Algorithms for Multiclass Learning with Corrupted Samples

Ioannis Iakovidis
7115142100008

Examination committee:
Christos Tzamos, Department of Informatics and Telecommunications, National and Kapodistrian University of Athens.
Dimitris Fotakis, Department of Electrical and Computer Engineering, National Technical University of Athens.
Aris T. Pagourtzis, Department of Electrical and Computer Engineering, National Technical University of Athens.

Supervisor:
Christos Tzamos, Associate Professor, Department of Informatics and Telecommunications, National and Kapodistrian University of Athens.

ABSTRACT

The success of modern Machine Learning Algorithms often relies on the availability of a large number of accurately labeled examples. However, the process of labeling examples is laborious and prone to unreliability in practice. As a result, extensive research has been conducted to develop Machine Learning Algorithms that can effectively handle corrupted samples. Nevertheless, previous works have primarily focused on addressing Binary Classification problems. In this thesis, we investigate Algorithms and Complexity for the challenges of Multiclass Classification with Coarse or Noisy labels. We examine the relationship between these problems and explore the computational complexity of these problems under different assumptions. Namely, we develop an algorithm for multiclass learning with agnostic label noise under the Gaussian distribution, which in terms is an algorithm for the Coarse label problem as well. Finally, we examine some special but commonly studied cases for the Coarse Label problem and develop polynomial time algorithms.

ΣΥΝΟΨΗ

Η επιτυχία των σύγχρονων Αλγορίθμων Μηχανικής Μάθησης συνήθως εξαρτάται από το γεγονός ότι υπάρχει ένας μεγάλος αριθμός σωστά ετικετοποιημένων παραδειγμάτων. Ωστόσο, στην πράξη, η διαδικασία ετικετοποίησης παραδειγμάτων είναι εξαιρετικά κοστοβόρα και συχνά μη αξιόπιστη. Για αυτό το λόγο υπάρχει ένας μεγάλος αριθμός έργων που αφορούν την ανάπτυξη Αλγορίθμων Μηχανικής Μάθησης που λειτουργούν με διεφθαρμένα δείγματα. Ωστόσο, οι προηγούμενες εργασίες απευθύνονται συνήθως σε προβλήματα Δυαδικής Κατηγοριοποίησης. Σε αυτήν τη διατριβή, μελετούμε Αλγορίθμους και Πολυπλοκότητα για τα προβλήματα Πολυταξικής Κατηγοριοποίησης με αδρές ή Θορυβώδεις ετικέτες. Εξετάζουμε τη σχέση ανάμεσα σε αυτά τα προβλήματα και διερευνούμε την υπολογιστική πολυπλοκότητα τους υπό διάφορες υποθέσεις. Συγκεκριμένα, αναπτύσσουμε έναν αλγόριθμο για πολυταξική μάθηση με αγνώστικούς θορύβους ετικέτας υπό την κανονική κατανομή, που συνεπάγει επίσης έναν αλγόριθμο για το πρόβλημα μάθησης με αδρές ετικέτες. Τέλος, εξετάζουμε μερικές ειδικές, αλλά συχνά μελετημένες περιπτώσεις για το πρόβλημα μάθησης με αδρές ετικέτες και αναπτύσσουμε αλγόριθμους πολυωνυμικού χρόνου.

ΕΥΧΑΡΙΣΤΙΕΣ

Θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή μου κ. Χρήστο Τζάμο και τους υποψήφιους διδάκτορες κ. Βασίλη Κοντονή και κ. Αλβέρτο Καλαβάση για την βοήθεια και την καθοδήγηση που μου παρείχαν μέσα από τις συμβουλές τους. Θα ήθελα επίσης να τους ευχαριστήσω για το χρόνο που αφιέρωσαν στις συζητήσεις μας για την υλοποίηση της εργασίας και για την καλύτερη εμβάθυνση στο θέμα. Θα ήθελα, ακόμη, να τους εκφράσω την ευγνωμοσύνη μου για όλες αυτές τις συζητήσεις που οδήγησαν στην διεύρυνση του τρόπου σκέψης μου και στην περαιτέρω εξέλιξη μου στον τομέα Είμαι επιπλέον ευγνώμων για την κατανόηση και το ενδιαφέρον που δείξανε για την εκπλήρωση των εξωσχολικών ακαδημαϊκών μου στόχων. Επίσης θα ήθελα να ευχαριστήσω τον καθηγητή κ. Δημήτρη Φωτάκη που με εισήγαγε σε αυτό τον τομέα και με σύστησε στο κ. Χρήστο Τζάμο. Ακόμα θα ήθελα να ευχαριστήσω όλα τα μέλη της τριμελούς επιτροπής κ. Χρήστο Τζάμο, κ. Δημήτριο Φωτάκη και κ. Αριστείδη Παγουρτζή για τις εποικοδομητικές συμβουλές και το χρόνο που αφιέρωσαν.

Τέλος, ευχαριστώ πάρα πολύ την οικογένεια μου, τους γονείς μου, την αδερφή μου, την κοπέλα μου Μελίτα και τους φίλους μου που ήταν και είναι πάντα δίπλα μου στηρίζοντας τις αποφάσεις μου και κάνοντας το καλύτερο για μένα με όλα τα μέσα.

CONTENTS

CHAPTER 1

INTRODUCTION

1.1 Probably Approximately Correct Model

In machine learning the problem of classification is to learn a concept function $c : \mathcal{X} \to [K], K \in \mathbb{N}$ from a collection of samples $\{x_i, c(x_i)\}_1^m$ that are drawn from a unknown distribution $x_i \sim D, \forall i \in [m]$. Or in other words, to construct an efficient algorithm A that according to the input sample outputs a function, or hypothesis, f_A that is close to c. However in order to formalize that goal, what we consider learnable, we have to further specify the above problem.

First of all, we need to define what we mean by close. For our case, the notion of closeness that is desired is to have a small probability of error on a random example. So we define the error of a function f under the distribution D as

$$\text{err}_D[f] = \Pr_{x \sim D}[f(x) \neq c(x)]$$

Also, another parameter of our problem should be a function space $\mathcal{H} \subseteq \{f \mid d : X \to [K]\}$ that we could search over. Clearly, this could not be done efficiently if $\mathcal{H}$ could be the space of all functions. Also if the learning space was too diverse (more complex) we would maybe have a lot of hypotheses that perfectly fit the data through interpolation but do much worse on future examples. Now our problem has taken a more concrete form:

$$\begin{aligned} \text{minimize}_f \quad & \text{err}_D[f] \\ & \text{s.t. } f \in \mathcal{H} \end{aligned}$$

However, as our function space $\mathcal{H}$ could be infinite it makes sense that we could only approximate f into a given accuracy ε. Furthermore, as our input is randomly sampled we could not be sure that we could even get a representative sample (for example there is a nonzero probability that our sample could be the same element of X). So we should allow for a probability of failure δ.

Another fact that we have to take into account is that we do not know the probability distribution D from which the instances are generated so our algorithm A has to perform well for all distributions D. Taking into account all of these considerations we have the following definition of learnability.

Definition 1.1 (PAC learnable)**.** A class of functions $\mathcal{H}$ is called PAC learnable if for every distribution D and $c \in \mathcal{H}$, there exists an algorithm A and a polynomial M such that, for every input $\varepsilon \geq 0$ and $\delta \geq 0$ (accuracy and probability of failure), if the algorithm is given as input a sample S of $m \geq M(\frac{1}{\varepsilon}, \frac{1}{\delta})$ idd pairs $\{(x_i, c(x_i))\}_1^m$, $x_i \sim D$ then it returns a hypothesis $h_{A(S)}$. Such that:

$$\Pr_{S \sim D^m} \left[\text{err}_D \left[h_{A(S)} \right] > \varepsilon \right] \leq \delta$$

If the runtime of the algorithm is polynomial with respect to $\frac{1}{\varepsilon}$ and $\frac{1}{\delta}$ then we call the class to be efficiently (or polynomially) PAC learnable. The above definition refers to the setting where the function that we seek to approximate belongs to our hypothesis class $\mathcal{H}$. This is called the realizable setting. However, in practice, we can not expect that we know the concept function space of a problem ahead of time. So all want is to approximate it with our search space as precisely as possible (agnostic setting).

Definition 1.2 (agnostically PAC learnable)**.** A class of functions $\mathcal{H}$ with domain $\mathcal{X}$ and image $\mathcal{Y}$ is called agnostically PAC learnable if, for every distribution D and $c : \mathcal{X} \to \mathcal{Y}$ function, there exists an algorithm A and a polynomial M such that, for every input $\varepsilon \geq 0$ and $\delta \geq 0$ (accuracy and probability of failure), if the algorithm is given as input a sample S of $m \geq M(\frac{1}{\varepsilon}, \frac{1}{\delta})$ idd pairs $\{(x_i, c(x_i))\}_1^m$, $x_i \sim D$ then it returns a hypothesis $h_{A(S)} \in \mathcal{H}$. Such that:

$$\Pr_{S \sim D^m} \left[\text{err}_D \left[h_{A(S)} \right] > \min_{\mathcal{H}} \text{err}_D \left[h \right] + \varepsilon \right] \leq \delta$$

In Statistical Learning Theory there is a theorem that bounds the number of examples needed by each class of functions in order for a learning algorithm to exist. These bound are achieved by the algorithm that achieves the minimum error in the input sample. Or else minimizes the empirical error:

$$\min_{\mathcal{H}} \sum_{1}^{m} \mathbb{1} \left(h(x_i) \neq y_i \right))$$

1.2 Efficent Realizable vs Agnostic Learning

In practice, the labeling process can be quite expensive and time-consuming. Hence even if we knew the precise concept class chances are that we could not be able to produce the number of samples that are needed in order to accurately learn it. And so we would have to resort to more sloppy labeling that could produce some errors and hence try to learn a concept agnostically outside of the class.

However minimizing the empirical error can not always be done efficiently, especially in the agnostic setting. In fact, learning agnostically even simple classes like halfspaces has been shown to be computationally hard [Dan15b]. As a result, we opt to study problems that are closer to the realizable setting.

Furthermore, we assume that the target concept is realizable within the class but we observe examples perturbed with some random process [MN06], [AL88]. In that case, we can get efficient algorithms for some classes in much more complex problems than realizable PAC learning. In a way these models of noise as a way to study the Spectrum between realizable and agnostic learning.

Another way to overcome the computational intractability of Agnostic Learning is to look at distribution-specific algorithms. For the cases where the samples are drawn

from well-behaved distributions like the Uniform, Normal, or Log-concave distributions Polynomial Time Approximations Schemes have been developed.

1.3 Learning with Coarse Labels

In this thesis, we will study the problem of Learning with Coarse Labels which is a weak supervision learning problem. In the following paragraphs, we will define the problem and summarize the existing algorithmic techniques. Our main motivation is the recent result [Fot+21] that produced an efficient algorithm for a wide range of corruptions.

Specifically, we will investigate the problem when one has to learn with samples of the form (x, S) where S is a subset of the label set $\mathcal{Y}$ containing the true label of x. We say that S is, therefore, a coarse label and our goal is to learn with good accuracy on the ground truth labels, a schematic representation of the problem can be found in figure 1.1. A motivation for this problem is that for labeling it can be that we ask the experts to identify a subset of the labels that the ground truth label belongs to, for example by asking yes or no questions if a random label is the correct one. As a result, because the problem set to the experts is more simple we can reduce the labeling cost as well as the label noise.

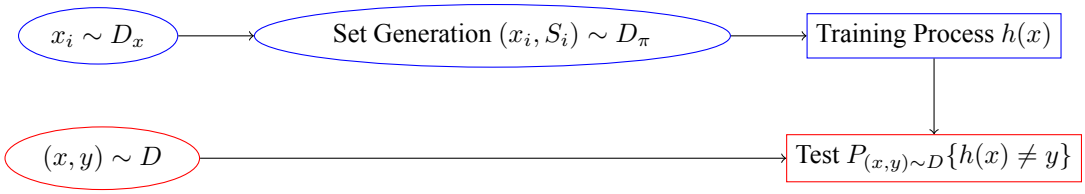

Figure 1.1: Learning With Coarse Labels

Of course, this problem would not be feasible if the subsets could be arbitrary, as some labels could never be separated and thus there would be no way to distinguish between them. So one needs an assumption that makes the sets fine enough in order for inference to be possible. Hence we have to formally define the set generation process in order to give a useful but doable condition. Specifically, we define the Coarse example generative process as follows.

Definition 1.3 (Generative Process for Coarse Examples)**.** Let $\mathcal{X}$ be an arbitrary domain, and let $\mathcal{Z} = \{1, \ldots, K\}$ be the discrete domain of all possible fine labels. We generate coarsely labeled examples as follows:

1. Draw a finely labeled example (x, z) from a distribution D on $\mathcal{X} \times \mathcal{Z}$.
2. Draw a coarsening partition $\mathcal{S}$ (of $\mathcal{Z}$) from a distribution π.
3. Find the unique set $S \in \mathcal{S}$ that contains the fine label z.
4. Observe the coarsely labeled example (x, S).

We denote D_π the distribution of the coarsely labeled example (x, S).

Notice that the ground truth label always belongs in the set and the marginal on x is the distribution D_x in both coarse and fine examples. Also, the coarsening partition

distribution is the same for all instances, or in other words we have that the observed set is conditionally independent of x when we condition on the fine label. Also as mentioned previously the partition distribution could be too coarse and not separate some labels at all and hence loss all of the information the original distribution had on associated labels as a result in such a case learning would not be possible. Thus we will study partition distributions that are information preserving i.e. the property that not much information is lost no matter what the fine label distribution is.

Definition 1.4 (Information Preserving Partition Distribution)**.** Let $\mathcal{Z}$ be any domain and let $\alpha \in (0, 1]$. We say that π is an α-information preserving partition distribution if for every two distributions D^1, D^2 supported on $\mathcal{Z}$, it holds that $\text{TV}\left(D^1_\pi, D^2_\pi\right) \geq \alpha \cdot \text{TV}\left(D^1, D^2\right)$, where $\text{TV}\left(D^1, D^2\right)$ is the total variation distance of D^1 and D^2.

Intuitively α is the fraction of information that has been preserved, as we have a corruption process $\alpha \leq 1$ so $1 - \alpha$ is the amount of information lost.

1.4 Statistical Query Reduction

The guarantee shown in [Fot+21] is that every algorithm that belongs to a certain model that learns given fine labeled examples can be simulated given coarse examples when the partition distribution is information preserving. Specifically, it is possible to efficiently simulate every algorithm in the Statistical Query (SQ) model by simulating the SQ oracle in polynomial time.

Definition 1.5 (Learnable in the Statistical Query Model [Kea98])**.** We say that a class $\mathcal{H}$ is PAC learnable in the SQ model if there exists an algorithm A and a polynomial p such that for all $c \in H$ target concepts and distributions D given access to an SQ oracle $\text{Stat}_D(q, \tau)$ with inputs ε, δ the following hold:

1. For any query (q, τ) query made by A, q is computable in $p(\frac{1}{\varepsilon}, \frac{1}{\delta})$ time and $\frac{1}{\tau}$ is bounded above by $p(\frac{1}{\varepsilon}, \frac{1}{\delta})$.
2. A halts in time $p(\frac{1}{\varepsilon}, \frac{1}{\delta})$.
3. And for the output of A, h we have that $\text{err}_D[h] \leq \varepsilon$ with probability at least $1 - \delta$.

Where $\text{Stat}_D(\chi, \tau)$ is an oracle that computes given a query (q, τ) returns $\mathop{\mathbb{E}}_{x \sim D}[q(x, c(x))]$ up to accuracy τ. The above definition can be easily modified for the agnostic case.

Essentially algorithms in this model do not have direct access to examples from the distribution D but only to statistics taken from D. Hence if we could simulate the SQ oracle using coarse examples we could run every algorithm in the SQ model. This model covers a wide variety of robust learning algorithms used in practice like Stochastic Gradient Descent.

Let (q, τ) an SQ it is true that:

$$\mathop{\mathbb{E}}_{(x,y)\sim D}[q(x, y)] = \sum_{1}^{k} \mathop{\mathbb{E}}_{(x,y)\sim D}[q(x, i)\,\mathbb{1}(y = i)]$$

And

$$\begin{aligned}\mathop{\mathbb{E}}_{(x,y)\sim D}[q(x,i)\mathbb{1}(y=i)] &= \int_X q(x,i)D(x,i)dx \\ &= \int_X q(x,i)D_x(x)D(i \mid x)dx\end{aligned}$$

By rejection sampling according to $q(x,i)$ we can get samples from the distribution:

$$D_x^f(x) = \frac{q(x,i)}{\mathop{\mathbb{E}}_{x\sim D_x}[q(x,i)]}D_x(x)$$

So we have:

$$\begin{aligned}\mathop{\mathbb{E}}_{(x,y)\sim D}[q(x,i)] &= \int_X \frac{q(x,i)}{D_x^f(x)}D_x(x)D_x^f(x)D(i \mid x)dx \\ &= \int_X \mathop{\mathbb{E}}_{x\sim D_x}[q(x,i)]\, D_x^f(x)D(i \mid x)dx \\ &= \mathop{\mathbb{E}}_{x\sim D_x}[q(x,i)]\,\mathrm{Pr}_{z\sim D_z^f}[z=i]\end{aligned}$$

Where D_z^f is the distribution of labels over accepted samples. So it still suffices to estimate $\mathrm{Pr}_{z\sim D_z^f}[z=i]$. The expected value of the query function is independent of the partition and can be estimated by unlabeled examples. Hence learning in this model has been reduced to estimating probabilities of having i as the label in subsampled distributions, an unsupervised problem.

1.5 Simulating the SQ oracle

To simulate the SQ oracle with coarse samples we need to be able to compute $\mathrm{Pr}_{z\sim D_z^f}[z=i]$ for different rejection sampling functions f. The authors of [Fot+21] showed that because the set generation process is information preserving we can do this by only processing the label subsets and not the instances. Hence we will solve the problem: Let D probability distribution on a set $[K]$, $K \in \mathbb{N}$ given samples from the following generative processes, $S \sim D_\pi$, estimate the probability of observing $i \in [K]$ from D up to accuracy ε.

Definition 1.6 (Generative process of Coarse Samples)**.** Let π an information preserving distribution of partitions over $[K]$, $K \in \mathbb{N}$. Consider the following process:

1. Draw a sample z from D.
2. Draw a coarsening partition P of $[K]$ from the distribution π
3. Observe $S \in P$ such that $z \in S$.

