users on pairs actually queried by the proposed algorithm in this user study. We selected first 50 pairs from total 199 queried pairs because the algorithm largely depends on the first selected high uncertainty data points. For each of selected 50 pairs, we newly conducted a user study that asked users to first answer uncertainty comparison and explicit labeling, and then answer which form of feedback is more preferred.

Furthermore, we also investigate the relationship between pair types and pair preferences.

Figure 3.17 shows the histogram of the selected 50 pairs that were first queried by the uncertainty comparison oracle during the execution of the proposed algorithm. The preference is aggregated by majority votes, thus resulting a total count of 50. We can observe that most of the queried pairs belong to 'E' type

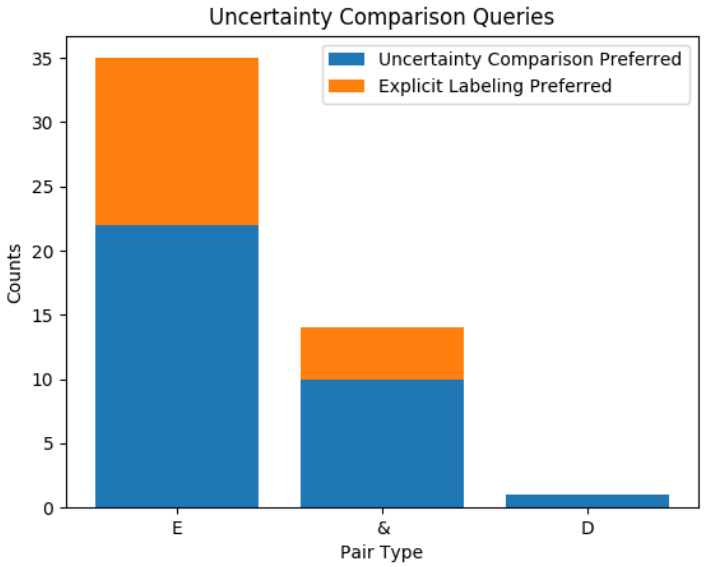


Figure 3.17: Histogram of queried data pairs.

and '&' type. The proposed algorithm did not query many 'D' type pairs at the beginning of its execution because it was working on selecting difficult (or uncertain) data points from easy ones.

From the figure, we can observe that for all pair types, the proposed uncertainty comparison oracle is more preferred, even for the 'E' type . For '&' pairs, uncertainty comparison is preferred because the user can simply pick the difficult one, without deciding its label. Although few are collected, it is the same for 'D' pairs. This indicates that for some application scenes, the proposed oracle is more preferred by users and the resulting algorithm is able to show better performance.

3.5.3 User opinions

At the end of each questionnaire, we also asked annotators to answer their opinions on these tasks in free text. We select some of representative opinions and list their English translation.

The following list shows advantages of positivity comparisons over explicit labeling.

- It is easy to choose between "NA" or "WO" even if you can't read the word.
- You can choose the one you can easily recognize.
- You can choose the letters by your feeling.
- Unlike direct judgments, there is no clear correct answer, so it is possible to create questions that are easy for anyone to answer.
- When it's not too curled up, it's easy to choose.

The following list shows disadvantages of positivity comparisons over explicit labeling.

- If you cannot read either of them, your selection criteria will be blurred.

- It is hard to judge a flaw when it's curled up.
- It is not sure if the decision is accurate.
- You need to stop and compare both images carefully, and may feel a great sense of hesitation before making a decision.
- Unlike direct judgement, there is no clear correct answer, and if neither letter is difficult to judge, you don't have to think about the answer. You can make a good choice.

The following list shows advantages of uncertainty comparisons over explicit labeling.

- It's easy to choose if you can read one or the other somehow.
- It's quick and intuitive and I understand it quickly.
- Can be narrowed down if both are recognized as "NA" or "WO".
- It's easy to imagine how easy it is to read by just the simple criterion of being able to read, and how easy it is to read by pronouncing it in your head.
- It is highly flexible and does not have any restrictions.

The following list shows disadvantages of uncertainty comparisons over explicit labeling.

- You can only seem to read them, but you can't tell whether you actually chose the correct answer or not.
- I don't know if other people can quickly recognize.
- If the words are not read as "NA" or "WO", I use the elimination method to select.
- When neither of them is likely to be readable, I tend to choose them at random.
- Unlike direct judgments, there is no clear correct answer, which makes it difficult to evaluate the competence of the annotator.

As we can see from above lists, it is difficult to choose when both images in a pair are not recognizable. This may affect the performance of the method dealing with explicit labels, as it assumes the labels are clean, and it may also affect the performance the existing method, as it is required to sort the whole dataset. However, this does not significantly downgrade the performance of the proposed algorithm, as either one in the pair satisfying the desired uncertainty. Moreover, it is interesting to see the various criterion used by annotators.

3.5.4 On non-ideal datasets

In this section, we discuss the characteristics of uncertainty comparison using other datasets we tested in user studies but did not show significant performance. On the Lancers platform, our user profile is still set to have the same type of industry of mass media and the description is similar to previous settings in this section.

We first conducted user experiments using face synthesis data, also known as DeepFake. The goal is to test the mental burden difference of explicit feedback and comparison feedback. However, we found that the annotation accuracy is as high as around 80%, and errors of collected annotation labels of explicit feedback are mostly false positive. That is to say, most of the errors are wrongly assign the 'real' label to a actually synthesized 'fake' data point, either a piece of video without sound or an image. Considering the high accuracy and imbalance of explicit feedback error, we restrained from conducting more experiments using these data.

We also conducted experiments on learning user preferences using a car dataset [87]. Here, we pay attention to preference learning instead of an objective classification task because the positivity comparison feedback has been extensively used in this field. However, when working on aggregated preference, we found that the explicit feedback is reported to be easier to answer than uncertainty comparison feedback since it concerns only personal preference. When working on individual preferences, the performance largely depends on each annotator. In addition, we also used a dataset consists of movie information for user study on preference learning. However, we found it is hard to design an intuitive question on uncertainty comparison for the task.

Inspired by the progress of user studies using the Kuzushiji dataset, we tested the MNIST version of deep fake. In order to improve the user experience, we concatenated multiple digits of same class in a row as a single data point. However, it is found to be too difficult for all forms of feedback and the annotation accuracies were barely over 50%.

In order to test on tasks that have a more ambiguous classification threshold than preference, we also conducted on satellite image classification of 'industrial land' and 'residential land', where normal annotators may lack professional knowledge required by the task. However, it turns out this dataset is easy for all forms of feedback. In additional, uncertainty comparison on the cat species classification task failed to provide significant performance improvement over existing methods.

Considering datasets used in previous sections, we conclude the following findings concerning the uncertainty comparison feedback:

- The question sentence needs to be easy to interpret.
- Data needs to have features that are easy to interpret, such as the readability of Kuzushiji or the intepretability of post titles.
- The task should not require too much professional knowledge which will results to be too difficult; it also should not be too easy so that learning from explicit labels will work well.
- The task should be designed for a objective standard instead of a subjective standard, such as preference.

Furthermore, during the above exploration process, we also found that refining the way of asking the same question can also improve the annotation results.

3.6 Conclusion

In this chapter, we address the problem of active classification using positivity comparison queries and propose a novel uncertainty comparison oracle, followed

by a noise-tolerant theoretical-guaranteed label inference algorithm. We then confirm the performance of the algorithm theoretically and empirically.

We believe this research will benefit researchers in all fields who are seeking for a more effective and less laborious annotation method for their unlabeled datasets. It can foster applications of machine learning by lowering the annotation barrier for people without specific professional knowledge. On the other hand, it can also benefit domain experts with professional knowledge by saving their time for more important tasks. Furthermore, collecting comparison information can potentially mitigate annotation biases of explicit labeling. In this sense, it can also serve the aim of protecting privacy by not querying the explicit class labels in some cases. However, for the negative side, it may harm the performance of downstream classification models when the comparison annotation is mostly incorrect.

Chapter 4

Learning from Triplet Comparisons

4.1 Introduction

As shown in Section 2.3 and Section 2.4, there are situations when only working with pairwise comparisons reaches performance limit and triplet comparisons is introduced for further performance improvement. Recent work on learning from triplet comparison feedback data has received increasing attention [67, 83]. It is usually argued in a similar way that humans perform better in the task of evaluating which instances are similar, rather than identifying each individual instance [142]. More importantly, it is also believed that humans can achieve much better and more reliable performance on assessing the similarity on a relative scale, such as "Instance A is more similar to instance B than to instance C", rather than on an absolute scale, such as "The similarity score between A and B is 0.9 while the one between A and C is 0.4", by Kleindessner et al. [83]. Collecting data in this manner has the advantage of avoiding the problem caused by individuals' different assessment scales. On the other hand, the collected absolute similarity scores may only provide information on a comparison level in some applications, e.g., sensor localization [97]. It was shown that keeping only the relative comparison information can help an algorithm be resilient against measurement errors and achieve high accuracy [154].

In this chapter, we focus on the problem of learning from triplet comparison data, which is a common form of comparison feedback data. Formally, a triplet comparison $(\mathbf{x}_a, \mathbf{x}_b, \mathbf{x}_c)$ contains the information that instance $\mathbf{x}_a$ is more similar to $\mathbf{x}_b$ than to $\mathbf{x}_c$. As one example, search-engine query logs can readily provide feedback in the form of triplet comparisons [133]. Given a list of website links $\{A, B, C\}$ for a query, if links A and B are clicked and the link C is not clicked, we can formulate a triplet comparison as (A, B, C). We can also collect unlabeled datasets first and collect triplet comparison afterwards, such as the instrument dataset [110] and the car dataset [83]. Note that data are collected in a totally unlabeled way in these applications.

As mentioned before, learning from triplet comparison data was initially studied in the context of metric learning [133], in which a consistent distance metric between two instances is assumed to be learned from data. The well-known triplet loss for face recognition was proposed in this line of research [131, 161]. Using this loss function, an inductive mapping function can be efficiently learned from triplet comparison image data.

At the same time, the problem of ordinal embedding has also been extensively studied [1, 147]. It aims to learn an embedding of the given instances to the Euclidean space that preserves the order given by the data. Algorithms for large scale ordinal embedding have been developed [3]. In addition, many other

problem settings have been considered for the situation of using only triplet comparison data, such as nearest neighbor search [62], kernel function construction [84] and outlier identification [85].

However, learning a binary classifier from triplet comparison data alone remained untouched until recently. A random forest construction algorithm [61] was proposed for both classification and regression. However, it requires a initially labeled dataset and needs to *actively* access a triplet comparison oracle many times. For *passively* collected triplet comparison data, a boosting based algorithm [122] was recently proposed without accessing a triplet comparison oracle. However, a set of labeled data is still indispensable to initiating the training process. To the best of our knowledge, method presented in this chapter is the first to tackle the problem of learning a classifier *only* from *passively* obtained triplet comparison data, without accessing either a labeled dataset or an oracle.

4.1.1 Organization

We show that we can successfully learn binary classifiers from only passively obtained triplet comparison data. We achieve this goal by developing a novel method for learning binary classifiers in this setting with theoretical justification. We use the direct risk minimization framework given for the classification problem. We then show that the classification risk can be empirically estimated in an unbiased way given *only* triplet comparison data. On the theoretical perspective, we establish an estimation error bound for the proposed empirical risk minimizer, showing that learning from triplet comparison data is consistent. Note that our method returns an *inductive* model, which is different from clustering and ordinal embedding, and can be applied to unseen test data points. The test data would consist of single instances instead of triplet comparisons since our primitive goal is to perform a binary classification task on unseen data points.

In following sections, we first review the ordinary fully supervised classification setting. Then, we introduce the formal problem setting and assumptions for the data generation process of triplet comparison data. Finally, we describe the proposed method for training binary classifiers from only passively obtained triplet comparison data.

4.2 Generation process of triplet comparison data

We first formulate the underlying generation process of triplet comparison data in order to perform empirical risk minimization.

Label possibilities for triplet comparison data

We assume that three samples in a triplet $(\mathbf{x}_a, \mathbf{x}_b, \mathbf{x}_c)$ are first generated independently, then shown to an annotator. The annotator can then mark the triplet to be proper / correct or not. Here, we denote the similarity between two samples $\mathbf{x}_a$ and $\mathbf{x}_b$ as σ_{ab}: the larger σ_{ab} is, the more similar two samples are. Then, a proper / correct triplet means $\sigma_{ab} \geq \sigma_{ac}$. Specifically, it means that the three labels (y_a, y_b, y_c) in a triplet appear to fall in one of the following cases:

$$\begin{aligned}\mathcal{Y}_1 \triangleq \{&(+1,+1,-1),(-1,-1,+1),(+1,+1,+1),\\ &(-1,-1,-1),(+1,-1,-1),(-1,+1,+1)\}.\end{aligned} \tag{4.1}$$

Otherwise, it means the first data point $\mathbf{x}_a$ is more similar to the third one $\mathbf{x}_c$ than to the second one $\mathbf{x}_b$. Therefore, the annotator will choose to mark the

triplet as not proper / incorrect. Similarly, this means (y_a, y_b, y_c) appears to fall in one of the following cases

$$\mathcal{Y}_2 \triangleq \{(+1, -1, +1), (-1, +1, -1)\}. \tag{4.2}$$

Assumptions on the generation process

Here, we describe what we assume on how a triplet is drawn from the underlying data distribution.

First, three data points are generated independently from the underlying joint density $p(x, y)$, then an initial unlabeled dataset $\mathcal{D} = \{(\mathbf{x}_a, \mathbf{x}_b, \mathbf{x}_c)\}$ are collected without knowing the underlying true labels (y_a, y_b, y_c). However, we assume we can then collect information about which case a triplet belongs to from annotators feedback. Notice that in the present problem setting, we assume annotators will always give rational feedback. This means annotators never recognizes samples with different labels to be more similar to each other. We believe this is a reasonable assumption as triplet comparison is usually answered with high accuracy, which will be shown later in this chapter. After receiving feedback from annotators, we can actually obtain the following two distinct datasets. The data the user chooses to keep the order, namely the set of triplets marked as proper / correct, is denoted as

$$\mathcal{D}_1 \triangleq \{(\mathbf{x}_a, \mathbf{x}_b, \mathbf{x}_c) | (y_a, y_b, y_c) \in \mathcal{Y}_1\}. \tag{4.3}$$

Similarly, the data the user chooses to flip the order, namely the set of triplets marked as not proper / incorrect is denoted as

$$\mathcal{D}_2 \triangleq \{(\mathbf{x}_a, \mathbf{x}_b, \mathbf{x}_c) | (y_a, y_b, y_c) \in \mathcal{Y}_2\}. \tag{4.4}$$

Note that the ratio of $n_1 \triangleq |\mathcal{D}_1|$ to $n_2 \triangleq |\mathcal{D}_2|$ is fixed because we assume the three samples in a triplet are generated independently from $p(x, y)$. Therefore, the ratio $\frac{n_1}{n_2}$ is only dependent on the underlying class prior probabilities, which are assumed to be fixed unknown values.

Although being collected from an original same set and separated based on annotator feedback, the two datasets $\mathcal{D}_1$ and $\mathcal{D}_2$ can be alternatively considered to be generated from two underlying distributions as indicated by the following lemma.

