Weakly Supervised Learning via Statistical Sufficiency

Giorgio Patrini

A thesis submitted for the degree of
Doctor of Philosophy
The Australian National University

December 2016

Except where otherwise indicated, this thesis is my own original work.

Giorgio Patrini
20 December 2016

Abstract

The Thesis introduces a novel algorithmic framework for weakly supervised learning, namely, for any any problem in between supervised and unsupervised learning, from the labels standpoint. Weak supervision is the reality in many applications of machine learning where training is performed with partially missing, aggregated-level and/or noisy labels. The approach is grounded on the concept of statistical sufficiency and its transposition to loss functions. Our solution is problem-agnostic yet constructive as it boils down to a simple two-steps procedure. First, estimate a sufficient statistic for the labels from weak supervision. Second, plug the estimate into a (newly defined) linear-odd loss function and learn the model by any gradient-based solver, with a simple adaptation. We apply the same approach to several challenging learning problems: (i) learning from label proportions, (ii) learning with noisy labels for both linear classifiers and deep neural networks, and (iii) learning from feature-wise distributed datasets where the entity matching function is unknown.

Contents

Chapter 1

Introduction

1.1 Thesis summary

Supervised learning is by far the most effective application of the machine learning paradigm. However, its success in modern real word challenges is undermined by the fundamental assumption that we have perfect knowledge of the target variable, the label, at training time. Despite the unprecedented pace of accumulation of digital datasets, *labeled data* is still rare, due to several reasons: supervision is often generated by costly human annotation; relevant information may be obfuscated by privacy mechanisms; or, merely, labels are only known at a higher level of instance/temporal granularity with the respect to the one required for training. In addition to those issues, data collection is affected by ubiquitous noise corruption of various nature and therefore supervision is never perfectly reliable. As a consequence, learning is often performed with sparse, aggregated-level and/or noisy training labels. We consider all those scenarios all under the name of *weakly supervised learning*.

In this Thesis, we approach this generic learning problem through the lens of Statistics and Learning Theory. Our solution is problem-agnostic yet constructive and it boils down to a simple two-steps procedure. First, estimate a sufficient statistic of the unknown labels from weak supervision; this quantity is called *mean operator*. Second, plug the estimate into a standard loss function and learn the model by any gradient-based solver. The key to the second step is the definition of a family of losses that we call *linear-odd*. Its elements can be computed without the need of any label, given their sufficient statistic, by virtue of what we term *loss factorization*. Several commonly used losses belong to the family, *e.g.* logistic and square. From the theoretical viewpoint, linear-odd losses shed new light on generalization bounds with Rademacher complexity: the contribution of the supervision is isolated into one single term, which accounts for the deviation of the sufficient statistic from its population mean. From the algorithmic viewpoint, we bypass the need of estimating actual labels and therefore circumvent the difficult bi-convex optimization problem arising from naïvely modeling labels as latent variables. To put this into practice, a well-behaved estimator of the sufficient statistic remains to be defined. This is achieved depending on the particular nature of the weak supervision and relative assumptions.

We study in detail three scenarios of weak supervision. The first is called *learning*

from label proportions, where nothing is known about the target variable but its proportion over subsets of the training set, the *bags*. This setting is inspired by several applications where individual labels are not available, but ratios and percents are easy to estimate for domain experts, or given by other sources in terms of aggregates, *e.g.* surveys, census data. Despite the poor label knowledge, it is possible to learn linear classifiers with strong theoretical guarantees and good practical performance. We work under a weak distinguish-ability assumption of bags, which relaxes a similar but stricter condition in literature. In light of our two-step framework, the problem is reduced to estimating the mean operator from the label proportions. We do so by least square minimization with a manifold regularizer, which expresses our geometrical assumption on the data. A finite sample guarantee is given: the estimator is all the better as the maximum feature vectors norm increases. We name this algorithm the Laplacian Mean Map (LMM): after estimation, a standard loss function computed with the mean operator is minimized. The model output of LMM enjoys a data dependent approximation bound with respect to an ideal classifier learned with full supervision.

We then propose an iterative algorithm, the Alternating Mean Map (AMM). It takes the solution of LMM as input and optimizes it further over the set of labelings consistent with the proportions, implementing coordinate descent minimization. This simple idea can highly improve the quality of the final model in practice, yet it does not suffer some of the drawbacks of the popular bi-convex iterative optimization. In fact, the LMM initializer performs significantly better than random vectors and is deterministic by definition. Moreover, computing the latent labels that match with the proportions while minimizing the loss can be efficiently done via sorting. Finally, we also formulate a specialized uniform convergence bound, involving a generalization of Rademacher complexity for learning from label proportions. The result includes a bag-wise surrogate risk for which we show that AMM optimizes a tractable bound. We experiment on UCI domains with up to hundreds of thousands of examples, comparing the algorithms to previous work. We simulate bags and their label proportions and retain labels for test set performance. Results display that AMM and LMM outperform the state of the art and sometimes even compete with the fully supervised learner, while requiring few proportions only. Tests on the largest domains display the scalability of both algorithms.

The second weakly supervised scenario is the case of *asymmetric label noise*. In the binary classification setting that we considered, the noise model assumes that one label is flipped into the other at random by class-dependent noise rates. The setting is a well-studied approximation of real-world corruption of categorical labels, by the effect of human mistakes in data annotation or automated data extraction. A known recipe exists for correcting the loss by making it unbiased under asymmetric label noise. Thanks to loss factorization, we can directly apply it to the mean operator. We demonstrate that any algorithm minimizing a linear-odd loss computed with the unbiased mean operator enjoys a generalization bound that tightens results from prior work. We also characterize the whole family of linear-odd losses with an approximate robustness property: the difference in average risk (under the noise) between

ideal model and ours cannot be arbitrarily large and the bound is data dependent by the mean operator. On the algorithmic side, we show how to adapt stochastic gradient descent and proximal algorithms to handle weak supervision. Once the mean operator has been estimated, the modification only requires a change of inputs and to sum up the mean operator to the model update. The theory is validated with experiments on UCI domains, on which we inject artificial noise. We assume to know the noise rates for the first step of estimation. Our approach of loss correction is effective: we obtain a significant gain in performance with respect to standard loss minimization with the same algorithm. Interestingly, in the presence of very high noise (one noise rate close to 50%) we still obtain sensible models, while learning with no loss correction often results in random guessing.

We study further the asymmetric label noise setting and consider multi-class classification with deep neural networks, including recurrent neural networks. Once more, the only component we operate on is the loss function, thus our solution is actually independent from any chosen architecture. Although, with neural networks, our two-step learning procedure requires more care. Here the sufficient statistic is computed upon intermediate feature representations that need to be learned and therefore cannot be estimated *a priori*. Still, even with multiple classes, what is sufficient for loss correction is the probability of flipping each class into any other, namely a transition matrix. Given the matrix, we propose two types of loss corrections; one follows the extension of the above idea to multi-class, the other is instead inspired by prior work on robust Deep Learning. Both corrections amounts at most to one inversion and multiplication of the noise matrix. Therefore, we show how to compute the noise rates from (noisy) data as the first step of learning. In particular, we adapt a recently published technique for noise estimation to the multi-class scenario. The whole learning process is summarized as follows: train the chosen neural network with standard loss using corrupted labels; exploit it to estimate the transition matrix; finally re-train the network with the corrected loss. Experiments on MNIST, IMDB, CIFAR-10, CIFAR-100 and a large scale dataset of clothing images show that a diversity of architectures — stacking dense, convolutional, pooling, dropout, batch normalization, word embedding, LSTM and residual layers — demonstrate noise robustness for image recognition and sentiment analysis. Incidentally, we also prove that, when ReLU is the only non-linearity, the loss curvature is immune to label noise.

Finally, we explore an extension of the framework to the challenging problem of *learning from distributed datasets*, where examples are "vertically" (feature-wise) partitioned and the "who-is-who" correspondence is unknown. As a motivating example, we may imagine two peer institutions, *e.g.* a bank and an insurance company, which aim to exploit the joint predictive power of their data assets (expressed on different feature spaces), based on the knowledge that they share many customers. Our goal is to learn a classifier in the cross product space of the two domains, in the hard case when no shared ID is available; companies may not know which partial views are about the same customer in the two datasets or, more likely, matching is not permitted by privacy regulation. We work under two assumptions: all labels are known

and some of the features are shared between the two domains. This can be seen as a peculiar setting of weakly supervised learning, since the mapping between each feature vector and relative label (although known) is effectively missing.

Traditionally, the problem would be approached by first solving (approximate) entity matching and subsequently learning the classifier in a standard manner. Instead, following the underlying philosophy of the Thesis, we bypass the problem of estimating the unknown variables and look for sufficient statistics. In this setting, we require a different sort of loss factorization, expressed by the recently introduced concept of Rademacher observations (rados). Those statistics closely related to the mean operator and indeed can be thought as sufficient for subsets of examples — not just for labels — in the data. We replace the minimization of a loss over examples, which requires entity resolution, by the *equivalent* minimization of a loss over rados. In general, the number of rados is exponential in the sample size. We show that a large subset of these rados does not require to perform entity matching and can be easily obtained linking together partial views of examples with the same value of shared features. With a focus on square loss, we prove that optimization on those rados has time and space complexities smaller than the algorithm minimizing the equivalent square loss on examples. Last, we relax the key assumption that the data is vertically partitioned among peers — in this case, we would not even know the existence of a solution to entity resolution. In this more general setting, experiments support the possibility of beating the optimal peer in hindsight.

1.2 Organization and originality

Except Chapter 2 on the background, the content of the Thesis is original or excerpt from novel results either published or submitted to academic conferences from myself and collaborators. In particular, Chapter 3 elaborates the theory of loss functions discussed in Patrini et al. [2016a] including some formal results from Patrini et al. [2014]; it can be thought as a toolbox of abstract results to be specialized for particular instances of weak supervision. Chapter 4 deals with the special case of learning from label proportions, that is the subject of Patrini et al. [2014]. Chapters 5 and 6 discuss the case of learning with asymmetric noisy labels as in Patrini et al. [2016a] (theory, linear and kernel models) and in Patrini et al. [2017] (neural networks, multi-class, noise estimation). Chapter 7 extends the theory from Chapter 3 to the problem of distributed learning. This is a revised presentation from the original work of Patrini et al. [2016b]; it also requires some background material from Nock et al. [2015], that we have co-authored. Results recalled from prior work are accompanied by the relative citations. Literature reviews and related work are topic-specific and presented at the end of each Chapter, along with most of the proofs and additional formal and experimental results.

1.3 First-author publications included in the Thesis

Patrini, G.; Nock, R.; Rivera, P.; and Caetano, T., (Almost) no label no cry. In *NIPS*, 2014.

Patrini, G.; Nielsen, F.; Nock, R.; and Carioni, M., Loss factorization, weakly supervised learning and label noise robustness. In *ICML*, 2016.

Patrini, G.; Nock, R.; Hardy, S.; and Caetano, T., Fast learning from distributed datasets without entity matching. In *IJCAI*, 2016.

Patrini, G.; Rozza, A.; Menon, A.; Nock, R.; and Qu, L., Making deep neural networks robust to label noise: a loss correction approach. Submitted to CVPR, 2017.

1.4 Contributed publications

Nock, R.; Patrini, G.; and Friedman, A., Rademacher observations, private data and boosting. In *ICML*, 2015.

Muzellec, B.; Nock, R.; Patrini, G.; and Nielsen, F., Tsallis regularized optimal transport and ecological inference. In *AAAI*, 2017.

Chapter 2

Background

The Chapter recalls basic notions of supervised Machine Learning, Learning Theory and Statistics, and presents a critical review of weakly supervised learning as tackled by prior work. Notations, formal statements and proofs of this introductory material are loosely inspired by the book of Shalev-Shwartz and Ben-David [2014].

2.1 Preliminary notation

We begin by defining some notation that will be used throughout the Thesis. $\mathbb{R}_+$ is the set of non-negative real numbers. For any $m \in \mathbb{N}$, the sequence of positive natural number up to m is $[m] \doteq \{1, \dots, m\}$. Boldfaces like $\boldsymbol{v}$ indicate (column) vectors; $\boldsymbol{v}_i$ (boldface) is the ith element of a sequence of vectors instead; v_i may either be the ith component of vector $\boldsymbol{v}$ or the ith element of a sequence of scalars $\{v_1, v_2, \dots\}$. $\mathbf{1}$ is the vector of all ones, with size determined by context, and $\boldsymbol{e}^i$ is the indicator vector, that is 1 only at the ith position.

Capital letters like A can indicate matrices, or sometimes constants scalar, depending on the context; $A_{i\cdot}$ and $A_{\cdot j}$ are respectively row $i \in [m]$ and column $j \in [n]$ of matrix $A \in \mathbb{R}^{m \times n}$. $\mathbf{0}$ is either a vector or a matrix filled with zeros, with size determined by context. I is the identity matrix. $\mathrm{diag}(\boldsymbol{v})$ is diagonal matrix with $\boldsymbol{v}$ on the diagonal. $\mathrm{tr}(A)$ is the trace of matrix A.

For any function $f : \mathbb{R} \to \mathbb{R}$, we denote its first derivative at x by $f'(x)$; the subdifferential set, that is, the set of all subdifferentials at point x is denoted by $\partial f(x)$. For vector functions $f : \mathbb{R}^d \to \mathbb{R}$, we indicate gradient at point $\boldsymbol{x}$ by $\nabla f(\boldsymbol{x})$.

We denote with $1\{p\}$ the indicator function of a predicate p, which is 1 when TRUE. $[v]_+$ is $\max(0, v)$, for any scalar v. Inner products are written by angular brackets $\langle \cdot, \cdot \rangle$. Sets and sequences are italic capital letter like $\mathcal{S}$. The probability of an event is denoted by $\mathbb{P}(\cdot)$. The expectation over a distribution $\mathcal{D}$ is denoted as $\mathbb{E}_{\mathcal{D}}[\cdot]$; the same notation is for empirical averages on a sample $\mathcal{S}$ as $\mathbb{E}_{\mathcal{S}}[\cdot] \doteq 1/|\mathcal{S}| \sum \cdot$. We also refer to the set $\Sigma_m \doteq \{-1, 1\}^m$, the m-times Cartesian product of $\{-1, 1\}$.

2.2 The supervised learning problem

In Supervised Machine Learning we learn a map between two spaces by looking at examples. Learning Theory studies the role of those objects within the process of learning. Let set $\mathcal{X}$ be the input space set $\mathcal{Y}$ the output space. Elements $\boldsymbol{x} \in \mathcal{X} \subseteq \mathbb{R}^d$ are called *feature vectors, observations* or *instances* and elements $y \in \mathcal{Y}$ are called *labels*. The function $h : \mathcal{X} \to \mathcal{Y}$, named *hypothesis* or *model*, belongs to a hypothesis space $\mathcal{H}$. Any pair $(x, y) \in \mathcal{X} \times \mathcal{Y}$ is an example.

It is usual to frame the problem of learning in the language of probability. We assume that examples are drawn *i.i.d* from an unknown but fixed distribution $\mathcal{D}$. A *(learning) sample* $\mathcal{S}$ of size m is a finite sequence of examples $\{(\boldsymbol{x}_i, y_i), i \in [m]\}$. Notice that despite the commonly used "set-like" notation, $\mathcal{S}$ is actually a sequence, which in particular implies that examples can be repeated. At the same time, since examples are drawn *i.i.d.*, their order does not strictly matter, except sometimes for notational convenience in defining algorithms. In regression, the output space is the real line, *i.e.* $\mathcal{Y} = \mathbb{R}$. In classification, the output space is instead discrete and labels are also called *classes*. For instance in binary classification, a main focus of this work, it is common to represent the output variable as taking values in $\{-1, 1\}$.

The goal of supervised learning is to find a hypothesis that generalizes well on unseen examples. Thus we first ought to define what is the quality measure for hypotheses. In classification, we refer to the *generalization error*, or *risk*, the average number of labels correctly predicted by the model: $\mathbb{E}_{\mathcal{D}}\, 1\{h(\boldsymbol{x}) = y\}$. We could therefore define a *learner* as any algorithm minimizing the generalization error as:

$$h^\star \doteq \operatorname*{argmin}_{h:\mathcal{X}\to\mathcal{Y}} \mathbb{E}_{\mathcal{D}}\, 1\{h(\boldsymbol{x}) \neq y\} \tag{2.1}$$

The minimizer of this problem is called *Bayes optimal* and its risk is the *Bayes risk*. This formulation is problematic for several reasons. First, we have mentioned that the distribution $\mathcal{D}$ is unknown to the learner. Learning can only make use of the sample $\mathcal{S}$, which provides an empirical version of the objective of Equation 2.1, namely the *empirical risk* $\mathbb{E}_{\mathcal{S}}\, 1\{h(\boldsymbol{x}) \neq y\}$. Second, we need to define what the model space $\mathcal{H}$ is. The choice of the model space is crucial for the success of learning, as discussed below. For most of the Thesis we consider one of the simplest hypothesis space, *i.e.* the set of linear classifiers. To make this explicit, we will denote the model by a vector $\boldsymbol{\theta} \in \mathbb{R}^d$ as $h(\cdot) \doteq \langle \boldsymbol{\theta}, \cdot \rangle$. This often allows to simplify formal arguments while still capturing essential properties. We introduce more complex model spaces in Chapter 6, that deals with deep neural networks, and Chapter 3, which briefly touches on non-parametric kernel methods. Third, in practice it would be difficult to optimize directly the error function, which is discontinuous and non-convex. Instead, it is common to resort to surrogate objectives called *loss functions*.

A loss $\ell : \mathcal{X} \times \mathcal{Y} \times \mathcal{H} \to \mathbb{R}$ measures the disagreement between labels and model predictions, although not necessarily by strict inequality, *e.g.* logistic and square losses (in formulae given below). By extension from the empirical risk, we call *empirical ℓ-risk*, or sometimes *surrogate* empirical risk, the average loss $\mathbb{E}_{S}\, \ell(y, h(\boldsymbol{x}))$.

We denote ℓ-risks and empirical counterparts by $R_{\mathcal{D},\ell}(h)$ and $R_{\mathcal{S},\ell}(h)$ respectively. The error itself can be thought as the risk a certain loss function, the $0/1$ *loss*, $\ell_{01} = 1\{yh(\boldsymbol{x}) < 0\}$. As a result, we now formulate the *empirical risk minimization* (ERM) framework, a fundamental paradigm in Machine Learning:

$$\underset{h\in\mathcal{H}}{\operatorname{argmin}}\ R_{\mathcal{S},\ell}(h) \tag{2.2}$$

One final important issue is hidden in the ERM formulation, that of *overfitting*. By searching for a model *in the learning sample*, we may end up learning all too well particular variations of $\mathcal{S}$, but with no guarantee that the model may generalize on unseen examples of $\mathcal{D}$. Intuitively, when the size of the learning sample is small relatively to the model complexity, the model could memorize the whole sample itself, and overfitting is a potential threat. When obtaining more data for learning is not an option, overfitting can be combatted in several ways. We may replace the hypothesis space $\mathcal{H}$ with one containing model with smaller capacity; or equivalently, we may express a "preference" over the elements of $\mathcal{H}$ by balancing the empirical risk with an additional term acting as a regularizer $\Omega : \mathcal{H} \to \mathbb{R}_+$. We update the learning framework of Equation 2.2 to the *structural risk minimization* to:

$$\underset{h\in\mathcal{H}}{\operatorname{argmin}}\ R_{\mathcal{S},\ell}(h) + \lambda \cdot \Omega(h) \tag{2.3}$$

with $\lambda > 0$ balancing the two contributions. As we show below, Ω may also be a function of on the learning sample, depending on the kind of prior knowledge one needs to express in the problem.

2.3 Learning Theory

Loss functions have been studied extensively in the Machine Learning and Statistics literature. It is common to assume some preconditions for defining losses that are amenable of study: non-negativity and convexity are definitely the most widely required. In our work, we do not strictly work under those conditions but we will make it explicit when particular results require them. Here we limit ourself to recall some definitions and properties that are relevant for what is discussed in the Thesis.

One first important requirement for loss functions is *properness*.

Definition 1. *A loss $\ell(y, h(\boldsymbol{x}))$ is* proper *when:*

$$\underset{h}{\operatorname{argmin}}\, \mathbb{E}_{y\sim p(y|\boldsymbol{x})}\ \ell(y, h(\boldsymbol{x})) = p(y|\boldsymbol{x})\ . \tag{2.4}$$

This requirement states that the optimal model should be the conditional probability of the class, given the observation. This way, models fitted via proper losses are actual probability estimators for the class.

We may define the domain of a loss function to be $\mathcal{X} \times \mathcal{Y} \times \mathcal{H}$. In practice, this parameterization is seldom necessary. It is common to focus on a subset of loss

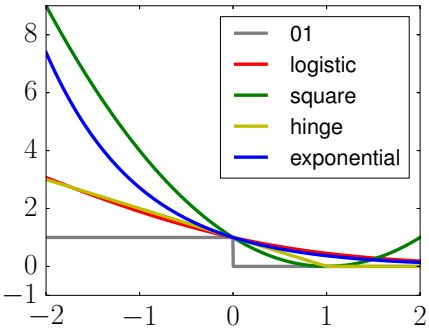


Figure 2.1: Losses as function of $x = yh(\boldsymbol{x})$. Logistic is scaled by $1/\log(2)$ to be an upper bound of 01 loss, which is irrelevant for optimization.

functions called *margin losses*.

Definition 2. *A* margin loss *is defined as* $\ell(yh(\boldsymbol{x})) \doteq \ell(\boldsymbol{x}, y, h)$.

When clear by the context, we simply use a generic scalar argument $\ell(x)$. Margin losses are implicitly *symmetric* since $\ell(yh(\boldsymbol{x}))) = \ell(-y \cdot (-h(\boldsymbol{x})))$. Examples are $0/1$ loss $1\{x < 0\}$, square loss $(1-x)^2$, logistic loss $\log(1+e^{-x})$ (logistic regression), hinge loss $[1-x]_+$ (SVM) and exponential loss e^{-x} (boosting). Figure 2.1 shows how these losses are all designed to be upper bounds of $0/1$ loss, a sensible strategy for optimizing the quantity that we ultimately wish to minimize. Among those margin losses, it is well known that hinge loss is not proper and cannot naturally be used as a class probability estimator [Platt, 1999].

Sometimes we will consider losses that are *strongly convex* functions, which is a common assumption to facilitate the derivation of desired upper bounds.

Definition 3. *Let* $\gamma > 0$. *A differentiable function* $f(\boldsymbol{x})$ *is* γ-strongly convex *if for any* $\boldsymbol{x}, \boldsymbol{x}' \in Dom(f)$ *we have:*

$$f(\boldsymbol{x}) - f(\boldsymbol{x}') \geq \langle \nabla f(\boldsymbol{x}'), \boldsymbol{x} - \boldsymbol{x}' \rangle + \frac{\gamma}{2} \left\| \boldsymbol{x} - \boldsymbol{x}' \right\|_2^2 \ . \tag{2.5}$$

Similarly, it is common to refer to *L-Lipschitz* functions, that are such that $|f(\boldsymbol{x}) - f(\boldsymbol{x}')| \leq L\|\boldsymbol{x} - \boldsymbol{x}'\|$ for any $\boldsymbol{x}, \boldsymbol{x}'$. A comprehensive discussion on loss functions for binary classification is Reid and Williamson [2010].

2.3.1 Generalization bounds and Rademacher complexity

We recall standard results on uniform convergence of learning, taken from Bartlett and Mendelson [2002]; Kakade et al. [2009]; Shalev-Shwartz and Ben-David [2014]. The aim is to provide generalization bounds for ERM. We ground the theory on the concept of Rademacher complexity, a modern data dependent form of complexity of the model space. Several formal results of the Thesis will extend what follows.

Definition 4. *Let σ be a random variable drawn* i.i.d. *from $\{\pm 1\}$ with uniform probability. The* empirical Rademacher complexity *of a hypothesis space $\mathcal{H}$ with regard to sample $\mathcal{S}$ of size m and loss ℓ is:*

$$\mathcal{R}(\ell \circ \mathcal{H} \circ \mathcal{S}) \doteq \mathbb{E}_{\sigma \sim \Sigma_m} \left[\frac{1}{m} \sup_{h \in \mathcal{H}} \sum_{i=1}^{m} \sigma_i \, \ell(y_i h(x_i)) \right] , \tag{2.6}$$

while its population version $\mathcal{R}(\ell \circ \mathcal{H}) \doteq \mathbb{E}_{\mathcal{S} \sim \mathcal{D}} \, \mathcal{R}(\ell \circ \mathcal{H} \circ \mathcal{S})$ is the Rademacher complexity.

We recall two types of bound that we will use in the Thesis. For the sake of completeness, we prove them in the Appendix of this Chapter; proofs rely heavily on the application of McDiarmid's inequality [McDiarmid, 1998].

Theorem 5. *Assume a bounded loss,* i.e. $|\ell(x)| \leq C, \forall x$. *Then for any $\delta \in (0,1)$, with probability at least $1 - \delta$ over the choice of $\mathcal{S}$, simultaneously for all $h \in \mathcal{H}$, we have:*

$$R_{\mathcal{D},\ell}(h) - R_{\mathcal{S},\ell}(h) \leq 2\mathcal{R}(\ell \circ \mathcal{H}) + C\sqrt{\frac{2}{m} \log \frac{1}{\delta}} \, . \tag{2.7}$$

Moreover, let the empirical risk minimizer be $\hat{h} = \operatorname{argmin}_{h \in \mathcal{H}} R_{\mathcal{S},\ell}(h)$. For any $\delta \in (0,1)$, with probability least $1 - \delta$ over the choice of $\mathcal{S}$, we have:

$$R_{\mathcal{D},\ell}(\hat{h}) - \inf_{h \in \mathcal{H}} R_{\mathcal{D},\ell}(h) \leq 4\mathcal{R}(\ell \circ \mathcal{H}) + 2C\sqrt{\frac{2}{m} \log \frac{1}{\delta}} \, . \tag{2.8}$$

Proof in 2.6.1. This first inequality limits the difference between risk and its empirical approximation; this probabilistic guarantee corroborates the idea of fitting the model on a finite but not too small sample $\mathcal{S}$. The second is a bound on the *excess ℓ-risk*, that is the difference between the risk of ERM model and the smallest risk achievable for models in $\mathcal{H}$, the second being unavoidable even for known $\mathcal{D}$. Those bounds have a typical shape often encountered in generalization results. Two terms contribute to their significance. The Rademacher complexity quantifies the capacity of the model space $\mathcal{H}$ of fitting all possible label assignments. While large capacity means better approximation and therefore lower empirical risk, this is not necessarily good for generalization risk on $\mathcal{D}$. In fact, low Rademacher complexity tightens the bounds. This is along the line of the classic results on the VC-dimension, which is indeed an upper bound of the Rademacher complexity. The remaining term is a statistical penalty due to requiring uniform convergence. The bound holds true for any $\mathcal{S}$, and in particular for samples not being much representative of $\mathcal{D}$, that is, drawn from areas with low probability mass on the support of $\mathcal{D}$. This is the role of δ in the probability inequality, while large m weights down the possibility of those bad choices of $\mathcal{S}$.

Overall the bounds decrease as $O(1/\sqrt{m})$. To see this, actually we need to compute the Rademacher complexity for a specific $\mathcal{H}$ (see Theorem 7 below). Additional conditions can guarantee a *fast rate* of convergence of $O(1/m)$, which is also known to

be the fastest rate achievable in learning [Vapnik, 1998]. Those improved guarantees will not be discussed in the Thesis.

Next, we specialize some of the previous results for Lipschitz loss functions and linear classifiers.

Lemma 6. *Let ℓ be a L-Lipschitz function. Then: $\mathcal{R}(\ell \circ \mathcal{H}) \leq L\mathcal{R}(\mathcal{H})$, where:*

$$\mathcal{R}(\mathcal{H}) \doteq \mathbb{E}_{S\sim\mathcal{D}}\mathbb{E}_{\sigma\sim\Sigma_m}\left[\frac{1}{m}\sup_{h\in\mathcal{H}}\sum_{i=1}^{m}\sigma_i\, h(x_i)\right] , \tag{2.9}$$

The result is also helpful to clarify what the Rademacher complexity computes: it measures the capacity of hypotheses in $\mathcal{H}$ by the supremum of correlation with random noise — the random variable σ. We will refer to both $\mathcal{R}(\mathcal{H})$ and $\mathcal{R}(\ell \circ \mathcal{H})$ (with dependency on ℓ) as Rademacher complexity. Second, we can give generalization bounds specialized for linear classifiers.

Theorem 7. *Let $\mathcal{X}$ be a vector space such that $\mathcal{X} = \{\boldsymbol{x} : \|\boldsymbol{x}\|_2 \leq X < \infty\}$. Let $\mathcal{H}$ be the space of bounded linear classifiers $\mathcal{H} = \{\boldsymbol{\theta} : \|\boldsymbol{\theta}\|_2 \leq H < \infty\}$. Assume a L-Lipschitz, bounded loss $|\ell(x)| \leq C, \forall x$. With probability at least $1-\delta$ over the choice of $\mathcal{S}$, simultaneously for all $h \in \mathcal{H}$, we have:*

$$R_{\mathcal{D},\ell}(h) - R_{\mathcal{S},\ell}(h) \leq \frac{2LXH}{\sqrt{m}} + C\sqrt{\frac{2}{m}\log\frac{1}{\delta}} . \tag{2.10}$$

Moreover, let the empirical risk minimizer be $\hat{h} = \operatorname{argmin}_{h\in\mathcal{H}} R_{\mathcal{S},\ell}(h)$. For any $\delta \in (0,1)$, with probability least $1-\delta$ over the choice of $\mathcal{S}$, we have:

$$R_{\mathcal{D},\ell}(\hat{h}) - \inf_{h\in\mathcal{H}} R_{\mathcal{D},\ell}(h) \leq \frac{4LXH}{\sqrt{m}} + 2C\sqrt{\frac{2}{m}\log\frac{1}{\delta}} . \tag{2.11}$$

Proof in 2.6.2. Finally, we have mentioned that Rademacher complexity is a data dependent measure. Although, previous results account for its population mean, not the empirical complexity. Yet it is simple to show that the empirical Rademacher complexity converges rapidly to its population mean by applying McDiarmid's inequality.

Lemma 8. *Assume a bounded loss,* i.e. *$|\ell(x)| \leq C, \forall x$. Let $\delta > 0$. We have with probability $1-\delta$:*

$$|\mathcal{R}(\ell \circ \mathcal{H}) - \mathcal{R}(\ell \circ \mathcal{H} \circ \mathcal{S})| \leq C\sqrt{\frac{2}{m}\log\frac{2}{\delta}} . \tag{2.12}$$

It follows that we can express all the above generalization bounds in terms of empirical Rademacher complexity at the price of worsening the statistical penalty.

2.3.2 Calibrated losses

The generalization bounds above express uniform convergence guarantees of the ERM with respect to the surrogate loss function chosen for learning. Ultimately, we wish to obtain similar bounds for the error, which is the measure of goodness of the hypothesis. To that objective, we define a subclass of margin losses called *calibrated* [Bartlett et al., 2006], which transforms the previous results in bounds for the generalization error. We introduce them by a sufficient and necessary condition for the case of convex losses.

Definition 9. *Let ℓ be a non-negative convex loss. ℓ is* (classification) calibrated *if and only if ℓ is differentiable in 0 and $\ell'(0) < 0$.*

Theorem 10. *Let ℓ be calibrated. Then, there exists a convex, non-decreasing, invertible function $\psi_\ell : [0,1] \to \mathbb{R}_+$, with $\psi_\ell(0) = 0$, such that for any $h \in \mathcal{H}$:*

$$\psi_\ell \left(R_{\mathcal{D},0/1}(h) - \inf_h R_{\mathcal{D},0/1}(h) \right) \leq R_{\mathcal{D},\ell}(h) - \inf_h R_{\mathcal{D},\ell}(h) \ . \tag{2.13}$$

It follows that any time we can prove a bound for the excess ℓ-risk, we immediately translate it to the error via ψ_ℓ^{-1}. For connections between proper and calibrated losses see Reid and Williamson [2010].