We denote the distribution of S as D_π

We can do this by Empirical Likelihood Maximization. The corresponding empirical log-likelihood objective after drawing N independent samples $S_1, \ldots, S_N$ from D_π is given by

$$\mathcal{L}_N(\boldsymbol{p}) = \frac{1}{N}\sum_{n=1}^{N}\log\left(\sum_{i\in S_n}\boldsymbol{p}_i\right).$$

Using concentration results one can make the total variation distance from the optimum of the $\mathcal{L}_N$ and the true probability vector of D smaller than ε with probability at least $1-\delta$ using N polynomial in $K, \frac{1}{\delta}, \frac{1}{\varepsilon}$. Thus we have the following theorem:

Theorem 1.7 (Proposition 7 [Fot+21])**.** Let $[K], K \in \mathbb{N}$ be a discrete domain and let D be a distribution supported on $[K]$. Moreover, let π be an α-information preserving partition distribution for some $\alpha \in (0,1]$. Then, with $N = \widetilde{O}\left(K/\left(\epsilon^2\alpha^2\right)\log(1/\delta)\right)$ samples from D_π and in time polynomial in the number of samples N, we can compute a distribution $\widetilde{D}$ supported on $[K]$ such that $\mathrm{TV}(\widetilde{D}, D) \leq \epsilon$

1.6 Overall Algorithm

Hence the overall algorithm runs the steps of a predefined SQ algorithm and for each query, it performs K mean estimations by the sample averages and K Empirical Likelihood Maximizations for subsampled distributions and then combines the results as shown before. From concentration results, one can show that the number of examples and hence the complexity is polynomial in $\frac{1}{\delta}, \frac{1}{\varepsilon}$. From this, we get the following results:

Theorem 1.8 (SQ from Coarsely Labeled Examples)**.** Consider a distribution D_π over coarsely labeled examples in $\mathbb{R}^d \times [K]$, with α-information preserving partition distribution π. Let $q : \mathbb{R}^d \times [K] \to [-1,1]$ be a query function, that can be evaluated on any input in time T, and $\tau, \delta \in (0,1)$. There exists an algorithm (Algorithm 1), that draws $N = \widetilde{O}\left(K^4/\left(\tau^3\alpha^2\right)\log(1/\delta)\right)$ coarsely labeled examples from D_π and, in poly (N,T) time, computes an estimate $\hat{r}$ such that, with probability at least $1-\delta$, it holds $\left|\mathop{\mathbb{E}}_{(x,z)\sim D}[q(x,z)] - \hat{r}\right| \leq \tau$.

Some important properties of the algorithm above are that it works for any hypothesis class that achieves an SQ algorithm, also it is agnostic on the partition distribution π, and assumes only information preservation.

1.7 Thesis Overview

However, this result does not necessarily hold for the setting where π is not an information-preserving partition or the coarsening generative process is instance-dependent. These are the main questions that we will be concerned with in this thesis.

Specifically, the thesis is structured in the following way:

- In chapter 2 we will see how learning problems under the presence of corruption with different assumptions are related to each other.
- In chapter 3 we study algorithms for learning under RCN.
- In chapter 4 we will show a reduction of the coarse labels learning to a simpler case where the combinatorial structure of the set is irrelevant.
- In chapter 5 we will design efficient algorithms for cases where there is a simple instance dependence in the coarsening generative process.
- In chapter 6 we will design a PTAS for agnostically learning multiclass linear classifiers.
- In chapter 7 we list further open problems that are of interest in this field.

CHAPTER 2

MAIN PROBLEMS

2.1 Problem Hierarchy

As we have seen the problem of learning with coarse labels has been solved with the assumption that the set generation process is information-preserving. And so one could recover the correct distribution label by looking only at the generated subsets [Fot+21]. We name this assumption α-IP (for α-information preserving). However, another weaker assumption is that the incorrect labels are not too frequent, [CST11]. We name this assumption ε-UB (that we have ε as an upper bound on the probability that an incorrect label belongs in the set).

However, in both cases these assumptions we have that the generated set does not depend on the point x but rather the ground truth label y. We can generalize both these assumptions by allowing the information-preserving constant and the probability bound to be variable with x but still having universal bounds for all x. This is parallel to learning with Massart noise [MN06]. We will name these problems $\alpha(x)$-IP and $\varepsilon(x)$-UB respectively. In the following figure, we see how those problems relate to each other. And in the coming paragraphs of this section, we will make an introduction and define each problem separately.

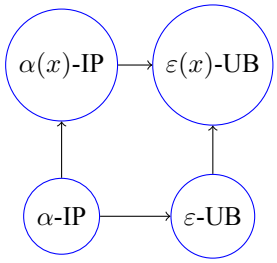

Figure 2.1: Problem Hierarchy

Notice that the problems form a lattice with edges from stronger to weaker assumptions. As in any such case, the existence of algorithms for the more general problems

solve also the more confined. Also hardness results in stronger assumptions carry to more general settings.

2.2 ε-UB Assumption

Definition 2.1 (ε-UB assumption)**.** We say a distribution, D_π, on $\mathcal{X} \times 2^K$ satisfies the ε-UB assumption if there exists an $\varepsilon < 1$ such that:

$$\Pr_{(x,S)\sim D_\pi}[z \in S \mid x] = \Pr_{(x,S)\sim D_\pi}[z \in S \mid x'] \leq \varepsilon, \forall x, x' \in \mathcal{X} : c(x) = c(x'), \forall z \neq c(x)$$

We assume that the ground truth labels are realizable by $c : \mathcal{X} \to [K]$. We will show later that solving this problem without the assumption of realizability is hard. When having this assumption we can solve the problem statistically by the analog of empirical risk minimization (superset risk minimization).

Theorem 2.2. If D_π satisfies the ε-UB assumption then we have that for all hypothesis h:

$$\Pr_{(x,S)\sim D_\pi}[h(x) \notin S] \leq \Pr_{(x,y)\sim D}[h(x) \neq y] \leq \frac{1}{1-\varepsilon} \Pr_{(x,S)\sim D_\pi}[h(x) \notin S]$$

Proof. The first inequality is trivial because for the prediction to be correct it must belong in the set. For the second we have that:

$$\begin{aligned}
\Pr_{(x,S)\sim D_\pi}[h(x) \notin S] &= \Pr_{(x,S)\sim D_\pi}[(h(x) \notin S) \wedge (h(x) \neq y)] \\
&= \Pr_{(x,S)\sim D}[(h(x) \neq y] \Pr_{(x,S)\sim D_\pi}[h(x) \notin S \mid h(x) \neq y] \\
&\geq \Pr_{(x,S)\sim D}[(h(x) \neq y]\,(1-\varepsilon)
\end{aligned}$$

□

Hence if someone finds a model that has a small probability of predicting outside the associated set (having a superset error), then he has a model with a low error. This theorem applies even in the instance-dependent setting (with assumption $\varepsilon(x)$-UB). And similar VC-dimension generalization results with respect to the empirical superset error apply also in this case.

Theorem 2.3. (Sample Complexity for Supperset Error [LD14]) Let $\theta = \log \frac{2}{1+\varepsilon}$ and suppose the Natarajan dimension of the hypothesis space H is d_H. Then if we have a sample S from D_π of size n, $n \geq n_0(H, \epsilon, \delta)$ where

$$n_0(H, \epsilon, \delta) = \frac{4}{\theta\epsilon}\left(d_H\left(\log(d_H) + 2\log K + \log\frac{1}{\theta\epsilon}\right) + \log\frac{1}{\delta} + 1\right)$$

Let ERM_S be any algorithm that minimizes the number of superset errors on S then $\text{err}_D[\text{ERM}_S(x)] \leq \epsilon$ with probability $1-\delta$

In this way, one can learn when given examples and sets whose distribution satisfies the ε-UB assumption. However the process of finding a model with minimum empirical superset error much like the ERM is not always efficient.

It is true that the ε-UB assumption is a generalization of the α-IP assumption. But there are examples that satisfy ε-UB but not α-IP. In the following example, we have four classes and for each example in a class, one of two sets is chosen with probability a half. This can be represented schematically or by a probability transition matrix of size $2^K \times K$ with $\Pr[S = T \mid y = i]$ being the elements.

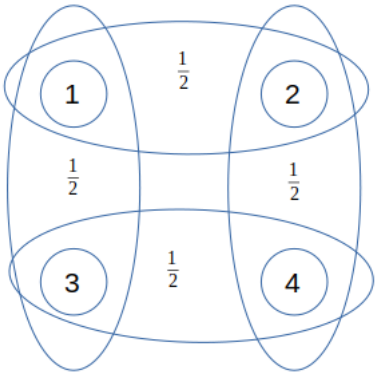

Figure 2.2: Example of a set generation process in ε-UB but not in α-IP

One can easily verify that this example does not satisfy the α-IP assumption as the probability distributions over the classes $(\frac{1}{2}, 0, 0, \frac{1}{2})$ and $(0, \frac{1}{2}, \frac{1}{2}, 0)$ both yield a uniform distribution over the sets. Moreover, observe that since we have two distributions with a positive total variation distance that map to the same distribution over subsets, given only the associated subsets the distribution over fine labels is not recoverable. Hence the unsupervised (statistical) problem is not solvable and the algorithm of [Fot+21] does not apply in this setting.

Also, observe that the above definition of the ε-UB setting was in the realizable setting. This was not for simplicity but for the fact that for the situation that we have soft labels, i.e. we have a probability for a label to be the ground truth for every $x \in \mathcal{X}$. We can have two distributions, for example, $(\frac{1}{2}, 0, 0, \frac{1}{2})$ and $(0, \frac{1}{2}, \frac{1}{2}, 0)$ in the above example that if they where the distribution of y given x they both lead to the same distribution on labels given x. Hence the problem is unidentifiable and so realizability is important for the validity of the problem.

2.3 ε-UB and Learning with RCN

The setting of learning with the ε-UB assumption is similar to learning with Random Classification Noise (RCN) [Kea98]. Where independent of the instance we have a fixed probability of the label being flipped to any other label. Let H a $K \times K$ matrix, the confusion matrix, such that $H_{ij} = \Pr[\tilde{y} = j \mid y = i]$, where $\tilde{y}$ is the observed label and y the ground truth. One can schematically represent the example generation process under this type of noise with the following figure.

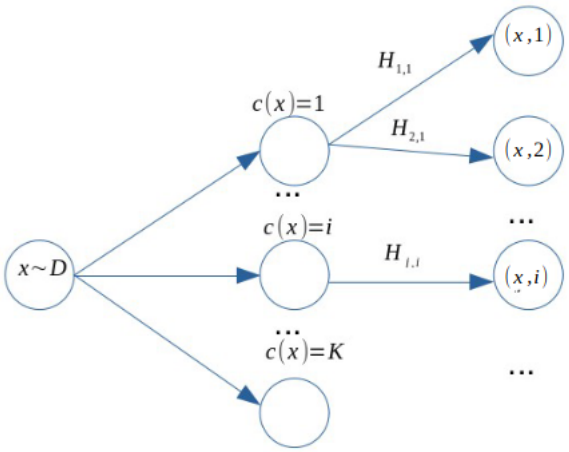

Figure 2.3: Example Generation Process under multiclass RCN noise

It is true that if one given coarse examples $\{(x, S)\}_1^m$ for each example keeps a random label in the set, $\hat{y}_i$. We would have a problem of learning with noise. Furthermore, as incorrect labels are less frequently in the set than the correct label we have that the probability that $\hat{y}_i$ is the correct label is greater than being any other one. One can prove that this is a reduction to learning with RCN with the additional assumption that the correct class is more probable. Of course, that kind of reduction loses some information about the problem, as the correct label is always in the set, and in a way we are amplifying the corruption process. But learning with RCN is a well-studied problem that is known to have efficient algorithms in a wide number of settings.

Theorem 2.4. There is an efficient reduction from learning with ε-UB coarse samples to learning with RCN.

Proof. Let $\{(x, S)\}_1^m$ be the coarse examples. By taking for each $i \in [m]$ $\hat{y}_i$ a random label from the set S_i. We can form a set of instance-label pairs $\{(x_i, \hat{y}_i)\}_1^m$. For which we can compute the probability that the observed label is j given that the ground truth label is i:

$$\begin{aligned}
\Pr[\widehat{y} = j \mid y = i, x] &= \Pr[\widehat{y} = j \mid y = i] \\
&= \mathop{\mathbb{E}}_{S}\left[\Pr[\widehat{y} = j \mid y = i, S]\right] \\
&= \mathop{\mathbb{E}}_{S}\left[\frac{1}{|S|}\mathbb{1}(j \in S) \mid y = i\right]
\end{aligned}$$

Let $H_{ij} = \Pr[\widehat{y} = j \mid y = i]$. As the instance is conditionally independent given the label we have a problem of learning under RCN with confusion matrix H.

Notice that this reduction preserves the instance distribution, as it does not modify the instances $\{x_i\}_1^m$. Also, we have that the probability that we observe the correct label

can be bounded away by a constant from the probability that we observe any other label.

$$\begin{aligned}
H_{i,j} &= \mathop{\mathbb{E}}_{S}\left[\frac{1}{|S|}\mathbb{1}\left(j \in S\right) \mid y = i\right] \\
&= \mathop{\mathbb{E}}_{S}\left[\frac{1}{|S|} \mid y = i\right] - \mathop{\mathbb{E}}_{S}\left[\frac{1}{|S|}\mathbb{1}\left(j \notin S\right) \mid y = i\right] \\
&= H_{i,i} - \mathop{\mathbb{E}}_{S}\left[\frac{1}{|S|}\mathbb{1}\left(j \notin S\right) \mid y = i\right] \\
&\leq H_{i,i} - \frac{1}{K}\mathop{\mathbb{E}}_{S}\left[\mathbb{1}\left(j \notin S\right) \mid y = i\right] \\
&\leq H_{i,i} - \frac{1-\varepsilon}{K}
\end{aligned}$$

□

Learning under RCN is a very widely studied topic that gave birth to learning with the framework of learning with Statistical Queries. And in general, the problem of learning under RCN is solved when an SQ algorithm exists for binary classification, as there is an algorithm to compute any Statistical Query from noisy data [Kea98]. However, for multiclass learning problems, there are some degenerate cases that are not yet understood (for further detail read 3).

2.4 Instance Dependent Set Generation

Both α-IP and ε-UB had the rather unrealistic assumption that the set generation process is the same for every instance $x \in \mathcal{X}$. This is definitely not the case in practice as one could easily get more noise closer to the decision boundary. Also, algorithms that operate with this kind of assumption can rely heavily on the knowledge of the unchanged rates and be in a way overtuned. Following we see the definitions for both settings.

Definition 2.5 ($\varepsilon(x)$-UB assumption)**.** We say a distribution, D_π, on $\mathcal{X} \times 2^K$ satisfies the $\varepsilon(x)$-UB assumption if there exists an $\varepsilon < 1$ such that:

$$\Pr_{(x,S)\sim D_\pi}\left[z \in S \mid x\right] \leq \varepsilon, \forall x \in \mathcal{X}, \forall z \neq c(x)$$

Definition 2.6 ($\alpha(x)$-IP assumption)**.** We say a distribution, D_π, on $\mathcal{X} \times 2^K$ satisfies the α-IP assumption. If there exists an $\alpha < 1$ and information preserving partition distributions, $\pi(x)$, for every $x \in \mathcal{X}$ with a constant less than α. That is composed with an example distribution D over $\mathcal{X} \times K$ generate D_π.

As in the case of learning ε-UB coarse samples, one can reduce both cases to learning with noise. But in this case, the noise rate is instance dependent. Specifically following the proof of the last paragraph one can easily show that both cases are reduced to learning with Massart noise.

However, distribution-independent PAC learning with Massart noise is known to be computationally hard even in simple classes like halfspaces. But one can rather get an error arbitrarily close to the noise threshold in polynomial time.

2.5 Further Observations

Summarizing we introduced three generalizations to the learning with coarse labels setting discussed in chapter 1 ([Fot+21]). And show their connections to the problem of learning with noise. In the following figure, we see the overall problem relationships.

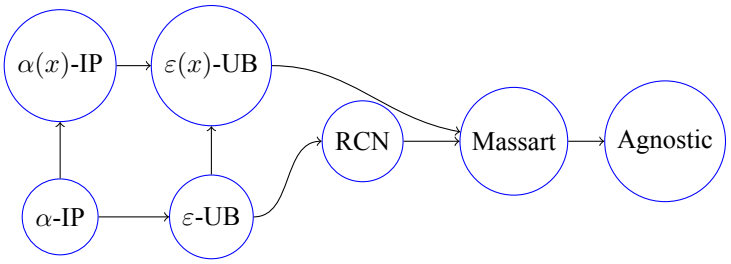

Figure 2.4: Problem Relationships

As in the reduction to learning with noise the distribution of the instances remains unchanged, we have that also agnostic-PAC learning algorithms will result in coarse learning algorithms (under $\varepsilon(x)$-UB). For the class of linear classifiers, we have that agnostic PAC learning can be achieved only in distribution-specific cases. Specifically, it has been proven that one can agnostically PAC learn half-spaces under the Gaussian distribution [Dia+22], [Kal+05], [Dan15a]. If any of these results are able to be extended in the multiclass classification case we would have PAC learning algorithms for learning with coarse labels as well.

CHAPTER 3

RANDOM CLASSIFICATION NOISE

In this chapter will study the problem of learning under RCN. Specifically, we will study how one can use the forward loss correction method to learn multiclass linear classifiers under RCN [Pat+17]. The method's main idea is to minimize the model mixed with the confusion matrix to match the noisy labels.

The main motivation of the study is [Kon+23] where it is shown that one can learn efficiently halfspaces under RCN (and Massart noise with knowledge of the flip probability at every point) with the use of this method.

First, we will discuss how learning with RCN differs from binary to multiclass classification and how this problem is usually solved. And then present an alternative solution for learning multiclass linear classifiers.

3.1 Binary vs Multiclass Label Noise

As we have seen learning under Random Classification Noise is the problem of learning with label noise that is independent of the instance described by a matrix, H. When Learning with noise one has to not take every example at face value and instead operate on statistics that reveal the true label over a region of the space.

Specifically learning under RCN in binary classification can be solved (for every function class that admits an SQ algorithm [BKW00]) as one can simulate any SQ using noisy data, [Kea98]. The most popular methods that are used to learn under RCN are backward loss correction methods. Specifically, if the confusion matrix, H, is invertible we can transform any statistic by the use of the inverse in order to get noiseless queries to another statistic. This can be used in an already SQ algorithm like perceptron or to transform a loss function in order to minimize a noiseless loss, [Blu+98], [Coh97],

[Nat+13]. Specifically, it is true that:

$$\begin{aligned}
\mathop{\mathbb{E}}_{(x,y)\sim\widetilde{D}}[g(x,y)] &= \mathop{\mathbb{E}}_{x\sim D_x}\left[\sum_{1}^{K} H_{c(x)j}g(x,j)\right] \\
&= \mathop{\mathbb{E}}_{x\sim D_x}\left[H_{c(x)}^T g(x)\right] \\
&= \mathop{\mathbb{E}}_{x\sim D_x}\left[\sum_{i=1}^{K} \mathbb{1}\left(c(x)=i\right) H_i^T g(x)\right] \\
&= \mathop{\mathbb{E}}_{x\sim D_x}\left[r^T Hg(x)\right],\ r : r_i = \mathbb{1}\left(c(x)=i\right)
\end{aligned}$$

Where $g(x)$ is the vector of $g(x,i), i \in [K]$ and H_i the i'th row of H. Then if one does choose g such that:

$$Hg(x) = G(x)$$

then $H_i^T g(x) = G(x,i)$. So

$$\begin{aligned}
\mathop{\mathbb{E}}_{(x,y)\sim\widetilde{D}}[g(x,y)] &= \mathop{\mathbb{E}}_{x\sim D_x}\left[H_{c(x)}^T g(x)\right] \\
&= \mathop{\mathbb{E}}_{x\sim D_x}[G(x,c(x))] \\
&= \mathop{\mathbb{E}}_{(x,y)\sim D}[G(x,y)]
\end{aligned}$$

Hence one can compute noiseless queries on G. And hence if there exists an SQ algorithm that solves the problem by the use of an appropriate g one can simulate the noiseless statistical queries. But that does rely on the solution of the system with the matrix H.