Lemma 5. *Corresponding to the data generation process described above, let*

$$\begin{aligned} p_1(\mathbf{x}_a, \mathbf{x}_b, \mathbf{x}_c) &= \frac{p(\mathbf{x}_a, \mathbf{x}_b, \mathbf{x}_c, (y_a, y_b, y_c) \in \mathcal{Y}_1)}{\pi_{\mathrm{T}}}, \\ p_2(\mathbf{x}_a, \mathbf{x}_b, \mathbf{x}_c) &= \pi_+ p_+(\mathbf{x}_a) p_-(\mathbf{x}_b) p_+(\mathbf{x}_c) + \pi_- p_-(\mathbf{x}_a) p_+(\mathbf{x}_b) p_-(\mathbf{x}_c), \end{aligned} \tag{4.5}$$

where $\pi_{\mathrm{T}} \triangleq 1 - \pi_+ \pi_-$, $\pi_+ \triangleq p(y = +1)$ *and* $\pi_- \triangleq p(y = -1)$ *are the class prior probabilities that satisfy* $\pi_+ + \pi_- = 1$; $p_+(x) \triangleq p(x|y = +1)$ *and* $p_-(x) \triangleq p(x|y = -1)$ *are class conditional probabilities. Then, it holds that*

$$\begin{aligned} \mathcal{D}_1 &= \{(\mathbf{x}_{1,a}, \mathbf{x}_{1,b}, \mathbf{x}_{1,c})\}_{i=1}^{n_1} \overset{\text{i.i.d.}}{\sim} p_1(\mathbf{x}_a, \mathbf{x}_b, \mathbf{x}_c), \\ \mathcal{D}_2 &= \{(\mathbf{x}_{2,a}, \mathbf{x}_{2,b}, \mathbf{x}_{2,c})\}_{i=1}^{n_2} \overset{\text{i.i.d.}}{\sim} p_2(\mathbf{x}_a, \mathbf{x}_b, \mathbf{x}_c). \end{aligned} \tag{4.6}$$

Detailed derivation is given in Section 5.4.

In order to prepare for later investigation, we then denote the above pointwise data collected from $\mathcal{D}_1$ and $\mathcal{D}_2$ by ignoring the triplet comparison relation as

$$\begin{aligned}\mathcal{D}_{1,a} &\triangleq \{\mathbf{x}_{1,a}\}_{i=1}^{n_1},\ \mathcal{D}_{1,b} \triangleq \{\mathbf{x}_{1,b}\}_{i=1}^{n_1},\ \mathcal{D}_{1,c} \triangleq \{\mathbf{x}_{1,c}\}_{i=1}^{n_1},\\ \mathcal{D}_{2,a} &\triangleq \{\mathbf{x}_{2,a}\}_{i=1}^{n_2},\ \mathcal{D}_{2,b} \triangleq \{\mathbf{x}_{2,b}\}_{i=1}^{n_2},\ \mathcal{D}_{2,c} \triangleq \{\mathbf{x}_{2,c}\}_{i=1}^{n_2}.\end{aligned} \tag{4.7}$$

We then express the marginal densities of above pointwise data by the following theorem.

Theorem 6. *Samples in $\mathcal{D}_{1,a}$, $\mathcal{D}_{1,c}$, $\mathcal{D}_{2,a}$ and $\mathcal{D}_{2,c}$ can be considered to be independently drawn from*

$$\tilde{p}_1(\mathbf{x}) = \pi_+ p_+(\mathbf{x}) + \pi_- p_-(\mathbf{x}), \tag{4.8}$$

samples in $\mathcal{D}_{1,b}$ can be considered to be independently drawn from

$$\tilde{p}_2(\mathbf{x}) = \frac{(\pi_+^3 + 2\pi_+^2\pi_-)p_+(\mathbf{x}) + (2\pi_+\pi_-^2 + \pi_-^3)p_-(\mathbf{x})}{\pi_{\mathrm{T}}}, \tag{4.9}$$

and samples in $\mathcal{D}_{2,b}$ can be considered to be independently drawn from

$$\tilde{p}_3(\mathbf{x}) = \pi_- p_+(\mathbf{x}) + \pi_+ p_-(\mathbf{x}). \tag{4.10}$$

Proof can be found in Section 5.7.

Theorem 6 indicates that from triplet comparison data, we can essentially obtain data points that can be drawn independently from three different distributions. We denote the three aggregated datasets as

$$\begin{aligned}\tilde{\mathcal{D}}_1 &= \mathcal{D}_{1,a} \cup \mathcal{D}_{1,c} \cup \mathcal{D}_{2,a} \cup \mathcal{D}_{2,c},\\ \tilde{\mathcal{D}}_2 &= \mathcal{D}_{1,b},\\ \tilde{\mathcal{D}}_3 &= \mathcal{D}_{2,b}.\end{aligned} \tag{4.11}$$

4.3 Unbiased risk estimator for triplet comparison data

We now attempt to express the classification risk,

$$R(f) \triangleq \mathop{\mathbb{E}}_{(\mathbf{x},y)\sim p(\mathbf{x},y)}[\ell(f(\mathbf{x}), y)], \tag{4.12}$$

on the basis of the three pointwise densities presented above.

The classification risk can be separately expressed as the expectations over $p_+(\mathbf{x})$ and $p_-(\mathbf{x})$. Although we do not have access to data drawn from these two distributions, we can obtain data from three related densities $\tilde{p}_1(\mathbf{x})$, $\tilde{p}_2(\mathbf{x})$, and $\tilde{p}_3(\mathbf{x})$ as indicated in Theorem 6. Letting

$$A \triangleq \frac{\pi_+^3 + 2\pi_+^2\pi_-}{\pi_{\mathrm{T}}}, \quad B \triangleq \frac{2\pi_+\pi_-^2 + \pi_-^3}{\pi_{\mathrm{T}}}, \tag{4.13}$$

we can express the relationship between these densities as

$$\begin{bmatrix}\tilde{p}_1(\mathbf{x})\\ \tilde{p}_2(\mathbf{x})\\ \tilde{p}_3(\mathbf{x})\end{bmatrix} = \begin{bmatrix}\pi_+ & \pi_-\\ A & B\\ \pi_- & \pi_+\end{bmatrix}\begin{bmatrix}p_+(\mathbf{x})\\ p_-(\mathbf{x})\end{bmatrix}. \tag{4.14}$$

Our goal is to solve the above equation so that we can express $p_+(\mathbf{x})$ and $p_-(\mathbf{x})$ in terms of the three densities from which we have i.i.d. data samples. To this end, we can rewrite the classification risk, which we want to minimize, in terms of $\tilde{p}_1(\mathbf{x})$, $\tilde{p}_2(\mathbf{x})$ and $\tilde{p}_3(\mathbf{x})$. An answer to Equation 4.14 is given by the following lemma.

Lemma 7. *We can express $p_+(\mathbf{x})$ and $p_-(\mathbf{x})$ in terms of $\tilde{p}_1(\mathbf{x})$, $\tilde{p}_2(\mathbf{x})$ and $\tilde{p}_3(\mathbf{x})$ as*

$$\begin{aligned} p_+(\mathbf{x}) &= \frac{1}{(ac-b^2)}\left((c\pi_+ - b\pi_-)\tilde{p}_1(\mathbf{x}) + (cA - bB)\tilde{p}_2(\mathbf{x}) + (c\pi_- - b\pi_+)\tilde{p}_3(\mathbf{x})\right), \\ p_-(\mathbf{x}) &= \frac{1}{(ac-b^2)}\left((a\pi_- - b\pi_+)\tilde{p}_1(\mathbf{x}) + (aB - bA)\tilde{p}_2(\mathbf{x}) + (a\pi_+ - b\pi_-)\tilde{p}_3(\mathbf{x})\right), \end{aligned} \tag{4.15}$$

provided $ac - b^2 \neq 0$ where

$$a \triangleq \pi_+^2 + A^2 + \pi_-^2, \quad b \triangleq 2\pi_+\pi_- + AB, \quad c \triangleq \pi_-^2 + B^2 + \pi_+^2.$$

Detailed derivation is given in Section 5.6.

As a result of the above lemma, we can express the classification risk using only triplet comparison data. Letting $\ell_+(\mathbf{x}) \triangleq \ell(f(\mathbf{x}), +1)$ and $\ell_-(\mathbf{x}) \triangleq \ell(f(\mathbf{x}), -1)$, we have the following theorem.

Theorem 8. *The classification risk can be equivalently expressed as*

$$\begin{aligned} R(f) = \frac{1}{(ac-b^2)}\{ &\mathop{\mathbb{E}}_{\mathbf{x}\sim\tilde{p}_1(\mathbf{x})}\left[\pi_{\text{test}}(c\pi_+ - b\pi_-)\,\ell_+(\mathbf{x}) + (1-\pi_{\text{test}})(a\pi_- - b\pi_+)\,\ell_-(\mathbf{x})\right] + \\ &\mathop{\mathbb{E}}_{\mathbf{x}\sim\tilde{p}_2(\mathbf{x})}\left[\pi_{\text{test}}(cA - bB)\,\ell_+(\mathbf{x}) + (1-\pi_{\text{test}})(aB - bA)\,\ell_-(\mathbf{x})\right] + \\ &\mathop{\mathbb{E}}_{\mathbf{x}\sim\tilde{p}_3(\mathbf{x})}\left[\pi_{\text{test}}(c\pi_- - b\pi_+)\,\ell_+(\mathbf{x}) + (1-\pi_{\text{test}})(a\pi_+ - b\pi_-)\,\ell_-(\mathbf{x})\right]\}, \end{aligned} \tag{4.16}$$

where $\pi_{\text{test}} \triangleq p_{\text{test}}(y=+1)$ denotes the class prior of the test dataset.

A proof is given in Section 5.7.

In this chapter, we consider the common case in which $\pi_{\text{test}} = \pi_+$, which means the test dataset shares the same class prior as the training dataset. However, even when $\pi_{\text{test}} \neq \pi_+$, which means the class prior shift [143] occurs, our method can still be used when π_{test} is known.

The process of obtaining the empirical risk minimizer of Equation 4.16: $\hat{f} = \arg\min R(f)$ is similar to other ERM-based learning approaches. As long as the risk representation that we want to minimize is continuous and differentiable with respect to the model parameters, such as the linear-in-parameter model or neural networks, we can use powerful stochastic optimization algorithms [80].

4.4 Estimation error bound

In this section, we establish an estimation error bound for the proposed unbiased risk estimator. Let $\mathcal{F} \subset \mathbb{R}^{\mathcal{X}}$ represent a function class specified by a model. First, let $\mathfrak{R}(\mathcal{F})$ be the (expected) Rademacher complexity of $\mathcal{F}$ which is defined as

$$\mathfrak{R}(\mathcal{F}) \triangleq \mathop{\mathbb{E}}_{Z_1,\cdots,Z_n\sim\mu}\mathop{\mathbb{E}}_{\boldsymbol{\sigma}}\left[\sup_{f\in\mathcal{F}}\frac{1}{n}\sum_{i=1}^{n}\sigma_i f(Z_i)\right] \tag{4.17}$$

where n is a positive integer, $Z_1, \cdots, Z_n$ are i.i.d. random variables drawn from a probability distribution with density μ, and $\boldsymbol{\sigma} = (\sigma_1, \cdots, \sigma_n)$ are Rademacher variables, which are random variables that take the value of $+1$ or -1 with even probabilities.

In general, we assume it holds that for any probability density μ, the specified model $\mathcal{F}$ satisfies $\mathfrak{R}(\mathcal{F}) \leq \frac{C_{\mathcal{F}}}{\sqrt{n}}$ for some constant $C_{\mathcal{F}} > 0$. Additionally, we use

$$f^* \triangleq \arg\min_{f \in \mathcal{F}} R(f) \tag{4.18}$$

to denote the true risk minimizer and

$$\hat{f} \triangleq \arg\min_{f \in \mathcal{F}} \hat{R}_{T,\ell}(f) \tag{4.19}$$

to denote the empirical risk minimizer.

Theorem 9. *Assume the following holds:*

- *The loss function ℓ is ρ-Lipschitz with respect to the first argument ($0 < \rho < \infty$).*
- *All functions in the model class $\mathcal{F}$ are bounded, i.e., there exists a constant C_b such that $||f||_\infty \leq C_b$ for any $f \in \mathcal{F}$.*

Let $C_\ell \triangleq \sup_{t \in \{\pm 1\}} \ell(C_b, t)$. Then for any $\delta > 0$, with probability at least $1 - \delta$ it holds that

$$R(\hat{f}) - R(f^*) \leq \left(\frac{2\rho C_{\mathcal{F}}}{\sqrt{n}} + \sqrt{\frac{C_\ell^2 \log \frac{2}{\delta}}{2n}} \right) \cdot \frac{C_R}{|ac - b^2|}, \tag{4.20}$$

where

$$\begin{aligned} C_R =& |\pi_{\text{test}}(c\pi_+ - b\pi_-)| + |(1 - \pi_{\text{test}})(a\pi_- - b\pi_+)| + |\pi_{\text{test}}(cA - bB)| + \\ & |(1 - \pi_{\text{test}})(aB - bA)| + |\pi_{\text{test}}(c\pi_- - b\pi_+)| + |(1 - \pi_{\text{test}})(a\pi_+ - b\pi_-)|. \end{aligned} \tag{4.21}$$

Since n appears in the denominator, it is obvious that when the class prior is fixed, the bound will get tighter as the amount of triplet comparison data increases. However, it is not clear how the bound will behave when we fix the amount of triplet comparison data and change the class prior. Thus in Figure 4.1, we show the behavior of the coefficient term $\frac{C_R}{|ac-b^2|}$ with respect to the same class prior of both training and test datasets. From the illustration, we can capture the rough trend that the bound gets tighter when the class prior becomes further from 0.5. We will further investigate this behavior in experiments.

4.5 On the class prior

In the previous sections, the class prior π_+ is assumed known. For this simple case, we can directly use the proposed algorithm to separate test data as well as identify correct classes. However, it may not be true for many real-world applications. There are two situations that can be considered. For the worst case, no information about the class prior is given. Although we still can estimate a result for the class prior from data and obtain a classifier that is able to separate data for different classes, we cannot identify the correct class without the information of which class has a higher class prior. A better situation is that we have the information of which class has a higher class prior. By setting this class as the positive class, we can successfully train a classifier to identify the correct class. Thus, we assume that the positive class has a higher class prior, which means $\pi_+ > \frac{1}{2}$.

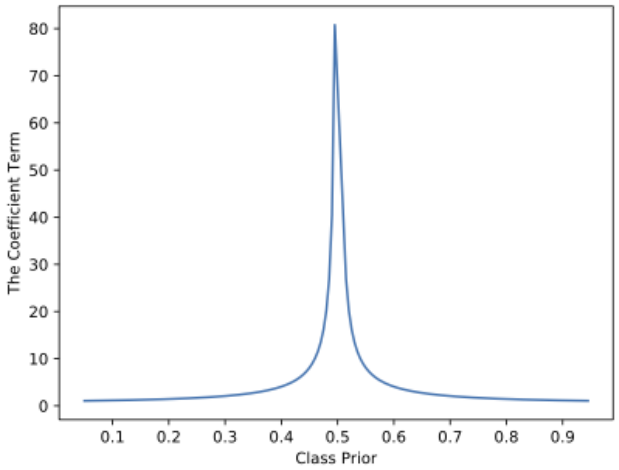


Figure 4.1: Behaviour of the coefficient term.

4.5.1 Class prior estimation from triplet comparison data

Noticing $\pi_{\mathrm{T}} = 1 - \pi_+ + \pi_+^2$, we can obtain $\pi_+^2 - \pi_+ + (1 - \pi_{\mathrm{T}}) = 0$. By assuming $\pi_+ > \pi_-$, we have

$$\pi_+ = \frac{1 + \sqrt{1 - 4(1 - \pi_{\mathrm{T}})}}{2}. \tag{4.22}$$

Since we can unbiasedly estimate π_{T} by $\frac{n_1}{n_1+n_2}$, the class prior π_+ can thus be estimated once the triplet comparison dataset is given.

4.6 User study

In this section, we first verify the motivation of using triplet comparisons for binary classification. To this end, we conducted user experiments on three forms of feedback: explicit labeling feedback, positivity pairwise comparison feedback and triplet comparison feedback.