2.4 Maximum likelihood, exponential family and sufficient statistics

So far we have seen learning within ERM. An alternative set up for learning problems is the notion of maximum likelihood estimation (MLE), a classic procedure for fitting models in Statistics. We start by assuming that our model belongs to a certain parametric probability distribution. Learning is then accomplished by fitting those parameters to the data by maximizing its likelihood. In particular, we can learn a binary classifier in the conditional (binary) exponential family parameterized by a vector $\boldsymbol{\theta} \in \mathbb{R}^d$:

$$p_{\boldsymbol{\theta}}(y|\boldsymbol{x}) = \exp\left(\langle \boldsymbol{\theta}, y\boldsymbol{x} \rangle - \log \sum_{y \in \mathcal{Y}} \exp \langle \boldsymbol{\theta}, y\boldsymbol{x} \rangle \right) \tag{2.14}$$

Here, y is the only random variable. The two terms in the exponent are the log-partition function, which normalizes the distribution such that it can sum to one, and the inner product between parameters and the sufficient statistic $y\boldsymbol{x}$. Under the *i.i.d.* assumption, we can maximize the likelihood of the data being generated by the

exponential family as:

$$\underset{\boldsymbol{\theta}}{\text{argmax}} \quad \prod_{i=1}^{m} \exp\left(\langle \boldsymbol{\theta}, y_i \boldsymbol{x}_i \rangle - \log \sum_{y \in \mathcal{Y}} \exp\langle \boldsymbol{\theta}, y \boldsymbol{x}_i \rangle\right) = \tag{2.15}$$

$$\underset{\boldsymbol{\theta}}{\text{argmax}} \quad \exp\left(\sum_{i=1}^{m} \langle \boldsymbol{\theta}, y_i \boldsymbol{x}_i \rangle - \sum_{i=1}^{m} \log \sum_{y \in \mathcal{Y}} \exp\langle \boldsymbol{\theta}, y \boldsymbol{x}_i \rangle\right) \ . \tag{2.16}$$

By taking log and a negation, the objective of 2.16 becomes:

$$\sum_{i=1}^{m} \log \sum_{y \in \mathcal{Y}} \exp\langle \boldsymbol{\theta}, y \boldsymbol{x}_i \rangle - \sum_{i=1}^{m} \langle \boldsymbol{\theta}, y_i \boldsymbol{x}_i \rangle \tag{2.17}$$

$$= \sum_{i=1}^{m} \log \sum_{y \in \mathcal{Y}} \exp\langle \boldsymbol{\theta}, y \boldsymbol{x}_i \rangle - \sum_{i=1}^{m} \log \exp\langle \boldsymbol{\theta}, y_i \boldsymbol{x}_i \rangle \tag{2.18}$$

$$= \sum_{i=1}^{m} \log \frac{\exp\langle \boldsymbol{\theta}, \boldsymbol{x}_i \rangle + \exp\langle \boldsymbol{\theta}, -\boldsymbol{x}_i \rangle}{\exp\langle \boldsymbol{\theta}, y_i \boldsymbol{x}_i \rangle} \tag{2.19}$$

$$= \sum_{i=1}^{m} \log\left(1 + \exp\left(-2 y_i \langle \boldsymbol{\theta}, \boldsymbol{x}_i \rangle\right)\right) \ . \tag{2.20}$$

Step 2.20 is true since $y \in \{\pm 1\}$. Finally, by re-parameterizing $\boldsymbol{\theta}$ and normalizing, we obtain logistic loss. These steps prove how the two approaches for learning, ERM and MLE, are equivalent for conditional exponential family and logistic loss. Additionally, Equation 2.17 shows how the loss splits into a linear term aggregating the labels sufficient statistics and another, label free term. We consider this equivalence between conditional exponential family and logistic loss as common knowledge (in spite of being unaware of any published material). One of the key theoretical contribution of the Thesis is to demonstrate how this is not a coincidence: we will introduce a broad new family of losses for which this decomposition is always valid.

2.4.1 Sufficient statistics

Why did we call the quantities $y\boldsymbol{x}$ and $\sum_i y_i \boldsymbol{x}_i$ sufficient statistics? A statistic is any function g computed from a sample $\mathcal{S}$. Intuitively, a statistic $g(\mathcal{S})$ is sufficient (for the parameter $\boldsymbol{\theta}$) when it aggregates all information of $\mathcal{S}$, such that no better model could be learned from $\mathcal{S}$ in place of g. We are mostly interested in sufficiency with regard to y, thus we will only look into this case. In formulas, a statistic g is sufficient for $\boldsymbol{\theta}$ with respect to any random variable Y with outcome y if:

$$\mathbb{P}(\boldsymbol{\theta}|y) = \mathbb{P}(\boldsymbol{\theta}|g(y)), \forall y \sim Y \ . \tag{2.21}$$

We provide an equivalent definition that, although somewhat less intuitive, is useful for conceiving some formal results in the Thesis.

Definition 11 (Sufficient statistic)**.** *Let Y a random variable. $g(Y)$ is a sufficient statistic*

for θ with regard to Y when for each pair of outcomes y, y' of Y we have:

$$\frac{\mathbb{P}(\theta|g(y))}{\mathbb{P}(\theta|g(y'))} \text{ does not depend on } y \Leftrightarrow g(y) = g(y') \ . \tag{2.22}$$

An additional equivalent characterization for sufficient statistics is provided by the classic Fisher-Neyman Theorem [Lehmann and Casella, 1998].

Theorem 12 (Fisher-Neyman)**.** *Let $p_{\boldsymbol{\theta}}(y)$ be the probability density function of y. Then g is sufficient for $\boldsymbol{\theta}$ if and only if two non-negative functions $p^{(i)}$ and $p^{(ii)}$ can be found such that:*

$$p_{\boldsymbol{\theta}}(y) = p^{(i)}_{\boldsymbol{\theta}}(g(y)) \cdot p^{(ii)}(y) \ . \tag{2.23}$$

In other words, the probability density factors in two functions, such that $\boldsymbol{\theta}$ interacts with the y only through g. This can be used to show that $y\boldsymbol{x}$ is indeed sufficient for $\boldsymbol{\theta}$ with regard to y in the case of the conditional exponential family (Equation 2.14). It holds that $p^{(ii)}(y|\boldsymbol{x}) = 1$, $g(y|\boldsymbol{x}) = y\boldsymbol{x}$ and $p^{(i)}_{\boldsymbol{\theta}}(\cdot|\boldsymbol{x}) = \exp(\langle\boldsymbol{\theta}, \cdot\rangle - \log\sum_{y\in\mathcal{Y}}\exp(\langle\boldsymbol{\theta}, y\boldsymbol{x}\rangle)$, since the value of y is not needed for computing $p^{(i)}_{\boldsymbol{\theta}}$.

2.5 Weakly supervised learning

This Section is halfway between an informal problem statement and a high level literature review for weakly supervised learning problems and relative frameworks for solutions. "Weakly supervised learning" is a non-standard yet widely used terminology to describe scenarios that sit somewhere in between of supervised and unsupervised learning; other literature refers to, for instance, indirect [Raghunathan et al., 2016] or distant supervision [Mintz et al., 2009; Surdeanu et al., 2012]. This class of learning problems relaxes one fundamental assumption of supervised learning: the learner does not have perfect labels, that is, there is no guarantee that each label is fully observable and that each observed label is free from mistakes. This informal definition is deliberately vague as it is meant to encompass a large diversity of learning settings. For example, labels may be missing as with *semi-supervision* [Chapelle et al., 2006] and *positive and unlabeled data* [du Plessis et al., 2015], *noisy* [Natarajan et al., 2013], aggregated as it happens in *multiple instance learning* [Dietterich et al., 1997] and *learning from label proportions* [Kuck and de Freitas, 2005], or given in a candidate set which include the only correct one, as in *partial labels* or *superset label learning* [Cour et al., 2011; Liu and Dietterich, 2014]. We also stress the fact that all these problems share the same objective of supervised learning, which is learning a classifier — or occasionally a regressor —, in contrast with "*fully unsupervised*" learning.

Those scenarios are rarely considered altogether; formal analysis and algorithmic solutions are usually proposed *ad-hoc*. In contrast, one of the goal of the Thesis is give an unified treatment. We will see how this level of abstraction is beneficial.

Formally, we model a weakly supervised problem by a corruption process that takes the original distribution $\mathcal{D}$ and produces a corrupted distribution $\widetilde{\mathcal{D}}$, from which we get a corrupted sample $\widetilde{\mathcal{S}}$:

$$\mathcal{D} \xrightarrow{corrupt} \widetilde{\mathcal{D}} \xrightarrow{sample} \widetilde{\mathcal{S}} \tag{2.24}$$

A fundamental assumption is that the marginal distribution of the features is untouched by corruption, that is, $\mathbb{P}_{\mathcal{D}}(x) = \mathbb{P}_{\widetilde{\mathcal{D}}}(x)$. This is all the formalism we need but we give more concrete examples of $\widetilde{\mathcal{S}}$ for some particular cases.

Example. In semi-supervised learning, the learner sees two subsamples, one fully labeled and the other unlabeled: $\widetilde{\mathcal{S}} = (\mathcal{S}_L, \mathcal{S}_U)$, with $\mathcal{S}_L = \{(\boldsymbol{x}_i, y_i), i \in [m_L]\}$ and $\mathcal{S}_U = \{\boldsymbol{x}_i, i \in [m_U]\}$. In positive and unlabeled learning, the scenario is similar but the labeled examples are always positive $\mathcal{S}_L = \{(\boldsymbol{x}_i, 1), i \in [m_L]\}$.

Example. When labels are noisy, it is usually assumed that the learner sees them all, although without guarantees of their truthfulness. Hence, $\widetilde{\mathcal{S}} = \{(\boldsymbol{x}_i, \tilde{y}_i), i \in [m]\}$, where $\tilde{y}$ is drawn from some corrupted label distribution $p(\tilde{y})$.

Example. It is less obvious how to formalize supervision given at aggregate level; a possible choice is the following. Let us consider the original learning sample given by two ordered sets, one containing observations and the other one the relative labels: $\mathcal{S} = (\mathcal{S}_X, \mathcal{S}_Y)$; observations are mapped to respective labels by indices. The corrupted sample leaves features vectors untouched us usual, yet it partitions them into n *bags* as $\mathcal{S}_X = \bigcup_{j \in [n]} \mathcal{S}_j$ with $\forall j, j' \ \mathcal{S}_j \cap \mathcal{S}_{j'} = \emptyset$. A certain type of supervision is defined at the level of bags by $\boldsymbol{\pi} \in \mathbb{R}^n$. The resulting corrupted sample is then $\widetilde{\mathcal{S}} = (\{\mathcal{S}_j\}_{j \in [n]}, \{\pi_j\}_{j \in [n]})$. In learning from label proportions, $\pi_j \in [0, 1]$ represents the fraction of positive labels for each bag j. In multiple instance labels, the supervision is even weaker and $\pi_j \in \{0, 1\}$ is 1 if at least one label in bag j is positive[1].

The difference between those learning settings is not crisp, for two reasons. On one hand, it is plausible that they can be combined by mixing the types of corruption. For instance, we can easily imagine that some labels are missing but those known are noisy; another common scenario is semi-supervised learning with additional prior knowledge on large unlabeled data, *e.g.* as in Bilenko et al. [2004]. On the other hand, some learning settings can be thought to be more general than others, without a total ordering in a hierarchy. For instance, the noisy label distribution $p(\tilde{y})$ may be modeling the event of "label suppression" so as to encompass missing labels as well [Menon et al., 2015]; multiple instance learning can be thought as a particular, less informative supervision with respecting to label proportions [Kuck and de Freitas, 2005]; at the same time, algorithms for semi-supervised learning have been shown to be apt for multiple instance learning [Zhou and Xu, 2007].

[1] Other encodings of supervision of MIL have been formalized. Also, MIL is sometimes intended as the problem of learning classifiers at bag level; we do not consider this scenario. See Foulds and Frank [2010]; Hernandez-Gonzalez et al. [2016] for more details.

2.5.1 Empirical risk minimization under weak supervision

What is the meaning of computing the empirical risk on the corrupted sample, $R_{\widetilde{\mathcal{S}},\ell}(h)$? By relaxing the assumption of "full supervision" the framework of ERM loses its sense. Depending on the nature of weak supervision, this quantity may be computable either only partially (semi-supervision) or completely but with little meaning (noisy labels); worse, when supervision is given at multi-instance level, there exists no obvious way to estimate the risk. Mathematically, the problem of searching for risk minimizer becomes ill-posed. We discuss a non-exhaustive overview of families of methods proposed in the past.

Case (i): some observations have individual labels, but some others do not. We adopt the definition of $\widetilde{\mathcal{S}} = \mathcal{S}_L \cup \mathcal{S}_U$ of semi-supervised learning for simplicity. We can formulate an optimization problem such as:

$$\operatorname*{argmin}_{h \in \mathcal{H}} R_{\mathcal{S}_L,\ell}(h) + \lambda \cdot \textsc{reg}(\widetilde{\mathcal{S}}, h) \tag{2.25}$$

The regularizer is intended to exploit the information of the additional unlabeled features vectors to bias the search in the model space. The idea is that the feature geometry, regardless of unknown labels, should be informative about the class distribution. Formally, one of three celebrated assumptions is required for this to be true: the smoothness, the cluster, or the manifold assumption [Chapelle et al., 2006]; the *causal direction* of data generation has also been taken into account for justifying the success of semi-supervised learning [Janzing et al., 2012]. Examples of this framework for learning with missing labels are: transductive SVM [Joachims, 1999], information regularization [Szummer and Jaakkola, 2002], label propagation as a regularizer [Zhu and Ghahramani, 2002; Bengio et al., 2006], entropy regularization [Grandvalet and Bengio, 2004], manifold regularization [Belkin et al., 2006], ladder network [Rasmus et al., 2015] and graph embedding [Weston et al., 2012; Yang et al., 2016].

Case (ii): noisy labels. Design a corrected loss $\tilde{\ell}$ satisfying a certain property of robustness with respect to the noise and minimize it instead of the original ℓ:

$$\operatorname*{argmin}_{h \in \mathcal{H}} R_{\widetilde{\mathcal{S}},\tilde{\ell}}(h) \tag{2.26}$$

Often, the robust loss is either non-convex [Masnadi-Shirazi et al., 2010; Ding and Vishwanathan, 2010; Ghosh et al., 2015] or requires certain knowledge of the label corruption $p(\tilde{y}|y)$ [Stempfel and Ralaivola, 2009; Natarajan et al., 2013], with the noticeable exception of the linear "unhinged" loss [van Rooyen et al., 2015].

Case (iii): aggregate supervision. An alternative to ERM in this case is non trivial, as demonstrated by the many present in literature. An intuitive approach in the event of lack of observation-level labels is to augment the search space to $\mathcal{Y}$, namely, by

modeling the unknown labels as latent variables. Although ultimately the learner output is a model, the estimation of the labels become a fundamental intermediate step for learning. This framework closely resembles and it is often implemented as the Expectation Maximization (EM) algorithm, sharing the well known drawbacks. Weak supervision helps the search in the model space by formulating constraints (hard of soft) to be satisfied by the model:

$$\underset{h\in\mathcal{H},y\in\mathcal{Y}}{\operatorname{argmin}} R_{(\mathcal{S}_X,\mathcal{S}_Y),\ell}(h) + \lambda \cdot \textsc{constr}(\mathcal{S}_X, \mathcal{S}_Y, h) \tag{2.27}$$

Examples of this approach are Yu et al. [2013] for LLP, Andrews et al. [2002] for MIL. This framework easily extends to the case of constraints on (functions of) features vectors, which may be loosely interpreted as weak supervision as well. In this line of work, particularly relevant for applications in NPL, constraints are enforced on particular features, such as the word "amazing" is a strong indicator of a highly rated movie review; a sample of those methods are constraint driven learning Chang et al. [2007], generalized expectation constraints Mann and McCallum [2008], posterior constraints Ganchev et al. [2010], and label regularization [Mann and McCallum, 2010; Ardehaly and Culotta, 2015]. A unifying framework of this kind of EM-based inference with constraints is elaborated by Samdani et al. [2012].

Case (iv): Prior work also attempts an agnostic treatment of weak supervision for ERM, as we envision in the Thesis. A broad, comparative formalization of weakly supervised problems is presented by Garcıa-Garcıa and Williamson [2011]. Joulin and Bach [2012] attack a version of the generic Problem 2.27 by a convex relaxation and a dedicated optimization algorithm based on semidefinite programming. Li et al. [2013] formulate a tighter and more scalable convex relaxation of a generic weakly supervised max-margin problem. Zantedeschi et al. [2016] propose the β-risk, a surrogate risk augmented with a matrix parameter β which captures the reliability of each instance label; models, labels and β are estimated iteratively. A last important mention is the work of Raghunathan et al. [2016] that, due to the strong similarity with our framework, will be discussed in detail in Section 3.7 of the next Chapter.

Several other ideas are popular in literature. These are for instance self-training and co-training [Blum and Mitchell, 1998; Nigam and Ghani, 2000], multi-view learning [de Sa, 2005] and graph cuts [Blum and Chawla, 2001] for semi-supervised learning, the framework of collective graphical models [Sheldon and Dietterich, 2011; Bernstein and Sheldon, 2016] and, more generally, any Bayesian treatment for incorporating weak supervision into a generative model [Seeger, 2000; Lawrence and Jordan, 2004; Kuck and de Freitas, 2005; Liang et al., 2009; Kingma et al., 2014]. We do not discuss those topics as they do not directly implement a solution for ERM – although some can be shown to link to it. Moreover, readers who are expert in those areas might foresee a conceptual link to multi-armed bandit [Auer et al., 2002] or Reinforcement Learning [Sutton and Barto, 1998]. Admittedly, the need to model the learning process in those settings is the result of the mismatch between observation

(or action) and label (reward). As we do not study in any case neither sequential data nor interactive scenarios, or in other words, we always assume *i.i.d.* samples, those learning setting are as well outside the scope of the Thesis.

2.6 Appendix: proofs

2.6.1 Proof of Theorem 5

The proof follows Shalev-Shwartz and Ben-David [2014]. Let us first recall McDiarmid's inequality [McDiarmid, 1998].

Lemma 13. *Let $\mathcal{Z}$ be a set and $f : \mathcal{Z}^m \to \mathbb{R}$ be a function of m variables such that for some $c > 0$, for all $i' \in [m]$ and for all $z_1, \ldots, z_m, z_{i'} \in \mathcal{Z}$ we have:*

$$|f(z_1, \ldots, z_i, \ldots, z_m) - f(z_1, \ldots, z_{i'}, \ldots, z_m)| \leq c \ . \tag{2.28}$$

Let now be $Z_1, \ldots, Z_m$ m independent random variables taking values in $\mathcal{Z}$. Then, with probability at least $1 - \delta$ we have:

$$f(Z_1, \ldots, Z_m) - \mathbb{E}[f(Z_1, \ldots, Z_m)] \leq c\sqrt{\frac{m}{2}\log\frac{1}{\delta}} \ . \tag{2.29}$$

We also make use of an additional Lemma that connects Rademacher complexity with the maximum deviation of empirical risk from the respective risk.

Lemma 14. *Let $\phi(\mathcal{S}) = \sup_{h\in\mathcal{H}} R_{\mathcal{D},\ell}(h) - R_{\mathcal{S},\ell}(h)$. Then:*

$$\mathbb{E}_{\mathcal{D}}\, \phi(\mathcal{S}) \leq 2\mathcal{R}(\ell \circ \mathcal{H}) \ . \tag{2.30}$$

For any bounded loss function $|\ell(c)| \leq C$, the function ϕ satisfies McDiarmid's inequality with $c = 2C/m$. Therefore:

$$\phi(\mathcal{S}) \leq \mathbb{E}_{\mathcal{D}}\, \phi(\mathcal{S}) + C\sqrt{\frac{2}{m}\log\frac{1}{\delta}} \leq 2\mathcal{R}(\ell \circ \mathcal{H}) + C\sqrt{\frac{2}{m}\log\frac{1}{\delta}} \ , \tag{2.31}$$

with probability at least $1 - \delta$. Clearly, for all $h \in \mathcal{H}$, $R_{\mathcal{D},\ell}(h) - R_{\mathcal{S},\ell}(h) \leq \phi(\mathcal{S})$, which proves the first statement. For the second statement, for any $h^\star \in \mathcal{H}$, we write:

$$\begin{aligned}
R_{\mathcal{D},\ell}(\hat{h}) - R_{\mathcal{D},\ell}(h^\star) &= R_{\mathcal{D},\ell}(\hat{h}) + \Big(-R_{\mathcal{S},\ell}(\hat{h}) + R_{\mathcal{S},\ell}(\hat{h})\Big) \\
&\quad + \Big(-R_{\mathcal{S},\ell}(h^\star) + R_{\mathcal{S},\ell}(h^\star)\Big) - R_{\mathcal{D},\ell}(h^\star) && (2.32)\\
&\leq \Big(R_{\mathcal{D},\ell}(\hat{h}) - R_{\mathcal{S},\ell}(\hat{h})\Big) + \Big(R_{\mathcal{S},\ell}(h^\star) - R_{\mathcal{D},\ell}(h^\star)\Big) && (2.33)\\
&\leq 2\phi(\mathcal{S}) && (2.34)
\end{aligned}$$

Step 2.33 holds because $\hat{h}$ is the empirical risk minimizer. Finally, we apply again inequality 2.31 and obtain the desired statement.

2.6.2 Proof of Theorem 7

We need to compute an upper bound of the Rademacher complexity for linear classifiers.

Lemma 15. *Let $\mathcal{X}$ be a vector space such that $\mathcal{X} = \{\boldsymbol{x} : \|\boldsymbol{x}\|_2 \leq X < \infty\}$. Let $\mathcal{H}$ be the space of bounded linear classifiers $\mathcal{H} = \{\boldsymbol{\theta} : \|\boldsymbol{\theta}\|_2 \leq H < \infty\}$. Then:*

$$\mathcal{R}(\mathcal{H}) \leq \frac{XH}{\sqrt{m}} \ . \tag{2.35}$$

Proof

$$m\ \mathcal{R}(\mathcal{H}) = m\ \mathbb{E}_{\mathcal{D},\sigma} \sup_{\boldsymbol{\theta}\in\mathcal{H}} \frac{1}{m} \sum_{i=1}^{m} \sigma_i \langle \boldsymbol{\theta}, \boldsymbol{x}_i \rangle \tag{2.36}$$

$$= \mathbb{E}_{\mathcal{D},\sigma} \sup_{\boldsymbol{\theta}\in\mathcal{H}} \sum_{i=1}^{m} \sigma_i \langle \boldsymbol{\theta}, \boldsymbol{x}_i \rangle \tag{2.37}$$

$$= \mathbb{E}_{\mathcal{D},\sigma} \sup_{\boldsymbol{\theta}:\|\boldsymbol{\theta}\|_2\leq B} \sum_{i=1}^{m} \sigma_i \langle \boldsymbol{\theta}, \boldsymbol{x}_i \rangle \tag{2.38}$$

$$= \mathbb{E}_{\mathcal{D},\sigma} \sup_{\boldsymbol{\theta}:\|\boldsymbol{\theta}\|_2\leq B} \langle \boldsymbol{\theta}, \sum_{i=1}^{m} \sigma_i \boldsymbol{x}_i \rangle \tag{2.39}$$

$$\leq H\ \mathbb{E}_{\mathcal{D},\sigma} \left[\left\| \sum_{i=1}^{m} \sigma_i \boldsymbol{x}_i \right\|_2 \right] \ . \tag{2.40}$$

The last Step is due to Cauchy-Schwartz inequality. Next, by Jensen's inequality it holds:

$$\mathbb{E}_\sigma \left[\left\| \sum_{i=1}^{m} \sigma_i \boldsymbol{x}_i \right\|_2 \right] = \mathbb{E}_\sigma \left[\left(\left\| \sum_{i=1}^{m} \sigma_i \boldsymbol{x}_i \right\|_2^2 \right)^{1/2} \right] \leq \left(\mathbb{E}_\sigma \left[\left\| \sum_{i=1}^{m} \sigma_i \boldsymbol{x}_i \right\|_2^2 \right] \right)^{1/2} \ . \tag{2.41}$$

Finally, since the variables σ_i are independent, we have:

$$\mathbb{E}_\sigma \left[\left\| \sum_{i=1}^{m} \sigma_i \boldsymbol{x}_i \right\|_2^2 \right] = \mathbb{E}_\sigma \left[\sum_{i,j} \sigma_i \sigma_j \langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle \right] \tag{2.42}$$

$$= \sum_{i\neq j} \langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle \mathbb{E}_\sigma \left[\sigma_i \sigma_j \right] + \sum_{i=1}^{m} \langle \boldsymbol{x}_i, \boldsymbol{x}_i \rangle \mathbb{E}_\sigma \left[\sigma_i^2 \right] \tag{2.43}$$

$$= \sum_{i=1}^{m} \|\boldsymbol{x}_i\|_2^2 \tag{2.44}$$

$$\leq mX^2 \ . \tag{2.45}$$

Altogether we have:

$$m\,\mathcal{R}(\mathcal{H}) \leq H\mathbb{E}_{\mathcal{D}}\left(mX^2\right)^{1/2} = \sqrt{m}XH\ , \tag{2.46}$$

which proves the Lemma. ■

To get Theorem 7, we combine the Theorem 5 and Lemma 6 with the last result.

Chapter 3

Weakly supervised learning and loss factorization

We introduce a new family of loss functions called linear-odd for the formal study of weak supervision. The central result is the statement of the Loss Factorization Theorem that isolates the effect of supervision into a label sufficient statistic. As a result to this finding, we can formulate a generic two-step approach for solving weakly supervised problem, by calling standard supervised algorithms based on gradient descent. This Chapter is the most abstract and can be thought as a toolbox of formal results and meta-algorithms to be specialized for particular instances of weak supervision. The Chapter, if not the Thesis in a whole, is inspired by the popular quote:

> *"One should solve the problem directly and never solve a more general problem as an intermediate step"* [Vapnik, 1998].

3.1 Linear-odd losses and Loss Factorization

In Chapter 2 we shown that MLE of the exponential family is equivalent to ERM with logistic loss. In doing so, we proven that logistic loss decomposes into two terms, one of which is linear. We now give a name to that statistical object and show some intriguing properties connecting the idea of sufficiency from Statistics to Learning Theory.

Definition 16. *The (empirical)* mean operator *of a learning sample $\mathcal{S}$ is $\boldsymbol{\mu}_{\mathcal{S}} \doteq \mathbb{E}_{\mathcal{S}}[y\boldsymbol{x}]$.*

In the following, we drop the dependency of $\mathcal{S}$ when clear by the context. The name *mean operator*, or mean map, is borrowed from the theory of Hilbert space embedding [Quadrianto et al., 2009]. In this Thesis, $\boldsymbol{\mu}$ will play the role of sufficient statistic for labels with regard to a set of losses. The next is motivated by Definition 11 by taking log-odd ratio.

Definition 17. *A function $T(\mathcal{S})$ is said to be a sufficient statistic for a variable y with regard to a set of losses $\mathcal{L}$ and a hypothesis space $\mathcal{H}$ when for any $\ell \in \mathcal{L}$, any $h \in \mathcal{H}$ and any two samples $\mathcal{S}$ and $\mathcal{S}'$ the empirical ℓ-risk is such that*[1]*:*

$$R_{\mathcal{S},\ell}(h) - R_{\mathcal{S}',\ell}(h) \text{ does not depend on } y \Leftrightarrow T(\mathcal{S}) = T(\mathcal{S}') \ . \tag{3.1}$$

We will often call any such quantity a *label sufficient statistic*. We now state and prove one of the main theoretical contributions of the Thesis, the Loss Factorization Theorem – by reminiscence of Fisher-Neyman Factorization Theorem 12. As a consequence, we establish sufficiency of mean operators for a large set of losses.

Theorem 18 (Factorization)**.** *Let $\mathcal{H}$ be the space of linear hypotheses. Assume that a loss ℓ is such that $\ell_o(x) \doteq (\ell(x) - \ell(-x))/2$ is linear. Then, for any sample $\mathcal{S}$ and hypothesis $h \in \mathcal{H}$ the empirical ℓ-risk can be written as:*

$$R_{\mathcal{S},\ell}(h) = \frac{1}{2} R_{\mathcal{S}_{2x},\ell}(h) + \ell_o(h(\boldsymbol{\mu})) \ , \tag{3.2}$$

where $\mathcal{S}_{2x} \doteq \{(\boldsymbol{x}_i, \sigma), i \in [m], \forall \sigma \in \mathcal{Y}\}$.

Proof We detail the proof as it clearly highlights the relevant elements allowing losses to factor. A key ingredient is an elementary fact from calculus: *any function writes (uniquely) as the sum of an even and an odd function*. Therefore, we can write a loss function $\ell(x)$ as:

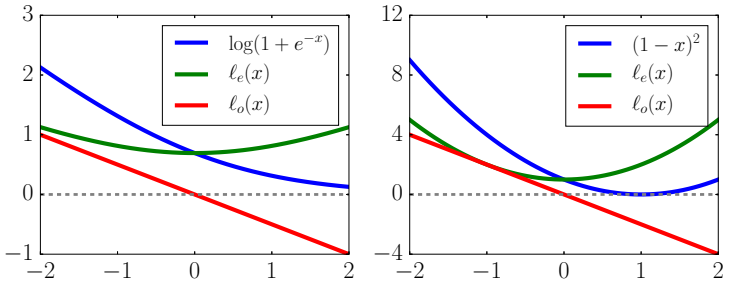
$$\begin{aligned} \ell(x) &= \frac{1}{2}\left[\ell(x) + \ell(-x) + \ell(x) - \ell(-x)\right] && (3.3)\\ &= \ell_e(x) + \ell_o(x) \ , && (3.4) \end{aligned}$$

where $\ell_e(x) \doteq \frac{1}{2}\left[\ell(x) + \ell(-x)\right]$ and $\ell_o(x) \doteq \frac{1}{2}\left[\ell(x) - \ell(-x)\right]$ are respectively an even and an odd function (see also Figure 3.1). The empirical risk is:

$$\begin{aligned} R_{\mathcal{S},\ell}(h) &= \mathbb{E}_{\mathcal{S}}[\ell(yh(\boldsymbol{x}))] && (3.5)\\ &= \frac{1}{2}\mathbb{E}_{\mathcal{S}}\Big[\ell(yh(\boldsymbol{x})) + \ell(-yh(\boldsymbol{x})) + \ell(yh(\boldsymbol{x})) - \ell(-yh(\boldsymbol{x}))\Big] && (3.6)\\ &= \frac{1}{2}\mathbb{E}_{\mathcal{S}}\left[\sum\nolimits_{\sigma\in\mathcal{Y}} \ell(\sigma h(\boldsymbol{x}))\right] + \mathbb{E}_{\mathcal{S}}\Big[\ell_o(yh(\boldsymbol{x}))\Big] && (3.7)\\ &= \frac{1}{2}\mathbb{E}_{\mathcal{S}_{2x}}\Big[\ell(\sigma h(\boldsymbol{x}))\Big] + \mathbb{E}_{\mathcal{S}}\Big[\ell_o(h(y\boldsymbol{x}))\Big] \ . && (3.8) \end{aligned}$$

Step 3.8 is due to the definition of $\mathcal{S}_{2x}$ and linearity of h. Finally, exploiting the properties of the loss assumed by the Theorem and the linearity of expectation, we

[1] Notice that the focus in the Definition is on the variable y (confront with Definition 11). This is convenient so as to stress that sufficiency is about a certain variable, usually the label, while h will rarely play a role in the Definition. Yet, the model h is the object that sufficiency is relevant *for*, since it is the goal of learning.



(a) logistic loss (b) square loss

Figure 3.1

have:

$$\mathbb{E}_{\mathcal{S}}\Big[\ell_o(h(y\boldsymbol{x}))\Big] = \ell_o(h(\boldsymbol{\mu}))\ , \tag{3.9}$$

which concludes the proof. ■

Factorization splits ℓ-risk in two parts. A first term is the ℓ-risk computed *on the same loss* on the "doubled sample" $\mathcal{S}_{2x}$ that contains each observation twice, labeled with opposite signs, hence it does not require any label knowledge. A second term is a function ℓ_o of h applied to the mean operator $\boldsymbol{\mu}$, which aggregates all sample labels. Also observe that ℓ_o is by construction an odd function, *i.e.* symmetric with respect to the origin. We call the losses satisfying the Theorem *linear-odd*.

Definition 19 (Linear-odd loss)**.** *A loss ℓ is a-*linear-odd *(a-*LOL*) when, for any $a \in \mathbb{R}$:*

$$\ell_o(x) = (\ell(x) - \ell(-x))/2 = ax\ . \tag{3.10}$$

Notice how this does not exclude losses that are non-smooth, non-convex, or non-proper. From now on, in this Chapter we consider $\mathcal{H}$ as the space linear hypotheses $h(\cdot) = \langle \boldsymbol{\theta}, \cdot \rangle$. As a consequence of Theorem 18, $\boldsymbol{\mu}$ is sufficient for all labels.

Corollary 20. *The mean operator $\boldsymbol{\mu}$ is a sufficient statistic for the label y with regard to* LOLS *and the space of linear classifiers $\mathcal{H}$.*

Proof in 3.5.1. The practical consequence of this Corollary is at the core of the applications in the Thesis: the single vector $\boldsymbol{\mu} \in \mathbb{R}^d$ summarizes all information concerning the linear relationship between y and $\boldsymbol{x}$ for losses that are LOL. But before dealing with the theoretical and practical consequences, we discuss the natural question at this point: how restrictive is the linear-odd condition?