If H is not invertible solving the system algebraically (constructing the function g from G) is impossible for all points x. Also, the problem should not be solved in general as some variations are unidentifiable (for example if H assigns labels uniformly at random).

For binary classification, if the noise rate (probability that a label flips) is less than $1/2$ (we have a greater probability of seeing the true label) we have that the matrix H is invertible as it is diagonally dominant. Not only that one can subsample the noise rates so to make them the same without losing invertibility [RDM06]. But for multiclass classification that is not the case. Below we see an example of noise rates where we have a greater probability of seeing the true label but the matrix is non-invertible.

$$T = \begin{bmatrix} \frac{1}{2} & \frac{1}{4} & \frac{1}{4} & 0 \\ \frac{1}{4} & \frac{1}{2} & 0 & \frac{1}{4} \\ \frac{1}{4} & 0 & \frac{1}{2} & \frac{1}{4} \\ 0 & \frac{1}{4} & \frac{1}{4} & \frac{1}{2} \end{bmatrix}$$

Hence it remains an open question if every identifiable case is also solvable for multiclass classification.

3.2 Forward Loss Correction

In the following paragraphs, we will investigate the possible use of the Forward Loss Correction method [Pat+17]. This a method commonly used in practice that has been found to give efficient algorithms for learning noisy perceptrons [Kon+23].

We assume that the data are linearly separable with margin $\gamma > 0$. Or there is a linear classifier $c : \mathbb{R}^d \to [K]$ with weights $w^* \in \mathbb{R}^{d \times K} : c(x) = \text{argmax}_{j \in [K]} \left(w_j^{*T} x\right)$ and

$$w_{c(x)}^{*T} x \geq w_j^{*T} x + \gamma, \forall x \in \text{Sup}\,(D)\,.$$

Also without loss of generality we can assume that $\sum_1^K w_j^* = 0$, as adding to all parameter vectors the same constant vector does not change the prediction outcomes. Let $g(x)$ be the one-hot representation of $c(x)$. We use a multiclass linear classifier composite with a softmax layer, i.e.

$$f(x; w) = \frac{1}{N} \begin{bmatrix} z_1 \\ \dots \\ z_K \end{bmatrix}, \text{ where } z_i = e^{w_i^T x} i \in [K],\ N = \sum_i z_i$$

Let the noisy labels be generated according to the matrix H such that $H_{ij} = \Pr\left[\hat{y} = j \mid y = i\right]$ and H_i is the i'th column of H. We will study the outcome of minimizing the cross entropy of the mixed model with the transition matrix and the observed noisy labels. Let the mixed model and cross-entropy loss be defined as follows:

$$\text{mix}(f(x, w), H)_i = H_i^T f(x, w)$$

$$ce(y, h(x)) = -\sum_{i=1}^{K} y_i \ln h(x)_i$$

As we will see later this objective in our setting is non-convex, hence we can not use a plain objective value argument and we have to make an ad-hoc analysis with the use of its gradients about the trajectory of gradient descent.

3.3 Expected Gradient Computation

First, we will derive the gradient in this setting. In the next paragraphs, we will use the following expressions to develop first-order methods. By the chain rule, we have that:

$$\begin{aligned}
\partial_{w_{jl}} ce(y, \text{mix}(f(x, w), H)) &= -\sum_{i=1}^{K} \frac{y_i}{\text{mix}(f(x, w), H)_i} \partial_{w_{jl}} \text{mix}(f(x, w), H)_i \\
&= -\sum_{i=1}^{K} \frac{y_i}{\text{mix}(f(x, w), H)_i} \partial_{w_{jl}} H_i^T f(x, w) \\
&= -\sum_{i=1}^{K} \frac{y_i}{\text{mix}(f(x, w), H)_i} H_i^T \partial_{w_{jl}} f(x, w)
\end{aligned}$$

And

$$\begin{aligned}
\partial_{w_{jl}} f_i(x, w) &= \partial_{w_{jl}} \frac{z_i}{N} \\
&= -\frac{z_i \partial_{w_{jl}} N}{N^2} + \partial_{w_{jl}} z_i \frac{1}{N} \\
&= -\frac{z_i z_j x_l}{N^2} + \frac{\mathbb{1}\,(i = j)\, z_j x_l}{N} \\
&= -f_i f_j x_l + \mathbb{1}\,(i = j)\, f_j x_l \\
&= (\mathbb{1}\,(i = j) - f_i) f_j x_l
\end{aligned}$$

Hence

$$\begin{aligned}
\partial_{w_{jl}} ce(y, \text{mix}(f(x,w), H)) &= -\sum_{i=1}^{K} \frac{y_i}{H_i^T f} H_i^T (e_j - f) f_j x_l \\
&= \left(-\sum_{i=1}^{K} \frac{y_i}{H_i^T f} H_{ji} + \sum_{i=1}^{K} \frac{y_i}{H_i^T f} H_i^T f \right) f_j x_l \\
&= \left(-\sum_{i=1}^{K} \frac{y_i}{H_i^T f} H_{ji} + \sum_{i=1}^{K} \frac{y_i}{H_i^T f} H_i^T f \right) f_j x_l \\
&= \left(-\sum_{i=1}^{K} \frac{y_i}{H_i^T f} H_{ji} + 1 \right) f_j x_l, \ \sum_i y_i = 1 \qquad (3.1) \\
&= \left(\sum_{i=1}^{K} \frac{H_{ji} H_i^T f - H_{ji} y_i}{H_i^T f} \right) f_j x_l, \ \sum_i H_{ji} = 1 \\
&= \left(\sum_{i=1}^{K} (H_i^T f - y_i) \frac{H_{ji}}{H_i^T f} \right) f_j x_l \qquad (3.2)
\end{aligned}$$

Now observe that we have $\mathbb{E}[y_i \mid x] = H_i^T g = \text{mix}(g(x), T))$. Hence

$$\begin{aligned}
\partial_{w_j} \mathbb{E}[ce(y, \text{mix}(f(x,w), H)) \mid x] &= \mathbb{E}\left[\partial_{w_j} ce(y, \text{mix}(f(x,w), H)) \mid x\right] \\
&= \left(\sum_{i=1}^{K} (H_i^T f - H_i^T g) \frac{H_{ji}}{H_i^T f} \right) f_j x \\
&= \left(\sum_{i=1}^{K} \frac{H_{ji} H_i^T}{H_i^T f} (f - g) \right) f_j x \\
&= \left(\sum_{i=1}^{K} \frac{H_{ji} H_i}{H_i^T f} \right)^T (f - g) f_j x \\
&= a_j^T (f - g) f_j x, \text{ where } a_j = \sum_{i=1}^{K} \frac{H_{ji} H_i}{H_i^T f} \qquad (3.3)
\end{aligned}$$

And so the expected norm squared of the gradient is the following:

$$\begin{aligned}
\mathbb{E}\left[\|\nabla ce\|_2^2 \mid x\right] &\geq \|\mathbb{E}[\nabla ce \mid x]\|_2^2 \\
&= \nabla ce^T \nabla ce \\
&= \sum_{j \in [K]} a_j^T (f - g)(f - g)^T a_j f_j^2 x^T x
\end{aligned}$$

3.4 Loss Landscape

If g is defined as the one-hot vector of true labels then it is clear that f can only reach g in a limit case. While if the labels were realized by the soft labels of a linear model then f could be exactly equal to g. Hence to attain a small loss the norm of the predictor must diverge towards infinity [Sou+22].

Theorem 3.1. $\|\nabla ce\|_F \to 0$ as $\|w\|_F \to \infty$

Proof. As $\|w\|_F \to \infty$ then f becomes a one-hot vector with the only coordinate being one the $\text{argmax}_{j\in[K]} \left(\frac{w_j^T}{\|w\|_F} x \right)$. Hence for each x there exists only one non-zero coordinate of f, call it j. So observing the gradient expression 3.2 we find that only the j'th coordinate can be different than 0. But

$$\begin{aligned}
\partial_{w_{jl}} ce(y, \text{mix}(f(x,w), H)) &= \left(\sum_{i=1}^{K} (H_i^T f - y_i) \frac{H_{ji}}{H_i^T f} \right) f_j x_l \\
&= \left(\sum_{i=1}^{K} (H_{ji} - y_i) \frac{H_{ji}}{H_{ji}} \right) x_l \\
&= \left(\sum_{i=1}^{K} (H_{ji} - y_i) \right) x_l \\
&= 0
\end{aligned}$$

One can also note that in this case, the coefficient vector of the expected gradient equals the vector of all ones $a_j f_j = 1$, if $f_j = 1$ and otherwise it is zero. □

In the figure below we can see the convex landscape for the one-dimensional and binary case. And the above result is clearly evident.

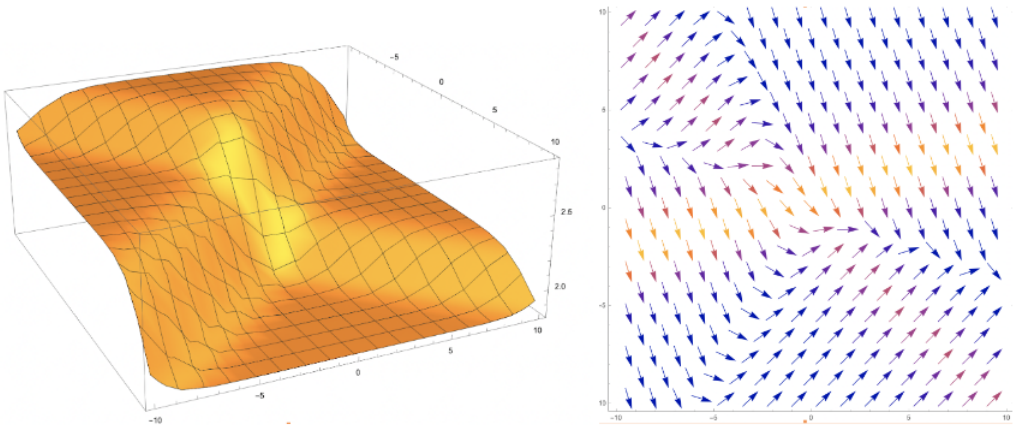

Figure 3.1: Loss landscape

So in order to argue that we converge to the true linear classifier. We will iteratively find a vector that the angle between it and w^* converges to 0. A popular example of this technique is the analysis of the perceptron.

3.5 Correlation with w^*

3.5.1 Correlation of Gradient Update with $w*$

Now we will analyze the gradient descent dynamics let $u_{tj} = -\eta_{tj}\partial_{w_j} ce(y, \text{mix}(f(x,w), H))$ be the SGD update for the parameters w_j when given the point (x, y) and u_t the overall update to the matrix w. Starting with the $w_0 = 0$ we will have that after T steps $w_{tj} = \sum_1^T u_{tj}$. As the convex landscape has no critical point we will have to show that our gradient rotates w in the correct direction. We have that in expectation updating with the point x at step t assuming that we have weights w:

$$
\begin{aligned}
\mathbb{E}\left[u_t \cdot w^* \mid x\right] &= \mathbb{E}\left[\sum_{j=1}^{K} u_{tj}^T w_j^* \mid x\right] \\
&= \sum_{j=1}^{K} \eta_{tj} a_j^T (g-f) f_j w_j^{*T} x \\
&= \left(\sum_{j=1}^{K} \eta_{tj} f_j (w_j^{*T} x) a_j^T\right)(g-f) \\
&= \left(\sum_{j=1}^{K} \eta_{tj} f_j (w_j^{*T} x) \sum_{i=1}^{K} \frac{H_{ji} H_i}{H_i^T f}\right)^T (g-f) \\
&= \left(\sum_{j=1}^{K}\sum_{i=1}^{K} \eta_{tj} f_j (w_j^{*T} x) \frac{H_{ji} H_i}{H_i^T f}\right)^T (g-f) \\
&= \left(\sum_{i=1}^{K} \frac{H_i}{H_i^T f} \sum_{j=1}^{K} \eta_{tj} f_j (w_j^{*T} x) H_{ji}\right)^T (g-f) \\
&= \left(\sum_{i=1}^{K} \frac{H_i}{H_i^T f} H_i^T r\right)^T (g-f),\ r \text{ col. vector s.t. } r_j = \eta_{tj} f_j (w_j^{*T} x) \\
&= r^T \left(\sum_{i=1}^{K} \frac{H_i H_i^T}{H_i^T f}\right)(g-f) \\
&= r^T \left(HDH^T\right)(g-f) \\
&= d^T \left(\Lambda H D H^T\right)(g-f)
\end{aligned}
$$

Where D a diagonal matrix with $\frac{1}{T_i^T f}$ on the diagonal, Λ the diagonal matrix of the step sizes multiplied with the soft labels $\Lambda = \text{diag}(\{\eta_i f_i\}_{i=1}^K)$, and d the vector of scores $w_i^* x$. From this update rule, it is not clear that the expected correlation between the update and the true vector is positive given the assumption that H leads to a statistically identifiable problem. Also, we do not have enough parameters to use the full information of H in the updates as Λ is diagonal.

3.5.2 Correlation of a Tuned Gradient Update

In this paragraph, we will examine if we could use a more complicated single-order method in order to solve our problem. Specifically, as we will see when we will examine the cases where H is invertible it would be beneficial if we were able to control more parameters and eventually use a step size matrix that involves the inverse of H. So we will examine the dynamics of the following gradient update:

$$w_{jt} = w_{j(t-1)} + A_j^T u = w_{j(t-1)} + \sum_{l=1}^{K} A_{jl} u_l$$

where u is the matrix of negative partial derivatives u_j one in each column and A is a matrix of weight coefficients, to be determined later.

$$\begin{aligned}
\mathbb{E}\left[Au \cdot w^* \mid x\right] &= \mathbb{E}\left[\sum_{j=1}^{K} w_j^{*T} A_j^T u \mid x\right] \\
&= \mathbb{E}\left[\sum_{j=1}^{K} w_j^{*T} \sum_{l=1}^{K} A_{jl} u_l \mid x\right] \\
&= \sum_{j=1}^{K}\sum_{l=1}^{K} w_j^{*T} A_{jl} a_l^T (g-f) f_l x \\
&= \left(\sum_{j=1}^{K}\sum_{l=1}^{K} w_j^{*T} x A_{jl} a_l^T f_l\right)(g-f) \\
&= \left(\sum_{l=1}^{K} a_l^T \sum_{j=1}^{K} d_j B_{jl}\right)(g-f),\ d_j = w_j^* x \text{ and } B = A \cdot \text{diag}(f_i) \\
&= \left(\sum_{l=1}^{K} a_l^T (B^T d)_l\right)(g-f) \\
&= \left(\sum_{l=1}^{K}\sum_{i=1}^{K} \frac{T_{ji} T_i^T}{T_i^T f} (B^T d)_l\right)(g-f) \\
&= \left(\sum_{i=1}^{K} \frac{T_i}{T_i^T f} T_i^T (B^T d)\right)^T (g-f) \\
&= d^T B \left(T D T^T\right)(g-f) \\
&= d^T \left(A \Lambda T D T^T\right)(g-f),\ \Lambda = \text{diag}(f_i)
\end{aligned} \tag{3.4}$$

Notice that we arrived at the same formula but now A is a full matrix instead of a diagonal.

3.5.3 Binary Classification Case

For the binary classification case the confusion matrix has the form:

$$T = \begin{bmatrix} a_1 & 1-a_1 \\ 1-a_2 & a_2 \end{bmatrix},\ \text{where } a_1, a_2 \in (1/2, 1]$$

And hence we have that

$$(T_1^T f T_2^T f)TDT^T = \begin{bmatrix} T_2^T f a_1^2 + T_1^T f (1-a_1)^2 & T_2^T f a_1 (1-a_2) + T_1^T f (1-a_1) a_2 \\ T_2^T f (1-a_2) a_1 + T_1^T f a_2 (1-a_1) & T_2^T f (1-a_2)^2 + T_1^T f a_2^2 \end{bmatrix}$$
$$= (T_2^T f (1-a_2) a_1 + T_1^T f a_2 (1-a_1)) \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$$
$$+ (a_1 + a_2 - 1)^2 \begin{bmatrix} f_2 & 0 \\ 0 & f_1 \end{bmatrix}$$

Thus by the assumption that all the weight vectors sum to 0, if we use the same step size for all dimensions, we have that we can ignore the first term. And so if we use as step size $\frac{T_1^T f T_2^T f}{f_1 f_2} \frac{1}{(a_1+a_2-1)^2}$ we have that:

$$\mathbb{E}\left[u_t \cdot w^* \mid x\right] = \sum_1^K (w_j^{*T} x)(g-f)_j$$

Now when f errs with respect to g at a point x by Lemma 3.2 we have a good correlation between the update and the optimal parameters.

3.5.4 Invertible Case

By 3.4 if we use $A = (T^T)^{-1} D^{-1} T^{-1} \Lambda^{-1}$ (T^T is invertible iff T and D, Λ are always invertible) then we would have that

$$\mathbb{E}\left[u_t \cdot w^* \mid x\right] = \sum_1^K (w_j^{*T} x)(g-f)_j$$

And so by 3.2 would have a good correlation with the optimal parameters.

Lemma 3.2. If f misclassifies x then we have that:

$$\sum_1^K (w_j^{*T} x)(g-f)_j \geq \frac{\gamma}{2}$$

Proof. Without loss of generality we have that $g_1(x) = 1$ and that $f_1(x) < \frac{1}{2}$. And so we have that:

$$\begin{aligned} \sum_1^K (w_j^{*T} x)(g-f)_j &= w_1^{*T} x(1-f_1) - \sum_{j\neq 1} w_j^{*T} x f_j \\ &= w_1^{*T} x(\sum_{j\neq 1} f_j) - \sum_{j\neq 1} w_j^{*T} x f_j \\ &= \sum_{j\neq 1} w_1^{*T} x f_j - \sum_{j\neq 1} w_j^{*T} x f_j \\ &= \sum_{j\neq 1} (w_j^{*T} x + \gamma) f_j - \sum_{j\neq 1} w_j^{*T} x f_j \\ &= \gamma \sum_{j\neq 1} f_j \\ &\geq \frac{\gamma}{2} \end{aligned}$$

□

Also, we have that $\mathbb{E}[u_t \cdot w^* \mid x] \geq 0$ as:

$$\begin{aligned}\sum_1^K (w_j^{*T} x)(g-f)_j &= w_y^{*T} x(1-f_y) - \sum_{j\neq y} w_j^{*T} x f_j \\ &= w_y^{*T} x(\sum_{j\neq y} f_j) - \sum_{j\neq y} w_j^{*T} x f_j \\ &= \sum_{j\neq y} (w_y^{*T} x - w_j^{*T} x) f_j \\ &\geq 0\end{aligned}$$

3.5.5 Non-Invertible Case

Here we will investigate what would be a valid solution concept for non-invertible matrices. If as in the intuition of the method, we could be able to efficiently generate an f such that its confusion under T is arbitrarily close to g. Then we notice that for the matrix

$$T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\ 0 & \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\ 0 & \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \end{bmatrix}$$

$\|Tf - Tg\|_2 \leq \varepsilon$ implies that $|f_1(x) - g_1(x)| \leq \varepsilon$ and hence for sufficiently small ε, $g_1(x) = 1$ if and only if $f_1(x) \geq \frac{1}{2}$. And if we predict the class 1 if $f_1(x) \geq \frac{1}{2}$ and otherwise 0 we could solve the problem of intersection of halfspaces efficiently.