4.6.1 Dataset

We choose to use the Oxford-IIIT pet dataset [120], whose classes and number of images of each class are listed as follows in Figure 4.2.

Among classes shown above, we choose to select two cat breeds, Birman and Ragdoll, for user studies. As shown below in Figure 4.3, these two breeds of cats have highly similar visual characteristics. We consider conducting binary classification for these two breeds would well simulate difficult situations to some extent.

4.6.2 Methods and query interface

Among all user studies, we prepare 10 problems of each of three forms of feedback. This is to say, a user need to answer 30 problems in total for each questionnaire. For the 10 problems of each type, we uniformly sample 10 images among which five images are from Birman images and another five images from Ragdoll images. For explicit labeling queries, each problem holds one image and the order is randomly shuffled. For pairwise comparison queries, each problem holds a pair

Breed	Count
American Bulldog	200
American Pit Bull Terrier	200
Basset Hound	200
Beagle	200
Boxer	199
Chihuahua	200
English Cocker Spaniel	196
English Setter	200
German Shorthaired	200
Great Pyrenees	200
Havanese	200
Japanese Chin	200
Keeshond	199
Leonberger	200
Miniature Pinscher	200
Newfoundland	196
Pomeranian	200
Pug	200
Saint Bernard	200
Samyoed	200
Scottish Terrier	199
Shiba Inu	200
Staffordshire Bull Terrier	189
Wheaten Terrier	200
Yorkshire Terrier	200
Total	4978

1.Dog Breeds

Breed	Count
Abyssinian	198
Bengal	200
Birman	200
Bombay	200
British Shorthair	184
Egyptian Mau	200
Main Coon	190
Persian	200
Ragdoll	200
Russian Blue	200
Siamese	199
Sphynx	200
Total	2371

2.Cat Breeds

Family	Count
Cat	2371
Dog	4978
Total	7349

3.Total Pets

Figure 4.2: Detailed Statistics of the Oxford-IIIT pet dataset [120].

Figure 4.3: Sample pictures of a Birman cat (left) and a Ragdoll cat (right).

of images and the pair if sequentially sampled without replacement from all 45 possible pairs. The same is for triplet comparison queries, which are sampled without replacement from all possible triplets. We conducted user studies under the following four different situations. They are discriminated based on how images are sampled for queries of different forms.

- All three types of queries use different sets of 10 images. This is to say, we prepared 30 different images in total, 15 for Birman and 15 for Ragdoll. As this number is too small to be a feasible classification feedback, we do not train classifiers in this experiment but only report annotation accuracies.
- We use the same set of 10 images for all types of queries. We also mix all 30 questions and shuffle their appearing order.

- With the similar use of images as the above situation, we consider to draw attention of users on assuring accuracy in this setting. Specifically, we added note to ask users answer with the correction selection as one can as possible. In this way, we can objectively evaluate the difficulty of different types of queries.
- With the similar setting of the above one, we change the layout of the options of triplet comparison feedback to make it appearing more similar to explicit labeling feedback.

Same as user studies of Chapter 3, we adopted the Lancers platform for recruiting annotators and collecting feedback. We also split questions for each experimental setting into separate crowdsourcing tasks with proper sizes ranging from 35 to 55 so that it is feasible to be posted on the platform. On the Lancers platform, our user profile is still set to have the same type of industry of mass media. The English translation of description of the task is:

> Please answer the following questions regarding the determination of type judgement of cats. Please also answer the difficulty level of the questions. At the end of the questionnaire, you will be asked to answer your approach in the judgment in a simple, free-text format.

For interface, We use a similar layout as the user study using the Kuzushiji dataset as shown in Figure 3.11, Figure 3.12 and Figure 3.13 in Chapter 3, which are composed of simple HTML [1] elements without fancy CSS [2] styles. Therefore, we omit to show screenshots in this section.

4.6.3 Results and discussion

In this section, we present the results of each study setting in a logical order.

Using different sets of images

Accuracy and difficulty results for all questions are reported in Figure 4.4. The overall difficulties for explicit label feedback, pairwise comparison feedback and triplet comparison feedback are 1.95, 1.86, 2.19, respectively. From these results, We can observe that although the triplet comparison feedback can offer a similarly significantly high accuracy as the pairwise counterpart, they usually suffer from higher evaluation difficulty, though not significant enough.

Although these results can support the usage of triplet comparison feedback for annotation of binary classification to some extend, there is possibility remaining that the evaluation difficulty may come from the innate difficulty of different images used for each type of feedback, which we examine in the following studies.

Using the same set of images

In order to get rid of the influence of the innate difficulty of different images, we choose to use the same set of 10 images for all types of questions. Furthermore, we also shuffle their appearing order to prevent it from influencing user annotation and evaluation.

[1] HyperText Markup Language
[2] Cascading Style Sheets

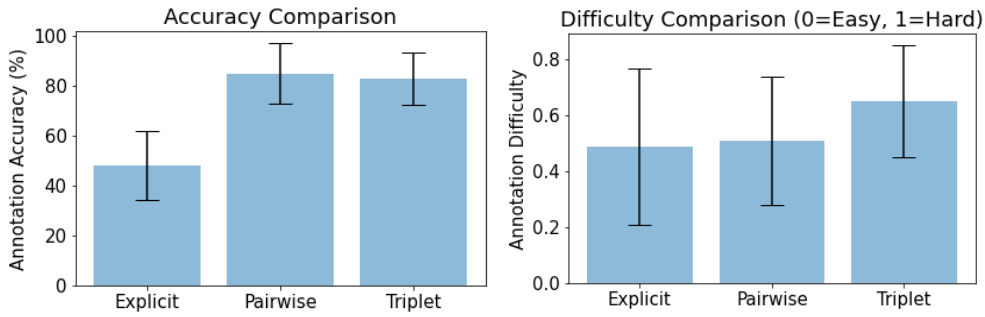


Figure 4.4: Accuracy and difficulty results using different sets of images.

In order to further clarify the results, we repeated the same user study with two different sets of 10 images. We are tested an alternative layout for questions on triplet comparison feedback.

Detailed results are shown in the following Figure 4.5, Figure 4.6 and Figure 4.7. The overall difficulty evaluation for each type of feedback is 1.85, 2.02, 2.13 when using the first set of images; 1.97, 2.01, 2.02 when using the second set of images; and 1.91, 2.02, 2.08 when using both sets of images.

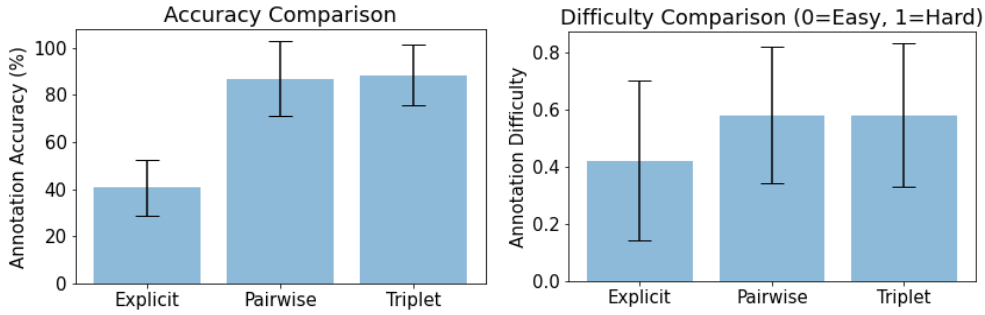


Figure 4.5: Accuracy and difficulty results using the first set of images.

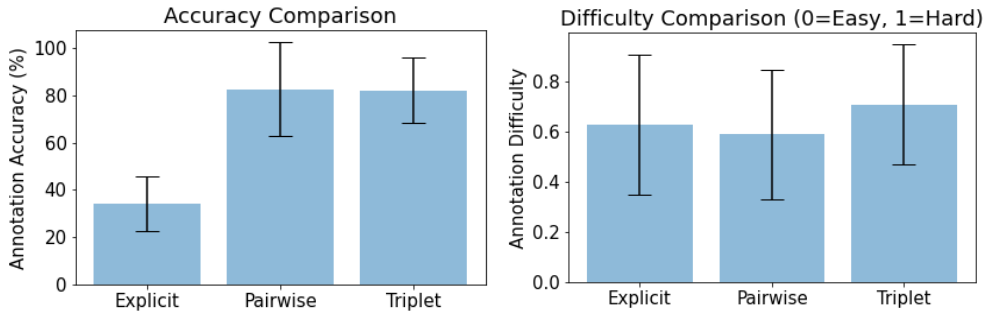


Figure 4.6: Accuracy and difficulty results using the second set of images.

First, for the accuracy of each type of feedback, the trend remains the same as in the previous user study that comparison feedback is significantly more accurate than explicit labeling feedback. This strengthen the motivation of incorporating such feedback to mitigate the errors caused by the explicit labeling in some certain applications.

Then, for the difficulty evaluation, we can see that the gap between explicit labeling feedback and comparison feedback clearly shrinks with larger number of images. Although these results can support the motivation of using comparison feedback, giving more accurate feedback with the same difficulty, we think this

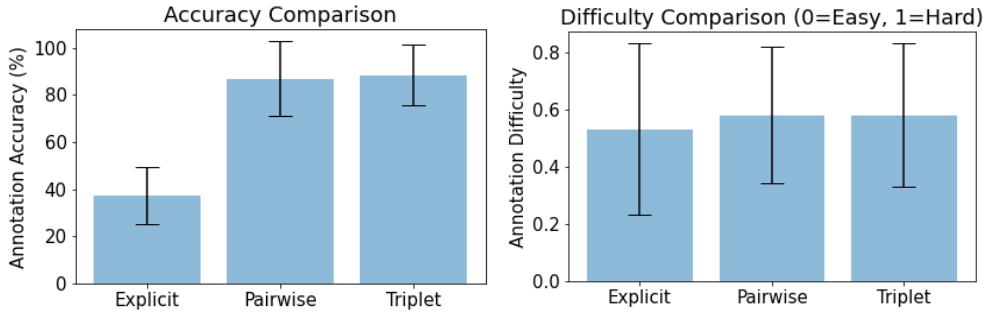


Figure 4.7: Accuracy and difficulty results using both sets of images.

may be caused by the inappropriate way users evaluate difficulty. When not thinking about the accuracy of annotation, a user may just give an answer of her mind and does not consider the question as a difficult one. Therefore, users may confuse the difficulty of accurately annotation and that of feedback type and cannot answer the later one separately. To this end, in order to evaluate the feedback difficulty in an accurate and objective way, we think it is necessary to let users keep in mind to keep the accuracy as high as possible. This is to be presented in the following section.

Using the same set of images with notification on accuracy

In order to separate the difficulty purely caused by the feedback form, we added the following notification at the beginning of the questionnaire to let users try their best effort to keep annotations as accurate as possible. The note is shown in users native language the we provide the English translation here.

> Note: Since we are experimenting with the difference in approach between paying according to the number of correct answers and paying all answers regardless to the number of correct answers, please answer as if you are being paid according to the number of correct answers this time. (Actually, you will be paid for all questions, including the ones you answered incorrectly.)

Results are mainly shown in Figure 4.8, and the overall difficulties for three types of feedback are 1.79, 1.96 and 2.25, respectively.

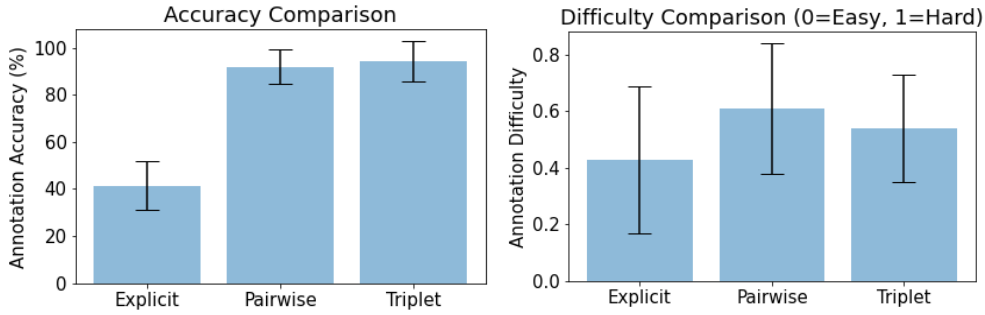


Figure 4.8: Accuracy and difficulty results using the first set of images with notification on accuracy.

We can observe that the difficulty of triplet comparison feedback decreased to be easier in this setting, and its accuracy remains the same as pairwise comparisons. The results indicate that triplet comparison is a favorable feedback for

binary classification, although users mentally consider it to be the hardest in the overall scale.

We further investigate the detailed bar chart of individual difficulty evaluations of each question in Figure 4.9. It can observed that, being different from the other two types of feedback, difficulty evaluations of the triplet comparison feedback shows a clear shift to the easier side.

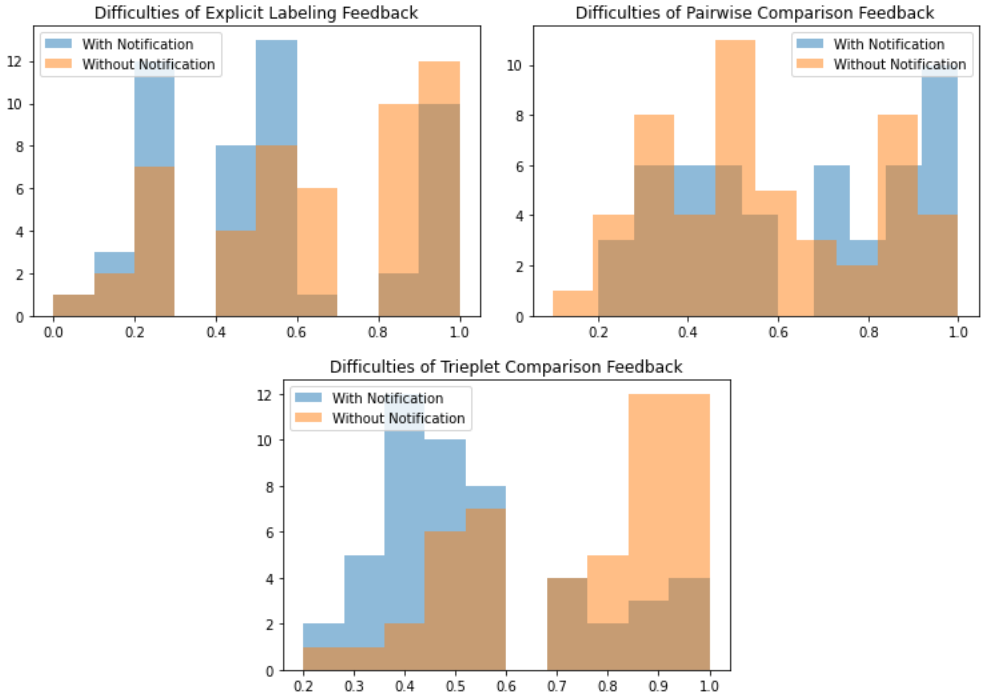


Figure 4.9: Difficulty evaluations

Using an alternative layout of question on triplet comparison feedback

Here we investigate how the presentation of the question would influence the feedback quality.

As shown in Figure 4.10 and Figure 4.11, the triplet comparison feedback shows consistent difficulty evaluation with the explicit labeling feedback instead of the pairwise comparison feedback shown in previous results. This indicates that the layout for presenting a feedback also plays an important and unignorable role in the quality of its collected annotation. However, we would like to note that changing on the presentations does not directly indicate changes on the trend of accuracy evaluations.

4.6.4 User opinions

Here, we present the collected user opinions in natural language how they intuitively feel about conducting annotation for each type of feedback.

The following lists user answers for advantages of triplet comparison feedback.