3.1.1 The extent of linear-odd losses

Many commonly used losses are linear-odd. We list several examples in Table 3.1 and discuss some in the following.

	loss	even function ℓ_e	odd function l_o
generic	$\ell(x)$	$\frac{1}{2}(\ell(x)+\ell(-x))$	$\frac{1}{2}(\ell(x)-\ell(-x))$
LOL	$\ell(x)$	$\frac{1}{2}(\ell(x)+\ell(-x))$	ax
ρ-loss	$\rho\lvert x\rvert-\rho x+1$	$\rho\lvert x\rvert+1$	$-\rho x\ (\rho\geq 0)$
unhinged	$1-x$	1	$-x$
perceptron	$\max(0,-x)$	$x\ \mathrm{sign}(x)$	$-x$
double-hinge	$\max(-x,1/2\max(0,1-x))$	†	$-x$
SPL	$a_l+l^\star(-x)/b_l$	$a_l+\frac{1}{2b_l}(l^\star(x)+l^\star(-x))$	$-x/(2b_l)$
logistic	$\log(1+e^{-x})$	$\frac{1}{2}\log(2+e^x+e^{-x})$	$-x/2$
square	$(1-x)^2$	$1+x^2$	$-2x$
Matsushita	$\sqrt{1+x^2}-x$	$\sqrt{1+x^2}$	$-x$

Table 3.1: Factorization of losses. † for reason of space, the even part of double-hinge is written here as: $\max(-x,1/2\max(0,1-x))+\max(x,1/2\max(0,1+x))$.

Example. For logistic loss it holds that (Figure 3.1a):

$$\ell_o(x)=\frac{1}{2}\log\frac{1+e^{-x}}{1+e^{x}} \tag{3.11}$$

$$=\frac{1}{2}\log\frac{e^{-\frac{x}{2}}(e^{\frac{x}{2}}+e^{-\frac{x}{2}})}{e^{\frac{x}{2}}(e^{-\frac{x}{2}}+e^{\frac{x}{2}})} \tag{3.12}$$

$$=-\frac{x}{2}\ . \tag{3.13}$$

Example. Unhinged loss $\ell(x)=1-x$ of van Rooyen et al. [2015], which is trivially linear-odd. Instead, the standard hinge loss $\ell(x)=[1-x]_+$ does not factor in a linear term.

Example. Double-hinge and perceptron losses are proven to be linear-odd in du Plessis et al. [2015]. See also Appendix 3.6.5.

Example. The class of *symmetric proper losses* (SPLS) [Nock and Nielsen, 2009], *e.g.* logistic, square and Matsushita losses, satisfies the linear-odd condition. Let ϕ be permissible generator, *i.e.* ϕ is strictly convex, differentiable and symmetric with respect to $1/2$ and with $\mathrm{dom}(\phi)\supseteq[0,1]$. SPLS are defined as $\ell(x)=a_\phi+\phi^\star(-x)/b_\phi$, where $\phi^\star$ is the convex conjugate of ϕ. Then, since $\phi^\star(-x)=\phi^\star(x)-x$, we have:

$$\ell_o(x)=\frac{1}{2}\left(a_\phi+\frac{\phi^\star(-x)}{b_\phi}-a_\phi-\frac{\phi^\star(x)}{b_\phi}\right) \tag{3.14}$$

$$=\frac{1}{2b_\phi}\left(\phi^\star(x)-x-\phi^\star(x)\right) \tag{3.15}$$

$$=-\frac{x}{2b_\phi}\ . \tag{3.16}$$

For a broader treatment of SPL, see Subsection 4.2.1 on the next Chapter.

One may question whether SPL and LOL are equivalent. Table 3.1 gives an answer in the negative already: there are examples of non-smooth functions that are LOL. The next result provides a more complete picture by giving an exhaustive characterization of the family of linear-odd losses.

Lemma 21. *The exhaustive class of linear-odd losses is in 1-to-1 mapping with a proper subclass of even functions,* i.e. $\ell(x) = \ell_e(x) + ax$, *with* ℓ_e *any even function.*

Proof in 3.5.2. Interestingly, the proposition also let us engineer losses that always factor: choose any even function ℓ_e with desired properties — it need not be convex nor smooth. The loss is then $\ell(x) = \ell_e(x) + ax$, with a to be chosen. For example, let $\ell_e(x) = \rho|x| + 1$, with $\rho > 0$. $\ell(x) = \ell_e(x) - \rho x$ is a LOL; furthermore, ℓ upper bounds 01 loss and intercepts it in $\ell(0) = 1$. Non-convex ℓ can be constructed similarly. Yet, not all non-differentiable losses can be crafted this way since they are not LOL.

From the optimization viewpoint, we may want to keep properties of ℓ after factorization. The good news is that we are dealing with the same ℓ plus a linear term. Thus, if the property of interest is closed under summation with linear functions, then it will hold true. An example is convexity: if ℓ is LOL and convex, so is the factored loss. The same is true for differentiability.

In Appendix 3.6, we also elaborate on the relevance of the mean operator relating it to the covariance between $\boldsymbol{x}$ and y (Subsection 3.6.1), discuss the generality of Factorization beyond LOLS and linear models (Subsection 3.6.2) and state sufficient and necessary conditions to bound other known losses, *e.g.* hinge and Huber, by LOLS (Subsection 3.6.3).

3.2 Generalization bounds

A consequence of working with LOLS is on generalization bounds. We first derive an improved upper bound to the Rademacher complexity of $\mathcal{H}$ computed on $\mathcal{S}_{2x}$.

Lemma 22. *Suppose m even. Suppose* $\mathcal{X} = \{\boldsymbol{x} : \|\boldsymbol{x}\|_2 \leq X\}$ *be the observations space, and* $\mathcal{H} = \{\boldsymbol{\theta} : \|\boldsymbol{\theta}\|_2 \leq H\}$ *be the space of linear hypotheses. Let* $\Sigma^{2m} \doteq \times_{j\in[2m]}\mathcal{Y}$. *Then the empirical Rademacher complexity:*

$$\mathcal{R}(\mathcal{H} \circ \mathcal{S}_{2x}) \doteq \mathbb{E}_{\sigma\sim\Sigma^2}\left[\sup_{\boldsymbol{\theta}\in\mathcal{H}} \frac{1}{2m} \sum_{i\in[2m]} \sigma_i \langle \boldsymbol{\theta}, \boldsymbol{x}_i \rangle\right] \tag{3.17}$$

of $\mathcal{H}$ *on* $\mathcal{S}_{2x}$ *satisfies:*

$$\mathcal{R}(\mathcal{H} \circ \mathcal{S}_{2x}) \quad \leq \quad v \cdot \frac{XH}{\sqrt{2m}}\,, \tag{3.18}$$

with $v \doteq \frac{1}{2} + \frac{1}{2}\sqrt{\frac{1}{2} - \frac{1}{m}}$.

Proof in 3.5.3. Notice that this holds for the Rademacher complexity as well, since sampling different learning samples from D does not change the bound. A vanilla

calculation of the complexity of a double sized sample by Lemma 15 would give a coefficient of $XH/\sqrt{2m}$. Here we multiply it by $v < (\sqrt{2}+1)/(2\sqrt{2}) \approx 0.85$. This is relevant when combined into the next Theorem. The result sheds new light on excess ℓ-risk bounds on Rademacher complexity with linear hypotheses.

Theorem 23. *Assume ℓ is a-LOL and L-Lipschitz. Suppose $\mathbb{R}^d \supseteq \mathcal{X} = \{\boldsymbol{x} : \|\boldsymbol{x}\|_2 \leq X < \infty\}$ and $\mathcal{H} = \{\boldsymbol{\theta} : \|\boldsymbol{\theta}\|_2 \leq H < \infty\}$. Let $c(X,H) \doteq \max_{y\in\mathcal{Y}} \ell(yXH)$ and $\hat{\boldsymbol{\theta}} \doteq \operatorname{argmin}_{\boldsymbol{\theta}\in\mathcal{H}} R_{\mathcal{S},\ell}(\boldsymbol{\theta})$. Then for any $\delta > 0$, with probability at least $1-\delta$:*

$$R_{\mathcal{D},\ell}(\hat{\boldsymbol{\theta}}) - R_{\mathcal{D},\ell}(\boldsymbol{\theta}^\star) \leq \frac{\sqrt{2}+1}{2} \cdot \frac{XHL}{\sqrt{m}} + c(X,H) \cdot \sqrt{\frac{1}{m}\log\left(\frac{1}{\delta}\right)} + 2|a|H \cdot \|\boldsymbol{\mu}_{\mathcal{D}} - \boldsymbol{\mu}_{\mathcal{S}}\|_2 \ , \tag{3.19}$$

or more explicitly:

$$R_{\mathcal{D},\ell}(\hat{\boldsymbol{\theta}}) - R_{\mathcal{D},\ell}(\boldsymbol{\theta}^\star) \leq \frac{\sqrt{2}+1}{2} \cdot \frac{XHL}{\sqrt{m}} + c(X,H) \cdot \sqrt{\frac{1}{m}\log\left(\frac{2}{\delta}\right)} + 2|a|XH \cdot \sqrt{\frac{d}{m}\log\left(\frac{2d}{\delta}\right)} \ . \tag{3.20}$$

Proof in 3.5.4. The former expression displays the contribution of the non-linear part of the loss, keeping aside what is missing: a deviation of the empirical mean operator from its population mean. When $\boldsymbol{\mu}$ is not known because of partial label knowledge, the choice of any estimator would affect the bound only through that norm discrepancy. The second expression highlights the interplay of the two loss components. $c(X,H)$ is the only non-linear term, which may well be predominant in the bound for fast-growing losses, *e.g.* strongly convex. Moreover, we confirm that the linear-odd part does not change the complexity and only affects the statistical penalty by a linear factor, with a dependency on d. We remark that we could as well obtain a bound in the shape of the first statement of Theorem 5; we opt for an excess-risk bound to compare easily with prior work in Chapter 5.

Linear-odd losses are also calibrated under mild conditions.

Lemma 24. *Every a-linear-odd loss that is non-negative, convex, differentiable in 0 and with $a < 0$ is calibrated.*

Proof in 3.5.5. Consider again Table 3.1. Those losses are all convex and all have $a < 0$. They are also differentiable in 0 with the exception of ρ-loss and "perceptron". This simple proof can be extended and used for unhinged loss, if we assume bounded models [van Rooyen et al., 2015]. As a consequence of Theorem 10 we know that the previous generalization bound for the ℓ-risk translates into a guarantee for the generalization error.

Meta-Algorithm 1: Weakly supervised classification via statistical sufficiency — the two-step procedure

Input: $\widetilde{\mathcal{S}}$, ℓ is a-LOL, $\mathcal{H}$, $\lambda > 0$

$$(i) \quad \hat{\boldsymbol{\mu}} \leftarrow \text{estimate from } (\widetilde{\mathcal{S}}, \mathcal{H})$$

$$(ii) \quad \hat{\boldsymbol{\theta}} \leftarrow \operatorname*{argmin}_{\boldsymbol{\theta} \in \mathcal{H}} \frac{1}{2m} \sum_{i=1}^{m} \sum_{\sigma \in \mathcal{Y}} \ell(\sigma \langle \boldsymbol{\theta}, \boldsymbol{x}_i \rangle) + a \langle \boldsymbol{\theta}, \hat{\boldsymbol{\mu}} \rangle + \lambda \Omega(\boldsymbol{\theta})$$

Output: $\hat{\boldsymbol{\theta}}$

3.3 A two-step procedure for weakly supervised algorithms

Recall that in weakly supervised scenarios we learn on $\widetilde{\mathcal{S}}$ with partially observable labels, but aim to generalize to $\mathcal{D}$. Let us assume to know an algorithm A that learns by ERM of a linear-odd loss ℓ over a sample $\mathcal{S}$ from the non-corrupted distribution $\mathcal{D}$. Can we use algorithm A as is for learning from $\widetilde{\mathcal{S}}$? By sufficiency, Corollary 20 leads to a principled two-step approach:

> i estimate $\hat{\boldsymbol{\mu}}$ from $\widetilde{\mathcal{S}}$
>
> ii run algorithm A with ℓ computed with the estimated $\hat{\boldsymbol{\mu}}$

Meta Algorithm 1 formalizes this two-step procedure. The framework is as abstract as possible. Step (ii) is the same for every possible weakly supervised scenario, while to implement Step (i) we need to know the particular type of supervision we are dealing with. In other words, any weakly supervised classification problem can be cast into the convex optimization problem of Step (ii), once the label sufficient statistic — or its estimator — is available. This direction has been explored by the work on learning from label proportions of Quadrianto et al. [2009, with logistic loss] and Patrini et al. [2014, symmetric proper losses], and in the setting of noisy labels by Gao et al. [2016, logistic loss]. The approach contrasts with *ad-hoc* optimization methods often aiming to recover the latent labels by coordinate descent and EM (see Chapter 2). Instead, the only difficulty here is to come up with a well-behaved estimator of $\boldsymbol{\mu}$ — a statistic independent from both model h and loss ℓ.

Meta Algorithm 1 comes with the generalization guarantees elaborated in this Chapter. Theorem 23 bounds in probability the ℓ-excess risk and, with Lemma 24, the true risk. It is more difficult to derive finite-sample formal results on the output of Meta Algorithm 1, unless we work under additional assumptions. One such result is given in Altun and Smola [2006]. We reformulate it for the whole class of convex

and differentiable linear-odd losses. The Lemma gives an answer to the interesting question of how much model $\hat{\boldsymbol{\theta}}$ may diverge from the one that would be computed from empirical mean operator $\boldsymbol{\mu} = \mathbb{E}_{\mathcal{S}}[y\boldsymbol{x}]$.

Lemma 25. *Let ℓ be a-LOL convex and differentiable and let $\lambda > 0$. Call $\hat{\boldsymbol{\theta}}$ and $\boldsymbol{\theta}^\star$ the minimizer of (ii) in Meta Algorithm 1 when, respectively, the empirical risk is computed with $\hat{\boldsymbol{\mu}}$ from the estimator of (i) and $\boldsymbol{\mu}$. Then*

$$\left\|\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}^\star\right\|_2 \leq \frac{|a|}{\lambda}\left\|\hat{\boldsymbol{\mu}} - \boldsymbol{\mu}\right\|_2 . \tag{3.21}$$

This Lemma transfers the quality of the sufficient statistic estimation to the quality of the final model. Therefore, by using convex and differentiable linear-odd losses, any well-behaved mean operator estimator would be enough for obtaining reliable loss minimizers. A tighter result than the previous Lemma is the following, a data dependent approximation bound. It requires twice differentiability and strong convexity.

Let us first define some useful quantities. Let $\boldsymbol{f}_k \in \mathbb{R}^m$ denote the vector encoding the k^{th} variable in $\mathcal{S}$: $f_{ki} = x_{ik}$. For any $k \in [d]$, let:

$$\overline{\boldsymbol{f}}_k \doteq \left(\frac{d}{\sum_k \|\boldsymbol{f}_k\|_2^2}\right)^{\frac{d-1}{2d}} \boldsymbol{f}_k \tag{3.22}$$

denote a normalization of vectors $\boldsymbol{f}_k$ in the sense that:

$$\frac{1}{d}\sum_k \left\|\overline{\boldsymbol{f}}_k\right\|_2^2 = \frac{1}{d}\left(\frac{d}{\sum_k \|\boldsymbol{f}_k\|_2^2}\right)^{1-\frac{1}{d}} \sum_k \|\boldsymbol{f}_k\|_2^2 \tag{3.23}$$

$$= \left(\frac{1}{d}\sum_k \|\boldsymbol{f}_k\|_2^2\right)^{\frac{1}{d}} . \tag{3.24}$$

Let $\overline{F}$ collect all vectors $\overline{\boldsymbol{f}}_k$ in column and let F collect all vectors $\boldsymbol{f}_k$ in column. Without loss of generality, we assume $F^\top F \succ 0$, *i.e.* $F^\top F$ positive definite (*i.e.* no feature is a linear combination of the others), implying that $\overline{F}^\top \overline{F} \succ 0$ as well, because the columns of $\overline{F}$ are just positive rescaling of the columns of V.

Theorem 26. *Hold the same conditions of Lemma 25 and additionally assume that ℓ is twice differentiable and γ-strongly convex with $\gamma > 0$. Let m be the learning sample size. Then the following holds:*

$$\left\|\boldsymbol{\theta}^\star - \hat{\boldsymbol{\theta}}\right\|_2 \leq \frac{|a|}{\lambda + \frac{1}{em}\gamma \operatorname{vol}^2(\overline{F})}\left\|\boldsymbol{\mu} - \hat{\boldsymbol{\mu}}\right\|_2 , \tag{3.25}$$

where $\operatorname{vol}(\overline{F}) \doteq \sqrt{\det \overline{F}^\top \overline{F}}$ *denote the volume of the (row/column) system of* $\overline{F}$.

Proof in 3.5.6. To see how large the denominator in 3.25 can be, consider the simple case where all eigenvalues of $\overline{F}^\top \overline{F}$, $\lambda_k(\overline{F}^\top \overline{F}) \in [\lambda_\circ \pm \delta]$ for small δ. In this

Algorithm 2: μSGD with L_2 regularization

Input: $\mathcal{S}_{2x}, \boldsymbol{\mu}$, ℓ is a-LOL, $\lambda > 0$, $T > 0$
$\boldsymbol{\theta}^0 \leftarrow \mathbf{0}$
For any $t = 1, \dots, T$:
  Pick $i \in [|\mathcal{S}_{2x}|]$ uniformly at random
  $\eta \leftarrow 1/(\lambda t)$
  Pick any $\boldsymbol{v} \in \partial \ell(y_i \langle \boldsymbol{\theta}^t, \boldsymbol{x}_i \rangle)$
  $\boldsymbol{\theta}^{t+1} \leftarrow (1 - \eta\lambda)\boldsymbol{\theta}^t - \eta(\boldsymbol{v} \; + a\boldsymbol{\mu}/2 \;)$
  $\boldsymbol{\theta}^{t+1} \leftarrow \min\left\{\boldsymbol{\theta}^{t+1}, \boldsymbol{\theta}^{t+1}/(\sqrt{\lambda} \cdot \|\boldsymbol{\theta}^{t+1}\|_2)\right\}$
Output: $\boldsymbol{\theta}^T$

Algorithm 3: μ proximal algorithm with $\Theta(\cdot)$ regularization

Input: $\mathcal{S}_{2x}, \boldsymbol{\mu}$, ℓ is a-LOL; $\lambda > 0$; $T > 0$
$\boldsymbol{\theta}^0 \leftarrow \mathbf{0}$
For any $t = 1, \dots, T$:
  $\eta \leftarrow 1/(\lambda t)$ or found by line search
  $\boldsymbol{\theta}^{t+1} \leftarrow \mathrm{prox}_{\lambda\Theta}\left(\boldsymbol{\theta}^t + \eta\left(\partial R_{\mathcal{S}_{2x},\ell}(\boldsymbol{\theta}^t) \; + a\boldsymbol{\mu}/2 \;\right)\right)$
Output: $\boldsymbol{\theta}^T$

case, $\mathrm{vol}^2(\overline{F})$ is proportional to the "average feature norm":

$$\frac{\det\left(\overline{F}^\top \overline{F}\right)}{m} = \frac{\mathrm{tr}\, V^\top V}{md} + o(\delta) = \frac{\sum_i \|\boldsymbol{x}_i\|_2^2}{md} + o(\delta) \; . \tag{3.26}$$

Theorem 41 in Chapter 4 characterizes further this data dependent results by exploiting additional assumptions.

We now consider two more concrete examples on how to take Step (ii) of the abstract Meta Algorithm 1 for well-known learning algorithms. Algorithm 2, μSGD, adapts SGD to weak supervision. For the sake of presentation, we work on a simple version of SGD based on subgradient descent with L_2 regularization from PEGASO of Shalev-Shwartz et al. [2011]. Given $\boldsymbol{\mu}$, changes are trivial: construct $\mathcal{S}_{2x}$ from $\widetilde{\mathcal{S}}$ and sum $a\boldsymbol{\mu}/2$ to the subgradients of each example of $\mathcal{S}_{2x}$. The only changes are highlighted in grey.

A second noticeable example is given by the family of proximal algorithms [Bach et al., 2012]. The same *modus operandi* leads to Algorithm 3, where the proximal map is $\mathrm{prox}_{\lambda\Theta}(\boldsymbol{x}) = \mathrm{argmin}_{\boldsymbol{x}'} \lambda\Theta(\boldsymbol{x}') + \frac{1}{2}\|\boldsymbol{x} - \boldsymbol{x}'\|_2^2$ and $\Theta(\cdot)$ is a regularizer, non necessarily smooth. With some abuse of notation, ∂R indicates any vector in the subdifferential set of the (decomposable) empirical risk. Once again, the adaptation works by summing $a\boldsymbol{\mu}/2$ in the gradient step and changing the input to $\mathcal{S}_{2x}$.

3.4 Discussion

We have presented novel theoretical tools apt to elaborate a learning theory for weakly supervised problems. The level of abstraction is high enough to let us formulate a Meta Algorithm that is problem agnostic but simply adaptable for exploiting well known gradient-based solvers. Finite sample bounds support the viability of our approach. The rest of the Thesis will put these ideas to use by specialization under different assumptions on the quality of supervision. Applications elsewhere may easily draw from this Chapter as well.

3.5 Appendix: proofs

3.5.1 Proof of Corollary 20

We need to show the double implication that defines sufficiency for y.

$\Rightarrow$) By Factorization Theorem 18, $R_{\mathcal{S},\ell}(h) - R_{\mathcal{S}',\ell}(h)$ is label independent only if the odd part cancels out.

$\Leftarrow$) If $\boldsymbol{\mu} = \boldsymbol{\mu}'$ then $R_{\mathcal{S},\ell}(h) - R_{\mathcal{S}',\ell}(h)$ is independent from the label, because the label only appears in the mean operator due to Factorization Theorem 18.

3.5.2 Proof of Lemma 21

Consider the class of losses satisfying $\ell(x) - \ell(-x) = 2ax$. For any element of the class, define $\ell_e(x) = \ell(x) - ax$, which is even. In fact we have:

$$\ell_e(-x) = \ell(-x) + ax = \ell(x) - 2ax + ax = \ell(x) - ax = \ell_e(x) \ . \tag{3.27}$$

3.5.3 Proof of Lemma 22

Suppose without loss of generality that $\boldsymbol{x}_i = \boldsymbol{x}_{m+i}$. The proof relies on the observation that $\forall \boldsymbol{\sigma} \in \mathcal{Y}^{2m}$:

$$\begin{aligned} \operatorname*{argsup}_{\boldsymbol{\theta}\in\mathcal{H}} \mathbb{E}_{\mathcal{S}}[\sigma(\boldsymbol{x})\langle\boldsymbol{\theta},\boldsymbol{x}\rangle] &= \frac{1}{2m} \operatorname*{argsup}_{\boldsymbol{\theta}\in\mathcal{H}} \sum_i \sigma_i \langle\boldsymbol{\theta},\boldsymbol{x}_i\rangle & (3.28)\\ &= \frac{\sup_{\mathcal{H}} \|\boldsymbol{\theta}\|_2}{\|\sum_i \sigma_i \boldsymbol{x}_i\|_2} \sum_i \sigma_i \boldsymbol{x}_i \ . & (3.29) \end{aligned}$$

In fact, this follows from Cauchy-Schwartz inequality – the classifier that maximizes the inner product with the sum is indeed proportional to the normalized sum, times the maximal norm of a classifier. So:

$$\begin{aligned} \mathcal{R}(\mathcal{H}\circ\mathcal{S}_{2x}) &= \mathbb{E}_{\mathcal{Y}^{2m}} \sup_{h\in\mathcal{H}} \{\mathbb{E}_{\mathcal{S}_{2x}}[\sigma(\boldsymbol{x})h(\boldsymbol{x})]\} & (3.30)\\ &= \frac{\sup_{\mathcal{H}} \|\theta\|_2}{2m} \cdot \mathbb{E}_{\mathcal{Y}^{2m}} \left[\frac{\left(\sum_{i=1}^{2m} \sigma_i \boldsymbol{x}_i\right)^\top \left(\sum_{i=1}^{2m} \sigma_i \boldsymbol{x}_i\right)}{\|\sum_{i=1}^{2m} \sigma_i \boldsymbol{x}_i\|_2} \right] & (3.31)\\ &= \sup_{\mathcal{H}} \|\theta\|_2 \cdot \mathbb{E}_{\mathcal{Y}^{2m}} \left[\frac{1}{2m} \cdot \left\| \sum_{i=1}^{2m} \sigma_i \boldsymbol{x}_i \right\|_2 \right] \ . & (3.32) \end{aligned}$$

Remark that whenever $\sigma_i = -\sigma_{m+i}$, $\boldsymbol{x}_i$ disappears in the sum, and therefore the max norm for the sum may decrease as well. This suggests to split the 2^{2m} assignations into 2^m groups of size 2^m, ranging over the possible number of observations taken into account in the sum. They can be factored by a weighted sum of contributions of

each subset of indices $\mathcal{I} \subseteq [m]$ ranging over the non-duplicated observations:

$$\mathbb{E}_{\mathcal{Y}^{2m}}\left[\frac{1}{m}\cdot\left\|\sum_{i=1}^{2m}\sigma_i\boldsymbol{x}_i\right\|_2\right] = \frac{1}{2^{2m}}\sum_{\mathcal{I}\subseteq[m]}\frac{2^{m-|\mathcal{I}|}}{2m}\cdot\sum_{\boldsymbol{\sigma}\in\mathcal{Y}^{|\mathcal{I}|}}\sqrt{2}\left\|\sum_{i\in\mathcal{I}}\sigma_i\boldsymbol{x}_i\right\|_2 . \tag{3.33}$$

$$= \frac{\sqrt{2}}{2^m}\sum_{\mathcal{I}\subseteq[m]}\frac{1}{2m}\cdot\underbrace{\frac{1}{2^{|\mathcal{I}|}}\cdot\sum_{\boldsymbol{\sigma}\in\mathcal{Y}^{|\mathcal{I}|}}\left\|\sum_{i\in\mathcal{I}}\sigma_i\boldsymbol{x}_i\right\|_2}_{u_{|\mathcal{I}|}} . \tag{3.34}$$

The $\sqrt{2}$ factor appears because of the fact that we now consider only the observations of $\mathcal{S}$. For any *fixed* $\mathcal{I}$, we renumber its observations in $[|\mathcal{I}|]$ for simplicity, and observe that, since $\sqrt{1+x} \leq 1 + x/2$:

$$u_{|\mathcal{I}|} = \frac{1}{2^{|\mathcal{I}|}}\sum_{\boldsymbol{\sigma}\in\mathcal{Y}^{|\mathcal{I}|}}\sqrt{\sum_{i\in\mathcal{I}}\|\boldsymbol{x}_i\|_2^2 + \sum_{i_1\neq i_2}\sigma_{i_1}\sigma_{i_2}\boldsymbol{x}_{i_1}^\top\boldsymbol{x}_{i_2}} \tag{3.35}$$

$$= \frac{\sqrt{\sum_{i\in\mathcal{I}}\|\boldsymbol{x}_i\|_2^2}}{2^{|\mathcal{I}|}}\sum_{\boldsymbol{\sigma}\in\mathcal{Y}^{|\mathcal{I}|}}\sqrt{1+\frac{\sum_{i_1\neq i_2}\sigma_{i_1}\sigma_{i_2}\boldsymbol{x}_{i_1}^\top\boldsymbol{x}_{i_2}}{\sum_{i\in\mathcal{I}}\|\boldsymbol{x}_i\|_2^2}} \tag{3.36}$$

$$\leq \frac{\sqrt{\sum_{i\in\mathcal{I}}\|\boldsymbol{x}_i\|_2^2}}{2^{|\mathcal{I}|}}\sum_{\boldsymbol{\sigma}\in\mathcal{Y}^{|\mathcal{I}|}}\left(1+\frac{\sum_{i_1\neq i_2}\sigma_{i_1}\sigma_{i_2}\boldsymbol{x}_{i_1}^\top\boldsymbol{x}_{i_2}}{2\sum_{i\in\mathcal{I}}\|\boldsymbol{x}_i\|_2^2}\right) \tag{3.37}$$

$$= \sqrt{\sum_{i\in\mathcal{I}}\|\boldsymbol{x}_i\|_2^2} + \frac{1}{2^{|\mathcal{I}|}\cdot 2\sum_{i\in\mathcal{I}}\|\boldsymbol{x}_i\|_2^2}\cdot\sum_{\boldsymbol{\sigma}\in\mathcal{Y}^{|\mathcal{I}|}}\sum_{i_1\neq i_2}\sigma_{i_1}\sigma_{i_2}\boldsymbol{x}_{i_1}^\top\boldsymbol{x}_{i_2} \tag{3.38}$$

$$= \sqrt{\sum_{i\in\mathcal{I}}\|\boldsymbol{x}_i\|_2^2} + \frac{1}{2^{|\mathcal{I}|}\cdot 2\sum_{i\in\mathcal{I}}\|\boldsymbol{x}_i\|_2^2}\cdot\sum_{i_1\neq i_2}\boldsymbol{x}_{i_1}^\top\boldsymbol{x}_{i_2}\cdot\underbrace{\left(\sum_{\boldsymbol{\sigma}\in\mathcal{Y}^{|\mathcal{I}|}}\sigma_{i_1}\sigma_{i_2}\right)}_{=0} \tag{3.39}$$

$$= \sqrt{\sum_{i\in\mathcal{I}}\|\boldsymbol{x}_i\|_2^2} \tag{3.40}$$

$$\leq \sqrt{|\mathcal{I}|}\cdot X . \tag{3.41}$$

Plugging this in Equation 3.34 yields:

$$\frac{1}{X}\cdot\mathbb{E}_{\mathcal{Y}^{2m}}\left[\frac{1}{m}\cdot\left\|\sum_{i=1}^{2m}\sigma_i\boldsymbol{x}_i\right\|_2\right] \leq \frac{\sqrt{2}}{2^m}\sum_{k=0}^{m}\frac{\sqrt{k}}{2m}\binom{m}{k} . \tag{3.42}$$

Since m is even:

$$\mathbb{E}_{\mathcal{Y}^{2m}}\left[\frac{1}{2m}\cdot\left\|\sum_{i=1}^{2m}\sigma_i\boldsymbol{x}_i\right\|_2\right] \leq \frac{\sqrt{2}}{2^m}\sum_{k=0}^{(m/2)-1}\frac{\sqrt{k}}{2m}\binom{m}{k} + \frac{\sqrt{2}}{2^m}\sum_{k=m/2}^{m}\frac{\sqrt{k}}{2m}\binom{m}{k} . \tag{3.43}$$

Notice that the left one trivially satisfies:

$$\begin{aligned}
\frac{\sqrt{2}}{2^m}\sum_{k=0}^{(m/2)-1}\frac{\sqrt{k}}{2m}\binom{m}{k} &\leq \frac{\sqrt{2}}{2^m}\sum_{k=0}^{(m/2)-1}\frac{1}{2m}\cdot\sqrt{\frac{m-2}{2}}\binom{m}{k} && (3.44)\\
&= \frac{1}{2}\cdot\sqrt{\frac{1}{m}-\frac{2}{m^2}}\cdot\frac{1}{2^m}\sum_{k=0}^{(m/2)-1}\binom{m}{k} && (3.45)\\
&\leq \frac{1}{4}\cdot\sqrt{\frac{1}{m}-\frac{2}{m^2}} && (3.46)
\end{aligned}$$

Also, the right one satisfies:

$$\begin{aligned}
\frac{\sqrt{2}}{2^m}\sum_{k=m/2}^{m}\frac{\sqrt{k}}{2m}\binom{m}{k} &\leq \frac{\sqrt{2}}{2^m}\sum_{k=m/2}^{m}\frac{\sqrt{m}}{2m}\binom{m}{k} && (3.47)\\
&= \frac{1}{\sqrt{2m}}\cdot\frac{1}{2^m}\sum_{k=m/2}^{m}\binom{m}{k} && (3.48)\\
&= \frac{1}{2}\cdot\frac{1}{\sqrt{2m}}\ . && (3.49)
\end{aligned}$$

We get:

$$\begin{aligned}
\frac{1}{X}\cdot\mathbb{E}_{y^{2m}}\left[\frac{1}{m}\cdot\left\|\sum_{i=1}^{2m}\sigma_i\boldsymbol{x}_i\right\|_2\right] &\leq \frac{1}{4}\cdot\sqrt{\frac{1}{m}-\frac{2}{m^2}}+\frac{1}{2}\cdot\sqrt{\frac{1}{2m}} && (3.50)\\
&= \frac{1}{\sqrt{2m}}\cdot\left(\frac{1}{2}+\frac{1}{2}\sqrt{\frac{1}{2}-\frac{1}{m}}\right)\ . && (3.51)
\end{aligned}$$

And finally:

$$\mathcal{R}(\mathcal{H}\circ\mathcal{S}_{2x}) \leq v\cdot\frac{XH}{\sqrt{2m}}\ , \tag{3.52}$$

with:

$$v \doteq \frac{1}{2}+\frac{1}{2}\sqrt{\frac{1}{2}-\frac{1}{m}}\ , \tag{3.53}$$

as claimed.

3.5.4 Proof of Theorem 23

We start by proving a helper Lemma, an application of McDiarmid 's inequality to evaluate the convergence of the empirical mean operator to its population counterpart.