3.6 Norm of the current guess

Here we will investigate how the norm of our parameter vector changes. Having a bound on the norm gives us a bound on the angle and hence the number of iterations till convergence. First of all, notice that:

$$\|w_T\|_F^2 = 2\sum_{t=1}^{T} v_t \cdot w_t + \sum_{i=1}^{T} \|v_t\|_F^2$$

Assuming that v_t is the update at step t. Hence we have find an upper bound on $\mathbb{E}[v_t \cdot w_t]$ and also balance the norm of the update vector. In general if $v_t = A^T \partial_w ce(y, \text{mix}(f(x,w), T))$ we have that:

$$\mathbb{E}[v_t \cdot w_t \mid x] = d^T (A\Lambda T D T^T)(g-f), \text{ from3.4}$$

where d vector in $\mathbb{R}^K$ such that $d_j = w_j^T x$. Now assuming that we are in the invertible case if A is set as $(T^T)^{-1} D^{-1} T^{-1} \Lambda^{-1}$ in order to have a high correlation with the true

parameters, then we have that:

$$\begin{aligned}
\mathbb{E}\left[v_t \cdot w_t \mid x\right] &= d^T(g-f) \\
&= d_y(1-\frac{e^{d_y}}{\sum e^{d_i}}) - \sum_{j\neq y} d_j \frac{e^{d_j}}{\sum e^{d_i}} \\
&= d_y(\sum_{j\neq y} f_j) - \sum_{j\neq y} d_j f_j \\
&= \sum_{j\neq y}(d_y - d_j) f_j \\
&= \sum_{j\neq y}(d_y - d_j)\frac{e^{d_j}}{\sum e^{d_i}} \\
&= \sum_{j\neq y}(d_y - d_j)\frac{1}{1+\sum_{i\neq j,y} e^{d_i-d_j} + e^{d_y-d_j}}
\end{aligned}$$

We will bound the above expression term by term if $d_y - d_j \leq 0$ then the term is non-positive. Furthermore if $d_y - d_j > 0$

$$\begin{aligned}
(d_y - d_j)\frac{1}{1+\sum_{i\neq j,y} e^{d_i-d_j} + e^{d_y-d_j}} &\leq (d_y - d_j)\frac{1}{e^{d_y-d_j}} \\
&\leq \frac{1}{e}
\end{aligned}$$

Hence the above expression is less than $(K-1)\frac{1}{e}$.

Now we have to bound $\|u_t\|_2$ we have that

$$\begin{aligned}
u_{tl} &= \sum_j A_{lj}\left(\sum_{i=1}^{K}(H_i^T f - y_i)\frac{H_{ji}}{H_i^T f}\right) f_j x \\
&= \sum_j A_{lj}\left(\sum_{i=1}^{K} H_i^T(f-q)\frac{H_{ji}}{H_i^T f}\right) f_j x \\
&= \sum_j A_{lj}\left(\sum_{i=1}^{K} H_i \frac{H_{ji}}{H_i^T f}\right)^T (f-q) f_j x \\
&= \sum_j A_{lj} a_j^T (f-q) f_j x
\end{aligned}$$

where q such that $H^T q = y$ as H is invertible such a vector exists. By using the analysis of paragraph 3.8.2 we have that:

$$\begin{aligned}
u_t^T u_t &= \sum_j u_{tj}^T u_{tj} \\
&= \sum_{j=1}^{K}\sum_{l=1}^{K} u_{tj}^T A_{jl} a_l^T (q-f) f_l x \\
&= d^T\left(A\Lambda T D T^T\right)(q-f)
\end{aligned}$$

where d is the vector of $u_{tj}^T x$. Thus by using the inverse matrix, we have that

$$\begin{aligned} u_t^T u_t &= d^T (q - f) \\ &= \sum_j (q - f)_j x^T u_{tj} \\ &= d'^T (q - f) \end{aligned}$$

Where d' is the vector of $(q - f)_j x^T x$ so if we assume that the point x are normalized in the ball of radius one, we have that

$$\begin{aligned} u_t^T u_t =& d'^T (q - f) \\ \leq& (q - f)^T (q - f) \\ =& (H^{-1^T} y - f)^T (H^{-1^T} y - f) \\ =& H^{-1^T} (y - H^T f)^T H^{-1^T} (y - H^T f) \\ =& \|(y - H^T f) H^{-1}\|_2^2 \\ \leq& (\max \lambda(H^{-1}))^2 \|(y - H^T f)\|_2^2 \\ \leq& O(K) \end{aligned}$$

We assume that the maximum eigenvalue from of the inverse is constant the last inequality holds from the fact that the matrix H is stochastic. Hence we have that:

$$\mathbb{E}\left[\|w_T\|_2^2\right] = O(KT)$$

3.7 Concentration Results

In this paragraph, we will show that since our update vector has a high correlation with the optimum without increasing the weight norm by much in expectation, we have that with high probability our guess will be steered to the optimum solution. For our concentration argument, we define the Martingale:

$$q_T = \sum_{t=1}^{T} (\mathbb{E}\left[u_t \mid F_{t-1}\right] - u_t)$$

Where let the filtration F_t be the randomness of the SGD update at step t and also let $q_0 = 0$.

From paragraph 3.5.4 and lemma 3.2 we have that if our current hypothesis has more than ε probability of error then

$$\begin{aligned} \mathbb{E}\left[u_t \mid F_{t-1}\right] \cdot w^* &= \mathop{\mathbb{E}}_{x \sim D_x} \left[u_t \cdot w^*\right] \\ &\geq \mathop{\mathbb{E}}_{x \sim D_x} \left[\frac{\gamma}{2} \mathbb{1}\left(\text{argmax} f_j(x) \neq g(x)\right)\right] \\ &= \frac{\gamma}{2} \Pr x \sim D_x \text{argmax} f_j(x) \neq g(x) \\ &\geq \frac{\varepsilon\gamma}{2} \end{aligned}$$

Hence

$$\sum_{1}^{T} \mathbb{E}\left[u_t \mid F_{t-1}\right] \cdot w^* \geq T\frac{\varepsilon\gamma}{2}$$

Furthermore, from the fact that $\|u_t\|_2 = O(\sqrt{K})$ we have that $\|\mathbb{E}\left[u_t \mid F_{t-1}\right] - u_t\|_2 = O(\sqrt{K})$ by triangle inequality. And hence by Cauchy-Schwarz and the assumption that $\|w^*\|_F = 1$ we have that $\|\mathbb{E}\left[u_t \cdot w^* \mid F_{t-1}\right] - u_t \cdot w^*\|_2 = O(\sqrt{K})$. Let c be a specific constant for bounding the above expectations.

From the above and the use of the Azuma-Hoeffding inequality, lemma 3.3, we have that

$$\begin{aligned}
&\Pr\left[q_T \cdot w^* \geq T\frac{\gamma\varepsilon}{4}\right] \leq e^{-\gamma^2\varepsilon^2 T/(c^2 128K)} \\
\Rightarrow &\Pr\left[\sum_{t=1}^{T}(\mathbb{E}\left[u_t \mid F_{t-1}\right] - u_t) \cdot w^* \geq T\frac{\gamma\varepsilon}{4}\right] \leq e^{-\gamma^2\varepsilon^2 T/(c^2 128K)} \\
\Rightarrow &\Pr\left[w_T \cdot w^* \leq \sum_{t=1}^{T}(\mathbb{E}\left[u_t \mid F_{t-1}\right] - u_t) \cdot w^* - T\frac{\gamma\varepsilon}{4}\right] \leq e^{-\gamma^2\varepsilon^2 T/(c^2 128K)} \\
\Rightarrow &\Pr\left[w_T \cdot w^* \leq T\frac{\gamma\varepsilon}{4}\right] \leq e^{-\gamma^2\varepsilon^2 T/(c^2 128K)}
\end{aligned}$$

Hence with T larger than $127c^2K\log(\frac{1}{\delta})\frac{1}{(\varepsilon\gamma)^2}$ we have that $w_T \cdot w^* \geq T\frac{\varepsilon\gamma}{4}$ with probability at least $1-\delta$. Also as $\mathbb{E}\left[\|w_T\|_F^2\right] = O(KT)$ by Markov's inequality we have that $\Pr\left[\|w_T\|_F^2 \geq O(\frac{1}{\delta}KT)\right] \leq \delta$.

Hence with probability 2δ:

$$\begin{aligned}
\cos(\theta(w^*, w_T)) &= \frac{w_T \cdot w^*}{\|w^*\|_2} \\
&\geq \frac{T\varepsilon\delta\gamma}{4\sqrt{KT}} \\
\Rightarrow T &= O(\frac{4K}{(\delta\varepsilon\gamma)^2})
\end{aligned}$$

So if $T = O(\frac{4K}{(\delta\varepsilon\gamma)^2})$ there exists an update that has error lower than ε with probability at least $1-\delta$.

So in the above paragraphs showed that learning halfspaces with information-preserving noise can be done efficiently by the use of the forward loss correction method. However, for that goal, we used a complicated update procedure that uses the inverse.

Lemma 3.3 (Azuma-Hoeffding)**.** Let $\xi^{(t)}$ be a martingale with bounded increments, i.e., $\left|\xi^{(t)} - \xi^{(t-1)}\right| \leq$ M. It holds that $\Pr\left[\xi^{(T)} \geq \xi^{(0)} + \lambda\right] \leq e^{-\lambda^2/\left(2M^2T\right)}$.

CHAPTER 4

LEARNING WITH COMPLEMENTARY LABELS

4.1 Reduction to Learning with Complementary Labels

The problem of learning with coarse or partial labels has a combinatorial structure given that the observed subsets can be arbitrary. In this paragraph, we will see a reduction that removes this combinatorial structure. And what results one can come up with, once the more straightforward problem is studied.

Specifically, one important variant of the problem of learning with coarse labels is one when all the sets have size $K-1$. Hence the algorithm receives pairs of an instance and a label counterexample that is not valid for that instance. In this variant, all the combinatorial structure of the sets is gone and we have kind of an inverse problem to learning with correct labels. This problem is called learning with complementary labels and has been greatly studied in the literature ([Ish+17], [Yu+18], [Zha+21]). However, despite the fact that this problem seems much simpler than the problem of learning with coarse labels, the two problems are equivalent.

Theorem 4.1. Distribution-independent learning with coarse samples with the assumption that $\Pr[|S| < K \mid x] \geq \beta$ can be reduced to distribution-independent learning with complementary labels.

Proof. Assuming that there is an efficient PAC learning algorithm for distribution-independent learning with complementary labels. And that its sample complexity is characterized by the polynomial p, on ϵ and δ, let $m = p(1/\epsilon', 2/\delta)$.

Then let $\Pr[|S| < K] = \alpha$ by Lemma 4.2 it suffices to draw $O(m/\alpha)$ (polynomial number) samples from D to have m informative examples with probability greater than $\delta/2$. Then let $\{(x_i, S_i)\}_1^m$ such samples.

We transform them to $\{(x_i, S_i')\}_1^m$, $S_i' = [K] - z$ where z is uniformly drawn from $\bar{S}_i$. Now as there exists an efficient PAC learning algorithm for this setting and m examples suffice, we can just run it and get an output hypothesis, h.

However, the distribution on X in the second problem has changed from D_x to D_x'. Points that are more probable to generate uninformative samples are being sampled less

often. As a result

$$D'_x = \frac{D_x \cdot \Pr\left[|S| < K \mid x\right]}{\Pr\left[|S| < K\right]}$$

Because of the assumption that $\Pr\left[|S| < K \mid x\right] \geq \beta$ then we have that

$$\begin{aligned}
\mathop{\mathbb{E}}_{x\sim D_x}\left[\mathbb{1}\left(h(x) \neq c(x)\right)\right] &= \int_X \mathbb{1}\left(h(x) \neq c(x)\right) D_x dx \\
&= \int_X \mathbb{1}\left(h(x) \neq c(x)\right) D_x \frac{\alpha \Pr\left[|S| < K \mid x\right]}{\alpha \Pr\left[|S| < K \mid x\right]} dx \\
&= \alpha \int_X \mathbb{1}\left(h(x) \neq c(x)\right) D'_x \frac{1}{\Pr\left[|S| < K \mid x\right]} dx \\
&\leq \frac{\alpha}{\beta} \mathop{\mathbb{E}}_{x\sim D_x}\left[\mathbb{1}\left(h(x) \neq c(x)\right)\right] \\
&\leq \frac{\alpha}{\beta}\epsilon'
\end{aligned}$$

If we set $\epsilon' = \epsilon\frac{\beta}{\alpha}$ we get an ϵ, δ hypothesis for coarse label learning. □

Lemma 4.2. Let D a distribution over domain X and a property $G \subseteq X$ with $\Pr\limits_{x\sim D}\left[x \in G\right] \geq \alpha$ then with $O(\frac{n}{\alpha}\log\frac{1}{\delta})$ samples from the distribution D we got that at least n samples will statisfy the property G with probability $1-\delta$.

Proof. We define the random variables $X_1, \ldots, X_m$ with $X_i = \mathbb{1}\left(x \in G\right)$. The X_i's are iid Bernoulli random variables with $\mathop{\mathbb{E}}\limits_{X_1\sim D}\left[X_1\right] \geq \alpha$. Let $\mathop{\mathbb{E}}\limits_{X_i\sim D}\left[\sum X_i\right] = \mu$ from the Chernoff Bound we have that:

$$\begin{aligned}
&\Pr\left[\sum X_i \leq (1-\varepsilon)\mu\right] < e^{-\frac{\varepsilon^2}{2}\mu} \\
\Rightarrow &\Pr\left[\sum X_i \leq (1-\varepsilon)m\alpha\right] < e^{-\frac{\varepsilon^2}{2}m\alpha} \\
\Rightarrow &\Pr\left[\sum X_i \leq (1-\varepsilon)8n\ln\frac{1}{\delta}\right] < e^{-\varepsilon^2 4n\ln\frac{1}{\delta}},\ m = \frac{8n}{\alpha}\ln\frac{1}{\delta} \\
\Rightarrow &\Pr\left[\sum X_i \leq n\right] < e^{-n\ln\frac{1}{\delta}},\ \delta < e^{-1/4}, \varepsilon = 1/2 \\
\Rightarrow &\Pr\left[\sum X_i \leq n\right] < e^{-\ln\frac{1}{\delta}},\ n \geq 1 \\
\Rightarrow &\Pr\left[\sum X_i < n\right] < \delta
\end{aligned}$$

□

Theorem 4.3. Distribution-independent learning with coarse samples with the assumption that $\Pr\left[z \in S \mid x\right] \leq \varepsilon$ can be reduced to distribution-independent learning with complementary labels $(x, \bar{y}), \bar{y} \neq y$ with $\Pr\left[z = \bar{y}\right] \geq (1-\varepsilon)\frac{1}{K-1}, \forall z \neq y$.

Proof. Obviously as $\Pr[|S| < K \mid x] \geq \Pr[z \notin S \mid z \neq y] \geq 1 - \varepsilon$. So by the above theorem, we have that the problem can be reduced to learning with complementary labels. But also the resulting counterexamples satisfy the assumption that each non-label has some lower bounded probability to be observed for every x.

$$\begin{aligned}
\Pr[z = \bar{y} \mid x] &= \Pr[z \notin S \wedge z = \bar{y} \mid x] \\
&= \Pr[z \notin S \mid x] \Pr[z = \bar{y} \mid x \wedge (z \notin S)] \\
&= \Pr[z \notin S \mid x] \, \mathbb{E}\left[\frac{1}{|\bar{S}|} \mid x \wedge (z \notin S)\right] \\
&\geq (1 - \varepsilon)\frac{1}{K-1} > 0
\end{aligned}$$

And so $\Pr[z \neq \bar{y} \mid x] \leq 1 - (1 - \varepsilon)\frac{1}{K-1} < 1$. And so this reduction does not lose the set generation assumption $\varepsilon(x)$-UB. □

Although the algorithm is not practical as it increases the ambiguity of the label for each example. It is important from a theoretical point of view as it eliminates all the combinatorial structures of the sets. Also in this setting, we have reduced the number of possible observed sets to K from 2^K hence backwards loss correction methods can be applied more efficiently with respect to the dependence on K. However, learning with complementary labels without additional assumptions is computationally hard.

Theorem 4.4. Proper PAC learning multiclass linear classifiers with complementary labels in the realizable setting is hard unless $NP = RP$.

Proof. It suffices to show a reduction from learning the intersection of two halfspaces to this problem. As learning the intersection of two halfspaces is hard unless $NP = RP$ [KS08],[KS06]. Assuming that there is a PAC learning algorithm for learning multiclass linear classifiers with complementary labels and $NP \neq RP$. Let p be the polynomial that characterizes the sample complexity.

Given $m = p(\frac{1}{\epsilon}, \frac{1}{\delta})$ samples from an unknown distribution D that is realized by the intersection of two unknown halfspaces, $w_1, w_2 \in \mathbb{R}^d$, $\{x_i, c(x_i)\}_1^m$. We can transform it into a sample of counterexamples from $x_i \sim D_x$ that is realized by a multiclass linear classifier. If $c(x_i) = +1$ we uniformly pick for $\bar{y_i}$ labels from $\{2, 3, 4\}$ and if $c(x_i) = -1$ we pick as $\bar{y_i} = 1$.

From this transformation me get $\{x_i, \bar{y_i}\}_1^m$. It is easy to check that the multiclass linear classifier that is defined by the vectors $w_1 + w_2, w_1 - w_2, w_2 - w_1, -(w_1 + w_2)$ realizes the above counterexamples. And if we learned this classifier to a high accuracy with high probability then by adding combinations of the vectors we could get a probably highly accurate hypothesis for the intersection of halfspaces. □

4.2 Strong Linear Separability

Here we will look at a stronger version of linear separability that will enable us to use binary classification algorithms. This stronger notion of separability has been used recently to make efficient bandit linear classifiers with a finite mistake bound [Bey+19].

Definition 4.5. We say that a distribution D on $\mathbb{R}^d \times [K]$ is strongly linearly separable if

$$w_y^T x > 0 \text{ and } w_j^T x < 0, \forall j \neq y, \forall (x,y) \in \text{Sup}(D)$$

In that case, the One versus All reduction to binary classification works.

Theorem 4.6. If D is strongly linearly separable then training we can solve the problem efficiently by training K binary linear classifiers.