- Compared to explicit evaluation and pairwise comparison, there are more areas that can be compared and the differences are easier to understand.
- There are three objects to compare, so it is easy to see the differences in features.

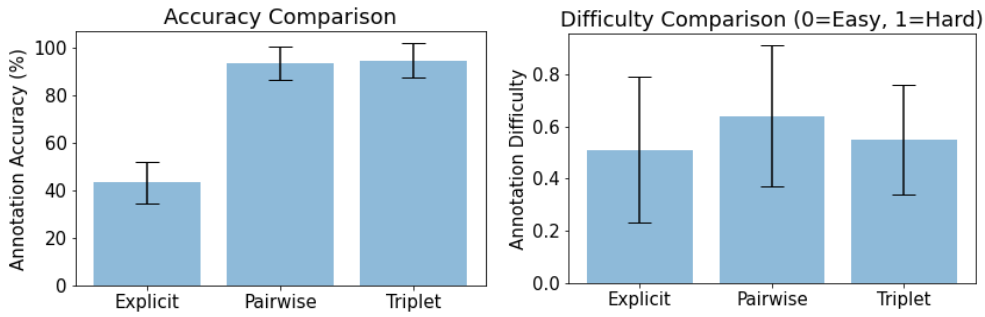


Figure 4.10: Accuracy and difficulty results using the new layout and the first set of images with notification on accuracy.

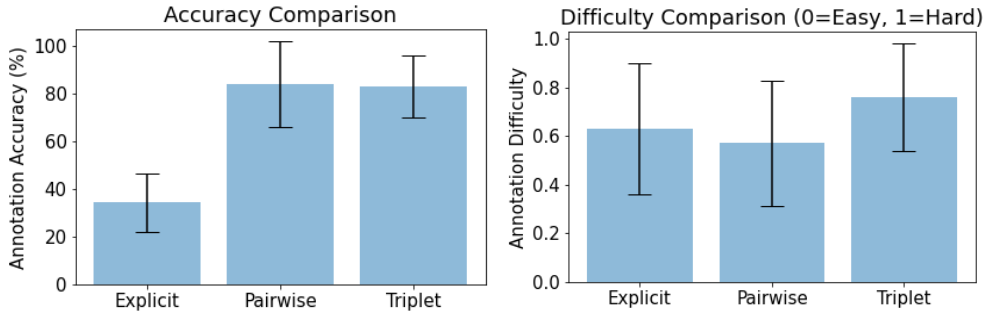


Figure 4.11: Accuracy and difficulty results using the new layout and the second set of images with notification on accuracy.

- If you're not sure what kind of cat you're looking for, just pick one that looks like it.
- Even if you don't have knowledge of the species, you can compare and consider their visual characteristics.
- Instead of clearly choosing between a Birman and a Ragdoll, it's easier to just choose the one that looks more like.

The following lists user answers for disadvantages of triplet comparison feedback.

- Some of the images do not show the features or are difficult to understand, making it difficult to distinguish between them.
- It's hard to make a choice when you compare it to a reference photo and neither one looks like the other.
- It's hard to choose if they're both similar.
- There is a lot of information out there, and that's where it gets hard to make a decision.
- The more information you have compared to others, the more your standard of "this is the way it is" will be shaken, which will lead to hesitation and difficulty in making decisions.

We can observe that most of the disadvantages listed above are not restricted to the triplet comparison, but also exist for other types of feedback. However, most of the advantages listed above are mainly due to unique design of the triplet

comparison feedback. Moreover, as we separate all triplets into two sets as indicated in the data generation process, the overall performance is essentially robust to annotations in such cases and will not drop significantly as long as the triplet belongs to the correct set.

4.7 Simulation study

In this section, we conducted experiments using real world datasets to evaluate and investigate the performance of the proposed method for triplet classification. This section being called simulation study is because the triplet comparison data is generated by simulation, instead of actually being collected from the real-world.

4.7.1 Baseline methods

- KMEANS: As a simple baseline, we used k-means clustering [103] with $k = 2$ on all the data instances of triplets while ignoring all the relation information.
- ITML: Information-theoretic metric learning [41] is a metric learning method that requires pairwise the relationship between data instances. From a triplet $(\mathbf{x}_a, \mathbf{x}_b, \mathbf{x}_c)$, we constructed pairwise constraints as $(\mathbf{x}_a, \mathbf{x}_b)$ being similar and $(\mathbf{x}_a, \mathbf{x}_c)$ being dissimilar. Using the metric returned by the algorithm, we conducted k-means clustering on test data. We used the identity matrix for prior knowledge and fix the slack variable as $\gamma = 1$.
- TL: Triplet loss [131] is a loss function proposed in the context of deep metric learning which can learn a metric directly from triplet comparison data. Using the metric returned by the algorithm, we conducted k-means clustering on test data.
- SERAPH: Semi-supervised metric learning paradigm with hyper sparsity [114] is a metric learning method based on entropy regularization. We formulated a pairwise relationship in the same manner as with ITML. Using the metric returned by ITML, we conducted k-means clustering on test data.
- SU: SU learning [12] is a method for learning a binary classifier from similarity and unlabeled data. We used the same method for estimating the class prior, and considered the less similar sample in a triplet as unlabeled data.

4.7.2 Datasets

UCI datasets: We used six datasets from the *UCI Machine Learning Repository* [5]. They are binary classification datasets and we use the given labels for further triplet comparison data generation.

Image datasets: We used the MNIST [95], the Fashion MNIST [153] and the CIFAR-10 [88] image datasets.

Although these datasets have labels, using the triplet comparison data composed of labeled data fulfills the purpose of experiments which is to assess whether the proposed method can work properly. As mentioned before, the proposed method can be applied to situations where we do not have access to the labels.

For all datasets, we randomly sampled from the original datasets to generate triplet comparisons maintaining the ratio of n_1 and n_2.

4.7.3 Proposed method

For the proposed method and existing methods that require a model architecture, we used a fully-connected neural network with only 1 hidden layer of width 100 and rectified linear units (ReLUs) [113] for all the datasets except for CIFAR-10. The width of the hidden layer was set to be 100 through out all experiments. Adam [80] was used for optimization. The neural network architecture used for CIFAR-10 is specified below. Two surrogate losses, namely the squared loss and the double hinge loss, were used as indicated in the results tables.

CNN Structure for CIFAR10

The following structure is used:

- Convolution (3 in/32 out-channels, kernel size 3) with ReLU.
- Convolution (32 in/32 out-channels, kernel size 3) with ReLU.
- Max-pooling (kernel size 2, stride 2).
- Repeat twice:
  - Convolution (32 in/32 out-channels, kernel size 3) with ReLU.
  - Convolution (32 in/32 out-channels, kernel size 3) with ReLU.
  - Max-pooling (kernel size 2, stride 2).
- Fully-connected (512 units) with ReLU.
- Fully-connected (1 unit).

4.7.4 Results

The proposed method estimates the unknown class prior first. For baseline methods, performances are measured by the clustering accuracy $1-\min(r, 1-r)$ where r is the error rate. The results of different triplet numbers are listed in Table 4.1, Table 4.2 and Table 4.3. The best and equivalent methods are shown in bold face on the one-sided t-test with a significance level of 5%. Also as shown in Figure 4.12, the performance of the proposed method arises with more training data and remains in a consist range with respect to the class prior, which follows the prediction by the theory in most of the cases.

Table 4.1: Experimental results with class prior as 0.7 and 1000 training triplets.

	Proposed Methods		Baselines				
Dataset	Squared	Double Hinge	KMEANS	ITML	TL	SERAPH	SU
adult	65.54 (0.41)	64.19 (0.61)	71.94 (0.10)	71.04 (1.00)	61.48 (1.36)	71.04 (1.00)	**75.88 (0.50)**
breast	**97.41 (0.28)**	**96.90 (0.31)**	96.20 (0.34)	95.84 (0.29)	93.87 (0.78)	96.72 (0.23)	65.26 (0.76)
diabetes	**70.71 (0.84)**	64.87 (0.74)	66.69 (0.70)	65.91 (0.69)	64.38 (1.60)	67.44 (0.78)	34.42 (0.73)
magic	61.75 (1.00)	**71.91 (0.39)**	65.08 (0.17)	64.79 (0.17)	65.42 (0.22)	64.96 (0.19)	34.77 (0.19)
phishing	**76.58 (0.30)**	74.95 (0.27)	63.43 (0.50)	63.75 (0.23)	57.85 (0.92)	63.42 (0.53)	34.17 (0.22)
spambase	**62.08 (1.87)**	**64.66 (1.04)**	**63.59 (0.24)**	**63.24 (0.31)**	59.59 (1.57)	**63.28 (0.34)**	60.27 (0.30)
mnist	79.86 (0.35)	**80.78 (0.34)**	65.24 (0.25)	0.00 (0.00)	58.26 (1.24)	0.00 (0.00)	50.80 (0.03)
fashion	89.73 (0.33)	**91.62 (0.33)**	74.90 (1.00)	0.00 (0.00)	76.83 (1.31)	0.00 (0.00)	49.85 (0.08)
cifar10	**76.39 (1.57)**	66.28 (2.51)	64.17 (0.01)	0.00 (0.00)	60.17 (1.26)	0.00 (0.00)	59.50 (0.50)
Count	5	5	1	1	0	1	1

Table 4.2: Experimental results with class prior as 0.7 and 500 training triplets.

	Proposed Methods		Baselines				
Dataset	Squared	Double Hinge	KMEANS	ITML	TL	SERAPH	SU
adult	62.72 (0.57)	59.74 (1.44)	71.44 (0.60)	71.79 (0.20)	58.53 (1.17)	70.54 (1.09)	**76.30 (0.04)**
breast	**96.90 (0.44)**	**96.53 (0.35)**	**96.28 (0.29)**	**96.79 (0.24)**	89.67 (1.97)	**96.68 (0.27)**	64.12 (0.91)
diabetes	**69.64 (0.68)**	67.08 (0.91)	66.27 (0.65)	64.87 (0.66)	63.15 (1.56)	67.44 (0.68)	33.90 (0.67)
magic	63.86 (1.44)	**70.37 (0.36)**	64.86 (0.15)	65.03 (0.13)	66.36 (0.30)	64.94 (0.14)	34.83 (0.15)
phishing	**75.52 (0.31)**	74.57 (0.37)	63.08 (0.47)	63.31 (0.41)	56.37 (1.18)	62.73 (0.76)	33.89 (0.20)
spambase	61.18 (1.11)	59.95 (1.38)	**63.55 (0.32)**	**64.17 (0.31)**	59.35 (1.48)	**63.53 (0.35)**	58.96 (0.44)
mnist	**74.23 (0.32)**	**75.19 (0.50)**	64.74 (0.55)	0.00 (0.00)	56.07 (0.87)	0.00 (0.00)	50.87 (0.26)
fashion	83.83 (0.55)	**87.86 (0.66)**	75.40 (0.34)	0.00 (0.00)	76.66 (1.39)	0.00 (0.00)	49.88 (0.08)
cifar10	**66.28 (1.77)**	**62.63 (2.53)**	**64.16 (0.01)**	0.00 (0.00)	61.26 (1.13)	0.00 (0.00)	59.05 (0.65)
Count	5	5	3	2	0	2	1

Table 4.3: Experimental results with class prior as 0.7 and 200 training triplets.

	Proposed Methods		Baselines				
Dataset	Squared	Double Hinge	KMEANS	ITML	TL	SERAPH	SU
adult	58.12 (0.90)	55.10 (1.00)	70.54 (1.50)	70.04 (1.17)	58.28 (0.94)	68.54 (1.67)	**75.27 (0.51)**
breast	**96.68 (0.32)**	**96.50 (0.35)**	**95.91 (0.34)**	**96.24 (0.24)**	94.27 (0.68)	**96.64 (0.28)**	66.20 (0.80)
diabetes	**69.25 (0.98)**	65.36 (0.89)	64.97 (0.87)	**67.27 (0.72)**	63.47 (1.22)	**67.11 (0.82)**	35.23 (0.94)
magic	60.54 (1.88)	**68.56 (0.53)**	64.88 (0.13)	65.15 (0.14)	66.31 (0.42)	64.97 (0.15)	34.60 (0.34)
phishing	**72.22 (0.62)**	**72.11 (0.65)**	63.70 (0.26)	63.71 (0.21)	57.02 (1.41)	63.17 (0.77)	34.03 (0.32)
spambase	57.69 (1.68)	55.74 (1.19)	**63.78 (0.34)**	**63.04 (0.35)**	60.78 (1.63)	**63.74 (0.25)**	58.92 (0.43)
mnist	67.14 (0.67)	**70.96 (0.53)**	64.49 (1.00)	0.00 (0.00)	57.88 (1.43)	0.00 (0.00)	50.10 (0.62)
fashion	76.67 (0.40)	**83.74 (0.55)**	74.90 (1.00)	0.00 (0.00)	73.24 (1.80)	0.00 (0.00)	47.97 (0.76)
cifar10	**63.14 (1.68)**	58.83 (2.16)	**64.16 (0.01)**	0.00 (0.00)	61.23 (1.18)	0.00 (0.00)	58.65 (0.66)
Count	4	5	3	3	0	3	1

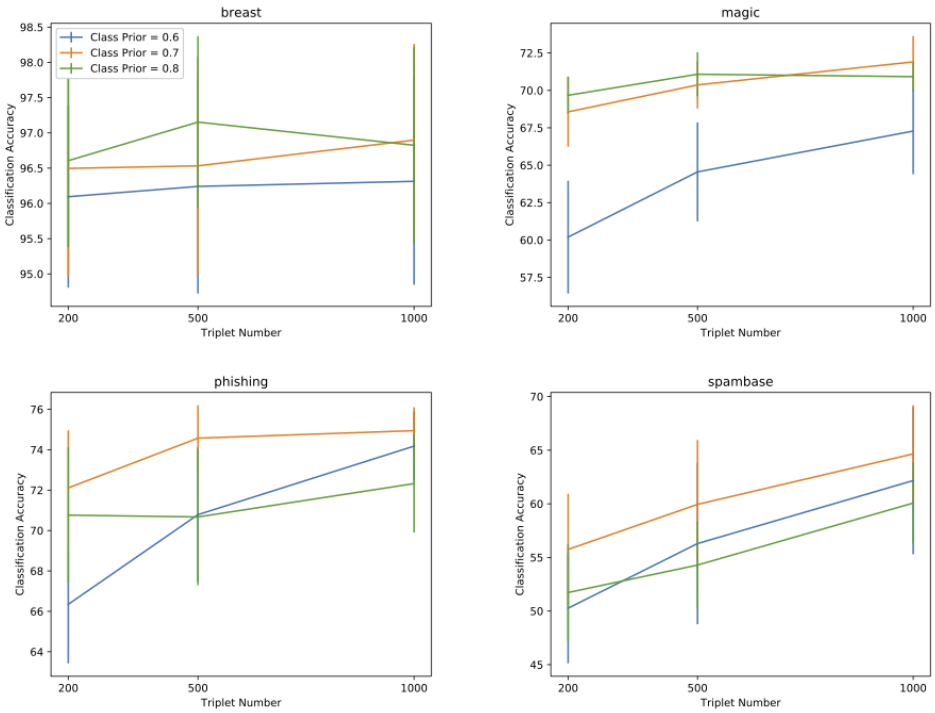


Figure 4.12: Average classification error and standard deviation over 20 trials.

4.8 Conclusion

In this chapter, we proposed a novel method for learning a classifier from *only* passively obtained triplet comparison data. We established an estimation error bound for the proposed method, and confirmed that the estimation error decreases as the amount of triplet comparison data increases. We also empirically confirmed that the performance of the proposed method surpassed multiple baseline methods on various datasets.

Chapter 5

Proofs

In this chapter, we present proofs for theorems presented in Chapter 3 and Chapter 4.