Lemma 27. *Suppose* $\mathbb{R}^d \supseteq \mathcal{X} = \{\boldsymbol{x} : \|\boldsymbol{x}\|_2 \leq X < \infty\}$ *be the observations space. Then for*

any $\delta > 0$ with probability at least $1 - \delta$:

$$\|\boldsymbol{\mu}_{\mathcal{D}} - \boldsymbol{\mu}_{\mathcal{S}}\|_2 \leq X \cdot \sqrt{\frac{d}{m} \log\left(\frac{d}{\delta}\right)} \ . \tag{3.54}$$

Proof Let $\mathcal{S}$ and $\mathcal{S}'$ be two learning samples that differ for only one example $(\boldsymbol{x}_i, y_i) \neq (\boldsymbol{x}_{i'}, y_{i'})$. Let first consider the one-dimensional case. We refer to the k-dimensional component of $\boldsymbol{\mu}$ with $\boldsymbol{\mu}^k$. For any $\mathcal{S}, \mathcal{S}'$ and any $k \in [d]$ it holds:

$$\left|\boldsymbol{\mu}^k_{\mathcal{S}} - \boldsymbol{\mu}^k_{\mathcal{S}'}\right| = \frac{1}{m}\left|\boldsymbol{x}^k_i y_i - \boldsymbol{x}^k_{i'} y_{i'}\right| \tag{3.55}$$

$$\leq \frac{X}{m}\left|y_i - y_{i'}\right| \tag{3.56}$$

$$\leq \frac{2X}{m} \ . \tag{3.57}$$

This satisfies the bounded difference condition of McDiarmid's inequality, which let us write for any $k \in [d]$ and any $\epsilon > 0$ that:

$$\mathbb{P}\left(\left|\boldsymbol{\mu}^k_{\mathcal{D}} - \boldsymbol{\mu}^k_{\mathcal{S}}\right| \geq \epsilon\right) \leq \exp\left(-\frac{m\epsilon^2}{2X^2}\right) \tag{3.58}$$

and the multi-dimensional case, by union bound:

$$\mathbb{P}\left(\exists k \in [d] : \left|\boldsymbol{\mu}^k_{\mathcal{D}} - \boldsymbol{\mu}^k_{\mathcal{S}}\right| \geq \epsilon\right) \leq d \exp\left(-\frac{m\epsilon^2}{2X^2}\right) \ . \tag{3.59}$$

Then by negation:

$$\mathbb{P}\left(\forall k \in [d] : \left|\boldsymbol{\mu}^k_{\mathcal{D}} - \boldsymbol{\mu}^k_{\mathcal{S}}\right| \leq \epsilon\right) \geq 1 - d \exp\left(-\frac{m\epsilon^2}{2X^2}\right) \ , \tag{3.60}$$

which implies that for any $\delta > 0$ with probability $1 - \delta$:

$$X\sqrt{\frac{2}{m}\log\left(\frac{d}{\delta}\right)} \geq \|\boldsymbol{\mu}_{\mathcal{D}} - \boldsymbol{\mu}_{\mathcal{S}}\|_\infty \geq d^{-1/2}\|\boldsymbol{\mu}_{\mathcal{D}} - \boldsymbol{\mu}_{\mathcal{S}}\|_2 \ . \tag{3.61}$$

This concludes the proof. ■

We now restate and prove Theorem 23.

Theorem 23 *Assume ℓ is a-LOL and L-Lipschitz. Suppose $\mathbb{R}^d \supseteq \mathcal{X} = \{\boldsymbol{x} : \|\boldsymbol{x}\|_2 \leq X < \infty\}$ be the observations space, and $\mathcal{H} = \{\boldsymbol{\theta} : \|\boldsymbol{\theta}\|_2 \leq H < \infty\}$ be the space of linear hypotheses. Let $c(X, H) \doteq \max_{y \in \mathcal{Y}} \ell(yXH)$. Let $\hat{\boldsymbol{\theta}} = \operatorname{argmin}_{\boldsymbol{\theta} \in \mathcal{H}} R_{\mathcal{S},\ell}(\boldsymbol{\theta})$. Then for any*

$\delta > 0$, with probability at least $1 - \delta$:

$$R_{\mathcal{D},\ell}(\hat{\boldsymbol{\theta}}) - R_{\mathcal{D},\ell}(\boldsymbol{\theta}^\star) \leq \frac{\sqrt{2}+1}{2} \cdot \frac{XHL}{\sqrt{m}} + c(X,H) \cdot \sqrt{\frac{1}{m}\log\left(\frac{1}{\delta}\right)} + 2|a|H \cdot \|\boldsymbol{\mu}_{\mathcal{D}} - \boldsymbol{\mu}_{\mathcal{S}}\|_2 \ , \tag{3.62}$$

or more explicitly:

$$R_{\mathcal{D},\ell}(\hat{\boldsymbol{\theta}}) - R_{\mathcal{D},\ell}(\boldsymbol{\theta}^\star) \leq \frac{\sqrt{2}+1}{2} \cdot \frac{XHL}{\sqrt{m}} + c(X,H) \cdot \sqrt{\frac{1}{m}\log\left(\frac{2}{\delta}\right)} + 2|a|XH \cdot \sqrt{\frac{d}{m}\log\left(\frac{2d}{\delta}\right)} \tag{3.63}$$

Proof Let $\boldsymbol{\theta}^\star = \operatorname{argmin}_{\boldsymbol{\theta}\in\mathcal{H}} R_{\mathcal{D},\ell}(\boldsymbol{\theta})$. We have:

$$\begin{aligned}
R_{\mathcal{D},\ell}(\hat{\boldsymbol{\theta}}) - R_{\mathcal{D},\ell}(\boldsymbol{\theta}^\star) &= \frac{1}{2}R_{\mathcal{D}_{2x},\ell}(\hat{\boldsymbol{\theta}}) + a\langle\hat{\boldsymbol{\theta}}, \boldsymbol{\mu}_{\mathcal{D}}\rangle - \frac{1}{2}R_{\mathcal{D}_{2x},\ell}(\boldsymbol{\theta}^\star) - a\langle\boldsymbol{\theta}^\star, \boldsymbol{\mu}_{\mathcal{D}}\rangle && (3.64)\\
&= \frac{1}{2}\left(R_{\mathcal{D}_{2x},\ell}(\hat{\boldsymbol{\theta}}) - R_{\mathcal{D}_{2x},\ell}(\boldsymbol{\theta}^\star)\right) + a\langle\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}^\star, \boldsymbol{\mu}_{\mathcal{D}}\rangle && (3.65)\\
&= \frac{1}{2}\left(R_{\mathcal{S}_{2x},\ell}(\hat{\boldsymbol{\theta}}) - R_{\mathcal{S}_{2x},\ell}(\boldsymbol{\theta}^\star)\right) + a\langle\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}^\star, \boldsymbol{\mu}_{\mathcal{D}}\rangle && (3.66)\\
&\quad + \frac{1}{2}\Big(R_{\mathcal{D}_{2x},\ell}(\hat{\boldsymbol{\theta}}) - R_{\mathcal{S}_{2x},\ell}(\hat{\boldsymbol{\theta}}) - R_{\mathcal{D}_{2x},\ell}(\boldsymbol{\theta}^\star) + R_{\mathcal{S}_{2x},\ell}(\boldsymbol{\theta}^\star)\Big) \ \Big\} A_1 \ . && (3.67)
\end{aligned}$$

Step 3.64 is obtained by the equality $R_{\mathcal{D},\ell}(\boldsymbol{\theta}) = \frac{1}{2}R_{\mathcal{D}_{2x},\ell}(\boldsymbol{\theta}) + a\langle\boldsymbol{\theta}, \boldsymbol{\mu}_{\mathcal{D}}\rangle$ for any $\boldsymbol{\theta}$. Applying the same equality with regard to $\mathcal{S}$, we have:

$$R_{\mathcal{D},\ell}(\hat{\boldsymbol{\theta}}) - R_{\mathcal{D},\ell}(\boldsymbol{\theta}^\star) \leq \underbrace{R_{\mathcal{S},\ell}(\hat{\boldsymbol{\theta}}) - R_{\mathcal{S},\ell}(\boldsymbol{\theta}^\star)}_{A_2} + \underbrace{a\langle\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}^\star, \boldsymbol{\mu}_{\mathcal{D}} - \boldsymbol{\mu}_{\mathcal{S}}\rangle}_{A_3} + A_1 \ . \tag{3.68}$$

A_2 is never more than 0 because $\hat{\boldsymbol{\theta}}$ is the minimizer of $R_{\mathcal{S},\ell}(\boldsymbol{\theta})$. From the Cauchy-Schwarz inequality and bounded models it holds true that:

$$A_3 \leq |a|\left\|\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}^\star\right\|_2 \cdot \left\|\boldsymbol{\mu}_{\mathcal{D}} - \boldsymbol{\mu}_{\mathcal{S}}\right\|_2 \leq 2|a|H\left\|\boldsymbol{\mu}_{\mathcal{D}} - \boldsymbol{\mu}_{\mathcal{S}}\right\|_2 \ . \tag{3.69}$$

We could treat A_1 by calling standard bounds based on Rademacher complexity on a sample with size $2m$. Since the complexity $\mathcal{R}(\mathcal{H} \circ \mathcal{S})$ does not depend on labels, its value would be the same — modulo the change of sample size — for both $\mathcal{S}$ and $\mathcal{S}_{2x}$, as they are computed with same loss and observations. However, the special structure of $\mathcal{S}_{2x}$ allows us to obtain a tighter structural complexity term, due to

cancellation effects as proven by Lemma 22. In order to exploit it, we first observe:

$$A_1 = \frac{1}{2}\Big(R_{\mathcal{D}_{2x},\ell}(\hat{\boldsymbol{\theta}}) - R_{\mathcal{S}_{2x},\ell}(\hat{\boldsymbol{\theta}}) - R_{\mathcal{D}_{2x},\ell}(\boldsymbol{\theta}^\star) + R_{\mathcal{S}_{2x},\ell}(\boldsymbol{\theta}^\star)\Big) \tag{3.70}$$

$$\leq \sup_{\boldsymbol{\theta}\in\mathcal{H}} |R_{\mathcal{D}_{2x},\ell}(\boldsymbol{\theta}) - R_{\mathcal{S}_{2x},\ell}(\boldsymbol{\theta})| \tag{3.71}$$

$$= \phi(\mathcal{S}_{2x}) \ . \tag{3.72}$$

Similarly to the proof of Theorem 5, we recall Lemma 6 and notice that for any bounded loss function, the function ϕ satisfies McDiarmid's inequality with $C = 1/m \cdot c(X,H)$. Therefore with probability $\geq 1-\delta$:

$$A_1 \leq 2L\ \mathbb{E}_{\mathcal{D}_{2x}}\ \mathcal{R}(\mathcal{H}\circ\mathcal{S}_{2x}) + \frac{c(X,H)}{m}\sqrt{m\log\frac{1}{\delta}} \tag{3.73}$$

$$\leq 2v\ \frac{XHL}{\sqrt{2m}} + c(X,H)\sqrt{\frac{1}{m}\log\frac{1}{\delta}} \tag{3.74}$$

$$\leq \frac{\sqrt{2}+1}{2}\frac{XHL}{\sqrt{m}} + c(X,H)\sqrt{\frac{1}{m}\log\frac{1}{\delta}} \tag{3.75}$$

where we also applied Lemmas 14 and 22 and the fact that:

$$v = \frac{1}{2} + \frac{1}{2}\sqrt{\frac{1}{2}-\frac{1}{m}} < \frac{\sqrt{2}+1}{2\sqrt{2}},\ \forall m > 1.$$

Finally, we get with probability at least $1-\delta$, $\delta > 0$ that:

$$R_{\mathcal{D},\ell}(\hat{\boldsymbol{\theta}}) - R_{\mathcal{D},\ell}(\boldsymbol{\theta}^\star) \leq \frac{\sqrt{2}+1}{2}\cdot\frac{XHL}{\sqrt{m}}$$
$$+ c(X,H)\cdot\sqrt{\frac{1}{m}\log\left(\frac{1}{\delta}\right)} + 2|a|H\cdot\|\boldsymbol{\mu}_{\mathcal{D}} - \boldsymbol{\mu}_{\mathcal{S}}\|_2 \ . \tag{3.76}$$

This proves the first statement. For the second statement, we apply Lemma 27 that provides the probabilistic bound for the norm discrepancy of the mean operators. We need to combine the two results. For any two events E, F it holds that:

$$1 - \mathbb{P}(E\wedge F) = \mathbb{P}(\neg(E\wedge F)) \tag{3.77}$$

$$= \mathbb{P}(\neg E \vee \neg F) \tag{3.78}$$

$$\leq \mathbb{P}(\neg E) + \mathbb{P}(\neg F) \tag{3.79}$$

Now consider that events:

$$E \doteq \left\{3.76\right\}\ , F \doteq \left\{\|\boldsymbol{\mu}_{\mathcal{D}} - \boldsymbol{\mu}_{\mathcal{S}}\|_2 \leq X\cdot\sqrt{\frac{d}{m}\log\left(\frac{2d}{\delta}\right)}\right\}$$

are true with probability at least $1-\delta/2$, or equivalently they are false with at most

$\delta/2$ probability. We write:

$$1 - \mathbb{P}\left(E \wedge F\right) \leq \delta/2 - \delta/2 = \delta \ , \tag{3.80}$$

and therefore with $1-\delta$ probability: $\mathbb{P}\left(E \wedge F\right) \geq 1-\delta$. Finally:

$$\begin{aligned} R_{\mathcal{D},\ell}(\hat{\boldsymbol{\theta}}) - R_{\mathcal{D},\ell}(\boldsymbol{\theta}^\star) \leq \frac{\sqrt{2}+1}{2} \cdot \frac{XHL}{\sqrt{m}} + \\ c(X,H) \cdot \sqrt{\frac{1}{m}\log\left(\frac{2}{\delta}\right)} + 2|a|XH \cdot \sqrt{\frac{d}{m}\log\left(\frac{2d}{\delta}\right)} \ . \end{aligned} \tag{3.81}$$

■

3.5.5 Proof of Lemma 24

Consider Definition 9. Since we assume convex and differentiable losses, it suffices to compute the derivative in 0.

$$\ell'(x) = \ell'_e(x) + a \tag{3.82}$$

and because ℓ'_e is an odd function, as derivative of an even function, $\ell'_e(0) = 0$. Therefore:

$$\ell'(0) < 0 \iff a < 0 \tag{3.83}$$

3.5.6 Proof of Theorem 26

Let us define:

$$R_{\mathcal{S}_X,\ell}(\boldsymbol{\theta},\boldsymbol{\mu}) = \frac{1}{2m}\left(\sum_{i\in[m]}\sum_{\sigma}\ell(\sigma\langle\boldsymbol{\theta},\boldsymbol{x}_i\rangle)\right) + a\langle\boldsymbol{\theta},\boldsymbol{\mu}\rangle \ . \tag{3.84}$$

Define also the regularized loss:

$$R_{\mathcal{S}_X,\ell}(\boldsymbol{\theta},\boldsymbol{\mu},\lambda) \doteq R_{\mathcal{S}_X,\ell}(\boldsymbol{\theta},\boldsymbol{\mu}) + \lambda\|\boldsymbol{\theta}\|_2^2 \ . \tag{3.85}$$

Our proof begins following the same first steps as the proof of Lemma 17 in Altun and Smola [2006] (recalled in Lemma 25) and adding the steps that handle the lower bound on ℓ''. Consider the following auxiliary function $A_\ell(\boldsymbol{\tau})$:

$$A_\ell(\boldsymbol{\tau}) \doteq \left(\nabla R_{\mathcal{S}_X,\ell}(\boldsymbol{\theta}^\star,\boldsymbol{\mu}) - \nabla R_{\mathcal{S}_X,\ell}(\hat{\boldsymbol{\theta}},\hat{\boldsymbol{\mu}})\right)^\top(\boldsymbol{\tau}-\hat{\boldsymbol{\theta}}) + \lambda\left\|\boldsymbol{\tau}-\hat{\boldsymbol{\theta}}\right\|_2^2 \ , \tag{3.86}$$

where the gradient of $R_{\mathcal{S}_X,\ell}$ is computed with respect to parameter $\boldsymbol{\theta}$. The gradient

of $A_\ell(.)$ with respect to $\boldsymbol{\tau}$ is:

$$\nabla A_\ell(\boldsymbol{\tau}) = \nabla R_{\mathcal{S}_X,\ell}(\boldsymbol{\theta}^\star, \boldsymbol{\mu}) - \nabla R_{\mathcal{S}_X,\ell}(\hat{\boldsymbol{\theta}}, \hat{\boldsymbol{\mu}}) + 2\lambda(\boldsymbol{\tau} - \hat{\boldsymbol{\theta}}) \ . \tag{3.87}$$

The gradient is $\mathbf{0}$ when $\boldsymbol{\tau} = \boldsymbol{\theta}^\star$:

$$\begin{aligned}
\nabla A_\ell(\boldsymbol{\theta}^\star) &= \nabla R_{\mathcal{S}_X,\ell}(\boldsymbol{\theta}^\star, \boldsymbol{\mu}) - \nabla R_{\mathcal{S}_X,\ell}(\hat{\boldsymbol{\theta}}, \hat{\boldsymbol{\mu}}) + 2\lambda(\boldsymbol{\theta}^\star - \hat{\boldsymbol{\theta}}) && (3.88)\\
&= \nabla R_{\mathcal{S}_X,\ell}(\boldsymbol{\theta}^\star, \boldsymbol{\mu}, \lambda) - \nabla R_{\mathcal{S}_X,\ell}(\hat{\boldsymbol{\theta}}, \hat{\boldsymbol{\mu}}, \lambda) && (3.89)\\
&= \mathbf{0} \ , && (3.90)
\end{aligned}$$

since both gradients are $\mathbf{0}$ because of the optimality of $\boldsymbol{\theta}^\star$ and $\hat{\boldsymbol{\theta}}$ with respect to $R_{\mathcal{S}_X,\ell}(\cdot, \boldsymbol{\mu}, \lambda)$ and $R_{\mathcal{S}_X,\ell}(\cdot, \hat{\boldsymbol{\mu}}, \lambda)$. Moreover, the Hessian H of A_ℓ is $HA_\ell(\boldsymbol{\tau}) = 2\lambda I \succeq 0$, thus A_ℓ is convex and therefore we can state that it is minimal at $\boldsymbol{\tau} = \boldsymbol{\theta}^\star$. Additionally, $A_\ell(\hat{\boldsymbol{\theta}}) = 0$ by definition. It comes thus $A_\ell(\boldsymbol{\theta}^\star) \leq 0$, which yields equivalently:

$$\begin{aligned}
0 &\geq \left(\nabla R_{\mathcal{S}_X,\ell}(\boldsymbol{\theta}^\star, \boldsymbol{\mu}) - \nabla R_{\mathcal{S}_X,\ell}(\hat{\boldsymbol{\theta}}, \hat{\boldsymbol{\mu}})\right)^\top (\boldsymbol{\theta}^\star - \hat{\boldsymbol{\theta}}) + \lambda \left\|\boldsymbol{\theta}^\star - \hat{\boldsymbol{\theta}}\right\|_2^2 && (3.91)\\
&= \left(\frac{1}{2m}\sum_y \sum_i \nabla \ell(y\langle \boldsymbol{\theta}^\star, \boldsymbol{x}_i\rangle) + a\boldsymbol{\mu} - \frac{1}{2m}\sum_y \sum_i \nabla \ell(y\langle \hat{\boldsymbol{\theta}}, \boldsymbol{x}_i\rangle) - a\hat{\boldsymbol{\mu}}\right)^\top (\boldsymbol{\theta}^\star - \hat{\boldsymbol{\theta}})\\
&\quad + \lambda \left\|\boldsymbol{\theta}^\star - \hat{\boldsymbol{\theta}}\right\|_2^2 && (3.92)\\
&= \frac{1}{2m} \underbrace{\left(\sum_y \sum_i \ell(y\langle \boldsymbol{\theta}^\star, \boldsymbol{x}_i\rangle) - \sum_y \sum_i \ell(y\langle \hat{\boldsymbol{\theta}}, \boldsymbol{x}_i\rangle)\right)^\top (\boldsymbol{\theta}^\star - \hat{\boldsymbol{\theta}})}_{\doteq \beta}\\
&\quad + a\left(\boldsymbol{\mu} - \hat{\boldsymbol{\mu}}\right)^\top (\boldsymbol{\theta}^\star - \hat{\boldsymbol{\theta}}) + \lambda \left\|\boldsymbol{\theta}^\star - \hat{\boldsymbol{\theta}}\right\|_2^2 \ . && (3.93)
\end{aligned}$$

Let us lower bound β. We have $\ell(y\langle \boldsymbol{\theta}^\star, \boldsymbol{x}\rangle) = y\ell'(y\langle \boldsymbol{\theta}^\star, \boldsymbol{x}\rangle)\boldsymbol{x}$, and a Taylor expansion brings that for any $\boldsymbol{\theta}^\star, \hat{\boldsymbol{\theta}}$, there exists some $\alpha \in [0,1]$ such that, defining:

$$u_{\alpha,i} \doteq y\langle \alpha\boldsymbol{\theta}^\star + (1-\alpha)\hat{\boldsymbol{\theta}}, \boldsymbol{x}_i\rangle \ , \tag{3.94}$$

we have:

$$\ell'(y\langle \boldsymbol{\theta}^\star \boldsymbol{x}_i\rangle) = \ell'(y\langle \hat{\boldsymbol{\theta}}, \boldsymbol{x}_i\rangle) + y\langle \boldsymbol{\theta}^\star - \hat{\boldsymbol{\theta}}, \boldsymbol{x}_i\rangle \ell''(u_{\alpha,i}) \ . \tag{3.95}$$

Thus we get:

$$\beta = \left(\sum_y \sum_i \ell(y\langle \boldsymbol{\theta}^\star, \boldsymbol{x}_i \rangle) - \sum_y \sum_i \ell(y\langle \hat{\boldsymbol{\theta}}, \boldsymbol{x}_i \rangle) \right)^\top (\boldsymbol{\theta}^\star - \hat{\boldsymbol{\theta}}) \tag{3.96}$$

$$= \left(\sum_y \sum_i y \left(\ell'(y\langle \boldsymbol{\theta}^\star, \boldsymbol{x}_i \rangle) - \ell'(y\langle \hat{\boldsymbol{\theta}}, \boldsymbol{x}_i \rangle) \right) \boldsymbol{x}_i \right)^\top (\boldsymbol{\theta}^\star - \hat{\boldsymbol{\theta}}) \tag{3.97}$$

$$= \left(\sum_y \sum_i (\boldsymbol{\theta}^\star - \hat{\boldsymbol{\theta}})^\top \boldsymbol{x}_i \ell''(u_{\alpha,i}) \boldsymbol{x}_i \right)^\top (\boldsymbol{\theta}^\star - \hat{\boldsymbol{\theta}}) \tag{3.98}$$

$$= 2 \sum_i \left((\boldsymbol{\theta}^\star - \hat{\boldsymbol{\theta}})^\top \boldsymbol{x}_i \right)^2 \ell''(u_{\alpha,i}) \tag{3.99}$$

$$\geq 2\gamma \sum_i \left((\boldsymbol{\theta}^\star - \hat{\boldsymbol{\theta}})^\top \boldsymbol{x}_i \right)^2 \tag{3.100}$$

$$= 2\gamma (\boldsymbol{\theta}^\star - \hat{\boldsymbol{\theta}})^\top S S^\top (\boldsymbol{\theta}^\star - \hat{\boldsymbol{\theta}}) \ , \tag{3.101}$$

where matrix $S \in \mathbb{R}^{d \times m}$ is formed by the observations of $\mathcal{S}_X$ in columns. Inequality 3.100 comes from the fact that we require γ-strongly convex losses that are also twice differentiable, which implies that the second derivative is lower bounded by γ.

Define $T \doteq (d / \sum_i \|\boldsymbol{x}_i\|_2^2) S S^\top$. Its trace satisfies $\operatorname{tr} T = d$. Let $\lambda_d \geq \lambda_{d-1} \geq ... \geq \lambda_1 > 0$ denote eigenvalues of T, with λ_1 strictly positive because $S S^\top = F^\top F \succ 0$. The Arithmetic Mean-Geometric inequality brings:

$$\prod_2^d \lambda_k \leq \left(\frac{1}{d-1} \sum_{k=2}^d \lambda_k \right)^{d-1} \tag{3.102}$$

$$= \left(\frac{\operatorname{tr} T - \lambda_1}{d-1} \right)^{d-1} \tag{3.103}$$

$$= \left(\frac{d - \lambda_1}{d-1} \right)^{d-1} \tag{3.104}$$

$$\leq \left(\frac{d}{d-1} \right)^{d-1} \ . \tag{3.105}$$

Multiplying both side by λ_1 and rearranging yields:

$$\lambda_1 \geq \left(\frac{d-1}{d} \right)^{d-1} \det T \tag{3.106}$$

Let $\lambda_\circ > 0$ denote the minimal eigenvalue of $S S^\top$. It satisfies:

$$\lambda_\circ = \left(\frac{1}{d} \sum_i \|\boldsymbol{x}_i\|_2^2 \right) \lambda_1 \tag{3.107}$$

and thus it comes from inequality 3.106:

$$\lambda_\circ \geq \left(\frac{d-1}{d}\right)^{d-1} \left(\frac{d}{\sum_i \|\boldsymbol{x}_i\|_2^2}\right)^{d-1} \det SS^\top \tag{3.108}$$

$$= \left(\frac{d-1}{d}\right)^{d-1} \det \left[\left(\frac{d}{\sum_i \|\boldsymbol{x}_i\|_2^2}\right)^{1-\frac{1}{d}} SS^\top\right] \tag{3.109}$$

$$= \left(\frac{d-1}{d}\right)^{d-1} \det \overline{F}^\top \overline{F} \tag{3.110}$$

$$= \left(\frac{d-1}{d}\right)^{d-1} \mathrm{vol}^2(\overline{F}) \tag{3.111}$$

$$\geq \frac{1}{e}\mathrm{vol}^2(\overline{F}) \ . \tag{3.112}$$

We have used notation $\mathrm{vol}(\overline{F}) \doteq \sqrt{\det \overline{F}^\top \overline{F}}$. Since:

$$(\boldsymbol{\theta}^\star - \hat{\boldsymbol{\theta}})^\top SS^\top (\boldsymbol{\theta}^\star - \hat{\boldsymbol{\theta}}) \geq \lambda_\circ \left\|\boldsymbol{\theta}^\star - \hat{\boldsymbol{\theta}}\right\|_2^2 \ , \tag{3.113}$$

combining 3.101 with 3.112 yields the following lower bound on β:

$$\beta \geq \frac{2}{e}\gamma \ \mathrm{vol}^2(\overline{F}) \left\|\boldsymbol{\theta}^\star - \hat{\boldsymbol{\theta}}\right\|_2^2 \ . \tag{3.114}$$

Going back to 3.93, we get:

$$\lambda \left\|\boldsymbol{\theta}^\star - \hat{\boldsymbol{\theta}}\right\|_2^2 + a\left(\boldsymbol{\mu} - \hat{\boldsymbol{\mu}}\right)^\top \left(\boldsymbol{\theta}^\star - \hat{\boldsymbol{\theta}}\right) + \frac{1}{em}\gamma \ \mathrm{vol}^2(\overline{F}) \left\|\boldsymbol{\theta}^\star - \hat{\boldsymbol{\theta}}\right\|_2^2 \leq 0 \ . \tag{3.115}$$

Since $a\left(\boldsymbol{\mu} - \hat{\boldsymbol{\mu}}\right)^\top \left(\boldsymbol{\theta}^\star - \hat{\boldsymbol{\theta}}\right) \leq |a| \left\|\boldsymbol{\mu} - \hat{\boldsymbol{\mu}}\right\|_2 \left\|\boldsymbol{\theta}^\star - \hat{\boldsymbol{\theta}}\right\|_2$, after chaining the inequalities and solving for $\left\|\boldsymbol{\theta}^\star - \hat{\boldsymbol{\theta}}\right\|_2$ we get:

$$\left\|\boldsymbol{\theta}^\star - \hat{\boldsymbol{\theta}}\right\|_2 \leq \frac{|a|}{\lambda + \frac{1}{em}\gamma \mathrm{vol}^2(\overline{F})} \left\|\boldsymbol{\mu} - \hat{\boldsymbol{\mu}}\right\|_2 \ , \tag{3.116}$$

as claimed.

3.6 Appendix: additional formal results

3.6.1 Mean and covariance operators

The intuition behind the relevance of the mean operator becomes clear once we rewrite it as follows.

Lemma 28. *Let* $\boldsymbol{\mu} \doteq \mathbb{E}[y\boldsymbol{x}]$ *and* $\mathbb{C}\mathrm{ov}[\boldsymbol{x}, y] \doteq \mathbb{E}_{\mathcal{S}}\left[(y\boldsymbol{x} - \boldsymbol{\mu})^2\right]$, *respectively the mean and the*

covariance operators. Let $\pi_+ \doteq \mathbb{E}_{\mathcal{S}} 1\{y > 0\}$ *be the positive label proportion of* $\mathcal{S}$*. Then:*

$$\boldsymbol{\mu} = \mathbb{C}\text{ov}[\boldsymbol{x}, y] + (2\pi_+ - 1)\mathbb{E}_{\mathcal{S}}[\boldsymbol{x}] \ . \tag{3.117}$$

Moreover, if observations are centered ($\mathbb{E}_{\mathcal{S}}[\boldsymbol{x}] = 0$) *or the labels are balanced* ($\pi_+ = 1/2$)*, then* $\boldsymbol{\mu} = \mathbb{C}\text{ov}[\boldsymbol{x}, y]$*.*

Proof

$$\begin{aligned}
\mathbb{C}\text{ov}[\boldsymbol{x}, y] &= \mathbb{E}_{\mathcal{S}}[y\boldsymbol{x}] - \mathbb{E}_{\mathcal{S}}[y]\mathbb{E}_{\mathcal{S}}[\boldsymbol{x}] && (3.118)\\
&= \boldsymbol{\mu} - \left(\frac{1}{m} \sum_{i:y_i>0} 1 - \frac{1}{m} \sum_{i:y_i<0} 1 \right) \mathbb{E}_{\mathcal{S}}[\boldsymbol{x}] && (3.119)\\
&= \boldsymbol{\mu} - (2\pi_+ - 1)\, \mathbb{E}_{\mathcal{S}}[\boldsymbol{x}] \ . && (3.120)
\end{aligned}$$

The second statement follows immediately. ■

We have come to the unsurprising fact that — when observations are centered — the covariance $\mathbb{C}\text{ov}[\boldsymbol{x}, y]$ is what we need to know about the labels for learning linear models. The rest of the loss (in light of Factorization) may be seen as a data dependent regularizer.

3.6.2 The generality of factorization

Factorization is ubiquitous for any (margin) loss, beyond the theory seen so far. The decomposition on even and odd functions is actually all we need to factor ℓ.

Theorem 29 (Factorization)**.** *For any sample* $\mathcal{S}$ *and hypothesis* h *the empirical* ℓ*-risk can be written as:*

$$R_{\mathcal{S},\ell}(h) = \frac{1}{2}\mathbb{E}_{\mathcal{S}} \left[\sum_{\sigma \in \mathcal{Y}} \ell(\sigma h(\boldsymbol{x})) \right] + \mathbb{E}_{\mathcal{S}} \Big[\ell_o(y h(\boldsymbol{x})) \Big] \tag{3.121}$$

where $\ell_o(\cdot)$ *is odd and* $\ell_e(\cdot) \doteq \sum_{\sigma \in \mathcal{Y}} \ell(\sigma h(\cdot))$ *is even and both uniquely defined.*

Its range of validity is exemplified by 01 loss, a non-convex discontinuous piecewise linear function, which factors as:

$$\ell_e(x) = \left\{ \begin{array}{cc} \frac{1}{2} & x \neq 0 \\ 1 & \text{otherwise} \end{array} \right. , \ \ \ell_o(x) = -\frac{1}{2}\,\text{sign}(x) \ . \tag{3.122}$$

It follows immediately that $\mathbb{E}_{\mathcal{S}}[\ell_o(\cdot)]$ is sufficient for y. However, ℓ_o is a function of model $\boldsymbol{\theta}$. This defeats the purpose of a sufficient statistic, which we aim to be computable from data only and it is the main reason to define LOLS.

Moreover, Factorization goes beyond simple linear models, and can also be formulated for RKHS. To show that, notice that we satisfy all hypotheses of the Representer Theorem [Schölkopf and Smola, 2002].

Theorem 30. *Let $h(\boldsymbol{x}) : \mathcal{X} \to \mathcal{H}$ be a feature map into a Reproducing Kernel Hilbert Space (RKHS) $\mathcal{H}$ with symmetric positive definite kernel $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, such that $h : \boldsymbol{x} \to k(\cdot, \boldsymbol{x})$. For any learning sample $\mathcal{S}$, the empirical ℓ-risk $R_{\mathcal{S},\ell}(h)$ with $\Omega : ||h||_{\mathcal{H}} \to \mathbb{R}^+$ regularization can be written as:*

$$\frac{1}{2}\mathbb{E}_{\mathcal{S}}\left[\sum_{\sigma\in\mathcal{Y}} \ell(\sigma h(\boldsymbol{x}))\right] + \mathbb{E}_{\mathcal{S}}\Big[\ell_o(yh(\boldsymbol{x}))\Big] + \Omega(||h||_{\mathcal{H}}) \tag{3.123}$$

and the optimal hypothesis admits a representation of the form $h(\boldsymbol{x}) = \sum_{i\in[m]} \alpha_i k(\boldsymbol{x}, \boldsymbol{x}_i)$.