Proof. As one can learn a linear efficiently with linear programming when having a polynomial number of samples. Let $m = \text{poly}\left(\frac{1}{\epsilon}, \frac{1}{\delta}\right)$ the required number of samples to get accuracy ϵ/K and confidence δ/K.

Let $S \sim D^m$ we can make $S_1, \ldots, S_K$ samples sets each one S_i by labeling the class i positive and all the others negative. As each set is linearly separable can use those to train K binary linear classifiers $w_1, \ldots, w_K$ with accuracy ϵ/K and confidence δ/K. Now for a new x if we predict a random label that has a positive inner product and we call our classifier h. We have that with probability $1-\delta$ all $\{w_i\}_1^K$ have error ϵ/K and:

$$\begin{aligned}
\Pr_{(x,y)\sim D}[h(x) \neq y] &\leq \Pr_{(x,y)\sim D}\left[w_y^T x > 0 \text{ or } \vee_{j\neq y}\, w_j^T x < 0\right] \\
&\leq \sum_1^K \Pr_{(x,y)\sim D}[w_i \text{ errs on } x] \\
&\leq \epsilon
\end{aligned}$$

□

It is true that multiclass linear classification can be reduced to binary classification by the use of a reduction to the strongly linearly separable case. This reduction is done with the use of the rational kernel [SSSS10].

Theorem 4.7 (Theorem 5 [Bey+19])**.** Let $(x_1, y_1), (x_2, y_2), \ldots, (x_T, y_T) \in \text{B}(\mathbf{0}, 1) \times \{1, 2, \ldots, K\}$ be a sequence of labeled examples that is weakly linearly separable with margin $\gamma > 0$. Let,

$$\gamma_1 = \frac{\left[376 \lceil \log_2(2K-2) \rceil \cdot \left\lceil \sqrt{\frac{2}{\gamma}} \right\rceil\right]^{\frac{-\lceil \log_2(2K-2)\rceil \cdot \sqrt{2/\gamma}\rceil}{2}}}{2\sqrt{K}},$$

$$\gamma_2 = \frac{\left(2^{s+1} r(K-1)(4s+2)\right)^{-(s+1/2)r(K-1)}}{4\sqrt{K}(4K-5)2^{K-1}},$$

where $r = 2\left\lceil \frac{1}{4}\log_2(4K-3)\right\rceil + 1$ and $s = \lceil \log_2(2/\gamma) \rceil$. There exists transformation ϕ such that the sequence of labeled examples transformed by ϕ, namely $(\phi(x_1), y_1), \ldots, (\phi(x_T), y_T)$, is strongly linearly separable with margin $\gamma' = \max\{\gamma_1, \gamma_2\}$.

This reduction is efficient however it deceases the margin. We will use this theorem to reduce our problem to important binary classification problems.

4.3 Positive and Unlabeled Learning

As we saw it suffices to investigate the learning with complementary labels problem. Now we will assume that we are also in the strongly linearly separable setting we described in the previous paragraph. Hence it suffices to train K linear classifiers one for

each class. However in this case the binary classifier that would classify as positive the i'th class and as negative all the rest, has to be trained with examples that certify that a point does not belong in the i'th class and points that correspond to different classes and their label is there for irrelevant for the classifier.

From the point of view of the i'th binary classifier the sample set is a set of negative, (when the non-label is i) and unlabeled examples (when the non-label is different than i). Hence it suffices to solve the problem of learning with positive (if we rename the labels) and unlabeled examples. With the additional assumption that we receive iid samples $(x, y), x \sim D_x$ such that when $y = -1$ we observe $\hat{y} = 0$ (ie unlabeled), and when $y = 1$ there is the lower bounded probability that we will observe the true label, otherwise we observe $\hat{y} = 0$.

The problem of learning with positive and unlabeled data (PU learning) is greatly studied in practice [HL20], [PNS14]. But in theory, it is only investigated when the distribution of positive examples is the conditional distribution, $D(x \mid y = 1)$, and the distribution of unlabeled examples is unchanged D_x, [DGL05]. And in this case, this problem can be reduced to learning with noise if one considers examples drawn with $2/3$ probability from the distribution of the positive examples and labeled positive, and with probability $1/3$ from D_x and labeled negative. And that can be solved for all SQ classes as we are talking about binary label noise that is bounded away from $1/2$ for a distribution close to D (for more details [DGL05] Proposition 1).

However, as in our case, we have an instance-dependent probability of a label being revealed, and hence we have a change in the positive example distribution. Particularly we observe positive examples from $D'_+ = \eta(x)D(x)/\mathop{\mathbb{E}}_D[\eta(x)]$ and unlabeled examples from D. And we will prove that the PU learning to noise reduction mentioned above leads to an instance-dependent Massart noise or Constant Partition Classification Noise [Dec97] with unknown partition. As a result, due to recent breakthroughs [DGT19], [Che+20], this gives an approximate algorithm for learning with coarse labels. However, this reduction maybe has lost information about the problem so we propose PU learning with a lower bounded probability of observing a positive example as an open problem.

Theorem 4.8. Let D be a distribution realized by a halfspace. The problem of learning from $x \sim D$ when the positive label is presented with probability $\eta(x)$ can be reduced to learning with Massart noise under a different distribution.

Proof. We construct the distribution D' by sampling a positive example from $D'(x)$ with probability $2/3$ and labeling it positive or sampling an unlabeled example from D with probability $1/3$ and labeling it negative. Then the resulting distribution is:

$$D'(x) = \begin{cases} \frac{D(x)}{3}, & c(x) = -1 \\ \frac{D(x)+2D_+(x)}{3}, & c(x) = +1 \end{cases}$$

And the noise rate is:

$$n(x) = \begin{cases} 0, & c(x) = -1 \\ \frac{D(x)}{2+D_+(x)}, & c(x) = +1 \end{cases}$$

Hence we have that the noise rate is at most $\frac{1}{2+\eta/\mathop{\mathbb{E}}_D[\eta(x)]}$. □

The distribution of the noisy examples is different than the original however it is close as it has non-trivial mass in every point that the original has.

CHAPTER 5

LEARNING WITH SIMPLE INSTANCE DEPENDENCIES

5.1 Unbiased Coarse Labels

In this paragraph, we will solve the problem of learning with coarse labels for a rather simple but diverse and widely researched setting [RW18], [Ish+17]. Specifically, we will study the setting where each label, other than the ground truth, is present in the set with the same probability. Or in other words consider the setting, $\eta(x)$-EP, (for $\eta(x)$-Equal Probabilities) where

$$\Pr_{(x,S)\sim D_\pi}[z \in S \mid x] = \eta(x) \leq \eta < 1, \forall z \neq c(x), \forall x \in \mathcal{X}$$

Also, assume that the ground truth labels follow a model $c(x)$. This is an instance-dependent setting and hence it is a specialization of the $\varepsilon(x)$-UB problem. Notice that the labels may be observed with the same probability but not independently, as for example two labels can be present half the time but not necessarily together. The difference from the general $\varepsilon(x)$-UB problem is the fact that, in this case, there is only one function that describes the problem, in the general setting we would have K.

For the purposes of the next paragraphs without loss of generality, we will ignore the fact that the distribution can give uninformative samples (i.e. $S = [K]$) because these cases can be easily adjusted for by the argument in Theorem 4.1 in Chapter 4.

Here we will prove that this setting can be reduced to the case where the probability is instance-independent.

Theorem 5.1. Given $\{(x_i, S_i)\}_1^m$ from the $\eta(x)$-EP setting there exists an constant α such that we can transform them to samples $\{(x_i, S_i')\}_1^m$ from the α-EP setting.

Proof. Define as π_x the distribution of the observed set $S_i \subseteq [K]$ when the instance is $x \in \mathcal{X}$. If for each sample set S_i each is replaced by the set $S_i' = [K] - z_i$, where z_i is drawn uniformly at random from $\bar{S}_i$. Let π_x' be the probability distribution of S_i' when

the instance is x. For the new set generation procedure we have that:

$$\begin{aligned}\Pr_{S_i' \sim \pi_x'}[z \notin S_i'] &= \Pr_{S_i' \sim \pi_x'}[z = z_i] \\ &= \mathop{\mathbb{E}}_{S_i \sim \pi_x}\left[\Pr_{S_i' \sim \pi_x'}[z = z_i \mid S_i]\right] \\ &= \mathop{\mathbb{E}}_{S_i \sim \pi_x}\left[\frac{\mathbb{1}(z_i \notin S_i)}{K - |S_i|}\right]\end{aligned}$$

Hence if we take the complimentary labels we have that the resulting probability distribution depends on the expected size of the complement of S. And thus could be instance dependent and different for different labels. However, we can apply rejection sampling with rate $\frac{K-|S_i|}{K}$ when preprocessing the samples. We will have a new marginal over the sets π_x'' and we have that:

$$\pi_x''(S) = \frac{\pi(x)(K - |S|)}{Z}$$

Where Z is a normalization factor so:

$$\begin{aligned}\mathop{\mathbb{E}}_{S \sim \pi_x''}\left[\frac{\mathbb{1}(z_i \notin S)}{K - |S|}\right] &= \sum_S \frac{\mathbb{1}(z_i \notin S)}{K - |S|}\pi_x''(S) \\ &= \sum_S \frac{\mathbb{1}(z_i \notin S)}{K - |S|}\frac{\pi(x)(K - |S|)}{Z} \\ &= \sum_S \frac{\mathbb{1}(z_i \notin S)}{Z}\end{aligned}$$

Which is constant for all labels z_i other than the ground truth. So $\Pr_{S_i' \sim \pi_x'}[z \notin S_i'] = \frac{1}{K-1}, \forall z \neq c(x)$ independent of the instance x. The above rejection sampling procedure chances the distribution D_x but as there is a lower bounded probability of accepting any point we have that the distribution is similar enough such that learning in this case can lead to learning in the original problem (as in the proof of Theorem 4.1). □

So even though the probability that each label is present depends on the instance from an instance-dependent transformation this can be reduced to a much simpler case. The resulting set distribution is equivalent to the case where for each instance we are presented with a uniformly chosen non-label and we can show that it is information preserving.

Theorem 5.2. The set generation process which we are presented with a uniformly chosen complementary label is information preserving.

Proof. Let $P \in \mathbb{R}^{2^k \times k}$ be the coarsion confusion matrix i.e. $P_{S,j} = \Pr_{S \sim \pi_j}[S]$. In our case, we have that P has only K non-zero columns (the sets of size $K - 1$). And for each corresponding set, the probability that it is chosen is $\frac{1}{K-1}$.

Let p, q probability distributions on the labels. From the structure of P we have that

$$\|P \cdot (p - q)\|_1 = \|(J_K - I_K)(p - q)\|_1$$

When J_K is all one's matrix of size $K \times K$. The matrix $(J_K - I_K)$ is known as the complement of the identity and it is always invertible. Hence we have that by simple norm-relationships

$$\begin{aligned}\|P \cdot (p-q)\|_1 &= \|(J_K - I_K)(p-q)\|_1 \\ &\geq \|(J_K - I_K)(p-q)\|_2 \\ &\geq \min_{\lambda \in Sp(J_K - I_K)} |\lambda| \|p-q\|_2 \\ &\geq \left(\frac{1}{\sqrt{K}} \min_{\lambda \in Sp(J_K - I_K)} |\lambda|\right) \|p-q\|_1\end{aligned}$$

□

So by applying the reduction to the instance-independent case and then the MLE algorithm, we have that:

Theorem 5.3. If the set generation process is such that every label i other than the ground truth appears in the set with probability $\eta(x)$ independently, for all points then there is an efficient coarse learning algorithm that uses samples $O(N\text{poly}(K))$. If there exists an efficient SQ learning algorithm that N statistical queries.

5.2 Labels presented IID

Now we will prove that if each label is present iid with probability less than η to S then the associated partition distribution is information preserving. In other words, we will investigate the setting where there exists one set generation distribution for each label π_y, and the sets are generated as such:

$$\Pr_{S \sim \pi_y}[z \in S] = \eta_z \leq \eta, \forall y, \forall z \neq y \text{ iid}$$

Theorem 5.4. The partition probability distribution in such a setting is information preserving.

Proof. Let p, q probability distributions on the labels. And let $P \in \mathbb{R}^{2^k \times k}$ be the coarsion confusion matrix i.e. $P_{S,j} = \Pr_{S \sim \pi_j}[S]$. Then it is true that:

$$(P \cdot p)_S = \Pr_{S \sim p_\pi}[S]$$

We want to lower bound we $\|P \cdot (p-q)\|_1$ with respect to $\|p-q\|_1$. As the l_1 and l_∞ norms are dual of each other we have that for any $v : \|v\|_\infty \leq 1$

$$\|P \cdot (p-q)\|_1 \geq v^T P \cdot (p-q)$$

Let $G = \{i : p_i - q_i \geq 0\}$. And set v such that $v_S = (|S \cap G| - |S \cap \bar{G}|)/K$. Let $g = Kv$ the not normalized vector. Then

$$g^T P p = \mathop{\mathbb{E}}_{S \sim p_\pi}[|S \cap G|] - \mathop{\mathbb{E}}_{S \sim p_\pi}\left[|S \cap \bar{G}|\right]$$

We can write $|S \cap G| = \sum_{i \in G} X_i$, for $X_i = \mathbb{1}(i \in S)$ and thus

$$\begin{aligned}\mathop{\mathbb{E}}_{S \sim p_\pi}[|S \cap G|] &= \sum_{i \in G} \Pr_{S \sim p_\pi}[i \in S] \\ &= \sum_{i \in G} (p_i + (1 - p_i)\eta_i) \\ &= \sum_{i \in G} (\eta_i + (1 - \eta_i)p_i)\end{aligned}$$

So we have that:

$$\begin{aligned}K\|P \cdot (p - q)\|_1 \geq &\sum_{i \in G} (\eta_i + (1 - \eta_i)p_i) - \sum_{i \in \bar{G}} (\eta_i + (1 - \eta_i)p_i) \\ &- \sum_{i \in G} (\eta_i + (1 - \eta_i)q_i) + \sum_{i \in \bar{G}} (\eta_i + (1 - \eta_i)p_i) \\ &= \sum_{i \in G} (1 - \eta_i)(p_i - q_i) + \sum_{i \in \bar{G}} (1 - \eta_i)(q_i - p_i) \\ &\geq (1 - \eta)\|p - q\|_1\end{aligned}$$

Thus $TV(Pp, Pq) \geq \frac{1-\eta}{K} TV(p, q)$ and so the partition distribution is information preserving. □

The intuition is that we set v as a tester to distinguish between the distributions Pp and Pq. And v has the function that counts the good elements, that are more probable in one distribution over the other, (versus the bad) in the observed set. By definition of the good and bad elements in terms of p and q the resulting test gives us an appropriate separation as observing good elements when the set is generated from Pp is more probable than observing when we have Pq. That difference with few calculations relates to the l_1 norm of p and q. For information-preserving coarsions we can apply the maximum likelihood algorithm and so we have the following theorems.

Theorem 5.5. If the set generation process is such that every label i other than the ground truth appears in the set with probability η_i independently, for all points then there is an efficient coarse learning algorithm that uses samples $O(N\text{poly}(K/(1 - \eta)))$. If there exists an efficient SQ learning algorithm that N statistical queries.

5.3 Learning with Hierarchically Structured Sets

As we have seen in chapters 4 and 5 one can simply ignore the combinatorial structure of the subsets increase the level of coarsening and only work with complementary labels. However, that may be sub-optimal, and well-structured sets could give us a more efficient training procedure. We study the setting when there is a Hierarchical partition of the labels and we with some probability observe the associated set (the one containing the ground true label) at some level of the hierarchy.

Specifically, if we observe data from the following generative model:

1. $y \sim D$
2. $P \sim \pi$

3. Observe S where $S \in P$ such that $y \in S$.

Where π is a distribution on the levels of a hierarchical partition scheme of the labels. For simplicity, we assume that the hierarchical partition scheme forms a complete binary tree. And with each sample, we observe with probability q_i the set on the i'th (with fine-grained samples being the 0'th level). Hence by simply forming the histogram of the fine-grained samples and ignoring the sets we could learn the distribution in $\theta(\frac{n}{q_0\epsilon^2})$ samples. Can we find better sample complexity than when using only the produced fine-grained labels? The following theorem answers this question negatively.

Theorem 5.6 (Hierarchical Distribution Learning)**.** We need $\Omega(\frac{n}{q_0\epsilon^2})$ hierarchically coarse samples to approximate a discrete distribution on n elements.

Proof. We can apply the same counterexample that proves that the sample complexity of learning a discrete distribution on n elements is $\Omega(\frac{n}{\epsilon^2})$. Let the hierarchical partitioning scheme form a binary tree and n even.

Now let z a vector and p_z a probability distribution such that:

$$p_z(i) = \frac{1-\epsilon z_i}{n} \text{ and } p_z(i+1) = \frac{1+\epsilon z_i}{n}$$

Assuming that we output a $p_{\hat{z}}$ distribution in this family.

Notice that each $\hat{z}_i$ that does not agree with the corresponding z_i contributes $\frac{2\epsilon}{n}$ to the total variation distance. Hence

$$\begin{aligned} d_{\text{TV}}(p_z, p_{\hat{z}}) &= \frac{2\epsilon}{n}\sum_{1}^{\frac{n}{2}}(\mathbb{1}(\hat{z}_i(S) = z_i))) \\ \Rightarrow \mathbb{E}[d_{\text{TV}}(p_z, p_{\hat{z}})] &= \frac{2\epsilon}{n}\sum_{1}^{\frac{n}{2}}\mathbb{E}[\mathbb{1}(\hat{z}_i(S) = z_i)] \\ &= \frac{2\epsilon}{n}\sum_{1}^{\frac{n}{2}}\mathbb{E}[\mathbb{E}[\mathbb{1}(\hat{z}_i(S) = z_i) \mid B_1 \dots B_{|S|}]] \end{aligned}$$

Where B_i is an indicator variable that signifies to which of the $\frac{n}{2}$ bins sample i belongs before the the coarsing process. The conditional distribution of S_j on B_j is the level distribution with level 0 being split into two events. Specifically the distribution of S_j conditional on B_j has domain $\Omega = \{2B_j, 2B_j+1, S_1, S_2, \dots S_{\lg n}\}$ with probabilities $\{q_0 \cdot \left(\frac{1-z_j\epsilon}{2}\right), q_0 \cdot \left(\frac{1+z_j\epsilon}{2}\right), q_1, \dots, q_{\lg n}\}$ where q_i is the probability that we draw level i from π.