5.1 Proof of Theorem 2

Theorem (Error rate bound)**.** *Suppose the following situations hold:*

1. *Condition 1 and Condition 2 hold for* $\epsilon_{pos}, \epsilon_{unc} \in [0, 0.5)$.
2. *There exist* $t = \Omega\left(\frac{\log 2}{2(0.5-\epsilon_{pos})^2}\right)$, $\epsilon > 0$, $D \subset \mathcal{X}$ *and* $n > \frac{t}{\epsilon}$, *where* $n = |D|$ *denotes the size of the initial unlabeled dataset.*

Then, there exist constants C_1 *and* C_2 *such that running Algorithm 1 on* D *with hyper-parameters* t *and* $m \geq \frac{C_1 \max(\log\log n, \log t)}{(0.5-\epsilon_{unc})^2}$, *with probability at least* $1-\delta$ *the following will hold:*

- *The error rate of inferred labels is bounded as* $|\{i \in [n] | \hat{y}_i \neq h^*(x_i)\}| \leq \epsilon n$.
- *The query complexity for* O_{pos} *is* $\mathcal{O}\left(\frac{n}{(0.5-\epsilon_{pos})^2}\right)$.
- *The query complexity for* O_{unc} *is* $\mathcal{O}\left(\frac{n \log\log n}{(0.5-\epsilon_{unc})^2}\right)$.

For simplicity, we denote $\delta \triangleq \delta(C_2, n, t, \epsilon_{pos})$.

Proof. Algorithm 1 consists of the following two steps

1. Selecting of relatively high uncertainty points.
2. Inferring labels by majority vote.

For step 1, the algorithm of [109] is executed using parameters $K = t$ and m. Directly adapting Theorem 1 in [109], we know that if $m \geq \frac{C_1 \max(\log\log n, \log t)}{(0.5-\epsilon_{\text{unc}})^2}$, then the correct top-$t$ points can be identified with probability at least $1 - \log n^{-C_2}$.

For step 2, we analyze the probability that a point $x \in D \setminus D'$ is correctly inferred. Without loss of generality, we assume the correct label for x is 1 and we calculate the probability that $\sum_{x_j \in D'} \mathrm{O}_{\text{pos}}(x, x_j) \geq \frac{1}{2}$.

Let $Z_j \triangleq \mathrm{O}_{\text{pos}}(x, x_j)$ denote the random variable representing the outcome of every query to O_{pos}. Because D' is assumed to be correctly identified by step 1, so $p(y|x) \geq p(y|x_j)$ should hold for every $x_j \in D'$. Thus, the expectation of Z_j

is $1-\epsilon_{\text{pos}}$. Note that Z_j only takes a value of either 0 or 1, thus by applying Hoeffding's inequality to $Z_1, Z_2, \cdots, Z_t$, we have

$$\Pr\left[\frac{1}{t}\sum_{j=1}^{t} Z_j - (1-\epsilon_{\text{pos}}) \leq -(0.5-\epsilon_{\text{pos}})\right] \leq \exp\left(-2t(0.5-\epsilon_{\text{pos}})^2\right). \tag{5.1}$$

This expresses the probability that $\frac{1}{t}\sum_{j=1}^{t} Z_j$ is smaller than 0.5.

Let $a \triangleq \exp\left(-2t(0.5-\epsilon_{\text{pos}})^2\right)$. Because t is selected so that $a \leq \frac{1}{2}$ and $\frac{1}{t}\sum_{j=1}^{t} Z_j$ is bounded within $[0,1]$, therefore for a single $x \in D \setminus D'$ it holds that

$$\begin{aligned}\Pr\left[\frac{1}{t}\sum_{j=1}^{t} Z_j \geq \frac{1}{2}\right] &\geq 1-a \\ &\geq \exp(-a(a+1)).\end{aligned} \tag{5.2}$$

This is because the assumption on the positivity comparison oracle is defined in a pointwise way so the above inequality can be derived.

In conclusion, for all data points in $D \setminus D'$ to be correctly labeled, the error rate $\epsilon = \frac{t}{n}$ can be achieved with probability at least $1-\delta$ where

$$\delta \triangleq 1 - (1 - \log n^{-C_2})\exp(-a(a+1)(n-t)). \tag{5.3}$$

For query complexities, as O_{pos} is queried $t(n-t)$ times, the query complexity of O_{pos} is $\mathcal{O}\left(\frac{n}{(0.5-\epsilon_{\text{pos}})^2}\right)$. Moreover, as indicated by Eq. (17) of Mohajer et al. [109], the query complexity of O_{unc} is $\mathcal{O}\left(\frac{n\log\log n}{(0.5-\epsilon_{\text{unc}})^2}\right)$. □

5.2 Proof of Theorem 3

Theorem (Generalization error bound for classifiers learned from downstream k-NN)**.** *Let the input and the output of Algorithm 1 be $D = \{x_i\}_{i=1}^n$ and $\hat{Y} = \{\hat{y}_i\}_{i=1}^n$. Let $\hat{f}(x;k)$ be the k-NN classifier obtained and $f^*(x) \triangleq \mathbb{1}_{\eta(x)\geq\frac{1}{2}}$ be the Bayes classifier. Suppose the following situations hold:*

1. *The conditions for Theorem 2 hold.*
2. *Assumption 1 holds with $\lambda > 0$ and $\omega > 0$.*
3. *Assumption 2 holds with $\alpha \geq 0$ and $C_\alpha \geq 1$.*

Then, using the same notations as Theorem 2, for $\delta' \in (0,1)$, $4\log(\frac{1}{\delta'}) + 1 \leq k \leq \frac{n}{2}$, with probability at least $(1-\delta)(1-\delta')$, it holds that

$$R(\hat{f}) \leq R(f^*) + C_\alpha\left(\frac{2\epsilon}{k} + \omega\left(\frac{2k}{n}\right)^\lambda\right)^{\alpha+1}. \tag{5.4}$$

Proof. First, we bound the difference between $\hat{f}(x;k)$ and $f(x)$. Similar to Reeve et al. [126], we define $\tilde{f}(x;k) = \mathbb{E}_{p(y|x)} = \frac{1}{k}\sum_{q=1}^{k} y_{\tau_q(x)}$.

Then we have

$$\left|\hat{f}(x;k) - f(x)\right| \leq \left|\hat{f}(x;k) - \tilde{f}(x;k)\right| + \left|\tilde{f}(x;k) - f(x)\right|. \tag{5.5}$$

For the first term in RHS, from Theorem 2, we know it is bounded by $\frac{2\epsilon}{k}$ with probability at least $1-\delta$. For the second term in right hand side, from Lemma

4.1 in Reeve et al. [126], we know it is bounded by $\omega\left(\frac{2k}{n}\right)^{\lambda}$ with probability at least $1-\delta'$ for $\delta'>0$ and $\frac{n}{2}\geq k\geq 4\log(\frac{1}{\delta'})+1$. Therefore, combing the two inequalities, we can derive the left hand side is bounded by $\Delta\triangleq\frac{2\epsilon}{k}+\omega\left(\frac{2k}{n}\right)^{\lambda}$ with probability at least $(1-\delta)(1-\delta')$. This means with at least the same probability, a randomly drawn point from $\mathcal{X}$ will fall in the set

$$\mathcal{X}'\triangleq\{x\in\mathcal{X}:|\hat{\eta}(x)-\eta(x)|\leq\Delta\}.$$

Therefore, it holds that

$$\begin{aligned}
&R(\hat{f})-R(f^*) && (5.6)\\
=&\int_{\mathcal{X}}\left|\eta(x)-\frac{1}{2}\right|\mathbb{1}_{\hat{f}(x)\neq f^*(x)}d\mu(x) && (5.7)\\
=&\int_{\mathcal{X}'}\left|\eta(x)-\frac{1}{2}\right|\mathbb{1}_{\hat{f}(x)\neq f^*(x)}d\mu(x) && (5.8)\\
&\big(\text{with probability at least}(1-\delta)(1-\delta')\big) && (5.9)\\
\leq&\int_{\mathcal{X}}\left|\eta(x)-\frac{1}{2}\right|\mathbb{1}_{|\eta(x)-\frac{1}{2}|\leq\Delta}d\mu(x) && (5.10)\\
\leq&C\Delta^{\alpha+1}. && (5.11)
\end{aligned}$$

□

5.3 Proof of Corollary 4

Corollary (Generalization error bound for classifiers learned with disagreement-based active learning)**.** *Suppose conditions for Theorem 2 hold. Then, when running Algorithm 2 with $\epsilon\in(0,1)$ and $\epsilon_i=\frac{1}{2^{i+2}}$, with probability at least $1-\delta$, the output $\hat{h}$ satisfies*

$$P_{x\sim\mathcal{P}_{\mathcal{X}}}[\hat{h}(x)\neq h^*(x)]\leq\epsilon. \tag{5.12}$$

Proof. Similar to the approach in Xu et al. [159], we use induction to show that at the end of every step i, $\mathbb{E}_{\mathcal{P}_{\mathcal{X}}}[h(x)\neq h^*(x)]\leq 4\epsilon_i$ always holds with probability at least $(1-\delta)^{\log(\frac{1}{\epsilon})}$ for a universal δ, which is obvious for $i=0$.

Then, with a little abusing of notations, we have

$$\begin{aligned}
&|x\in S_i:h(x)\neq h^*(x)| && (5.13)\\
=&|x\in D_i:h(x)\neq h^*(x)| && (5.14)\\
\leq&|x\in D_i:h(x)\neq\hat{y}|+|x\in D_i:h^*(x)\neq\hat{y}| && (5.15)\\
=&2\epsilon_i|S_i|. && (5.16)
\end{aligned}$$

Therefore, it holds that $\mathrm{P}_{x\sim S_i}[h(x)\neq h^*(x)]=\frac{|x\in S_i:h(x)\neq h^*(x)|}{|S_i|}\leq 2\epsilon_i$. Having $c_0\in(1,\infty)$ and $\gamma\in(0,1)$, using Lemma 3.1 from [64], we have $\mathrm{P}_{x\sim\mathcal{P}_{\mathcal{X}}}[h(x)\neq h^*(x)]\leq 4\epsilon_i$ with probability at least $1-\gamma$, providing $c_0\frac{d\log(\frac{|S_i|}{d})+\log(\frac{1}{\gamma})}{|S_i|}\leq\epsilon_i$. Setting $\gamma=1-(1-\delta)^{\log(2\epsilon)}$, We have $\mathrm{P}_{\mathcal{P}_{\mathcal{X}}}[\hat{h}(x)\neq h^*(x)]\leq\epsilon$ with probability at least $(1-\delta)^{\log(\frac{1}{\epsilon})}(1-\delta)^{\log(2\epsilon)}=1-\delta$ at the end of the algorithm. □

5.4 Proof of Lemma 5

Lemma. *Corresponding to the data generation process described above, let*

$$\begin{aligned}
p_1(\mathbf{x}_a,\mathbf{x}_b,\mathbf{x}_c)&=\frac{p(\mathbf{x}_a,\mathbf{x}_b,\mathbf{x}_c,(y_a,y_b,y_c)\in\mathcal{Y}_1)}{\pi_{\mathrm{T}}},\\
p_2(\mathbf{x}_a,\mathbf{x}_b,\mathbf{x}_c)&=\pi_+p_+(x_a)p_-(x_b)p_+(x_c)+\pi_-p_-(x_a)p_+(x_b)p_-(x_c),
\end{aligned}$$

where $\pi_{\mathrm{T}} \triangleq 1 - \pi_+\pi_-$, $\pi_+ \triangleq p(y=+1)$ *and* $\pi_- \triangleq p(y=-1)$ *are the class prior probabilities that satisfy* $\pi_+ + \pi_- = 1$*;* $p_+(x) \triangleq p(x|y=+1)$ *and* $p_-(x) \triangleq p(x|y=-1)$ *are class conditional probabilities. Then, it holds that*

$$\mathcal{D}_1 = \{(x_{1,a}, x_{1,b}, x_{1,c})\}_{i=1}^{n_1} \overset{\text{i.i.d.}}{\sim} p_1(\mathbf{x}_a, \mathbf{x}_b, \mathbf{x}_c),$$
$$\mathcal{D}_2 = \{(x_{2,a}, x_{2,b}, x_{2,c})\}_{i=1}^{n_2} \overset{\text{i.i.d.}}{\sim} p_2(\mathbf{x}_a, \mathbf{x}_b, \mathbf{x}_c).$$

Proof. From the data generation process, we can consider the generation distribution for data of $\mathcal{D}_1$ as

$$\begin{aligned} p_1(\mathbf{x}_a, \mathbf{x}_b, \mathbf{x}_c) &= p(\mathbf{x}_a, \mathbf{x}_b, \mathbf{x}_c | (y_a, y_b, y_c) \in \mathcal{Y}_1) \\ &= \frac{p(\mathbf{x}_a, \mathbf{x}_b, \mathbf{x}_c, (y_a, y_b, y_c) \in \mathcal{Y}_1)}{p((y_a, y_b, y_c) \in \mathcal{Y}_1)} \\ &= \frac{p(\mathbf{x}_a, \mathbf{x}_b, \mathbf{x}_c, (y_a, y_b, y_c) \in \mathcal{Y}_1)}{\pi_+^3 + 2\pi_+^2\pi_- + 2\pi_+\pi_-^2 + \pi_-^3}. \end{aligned} \tag{5.17}$$

Note that the denominator in Equation 5.17 can be rewritten as

$$\begin{aligned} \pi_{\mathrm{T}} &\triangleq \pi_+^3 + 2\pi_+^2\pi_- + 2\pi_+\pi_-^2 + \pi_-^3 \\ &= (\pi_+^3 + \pi_-^3) + 2(\pi_+^2\pi_- + \pi_+\pi_-^2) \\ &= \pi_+^2 + \pi_+\pi_- + \pi_-^2 \\ &= 1 - \pi_+\pi_-, \end{aligned} \tag{5.18}$$

then we have

$$p_1(\mathbf{x}_a, \mathbf{x}_b, \mathbf{x}_c) = \frac{p(\mathbf{x}_a, \mathbf{x}_b, \mathbf{x}_c, (y_a, y_b, y_c) \in \mathcal{Y}_1)}{\pi_{\mathrm{T}}}. \tag{5.19}$$

Moreover, the distribution $p(\mathbf{x}_a, \mathbf{x}_b, \mathbf{x}_c, (y_a, y_b, y_c) \in \mathcal{Y}_1)$ at the numerator of Equation 5.19 can be explicitly expressed as

$$\begin{aligned} &p(\mathbf{x}_a, \mathbf{x}_b, \mathbf{x}_c, (y_a, y_b, y_c) \in \mathcal{Y}_1) \\ =&\pi_+^3 p_+(x_a)p_+(x_b)p_+(x_c) + \pi_+^2\pi_- p_+(x_a)p_+(x_b)p_-(x_c) + \\ &\pi_+\pi_-^2 p_+(x_a)p_-(x_b)p_-(x_c) + \pi_+^2\pi_- p_-(x_a)p_+(x_b)p_+(x_c) + \\ &\pi_+\pi_-^2 p_-(x_a)p_-(x_b)p_+(x_c) + \pi_-^3 p_-(x_a)p_-(x_b)p_-(x_c), \end{aligned} \tag{5.20}$$

from the assumption that three instances in each triplet comparison is generated independently.