It follows that all results of this Chapter may be read in the context of non-parametric models, with the *kernel* mean operator as sufficient statistic.

Finally, we have formulated the Factorization Theorem for classification problems. We can show a similar statement for regression with square loss $\ell(\langle\boldsymbol{\theta}, \boldsymbol{x}_i\rangle, y) = (\langle\boldsymbol{\theta}, \boldsymbol{x}_i\rangle - y_i)^2$:

$$\mathbb{E}_{\mathcal{S}}[(\langle\boldsymbol{\theta}, \boldsymbol{x}\rangle - y)^2] = \mathbb{E}_{\mathcal{S}}\left[\langle\boldsymbol{\theta}, \boldsymbol{x}\rangle^2\right] + \mathbb{E}_{\mathcal{S}}\left[y^2\right] - 2\langle\boldsymbol{\theta}, \boldsymbol{\mu}\rangle \ . \tag{3.124}$$

Taking the minimizers on both sides we obtain:

$$\operatorname*{argmin}_{\boldsymbol{\theta}} \mathbb{E}_{\mathcal{S}}[(\langle\boldsymbol{\theta}, \boldsymbol{x}\rangle - y)^2] = \operatorname*{argmin}_{\boldsymbol{\theta}} \mathbb{E}_{\mathcal{S}}\left[\langle\boldsymbol{\theta}, \boldsymbol{x}\rangle^2\right] - 2\langle\boldsymbol{\theta}, \boldsymbol{\mu}\rangle \tag{3.125}$$

$$= \operatorname*{argmin}_{\boldsymbol{\theta}} \left\|X^\top\boldsymbol{\theta}\right\|_2^2 - 2\langle\boldsymbol{\theta}, \boldsymbol{\mu}\rangle \ . \tag{3.126}$$

This last result opens further applications yet, in the Thesis, we keep our focus on classification problems.

3.6.3 Factorization of non linear-odd losses

When ℓ_o is not linear, we can find upper bounds in the form of affine functions. It suffices to be continuous and have asymptotes at $\pm\infty$.

Lemma 31. *Let the loss ℓ be continuous. Suppose that it has asymptotes at $\pm\infty$,* i.e. *there exist $c_1, c_2 \in \mathbb{R}$ and $d_1, d_2 \in \mathbb{R}$ such that:*

$$\lim_{x\to+\infty} \ell(x) - c_1 x - d_1 = 0, \ \lim_{x\to-\infty} \ell(x) - c_2 x - d_2 = 0 \tag{3.127}$$

then there exists $q \in \mathbb{R}$ such that: $\ell_o(x) \leq \frac{c_1+c_2}{2}x + q$.

Proof One can compute the limits at infinity of ℓ_o to get:

$$\lim_{x\to+\infty} \ell_o(x) - \frac{c_1 + c_2}{2}x = \frac{d_1 - d_2}{2} \tag{3.128}$$

and:

$$\lim_{x\to-\infty} \ell_o(x) - \frac{c_1 + c_2}{2}x = \frac{d_2 - d_1}{2} \ . \tag{3.129}$$

Then $q \doteq \sup\{\ell_o(x) - \frac{c_1+c_2}{2}x\} < +\infty$ as ℓ_o is continuous. Thus $\ell_o(x) - \frac{c_1+c_2}{2}x \leq q$. ■

The Lemma covers many cases of practical interest outside the class of LOLS, *e.g.* hinge, absolute and Huber losses. Exponential loss is the exception since $\ell_o(x) = -\sinh(x)$ cannot be bounded. Consider for instance hinge loss: $\ell(x) = [1-x]_+$ is not differentiable in 1, however it is continuous with asymptotes at $\pm\infty$. Therefore, for any $\boldsymbol{\theta}$ its empirical risk is bounded as:

$$R_{\mathcal{S},hinge}(\boldsymbol{\theta}) \leq \frac{1}{2}R_{\mathcal{S}_{2x},hinge}(\boldsymbol{\theta}) - \frac{1}{2}\langle\boldsymbol{\theta},\boldsymbol{\mu}\rangle + q \ , \tag{3.130}$$

since $c_1 = 0$ and $c_2 = 1$. An alternative proof of this result on hinge is provided next, giving the exact value of $q = 1/2$. The odd term for hinge loss is:

$$\ell_o(x) = \frac{1}{2}\left([1-x]_+ - [1+x]_+\right) \tag{3.131}$$

$$= \frac{1}{4}\left(-2x + |1-x| - |1+x|\right) \tag{3.132}$$

due to an arithmetic trick for the max function:

$$\max(a,b) = (a+b)/2 + |b-a|/2 \ . \tag{3.133}$$

Then for any x:

$$|1-x| \leq |x| + 1, \tag{3.134}$$

$$|1+x| \geq |x| - 1 \tag{3.135}$$

and therefore:

$$\ell_o(x) \leq \frac{1}{4}(-2x + |x| + 1 - |x| + 1) = \frac{1}{2}(1-x) \ . \tag{3.136}$$

We also provide a "if-and-only-if" version of Lemma 31 fully characterizing which losses can be upper bounded by a LOL.

Lemma 32. *Let $\ell : \mathbb{R} \to \mathbb{R}$ a continuous function. Then there exists $c_1, d_1, d_2 \in \mathbb{R}$ such that:*

$$\limsup_{x\to+\infty} \ell_o(x) - c_1 x - d_1 = 0 \tag{3.137}$$

and:

$$\limsup_{x\to-\infty} \ell_o(x) - c_1 x - d_2 = 0 \ , \tag{3.138}$$

if and only if there exists $q, q' \in \mathbb{R}$ such that $\ell_o(x) \leq q'x + q$ for every $x \in \mathbb{R}$.

Proof

$\Rightarrow$) Suppose that such limits exist and they are zero for some c_1, d_1, d_2. Let prove

that ℓ_o is bounded from above by a line. It holds that:

$$q = \sup_{x \in \mathbb{R}} \{\ell_o(x) - c_1 x\} < \infty \ , \tag{3.139}$$

because ℓ_o is continuous. So for every $x \in \mathbb{R}$:

$$\ell_o(x) \leq c_1 x + q \ . \tag{3.140}$$

In particular we can take c_1 as the angular coefficient of the line.

$\Leftarrow$) Vice versa we proceed by contradiction. Suppose that there exists $q, q' \in \mathbb{R}$ such that ℓ_o is bounded from above by $\ell(x) = q'x + q$. Suppose in addition that the conditions on the asymptotes (3.137) and (3.138) are false. This implies either the existence of a sequence $x_n \to +\infty$ such that:

$$\lim_{n \to \infty} \ell_o(x_n) - q' x_n \to \pm\infty \ , \tag{3.141}$$

or the existence of another sequence $x'_n \to -\infty$:

$$\lim_{n \to \infty} \ell_o(y_n) - q' x'_n \to \pm\infty \ . \tag{3.142}$$

On one hand, if at least one of these two limits is $+\infty$ then we already reach a contradiction, because $\ell_o(x)$ is supposed to be bounded from above by $\ell(x) = q'x + q$. Suppose on the other hand that $x_n \to +\infty$ is such that:

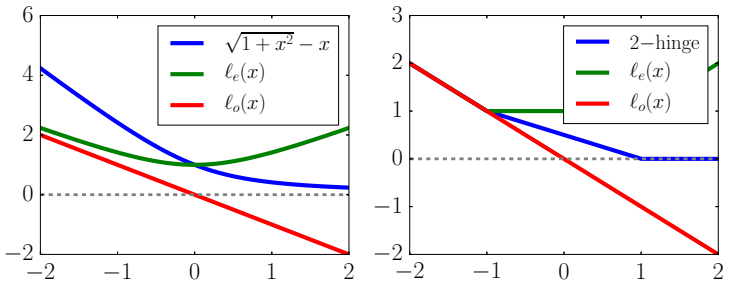
$$\lim_{n \to +\infty} \ell_o(x_n) - q' x_n \to -\infty \ . \tag{3.143}$$

Then defining $x'_n = -x_n$ we have:

$$\lim_{n \to +\infty} \ell_o(w_n) - m x'_n \to +\infty \ , \tag{3.144}$$

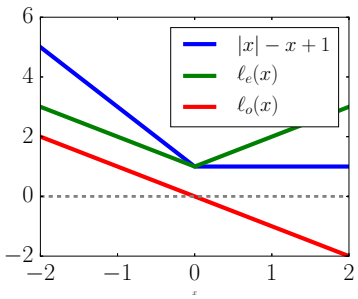
and for the same reason as above we reach a contradiction. ■

3.6.4 More graphs on linear and non-linear-odd losses

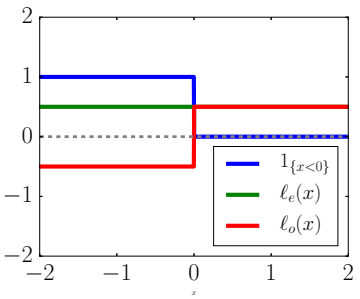


(a) Matsushita loss

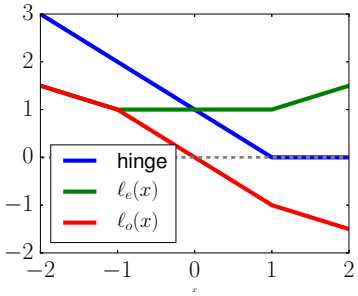
(b) double-hinge loss



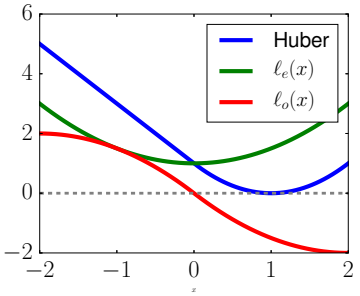
(c) ρ-loss, ρ=1



(d) 0-1 loss



(e) hinge loss



(f) Huber loss

Figure 3.2: Linear-odd: Matsushita, double-hinge, ρ-loss. Non-linear-odd: 0-1, hinge, Huber.

3.6.5 The linear-odd losses of du Plessis et al. [2015]

The work of du Plessis et al. [2015] shows that a linear-odd condition on a *convex* ℓ allows one to derive a tractable, *i.e.* still convex, loss for learning with *positive and unlabeled data*. The approach is similar to ours as it isolates a label-free term in the loss, with the goal of leveraging on the unlabeled examples too. Their manipulation of the loss *is not* equivalent to Theorem 18 though, as we explain here. Beside that, since we reason at the higher level of weakly supervised learning, we may frame

a solution for this setting by calling $\boldsymbol{\mu}$SGD on $\hat{\boldsymbol{\mu}}$ defined above or by building on estimators derived from geometrical ideas discussed in Chapter 4.

For the sake of completeness, we review the use of convex LOLS in the cited work. Let $\pi_+ \doteq \mathbb{P}(y=1)$ and let $\mathcal{D}_+$ and $\mathcal{D}_-$ respectively the set of positive and negative examples in $\mathcal{D}$. Consider first:

$$\mathbb{E}_{(\boldsymbol{x},\cdot)\sim\mathcal{D}}\left[\ell(-\langle\boldsymbol{\theta},\boldsymbol{x}\rangle)\right] \tag{3.145}$$

$$= \pi_+\mathbb{E}_{(\boldsymbol{x},\cdot)\sim\mathcal{D}_+}\left[\ell(-\langle\boldsymbol{\theta},\boldsymbol{x}\rangle)\right] + (1-\pi_+)\mathbb{E}_{(\boldsymbol{x},\cdot)\sim\mathcal{D}_-}\left[\ell(-\langle\boldsymbol{\theta},\boldsymbol{x}\rangle)\right] \tag{3.146}$$

Then, it is also true that:

$$\mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}\left[\ell(y\langle\boldsymbol{\theta},\boldsymbol{x}\rangle)\right] \tag{3.147}$$

$$= \pi_+\mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}_+}\left[\ell(y\langle\boldsymbol{\theta},\boldsymbol{x}\rangle)\right] + (1-\pi_+)\mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}_-}\left[\ell(y\langle\boldsymbol{\theta},\boldsymbol{x}\rangle)\right] \ . \tag{3.148}$$

Solve Equation 3.145 for:

$$(1-\pi_+)\mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}_-}\left[\ell(y\langle\boldsymbol{\theta},\boldsymbol{x}\rangle)\right] = (1-\pi_+)\mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}_-}\left[-\ell(-\langle\boldsymbol{\theta},\boldsymbol{x}\rangle)\right] \tag{3.149}$$

and substitute it into Equation 3.147 so as to obtain:

$$\mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}\left[\ell(y\langle\boldsymbol{\theta},\boldsymbol{x}\rangle)\right] \tag{3.150}$$

$$= \pi_+\mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}_+}\left[\ell(y\langle\boldsymbol{\theta},\boldsymbol{x}\rangle)\right] + \mathbb{E}_{(\boldsymbol{x},\cdot)\sim\mathcal{D}}\left[\ell(-\langle\boldsymbol{\theta},\boldsymbol{x}\rangle)\right] - \pi_+\mathbb{E}_{(\boldsymbol{x},\cdot)\sim\mathcal{D}_+}\left[\ell(-\langle\boldsymbol{\theta},\boldsymbol{x}\rangle)\right] \tag{3.151}$$

$$= \pi_+\left(\mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}_+}\left[\ell(+\langle\boldsymbol{\theta},\boldsymbol{x}\rangle)\right] - \mathbb{E}_{(\boldsymbol{x},\cdot)\sim\mathcal{D}_+}\left[\ell(-\langle\boldsymbol{\theta},\boldsymbol{x}\rangle)\right]\right) + \mathbb{E}_{(\boldsymbol{x},\cdot)\sim\mathcal{D}}\left[\ell(-\langle\boldsymbol{\theta},\boldsymbol{x}\rangle)\right] \tag{3.152}$$

$$= \frac{\pi_+}{2}\,\mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}_+}\left[\ell_o(+\langle\boldsymbol{\theta},\boldsymbol{x}\rangle)\right] + \mathbb{E}_{(\boldsymbol{x},\cdot)\sim\mathcal{D}}\left[\ell(-\langle\boldsymbol{\theta},\boldsymbol{x}\rangle)\right] \ , \tag{3.153}$$

by our usual definition of $\ell_o(x) = \frac{1}{2}(\ell(x) - \ell(-x))$. Recall that one of the goals of the authors is to conserve the convexity of this new crafted loss function. Then, du Plessis et al. [2015, Theorem 1] proceed stating that when ℓ_o is convex, it must also be linear. And therefore they must focus on LOLS. Theorem 1 of du Plessis et al. [2015] is immediate from the point of view of our theory: in fact, an odd function can be convex or concave only if it also linear. The resulting expression based on the fact $\ell(x) - \ell(-x) = 2ax$ simplifies into:

$$\mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}\left[\ell(y\langle\boldsymbol{\theta},\boldsymbol{x}\rangle)\right] = a\pi_+\mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}_+}\left[y\langle\boldsymbol{\theta},\boldsymbol{x}\rangle\right] + \mathbb{E}_{(\boldsymbol{x},\cdot)\sim\mathcal{D}}\left[\ell(-\langle\boldsymbol{\theta},\boldsymbol{x}\rangle)\right] \tag{3.154}$$

$$= a\pi_+\boldsymbol{\mu}_{\mathcal{D}_+} + \mathbb{E}_{(\boldsymbol{x},\cdot)\sim\mathcal{D}}\left[\ell(-\langle\boldsymbol{\theta},\boldsymbol{x}\rangle)\right] \ . \tag{3.155}$$

where $\boldsymbol{\mu}_{\mathcal{D}_+}$ is a mean operator computed on positive examples only. Notice how the second term is instead label independent, although it is not an even function as in the Factorization Theorem.

3.7 References

The name *mean operator* in this context was originally used by Quadrianto et al. [2009]. In the theory of Hilbert space embedding, its importance is due to the injectivity of the map — under conditions on the kernel — which is used in applications such as two-sample and independence tests, feature extraction and covariate shift [Smola et al., 2007].

Factorization of logistic loss has been derived in other contexts in literature [Jaakkola and Jordan, 2000; Gao et al., 2016]. The work of Zantedeschi et al. [2016] implicitly utilizes Loss Factorization for formulating the β-risk, an augmented version of surrogate risks incorporating a confidence parameter for each individual label.

In most of this Chapter, we kept the lighter notation of linear classifiers, but nothing should prevent the extension of our results to non-parametric models, exchanging $\boldsymbol{x}$ with an implicit feature map $h(\boldsymbol{x})$ recalling Theorem 30. A kernelized version of the mean-covariance operators Lemma 28 is given in Song et al. [2009]. A version of Theorem 23 may be derived for RKHS on top of Bartlett and Mendelson [2002]; Kakade et al. [2009].

3.7.1 The two-step procedure of Raghunathan et al. [2016]

The publication of our Patrini et al. [2016a] happened concurrently with Raghunathan et al. [2016], which shares many insights with our methodology. The authors design the same two-step procedure that amounts to sufficient statistic estimation followed by convex optimization. In fact, the underlying motivation is to avoid non-convex optimization and EM when labels are only known by indirect supervision. The learning framework of choice is MLE of graphical models, allowing an easy extension of the approach to structured label spaces — this is, in particular, an aspect not touched by our Thesis.

The two examples of weak supervision focus of Raghunathan et al. [2016] are close to what we consider in the next Chapters. The scenario of *learning under local privacy* can be mapped to the one of noisy labels; in particular, label corruption is dictated by a randomization schema that achieves differential privacy. The label sufficient statistic is estimated in a way that resembles Equation 5.2 in Chapter 5, with the same guarantee of unbiasedness. The setting of *learning with lightweight annotation* is a form of aggregated-level supervision, although in the context of structured prediction. The sufficient statistic is obtained by solving a linear system, in a similar fashion of Equation 4.7 and followers in Chapter 4.

Finally, Raghunathan et al. [2016] elaborate on the statistical efficiency of the proposed estimators and respective models, when compared to a model fitted by MLE. The two-step procedure is also studied by a geometrical viewpoint. The problem is mapped to the minimization of a KL divergence on an exponential family defined over a supervision function of y, not y itself, which makes the problem of MLE non-convex. It is shown that the two-step approach first optimizes over a relaxed set — the sufficient statistic estimation — and then projects onto the space of the exponen-

tial family, therefore bypassing the non-convex objective by an approximate solution.

3.7.2 Learning reductions

We have described the two-step algorithm as a procedure for *casting* a weakly supervised learning problem into an optimization problem, that is equivalent to a fully supervised problem. This is achieved via the equivalence expressed by Factorization. Solving a Machine Learning problem by solutions to other learning problems is a *learning reduction* [Beygelzimer et al., 2015]. Our work can be interpreted as such.

Following Beygelzimer et al. [2005], we define a supervised classification task as a triple $(\mathcal{K}, \mathcal{Y}, \ell)$, with supervised advice $\mathcal{K}$, predictions space $\mathcal{Y}$ and loss ℓ. A learning reduction is a procedure that casts a task $(\mathcal{K}, \mathcal{Y}, \ell)$ into a simpler "reduced' task $(\mathcal{K}', \mathcal{Y}', \ell')$. We assume that ℓ is linear-odd and K represents any sort of weak supervision. Then, our two-step procedure is a reduction to binary classification $(\mathcal{Y}, \mathcal{Y}, \ell)$. The reduction of Meta algorithm 1 is somehow simple, in the sense that $\mathcal{Y}$ does not change *and neither does* ℓ. Algorithms 2 and 3 are examples of learning reductions from any weakly supervised classification problem to binary classification. Yet we actually modify the internal code of the "reduced learner", the binary classifier, which contrasts with the concept of reduction. However, take as example Algorithm 2. We could as well write subgradients step as

$$\frac{1}{2}\left(\partial\ell(\langle\boldsymbol{\theta}^t, \boldsymbol{x}_i\rangle) + \partial\ell(-\langle\boldsymbol{\theta}^t, \boldsymbol{x}_i\rangle)\right) + a\boldsymbol{\mu} \ , \tag{3.156}$$

which equals $\partial\ell$, and thus the fully supervised SGD would be effectively untouched.

Chapter 4

Learning from label proportions

This Chapter is devoted to the problem of learning from label proportions. Loss Factorization is put to use by showing that we can learn from label proportions only via the estimation of the label sufficient statistic. Several formal guarantees are provided along with two practical algorithms which fill the gap left open the in last Chapter. In particular, we build on the Mean Map of [Quadrianto et al., 2009], relaxing its assumptions and yet proving finite sample guarantees for the estimation of the sufficient statistic. We also introduce a generalization of Rademacher complexity that is more meaningful in this learning setting. We show that our approach is particularly effective by extensive experiments simulating the construction of the instance groups and relative label proportions. The performance success is measured both against prior work and to the closeness to a fully supervised "Oracle".

4.1 Motivation

We are interested in learning a binary classifier with supervision at the level of groups of observations, called *bags*. The type of information we assume available is the *label proportions* per bag, indicating the fraction of positive binary labels of its observations. We refer to this framework as *learning from label proportions* (LLP) [Quadrianto et al., 2009; Yu et al., 2014b]. Several applications fit the LLP abstraction:

(i) Only aggregated labels can be obtained due to the physical limits of measurement tools or constraints of communication and storage, as in particle mass spectrometry and high energy physics [Chen et al., 2006; Musicant et al., 2007], quality control in metallurgy [Stolpe and Morik, 2011] and embryos implantation for human assisted reproduction [Hernández-González et al., 2016]. Similarly, in sentiment analysis positive/negative rates are given to whole documents, but we may be interested in attributing the sentiment to individual sentences [Kotzias et al., 2015].

(ii) Labels existed once but they are now given in an aggregated fashion for privacy-preserving reasons, as in medical databases [Bhowmik et al., 2015], fraud detection [Rüping, 2010], banking [Ma et al., 2016], house price market, election results, census data, etc.

(iii) The problem is semi- or unsupervised but it is simple to obtain additional supervision for the unlabeled observations in the form of expectation from other data source or by domain experts [Quadrianto et al., 2009; Liang et al., 2009]. This is sometimes called *distant supervision*, where individual labels are not only unknown but also virtually never measured or recorded. Supervision is given by pairing two or more data domains: on the one side, the individual feature vectors and, on the other side, aggregated variables from an additional related source of information. For instance, Twitter data can be paired with constraints from county demographics, trends in first names and exemplar Twitter accounts strongly associated with a class label [Mohammady and Culotta, 2014; Ardehaly and Culotta, 2015]. To give another example, advertising experts can estimate the rate of successful of ads at a campaign level, while the learning problem concerns the effectiveness of each individual banner and links shown to users [Wager et al., 2015]. Aggregate labels are frequently computed in terms of ratios or percents, *i.e.* proportions.

(iv) In Computer Vision applications, supervision is frequently only available to a higher level than the one interesting for prediction. For example in object detection, image segmentation and region recognition, a predictor should classify image patches or pixels, but those are seldom labeled in annotated image datasets [Kuck and de Freitas, 2005; Chen et al., 2009]. The problem of detecting events in videos by classifying singular frames is characterized by the same kind of weak supervision [Lai et al., 2014]. An example is the case of predicting per-pixel ice concentration on satellite images, when only region-level concentration is available from experts annotation [Li and Taylor, 2015]. Methods for LLP have also been used for learning "human nameable properties" of images [Yu et al., 2014a].

4.2 Learning setting

In learning from label proportions, we do not observe directly $\mathcal{S}$ but $\mathcal{S}_X$, which denotes the learning sample without any label. We are given its partition in $n > 1$ bags, that is $\mathcal{S}_X = \bigcup_j \mathcal{S}_j, j \in [n]$ and $\forall j \neq j', \mathcal{S}_j \cap \mathcal{S}_j' = \emptyset$. This is by no means restrictive: the case when bags constitute a *cover* instead of a partition of $\mathcal{S}_X$ can be reduced to the defined setting by copying examples among bags and by reweighing the loss accordingly. The "bag assignment function" that partitions $\mathcal{S}$ is unknown but fixed. We do not assume any knowledge about how observations have been partitioned, in spite of the fact that it is easy to imagine several applications for which something about the process is known. Each bag j is provided with its respective *label proportion*:

$$\pi_j \doteq \mathbb{E}_{\mathcal{S}_j}[y = 1] \tag{4.1}$$

and bag proportions $p_j \doteq m_j/m$ with $m_j = |\mathcal{S}_j|$. For the sake of clarity, in this Chapter we denote use i, j and k to refer respectively to examples, bags and features.

loss name	$\ell(x)$	$-\phi(x)$
logistic loss	$\log(1+\exp(-x))$	$-x\log x-(1-x)\log(1-x)$
square loss	$(1-x)^2$	$x(1-x)$
Matsushita loss	$-x+\sqrt{1+x^2}$	$\sqrt{x(1-x)}$

Table 4.1: Correspondence between permissible function ϕ and loss ℓ.

$\mathcal{H}$ is be the space of linear classifiers — for a kernelized version of our algorithms we point to the formulation of Quadrianto et al. [2009] that constitutes the basis of our algorithmic contributions.

4.2.1 Symmetric proper losses

We briefly discuss a particular class of losses that are called *symmetric proper losses* (SPL). While not necessary for the statement of the algorithms, some theoretical guarantees derived in this Chapter assume to work with the convenient functional shape of SPLs. Conceptually, we are not moving away from the framework presented in Chapter 3, since the class linear-odd losses has been shown to strictly include the one defined here. Symmetric proper losses are axiomatized in Nock and Nielsen [2009].

Definition 33 (Symmetric proper losses)**.** *A loss $\ell(y,h(\boldsymbol{x}))$ is called* symmetric proper *(SPL) if it is:*

(i) lower bounded
(ii) proper
(iii) symmetric, that is, $\ell(y,h(\boldsymbol{x}))=\ell(-y,-h(\boldsymbol{x}))$
(iv) twice differentiable

Notice that since we consider margin losses (Definition 2), condition (iii) is always satisfied. There exists a bijection between the set of symmetric proper losses and a set of *permissible* functions [Kearns and Mansour, 1996] ϕ which are differentiable, strictly convex, symmetric about $1/2$ and with $\mathrm{dom}(\phi)\supseteq[0,1]$. We also let $b_\phi \doteq \phi(1/2)-a_\phi>0$ and $\phi^\star$ be the convex conjugate of ϕ. The bijection is the property that we need, formalized in the next Lemma from Nock and Nielsen [2009].

Lemma 34. *A function ℓ is a* symmetric proper loss *if and only if there exists a* permissible *function ϕ such that:*

$$\ell(x)=a_\phi+\frac{\phi^\star(-x)}{b_\phi}\ . \tag{4.2}$$

This representation of SPL, which is originally obtained through Bregman divergences in Nock and Nielsen [2009], is sufficient to prove that SPLs factor the same way as linear-odd losses; we have shown that in Chapter 3. The original, more involved proof of Factorization for SPL is contained Patrini et al. [2014], Lemma 1.

4.3 Estimating the sufficient statistic

In a direct implementation of our two-step framework (Meta Algorithm 1), the estimation of the mean operator appears to be the learning bottleneck for LLP. The mean operator is in fact unknown since we cannot recover it simply from the label proportions. We start by expanding the mean operator in its bag-wise label-wise components:

$$\begin{aligned}
\boldsymbol{\mu} &= \mathbb{E}_{\mathcal{S}}\left[y\boldsymbol{x}\right] && (4.3)\\
&= \sum_{j=1}^{n} p_j \, \mathbb{E}_{\mathcal{S}_j}[y\boldsymbol{x}] && (4.4)\\
&= \sum_{j=1}^{n} p_j \left(\mathbb{E}_{\mathcal{S}_j}[\boldsymbol{x}, y=1] + \mathbb{E}_{\mathcal{S}_j}[\boldsymbol{x}, y=-1]\right) && (4.5)\\
&= \sum_{j=1}^{n} p_j \left(\pi_j \, \mathbb{E}_{\mathcal{S}_j}[\boldsymbol{x}|y=1] - (1-\pi_j) \, \mathbb{E}_{\mathcal{S}_j}[\boldsymbol{x}|y=-1]\right) \ . && (4.6)
\end{aligned}$$

The quantities p_j and π_j are known. So our problem is turned into the estimation of $2n$ vectors of unknowns $\mathbb{E}_{\mathcal{S}_j}[\boldsymbol{x}|y] \in \mathbb{R}^d$. Those are the average feature vector — the center — for each bag conditioned to the label sign. How can we use the information available to compute those centers? We come up with a linear system of equations of which they are the only unknowns. By law of total probability:

$$\mathbb{E}_{\mathcal{S}}[\boldsymbol{x}] = \sum_{y \in \mathcal{Y}} \pi_j \, \mathbb{E}_{\mathcal{S}_j}[\boldsymbol{x}|y] \ , \tag{4.7}$$

where every $\mathbb{E}_{\mathcal{S}_j}[\boldsymbol{x}]$ can be computed even without label knowledge as they are simply the bag-wise average feature vectors. For convenience, we rewrite the system in matrix form. Let $\boldsymbol{b}_j^y = \mathbb{E}_{\mathcal{S}_j}[\boldsymbol{x}|y]$ and $\boldsymbol{b}_j = \mathbb{E}_{\mathcal{S}_j}[\boldsymbol{x}]$. The $2n$ $\boldsymbol{b}_j^y$s are solutions of:

$$B - \Pi^\top B^{\pm} = \mathbf{0} \ , \tag{4.8}$$

where $B \doteq [\boldsymbol{b}_1|\boldsymbol{b}_2|...|\boldsymbol{b}_n]^\top \in \mathbb{R}^{n \times d}$, $\Pi \doteq [\operatorname{diag}(\boldsymbol{\pi})|\operatorname{diag}(\mathbf{1}-\boldsymbol{\pi})]^\top \in \mathbb{R}^{2n \times n}$ and $\boldsymbol{\pi}$ stacks together all label proportions in a vector. $B^{\pm} \in \mathbb{R}^{2n \times d}$ is the matrix of unknowns:

$$B^{\pm} \doteq \Big[\underbrace{\boldsymbol{b}_1^{+1}|\boldsymbol{b}_2^{+1}|...|\boldsymbol{b}_n^{+1}}_{(B^+)^\top} \Big| \underbrace{\boldsymbol{b}_1^{-1}|\boldsymbol{b}_2^{-1}|...|\boldsymbol{b}_n^{-1}}_{(B^-)^\top}\Big]^\top \ . \tag{4.9}$$

System 4.8 is under-determined, as it is made of $n \times d$ equations on $2n \times d$ unknowns. This fact should not be a surprise as it expresses the ill-posed nature of the problem of learning a classifier with label proportions only. To solve the system we need to resort to additional assumptions. We remark that solving System 4.8 accounts for the first estimation step, followed by Equation 4.3 to recover the mean operator. We now focus on solving the linear system.

Algorithm 4: Mean Map (MM)

Input: $\mathcal{S}_j, \pi_j, p_j, \forall j \in [n]$, ℓ is a-LOL, $\lambda > 0$
$\hat{B}^{\pm} \leftarrow \Pi^{\dagger} B$
$\hat{\boldsymbol{\mu}} \leftarrow \sum_{y \in \mathcal{Y}} y p(y) \hat{\boldsymbol{b}}^y$
$\hat{\boldsymbol{\theta}} \leftarrow \operatorname{argmin}_{\boldsymbol{\theta}} \frac{1}{2m} \sum_{i=1}^{m} \sum_{y \in \mathcal{Y}} \ell(y \langle \boldsymbol{\theta}, \boldsymbol{x}_i \rangle) + a \langle \boldsymbol{\theta}, \hat{\boldsymbol{\mu}} \rangle + \lambda \|\boldsymbol{\theta}\|_2^2$
Output: $\hat{\boldsymbol{\theta}}$

4.4 Mean Map algorithm of Quadrianto et al. [2009]

Quadrianto et al. [2009] originally propose this framework for LLP. It enforces a *homogeneity assumption*, that is a statement of conditional independence for j:

(**A4.0**) $\forall j,\ \mathbb{E}_{\mathcal{S}_j}[\boldsymbol{x}|y] = \mathbb{E}_{\mathcal{S}}[\boldsymbol{x}|y]$.

Under this hypothesis, the number of unknowns falls to $2d$. This is enough to obtain a system that is not under-determined, granted that $n \geq 2$. In this case, the linear system is simplified by defining $B^{\pm} \doteq [\boldsymbol{b}^{+1}, \boldsymbol{b}^{-1}]^{\top} \in \mathbb{R}^{n \times d}$ and $\Pi \doteq [\boldsymbol{\pi} | \mathbf{1} - \boldsymbol{\pi}] \in \mathbb{R}^{n \times 2}$, and the solution is found by pseudo-inversion as $\hat{B}^{\pm} = \Pi^{\dagger} B$. Let $\hat{\boldsymbol{b}}^y$ denote the rows of $\hat{B}^{\pm}$ in lieu of the true $\boldsymbol{b}^y = \mathbb{E}_{\mathcal{S}}[\boldsymbol{x}|y]$. The mean operator is then estimated by:

$$\hat{\boldsymbol{\mu}} = \sum_{y \in \mathcal{Y}} y p(y) \hat{\boldsymbol{b}}^y \ , \tag{4.10}$$

where $p(y)$ is the probability of the label over the whole $\mathcal{S}$ and can be derived from $\boldsymbol{\pi}$. This is the effectively the first example of implementation of the Meta Algorithm 1. We state it in Algorithm 4. The last step of convex optimization is expressed with L_2 regularization.