So $\hat{z}_i$ has to distinguish between two distributions P_0 and P_1 on Ω, $P_0 = \{q_0 \cdot \left(\frac{1-\epsilon}{2}\right), q_0 \cdot \left(\frac{1+\epsilon}{2}\right), q_1, \dots, q_{\lg n}\}$ and $P_1 = \{q_0 \cdot \left(\frac{1+\epsilon}{2}\right), q_0 \cdot \left(\frac{1-\epsilon}{2}\right), q_1, \dots, q_{\lg n}\}$. From [DK19]

$$\mathbb{E}[\mathbb{1}(\hat{z}_i(S) = z_i) \mid B_1 \dots B_{|S|}] \geq \frac{1}{2} - \frac{d_{TV}(P_0^{k_i}, P_1^{k_i})}{2}$$

So from Pinsker's Inequality

$$d_{TV}(P_0^{k_i}, P_1^{k_i}) \leq \sqrt{\frac{D_{\text{KL}}(P_0^{k_i} \mid P_1^{k_i})}{2}} = \sqrt{k_i\frac{D_{\text{KL}}(P_0 \mid P_1)}{2}} = \sqrt{k_i q_0 \frac{\epsilon^2}{1-\epsilon}} \leq \sqrt{2q_0k_i}\epsilon, \text{for } e \leq \frac{1}{2}$$

Because only the terms of the level 0 events contribute to the KL divergence. And $\mathbb{E}[k_i] = \frac{2k}{n}$, k_i is the number of samples that belong in the bin i before the coarsing process. So

$$\begin{aligned}
\mathbb{E}[\mathbb{E}[\mathbb{1}(\hat{z}_i(S) = z_i) \mid B_1 \dots B_{|S|}]] &\geq \frac{1}{2} - \mathbb{E}[\sqrt{k_i}]\sqrt{2q_0}\epsilon \\
&\geq \frac{1}{2} - \sqrt{\frac{2k}{n}}\sqrt{2q_0}\epsilon, \text{ Jensen} \\
&= \frac{1}{2} - 2\sqrt{kq_0/n}\epsilon \\
\Rightarrow \mathbb{E}[d_{\mathrm{TV}}(p_z, p_{\hat{z}})] &\geq \epsilon(1 - 4\epsilon\sqrt{kq_0/n})
\end{aligned}$$

So for $\mathbb{E}[d_{\mathrm{TV}}(p_z, p_{\hat{z}})] < \epsilon/2$ we need a number of samples $k \geq \frac{1}{64}\frac{n}{\epsilon^2 q_0}$ □

Hence in order to learn the distribution to total variation distance ε we need to observe the last layer of the tree a sufficient number of times.

CHAPTER 6

LINEAR MULTICLASS CLASSIFIERS WITH AGNOSTIC NOISE

Learning linear classifiers with no distributional assumptions is computationally hard in the agnostic case [Dan15b]. However given the fact that the instances come from well-behaved distributions like the Uniform, Normal, or Log-concave efficient algorithms can be designed for agnostically learning with many geometric concepts (like convex sets, intersections of halfspaces and halfspaces) [Kal+05], [Dan15a], [KOS08].

These algorithms rely on the fact that under these distributions there exist polynomials of low-degree that approximate every function in the function class. And as polynomial regression can be formulated with a convex program we can solve it and learn improperly by predicting with the regression polynomial. In this chapter, we will develop an algorithm that learns under the Normal Distribution by using this technique. The algorithm that we will design will not be polynomial (as this is not achievable even in the binary classification case) but it will run in polynomial time for any fixed error input ε.

Specifically, we will use the framework of [Kal+05] and reduce the problem to binary classification. However common reductions from multiclass to binary classification like One Versus All and All Pairs will fail to reach the optimum accuracy [DB95], [ASS01], [DSS12]. For that reason, we will develop a novel objective function such that the resulting polynomials after minimization are easier to round and therefore predict. Finally, we will prove that our new framework will work for a variety of problems like agnostic learning with coarse labels.

6.1 Approximating Polynomials for Related Concepts

First, consider the class regions of a linear multiclass learning model. As we predict j whenever $w_j^{*T}x - t_j^* > w_i^{*T}x - t_i^*, \forall i \neq j$ we have that the region of the class j is the intersections of the halfspaces with normal vectors $w_j^* - w_i^*, \forall i \neq j$ and offsets $t_j^* - t_i^*$. Hence if we could learn this intersection of $K-1$ halfspaces we could apply a one versus-all prediction classifier. But, as we have mentioned in previous chapters learning intersections of halfspaces is computationally hard even in the realizable setting. However, under distributional assumptions, this problem is solvable even when

we have agnostic noise. Specifically, we will use the following theorems for approximating halfspaces and intersections of halfspaces.

Theorem 6.1 ([KOS08])**.** For any $f : \mathbb{R}^n \to \{-1, 1\}$ intersection of halfspaces. There exist a polynomial $p : \mathbb{R}^n \to \mathbb{R}$ with degree $d = O(\frac{\log K}{\varepsilon^2})$ such that

$$\mathop{\mathbb{E}}_{x \sim \mathcal{N}^n} \left[(p(x) - f)^2\right] \leq \varepsilon$$

From the Cauchy-Schwartz inequality $\mathop{\mathbb{E}}_{x \sim D} [\|p(x) - c(x)\|_1] \leq \sqrt{\mathop{\mathbb{E}}_{x \sim D} [\|p(x) - c(x)\|_2^2]}$ we can immediately get the following corollary.

Corollary 6.2 ([KOS08])**.** For any $f : \mathbb{R}^n \to \{-1, 1\}$ intersection of halfspaces. There exist a polynomial $p : \mathbb{R}^n \to \mathbb{R}$ with degree $d = O(\frac{\log K}{\varepsilon^4})$ such that

$$\mathop{\mathbb{E}}_{x \sim \mathcal{N}^n} [|p(x) - f|] \leq \varepsilon$$

Theorem 6.3 ([KOS08])**.** For any $f : \mathbb{R}^n \to \{-1, 1\}$ halfspace. There exist a polynomial $p : \mathbb{R}^n \to \mathbb{R}$ with degree $d = O(\frac{1}{\varepsilon^2})$ such that the l_1-norm under the Gaussian distribution is less than ε.

These theorems describe the existence of low-degree polynomials that approximate halfspaces and intersections of halfspaces. Moreover, the theorems above are almost tight and there are lower bounds that characterize that this is the near-optimal (optimal in the case of halfspaces) in terms of degree polynomials that approximate these concepts [DKN09], [Hsu+22]. Also one can prove Statistical Query lower bounds using lower bounds on the degree of approximation by polynomials [Dia+21] in the case of binary classification. Therefore we have strong evidence to postulate that the algorithms that we have designed are not far from the optimal algorithms in the SQ framework.

6.2 One Versus All Learning Algorithm

We will investigate the guarantees of the following learning algorithm 1. Summarizing this algorithm learns a polynomial that approximates the region of every class then rounds the answer according to a threshold and it predicts according to the One Versus All (OVA) paradigm 2. As sign (ϕ), where ϕ a proposition, we define the function that is 1 if ϕ is true and -1 if ϕ is false.

Algorithm 1: OVAtrain(d degree, $S = \{(x_i, y_i)\}_1^m$ examples)

Result: $h_1, \ldots, h_K$ polynomial threshold functions

for $i = 1 : K$ **do**

  Let $z_j = \text{sign}\,(y_j = i)\,, j = 1, \ldots, m$ and let $S' = \{(x_j, z_j)\}_1^m$

  Find the polynomial of degree d than minimizes the l_1 distance over S' i.e.

$$\min_{p_i:\deg(p_i)\leq d} \frac{1}{m}\sum_1^m |p_i(x_j) - z_j|$$

  Choose $t_i \in [-1, 1]$ such that we minimize

$$\frac{1}{m}\sum_1^m \mathbb{1}\left(\text{sign}\left(p_i(x_j) - t_i\right) \neq z_j\right)$$

end

return $\text{sign}\,(p_1(x) - t_1)\,, \ldots, \text{sign}\,(p_K(x) - t_K)$

Algorithm 2: OVAtest($h_1, \ldots, h_K, x$) or $h(x)$

Result: $\hat{y}$ prediction of a sample x

for $i = 1 : K$ **do**

  If $h_i(x) \geq 0$

  return i

end

return 1

Theorem 6.4. Assuming that there exists a polynomial of degree d that approximates every intersection of halfspaces to l_1-error ε under the distribution D_X then we have that, the training algorithm 1 with the prediction algorithm 2, by the use of $N = O(\frac{n^d}{\varepsilon})$ samples in poly (N) time output a hypothesis with expected error less than $K\varepsilon + 2OPT$.

Furthermore, we can boost the algorithm such that the error is less than $K\varepsilon + 2OPT$ with probability at least $1 - \delta$ by the use of polynomial in $\frac{1}{\varepsilon}, \frac{1}{\delta}$ number of samples in polynomial time.

Proof. We will set d as the degree and we use $O(\frac{n^d}{\varepsilon})$ samples (the VC-dimension of degree d polynomial threshold functions is $O(n^d)$) and run the l_1-regression algorithm 1. It is true that for all $i \in [K]$:

$$\frac{1}{m}\sum_1^m \mathbb{1}\left(h_i(x_j) \neq \text{sign}\,(y_j = i)\right) \leq \frac{1}{2m}\sum_1^m |p_i(x_j) - \text{sign}\,(y_j = i)\,|$$

Because, we would make a mistake with respect to sign $(y = i)$ if and only if the threshold lay inside the interval $[p_i(x), \text{sign}\,(y = i)]$. Thus a random threshold would achieve error $\mathop{\mathbb{E}}_{(x,y)\sim D}\left[|p_i(x) - \text{sign}\,(y = i)\,|\right]/2$ in expectation thus by the probabilistic method there exists a threshold that has a lower error and thus the optimum has a lower error with certainty.

Now let c be the optimal multiclass linear model and c_i the intersections of halfspaces that each class defines. As we choose the polynomials that minimize the l_1

error we will have polynomials $p_1, \ldots, p_K$ that have better l_1 distance from the sign functions than the polynomials that approximate the intersections of halfspaces $\{c_i\}_1^K$, $p_1^*, \ldots, p_K^*$. In other words, we have that

$$\begin{aligned}
\frac{1}{m}\sum_1^m |p_i(x_j) - \text{sign}\,(y_j = i)\,| &\leq \frac{1}{m}\sum_1^m |p_i^*(x_j) - \text{sign}\,(y_j = i)\,| \\
&\leq \frac{1}{m}\sum_1^m |p_i^*(x_j) - c_i(x_j)| + \frac{1}{m}\sum_1^m |\text{sign}\,(y_j = i) - c_i(x_j)| \\
&\leq \frac{1}{m}\sum_1^m |p_i^*(x_j) - c_i(x_j)| + \frac{1}{m}\sum_1^m |\text{sign}\,(y_j = i) - c_i(x_j)| \\
&= \frac{1}{m}\sum_1^m |p_i^*(x_j) - c_i(x_j)| + \frac{2}{m}\sum_1^m \mathbb{1}\left(\text{sign}\,(y = i) \neq c_i(x)\right), \forall i \in [K]
\end{aligned}$$

The last inequality is because when an error occurs then the l_1 error is equal to 2. From the above my taking the expectation we have that

$$\Pr_{(x,y)\sim D}\left[h_i(x) \neq \text{sign}\,(y = i)\right] \leq \frac{\varepsilon}{2} + \Pr_{(x,y)\sim D}\left[\text{sign}\,(y = i) \neq c_i(x)\right]$$

As we chose the number of samples to be related to the VC dimension of PTFs of degree at most d we have that the empirical error is $\varepsilon/2$ close to the above expectation in an expected sample S. Thus

$$\mathop{\mathbb{E}}_S\left[\Pr_{(x,y)\sim D}\left[h_i(x) \neq \text{sign}\,(y = i)\right]\right] \leq \varepsilon + \mathop{\mathbb{E}}_S\left[\Pr_{(x,y)\sim D}\left[\text{sign}\,(y = i) \neq c_i(x)\right]\right], \forall i \in [K]$$

So in expectation over the sample set S we have that

$$\begin{aligned}
\Pr_{(x,y)\sim D}\left[h(x) \neq y\right] &\leq \Pr_{(x,y)\sim D}\left[h_y(x) < 0 \text{ or } \bigvee_{i\neq y} h_i(x) \geq 0\right] \\
&= \Pr_{(x,y)\sim D}\left[\bigvee_i h_i \neq \text{sign}\,(y = i)\right] \\
&\leq \sum_i \Pr_{(x,y)\sim D}\left[h_i(x) \neq \text{sign}\,(y = i)\right] \\
&\leq \sum_i \left(\varepsilon + \Pr_{(x,y)\sim D}\left[\text{sign}\,(y = i) \neq c_i(x)\right]\right) \\
&= K\varepsilon + \sum_i \Pr_{(x,y)\sim D}\left[\text{sign}\,(y = i) \neq c_i(x)\right] \\
&= K\varepsilon + 2OPT
\end{aligned}$$

The last equality is because as $\{c_i\}_1^K$ constitute a partition of $\mathbb{R}^n$ only one of them can be equal to 1 the same is true for sign $(y = i)$. Hence if we have a mistake then the sum of the indicators $\mathbb{1}\left(c_i(x) \neq \text{sign}\,(y = i)\right)$ is equal to 2 the one that corresponds to the correct label and the one of the prediction. But if the is no mistake then all events are

falls thus the sum of the indicators is 0.

$$\begin{aligned} OPT &= \Pr_{(x,y)\sim D}[\text{sign}\,(y \neq c(x))] \\ &= \Pr_{(x,y)\sim D}[\exists i : c_i(x) \neq \text{sign}\,(y = i)] \\ &= \mathop{\mathbb{E}}_{(x,y)\sim D}[\mathbb{1}\,(\exists i : c_i(x) \neq \text{sign}\,(y = i))] \\ &= \mathop{\mathbb{E}}_{(x,y)\sim D}\left[\frac{1}{2}\sum_i \mathbb{1}\,(c_i(x) \neq \text{sign}\,(y = i))\right] \\ &= \frac{1}{2}\sum_i \Pr_{(x,y)\sim D}[\text{sign}\,(y = i) \neq c_i(x)] \end{aligned}$$

Furthermore, high accuracy on an expected sample can be easily boosted to high accuracy with high confidence in a standard way [Kal+05].

□

Hence from theorem 6.2 we have the following corollary.

Corollary 6.5. For D_x being the normal distribution we have that if run the algorithm above with $d = \widetilde{O}(\frac{K^4 \log K}{\varepsilon^4})$ and using n^d samples then with probability at least $1 - \delta$ we get a hypothesis with error at most $2OPT + \varepsilon$ in time polynomial on n^d, $\frac{1}{\varepsilon}$, $\frac{1}{\delta}$.

If we assume that K is constant then we have an $n^{\frac{1}{\varepsilon^4}}$ learning algorithm. Notice that the analysis carried out above works for any prediction rule that returns an arbitrary label that has been predicted by the binary models. Also by applying distribution-specific agnostic boosting algorithms [Fel09] combined with the corollary above we can get the optimum accuracy, but only for the case when the optimum is less than $1/4$.

In the following paragraphs, we will see how we can get optimum accuracy by designing a more multiclass-specific task and rounding technique.

6.3 One Versus All Shortcomings

In the coming paragraphs, we will use the vector notation for all classifiers rounded and real-valued. So let p, p^* be K dimensional vectors of polynomials. Let c be the K dimensional one-hot vector of the optimum classifier and y the one-hot vector of labels.

In the one versus all paradigm essentially we trained and combined each classifier separately and as the task was well decomposable to binary tasks we got good accuracy. Essentially we got regression polynomials $\{p_i\}_1^K$ with l_1 error lower than the l_1 error of the true classifiers $\{c_i\}_1^K$.

$$\frac{1}{m}\sum_1^m \|p(x_j) - y_j\|_1 \leq \frac{1}{m}\sum_1^m \|p^*(x_j) - c(x_j)\|_1 + \frac{1}{m}\sum_1^m \|c(x_j) - y_j\|_1, \forall i \in [K]$$

In the coming discussion, we will not consider the l_1 error of p^* with c, as by taking a sufficient degree we can force it to be less than ε. And as we rounded each classifier separately we were led to a classifier a collection of classifiers $\{h_i\}_1^K$ that in l_1 error less than that of c.

$$\frac{1}{m}\sum_1^m \|h(x_j) - y_j\|_1 \leq \frac{1}{m}\sum_1^m \|p^*(x_j) - c(x_j)\|_1 + \frac{1}{m}\sum_1^m \|c(x_j) - y_j\|_1, \forall i \in [K]$$

However, the problem with the collection of classifiers h is that it does not necessarily lie in the simplex, so we can not predict by sampling. Hence we need to project to it without increasing the l_1 distance from y a one-hot vector that belongs in the simplex. Unfortunately, the projection lemma that is utilized for the projected gradient descent guarantees does not apply here. As the vector $y = [1, 0, 0]$ and $h = [1, 1/2, 1/2]$ have $\|h - y\|_1 = 1$ and both $[1, 0, 0]$ and $[0, 1/2, 1/2]$ are valid projections but have l_1 distances from y being 0 and 2 respectively.

Lemma 6.6 (l_2 Projection Lemma)**.** Let $C \subseteq \mathbb{R}^d$ any convex body let $y \in C$ and $x \notin C$ then for the projection of x to C:

$$\Pi_C(x) = \text{argmin}_{z \in C} \|x - z\|_2$$

we have that $\|\Pi_C(x) - y\|_2 \leq \|x - y\|_2$.

Hence we would need to make a custom projection method that does not blow up the l_1 error from all one-hot vectors. Let us consider projecting p, we can truncate all negative values to zero by only decreasing the l_1 norm also assuming that the sum of the vectors is greater than 1 then we can round by decreasing all the weights by the same amount. As y has only one non-zero coordinate we have that we decrease the l_1 distance by $(K-1)x$ if we subtract from all coordinates x and remain nonnegative. As soon as one coordinate becomes 0 we keep it constant and continue from the rest.

Hence there exists a way to project onto the simplex while keeping decreasing the l_1 norm. The only other case that we need to consider is when having the rounded classifier h is when all h_i's are 0. So we have that $\|h - y\|_1 = 1$ Then we have no information so the only logical thing to do is to flip a random K-sided coin and output $\hat{h}$ the corresponding one-hot vector. That way we will have a probability of $1 - 1/K$ to make a mistake and hence have $\|\hat{h} - y\|_1 = 2$. Thus we have increased the l_1 distance from y by 1 for an $(1 - \frac{1}{K})\Pr[h = 0]$ fraction of examples. This is the fundamental issue for the naive One Versus All reduction, that the classifiers have not been incentivized to collaborate and produce rounded outcomes and thus they can all respond negatively that does not give any information about the example.

Theorem 6.7. For any $A : \{0,1\}^K \times \{0,1\}^r \to \{0,1\}^K$ multiclass to binary One Versus All reduction algorithm using r random bits there exists $h_1, \ldots, h_K : \mathbb{R}^n \to \{0,1\}$ classifiers such that for all i it holds

$$\Pr_{(x,y)\sim D}[h_i(x) \neq y_i] \leq \Pr_{(x,y)\sim D}[c_i^*(x) \neq y_i] + \epsilon$$

i.e. they are optimal in classification error.But there exists a distribution D on $\mathbb{R}^d \times \{0,1\}^K$ for which

$$\Pr_{(x,y)\sim D, r\sim\{0,1\}^r}[A(h_1(x), \ldots, h_K(x), r) \neq y] \geq 2(1 - \frac{1}{K})OPT + \varepsilon.$$

Proof. We define a distribution D as a product of a distribution on one point x and a distribution that belongs to a family of distributions $\mathcal{F}$. Where $\mathcal{F} = \{\frac{1}{2}(e_i + e_j) : i, j \in [K], i \neq j\}$ and $\{e_i\}_1^K$ are the canonical basis vectors of $\mathbb{R}^K$. The all-zero classifies have optimal classification error for each distribution in $\mathcal{F}$ as they observe each label with probability $\frac{1}{2}$ in the one versus all distributions. Thus we have to prove that the classifier $A(0, \ldots, 0, r)$ has small accuracy against one distribution from the family $\mathcal{F}$.