Similarly, the underlying density for data of $\mathcal{D}_2$ can be expressed as

$$\begin{aligned} p_2(\mathbf{x}_a, \mathbf{x}_b, \mathbf{x}_c) &= p(\mathbf{x}_a, \mathbf{x}_b, \mathbf{x}_c | (y_a, y_b, y_c) \in \mathcal{Y}_2) \\ &= \frac{p(\mathbf{x}_a, \mathbf{x}_b, \mathbf{x}_c, (y_a, y_b, y_c) \in \mathcal{Y}_2)}{p((y_a, y_b, y_c) \in \mathcal{Y}_2)} \\ &= \frac{\pi_+^2\pi_- p_+(x_a)p_-(x_b)p_+(x_c) + \pi_+\pi_-^2 p_-(x_a)p_+(x_b)p_-(x_c)}{\pi_+^2\pi_- + \pi_+\pi_-^2} \\ &= \pi_+ p_+(x_a)p_-(x_b)p_+(x_c) + \pi_- p_-(x_a)p_+(x_b)p_-(x_c). \end{aligned} \tag{5.21}$$

□

5.5 Proof of Theorem 6

Theorem. *Samples in $\mathcal{D}_{1,a}$, $\mathcal{D}_{1,c}$, $\mathcal{D}_{2,a}$ and $\mathcal{D}_{2,c}$ can be considered to be independently drawn from*

$$\tilde{p}_1(x) = \pi_+ p_+(x) + \pi_- p_-(x), \tag{5.22}$$

samples in $\mathcal{D}_{1,b}$ can be considered to be independently drawn from

$$\tilde{p}_2(x) = \frac{(\pi_+^3 + 2\pi_+^2\pi_-)p_+(x) + (2\pi_+\pi_-^2 + \pi_-^3)p_-(x)}{\pi_{\mathrm{T}}}, \tag{5.23}$$

and samples in $\mathcal{D}_{2,b}$ can be considered to be independently drawn from

$$\tilde{p}_3(x) = \pi_- p_+(x) + \pi_+ p_-(x). \tag{5.24}$$

Proof. For simplicity, we give the proof of $\mathcal{D}_{2,a}$ and the other 5 cases follow the similar proof. Noticing

$$\mathcal{D}_2 \underset{i.i.d.}{\sim} p_2(\mathbf{x}_a, \mathbf{x}_b, \mathbf{x}_c) = \pi_+ p_+(x_a)p_-(x_b)p_+(x_c) + \pi_- p_-(x_a)p_+(x_b)p_-(x_c). \tag{5.25}$$

In order to decompose the triplet comparison data distribution into pointwise distribution, we marginalize $p_2(\mathbf{x}_a, \mathbf{x}_b, \mathbf{x}_c)$ with respect to x_b and x_c:

$$\begin{aligned}
& \int p_2(\mathbf{x}_a, \mathbf{x}_b, \mathbf{x}_c) dx_b dx_c \\
= \;& \pi_+ p_+(x_a) \int p_-(x_b) dx_b \int p_+(x_c) dx_c + \pi_- p_-(x_a) \int p_+(x_b) dx_b \int p_-(x_c) dx_c \\
= \;& \pi_+ p_+(x_a) \int \frac{p(x_b, y=-1)}{p(y=-1)} dx_b \int \frac{p(x_c, y=+1)}{p(y=+1)} dx_c + \\
& \quad \pi_- p_-(x_a) \int \frac{p(x_b, y=+1)}{p(y=+1)} dx_b \int \frac{p(x_c, y=-1)}{p(y=-1)} dx_c \\
= \;& \pi_+ p_+(x_a) + \pi_- p_-(x_a) \\
= \;& \tilde{p}_1(x_a)
\end{aligned} \tag{5.26}$$

□

5.6 Proof of Lemma 7

Lemma. *We can express $p_+(x)$ and $p_-(x)$ in terms of $\tilde{p}_1(x)$, $\tilde{p}_2(x)$ and $\tilde{p}_3(x)$ as*

$$\begin{aligned}
p_+(x) &= \frac{1}{(ac - b^2)} \left((c\pi_+ - b\pi_-)\tilde{p}_1(x) + (cA - bB)\tilde{p}_2(x) + (c\pi_- - b\pi_+)\tilde{p}_3(x) \right), \\
p_-(x) &= \frac{1}{(ac - b^2)} \left((a\pi_- - b\pi_+)\tilde{p}_1(x) + (aB - bA)\tilde{p}_2(x) + (a\pi_+ - b\pi_-)\tilde{p}_3(x) \right),
\end{aligned}$$

provided $ac - b^2 \neq 0$ where

$$a \triangleq \pi_+^2 + A^2 + \pi_-^2, \quad b \triangleq 2\pi_+\pi_- + AB, \quad c \triangleq \pi_-^2 + B^2 + \pi_+^2.$$

Proof. Notice that the equation has an infinite number of solutions. Letting

$$T \triangleq \begin{bmatrix} \pi_+ & \pi_- \\ A & B \\ \pi_- & \pi_+ \end{bmatrix}, \tag{5.27}$$

we resort to finding the Moore-Penrose pseudo inverse [111, 121], which provides the minimum Euclidean norm solution to the above system of linear equations.

Let T^* denote the conjugate transpose. We have

$$T^*T = \begin{bmatrix} \pi_+^2 + A^2 + \pi_-^2 & 2\pi_+\pi_- + AB \\ 2\pi_+\pi_- + AB & \pi_-^2 + B^2 + \pi_+^2 \end{bmatrix} = \begin{bmatrix} a & b \\ b & c \end{bmatrix}. \tag{5.28}$$

In the next step, we need to take the inverse of the above 2×2 matrix. To achieve a proper inverse matrix, we need to introduce another assumption that $\pi_+ \neq \frac{1}{2}$, which guarantees $ac - b^2 \neq 0$. Then

$$(T^*T)^{-1} = \frac{1}{(ac - b^2)} \begin{bmatrix} c & -b \\ -b & a \end{bmatrix}. \tag{5.29}$$

Finally, the Moore-Penrose pseudo inverse is given by

$$(T^*T)^{-1}T^* = \frac{1}{(ac - b^2)} \begin{bmatrix} c\pi_+ - b\pi_- & cA - bB & c\pi_- - b\pi_+ \\ -b\pi_+ + a\pi_- & -bA + aB & -b\pi_- + a\pi_+ \end{bmatrix}. \tag{5.30}$$

Thus, we can express $p_+(x)$ and $p_-(x)$ in terms of $\tilde{p}_1(x)$, $\tilde{p}_2(x)$ and $\tilde{p}_3(x)$ as

$$\begin{aligned} p_+(x) &= \frac{1}{(ac - b^2)} \left((c\pi_+ - b\pi_-)\tilde{p}_1(x) + (cA - bB)\tilde{p}_2(x) + (c\pi_- - b\pi_+)\tilde{p}_3(x) \right), \\ p_-(x) &= \frac{1}{(ac - b^2)} \left((a\pi_- - b\pi_+)\tilde{p}_1(x) + (aB - bA)\tilde{p}_2(x) + (a\pi_+ - b\pi_-)\tilde{p}_3(x) \right). \end{aligned} \tag{5.31}$$

□

5.7 Proof of Theorem 8

Theorem. *The classification risk can be equivalently expressed as*

$$\begin{aligned} R(f) = \frac{1}{(ac - b^2)} \{ & \mathop{\mathbb{E}}_{x \sim \tilde{p}_1(x)} [\pi_{\text{test}}(c\pi_+ - b\pi_-)\, \ell_+(x) + (1 - \pi_{\text{test}})(a\pi_- - b\pi_+)\, \ell_-(x)] + \\ & \mathop{\mathbb{E}}_{x \sim \tilde{p}_2(x)} [\pi_{\text{test}}(cA - bB)\, \ell_+(x) + (1 - \pi_{\text{test}})(aB - bA)\, \ell_-(x)] + \\ & \mathop{\mathbb{E}}_{x \sim \tilde{p}_3(x)} [\pi_{\text{test}}(c\pi_- - b\pi_+)\, \ell_+(x) + (1 - \pi_{\text{test}})(a\pi_+ - b\pi_-)\, \ell_-(x)] \}, \end{aligned}$$

where $\pi_{\text{test}} \triangleq p_{\text{test}}(y = +1)$ *denotes the class prior of the test dataset.*

Proof. Using Equation 4.15, we can rewrite the classification risk as

$$\begin{aligned} R_\ell(f) &= \mathop{\mathbb{E}}_{p(x,y)} [\ell(f(x), y)] \\ &= \pi_{\text{test}} \mathop{\mathbb{E}}_{p_+(x)} [\ell_+(x)] + (1 - \pi_{\text{test}}) \mathop{\mathbb{E}}_{p_-(x)} [\ell_-(x)] \\ &= \frac{\pi_{\text{test}}}{(ac - b^2)} \{ (c\pi_+ - b\pi_-) \mathop{\mathbb{E}}_{\tilde{p}_1(x)} [\ell_+(x)] + (cA - bB) \mathop{\mathbb{E}}_{\tilde{p}_2(x)} [\ell_+(x)] + \\ &\qquad (c\pi_- - b\pi_+) \mathop{\mathbb{E}}_{\tilde{p}_3(x)} [\ell_+(x)] \} + \\ &\quad \frac{1 - \pi_{\text{test}}}{(ac - b^2)} \{ (a\pi_- - b\pi_+) \mathop{\mathbb{E}}_{\tilde{p}_1(x)} [\ell_-(x)] + (aB - bA) \mathop{\mathbb{E}}_{\tilde{p}_2(x)} [\ell_-(x)] + \\ &\qquad (a\pi_+ - b\pi_-) \mathop{\mathbb{E}}_{\tilde{p}_3(x)} [\ell_-(x)] \}, \end{aligned} \tag{5.32}$$

which can be then simplified as Equation 4.16. □

5.8 Proof of Theorem 9

Theorem. *Assume the following holds:*

- *The loss function ℓ is ρ-Lipschitz with respect to the first argument ($0 < \rho < \infty$).*
- *All functions in the model class $\mathcal{F}$ are bounded, i.e., there exists a constant C_b such that $||f||_\infty \leq C_b$ for any $f \in \mathcal{F}$.*

Let $C_\ell \triangleq \sup_{t\in\{\pm 1\}} \ell(C_b, t)$. Then for any $\delta > 0$, with probability at least $1-\delta$ it holds that

$$R(\hat{f}) - R(f^*) \leq \left(\frac{2\rho C_{\mathcal{F}}}{\sqrt{n}} + \sqrt{\frac{C_\ell^2 \log \frac{2}{\delta}}{2n}} \right) \cdot \frac{C_R}{|ac - b^2|},$$

where

$$\begin{aligned} C_R =& |\pi_{\text{test}}(c\pi_+ - b\pi_-)| + |(1-\pi_{\text{test}})(a\pi_- - b\pi_+)| + |\pi_{\text{test}}(cA - bB)| + \\ & |(1-\pi_{\text{test}})(aB - bA)| + |\pi_{\text{test}}(c\pi_- - b\pi_+)| + |(1-\pi_{\text{test}})(a\pi_+ - b\pi_-)|. \end{aligned}$$

Proof. Letting

$$\begin{aligned} C_1 &\triangleq \frac{\pi_{\text{test}}}{(c\pi_+ - b\pi_-)(ac-b^2)}, \quad C_2 \triangleq \frac{1-\pi_{\text{test}}}{(a\pi_- - b\pi_+)(ac-b^2)}, \\ C_3 &\triangleq \frac{\pi_{\text{test}}}{(cA - bB)(ac-b^2)}, \quad C_4 \triangleq \frac{(1-\pi_{\text{test}})}{(aB - bA)(ac-b^2)}, \\ C_5 &\triangleq \frac{\pi_{\text{test}}}{(c\pi_- - b\pi_+)(ac-b^2)}, \quad C_6 \triangleq \frac{(1-\pi_{\text{test}})}{(a\pi_+ - b\pi_-)(ac-b^2)}, \end{aligned}$$

and

$$\begin{aligned} R_a(f) &= \mathop{\mathbb{E}}_{x\sim\tilde{p}_1(x)} [C_1 \ell(f(x), +1) + C_2 \ell(f(x), -1)], \\ R_b(f) &= \mathop{\mathbb{E}}_{x\sim\tilde{p}_2(x)} [C_3 \ell(f(x), +1) + C_4 \ell(f(x), -1)], \\ R_c(f) &= \mathop{\mathbb{E}}_{x\sim\tilde{p}_3(x)} [C_5 \ell(f(x), +1) + C_6 \ell(f(x), -1)], \end{aligned} \tag{5.33}$$

we can simplify the unbiased risk estimator info the form

$$R(f) = R_a(f) + R_b(f) + R_c(f). \tag{5.34}$$

Then

$$R(\hat{f}) - R(f^*) \leq 2 \sup_{f\in\mathcal{F}} |R_a(f) - \hat{R}_a(f)| + 2 \sup_{f\in\mathcal{F}} |R_b(f) - \hat{R}_b(f)| + 2 \sup_{f\in\mathcal{F}} |R_c(f) - \hat{R}_c(f)|.$$

For the first term,

$$\begin{aligned}
\sup_{f\in\mathcal{F}} |R_a(f) - \hat{R}_a(f)| &= \sup_{f\in\mathcal{F}} \left| \mathop{\mathbb{E}}_{p_a(x)} [C_1\ell(f(x),+1) + C_2\ell(f(x),-1)] - \frac{1}{n}\sum_{i=1}^{n} \hat{L} \right| \\
&\leq |C_1| \sup_{f\in\mathcal{F}} \left| \mathop{\mathbb{E}}_{p_a(x)} [\ell(f(x),+1)] - \frac{1}{n}\sum_{i=1}^{n} \ell(\widehat{f(x),+1}) \right| \\
&\quad + |C_2| \sup_{f\in\mathcal{F}} \left| \mathop{\mathbb{E}}_{p_a(x)} [\ell(f(x),-1)] - \frac{1}{n}\sum_{i=1}^{n} \ell(\widehat{f(x),-1}) \right| \\
&\leq |C_1| 2\mathcal{R} + |C_1| \sqrt{\frac{C_\ell^2 \log\frac{2}{\delta}}{2n}} + |C_2| 2\mathcal{R} + |C_2| \sqrt{\frac{C_\ell^2 \log\frac{2}{\delta}}{2n}} \\
&= (|C_1| + |C_2|) \left(\frac{2\rho C_{\mathcal{F}}}{\sqrt{n}} + \sqrt{\frac{C_\ell^2 \log\frac{2}{\delta}}{2n}} \right)
\end{aligned} \tag{5.35}$$

Combining three terms, Theorem 3 is proven. □

Chapter 6

Conclusion and Future work

In this chapter, we conclude the thesis and present several possible directions for future research.

6.1 Conclusion

This thesis is dedicated to investigating and expanding the possibility of learning from pairwise comparisons. In this context, we focus on the problem of binary classification, which is a fundamental problem for machine learning. Specifically, we investigated to what extent a comparison feedback can provide useful information for binary classification while maintaining user annotation feasibility by proposing a new form of comparison feedback and corresponding algorithms. On another direction, we investigate the possibility of using the triplet comparison *only* for binary classification by proposing a reasonable data generation process and a corresponding feasible ERM-based algorithm. We demonstrated the effectiveness of both approaches through user studies and extensive simulation studies.

Figure 6.1 shows how the methods proposed by this thesis is related to other methods on learning binary classifiers from the point of view of available supervision information. First, when most of explicit labels can be achieved accurately, we can rely on traditional standard methods of learning classifiers, being either passive or active. When this is not possible, we should verify whether explicit labels belong to one class is available. If it is the case, we can use methods such as positive-unlabeled learning mentioned before in weakly-supervised learning. If this is not available, we can turn to using alternative forms of feedback, which is the focus of this thesis. When we cannot interactively query annotation, the method proposed by Chapter 4 serves as a good candidate in this situation as it works on passively collected triplet comparison data alone. On the other hand, we need to further verify whether the task at hand is suitable for uncertainty comparison or not. This can be achieved by a small scale feasibility user study. If it is true, then the method proposed by Chapter 3 can be adopted for learning, otherwise the existing method of Xu et al. [159] can be used with the drawback of using unreliable feedback.

We summarize contributions of this thesis as follows.