4.5 Laplacian Mean Map

Our proposal, the *Laplacian Mean Map* (LMM) algorithm, aims to solve System 4.8 at a larger extent. In order to do that, we relax the homogeneity assumption as:

(**A4.1**) $\forall j, j'$ if $j \approx j'$ then $\mathbb{E}_{\mathcal{S}_j}[\boldsymbol{x}|y] \approx \mathbb{E}_{\mathcal{S}_{j'}}[\boldsymbol{x}|y]$.

That is, instead of setting every $\mathbb{E}_{\mathcal{S}_j}[\boldsymbol{x}|y]$ equal for each bag j for a given y, we assume them close when bags are similar, while none is constrained to be equal. The similarity between bags j and j' is a domain-specific parameter of the algorithm and we will refer to it by $v_{j,j'} \geq 0$; we discuss the choice of v below. To incorporate the assumption into the estimation, we solve System 4.8 by least square minimization

with regularization as:

$$\underset{\boldsymbol{b}_j^1, \boldsymbol{b}_j^{-1}}{\operatorname{argmin}} \sum_j \left(\boldsymbol{b}_j - \pi_j \boldsymbol{b}_j^1 + (1 - \pi_j) \boldsymbol{b}_j^{-1} \right)^2 + \gamma \sum_{j,j'} v_{j,j'} \left[\left(\boldsymbol{b}_j^1 - \boldsymbol{b}_{j'}^1 \right)^2 + \left(\boldsymbol{b}_j^{-1} - \boldsymbol{b}_{j'}^{-1} \right)^2 \right] . \tag{4.11}$$

Depending on the regularization strength $\gamma \geq 0$, the more the bags are similar — larger $v_{j,j'}$ — the more their relative estimate of $\boldsymbol{b}_j^{\pm}$ are close to each other. We can restate the problem in matrix form by the Laplacian of the symmetric matrix $V \in \mathbb{R}^{n \times n}$, which is the adjacency matrix of the graph induced by the similarity $v_{j,j'}$. The Laplacian is defined as $L_a = D - V$, and D is a diagonal matrix such that $D_j = \sum_{j'}^{n} v_{j,j'}$. For any vector $\boldsymbol{u} \in \mathbb{R}^n$, it holds that:

$$\begin{aligned} \boldsymbol{u}^\top L_a \boldsymbol{u} &= \boldsymbol{u}^\top D \boldsymbol{u} - \boldsymbol{u}^\top V \boldsymbol{u} && (4.12) \\ &= \sum_j D_j u_j^2 - \sum_{j,j'} v_{j,j'} u_j u_{j'} && (4.13) \\ &= \frac{1}{2} \left(\sum_j D_j u_j^2 - 2 \sum_{j,j'} v_{j,j'} u_j u_{j'} + \sum_{j'} D_{j'} u_{j'}^2 \right) && (4.14) \\ &= \sum_{j,j'} v_{j,j'} \left(u_j - u_{j'} \right)^2 && (4.15) \end{aligned}$$

Expression 4.15 is obtained by applying the definition of D_j. This is a standard result in Spectral Graph Theory [Von Luxburg, 2007]. We make use of this equivalence for both $\boldsymbol{b}_j^1$ and $\boldsymbol{b}_j^{-1}$, but separately, as the two vectors are subject to distinct constraints. Thanks to the structure of matrix $B^{\pm}$, we can define:

$$L \doteq \varepsilon I + \left[\begin{array}{c|c} L_a & \mathbf{0} \\ \mathbf{0} & L_a \end{array} \right] \in \mathbb{R}^{2n \times 2n} , \tag{4.16}$$

such that[1]

$$\hat{B}^{\pm} = \underset{X \in \mathbb{R}^{2n \times d}}{\operatorname{argmin}} \ \operatorname{tr} \left(B - \Pi^\top X \right)^\top D_{\boldsymbol{w}} \left(B - \Pi^\top X \right) + \gamma \ \operatorname{tr} X^\top L X . \tag{4.17}$$

$D_{\boldsymbol{w}} \doteq \operatorname{diag}(\boldsymbol{w})$ is a user-fixed bias matrix with non-negative element on the diagonal $\boldsymbol{w}$. For example, we can re-weight the importance of the linear equations by the size of the respective bags m_j, by $\boldsymbol{w} = \boldsymbol{p}$. The second term has the form of a manifold regularizer of Belkin et al. [2006], which allows us to reinterpret our assumption from a geometrical standpoint: the bag label-wise feature averages $\mathbb{E}_{\mathcal{S}_j}[\boldsymbol{x}|y]$ live on a low-dimensional manifold parameterized by the unidimensional similarity function $v_{j,j'}$. The Laplacian matrix L_a is an empirical, discrete approximation of the manifold.

[1] with $\epsilon > 0$ to assure non-singularity and numerical stability, see the proof of Theorem 35 in Appendix 4.11.2.

Algorithm 5: Laplacian Mean Map (LMM)

Input: $\mathcal{S}_j, \pi_j, p_j, \forall j \in [n]$, ℓ is a-LOL, $\gamma > 0$, $D_{\boldsymbol{w}}$, L, $\lambda > 0$

$\hat{B}^{\pm} \leftarrow \left(\Pi D_{\boldsymbol{w}} \Pi^\top + \gamma L\right)^{-1} \Pi D_{\boldsymbol{w}} B$

$\hat{\boldsymbol{\mu}} \leftarrow \frac{1}{m} \sum_{j=1}^{n} p_j (\pi_j \hat{\boldsymbol{b}}_j^1 - (1-\pi_j)\hat{\boldsymbol{b}}_j^{-1})$

$\hat{\boldsymbol{\theta}} \leftarrow \operatorname{argmin}_{\boldsymbol{\theta}} \frac{1}{2m} \sum_{i=1}^{m} \sum_{y \in \mathcal{Y}} \ell(y\langle \boldsymbol{\theta}, \boldsymbol{x}_i \rangle) + a\langle \boldsymbol{\theta}, \hat{\boldsymbol{\mu}} \rangle + \lambda \|\boldsymbol{\theta}\|_2^2$

Output: $\hat{\boldsymbol{\theta}}$

The size of the Laplacian is $O(n^2)$, which is small compared to $O(m^2)$ if there are not many bags. This is in contrast with traditional approaches for semi-supervised learning, where the Laplacian matrix is formed on top of similarity between examples, instead of bags [Belkin et al., 2006].

Problem 4.17 admits global optimum in closed form.

Theorem 35. *The solution to Problem 4.17*

$$\operatorname*{argmin}_{X \in \mathbb{R}^{2n \times d}} \operatorname{tr}\left(B - \Pi^\top X\right)^\top D_{\boldsymbol{w}} \left(B - \Pi^\top X\right) + \gamma \operatorname{tr} X^\top L X \tag{4.18}$$

is

$$\hat{B}^{\pm} \doteq \left(\Pi D_{\boldsymbol{w}} \Pi^\top + \gamma L\right)^{-1} \Pi D_{\boldsymbol{w}} B \tag{4.19}$$

Proof in 4.11.1. This result explains the role of the penalty εI in 4.16 as $\Pi D_{\boldsymbol{w}} \Pi^\top$ and L have respectively n- and (≥ 1)-dim null spaces, so the inversion may not be possible. Even when this does not happen exactly, this may incur numerical instabilities in computing the inverse. For domains where this risk exists, picking a small $\varepsilon > 0$ solves the problem.

Let $\hat{\boldsymbol{b}}_j^y$ denote the rows of $\hat{B}^{\pm}$, the solution of Problem 4.17, in lieu of the true $\boldsymbol{b}_j^y = \mathbb{E}_{\mathcal{S}_j}[\boldsymbol{x}|y]$. Those statistics are used to estimate the mean operator by Equation 4.6. We state the Laplacian Mean Map algorithm (LMM) in Algorithm 5, a second example of our two-step estimation/minimization procedure.

4.6 Estimation: formal guarantees

We have shown how to estimate the sufficient statistic. Although, the set up of Problem 4.17 is strongly dependent on our assumptions. It is therefore relevant to study formal guarantees of our estimator. We compare $\boldsymbol{\mu}_j \doteq \pi_j \boldsymbol{b}_j^+ - (1-\pi_j)\boldsymbol{b}_j^-$ to our estimates $\hat{\boldsymbol{\mu}}_j \doteq \pi_j \hat{\boldsymbol{b}}_j^+ - (1-\pi_j)\hat{\boldsymbol{b}}_j^-$, $\forall j \in [n]$, granted that $\boldsymbol{\mu} = \sum_j p_j \boldsymbol{\mu}_j$ and $\hat{\boldsymbol{\mu}} = \sum_j p_j \hat{\boldsymbol{\mu}}_j$.

Theorem 36. *Suppose that γ satisfies $\gamma\sqrt{2} \leq (\varepsilon(2n)^{-1} + \max_{j \neq j'} v_{jj'}) / \min_j w_j$. Let $M \doteq$*

$[\boldsymbol{\mu}_1|\boldsymbol{\mu}_2|...|\boldsymbol{\mu}_n]^\top \in \mathbb{R}^{n\times d}$, $\hat{M} \doteq [\hat{\boldsymbol{\mu}}_1|\hat{\boldsymbol{\mu}}_2|...|\hat{\boldsymbol{\mu}}_n]^\top \in \mathbb{R}^{n\times d}$ *and:*

$$\varsigma(V, B^{\pm}) \doteq \left(\frac{\varepsilon}{2n} + \max_{j\neq j'} v_{jj'}\right)^2 \|B^{\pm}\|_F \ .$$

The following holds:

$$\|M - \hat{M}\|_F \leq \frac{\sqrt{n/2}}{\min_j w_j^2} \cdot \varsigma(V, B^{\pm}) \ . \tag{4.20}$$

Proof in 4.11.2. The multiplicative factor to ς in 4.20 is roughly $O(n^{5/2})$ when there is no large discrepancy in the bias matrix $D_{\boldsymbol{w}}$, so the upper bound is driven by $\varsigma(.,.)$ when there are not many bags. We study its variations when the "distinguish-ability" between bags increases. This setting is interesting because in this case we may kill two birds with one stone, with the estimation of M *and* the subsequent learning problem potentially easier, in particular for linear separators. We consider two examples for $v_{jj'}$, the first being the normalized association [Shi and Malik, 2000]:

$$v_{jj'}^{nc} \doteq \frac{1}{2}\left(\frac{\textsc{assoc}(\mathcal{S}_j, \mathcal{S}_j)}{\textsc{assoc}(\mathcal{S}_j, \mathcal{S}_j \cup \mathcal{S}_{j'})} + \frac{\textsc{assoc}(\mathcal{S}_{j'}, \mathcal{S}_{j'})}{\textsc{assoc}(\mathcal{S}_{j'}, \mathcal{S}_j \cup \mathcal{S}_{j'})}\right) = \textsc{Nassoc}(\mathcal{S}_j, \mathcal{S}_{j'}) \ , \tag{4.21}$$

$$v_{jj'}^{G,s} \doteq \exp(-\|\boldsymbol{b}_j - \boldsymbol{b}_{j'}\|_2/s) \ , s > 0 \ . \tag{4.22}$$

Here, $\textsc{assoc}(\mathcal{S}_j, \mathcal{S}_{j'}) \doteq \sum_{\boldsymbol{x}\in\mathcal{S}_j, \boldsymbol{x}'\in\mathcal{S}_{j'}} \|\boldsymbol{x} - \boldsymbol{x}'\|_2^2$. To put these two similarity measures in the context of Theorem 36, consider the setting where we can make assumption:

(**A4.2**) $\exists \kappa > 0$ such that $\forall j, j' \ \|\boldsymbol{b}_j - \boldsymbol{b}_{j'}\|_2^2 \geq \kappa \max_{y,j} \|\boldsymbol{b}_j^y\|_2^2$.

This is a weak distinguish-ability property as if no such κ exists, then the centers of distinct bags may just be confounded. Consider also the additional assumption that:

(**A4.3**) $\exists \kappa' > 0$ such that $\max_j d_j^2 \leq \kappa', \forall j \in [n]$, where $d_j \doteq \max_{\boldsymbol{x}_i, \boldsymbol{x}_i' \in \mathcal{S}_j} \|\boldsymbol{x}_i - \boldsymbol{x}_{i'}\|_2$ is a bag's diameter.

In the following Lemma, the *little-o* notation is with respect to the "largest" unknown in Equation 4.11, *i.e.* $\max_{y,j} \|\boldsymbol{b}_j^y\|_2$.

Lemma 37. *There exists $\varepsilon_* > 0$ such that $\forall \varepsilon \leq \varepsilon_*$, the following holds:*

*(i) $\varsigma(V^{nc}, B^{\pm}) = o(1)$ under assumptions (**A4.2** + **A4.3**);*
*(ii) $\varsigma(V^{G,s}, B^{\pm}) = o(1)$ under assumption (**A4.2**), $\forall s > 0$.*
(iii) $\varsigma(V^{G,s}, B^{\pm})$ converges faster than $\varsigma(V^{nc}, B^{\pm})$.

Proof in 4.11.3. Hence, provided a weak (**A4.2**) or stronger (**A4.2**+**A4.3**) distinguish-ability assumption holds, the divergence between M and $\hat{M}$ gets smaller with the

increase of the norm of the unknowns $\boldsymbol{b}_j^y$. The following Lemma shows that both similarities also partially encode the hardness of solving the classification problem with linear separators, so that the manifold regularizer "limits" the distortion of the $\hat{\boldsymbol{b}}_j^y$s between two bags that tend not to be linearly separable.

Lemma 38. *Take $v_{jj'} \in \{v_{jj'}^{G,s}, v_{jj'}^{nc}\}$. There exists $0 < \kappa_l < \kappa_n < 1$ such that:*

(i) if $v_{jj'} > \kappa_n$ then $\mathcal{S}_j, \mathcal{S}_{j'}$ are not linearly separable;
(ii) if $v_{jj'} < \kappa_l$ then $\mathcal{S}_j, \mathcal{S}_{j'}$ are linearly separable.

Proof in 4.11.4. This Lemma is an advocacy to fit s of $v_{jj'}^{G,s}$ in a data dependent way.

A question may be raised as to whether finite sample approximation results like Theorem 36 can be proven for the Mean Map estimator. The following answers in the negative.

Lemma 39. *For any $\gamma > 0$, the Mean Map estimator (Equation 4.10) cannot guarantee:*

$$\frac{\|\hat{\boldsymbol{\mu}} - \boldsymbol{\mu}\|_2}{\max_{y,j} \|\boldsymbol{b}_j^y\|_2} \leq 2 - \gamma \ , \tag{4.23}$$

*even when (**A4.2** + **A4.3**) hold.*

Proof in 4.11.5. It is not hard to check that a randomized procedure that builds $\hat{\mu} \doteq y\boldsymbol{x}$ for some random $(\boldsymbol{x}, y) \in \mathcal{S}$ guarantees $O(2+\gamma)$ approximability when some bags are close to the convex hull of $\mathcal{S}$, for small $\gamma > 0$. In contrast, the Mean Map estimation may be very poor in that respect.

We now move to consider the quality of the model. The following Theorem shows a data dependent approximation bound that refines Theorem 26 under the assumption that we use SPLs that satisfy an additional requirement.

Definition 40. *Let $\phi'(x)$ be the derivative of $\phi(x)$ with regard to scalar x. We say that a symmetric proper loss ℓ is* proper scoring compliant *(PSC) when:*

$$\forall \boldsymbol{x}_i \in \mathcal{S}, \ \ \langle \boldsymbol{\theta}^\star, \boldsymbol{x}_i \rangle \text{ and } \langle \hat{\boldsymbol{\theta}}, \boldsymbol{x}_i \rangle \in \phi'([0,1]), \ .$$

This condition always holds for logistic and Matsushita losses for which $\phi'([0,1]) = \mathbb{R}$. For other losses such as square loss for which $\phi'([0,1]) = [-1,1]$, shrinking the observations in a ball of sufficiently small radius is sufficient to ensure this. Proof in 4.11.6.

Theorem 41. *Let $X \doteq \max_i \|\boldsymbol{x}_i\|_2$. Let ℓ be SPL and PSC, and let ϕ', ϕ'' be first and second derivative of ϕ. Let $\boldsymbol{f}_k \in \mathbb{R}^m$ denote the vector encoding the k^{th} feature variable in $\mathcal{S}$: $f_{ki} = x_{ik}$, with $k \in [d]$. Let $\overline{F}$ denote the feature matrix with column-wise normalized feature vectors:*

$$\overline{\boldsymbol{f}}_k \doteq \left(\frac{d}{\sum_{k'} \|\boldsymbol{f}_{k'}\|_2^2} \right)^{(d-1)/(2d)} \cdot \boldsymbol{f}_k \ . \tag{4.24}$$

Without loss of generality, we assume $\overline{F}^\top \overline{F}$ positive definite. Call $\hat{\boldsymbol{\theta}}$ and $\boldsymbol{\theta}^\star$ the minimizer of problem (ii) in Meta Algorithm 1 when, respectively, the empirical risk is computed with $\hat{\boldsymbol{\mu}}$ from the estimator of (i) and $\boldsymbol{\mu} = \mathbb{E}_{\mathcal{S}}[y\boldsymbol{x}]$. We have that:

$$\left\|\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}^\star\right\|_2^2 \leq \frac{1}{\lambda + q} \left\|\hat{\boldsymbol{\mu}} - \boldsymbol{\mu}\right\|_2^2 \ , \tag{4.25}$$

with:

$$q \doteq \frac{1}{em} \cdot \frac{\det\left(\overline{F}^\top \overline{F}\right)}{b_\phi \ \phi''\left(\phi'^{-1}\left(\frac{\overline{q}}{b_\phi \ \lambda}\right)\right)} \ (> 0) \ , \tag{4.26}$$

for some $\overline{q}$ in the interval $[\pm(X + \max\{\|\boldsymbol{\mu}\|_2, \|\hat{\boldsymbol{\mu}}\|_2\})]$.

4.7 Alternating Mean Map

We design a second algorithm for LLP called Alternating Mean Map (AMM). This algorithms does not belong to the family of algorithms defined by our two-step procedure stated in Meta Algorithm 1. Instead, it follows the more conventional strategy of coordinate descent for weakly supervised learning. Estimation of sufficient statistic and model learning are interleaved. Once we learn a new model, it can be applied to provide a better estimation of the mean operator. We do so by estimating the unknown labels under the constraints given by the label proportions. LMM can be used to initialize the alternating procedure. In the experiments of Section 4.9 we also demonstrate that is the best choice for initialization.

Let us denote the set of labelings that are *consistent* with the proportions $\boldsymbol{\pi}$ as:

$$\Sigma_{\boldsymbol{\pi}} \doteq \left\{ \boldsymbol{\sigma} \in \Sigma_m : \sum_{i:\boldsymbol{x}_i \in \mathcal{S}_j} \sigma_i = (2\pi_j - 1)m_j, \forall j \in [n] \right\} \ , \tag{4.27}$$

and the (possibly biased) mean operator computed as $\boldsymbol{\mu}(\boldsymbol{\sigma}) \doteq (1/m) \sum_i \sigma_i \boldsymbol{x}_i$ from some $\boldsymbol{\sigma} \in \Sigma_{\boldsymbol{\pi}}$. Notice that the true mean operator is $\boldsymbol{\mu} = \boldsymbol{\mu}(\boldsymbol{\sigma})$ for at least one $\boldsymbol{\sigma} \in \Sigma_{\boldsymbol{\pi}}$. The objective function augmented with latent variables $\boldsymbol{\sigma}$ is the following:

$$\frac{1}{2m} \sum_{i=1}^{m} \sum_{y \in \mathcal{Y}} \ell(y\langle \boldsymbol{\theta}, \boldsymbol{x}_i \rangle) + a\langle \boldsymbol{\theta}, \boldsymbol{\mu}(\boldsymbol{\sigma}) \rangle + \lambda \|\boldsymbol{\theta}\|_2^2 \ . \tag{4.28}$$

This approach fits into the framework of Problem 2.27 for ERM under weak supervision. An important insight is that, by Factorization, $\boldsymbol{\sigma}$ affects the empirical risk only through the value of $\langle \boldsymbol{\theta}, \boldsymbol{\mu}(\boldsymbol{\sigma}) \rangle$. Therefore, for a fixed model $\boldsymbol{\theta}$, the optimum is found by optimizing a linear function in the space of consistent labelings $\Sigma_{\boldsymbol{\pi}}$.

The Alternating Mean Map algorithm (Algorithm 6) starts with the output of LMM and then optimizes it further over the set of consistent labelings. We give two possible implementations, corresponding to either a min-max or a min-min learning

Algorithm 6: Alternating Mean Map (AMM$^{\text{OPT}}$)

Input: LMM params, optimization strategy OPT $\in \{\min, \max\}$, convergence predicate PR
$\boldsymbol{\theta}_0 \leftarrow LMM(\text{params})$
repeat for $t = 0, 1, \ldots$
(a) $\quad \boldsymbol{\sigma}_t \leftarrow \arg \text{OPT}_{\boldsymbol{\sigma} \in \Sigma_\pi} \langle \boldsymbol{\theta}, \boldsymbol{\mu}(\boldsymbol{\sigma}) \rangle$
(b) $\quad \boldsymbol{\theta}_t \leftarrow \arg\min_{\boldsymbol{\theta}} \frac{1}{2m} \sum_{i=1}^{m} \sum_{y \in \mathcal{Y}} \ell(y \langle \boldsymbol{\theta}, \boldsymbol{x}_i \rangle) + a \langle \boldsymbol{\theta}, \boldsymbol{\mu}(\boldsymbol{\sigma}) \rangle + \lambda \|\boldsymbol{\theta}\|_2^2$
until predicate PR is true
Output: $\hat{\boldsymbol{\theta}}$ (see text)

strategy. At each iteration, in Step (a) we first pick a consistent labeling in Σ_π that is the best (OPT = min) or the worst (OPT = max) for the current classifier, and then in Step (b) we fit the classifier $\hat{\boldsymbol{\theta}}$ on the given set of labels. We iterate until a convergence predicate is met, which tests whether the empirical risk difference with the previous iteration is small enough (AMM$^{\text{min}}$), or the number of iterations exceeds a user-specified limit (AMM$^{\text{max}}$). The returned classifier $\hat{\boldsymbol{\theta}}$ is the one with the smallest empirical risk; in the case of AMM$^{\text{min}}$, the output is the last learned model since the risk cannot increase.

Step (b) is a convex minimization with no technical difficulty. The label inference Step (a) is combinatorial as we optimize a linear function for integer variables. Yet, it can be solved in time almost linear in m, exploiting a trick similar to one elaborated in Yu et al. [2013]. The insight is that the search for labels consistent with each label proportion can be done via sorting. Moreover, we are optimizing a linear objective that decomposes by bag. Therefore, bag by bag, we can sort observations by the value of the inner product with the current model and label them as positive until we match the label proportion for the bag. The next Lemma is formally proven in 4.11.7.

Lemma 42. *The running time of Step (a) in AMM is* $O(m \log m)$.

4.8 Generalization bounds

We now study generalization bounds for LLP. We are not aware of much prior work on the topic. The only known generalization theory for LLP is in Yu et al. [2014b] that formulate uniform convergence bounds but with a focus on coarser grained problems — the estimation of bag label proportions – not directly the classifiers.

We obtain a first bound by applying Theorem 23 and utilizing Theorem 36 for evaluating how generalization degrades when we estimate the mean operator with LMM. The next Theorem holds for linear classifiers.

Theorem 43. *Let* $\varepsilon = 0$. *Hold the same assumptions and notation from Theorems 23 and 36. Let* ℓ *be* a-LOL. *Let* $\text{Harm}(m) \doteq \frac{n}{\sum_{j \in [n]} 1/m_j}$ *be the harmonic mean of the bag sizes. For*

any $\delta > 0$*, with probability at least* $1 - \delta$*:*

$$R_{\mathcal{D},\ell}(\hat{\boldsymbol{\theta}}) - R_{\mathcal{D},\ell}(\boldsymbol{\theta}^\star) \leq \frac{\sqrt{2}+1}{2} \cdot \frac{XHL}{\sqrt{m}} + c(X,H) \cdot \sqrt{\frac{1}{m} \log\left(\frac{2}{\delta}\right)}$$
$$+ 2|a|H \left(X\sqrt{\frac{d}{m}\log\left(\frac{2d}{\delta}\right)} + X^{1/4} \frac{\left(\max_{j \neq j'} v_{jj'}\right)^2}{\min_j w_j^2} \, n^{3/2} \sqrt{\frac{1}{\mathrm{Harm}(m)}} \right) \; . \tag{4.29}$$

Proof in 4.11.8. The Theorem highlights the effect of the mean operator as estimated by LMM, which trades off number of bags and (the harmonic mean of) their sizes. In particular, if we assume all bags with same size, we have $\forall j, m_j = m/n$ and therefore:

$$n^{3/2}\sqrt{\frac{1}{n}\sum_{j\in[n]} \frac{n}{m}} = n^{3/2}\sqrt{\frac{n}{m}} = \frac{n^2}{\sqrt{m}} \; . \tag{4.30}$$

Once again, the rate of convergence is $O(\sqrt{m})$. The bound can be improved as we did above for Theorem 36 once we assume (**A4.2**) or (**A4.3**). The condition $\varepsilon = 0$ is not a requirement but it simplifies the bound shape. The result confirms that novel bounds can be derived by simply plugging in any known guarantee on the mean operator estimators. We will show the same convenience in Chapter 5 analyzing the case of noisy labels.

We formulate an additional, more generic bound in the following. This result fits well with AMM and various algorithms that work by minimizing the *bag empirical* ℓ*-risk* $\mathbb{E}_{\sigma\sim\Sigma_\pi}\mathbb{E}_{\mathcal{S}}[\ell(\sigma h(\boldsymbol{x}))]$. This quantity is often part of the objective of LLP algorithms inspired by EM. AMM$^{\text{min}}$ and AMM$^{\text{max}}$ respectively minimize a lower bound and an upper bound of this risk. Here the model space $\mathcal{H}$ is generic and not restricted to linear classifiers. The bound relies on a generalization of Rademacher complexity that better suits the LLP setting.

Definition 44. *The* bag empirical Rademacher complexity *of a hypothesis space* $\mathcal{H}$ *with regard to sample* $\mathcal{S}$ *of size* m *and loss* ℓ *is:*

$$\mathcal{R}^b(\ell \circ \mathcal{H} \circ \mathcal{S}) \doteq \mathbb{E}_{\boldsymbol{\sigma}\sim\Sigma_m}\left[\sup_{h\in\mathcal{H}} \mathbb{E}_{\boldsymbol{\sigma}'\sim\Sigma_\pi}\left[\frac{1}{m}\sum_{i=1}^{m} \sigma \, \ell(\sigma' h(\boldsymbol{x}_i))\right]\right] \; . \tag{4.31}$$

and the respective bag Rademacher complexity *is* $\mathcal{R}^b(\ell \circ \mathcal{H}) \doteq \mathbb{E}_{\mathcal{S}\sim\mathcal{D}}\mathcal{R}^b(\ell \circ \mathcal{H} \circ \mathcal{S})$.

Notice that the usual empirical Rademacher complexity (Definition 4) equals $\mathcal{R}^b$ when $|\Sigma_\pi| = 1$. It is also useful to define another type of complexity for the LLP, related with the variation of the label proportions per bag. Let $\mathcal{I}_1^{\prime 2}$ and $\mathcal{I}_2^{\prime 2}$ be two m-size *i.i.d.* samples of $[2m]$ and $\mathcal{S}(\mathcal{I}_1^{\prime 2})$ and $\mathcal{S}(\mathcal{I}_2^{\prime 2})$ be the size-m subsets of $\mathcal{S}$ corresponds to those indices. Take $l = 1, 2$ and any $\boldsymbol{x}_i \in \mathcal{S}$. If $i \notin \mathcal{I}_l^{\prime 2}$ then $\pi^s_{|l}(\boldsymbol{x}_i) = \pi^l_{|l}(\boldsymbol{x}_i)$ is $\boldsymbol{x}_i$'s bag's label proportion measured on $\mathcal{S}\backslash\mathcal{S}(\mathcal{I}_l^{\prime 2})$. Else, $\pi^s_{|2}(\boldsymbol{x}_i)$ is its bag's label proportion measured on $\mathcal{S}(\mathcal{I}_2^{\prime 2})$ and $\pi^l_{|1}(\boldsymbol{x}_i)$ is its label, *i.e.* a bag's label proportion that would contain only $\boldsymbol{x}_i$. Finally, $\sigma_1(\boldsymbol{x}) \doteq 2 \cdot 1\{\boldsymbol{x} \in \mathcal{S}(\mathcal{I}_1^{\prime 2})\} - 1 \in \Sigma_1$.

Definition 45. *The* label proportion complexity *of $\mathcal{H}$ is:*

$$L(\mathcal{H}) \doteq \mathbb{E}_{\mathcal{D}_{2m}} \mathbb{E}_{\mathcal{I}_1^{/2}, \mathcal{I}_2^{/2}} \sup_{h \in \mathcal{H}} \frac{1}{m} \sum_{i=1}^{m} \left[\sigma_1(\boldsymbol{x}_1) \left(\pi^s_{|2}(\boldsymbol{x}_i) - \pi^l_{|1}(\boldsymbol{x}_i) \right) h(\boldsymbol{x}_i) \right] \ . \tag{4.32}$$

$L(\mathcal{H})$ tends to be smaller as classifiers in $\mathcal{H}$ have small magnitude on bags whose label proportion is close to $1/2$. Despite similar shapes, $\mathcal{R}^b$ and L behave differently: when bags are pure, $\pi_j \in \{0, 1\}, \forall j$, $L = 0$. When bags are most impure, $\pi_j = 1/2, \forall j$, $\mathcal{R}^b = 0$. As bags are impure, the bag empirical ℓ-risk also tends to increase. We can formulate a generalization bound for the bag empirical ℓ-risk involving a balance of the two complexity measures.

Theorem 46. *Suppose $\exists H \geq 0, |h(\boldsymbol{x})| \leq H$. Then, for any* SPL *$\ell$ and any $0 < \delta \leq 1$, with probability $> 1 - \delta$, the following bound holds over all $h \in \mathcal{H}$:*

$$\begin{aligned} R_{\mathcal{D},\ell}(h) - \mathbb{E}_{\sigma \sim \Sigma_\pi} \mathbb{E}_{\mathcal{S}}[\ell(\sigma h(\boldsymbol{x}))] &\leq 2\mathcal{R}^b(\ell \circ \mathcal{H} \circ \mathcal{S}) \\ &+ L(\mathcal{H}) + 4 \left(\frac{2H}{b_\phi} + 1 \right) \sqrt{\frac{1}{2m} \log \frac{2}{\delta}} \ . \end{aligned} \tag{4.33}$$

Furthermore, under PSC*, we have for any* SPL *ℓ:*

$$\mathcal{R}^b(\ell \circ \mathcal{H} \circ \mathcal{S}) \leq 2 b_\phi \ \mathbb{E}_{\Sigma_m} \sup_{h \in \mathcal{H}} \{ \mathbb{E}_{\mathcal{S}}[\sigma(\boldsymbol{x})(\pi(\boldsymbol{x}) - 1/2) h(\boldsymbol{x})] \} \ . \tag{4.34}$$

Proof in 4.11.9. This bound is the LLP equivalent to the first statement of Theorem 5, limiting the difference between ℓ-risk and its empirical counterpart, which is here averaged in the set of labels consistent with the given proportions.

4.9 Experiments

4.9.1 Algorithms

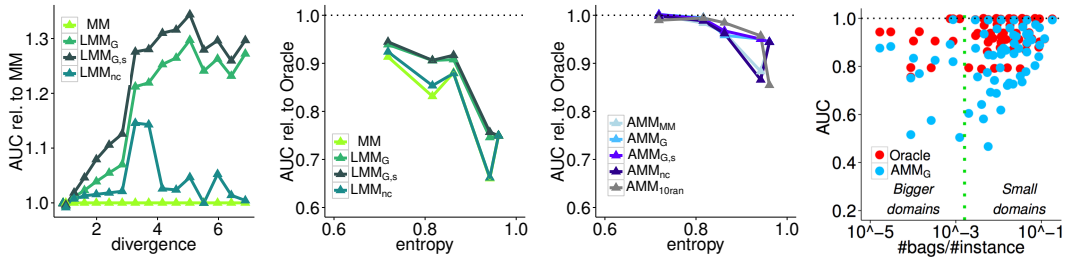
We compare LMM, AMM (ℓ = logistic loss) to the original MM [Quadrianto et al., 2009], InvCal [Rüping, 2010], conv-$\propto$SVM and alter-$\propto$SVM [Yu et al., 2013] with linear kernels. To obtain strong baselines, we test several additional initializations for AMM: the edge mean map estimator (AMM_{EMM}):

$$\hat{\mu} = \frac{1}{m^2} \left(\sum_i y_i \right) \left(\sum_i \boldsymbol{x}_i \right) \ , \tag{4.35}$$

the constant estimator (AMM_1):

$$\hat{\mu} = \mathbf{1} \ , \tag{4.36}$$

and finally $\text{AMM}_{10\text{ran}}$ which runs 10 random initial models ($\|\boldsymbol{\theta}_0\|_2 \leq 1$) and selects the one with smallest risk. This procedure is akin to alter-$\propto$SVM.