Let $(p_1, \ldots, p_K)$ the probability distributions on labels that $A(0, \ldots, 0, r)$ generates given random bits r. The average classification error over the class $\mathcal{F}$ is

$$\begin{aligned}
\frac{1}{\binom{K}{2}} \sum_{i>j} \left(1 - \frac{p_i + p_j}{2}\right) &= 1 - \frac{1}{\binom{K}{2}} \sum_{i>j} \left(\frac{p_i + p_j}{2}\right) \\
&= 1 - \frac{1}{\binom{K}{2}} \sum_{i>j} p_i \\
&= 1 - \frac{1}{2\binom{K}{2}} \sum_{i \neq j} p_i \\
&= 1 - \frac{1}{K(K-1)} (K-1) \sum_{i=1}^{K} p_i \\
&= 1 - \frac{1}{K} \\
&= 2\left(1 - \frac{1}{K}\right)\frac{1}{2}
\end{aligned}$$

Hence by the probabilistic method, we have that there exists a distribution in the family for which classification error is greater than $2\left(1 - \frac{1}{K}\right)\frac{1}{2}$. The result follows as $OPT = \frac{1}{2}$. □

6.4 An Optimal Agnostic Learner

In this section, we will describe a way to overcome this obstacle and get optimum accuracy with the same complexity as the OVA algorithm.

6.4.1 Rounding Procedure

We will consider minimizing the objective function $L(p) = \mathop{\mathbb{E}}_{(x,y)\sim D}[l(p(x), y)]$ where

$$l(p, y) = \|p - y\|_1 + |1 - \sum_{i=1}^{K} p| + \sum_{i=1}^{K} (-p_i)_+$$

In that way, we force our p_i's to sum up to one and also to have positive values. Specifically, it is true that for this objective the function value gets smaller when we project to the simplex using algorithm 3. Hence even if we receive a point x that the polynomial values do not form a probability distribution then we could round without increasing the objective value.

Algorithm 3: $R(p \in \mathbb{R}^K)$

Result: $p' \in \Delta_K$ a probability distribution vector

if $p \in \Delta_K$ **then**
  return p
end
$\forall i : p_i < 0$ set $p_i = 0$
let $s = \sum_{i=1}^{K} p_i$
if $s < 1$ **then**
  $\forall i, p_i = p_i + \frac{s}{K}$
end
else if $s > 1$ **then**
  while $s > 1$ **do**
    Let $S = \{i : p_i > 0\}$
    $\forall i \in S, p_i = p_i - \min\left(\frac{s-1}{K}, \min_{j \in S} p_j\right)$
    $s = \sum_{i=1}^{K} p_i$
  end
end
return p

Theorem 6.8. Let $y \in \{e_i\}_1^K$. There exists a procedure A such that from any p we could get a vector $p' \in \Delta_K$ such that $l(p', y) \leq l(p, y)$.

Proof. We will use the algorithm 3. If there exist coordinates of p that are negative we can increase them separately by that amount by only decreasing the objective value. Let $p_i = -x$ then by making $p_i = 0$ we decrease the first and third terms by x and as the middle term has slope 1 for p_i so we decrease the function by at least x. Thus the third term drops to zero.

Now if the $\sum_i p_i = x \neq 1$ is larger than 1 then decreasing it in any way to 1 drops the second term. If the sum is greater than one we can use the technique of the previous paragraph to also decrease the l_1 difference. If $x < 1$ then by increasing every coordinate by x/K we increase the l_1 difference by at most x which is also the decrease of the second term. □

The theorem below connects the objective value loss and the classification error of a rounded hypothesis.

Theorem 6.9. There exists a rounding procedure that given for any hypothesis $h : \mathbb{R}^n \to \mathbb{R}^K$ it outputs a hypothesis $\hat{h} : \mathbb{R}^n \to \{e_i\}_1^K$ such that

$$\Pr_{(x,y)\sim D}\left[\hat{h}(x) \neq y\right] \leq \frac{L(h)}{2}$$

Proof. Assume the hypothesis $\hat{h}$ that given any x computes $h(x)$ and rounds by running the Algorithm 3 and then samples according to that probability distribution (Algorithm

5). With the use of Theorem 6.8 we have that:

$$\begin{aligned} L(h) &= \mathop{\mathbb{E}}_{(x,y)\sim D}\left[\|h(x)-y\|_1 + |1-\sum h_i(x)| + \sum(-h_i(x))_+\right] \\ &\geq \mathop{\mathbb{E}}_{(x,y)\sim D}\left[\|R(h(x))-y\|_1 + |1-\sum R(h(x))_i| + \sum(-R(h(x))_i)_+\right] \\ &= \mathop{\mathbb{E}}_{(x,y)\sim D}[\|R(h(x))-y\|_1] \end{aligned}$$

Now let's consider the relationship between, the accuracy that a random one hot vector sampled from a distribution p has being equal to a fixed one hot vector y, and the l_1 distance between p and y.

$$\begin{aligned} \|p-y\|_1 &= \sum_{i=1}^{K}|p_i-y_i| \\ &= |p_j-1| + \sum_{i\neq j}|p_i| \\ &= 2\sum_{i\neq j}p_i \\ &= 2\Pr_{i\sim p}[i\neq j] \end{aligned}$$

Hence

$$\begin{aligned} L(h) &\geq \mathop{\mathbb{E}}_{(x,y)\sim D}[\|R(h(x))-y\|_1] \\ &= \mathop{\mathbb{E}}_{(x,y)\sim D}\left[2\Pr_{\hat{y}\sim R(h(x))}[\hat{y}\neq y]\right] \\ &= 2\Pr_{(x,y)\sim D}\left[\hat{h}(x)\neq y\right] \end{aligned}$$

The last probability also takes into account the random coins used by $\hat{h}$. □

6.4.2 Overall Algorithm

The pseudocode for the training and prediction routines is given in Algorithm 4 and 5. To give guarantees about the algorithms we will first show that there exists polynomials of low degree that achieve a small loss.

Theorem 6.10. Let D a distribution on $\mathbb{R}^d \times [K]$ such that the x-marginal is a standard normal. There exist polynomials $p_1, \ldots, p_K$ with degree at most $O(\frac{K^2 \log K}{\varepsilon^4})$ that achieve $L(p) \leq 2OPT + 3\varepsilon$.

Proof. For any vector of polynomials $p = [p_1, \ldots, p_K]$ and c optimum multiclass clas-

sifier we have that

$$\begin{aligned}
L(p) &= \mathop{\mathbb{E}}_{(x,y)\sim D}\left[\|p-y\|_1 + |1-\sum p_i| + \sum(-p_i)_+\right] \\
&= \mathop{\mathbb{E}}_{(x,y)\sim D}\left[\|c-y\|_1 + \|c-p\|_1 + |1-\sum p_i| + \sum(-p_i)_+\right] \\
&= \mathop{\mathbb{E}}_{(x,y)\sim D}\left[\|c-y\|_1 + \|p-y\|_1 + |\sum c_i - \sum p_i| + \sum(-p_i)_+\right] \\
&\leq \mathop{\mathbb{E}}_{(x,y)\sim D}\left[\|c-y\|_1 + 2\|p-c\|_1 + \sum(-p_i)_+\right] \\
&\leq \mathop{\mathbb{E}}_{(x,y)\sim D}\left[\|c-y\|_1 + 3\|p-c\|_1\right]
\end{aligned}$$

The last inequality is true because the negative coordinates of p also contribute to the l_1 the same amount, as c has only one non-zero coordinate.

Now as the x-marginal is the standard normal distribution and each coordinate of c represents an intersection of halfspaces from Theorem 6.1 we have that there exist polynomials of degree at most $O(\frac{\log K}{\varepsilon^2})$ that can approximate each coordinate to l_2^2 distance ε. Thus with degree $O(\frac{K^2 \log K}{\varepsilon^4})$ we can approximate the vector c to l_2^2 distance ε^2. Combining the fact that

$$OPT = \Pr_{x\sim D_x}[c(x) \neq y] = \mathop{\mathbb{E}}_{(x,y)\sim D}\left[\|c-y\|_1\right]/2$$

we have that $L(p) \leq 2OPT + 3\varepsilon$. □

Now given the Theorems 6.9, 6.10 the only thing left to show is that minimization of the objective function L can be done efficiently and with a small number of samples. In the work [Kal+05] as well as in the One Versus All analysis, optimization is done on the empirical l_1 loss and then VC dimension arguments are used in order uniformly bound the population loss. Here as we use a more complicated rounding procedure that does not produce polynomial threshold functions this is not an option.

To bound the complexity of minimization of L we will use a custom concentration argument and also the fact that the empirical analog of L is a convex program (that can be minimized efficiently).

Theorem 6.11. With $N = \frac{(nd)^{O(d)}}{K\varepsilon^2}$ samples and poly $\left(N, n^d, \frac{1}{\varepsilon}\right)$ runtime where $d = \frac{K^2 \log K}{\varepsilon^4}$, we can compute a vector of polynomials $\hat{p}$ such that

$$L(\hat{p}) \leq \min_{p:deg(p_i)<d} L(p) + O(\varepsilon)$$

Proof. First, we show that there exists an almost optimal polynomial that has bounded coefficients. From the proof of the last theorem 6.10, we have that there exists a vector of polynomials p^* of degree $d = O(\frac{K^2 \log K}{\varepsilon^4})$ that is ε-optimal for L and also for every i

$$\begin{aligned}
\mathop{\mathbb{E}}_{(x,y)\sim D}\left[p_i^*(x)^2\right] &= \mathop{\mathbb{E}}_{(x,y)\sim D}\left[(p_i^*(x) - c(x) + c(x))^2\right] \\
&= \mathop{\mathbb{E}}_{(x,y)\sim D}\left[2(p_i^*(x) - c(x))^2 + 2c(x)^2)\right] \\
&= 2\frac{\varepsilon^2}{K} + 2 \\
&\leq 3
\end{aligned}$$

If we write $p_i(x)$ in the Hermite basis $p_i(x) = \sum_{a \in \mathbb{N}} c_a H_a(x)$, it holds that $\sum_a c_a^2 = \mathop{\mathbb{E}}_{(x,y)\sim D}\left[p_i^*(x)^2\right]$. As the one-dimensional Hermite polynomials of degree k are

$$h_k(z) = \sum_{m=0}^{\lfloor \frac{K}{2} \rfloor} \frac{(-1)^m z^{k-2m}}{m!(n-2m)!2^m}$$

Thus, each monomial has a coefficient absolute bounded by 2^k. Therefore, the maximum coefficient of a multidimensional Hermite polynomial $H_a(x)$ is $2|a|$, thus the maximum coefficient of the polynomial is bounded by a constant $B = O(2^d)$. Let P be the set of polynomials of degree d in n dimensions with coefficients bounded by B also let $p_i(x) = \sum_j m_j(x) a_{ij}$ the sum of monomials form of p_i.

Let S be a set of N samples. We define the empirical loss of a vector of polynomials as

$$L_N(p) = \frac{1}{N} \sum_{i=1}^{N} l(p(x_i), y_i)$$

Our goal is to show that the empirical loss $L_N(p)$ is close to the population loss $L(p)$ for the output polynomial p of the optimization algorithm for L_N. The minimization of L_N subject to $|a| < B$ for all a coefficients can be formulated as a linear program (convex) and thus it can be solved with additive error ε in poly $\left(N, n^d, \frac{1}{\varepsilon}\right)$. Define the random variables $\{X_i\}_{i=1}^K$ such that

$$X_i = |\frac{1}{N} \sum_{(x,y)\in S} |p_i(x) - y| - \mathop{\mathbb{E}}_{(x,y)\sim D} [|p_i(x) - y|] \,|$$

As X_i is non-negative by Markov's Inequality we have

$$\begin{aligned}
\Pr_{S\sim D^N}[X_i \geq \delta] &\leq \frac{\mathop{\mathbb{E}}_{S\sim D^N}[X_i]}{\delta} \\
&\leq \frac{\sqrt{\mathop{\mathbb{E}}_{S\sim D^N}[X_i^2]}}{\delta} \\
&= \frac{\sqrt{\mathop{\mathrm{Var}}_{S\sim D^N}[X_i]}}{\delta} \\
&= \frac{\sqrt{\mathop{\mathrm{Var}}_{S\sim D}[X_i]}}{\sqrt{N}\delta} \\
&\leq \frac{\sqrt{\mathop{\mathbb{E}}_{S\sim D}\left[\left(|p_i(x)-y| - \mathop{\mathbb{E}}_{(x,y)\sim D}[|p_i(x)-y|]\right)^2\right]}}{\sqrt{N}\delta} \\
&\leq \frac{\sqrt{2\mathop{\mathbb{E}}_{S\sim D}\left[|p_i(x)-y|^2\right] + 2\left(\mathop{\mathbb{E}}_{(x,y)\sim D}[|p_i(x)-y|]\right)^2}}{\sqrt{N}\delta} \\
&\leq \frac{2\sqrt{2\mathop{\mathbb{E}}_{x\sim\mathcal{N}^d}\left[p_i(x)^2\right]+2}}{\sqrt{N}\delta} \\
&\leq \frac{2\sqrt{2\mathop{\mathbb{E}}_{x\sim\mathcal{N}^d}\left[\left(\sum_j a_{ij}m_j(x)\right)^2\right]+2}}{\sqrt{N}\delta} \\
&\leq \frac{2\sqrt{4\mathop{\mathbb{E}}_{x\sim\mathcal{N}^d}\left[\left(\sum_j a_{ij}^2 m_j(x)^2\right)\right]+2}}{\sqrt{N}\delta} \\
&\leq \frac{2\sqrt{4B^2\sum_j \mathop{\mathbb{E}}_{x\sim\mathcal{N}^d}[(m_j(x)^2)]+2}}{\sqrt{N}\delta} \\
&\leq \frac{2\sqrt{4B^2\sum_j \mathop{\mathbb{E}}_{x\sim\mathcal{N}^d}\left[(\|x\|_2^{2j})\right]+2}}{\sqrt{N}\delta} \\
&\leq \frac{2\sqrt{4B^2\sum_j O(j^j)+2}}{\sqrt{N}\delta} \\
&\leq \frac{2\sqrt{4B^2 n^{d+1}O(d^d)+2}}{\sqrt{N}\delta} \\
&\leq O(\frac{\sqrt{B^2 n^{d+1}d^d}}{\sqrt{N}\delta})
\end{aligned}$$

Here we used the fact that the number of coefficients of all degree d multivariate polynomial in n dimensions is n^{d+1}. Using $N = O(\frac{B^2 n^{d+1}d^d}{\varepsilon^2}) = \frac{(nd)^{O(d)}}{\varepsilon^2}$ we can make

this probability constant. By with $N = \frac{(nd)^{O(d)}}{K\varepsilon^2}$ we can do this for all X_i simultaneously. Working in the same way we can do this for the other two components of L. Thus using $N = \frac{(nd)^{O(d)}}{K\varepsilon^2}$ we can have that $\forall p$ with constant probability

$$|L_N(p) - L(p)| \leq \varepsilon$$

Now by solving the linear program, we get a polynomial p such that

$$\begin{aligned} & L_N(p) \leq \min_{q \in P} L_N(q) + \varepsilon \\ \Rightarrow & L_N(p) \leq L_N(p^*) + \varepsilon \\ \Rightarrow & L(p) - \varepsilon \leq L_N(p^*) + \varepsilon \\ \Rightarrow & L(p) \leq L(p^*) + 3\varepsilon \\ \Rightarrow & L(p) \leq 2OPT + O(\varepsilon) \end{aligned}$$

with constant probability. Substituting the degree of p^* we get $N = \left(n\frac{K^2 \log K}{\varepsilon^4}\right)^{O(\frac{K^2 \log K}{\varepsilon^4})} / \varepsilon^2$. □

The above result holds with constant probability but it can be efficiently boosted to δ by only multiplying the number of samples with $O(\log \frac{1}{\delta})$ ([SSBD14] Chapter 13 exercise 1). Also, the formulation of the minimization of L_N as a linear program is nearly the same as the formulation of l_1 polynomial regression as a linear program.

Algorithm 4: AgnosticLearner(d bound degree, $S = \{(x_i, y_i)\}_1^m$ examples, B bound on the coefficients)

Result: p polynomial

Find an ε approximate minimizer $p = \sum_{a \subseteq \mathbb{N}} c_a x^a$ of the following problem via linear programming

$$\begin{aligned} & \min_{p_i : \deg(p_i) \leq d} \frac{1}{|S|} \sum_i l(p(x_i), y_i) \\ & \text{s.t. } |c_a| \leq B \end{aligned}$$

return p

Algorithm 5: Predict(p, x) or $\hat{h}(x)$

Result: $\hat{y}$ prediction of a sample x

Compute $h = p(x)$

Round according to R (Algorithm 3)

return $\hat{y} \sim R(h)$

6.5 Approximating The Multiclass Model

The approach in the previous paragraph was to minimize a function of the one-hot vectors describing the labels. This is the advisable approach as minimizing the distance from the integer-valued labels takes to account the relative distance between the labels. And as the labels represent categorical values they should be on orthogonal directions

like the one-hot vectors. However, despite that, we tightly bound the degree of the multiclass polynomial that approximates the integer-valued labels. But we note that this approach leads to $O(K \cdot OPT)$ accuracy.

6.5.1 Existance of a low-degree approximating polynomial

We saw the guarantee of the One Versus all learning algorithm was $2OPT + \varepsilon$ and the times two factor on the approximation error came from the fact that we trained multiple learners in this section we will describe how one can construct a polynomial that has small l_1 error with respect to the multiclass labels.

Specifically, we will show that there is a linear combination of the intersection of halfspace polynomials that has ε error with respect to the l_1 norm. Consider the $K \times K$ matrix H

$$H = \begin{bmatrix} 1 & -1 & \dots & -1 \\ -1 & 1 & \dots & -1 \\ & & \dots & \\ -1 & -1 & \dots & 1 \end{bmatrix} = 2I - J$$

where J is the all-ones matrix. The above matrix has eigenvalues 2 and $2 - K$ as J has eigenvalues 0 (with algebraic multiplicity $K - 1$) and K (with algebraic multiplicity 1). Hence the system $Hx = b$ is always solvable for every $K \geq 3$.

Theorem 6.12. For every K-class linear model c there exists a polynomial p of degree $O(\frac{1}{\varepsilon^4})$ that approximates it to error ε in l_1 distance with respect to the Gaussian distribution.