- **Uncertainty comparison**: In Chapter 3, we first proposed the new form of pairwise uncertainty comparison feedback. The motivation for this new form of feedback is justified by user studies confirming its annotation feasibility, robustness, and accuracy. Then, we proposed a corresponding pseudo-label assigning algorithm that can actively selects pairs to query and provided theoretical justification by establishing an error rate bound under mild assumptions. We further develop it into insufficient budget

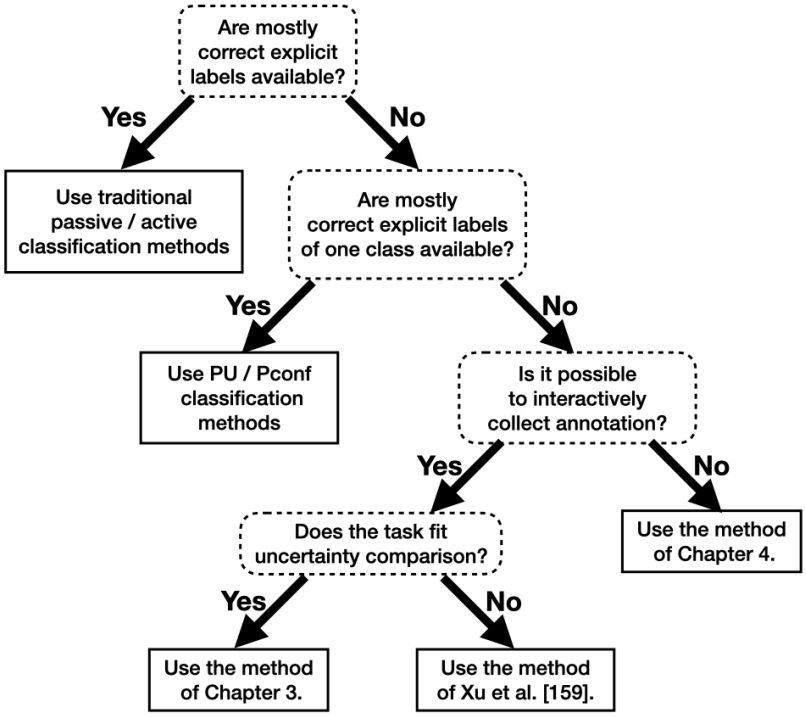


Figure 6.1: A flowchart for method selection.

cases by plugging into an active learning framework, and development corresponding theoretical justification. By extensive simulation studies, we confirmed the satisfying performance it can obtain when executed in an ideal setting.

- **Triplet comparison**: In Chapter 4, we realized binary classification using only triplet comparison data. Assuming knowing the class prior of the underlying data distribution and conducting proper matrix computation, we built an ERM-based classification risk estimator and established the generalization error bound of its empirical counterpart. User studies justified the motivation for using triplet comparison feedback for annotation and simulation studies confirmed its performance on larger scale of data.

From above results, we can conclude that various feasible form of comparison data can be used for binary classification, indicating the broader usage of alternative forms of feedback data in machine learning.

6.2 Future Work

In this section, we discuss and present several possible directions for future research.

- **Development of more useful feedback forms**: Considering the emerging of new application sciences, the characteristics of data in these corresponding new domains would be different from what we know. Therefore, it is necessary to develop new forms of feedback that is adaptable to the new data domain, user-friendly for large scale annotation, and most importantly robust to annotation noises.

- **Investigation on fundamental properties of comparison feedback for classification**: Uncertainty comparison and triplet comparison used by us focus on different aspects of classification information of each data point. Although we give an affirmative answer to whether we can conduct classification by the form of comparison feedback only, this is applicable for other feedback forms. This is to say, it is necessary to development a universal qualification tool for evaluating the usefulness of the information provided by a feedback form. Without accomplishing this, we cannot combine to use multiple forms of feedback effectively. The scope of this tool needs not to be restricted to the simplest binary case, or even to the classification problem.

- **A unified active learning framework from comparison data**: Active learning from alternative forms of comparisons still relies on designing an ad-hoc algorithm for each case, which is once again proved in Chapter 3. With the realization of the above quantitative analysis tool, it is necessary to develop a unified active learning framework for working with various forms of feedback, in order to achieve a better query complexity, thus be able to use the annotation resource more effectively.

From a unified point of view, this thesis takes the theoretical approach to a practical problem using indirect information when direction information is unavailable. However, theoretical wisdom may not be useful when being naively applied to practical problems. In the field of machine learning, problems such as prediction fairness or model interpretability rise when applying theoretically matured methods into real world applications. This shortcoming can be seen rising from only using benchmark datasets when evaluating algorithms. Therefore, extending the existing theoretical framework to also cover the behaviour of users is an essential and important future work to be considered. In the case of this thesis, problems may rise such as indirect feedback charges more burden to users, or different ways of asking for the same feedback give different results. In summary, a theory framework that take various aspects of user behaviour into consideration is expected.

References

[1] Sameer Agarwal, Josh Wills, Lawrence Cayton, Gert Lanckriet, David Kriegman, and Serge Belongie. Generalized non-metric multidimensional scaling. In *Artificial Intelligence and Statistics*, pages 11–18, 2007.

[2] Naomi S Altman. An introduction to kernel and nearest-neighbor nonparametric regression. *The American Statistician*, 46(3):175–185, 1992.

[3] Jesse Anderton and Javed Aslam. Scaling up ordinal embedding: A landmark approach. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 282–290, Long Beach, California, USA, 09–15 Jun 2019. PMLR.

[4] Jordan T Ash, Surbhi Goel, Akshay Krishnamurthy, and Dipendra Misra. Investigating the role of negatives in contrastive representation learning. *arXiv preprint arXiv:2106.09943*, 2021.

[5] Arthur Asuncion and David Newman. UCI machine learning repository, 2007.

[6] Ricardo Baeza-Yates, Berthier Ribeiro-Neto, et al. *Modern information retrieval*, volume 463. ACM press New York, 1999.

[7] Maria-Florina Balcan, Alina Beygelzimer, and John Langford. Agnostic active learning. *Journal of Computer and System Sciences*, 75(1):78–89, 2009.

[8] Maria-Florina Balcan, Avrim Blum, and Nathan Srebro. Improved guarantees for learning via similarity functions. 2008.

[9] Maria-Florina Balcan, Avrim Blum, and Nathan Srebro. A theory of learning with similarity functions. *Machine Learning*, 72(1):89–112, 2008.

[10] Maria Florina Balcan and Steve Hanneke. Robust interactive learning. In *Conference on Learning Theory*, pages 20–1, 2012.

[11] Han Bao, Yoshihiro Nagano, and Kento Nozawa. Sharp learning bounds for contrastive unsupervised representation learning. *arXiv preprint arXiv:2110.02501*, 2021.

[12] Han Bao, Gang Niu, and Masashi Sugiyama. Classification from pairwise similarity and unlabeled data. In *International Conference on Machine Learning*, pages 452–461. PMLR, 2018.

[13] Han Bao, Takuya Shimada, Liyuan Xu, Issei Sato, and Masashi Sugiyama. Similarity-based classification: Connecting similarity learning to binary classification. *arXiv preprint arXiv:2006.06207*, 2020.

[14] Horace Barlow. Redundancy reduction revisited. *Network: computation in neural systems*, 12(3):241, 2001.

[15] Peter L Bartlett, Michael I Jordan, and Jon D McAuliffe. Convexity, classification, and risk bounds. *Journal of the American Statistical Association*, 101(473):138–156, 2006.

[16] Peter L Bartlett and Shahar Mendelson. Rademacher and gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3(Nov):463–482, 2002.

[17] John Gilbert Beebe-Center, MS Rogers, and DN O'connell. Transmission of information about sucrose and saline solutions through the sense of taste. *The Journal of Psychology*, 39(1):157–160, 1955.

[18] Aurélien Bellet. Supervised metric learning with generalization guarantees. *arXiv preprint arXiv:1307.4514*, 2013.

[19] Aurélien Bellet, Amaury Habrard, and Marc Sebban. Metric learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 9(1):1–151, 2015.

[20] Viktor Bengs, Róbert Busa-Fekete, Adil El Mesaoudi-Paul, and Eyke Hüllermeier. Preference-based online learning with dueling bandits: A survey. *J. Mach. Learn. Res.*, 22:7–1, 2021.

[21] Alina Beygelzimer, Daniel J Hsu, John Langford, and Chicheng Zhang. Search improves label for active learning. In *Advances in Neural Information Processing Systems*, pages 3342–3350, 2016.

[22] Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.

[23] Olivier Bousquet and André Elisseeff. Stability and generalization. *The Journal of Machine Learning Research*, 2:499–526, 2002.

[24] Ralph Allan Bradley and Milton E Terry. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345, 1952.

[25] Mark Braverman, Jieming Mao, and S Matthew Weinberg. Parallel algorithms for select and partition with noisy comparisons. In *Proceedings of the forty-eighth annual ACM symposium on Theory of Computing*, pages 851–862, 2016.

[26] Kay Henning Brodersen, Cheng Soon Ong, Klaas Enno Stephan, and Joachim M Buhmann. The balanced accuracy and its posterior distribution. In *2010 20th international conference on pattern recognition*, pages 3121–3124. IEEE, 2010.

[27] Qiong Cao, Yiming Ying, and Peng Li. Distance metric learning revisited. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 283–298. Springer, 2012.

[28] Yuzhou Cao, Lei Feng, Senlin Shu, Yitian Xu, Bo An, Gang Niu, and Masashi Sugiyama. Multi-class classification from single-class data with confidences. *arXiv preprint arXiv:2106.08864*, 2021.

[29] Yuzhou Cao, Lei Feng, Yitian Xu, Bo An, Gang Niu, and Masashi Sugiyama. Learning from similarity-confidence data. *arXiv preprint arXiv:2102.06879*, 2021.

[30] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *arXiv preprint arXiv:2006.09882*, 2020.

[31] Abhijnan Chakraborty, Bhargavi Paranjape, Sourya Kakarla, and Niloy Ganguly. Stop clickbait: Detecting and preventing clickbaits in online news media. In *Advances in Social Networks Analysis and Mining (ASONAM)*, pages 9–16. IEEE, 2016.

[32] Ratthachat Chatpatanasiri, Teesid Korsrilabutr, Pasakorn Tangchanachaianan, and Boonserm Kijsirikul. A new kernelization framework for mahalanobis distance learning algorithms. *Neurocomputing*, 73(10-12):1570–1579, 2010.

[33] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.

[34] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020.

[35] Yuxin Chen and Changho Suh. Spectral mle: Top-k rank aggregation from pairwise comparisons. In *International Conference on Machine Learning*, pages 371–380. PMLR, 2015.

[36] Davide Chicco. Siamese neural networks: An overview. *Artificial Neural Networks*, pages 73–94, 2021.

[37] Sumit Chopra, Raia Hadsell, and Yann LeCun. Learning a similarity metric discriminatively, with application to face verification. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 539–546. IEEE, 2005.

[38] Wei Chu and Zoubin Ghahramani. Preference learning with gaussian processes. In *International Conference on Machine Learning*, page 137–144, 2005.

[39] Tarin Clanuwat, Mikel Bober-Irizar, Asanobu Kitamoto, Alex Lamb, Kazuaki Yamamoto, and David Ha. Deep learning for classical japanese literature. *preprint*, 2018.

[40] Tarin Clanuwat, Mikel Bober-Irizar, Asanobu Kitamoto, Alex Lamb, Kazuaki Yamamoto, and David Ha. Deep learning for classical japanese literature. *preprint*, 2018.

[41] Jason V Davis, Brian Kulis, Prateek Jain, Suvrit Sra, and Inderjit S Dhillon. Information-theoretic metric learning. In *Proceedings of the 24th international conference on Machine learning*, pages 209–216, 2007.

[42] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[43] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics, 2019.

[44] Carl Doersch, Abhinav Gupta, and Alexei A Efros. Unsupervised visual representation learning by context prediction. In *Proceedings of the IEEE international conference on computer vision*, pages 1422–1430, 2015.

[45] Alexey Dosovitskiy, Philipp Fischer, Jost Tobias Springenberg, Martin Riedmiller, and Thomas Brox. Discriminative unsupervised feature learning with exemplar convolutional neural networks. *IEEE transactions on pattern analysis and machine intelligence*, 38(9):1734–1747, 2015.

[46] Marthinus Du Plessis, Gang Niu, and Masashi Sugiyama. Convex formulation for learning from positive and unlabeled data. In *International conference on machine learning*, pages 1386–1394. PMLR, 2015.

[47] Marthinus C Du Plessis, Gang Niu, and Masashi Sugiyama. Analysis of learning from positive and unlabeled data. *Advances in neural information processing systems*, 27:703–711, 2014.

[48] Marthinus Christoffel Du Plessis, Gang Niu, and Masashi Sugiyama. Clustering unclustered data: Unsupervised binary labeling of two datasets having different class balances. In *2013 Conference on Technologies and Applications of Artificial Intelligence*, pages 1–6. IEEE, 2013.

[49] Charles Elkan and Keith Noto. Learning classifiers from only positive and unlabeled data. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 213–220, 2008.

[50] Lei Feng, Senlin Shu, Nan Lu, Bo Han, Miao Xu, Gang Niu, Bo An, and Masashi Sugiyama. Pointwise binary classification with pairwise confidence comparisons. In *International Conference on Machine Learning*, pages 3252–3262. PMLR, 2021.

[51] Leon Festinger. A theory of social comparison processes. *Human relations*, 7(2):117–140, 1954.

[52] Nicholas Frosst, Nicolas Papernot, and Geoffrey Hinton. Analyzing and improving representations with the soft nearest neighbor loss. In *International Conference on Machine Learning*, pages 2012–2020. PMLR, 2019.

[53] Johannes Fürnkranz and Eyke Hüllermeier. Preference learning and ranking by pairwise comparison. In *Preference learning*, pages 65–82. Springer, 2010.

[54] Wei Gao, Bin-Bin Yang, and Zhi-Hua Zhou. On the resistance of nearest neighbor to random noisy labels. *arXiv preprint arXiv:1607.07526*, 2016.

[55] WR Garner. An informational analysis of absolute judgments of loudness. *Journal of experimental psychology*, 46(5):373, 1953.

[56] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. *arXiv preprint arXiv:1803.07728*, 2018.

[57] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

[58] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent: A new approach to self-supervised learning. *arXiv preprint arXiv:2006.07733*, 2020.

[59] Zheng-Chu Guo and Yiming Ying. Guaranteed classification via regularized similarity learning. *Neural computation*, 26(3):497–522, 2014.

[60] Michael Gutmann and Aapo Hyvärinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 297–304. JMLR Workshop and Conference Proceedings, 2010.

[61] Siavash Haghiri, Damien Garreau, and Ulrike Luxburg. Comparison-based random forests. In *International Conference on Machine Learning*, pages 1866–1875, 2018.

[62] Siavash Haghiri, Debarghya Ghoshdastidar, and Ulrike von Luxburg. Comparison based nearest neighbor search. *arXiv preprint arXiv:1704.01460*, 2017.

[63] Bo Han, Quanming Yao, Xingrui Yu, Gang Niu, Miao Xu, Weihua Hu, Ivor Tsang, and Masashi Sugiyama. Co-teaching: Robust training of deep neural networks with extremely noisy labels. In *NeurIPS*, pages 8535–8545, 2018.

[64] Steve Hanneke et al. Theory of disagreement-based active learning. *Foundations and Trends® in Machine Learning*, 7(2-3):131–309, 2014.

[65] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9729–9738, 2020.

[66] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[67] Eric Heim. *Efficiently and Effectively Learning Models of Similarity from Human Feedback*. PhD thesis, University of Pittsburgh, 2016.