(a) (b) (c) (d)

Figure 4.1: Relative AUC (w.r.t. MM) as homogeneity assumption is violated (a). Relative AUC (w.r.t. Oracle) vs entropy on *heart* for LMM(b), $\text{AMM}^{\min}$(c). Relative AUC vs n/m for $\text{AMM}^{\min}_{G,s}$ (d).

Matrix V (Equations 4.21, 4.22) used is indicated in subscript: $\text{LMM}/\text{AMM}_{\text{G}}$, $\text{LMM}/\text{AMM}_{\text{G,s}}$, $\text{LMM}/\text{AMM}_{\text{nc}}$ respectively denote $v^{G,s}$ with $s = 1$, $v^{G,s}$ with s learned on cross validation and v^{nc}. We split the algorithms in two groups, *one-shot* and *iterative*. The latter, including AMMand (conv/alter)-$\propto$SVM, iteratively optimize a cost over labelings. The former (LMM, InvCal) do not and are thus much faster.

The range of hyper-parameters for cross validation are $\lambda = \lambda' m$ with $\lambda' \in \{0\} \cup 10^{\{0,1,2\}}$, $\gamma \in 10^{-\{2,1,0\}}$, $\sigma \in 2^{-\{2,1,0\}}$ for mean operator algorithms. We run all experiments with $D_w = I$ and $\varepsilon = 0$. For $\propto$SVM and InvCal, we used an implementation provided by the authors of [Yu et al., 2013]. Since we test on similar domains (6 are actually the same), hyper-parameters ranges for InvCal and $\propto$SVM are taken from [Yu et al., 2013]. To avoid an additional source of complexity in the analysis, we cross-validate all hyper-parameters using the knowledge of all labels of the validation sets — information that generally would not be accessible in real world applications. For all our experiments, the testing metric is the AUC.

4.9.2 Simulated domains

We generate 16 domains that gradually move away the $\boldsymbol{b}_j^\sigma$ away from each other, thus increasingly violating the homogeneity assumption (**A4.0**). The degree of violation is measured as $\|B^{\pm} - \overline{B^{\pm}}\|_F$, where $\overline{B^{\pm}}$ is the "homogeneity assumption matrix", that replaces all $\boldsymbol{b}_j^\sigma$ by $\boldsymbol{b}^\sigma$ for $\sigma \in \{-1, 1\}$, see Equation 4.8. Figure 4.1 (a) displays the AUC ratios LMM with respect to MM. It shows that LMM is all the better with respect to MM as the homogeneity assumption is violated. Furthermore, learning the width parameter s in LMM improves the results.

The MM algorithm was shown to learn a model with zero accuracy on a toy domain crafted in Yu et al. [2013]. We reproduce the experiment and test all our methods. In Table 4.2 we report performance of all measured in transductive setting, *i.e.* on the same training set. Although none of the distances used in our experiments in LMM leads reasonable accuracy in the toy dataset (not reported in the Table), $\text{AMM}^{\max}$ initialized with *any* starting point learns *in one step* a model which perfectly

	AMM^{min}	AMM^{max}
EMM	100.00	100.00
MM	8.46	100.00
LMM_{G}	8.46	100.00
$\text{LMM}_{\text{G,s}}$	8.46	100.00
LMM_{nc}	8.46	100.00
1	8.46	100.00
10ran	100.00	100.00

Table 4.2: AUC on the toy dataset of Yu et al. [2013]

classifies all the instances. We also notice that EMM returns an optimal classifier by itself (not reported in Table 4.2).

4.9.3 UCI domains

We convert 10 small domains ($m \leq 1000$) and 4 bigger ones ($m > 8000$) from UCI [Bache and Lichman, 2013] into the LLP setting. We cast to one-against-all classification when the problem is multi-class. On large domains, the bag assignment function is inspired by [Yu et al., 2014b]: we craft bags according to a selected categorical feature, and then we remove that feature from the data. This conforms to the idea that bag assignment is structured and non random in real-world problems. Most of our small domains however do not have many features, so instead of clustering on one feature and then discarding it, we run k-MEANS on the whole data to make the bags, for $k = n \in 2^{[5]}$.

We perform 5-folds nested CV comparisons on the 10 domains = 50 AUC values for each algorithm. Full experimental results are given in Appendix 7.9, including runtime and details by domain. Table 4.3 synthesizes the results, splitting one-shot and iterative algorithms:

(1) $\text{LMM}_{\text{G,s}}$ outperforms all one-shot algorithms.

(2) LMM_{G} and $\text{LMM}_{\text{G,s}}$ are competitive with many iterative algorithms, but lose against their AMM counterpart, which proves that additional optimization over labels is beneficial.

(3) AMM_G and $AMM_{G,s}$ are confirmed as the best variant of AMM, the first being the best in this case.

(4) Surprisingly, all mean map algorithms, even one-shots, are clearly superior to ∝SVMs. Further results reveal that ∝SVM performances are dampened by learning classifiers with the "inverted polarity" — *i.e.* flipping the sign of the classifier improves its performances. See Appendix 4.12.2.

(5) Figure 4.1 (b, c) presents the AUC relative to the Oracle — which learns the classifier knowing all labels and minimizing the logistic loss —, as a function of the Gini entropy of bag assignment, $gini(\mathcal{S}) \doteq 4\mathbb{E}_j[\pi_j(1-\pi_j)]$. For an entropy close to 1, we were expecting a drop in performances. The unexpected is that on some domains, large entropies ($\geq .8$) do not prevent AMM^{min} to compete with the Oracle. No such

pattern clearly emerges for $\propto$SVM and $\text{AMM}^{\max}$.

We now consider the 4 bigger datasets. We adopt a 1/5 hold-out method. Scalability results display that every method using v^{nc} and $\propto$SVM is not scalable to big domains. Table 4.4 presents the results on the big domains, distinguishing the feature used for bag assignment. $\text{AMM}_{10\text{ran}}$ does not appear because of clearly inferior performance. Big domains confirm the efficacy of LMM+AMM. No approach clearly outperforms the rest, although $\text{LMM}_{\text{G,s}}$ is often the best one-shot and $\text{AMM}_{\text{G}}^{\min}$ and $\text{AMM}_{\text{G,s}}^{\min}$ outperform all the methods several times.

Figure 4.1 (d) gives the AUC of $\text{AMM}_{\text{G}}^{\min}$ over the Oracle for *all* domains, as a function of the "degree of supervision", n/m (=1 if the problem is fully supervised). Noticeably, on 90% of the runs, $\text{AMM}_{\text{G}}^{\min}$ gets an AUC representing at least 70% of the Oracle's. Results on big domains can be remarkable: on the census domain with bag assignment on *race*, 5 proportions are sufficient for an AUC 5 points below the Oracle's — which learns with 200K labels.

4.10 Discussion

We have shown methods that can learn with label proportions successfully. By sufficiency we resort to standard learning procedures for binary classification. This is implemented as the LMM algorithm which estimates the mean operator via a Laplacian-based manifold regularizer relaxing the independence assumption of Quadrianto et al. [2009]. We show that under a weak distinguish-ability assumption between bags, our estimation of the mean operator is all the better as the maximal observation norm increase. This, as we show, cannot hold for the MM algorithm of Quadrianto et al. [2009]. Generalization bound are easily derived specializing the theory of Chapter 3.

We have also provided an iterative algorithm, AMM, that takes as input the solution of LMM and optimizes it further over the set of consistent labelings. We ground the algorithm in a uniform convergence result involving a generalization of Rademacher complexities for the LLP setting. The bound involves a bag surrogate risk for which we show that AMM optimizes tractable bounds.

Experiments display results that are superior to the state of the art, with algorithms that scale to big domains at affordable computational costs. Performances sometimes compete with the Oracle's — that learns knowing all labels —, even on big domains. On one side, such experimental finding are encouraging for Machine Learning applications where the given supervision is only coarse-grain. On the other side, we uncover severe implications on the reliability of privacy preserving aggregation techniques with simple group statistics like proportions, still common for the public release of summaries on sensitive data. Future work shall study the extent to which LLP methods may threaten anonymity of people and sensitivity of their attributes when polls-like results are released on the Internet and when additional individual information on those individuals is available.

	algorithm	MM	LMM			InvCal	AMM^{min}				AMM^{max}				*conv-*
		G	G,s	nc	MM	G	G,s	10ran	MM	G	G,s	10ran	$\propto$*SVM*		
LMM	G	**36**/4													
LMM	G,s	**38**/3	**30**/6												
LMM	nc	**28**/12	3/**37**	2/**37**											
	InvCal	4/**46**	3/**47**	4/**46**	4/**46**										
AMM^{min}	MM	**33**/16	26/24	25/25	32/18	**46**/4									
AMM^{min}	G	**38**/11	**35**/14	**30**/20	**37**/13	**47**/3	**31**/7								
AMM^{min}	G,s	**35**/14	**33**/17	**30**/20	**35**/15	**47**/3	**24**/11	7/**15**							
AMM^{min}	10ran	27/22	24/26	22/28	26/24	**44**/6	20/**30**	16/**34**	19/**31**						
AMM^{max}	MM	25/25	23/**27**	22/**28**	25/25	**45**/5	15/**35**	13/**37**	13/**37**	8/**42**					
AMM^{max}	G	27/23	22/**28**	21/**28**	**26**/24	**45**/5	17/**33**	14/**36**	14/**36**	10/40	13/14				
AMM^{max}	G,s	25/25	21/**29**	22/**28**	24/**26**	**45**/5	15/**35**	13/**37**	13/**37**	12/38	15/22	16/22			
AMM^{max}	10ran	23/**27**	21/**29**	19/**31**	24/**26**	**50**/0	19/**31**	15/**35**	17/**33**	7/**43**	19/**30**	20/**29**	17/**32**		
SVM	conv-$\propto$	21/**29**	2/**48**	2/**48**	2/**48**	2/**48**	4/**46**	3/**47**	3/**47**	4/**46**	3/**47**	3/**47**	4/**46**	0/**50**	
SVM	alter-$\propto$	0/**50**	0/**50**	0/**50**	0/**50**	20/**30**	0/**50**	0/**50**	0/**50**	3/**47**	3/**47**	2/**48**	1/**49**	0/**50**	27/23

↙ *e.g.* AMM$^{min}_{G,s}$ wins on AMM$^{min}_{G}$ 7 times, loses **15**, with 28 ties

Table 4.3: Small domains results. #win/#lose for row vs column. Bold faces means p-val $< .001$ for Wilcoxon signed-rank tests. Top-left subtable is for one-shot methods, bottom-right iterative ones, bottom-left compare the two. Italic are the best results. Grey cells highlight the best of all (AMM$^{min}_{G}$).

	algorithm	mushroom: 8124 × 108			adult: 48842 × 89			marketing: 45211 × 41			census: 299285 × 381		
		I(6)	II(7)	III(10)	IV(5)	V(16)	VI(42)	V(4)	VII(4)	VIII(12)	IV(5)	VIII(9)	VI(42)
	EMM	55.61	59.80	76.68	43.91	47.50	66.61	**63.49**	**54.50**	44.31	56.05	56.25	57.87
	MM	51.99	**98.79**	5.02	80.93	76.65	74.01	54.64	50.71	49.70	75.21	**90.37**	75.52
	LMM_G	73.92	98.57	14.70	81.79	78.40	78.78	54.66	51.00	51.93	75.80	71.75	**76.31**
	$LMM_{G,s}$	**94.91**	98.24	**89.43**	**84.89**	**78.94**	**80.12**	49.27	51.00	**65.81**	**84.88**	60.71	69.74
AMM^{min}	AMM_{EMM}	85.12	99.45	69.43	49.97	56.98	70.19	61.39	55.73	43.10	87.86	87.71	40.80
AMM^{min}	AMM_{MM}	89.81	99.01	15.74	**83.73**	77.39	80.67	52.85	**75.27**	58.19	89.68	84.91	68.36
AMM^{min}	AMM_G	89.18	99.45	50.44	83.41	**82.55**	**81.96**	51.61	75.16	57.52	87.61	88.28	76.99
AMM^{min}	$AMM_{G,s}$	89.24	**99.57**	3.28	81.18	78.53	**81.96**	52.03	75.16	53.98	**89.93**	83.54	52.13
AMM^{min}	AMM_1	**95.90**	98.49	97.31	81.32	75.80	80.05	65.13	64.96	66.62	89.09	**88.94**	56.72
AMM^{max}	AMM_{EMM}	93.04	3.32	26.67	54.46	69.63	56.62	51.48	55.63	57.48	71.20	77.14	66.71
AMM^{max}	AMM_{MM}	59.45	55.16	**99.70**	82.57	71.63	81.39	48.46	51.34	56.90	50.75	66.76	58.67
AMM^{max}	AMM_G	95.50	65.32	99.30	82.75	72.16	81.39	50.58	47.27	34.29	48.32	67.54	**77.46**
AMM^{max}	$AMM_{G,s}$	95.84	65.32	84.26	82.69	70.95	81.39	**66.88**	47.27	34.29	80.33	74.45	52.70
AMM^{max}	AMM_1	95.01	73.48	1.29	75.22	67.52	77.67	66.70	61.16	**71.94**	57.97	81.07	53.42
	Oracle	99.82	99.81	99.8	90.55	90.55	90.50	79.52	75.55	79.43	94.31	94.37	94.45

Table 4.4: AUC on big domains (*name*: #instances×#features). I=*cap-shape*, II=*habitat*, III=*cap-color*, IV=*race*, V=*education*, VI=*country*, VII=*poutcome*, VIII=*job* (number of bags); for each feature, the best result over one-shot, and over iterative algorithms is bold faced.

4.11 Appendix: proofs

4.11.1 Proof of Lemma 35

Using the fact that D_w and L are symmetric, we have:

$$\frac{\partial \ell(L, X)}{\partial X} \tag{4.37}$$

$$= -2\frac{\partial}{\partial X} \operatorname{tr} B^\top D_w \Pi^\top X + \frac{\partial}{\partial X} \operatorname{tr} X^\top \Pi D_w \Pi^\top X + \gamma \frac{\partial}{\partial X} \operatorname{tr} X^\top L X \tag{4.38}$$

$$= -2\Pi D_w B + 2\Pi D_w \Pi^\top X + 2\gamma L X = 0 \ , \tag{4.39}$$

out of which $\hat{B}^{\pm}$ follows.

4.11.2 Proof of Theorem 36

We let $\Pi_o \doteq [\operatorname{diag}(\boldsymbol{\pi}) | \operatorname{diag}(\boldsymbol{\pi} - \mathbf{1})]^\top N$ an orthonormal system ($n_{jj} = (\pi_j^2 + (1 - \pi_j)^2)^{-1/2}, \forall j \in [n]$ and 0 otherwise). Let $\mathbb{K}_{\Pi_o}$ be the n-dim subspace of $\mathbb{R}^d$ generated by Π_o. The proof of Theorem 36 exploits the following Lemma, which assumes that ε is any > 0 real for L in 4.16 to be $\succ 0$. When $\varepsilon = 0$, the result of Theorem 36 still holds but follows a different proof.

Lemma 47. *Let $A \doteq \Pi D_w \Pi^\top$ and L defined as in 4.16. Denote for short:*

$$U \doteq \left(L^{-1} A + \gamma^{-1} I \right)^{-1} \ . \tag{4.40}$$

Suppose there exists $\xi > 0$ such that for any $\boldsymbol{x} \in \mathbb{R}^{2n}$, the projection of $U\boldsymbol{x}$ in $\mathbb{K}_{\Pi_o}$, $\boldsymbol{x}_{U,o}$, satisfies:

$$\|\boldsymbol{x}_{U,o}\|_2 \leq \xi \|\boldsymbol{x}\|_2 \ . \tag{4.41}$$

Then:

$$\left\| M - \hat{M} \right\|_F \leq \gamma \xi \left\| B^{\pm} \right\|_F \ . \tag{4.42}$$

Proof Combining Lemma 35 and Equation 4.8, we get

$$B^{\pm} - \hat{B}^{\pm} = -\left((A + \gamma L)^{-1} A - I \right) B^{\pm} \tag{4.43}$$

$$= \left((\gamma L)^{-1} A + I \right)^{-1} B^{\pm} \ . \tag{4.44}$$

Define the following permutation matrix:

$$C \doteq \left[\begin{array}{c|c} 0 & I \\ I & 0 \end{array} \right] \in \mathbb{R}^{2n \times 2n} \ . \tag{4.45}$$

$A \doteq \Pi D_w \Pi^\top$ is not invertible but diagonalizable. Its (orthonormal) eigenvectors can

be partitioned in two matrices P_o and P such that:

$$P_o \doteq [\text{diag}(\boldsymbol{\pi} - \mathbf{1})|\,\text{diag}(\boldsymbol{\pi})]^\top N = C\Pi_o \in \mathbb{R}^{2n\times n} \text{ (eigenvalues 0)} , \tag{4.46}$$

$$P \doteq \Pi N \in \mathbb{R}^{2n\times n} \text{ (eigenvalues } w_j(\pi_j^2 + (1-\pi_j)^2), \forall j) . \tag{4.47}$$

We have:

$$M - \hat{M} = P_o^\top C B^\pm - P_o^\top C \hat{B}^\pm \tag{4.48}$$

$$= P_o^\top C \left((\gamma L)^{-1} A + I\right)^{-1} B^\pm \tag{4.49}$$

$$= \Pi_o^\top \left((\gamma L)^{-1} A + I\right)^{-1} B^\pm \tag{4.50}$$

$$= \gamma \Pi_o^\top \left(L^{-1} A + \gamma^{-1} I\right)^{-1} B^\pm . \tag{4.51}$$

Equation 4.50 follows from the fact that C is idempotent. Plugging Frobenius norm in 4.51, we obtain:

$$\left\|M - \hat{M}\right\|_F^2 = \gamma^2 \left\|\Pi_o^\top \left(L^{-1} A + \gamma^{-1} I\right)^{-1} B^\pm\right\|_F^2 \tag{4.52}$$

$$= \gamma^2 \sum_{k=1}^{d} \left\|\Pi_o^\top \left(L^{-1} A + \gamma^{-1} I\right)^{-1} \boldsymbol{b}_k^\pm\right\|_2^2 \tag{4.53}$$

$$\leq \gamma^2 \xi^2 \sum_{k=1}^{d} \left\|\boldsymbol{b}_k^\pm\right\|_2^2 \tag{4.54}$$

$$= \gamma^2 \xi^2 \left\|B^\pm\right\|_F^2 , \tag{4.55}$$

which yields 4.42. In 4.54, $\boldsymbol{b}_k^\pm$ denotes *column* k in $B^\pm$. Inequality 4.54 makes use of assumption 4.41. ■

To ensure $\|\boldsymbol{x}_{U,o}\|_2 \leq \xi \|\boldsymbol{x}\|_2$, it is sufficient that $\|U\boldsymbol{x}\|_2 \leq \xi \|\boldsymbol{x}\|_2$, and since $\|U\boldsymbol{x}\|_2 \leq \|U\|_F \|\boldsymbol{x}\|_2$, it is sufficient to show that:

$$\left\|U_\xi^{-1}\right\|_F^2 \leq 1 , \tag{4.56}$$

with $U_\xi \doteq L_\xi^{-1} A + \xi\gamma^{-1} I$, for relevant choices of ξ. We have let $L_\xi \doteq (1/\xi)L$. Let $0 \leq \lambda_1(.) \leq ... \leq \lambda_{2n}(.)$ denote the ordered eigenvalues of a positive-semidefinite matrix in $\mathbb{R}^{2n\times 2n}$. It follows that, since L is symmetric positive definite, we have:

$$\lambda_j(L_\xi^{-1} A) \geq \frac{\lambda_j(A)}{\lambda_{2n}(L_\xi)} \ (\geq 0) \ , \forall j \in [2n] . \tag{4.57}$$

We have used Equation 4.46. Weyl's Theorem [Horn and Johnson, 2012, Chapter 4]

then brings:

$$\lambda_j(U_\xi^{-1}) \leq \frac{\lambda_{2n}(L_\xi)}{\lambda_j(A) + \xi\gamma^{-1}\lambda_{2n}(L_\xi)} \leq \begin{cases} \xi^{-1}\gamma & \text{if} \quad j \in [n] \\ \frac{\lambda_{2n}(L_\xi)}{\lambda_j(A)} & \text{otherwise} \end{cases} . \tag{4.58}$$

Gershgorin's Theorem [Horn and Johnson, 2012, Chapter 6] brings $\lambda_{2n} \leq (1/\xi)(\varepsilon + \max_j \sum_{j'} |l_{jj'}|)$, and furthermore the eigenvalues of A satisfy $\lambda_j \geq w_j/2, \forall j \geq n+1$. We thus have:

$$\left\| U_\xi^{-1} \right\|_F^2 \leq \frac{n\gamma^2}{\xi^2} + \frac{4n\left(\varepsilon + \max_j \sum_{j'} |l_{jj'}|\right)^2}{\xi^2 \min_j w_j^2} . \tag{4.59}$$

In 4.58 and 4.59, we have used the eigenvalues of A given in Equations 4.46 and 4.47. Assuming:

$$\gamma \leq \frac{\xi}{\sqrt{2n}} , \tag{4.60}$$

a sufficient condition for the right-hand side of 4.59 to be ≤ 1 is that:

$$\xi \geq \frac{\varepsilon + \max_j \sum_{j'} |l_{jj'}|}{2\sqrt{n} \min_j w_j} . \tag{4.61}$$

To finish up the proof, recall that $L = D - V$ with $d_{jj} \doteq \sum_{j,j'} v_{jj'}$ and the coordinates $v_{jj'} \geq 0$. Hence,

$$\sum_{j'} |l_{jj'}| = 2 \sum_{j \neq j'} v_{jj'} \tag{4.62}$$

$$\leq 2n \max_{j \neq j'} v_{jj'}, \forall j \in [n] . \tag{4.63}$$

The proof is concluded by plugging this upper bound in 4.61 to choose ξ, then taking the maximal value for γ in 4.60 and finally solving the upper bound in 4.42.

4.11.3 Proof of Lemma 37

We first consider the normalized association criterion in 4.21:

$$v_{jj'}^N \doteq \frac{1}{2}\left(\frac{\textsc{assoc}(\mathcal{S}_j, \mathcal{S}_j)}{\textsc{assoc}(\mathcal{S}_j, \mathcal{S}_j \cup \mathcal{S}_{j'})} + \frac{\textsc{assoc}(\mathcal{S}_{j'}, \mathcal{S}_{j'})}{\textsc{assoc}(\mathcal{S}_{j'}, \mathcal{S}_j \cup \mathcal{S}_{j'})}\right) , \tag{4.64}$$

$$\textsc{assoc}(\mathcal{S}_j, \mathcal{S}_{j'}) \doteq \sum_{\boldsymbol{x} \in \mathcal{S}_j, \boldsymbol{x}' \in \mathcal{S}_{j'}} \|\boldsymbol{x} - \boldsymbol{x}'\|_2^2 . \tag{4.65}$$

Remark that:

$$\|\boldsymbol{b}_j - \boldsymbol{b}_{j'}\|_2^2 = \left\| \frac{1}{m_j} \sum_{\boldsymbol{x}_i \in \mathcal{S}_j} \boldsymbol{x}_i - \frac{1}{m_{j'}} \sum_{\boldsymbol{x}_{i'} \in \mathcal{S}_{j'}} \boldsymbol{x}_{i'} \right\|_2^2 \tag{4.66}$$

$$= \frac{1}{m_j^2} \left\| \sum_{\boldsymbol{x}_i \in \mathcal{S}_j} \boldsymbol{x}_i \right\|_2^2 + \frac{1}{m_{j'}^2} \left\| \sum_{\boldsymbol{x}_{i'} \in \mathcal{S}_{j'}} \boldsymbol{x}_{i'} \right\|_2^2 - \frac{2}{m_j m_{j'}} \left(\sum_{\boldsymbol{x}_i \in \mathcal{S}_j} \boldsymbol{x}_i \right)^\top \left(\sum_{\boldsymbol{x}_{i'} \in \mathcal{S}_{j'}} \boldsymbol{x}_{i'} \right)$$

$$= \frac{1}{m_j^2} \left\| \sum_{\boldsymbol{x}_i \in \mathcal{S}_j} \boldsymbol{x}_i \right\|_2^2 + \frac{1}{m_{j'}^2} \left\| \sum_{\boldsymbol{x}_{i'} \in \mathcal{S}_{j'}} \boldsymbol{x}_{i'} \right\|_2^2 - \frac{2}{m_j m_{j'}} \sum_{\boldsymbol{x}_i \in \mathcal{S}_j, \boldsymbol{x}_{i'} \in \mathcal{S}_{j'}} \boldsymbol{x}_i^\top \boldsymbol{x}_{i'} \tag{4.67}$$

$$\leq \frac{1}{m_j} \sum_{\boldsymbol{x}_i \in \mathcal{S}_j} \|\boldsymbol{x}_i\|_2^2 + \frac{1}{m_{j'}} \sum_{\boldsymbol{x}_{i'} \in \mathcal{S}_{j'}} \|\boldsymbol{x}_{i'}\|_2^2 - \frac{2}{m_j m_{j'}} \sum_{\boldsymbol{x}_i \in \mathcal{S}_j, \boldsymbol{x}_{i'} \in \mathcal{S}_{j'}} \boldsymbol{x}_i^\top \boldsymbol{x}_{i'} \tag{4.68}$$

$$= \frac{1}{m_j m_{j'}} \sum_{\boldsymbol{x}_i \in \mathcal{S}_j, \boldsymbol{x}_{i'} \in \mathcal{S}_{j'}} \|\boldsymbol{x}_i - \boldsymbol{x}_{i'}\|_2^2 \tag{4.69}$$

$$\underbrace{+ \frac{m_{j'} - 1}{m_j m_{j'}} \sum_{\boldsymbol{x}_i \in \mathcal{S}_j} \|\boldsymbol{x}_i\|_2^2 + \frac{m_j - 1}{m_j m_{j'}} \sum_{\boldsymbol{x}_{i'} \in \mathcal{S}_{j'}} \|\boldsymbol{x}_{i'}\|_2^2 - \frac{1}{m_j m_{j'}} \sum_{\boldsymbol{x}_i \in \mathcal{S}_j, \boldsymbol{x}_{i'} \in \mathcal{S}_{j'}} \boldsymbol{x}_i^\top \boldsymbol{x}_{i'}}_{\doteq a}$$

$$\leq \frac{2}{m_j m_{j'}} \sum_{\boldsymbol{x}_i \in \mathcal{S}_j, \boldsymbol{x}_{i'} \in \mathcal{S}_{j'}} \|\boldsymbol{x}_i - \boldsymbol{x}_{i'}\|_2^2 \tag{4.70}$$

$$= \frac{2}{m_j m_{j'}} \textsc{assoc}(\mathcal{S}_j, \mathcal{S}_{j'}) \ . \tag{4.71}$$

Equation 4.68 exploits the fact that $\left(\sum_{j=1}^n a_j\right)^2 \leq n \left(\sum_{j=1}^n a_j^2\right)$ and Equation 4.70 is due to $a \leq (m_j m_{j'})^{-1} \sum_{\boldsymbol{x}_i \in \mathcal{S}_j, \boldsymbol{x}_{i'} \in \mathcal{S}_{j'}} \|\boldsymbol{x}_i - \boldsymbol{x}_{i'}\|_2^2$. We thus have:

$$\frac{\textsc{assoc}(\mathcal{S}_j, \mathcal{S}_j)}{\textsc{assoc}(\mathcal{S}_j, \mathcal{S}_j \cup \mathcal{S}_{j'})} = \frac{\textsc{assoc}(\mathcal{S}_j, \mathcal{S}_j)}{\textsc{assoc}(\mathcal{S}_j, \mathcal{S}_j) + \textsc{assoc}(\mathcal{S}_j, \mathcal{S}_{j'})} \tag{4.72}$$

$$\leq \frac{\textsc{assoc}(\mathcal{S}_j, \mathcal{S}_j)}{\textsc{assoc}(\mathcal{S}_j, \mathcal{S}_j) + \frac{m_j m_{j'}}{2} \|\boldsymbol{b}_j - \boldsymbol{b}_{j'}\|_2^2} \tag{4.73}$$

$$\leq \frac{\kappa' m_j}{\kappa' m_j + \frac{m_j m_{j'}}{2} \|\boldsymbol{b}_j - \boldsymbol{b}_{j'}\|_2^2} \tag{4.74}$$

$$= \frac{1}{1 + \frac{m_{j'}}{2\kappa'} \|\boldsymbol{b}_j - \boldsymbol{b}_{j'}\|_2^2} \ . \tag{4.75}$$

Equation 4.73 uses 4.71 and 4.74 uses assumption (**A4.3**). Equation 4.74 also holds when permuting j and j', so we get:

$$\begin{aligned}
\varsigma(V^{NC}, B^{\pm}) &\leq \max_{j \neq j'} \left(\frac{\varepsilon}{2n} + \frac{1}{1 + \frac{m_j}{2\kappa'} \|\boldsymbol{b}_j - \boldsymbol{b}_{j'}\|_2^2} + \frac{1}{1 + \frac{m_{j'}}{2\kappa'} \|\boldsymbol{b}_j - \boldsymbol{b}_{j'}\|_2^2} \right)^2 \|B^{\pm}\|_F \\
&\leq \left(\frac{\varepsilon}{2n} + \frac{1}{1 + \frac{\min_j m_j}{2\kappa'} \min_{j,j'} \|\boldsymbol{b}_j - \boldsymbol{b}_{j'}\|_2^2} \right)^2 \|B^{\pm}\|_F && (4.76) \\
&\leq \left(\frac{\varepsilon^2}{2n^2} + 2 \left(\frac{1}{1 + \frac{\min_j m_j}{2\kappa'} \min_{j,j'} \|\boldsymbol{b}_j - \boldsymbol{b}_{j'}\|_2^2} \right)^2 \right) \|B^{\pm}\|_F && (4.77) \\
&\leq \frac{\varepsilon^2}{2n^2} d \max_{\sigma,j} \|\boldsymbol{b}_j^{\sigma}\|_2 + \frac{4\kappa' d \max_{\sigma,j} \|\boldsymbol{b}_j^{\sigma}\|_2}{\min_{j,j'}^2 \|\boldsymbol{b}_j - \boldsymbol{b}_{j'}\|_2^2} && (4.78) \\
&\leq \frac{\varepsilon^2}{2n^2} d \max_{\sigma,j} \|\boldsymbol{b}_j^{\sigma}\|_2 + \frac{4\kappa' d}{\kappa^2 \max_{\sigma,j} \|\boldsymbol{b}_j^{\sigma}\|_2} && (4.79) \\
&= f^{NC} \left(\max_{\sigma,j} \|\boldsymbol{b}_j^{\sigma}\|_2 \right) && (4.80) \\
&= o(1) \ , && (4.81)
\end{aligned}$$

where the last inequality uses assumption (**A4.2**), and 4.77 uses the property that $(a+b)^2 \leq 2a^2 + 2b^2$. We have let:

$$f^{NC}(x) \doteq \frac{\varepsilon^2}{2n^2} dx + \frac{4\kappa' d}{\kappa x} \ , \tag{4.82}$$

which is indeed $o(1)$ if $\varepsilon = o(n^2/\sqrt{x})$. This proves the Lemma for $\varsigma(V^{NC}, B^{\pm})$. The case of $\varsigma(V^{G,s}, B^{\pm})$ is easier, as:

$$\begin{aligned}
\exp\left(-\frac{\|\boldsymbol{b}_j - \boldsymbol{b}_{j'}\|_2}{s} \right) &\leq \exp\left(-\frac{\min_{j'',j'''} \|\boldsymbol{b}_{j''} - \boldsymbol{b}_{j'''}\|_2}{s} \right) && (4.83) \\
&\leq \exp\left(-\frac{\kappa}{s} \max_{\sigma,j} \|\boldsymbol{b}_j^{\sigma}\|_2 \right) \ , && (4.84)
\end{aligned}$$

from assumption (**A4.2**) alone, which gives:

$$\varsigma(V^{G,s}, B^{\pm}) \leq \|B^{\pm}\|_F \left(\frac{\varepsilon}{2n} + \exp\left(-\frac{\kappa}{s}\max_{\sigma,j}\|\boldsymbol{b}_j^{\sigma}\|_2\right)\right)^2 \tag{4.85}$$

$$\leq \|B^{\pm}\|_F \left(\frac{\varepsilon^2}{2n^2} + 2\exp\left(-\frac{2\kappa}{s}\max_{\sigma,j}\|\boldsymbol{b}_j^{\sigma}\|_2\right)\right) \tag{4.86}$$

$$\leq d\max_{\sigma,j}\|\boldsymbol{b}_j^{\sigma}\|_2 \left(\frac{\varepsilon^2}{2n^2} + 2\exp\left(-\frac{2\kappa}{s}\max_{\sigma,j}\|\boldsymbol{b}_j^{\sigma}\|_2\right)\right) \tag{4.87}$$

$$= f^G\left(\max_{\sigma,j}\|\boldsymbol{b}_j^{\sigma}\|_2\right) \tag{4.88}$$

$$= o(1)\ , \tag{4.89}$$

as claimed. We have let $f^G(x) \doteq \frac{\varepsilon^2}{2n^2}dx + dx\exp(-2\kappa x/s)$, which is indeed $o(1)$ if $\varepsilon = o(n^2/\sqrt{x})$.