Proof. Let c_i the partition to intersections of halfspaces of c. From theorem **??** we have $p_1, \dots, p_K$ polynomials that approximate c_i to error $\varepsilon_1, \dots, \varepsilon_K$ respectively in l_1 distance with respect to the Gaussian distribution. We will show that the polynomial $p = v^T P(x)$ where P the vector with coordinates $\{p_i(x)\}$ and $v : Hv = b$ with b the

vector with values $1, \ldots, K$ satisfies the requirements of the theorem.

$$\begin{aligned}
\mathop{\mathbb{E}}_{x\sim N^n}[|p(x)-c(x)|] &= \sum_{i=1}^{K} \mathop{\mathbb{E}}_{x\sim N^n}[|p(x)-c(x)|\cdot \mathbb{1}(c_i(x)=1)] \\
&= \sum_{i=1}^{K} \mathop{\mathbb{E}}_{x\sim N^n}\left[|v^T P - b_i|\cdot \mathbb{1}(c_i(x)=1)\right] \\
&= \sum_{i=1}^{K} \mathop{\mathbb{E}}_{x\sim N^n}\left[|v^T P - v^T H_i|\cdot \mathbb{1}(c_i(x)=1)\right], \ H_i \text{ the i'th row of } H \\
&\leq \sum_{i=1}^{K}\sum_{j=1}^{K} |v_j| \mathop{\mathbb{E}}_{x\sim N^n}[|p_j - H_{ij}|\cdot \mathbb{1}(c_i(x)=1)] \\
&= \sum_{j=1}^{K} |v_j| \sum_{i=1}^{K} \mathop{\mathbb{E}}_{x\sim N^n}[|p_j - c_j(x)|\cdot \mathbb{1}(c_i(x)=1)] \\
&= \sum_{j=1}^{K} |v_j| \mathop{\mathbb{E}}_{x\sim N^n}[|p_j - c_j(x)|] \\
&= \sum_{j=1}^{K} |v_j|\varepsilon_j \\
&\leq \varepsilon
\end{aligned}$$

If we set $\varepsilon_j = \frac{\varepsilon}{|v_j|}$. As p is a linear combination of the polynomials $p_1, \ldots, p_K$ so we have that $\deg(p) \leq \max_i \deg(p_i)$. Hence we have that the degree of p is $O(\frac{\max_i v_i^4 \log K}{\varepsilon^4})$. The values $\{|v_i|\}_1^K$ are independent with respect to ε but they depend on K. □

6.5.2 Lower Bounds on the Approximation Degree

This algorithm may not be tight as we solve the much harder problem of learning with intersections of halfspaces instead of directly approximating the multiclass linear classifier. However, in this paragraph, we will show that the polynomial that we specified is optimal with respect to the degree. This intuitively makes sense as the function of the multiclass linear model can be partitioned to regions that are intersections of halfspaces.

Specifically, we will show that if a multiclass linear model could be approximated to error ε with a low degree polynomial then intersections of halfspaces could be approximated also by a polynomial with the same degree. This reduction along with lower bounds on the degree of polynomials that approximate intersections of halfspaces can give us a lower bound for the degree of polynomials that can sufficiently approximate multiclass linear models.

Theorem 6.13. If for every function in the class of K-class linear models, there exists a polynomial of degree at most $d_K(\frac{1}{\varepsilon})$ that approximates it to l_1 error ε then there exists a polynomial of degree at most $d_{K+1}(O(\frac{1}{\varepsilon}))$ that approximates every intersection of K halfspaces.

Proof. Assuming that there exist polynomials of degree at most d_K that approximate every K-class linear model. Let c an intersection of K halfspaces $c(x) = (a_1^T x > 0) \wedge$

$\cdots \wedge (a_K^T x > 0)$. Consider the multiclass linear model f_w with weights $w \in \mathbb{R}^{n \times (K+1)}$

$$\begin{aligned} w_1 &= a_1 + \cdots + a_K \\ w_2 &= -a_1 + \cdots + a_K \\ &\cdots \\ w_{K+1} &= a_1 + \cdots - a_K \end{aligned}$$

We have that x is classified as 1 if and only if

$$\begin{aligned} & w_1^T x > w_i^T x, \forall i \in \{2, \ldots, K+1\} \\ \iff & a_i^T x \geq 0, \forall i \in [K] \\ \iff & c(x) = 1 \end{aligned}$$

From our assumption, there exist polynomials of degree at most d_{K+1} that approximate w as well as all renamings of the labels of w. We will show that there exists a linear combination of the cyclic renamings of w that approximates c. Consider the following linear system $Hv = b$ where

$$H = \begin{bmatrix} 1 & 2 & \ldots & K+1 \\ K+1 & 1 & \ldots & K \\ & & \ldots & \\ 2 & 3 & \ldots & 1 \end{bmatrix} \text{ and } b = \begin{bmatrix} 1 \\ -1 \\ \ldots \\ -1 \end{bmatrix}$$

it is easy to see that the above system is always invertible due to the circulant nature of H. Consider the polynomial $p = v^T P$ where P is the vector of polynomials that approximate the associated cyclic renamings of w. Let $w = w^1, \ldots, w^{K+1}$ signify the $K+1$ cyclic renamings of w and let $p_1, \ldots, p_K$ the polynomials that approximate them to error $\{\varepsilon_i\}_1^{K+1}$. We have that

$$\begin{aligned} \mathop{\mathbb{E}}_{x \sim N^n} \left[|c(x) - v^T P| \right] &= \sum_{i=1}^{K+1} \mathop{\mathbb{E}}_{x \sim N^n} \left[|c(x) - v^T P| \cdot \mathbb{1}\left(f_w(x) = i\right) \right] \\ &= \sum_{i=1}^{K+1} \mathop{\mathbb{E}}_{x \sim N^n} \left[|b_i - v^T P| \cdot \mathbb{1}\left(f_w(x) = i\right) \right] \\ &= \sum_{i=1}^{K+1} \mathop{\mathbb{E}}_{x \sim N^n} \left[|v^T H_i - v^T P| \cdot \mathbb{1}\left(f_w(x) = i\right) \right] \\ &\leq \sum_{i=1}^{K+1} \sum_{j=1}^{K+1} |v_j| \mathop{\mathbb{E}}_{x \sim N^n} \left[|H_{ij} - p_j(x)| \cdot \mathbb{1}\left(f_w(x) = i\right) \right] \\ &= \sum_{i=1}^{K+1} \sum_{j=1}^{K+1} |v_j| \mathop{\mathbb{E}}_{x \sim N^n} \left[|f_{w^j}(x) - p_j(x)| \cdot \mathbb{1}\left(f_w(x) = i\right) \right] \\ &= \sum_{j=1}^{K+1} |v_j| \mathop{\mathbb{E}}_{x \sim N^n} \left[|f_{w^j}(x) - p_j(x)| \right] \\ &\leq \sum_{j=1}^{K+1} |v_j| \varepsilon_j \\ &= \varepsilon \end{aligned}$$

The last inequality can be achieved if we set $\varepsilon_i = \varepsilon/|v_i|, \forall i \in [K]$, thus $\frac{1}{\varepsilon_i} = O(\frac{1}{\varepsilon})$. As p is a linear combination of the polynomials $\{p_i\}_{K+1}$ we have that its degree is at most the maximum so at most $d_{K+1}(O(\frac{1}{\varepsilon}))$. □

From the above theorem, we can see that in order approximate multiclass linear models with sufficient accuracy we need to approximate the corresponding intersections of halfspaces like the method of the above paragraph.

CHAPTER 7

FUTURE WORK AND OVERVIEW

Concluding we studied algorithms for multiclass learning in the presence of corruption and produced efficient algorithms for specific assumptions for learning with coarse and even noisy data. However, despite our efforts, there are still a large number of important open problems one should consider.

As multiclass learning is underrepresented in the literature there is a large number of problems that can be investigated. We divide these problems into two categories: semi-supervised multiclass learning and multiclass learning in the presence of corruption.

In the first category, we classify learning problems that concern the setting of learning with Coarse Labels. Specifically the problems of learning under the ε-UB, $\varepsilon(x)$-UB and $\alpha(x)$-IP problems. As well as the problem of PU learning when we always observe the positive label with some lower bounded probability. These settings can help speed up the Machine Learning pipeline as we would need substantially less labeling effort by learning in a semi-supervised manner.

In the second category, we classify learning problems where the labels have been imposed on some type of corruption. Namely the problems of learning with multiclass RCN and Massart noise. These settings can help the development of more robust trustworthy algorithms as in practice the realizable setting is quite unrealistic.

REFERENCES

[AL88] Dana Angluin and Philip Laird. ``Learning from noisy examples". In: *Machine Learning* 2 (1988), pp. 343–370.

[ASS01] Erin L. Allwein, Robert E. Schapire, and Yoram Singer. ``Reducing Multiclass to Binary: A Unifying Approach for Margin Classifiers". In: *J. Mach. Learn. Res.* 1 (2001), pp. 113–141. ISSN: 1532-4435. DOI: 10.1162/15324430152733133. URL: https://doi.org/10.1162/15324430152733133.

[Bey+19] Alina Beygelzimer et al. ``Bandit Multiclass Linear Classification: Efficient Algorithms for the Separable Case". In: *CoRR* abs/1902.02244 (2019). arXiv: 1902.02244. URL: http://arxiv.org/abs/1902.02244.

[BKW00] Avrim Blum, Adam Kalai, and Hal Wasserman. ``Noise-Tolerant Learning, the Parity Problem, and the Statistical Query Model". In: *CoRR* cs.LG/0010022 (2000). URL: https://arxiv.org/abs/cs/0010022.

[Blu+98] Avrim Blum et al. ``A polynomial-time algorithm for learning noisy linear threshold functions". In: *Algorithmica* 22.1 (1998), pp. 35–52.

[Che+20] Sitan Chen et al. ``Classification Under Misspecification: Halfspaces, Generalized Linear Models, and Connections to Evolvability". In: *CoRR* abs/2006.04787 (2020). arXiv: 2006.04787. URL: https://arxiv.org/abs/2006.04787.

[Coh97] E. Cohen. ``Learning Noisy Perceptrons by a Perceptron in Polynomial Time". In: *Proceedings of the 38th Annual Symposium on Foundations of Computer Science*. FOCS '97. USA: IEEE Computer Society, 1997, p. 514. ISBN: 0818681977.

[CST11] Timothee Cour, Ben Sapp, and Ben Taskar. ``Learning from Partial Labels". In: *Journal of Machine Learning Research* 12.42 (2011), pp. 1501–1536. URL: http://jmlr.org/papers/v12/cour11a.html.

[Dan15a] Amit Daniely. *A PTAS for Agnostically Learning Halfspaces*. 2015. arXiv: 1410.7050 [cs.DS].

[Dan15b] Amit Daniely. ``Complexity Theoretic Limitations on Learning Halfspaces". In: *CoRR* abs/1505.05800 (2015). arXiv: 1505.05800. URL: http://arxiv.org/abs/1505.05800.

[DB95] Thomas G. Dietterich and Ghulum Bakiri. ``Solving Multiclass Learning Problems via Error-Correcting Output Codes". In: *J. Artif. Int. Res.* 2.1 (1995), pp. 263–286. ISSN: 1076-9757.

[Dec97] Scott E. Decatur. ``PAC Learning with Constant-Partition Classification Noise and Applications to Decision Tree Induction". In: *Proceedings of the Sixth International Workshop on Artificial Intelligence and Statistics*. Ed. by David Madigan and Padhraic Smyth. Vol. R1. Proceedings of Machine Learning Research. Reissued by PMLR on 30 March 2021. PMLR, 1997, pp. 147–156. URL: https://proceedings.mlr.press/r1/decatur97a.html.

[DGL05] François Denis, Rémi Gilleron, and Fabien Letouzey. ``Learning from positive and unlabeled examples". In: *Theoretical Computer Science* 348.1 (2005), pp. 70–83.

[DGT19] Ilias Diakonikolas, Themis Gouleakis, and Christos Tzamos. ``Distribution-Independent PAC Learning of Halfspaces with Massart Noise". In: *CoRR* abs/1906.10075 (2019). arXiv: 1906.10075. URL: http://arxiv.org/abs/1906.10075.

[Dia+21] Ilias Diakonikolas et al. ``The Optimality of Polynomial Regression for Agnostic Learning under Gaussian Marginals". In: *CoRR* abs/2102.04401 (2021). arXiv: 2102.04401. URL: https://arxiv.org/abs/2102.04401.

[Dia+22] Ilias Diakonikolas et al. ``Learning General Halfspaces with Adversarial Label Noise via Online Gradient Descent". In: *Proceedings of the 39th International Conference on Machine Learning*. Ed. by Kamalika Chaudhuri et al. Vol. 162. Proceedings of Machine Learning Research. PMLR, 2022, pp. 5118–5141. URL: https://proceedings.mlr.press/v162/diakonikolas22b.html.

[DK19] Ilias Diakonikolas and Vasilis Kontonis. *Lecture Notes on Computational Learning Theory*. 2019.

[DKN09] Ilias Diakonikolas, Daniel M. Kane, and Jelani Nelson. ``Bounded Independence Fools Degree-2 Threshold Functions". In: *CoRR* abs/0911.3389 (2009). arXiv: 0911.3389. URL: http://arxiv.org/abs/0911.3389.

[DSS12] Amit Daniely, Sivan Sabato, and Shai Shwartz. ``Multiclass learning approaches: A theoretical comparison with implications". In: *Advances in Neural Information Processing Systems* 25 (2012).

[Fel09] Vitaly Feldman. ``Distribution-Specific Agnostic Boosting". In: *CoRR* abs/0909.2927 (2009). arXiv: 0909.2927. URL: http://arxiv.org/abs/0909.2927.

[Fot+21] Dimitris Fotakis et al. ``Efficient Algorithms for Learning from Coarse Labels". In: *CoRR* abs/2108.09805 (2021). arXiv: 2108.09805. URL: https://arxiv.org/abs/2108.09805.

[HL20] Zayd Hammoudeh and Daniel Lowd. ``Learning from Positive and Unlabeled Data with Arbitrary Positive Shift". In: *Advances in Neural Information Processing Systems*. Ed. by H. Larochelle et al. Vol. 33. Curran Associates, Inc., 2020, pp. 13088–13099. URL: https://proceedings.neurips.cc/paper_files/paper/2020/file/98b297950041a42470269d56260243a1-Paper.pdf.

[Hsu+22] Daniel Hsu et al. *Near-Optimal Statistical Query Lower Bounds for Agnostically Learning Intersections of Halfspaces with Gaussian Marginals*. 2022. arXiv: 2202.05096 [cs.LG].

[Ish+17] Takashi Ishida et al. ``Learning from complementary labels". In: *Advances in neural information processing systems* 30 (2017).

[Kal+05] A.T. Kalai et al. ``Agnostically learning halfspaces". In: *46th Annual IEEE Symposium on Foundations of Computer Science (FOCS'05)*. 2005, pp. 11–20. DOI: 10.1109/SFCS.2005.13.

[Kea98] Michael Kearns. ``Efficient noise-tolerant learning from statistical queries". In: *Journal of the ACM (JACM)* 45.6 (1998), pp. 983–1006.

[Kon+23] Vasilis Kontonis et al. *SLaM: Student-Label Mixing for Semi-Supervised Knowledge Distillation*. 2023. arXiv: 2302.03806 [cs.LG].

[KOS08] Adam R. Klivans, Ryan O'Donnell, and Rocco A. Servedio. ``Learning Geometric Concepts via Gaussian Surface Area". In: *2008 49th Annual IEEE Symposium on Foundations of Computer Science*. 2008, pp. 541–550. DOI: 10.1109/FOCS.2008.64.

[KS06] Adam R. Klivans and Alexander A. Sherstov. ``Cryptographic Hardness for Learning Intersections of Halfspaces". In: *2006 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS'06)*. 2006, pp. 553–562. DOI: 10.1109/FOCS.2006.24.

[KS08] Subhash Khot and Rishi Saket. ``On Hardness of Learning Intersection of Two Halfspaces". In: *Proceedings of the Fortieth Annual ACM Symposium on Theory of Computing*. STOC '08. Victoria, British Columbia, Canada: Association for Computing Machinery, 2008, pp. 345–354. ISBN: 9781605580470. DOI: 10.1145/1374376.1374426. URL: https://doi.org/10.1145/1374376.1374426.

[LD14] Liping Liu and Thomas Dietterich. ``Learnability of the Superset Label Learning Problem". In: *Proceedings of the 31st International Conference on Machine Learning*. Ed. by Eric P. Xing and Tony Jebara. Vol. 32. Proceedings of Machine Learning Research 2. Bejing, China: PMLR, 2014, pp. 1629–1637. URL: https://proceedings.mlr.press/v32/liug14.html.

[MN06] Pascal Massart and É lodie Nédélec. ``Risk bounds for statistical learning". In: *The Annals of Statistics* 34.5 (2006). DOI: 10.1214/009053606000000786. URL: https://doi.org/10.1214%2F009053606000000786.

[Nat+13] Nagarajan Natarajan et al. ``Learning with Noisy Labels". In: *Advances in Neural Information Processing Systems*. Ed. by C.J. Burges et al. Vol. 26. Curran Associates, Inc., 2013. URL: https://proceedings.neurips.cc/paper_files/paper/2013/file/3871bd64012152bfb53fdf04b401193f-Paper.pdf.

[Pat+17] Giorgio Patrini et al. ``Making deep neural networks robust to label noise: A loss correction approach". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 1944–1952.

[PNS14] Marthinus C du Plessis, Gang Niu, and Masashi Sugiyama. ``Analysis of Learning from Positive and Unlabeled Data". In: *Advances in Neural Information Processing Systems*. Ed. by Z. Ghahramani et al. Vol. 27. Curran Associates, Inc., 2014. URL: https://proceedings.neurips.cc/paper_files/paper/2014/file/35051070e572e47d2c26c241ab88307f-Paper.pdf.

[RDM06] Liva Ralaivola, François Denis, and Christophe Magnan. ``CN = CPCN." In: vol. 148. June 2006, pp. 721–728. DOI: 10.1145/1143844.1143935.

[RW18] Brendan van Rooyen and Robert C. Williamson. ``A Theory of Learning with Corrupted Labels". In: *Journal of Machine Learning Research* 18.228 (2018), pp. 1–50. URL: http://jmlr.org/papers/v18/16-315.html.

[Sou+22] Daniel Soudry et al. *The Implicit Bias of Gradient Descent on Separable Data*. 2022. arXiv: 1710.10345 [stat.ML].

[SSBD14] Shai Shalev-Shwartz and Shai Ben-David. *Understanding Machine Learning: From Theory to Algorithms*. USA: Cambridge University Press, 2014. ISBN: 1107057132.

[SSSS10] Shai Shalev-Shwartz, Ohad Shamir, and Karthik Sridharan. *Learning Kernel-Based Halfspaces with the Zero-One Loss*. 2010. arXiv: 1005.3681 [cs.LG].

[Yu+18] Xiyu Yu et al. ``Learning with biased complementary labels". In: *Proceedings of the European conference on computer vision (ECCV)*. 2018, pp. 68–83.

[Zha+21] Yiyang Zhang et al. ``Learning from a complementary-label source domain: theory and algorithms". In: *IEEE Transactions on Neural Networks and Learning Systems* 33.12 (2021), pp. 7667–7681.