[68] Charles AR Hoare. Quicksort. *The Computer Journal*, 5(1):10–16, 1962.

[69] Godfrey M Hochbaum. Certain personality aspects and pressures to uniformity in social group. *Unpublished doctoral thesis. Univ. of Minn*, 1953.

[70] Yu-Guan Hsieh, Gang Niu, and Masashi Sugiyama. Classification from positive, unlabeled and biased negative data. In *International Conference on Machine Learning*, pages 2820–2829. PMLR, 2019.

[71] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. PMLR, 2015.

[72] Takashi Ishida, Gang Niu, Weihua Hu, and Masashi Sugiyama. Learning from complementary labels. *arXiv preprint arXiv:1705.07541*, 2017.

[73] Takashi Ishida, Gang Niu, Aditya Menon, and Masashi Sugiyama. Complementary-label learning for arbitrary losses and models. In *International Conference on Machine Learning*, pages 2971–2980. PMLR, 2019.

[74] Takashi Ishida, Gang Niu, and Masashi Sugiyama. Binary classification from positive-confidence data. *arXiv preprint arXiv:1710.07138*, 2017.

[75] Daniel M Kane, Shachar Lovett, Shay Moran, and Jiapeng Zhang. Active classification with comparison queries. In *Annual Symposium on Foundations of Computer Science (FOCS)*, pages 355–366. IEEE, 2017.

[76] Masahiro Kato, Takeshi Teshima, and Junya Honda. Learning from positive and unlabeled data with a selection bias. In *International conference on learning representations*, 2018.

[77] Mahmut Kaya and Hasan Şakir Bilge. Deep metric learning: A survey. *Symmetry*, 11(9):1066, 2019.

[78] Christopher Kent and Koen Lamberts. An exemplar account of the bow and set-size effects in absolute identification. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 31(2):289, 2005.

[79] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. *arXiv preprint arXiv:2004.11362*, 2020.

[80] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[81] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[82] Ryuichi Kiryo, Gang Niu, Marthinus C du Plessis, and Masashi Sugiyama. Positive-unlabeled learning with non-negative risk estimator. *arXiv preprint arXiv:1703.00593*, 2017.

[83] Matthäus Kleindeßner. *Machine Learning in a Setting of Ordinal Distance Information*. PhD thesis, Eberhard Karls Universität Tübingen, 2017.

[84] Matthäus Kleindessner and Ulrike von Luxburg. Kernel functions based on triplet comparisons. In *Advances in Neural Information Processing Systems*, pages 6807–6817, 2017.

[85] Matthäus Kleindessner and Ulrike Von Luxburg. Lens depth function and k-relative neighborhood graph: versatile tools for ordinal data analysis. *The Journal of Machine Learning Research*, 18(1):1889–1940, 2017.

[86] ET Klemmer and Frederick C Frick. Assimilation of information from dot and matrix patterns. *Journal of Experimental Psychology*, 45(1):15, 1953.

[87] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *4th International IEEE Workshop on 3D Representation and Recognition (3dRR-13)*, Sydney, Australia, 2013.

[88] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. *preprint*, 2009.

[89] Brian Kulis et al. Metric learning: A survey. *Foundations and trends in machine learning*, 5(4):287–364, 2012.

[90] Brian Kulis, Mátyás A Sustik, and Inderjit S Dhillon. Low-rank kernel learning with bregman matrix divergences. *Journal of Machine Learning Research*, 10(2), 2009.

[91] James T Kwok and Ivor W Tsang. Learning with idealized kernels. In *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*, pages 400–407, 2003.

[92] Yongchan Kwon, Wonyoung Kim, Masashi Sugiyama, and Myunghee Cho Paik. Principled analytic classifier for positive-unlabeled learning via weighted integral probability metric. *Machine Learning*, 109(3):513–532, 2020.

[93] Yves Lacouture and AAJ Marley. Choice and response time processes in the identification and categorization of unidimensional stimuli. *Perception & psychophysics*, 66(7):1206–1226, 2004.

[94] Phuc H Le-Khac, Graham Healy, and Alan F Smeaton. Contrastive representation learning: A framework and review. *IEEE Access*, 2020.

[95] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[96] Katia Levecque, Frederik Anseel, Alain De Beuckelaer, Johan Van der Heyden, and Lydia G isle. Work organization and mental health problems in phd students. *Research Policy*, 46(4):868–879, 2017.

[97] Chong Liu, Kui Wu, and Tian He. Sensor localization with ring overlapping based on comparison of received signal strength indicator. In *2004 IEEE International Conference on Mobile Ad-hoc and Sensor Systems (IEEE Cat. No. 04EX975)*, pages 516–518. IEEE, 2004.

[98] Andrea Locatelli, Maurilio Gutzeit, and Alexandra Carpentier. An optimal algorithm for the thresholding bandit problem. In *International Conference on Machine Learning*, pages 1690–1698. PMLR, 2016.

[99] Nan Lu, Shida Lei, Gang Niu, Issei Sato, and Masashi Sugiyama. Binary classification from multiple unlabeled datasets via surrogate set classification. In *International Conference on Machine Learning*, pages 7134–7144. PMLR, 2021.

[100] Nan Lu, Gang Niu, Aditya Krishna Menon, and Masashi Sugiyama. On the minimal supervision for training any binary classifier from only unlabeled data. *arXiv preprint arXiv:1808.10585*, 2018.

[101] Nan Lu, Tianyi Zhang, Gang Niu, and Masashi Sugiyama. Mitigating overfitting in supervised classification from two unlabeled datasets: A consistent risk correction approach. In *International Conference on Artificial Intelligence and Statistics*, pages 1115–1125. PMLR, 2020.

[102] R Duncan Luce. *Individual choice behavior: A theoretical analysis*. Courier Corporation, 2012.

[103] J. Macqueen. Some methods for classification and analysis of multivariate observations. In *In 5-th Berkeley Symposium on Mathematical Statistics and Probability*, pages 281–297, 1967.

[104] P. C. MAHALANOBIS. On the generalized distance in statistics. *Proceedings of the National Institute of Sciences (Calcutta)*, 2:49–55, 1936.

[105] Enno Mammen, Alexandre B Tsybakov, et al. Smooth discrimination analysis. *The Annals of Statistics*, 27(6):1808–1829, 1999.

[106] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[107] George A Miller. The magical number seven, plus or minus two: Some limits on our capacity for processing information. *Psychological review*, 63(2):81, 1956.

[108] Tom Michael Mitchell. *The discipline of machine learning*, volume 9. Carnegie Mellon University, School of Computer Science, Machine Learning ⋯, 2006.

[109] Soheil Mohajer, Changho Suh, and Adel Elmahdy. Active learning for top-k rank aggregation from noisy comparisons. In *International Conference on Machine Learning*, pages 2488–2497, 2017.

[110] Stefan Mojsilovic and Antti Ukkonen. *Relative distance comparisons with confidence judgements*, pages 459–467.

[111] Eliakim H Moore. On the reciprocal of the general algebraic matrix. *Bull. Am. Math. Soc.*, 26:394–395, 1920.

[112] Kevin Musgrave, Serge Belongie, and Ser-Nam Lim. A metric learning reality check. In *European Conference on Computer Vision*, pages 681–699. Springer, 2020.

[113] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 807–814, 2010.

[114] Gang Niu, Bo Dai, Makoto Yamada, and Masashi Sugiyama. Information-theoretic semi-supervised metric learning via entropy regularization. *Neural computation*, 26(8):1717–1762, 2014.

[115] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *European conference on computer vision*, pages 69–84. Springer, 2016.

[116] Mehdi Noroozi, Hamed Pirsiavash, and Paolo Favaro. Representation learning by learning to count. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5898–5906, 2017.

[117] Kento Nozawa and Issei Sato. Understanding negative samples in instance discriminative self-supervised representation learning. *arXiv preprint arXiv:2102.06866*, 2021.

[118] Hyun Oh Song, Yu Xiang, Stefanie Jegelka, and Silvio Savarese. Deep metric learning via lifted structured feature embedding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4004–4012, 2016.

[119] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[120] Omkar M. Parkhi, Andrea Vedaldi, Andrew Zisserman, and C. V. Jawahar. Cats and dogs. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

[121] B Y R. Penrose and Communicated J. A. Todd. A generalized inverse for matrices, 1954.

[122] Michaël Perrot and Ulrike von Luxburg. Boosting for comparison-based learning. *arXiv preprint arXiv:1810.13333*, 2018.

[123] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

[124] Irwin Pollack. The information of elementary auditory displays. *The Journal of the Acoustical Society of America*, 24(6):745–749, 1952.

[125] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. *arXiv preprint arXiv:2103.00020*, 2021.

[126] Henry Reeve and Ata Kaban. Fast rates for a kNN classifier robust to unknown asymmetric label noise. In *International Conference on Machine Learning*, pages 5401–5409, 2019.

[127] Tomoya Sakai, Marthinus Christoffel Plessis, Gang Niu, and Masashi Sugiyama. Semi-supervised classification based on classification from positive and unlabeled data. In *International conference on machine learning*, pages 2998–3006. PMLR, 2017.

[128] Ruslan Salakhutdinov and Geoff Hinton. Learning a nonlinear embedding by preserving class neighbourhood structure. In *Artificial Intelligence and Statistics*, pages 412–419. PMLR, 2007.

[129] Nikunj Saunshi, Orestis Plevrakis, Sanjeev Arora, Mikhail Khodak, and Hrishikesh Khandeparkar. A theoretical analysis of contrastive unsupervised representation learning. In *International Conference on Machine Learning*, pages 5628–5637. PMLR, 2019.

[130] Bernhard Schölkopf, Alexander J Smola, Francis Bach, et al. *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press, 2002.

[131] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823, 2015.

[132] Erich Schubert and Peter J Rousseeuw. Faster k-medoids clustering: improving the pam, clara, and clarans algorithms. In *International Conference on Similarity Search and Applications*, pages 171–187. Springer, 2019.

[133] Matthew Schultz and Thorsten Joachims. Learning a distance metric from relative comparisons. *Advances in neural information processing systems*, 16:41–48, 2004.

[134] Burr Settles. Active learning literature survey. 2009.

[135] Nihar B Shah and Martin J Wainwright. Simple, robust and optimal ranking from pairwise comparisons. *The Journal of Machine Learning Research*, 18(1):7246–7283, 2017.

[136] Takuya Shimada, Han Bao, Issei Sato, and Masashi Sugiyama. Classification from pairwise similarities/dissimilarities and unlabeled data via empirical risk minimization. *Neural Computation*, 33(5):1234–1268, 2021.

[137] Kazuhiko Shinoda, Hirotaka Kaji, and Masashi Sugiyama. Binary classification from positive data with skewed confidence. *arXiv preprint arXiv:2001.10642*, 2020.

[138] Oriane Siméoni, Mateusz Budnik, Yannis Avrithis, and Guillaume Gravier. Rethinking deep active learning: Using unlabeled data at model training. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 1220–1227. IEEE, 2021.

[139] Josef Sivic and Andrew Zisserman. Efficient visual search of videos cast as text retrieval. *IEEE transactions on pattern analysis and machine intelligence*, 31(4):591–606, 2008.

[140] Kihyuk Sohn. Improved deep metric learning with multi-class n-pair loss objective. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett,

editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016.

[141] Bharath K Sriperumbudur, Kenji Fukumizu, Arthur Gretton, Bernhard Schölkopf, and Gert RG Lanckriet. On the empirical estimation of integral probability metrics. *Electronic Journal of Statistics*, 6:1550–1599, 2012.

[142] Neil Stewart, Gordon DA Brown, and Nick Chater. Absolute identification by relative judgment. *Psychological review*, 112(4):881, 2005.

[143] Masashi Sugiyama. Learning under non-stationarity: Covariate shift adaptation by importance weighting. In *Handbook of Computational Statistics*, pages 927–952. Springer, 2012.

[144] Balázs Szörényi, Róbert Busa-Fekete, Adil Paul, and Eyke Hüllermeier. Online rank elicitation for plackett-luce: A dueling bandits approach. 2015.

[145] Wilson L Taylor. "cloze procedure": A new tool for measuring readability. *Journalism quarterly*, 30(4):415–433, 1953.

[146] Alexander B Tsybakov. Optimal aggregation of classifiers in statistical learning. *The Annals of Statistics*, 32(1):135–166, 2004.

[147] Laurens Van Der Maaten and Kilian Weinberger. Stochastic triplet embedding. In *2012 IEEE International Workshop on Machine Learning for Signal Processing*, pages 1–6. IEEE, 2012.

[148] Vladimir Vapnik. Statistical learning theory. *New York*, 3, 1998.

[149] Kilian Q Weinberger, John Blitzer, and Lawrence K Saul. Distance metric learning for large margin nearest neighbor classification. In *Advances in neural information processing systems*, pages 1473–1480, 2006.

[150] Kilian Q Weinberger and Lawrence K Saul. Fast solvers and efficient implementations for distance metric learning. In *Proceedings of the 25th international conference on Machine learning*, pages 1160–1167, 2008.

[151] Kilian Q Weinberger and Lawrence K Saul. Distance metric learning for large margin nearest neighbor classification. *Journal of machine learning research*, 10(2), 2009.

[152] Thomas A Wills. Downward comparison principles in social psychology. *Psychological bulletin*, 90(2):245, 1981.

[153] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *preprint*, 2017.

[154] Ling Xiao, Renfa Li, and Juan Luo. Sensor localization based on nonmetric multidimensional scaling. *STRESS*, 2:1, 2006.

[155] Eric Xing, Michael Jordan, Stuart J Russell, and Andrew Ng. Distance metric learning with application to clustering with side-information. *Advances in neural information processing systems*, 15:521–528, 2002.

[156] Liyuan Xu, Junya Honda, Gang Niu, and Masashi Sugiyama. Uncoupled regression from pairwise comparison data. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

[157] Miao Xu, Bingcong Li, Gang Niu, Bo Han, and Masashi Sugiyama. Revisiting sample selection approach to positive-unlabeled learning: Turning unlabeled data into positive rather than negative. *arXiv preprint arXiv:1901.10155*, 2019.

[158] Yichong Xu, Xi Chen, Aarti Singh, and Artur Dubrawski. Thresholding bandit problem with both duels and pulls. In *International Conference on Artificial Intelligence and Statistics*, pages 2591–2600. PMLR, 2020.

[159] Yichong Xu, Hongyang Zhang, Kyle Miller, Aarti Singh, and Artur Dubrawski. Noise-tolerant interactive learning using pairwise comparisons. In *Advances in Neural Information Processing Systems*, pages 2431–2440, 2017.

[160] Yiming Ying and Peng Li. Distance metric learning with eigenvalue optimization. *The Journal of Machine Learning Research*, 13(1):1–26, 2012.

[161] Baosheng Yu, Tongliang Liu, Mingming Gong, Changxing Ding, and Dacheng Tao. Correcting the triplet selection bias for triplet loss. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 71–87, 2018.

[162] Jure Zbontar, Li Jing, Ishan Misra, Yann Lecun, and Stephane Deny. Barlow twins: Self-supervised learning via redundancy reduction. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 12310–12320. PMLR, 18–24 Jul 2021.

[163] Changshui Zhang, Feiping Nie, and Shiming Xiang. A general kernelization framework for learning algorithms based on kernel pca. *Neurocomputing*, 73(4-6):959–967, 2010.

[164] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. arxiv preprint (in iclr 2017). *arXiv preprint arXiv:1611.03530*, 2016.

[165] Yivan Zhang, Nontawat Charoenphakdee, and Masashi Sugiyama. Learning from indirect observations. *arXiv preprint arXiv:1910.04394*, 2019.

[166] Yivan Zhang, Nontawat Charoenphakdee, Zhenguo Wu, and Masashi Sugiyama. Learning from aggregate observations. *arXiv preprint arXiv:2004.06316*, 2020.