For the last statement, remark that we shall have in general $f^G(x) \leq f^{NC}(x)$ and even $f^G(x) = o(f^{NC}(x))$ if $\varepsilon = 0$, so we may expect better convergence in the case of $V^{G,s}$ as $\max_{\sigma,j}\|\boldsymbol{b}_j^{\sigma}\|_2$ grows.

4.11.4 Proof of Lemma 38

We first restate the Lemma in a more explicit way, that shall provide explicit values for κ_l and κ_n.

Lemma 48. *There exist $\kappa_{jj'}$ and $s_{jj'}$ depending on $d_j, d_{j'}$, and $\kappa'_{jj'} > 1$ depending on $m_j, m_{j'}$, such that:*

- *If $v_{jj'}^{G,s_{jj'}} > \exp(-1/4)$ then $\mathcal{S}_j, \mathcal{S}_{j'}$ are not linearly separable;*
- *If $v_{jj'}^{G,s_{jj'}} < \exp(-64)$ then $\mathcal{S}_j, \mathcal{S}_{j'}$ are linearly separable;*
- *If $v_{jj'}^{NC} > \kappa_{jj'}$ then $\mathcal{S}_j, \mathcal{S}_{j'}$ are not linearly separable;*
- *If $v_{jj'}^{NC} < \kappa_{jj'}/\kappa'_{jj'}$ then $\mathcal{S}_j, \mathcal{S}_{j'}$ are linearly separable.*

Proof We first consider the normalized association criterion in 4.21, and we prove the Lemma for the following expressions of $\kappa_{jj'}$ and $\kappa'_{jj'}$:

$$\kappa_{jj'} \doteq \frac{16}{2 + \frac{d_{jj'}^2}{2d_{j'}^2}} + \frac{16}{2 + \frac{d_{jj'}^2}{2d_j^2}}\ , \tag{4.90}$$

$$\kappa'_{jj'} \doteq 512\max\{m_j, m_{j'}\}\ , \tag{4.91}$$

with $d_{jj'} \doteq \max\{d_j, d_{j'}\}$ and $d_j \doteq \max_{\boldsymbol{x},\boldsymbol{x}'\in\mathcal{S}_j}\|\boldsymbol{x} - \boldsymbol{x}'\|_2$, $\forall j \neq j' \in [n]$. For any bag $\mathcal{S}_j$, we let $(\boldsymbol{b}_j^{\star}, r_j) \doteq MEB(\mathcal{S}_j)$ denote the minimum enclosing ball (MEB) for bag $\mathcal{S}_j$ and

distance L_2, that is, r_j is the smallest unique real such that:

$$\exists!\ \boldsymbol{b}_j^\star : d(\boldsymbol{x}, \boldsymbol{b}_j^\star) \doteq \|\boldsymbol{x} - \boldsymbol{b}_j^\star\|_2 \leq r_j, \forall \boldsymbol{x} \in \mathcal{S}_j\ . \tag{4.92}$$

We have let $d(\boldsymbol{x}, \boldsymbol{b}_j^\star) \doteq \|\boldsymbol{x} - \boldsymbol{b}_j^\star\|_2$. We are going to prove a first result involving the MEBs of $\mathcal{S}_j$ and $\mathcal{S}_{j'}$, and then will translate the result to the Lemma's statement. The following properties follows from standard properties of MEBs and the fact that $d(.,.)$ is a distance (they hold for any $j \neq j'$):

(a) $d(\boldsymbol{x}, \boldsymbol{x}') \leq 2r_j\ , \forall \boldsymbol{x}, \boldsymbol{x}' \in \mathcal{S}_j$;

(b) If bags $\mathcal{S}_j$ and $\mathcal{S}_{j'}$ are linearly separable, then $\forall \boldsymbol{x} \in \mathrm{co}(\mathcal{S}_j)$, $\exists \boldsymbol{x}' \in \mathcal{S}_{j'}$ such that $d(\boldsymbol{x}, \boldsymbol{x}') \geq \max\{r_j, r_{j'}\}$; here, "co" denotes the convex closure;

(c) If bags $\mathcal{S}_j$ and $\mathcal{S}_{j'}$ are linearly separable, then $d(\boldsymbol{b}_j, \boldsymbol{b}_{j'}) \geq \max\{r_j, r_{j'}\}$, where $\boldsymbol{b}_j$ and $\boldsymbol{b}_{j'}$ are the bags average;

(d) $\forall \boldsymbol{x} \in \mathcal{S}_j, \exists \boldsymbol{x}' \in \mathcal{S}_j$ s.t. $d(\boldsymbol{x}, \boldsymbol{x}') \geq r_j$;

(e) $d(\boldsymbol{x}, \boldsymbol{x}') \leq 2\max\{r_j, r_{j'}\} + d(\boldsymbol{b}_j^\star, \boldsymbol{b}_{j'}^\star), \forall \boldsymbol{x} \in \mathrm{co}(\mathcal{S}_j), \forall \boldsymbol{x}' \in \mathrm{co}(\mathcal{S}_{j'})$.

Let us define:

$$\textsc{assoc}(\mathcal{S}_j, \mathcal{S}_{j'}) \doteq \sum_{\boldsymbol{x} \in \mathcal{S}_j, \boldsymbol{x}' \in \mathcal{S}_{j'}} d^2(\boldsymbol{x}, \boldsymbol{x}')\ . \tag{4.93}$$

We remark that, assuming that each bag contains at least two elements without loss of generality:

$$v_{jj'}^{NC} = \frac{1}{2}\left(\frac{1}{1 + \frac{\textsc{assoc}(\mathcal{B}_j, \mathcal{B}_{j'})}{\textsc{assoc}(\mathcal{B}_j, \mathcal{B}_j)}} + \frac{1}{1 + \frac{\textsc{assoc}(\mathcal{B}_j, \mathcal{B}_{j'})}{\textsc{assoc}(\mathcal{B}_{j'}, \mathcal{B}_{j'})}}\right)\ . \tag{4.94}$$

We have $\textsc{assoc}(\mathcal{S}_j, \mathcal{S}_j) \leq 4m_j r_j^2$ and $\textsc{assoc}(\mathcal{S}_{j'}, \mathcal{S}_{j'}) \leq 4m_{j'} r_{j'}^2$ (because of (a)), and also $\textsc{assoc}(\mathcal{S}_j, \mathcal{S}_{j'}) \geq \max\{m_j, m_{j'}\}\max\{r_j^2, r_{j'}^2\}$ when $\mathcal{S}_j$ and $\mathcal{S}_{j'}$ are linearly separable (because of (b)), which yields in this case:

$$v_{jj'}^{NC} \leq \frac{1}{2 + \frac{\max\{m_j, m_{j'}\}\max\{r_j^2, r_{j'}^2\}}{2m_j r_j^2}} + \frac{1}{2 + \frac{\max\{m_j, m_{j'}\}\max\{r_j^2, r_{j'}^2\}}{2m_{j'} r_{j'}^2}} \tag{4.95}$$

$$\leq \frac{1}{2 + \frac{\max\{r_j^2, r_{j'}^2\}}{2r_j^2}} + \frac{1}{2 + \frac{\max\{r_j^2, r_{j'}^2\}}{2r_{j'}^2}}\ . \tag{4.96}$$

Let us name $\kappa_{jj'}^\circ$ the right-hand side of 4.96. It follows that when $v_{jj'}^{NC} > \kappa_{jj'}^\circ$, $\mathcal{S}_j$ and $\mathcal{S}_{j'}$ are not linearly separable.

On the other hand, we have $\textsc{assoc}(\mathcal{S}_j, \mathcal{S}_j) \geq m_j r_j^2$ and $\textsc{assoc}(\mathcal{S}_{j'}, \mathcal{S}_{j'}) \geq m_{j'} r_{j'}^2$ (because of (d)), and also:

$$\textsc{assoc}(\mathcal{S}_j, \mathcal{S}_{j'}) \leq m_j m_{j'} (2\max\{r_j, r_{j'}\} + d(\boldsymbol{b}_j^\star, \boldsymbol{b}_{j'}^\star))^2 \tag{4.97}$$

$$\leq m_j m_{j'} (4\max\{r_j^2, r_{j'}^2\} + 2d^2(\boldsymbol{b}_j^\star, \boldsymbol{b}_{j'}^\star)) \ , \tag{4.98}$$

because of (e) and the fact that $(a+b)^2 \leq 2a^2 + 2b^2$. It follows that $\forall j \neq j'$:

$$v_{jj'}^{NC} \geq \frac{1}{2 + \frac{2m_{j'}(4\max\{r_j^2, r_{j'}^2\} + 2d^2(\boldsymbol{b}_j^\star, \boldsymbol{b}_{j'}^\star))}{r_j^2}} + \frac{1}{2 + \frac{2m_j(4\max\{r_j^2, r_{j'}^2\} + 2d^2(\boldsymbol{b}_j^\star, \boldsymbol{b}_{j'}^\star))}{r_{j'}^2}} \ . \tag{4.99}$$

For any $j \neq j'$, when $d^2(\boldsymbol{b}_j^\star, \boldsymbol{b}_{j'}^\star) \leq 4\max\{r_j^2, r_{j'}^2\}$, then we have from 4.99:

$$v_{jj'}^{NC} \geq \frac{1}{2 + \frac{16 m_{j'} \max\{r_j^2, r_{j'}^2\}}{r_j^2}} + \frac{1}{2 + \frac{16 m_j \max\{r_j^2, r_{j'}^2\}}{r_{j'}^2}} \tag{4.100}$$

$$> \kappa_{jj'}^\circ / (32\max\{m_j, m_{j'}\}) \ . \tag{4.101}$$

Hence, $v_{jj'}^{NC} \leq \kappa_{jj'}^\circ / (32\max\{m_j, m_{j'}\})$ implies $d(\boldsymbol{b}_j^\star, \boldsymbol{b}_{j'}^\star) > 2\max\{r_j, r_{j'}\}$, implying $d(\boldsymbol{b}_j^\star, \boldsymbol{b}_{j'}^\star) > r_j + r_{j'}$, which is a sufficient condition for the linear separability of $\mathcal{S}_j$ and $\mathcal{S}_{j'}$.

We can relate the linear separability of $\mathcal{S}_j$ and $\mathcal{S}_{j'}$ to the value of $v_{jj'}^{NC}$ with respect to $\kappa_{jj'}^\circ$ defined in 4.96. To remove the dependence in the MEB parameters and obtain the statement of the Lemma, we just have to remark that $d_j^2/4 \leq r_j^2 \leq 4d_j^2, \forall j \in [n]$, which yields $\kappa_{jj'}/16 \leq \kappa_{jj'}^\circ \leq \kappa_{jj'}$. Therefore, when $v_{jj'}^{NC} > \kappa_{jj'}$, it follows that $v_{jj'}^{NC} > \kappa_{jj'}^\circ$ and $\mathcal{S}_j$ and $\mathcal{S}_{j'}$ are not linearly separable. On the other hand, when $v_{jj'}^{NC} \leq \kappa_{jj'}/(16 \times 32\max\{m_j, m_{j'}\}) = \kappa_{jj'}/\kappa'_{jj'}$, then $v_{jj'}^{NC} \leq \kappa_{jj'}^\circ / (32\max\{m_j, m_{j'}\})$ and the bags $\mathcal{S}_j$ and $\mathcal{S}_{j'}$ are linearly separable. This achieves the proof of Lemma 38 for the normalized association criterion in 4.21.

The proof for $v_{jj'}^{G,s}$ is shorter, and we prove it for:

$$s_{j,j'} = \max\{d_j, d_{j'}\} \ . \tag{4.102}$$

We have $(1/2)\max\{d_j, d_{j'}\} \leq \max\{r_j, r_{j'}\} \leq 2\max\{d_j, d_{j'}\}$. Hence, because of (c) above, if $\mathcal{S}_j$ and $\mathcal{S}_{j'}$ are linearly separable, then $v_{jj'}^{G,s} \leq 1/e^{1/4}$; so, when $v_{jj'}^{G,s} > 1/e^{1/4}$, the two bags are not linearly separable. On the other hand, if $d(\boldsymbol{b}_j^\star, \boldsymbol{b}_{j'}^\star) \leq 2\max\{r_j, r_{j'}\}$, then because of (e) above $d(\boldsymbol{b}_j, \boldsymbol{b}_{j'}) \leq 4\max\{r_j, r_{j'}\} \leq 8\max\{d_j, d_{j'}\}$, and so $v_{jj'}^{G,s} \geq 1/e^{64}$. This implies that if $v_{jj'}^{G,s} < 1/e^{64}$, then $d(\boldsymbol{b}_j^\star, \boldsymbol{b}_{j'}^\star) > 2\max\{r_j, r_{j'}\} \geq r_j + r_{j'}$, and thus the two bags are linearly separable, as claimed. This achieves the proof of Lemma 48. ■

This achieves the proof of Lemma 38.

4.11.5 Proof of lemma 39

Let $x > 0, \epsilon \in (0,1), p \in (0,1), p \neq 1/2$. We create a dataset from four observations, $\{(\boldsymbol{x}_1 = 0, 1), (\boldsymbol{x}_2 = 0, -1), (\boldsymbol{x}_3 = x, 1), (\boldsymbol{x}_4 = x, -1)\}$. There are two bags, $\mathcal{S}_1$ takes $1-\epsilon$ of $\boldsymbol{x}_2$ and ϵ of $\boldsymbol{x}_1$. $\mathcal{S}_2$ takes ϵ of $\boldsymbol{x}_4$ and $1-\epsilon$ of $\boldsymbol{x}_3$. The label-wise estimators $\hat{\boldsymbol{b}}^\sigma$ of Quadrianto et al. [2009] are solution of:

$$\begin{bmatrix} \hat{\boldsymbol{b}}^+ \\ \hat{\boldsymbol{b}}^- \end{bmatrix} = \left(\begin{bmatrix} 1-\epsilon & \epsilon \\ \epsilon & 1-\epsilon \end{bmatrix}^\top \begin{bmatrix} 1-\epsilon & \epsilon \\ \epsilon & 1-\epsilon \end{bmatrix} \right)^{-1} \begin{bmatrix} 1-\epsilon & \epsilon \\ \epsilon & 1-\epsilon \end{bmatrix}^\top \begin{bmatrix} x \\ 0 \end{bmatrix} \tag{4.103}$$

$$= \frac{1}{1-2\epsilon} \begin{bmatrix} (1-\epsilon)x \\ \epsilon x \end{bmatrix} \tag{4.104}$$

On the other hand, the true quantities are:

$$\begin{bmatrix} \boldsymbol{b}^+ \\ \boldsymbol{b}^- \end{bmatrix} = \begin{bmatrix} (1-\epsilon)x \\ \epsilon x \end{bmatrix} . \tag{4.105}$$

We now mix classes in $\mathcal{S}$ and pick bag proportions $q \doteq \mathbb{P}_{\mathcal{S}}[\boldsymbol{x} \in \mathcal{S}_1]$ and $1-q = \mathbb{P}_{\mathcal{S}}[\boldsymbol{x} \in \mathcal{S}_2]$. We have the class proportions defined by $\mathbb{P}_{\mathcal{S}}[y = +1] = \epsilon q + (1-\epsilon)(1-q) \doteq p$. Then:

$$\|\hat{\boldsymbol{\mu}} - \boldsymbol{\mu}\|_2 = \left\| p(1-\epsilon) \left(\frac{1}{1-2\epsilon} - 1 \right) x - (1-p)\epsilon \left(\frac{1}{1-2\epsilon} - 1 \right) x \right\|_2 \tag{4.106}$$

$$= \frac{2\epsilon|p-\epsilon|}{1-2\epsilon} x \tag{4.107}$$

$$= 2\epsilon(1-q)x . \tag{4.108}$$

Furthermore, $\max_i \left\|\boldsymbol{b}_i^\sigma\right\|_2 = x$. We get:

$$\frac{\|\hat{\boldsymbol{\mu}} - \boldsymbol{\mu}\|_2}{\max_i \left\|\boldsymbol{b}_i^\sigma\right\|_2} = 2\epsilon(1-q) . \tag{4.109}$$

Picking ϵ and $(1-q)$ both $> \sqrt{1-(\gamma/2)}$ is sufficient to have Equation 4.109 $> 2-\gamma$ for any $\gamma > 0$. Remark that both assumptions (**A4.2**) and (**A4.3**) hold for any $\kappa < 1$ and any $\kappa' > 0$.

4.11.6 Proof of Theorem 41

Let us start by recalling some notation.

$$R_{\mathcal{S}_X,\ell}(\boldsymbol{\theta},\boldsymbol{\mu}) = \frac{1}{2m}\left(\sum_{i\in[m]}\sum_{\sigma}\ell(\sigma\langle\boldsymbol{\theta},\boldsymbol{x}_i\rangle)\right) + a\langle\boldsymbol{\theta},\boldsymbol{\mu}\rangle \ . \tag{4.110}$$

Also define the regularized loss:

$$R_{\mathcal{S}_X,\ell}(\boldsymbol{\theta},\boldsymbol{\mu},\lambda) \doteq R_{\mathcal{S}_X,\ell}(\boldsymbol{\theta},\boldsymbol{\mu}) + \lambda\|\boldsymbol{\theta}\|_2^2 \ . \tag{4.111}$$

The proof builds upon Lemma 26, which bound is refined for SPLS. First, we prove a helper Lemma for Lipschitz losses.

Lemma 49. *Let ℓ be L-Lipschitz and a-LOL. Fix $\lambda > 0$, and let $X \doteq \max_i \|\boldsymbol{x}_i\|_2$. Let $\boldsymbol{\theta}' \doteq \operatorname{argmin}_{\boldsymbol{\theta}} R_{\mathcal{S}_X,\ell}(\boldsymbol{\theta},\boldsymbol{\mu}',\lambda)$ where $\boldsymbol{\mu}'$ is any vector in $\mathbb{R}^d$. Then:*

$$\|\boldsymbol{\theta}'\|_2 \leq \frac{LX + |a|\|\boldsymbol{\mu}'\|_2}{\lambda} \ . \tag{4.112}$$

Proof Let us define a shrinking of the optimal solution $\boldsymbol{\theta}'$, $\boldsymbol{\theta}_\alpha \doteq \alpha\boldsymbol{\theta}'$ for $\alpha \in (0,1)$. We have:

$$\begin{aligned}
R_{\mathcal{S}_X,\ell}(\boldsymbol{\theta}_\alpha,\boldsymbol{\mu}',\lambda) &= \frac{1}{2m}\left(\sum_i\sum_\sigma \ell(\sigma\langle\boldsymbol{\theta}_\alpha,\boldsymbol{x}_i\rangle)\right) + a\langle\boldsymbol{\theta}_\alpha,\boldsymbol{\mu}'\rangle + \lambda\|\boldsymbol{\theta}_\alpha\|_2^2 && (4.113)\\
&= \frac{1}{2m}\left(\sum_i\sum_\sigma \ell(\sigma\alpha\langle\boldsymbol{\theta}',\boldsymbol{x}_i\rangle)\right) + a\alpha\langle\boldsymbol{\theta}',\boldsymbol{\mu}'\rangle + \lambda\alpha^2\|\boldsymbol{\theta}'\|_2^2 && (4.114)\\
&\leq \frac{1}{2m}\left(\sum_i\sum_\sigma \ell(\sigma\langle\boldsymbol{\theta}',\boldsymbol{x}_i\rangle) + L\left|\sigma\alpha\langle\boldsymbol{\theta}',\boldsymbol{x}_i\rangle - \sigma\langle\boldsymbol{\theta}',\boldsymbol{x}_i\rangle\right|\right)\\
&\quad + a\alpha\langle\boldsymbol{\theta}',\boldsymbol{\mu}'\rangle + \lambda\alpha^2\|\boldsymbol{\theta}'\|_2^2 && (4.115)\\
&= \frac{1}{2m}\left(\sum_i\sum_\sigma \ell(\sigma\langle\boldsymbol{\theta}',\boldsymbol{x}_i\rangle)\right) + \frac{L(1-\alpha)}{m}\sum_i|\langle\boldsymbol{\theta}',\boldsymbol{x}_i\rangle|\\
&\quad + a\alpha\langle\boldsymbol{\theta}',\boldsymbol{\mu}'\rangle + \lambda\alpha^2\|\boldsymbol{\theta}'\|_2^2 \ , && (4.116)
\end{aligned}$$

where 4.115 holds because ℓ is L-Lipschitz. To have Equation 4.116 smaller than $R_{\mathcal{S}_X,\ell}(\boldsymbol{\theta}',\boldsymbol{\mu}',\lambda)$, we need equivalently:

$$\frac{L(1-\alpha)}{m}\sum_i|\langle\boldsymbol{\theta}',\boldsymbol{x}_i\rangle| + a\alpha\langle\boldsymbol{\theta}',\boldsymbol{\mu}'\rangle + \lambda\alpha^2\|\boldsymbol{\theta}'\|_2^2 \leq a\langle\boldsymbol{\theta}',\boldsymbol{\mu}'\rangle + \lambda\|\boldsymbol{\theta}'\|_2^2 \ , \tag{4.117}$$

that is:

$$\frac{L(1-\alpha)}{m}\sum_i|\langle\boldsymbol{\theta}',\boldsymbol{x}_i\rangle| - a(1-\alpha)\langle\boldsymbol{\theta}',\boldsymbol{\mu}'\rangle \leq \lambda(1-\alpha^2)\|\boldsymbol{\theta}'\|_2^2 \ , \tag{4.118}$$

and to find an $\alpha \in (0,1)$ such that this holds, because of Cauchy-Schwartz inequality, it is sufficient that:

$$(1-\alpha)(LX - a\boldsymbol{\mu}') \leq (1-\alpha)(LX + |a|\boldsymbol{\mu}') \leq \lambda(1-\alpha^2)\|\boldsymbol{\theta}'\|_2 \ , \tag{4.119}$$

that is

$$\|\boldsymbol{\theta}'\|_2 \geq \frac{LX + |a|\|\boldsymbol{\mu}'\|_2}{\lambda(1+\alpha)} \ . \tag{4.120}$$

Finally, whenever $\|\boldsymbol{\theta}'\|_2 > (LX + |a|\|\boldsymbol{\mu}'\|_2)/\lambda$, there is a shrinking of the optimal solution of Equation 4.110 that further decreases the empirical risk, thus contradicting its optimality. This ends the proof of Lemma 49. ■

We now combine Theorem 26 and Lemma 49. Recall that we assume SPLS. Since they are differentiable by definition, we can verify that they are also Lipschitz by taking the first derivative. We have:

$$\ell'(x) = -\frac{1}{b_\phi}(\phi^\star)'(-x) = -\frac{1}{b_\phi}(\phi')^{-1}(-x) \in [-1/b_\phi, 0] \ , \tag{4.121}$$

for any $x \in \phi'([0,1])$ and thus ℓ is $1/b_\phi$-Lipschitz. Also, we know that $a = -\frac{1}{2b_\phi}$ with $b_\phi > 0$ for SPLS (Table 3.1). Therefore, Lemma 49 tells us that:

$$\|\boldsymbol{\theta}'\|_2 \leq \frac{X + \|\boldsymbol{\mu}'\|_2}{b_\phi \lambda} \ . \tag{4.122}$$

We can obtain an explicit value instead of the generic γ denoting strong convexity in Theorem 26. Let us compute the second derivate for any SPL. By the rule of derivate of an inverse function, we differentiate Equation 4.121:

$$\ell''(x) = -\frac{1}{b_\phi}\left((\phi')^{-1}\right)'(-x) \tag{4.123}$$

$$= -\frac{1}{b_\phi \ \phi''\left((\phi')^{-1}(-x)\right)} \tag{4.124}$$

Recall Equation 3.100 in the proof of Theorem 26. We compute a lower bound of the second derivative ℓ'' by bounding its argument for any $\alpha \in [0,1]$:

$$\left|\pm\langle \alpha\boldsymbol{\theta}^\star + (1-\alpha)\hat{\boldsymbol{\theta}}, \boldsymbol{x}_i\rangle\right| \leq (\alpha\|\boldsymbol{\theta}^\star\|_2 + (1-\alpha)\|\hat{\boldsymbol{\theta}}\|_2)X \tag{4.125}$$

$$\leq \frac{X + \alpha\|\boldsymbol{\mu}\|_2 + (1-\alpha)\|\hat{\boldsymbol{\mu}}\|_2}{b_\phi\lambda} \tag{4.126}$$

$$\leq \frac{X + \max\{\|\boldsymbol{\mu}\|_2, \|\hat{\boldsymbol{\mu}}\|_2\}}{b_\phi\lambda} \ , \tag{4.127}$$

where Inequality 4.125 follows from Cauchy-Schwartz inequality. 4.126 uses Equa-

tion 4.122. $\boldsymbol{\mu}$ and $\hat{\boldsymbol{\mu}}$ are the parameters of $R_{\mathcal{S}_X,\ell}(\cdot,\boldsymbol{\mu},\lambda)$ and $R_{\mathcal{S}_X,\ell}(\cdot,\hat{\boldsymbol{\mu}},\lambda)$ in Lemma 26, and $\hat{\boldsymbol{\theta}}$ and $\boldsymbol{\theta}^\star$ the respective minimizers. This achieves the proof of Theorem 41 for the interval:

$$[\pm(X + \max\{\|\boldsymbol{\mu}\|_2, \|\hat{\boldsymbol{\mu}}\|_2\})] \ . \tag{4.128}$$

4.11.7 Proof of Lemma 42

We elaborate the proof for optimization strategy $\text{OPT} = \min$. The case $\text{OPT} = \max$ flips the choice of the label in Step 1. For simplicity, let us define, as in the last proof:

$$R_{\mathcal{S}_X,\ell}(\boldsymbol{\theta},\boldsymbol{\mu}) = \frac{1}{2m}\left(\sum_{i\in[m]}\sum_{\sigma}\ell(\sigma\langle\boldsymbol{\theta},\boldsymbol{x}_i\rangle)\right) + a\langle\boldsymbol{\theta},\boldsymbol{\mu}\rangle \ . \tag{4.129}$$

To minimize $R_{\mathcal{S}_X,\ell}(\boldsymbol{\theta}_t,\boldsymbol{\mu}(\boldsymbol{\sigma}))$ over $\boldsymbol{\sigma}\in\Sigma_{\boldsymbol{\pi}}$, we just have to find:

$$\boldsymbol{\sigma}_* \in \operatorname*{argmax}_{\boldsymbol{\sigma}\in\Sigma_{\boldsymbol{\pi}}}\left\langle\boldsymbol{\theta},\sum_i\sigma_i\boldsymbol{x}_i\right\rangle, \tag{4.130}$$

and we can do that bag by bag. Algorithm 7 presents the labeling (notation $(m) \doteq \{1,2,...,m-1\}$). Remark that the time complexity for one bag is $O(m_j\log m_j)$ due to the ordering (Step 1), so the overall complexity is indeed $O(m\max_i\log m_i)$.

Lemma 50. *Let $\boldsymbol{\sigma}_* \doteq \{\sigma_1^*,\sigma_2^*,...,\sigma_m^*\}$ be the set of labels obtained after running LA($\boldsymbol{\theta},\mathcal{S}_j,m_j^+$) for $j=1,2,...,n$. Then $\boldsymbol{\sigma}_*\in\operatorname{argmax}_{\boldsymbol{\sigma}\in\Sigma_{\boldsymbol{\pi}}}\boldsymbol{\theta}^\top\sum_i\sigma_i\boldsymbol{x}_i$.*

Proof The total edge, $\langle\boldsymbol{\theta},\sum_i\sigma_i\boldsymbol{x}_i\rangle$ (for any $\boldsymbol{\sigma}\in\Sigma_{\boldsymbol{\pi}}$), can be summable bag-wise with regard to the coordinates of $\boldsymbol{\sigma}$. Consider the optimal set:

$$\{\boldsymbol{\sigma}^\star\}_{\mathcal{B}} \doteq \operatorname*{argmax}_{\boldsymbol{\sigma}\in\{-1,1\}^{m'}:\ \langle\mathbf{1},\boldsymbol{\sigma}\rangle=2m^+-m'}\left\langle\boldsymbol{\theta},\sum\nolimits_{\boldsymbol{x}_i\in\mathcal{B}}\sigma_i\boldsymbol{x}_i\right\rangle \ ,$$

for some bag $\mathcal{B}=\{\boldsymbol{x}_i, i=1,2,...,m'\}$, with constraint $m^+\in[m']$. This set contains the label assignment $\boldsymbol{\sigma}_*$ returned by Algorithm 7 LA($\boldsymbol{\theta},\mathcal{B},m^+$), a property that follows from two simple observations:

1. Consider any observation $\boldsymbol{x}_i$ of bag $\mathcal{B}$; for any optimal labeling $\boldsymbol{\sigma}^\star$ of $\mathcal{B}$, let $m'^+ \doteq m^+ - \mathrm{I}(\sigma_i^\star=1)$. Define the set $\{\boldsymbol{\sigma}'^\star\}_i$ of optimal labelings of $\mathcal{B}\backslash\{\boldsymbol{x}_i\}$ with constraint $m'^+ \doteq m^+ - \mathrm{I}(\sigma_i^\star=1)$. Then this set coincides with the set created by taking the elements of $\{\boldsymbol{\sigma}^\star\}_{\mathcal{B}}$ to which we drop coordinate i. This follows from the per-observation summability of the total edge with regard to labels.

2. Assume $m^+\in(m')$. $\forall i^*\in\operatorname{argmax}_i|\langle\boldsymbol{\theta},\boldsymbol{x}_i\rangle|$, there exists an optimal assignment $\boldsymbol{\sigma}^\star$ such that $\sigma_{i^*}^\star = \operatorname{sign}\langle\boldsymbol{\theta},\boldsymbol{x}_{i^*}\rangle$. Otherwise, starting from any optimal assignment $\boldsymbol{\sigma}^\star$, we can flip the label of $\boldsymbol{x}_{i^*}$ and the label of any other $\boldsymbol{x}_i$ for which

Algorithm 7: Label Assignation (LA)

Input: $\boldsymbol{\theta} \in \mathbb{R}^d$, a bag $\mathcal{B} = \{\boldsymbol{x}_i \in \mathbb{R}^d, i = 1, 2, ..., m\}$, bag size $m^+ \in [m]$
If $\mathcal{B} = \emptyset$ **then** stop
Else if $m^+ \notin (m)$
    $y_i \leftarrow \mathrm{I}(m^+ = m) - \mathrm{I}(m^+ = 0), \forall i = 1, 2, ..., m$
Else
    1 $i^* \leftarrow \mathrm{argmax}_i \, |\langle \boldsymbol{\theta}, \boldsymbol{x}_i \rangle|$
    2 $y_{i^*} \leftarrow \mathrm{sign}\langle \boldsymbol{\theta}, \boldsymbol{x}_{i^*} \rangle$
    3 $\mathrm{LA}(\boldsymbol{\theta}, \mathcal{B} \backslash \{\boldsymbol{x}_{i^*}\}, m^+ - \mathrm{I}(y_{i^*} = 1))$

$\sigma_i^\star \neq \sigma_{i^*}^\star$, and get a label assignment that satisfies constraint m^+ and cannot be worse than $\boldsymbol{\sigma}^\star$, and is thus optimal, a contradiction.

Hence, $\mathrm{LA}(\boldsymbol{\theta}, \mathcal{B}, m^+)$ picks at each iteration a label that matches one in a subset of optimal labelings, and the recursive call preserves the subset of optimal labelings. Since when $m^+ \notin (m)$ the solution returned by $\mathrm{LA}(\boldsymbol{\theta}, \mathcal{B}, m^+)$ is obviously optimal, we end up when the current $\mathcal{B}$ is empty with $\boldsymbol{\sigma}_* \in \mathrm{argmax}_{\sigma \in \Sigma_\pi} \langle \boldsymbol{\theta}, \sum_i \sigma_i \boldsymbol{x}_i \rangle$, as claimed. ■

4.11.8 Proof of Theorem 43

We have to combine Theorems 23 (first statement) and 36. The link between the two is the following bound on the norm discrepancy between the population mean operator and its estimator by LMM:

$$\|\boldsymbol{\mu}_\mathcal{D} - \hat{\boldsymbol{\mu}}_\mathcal{S}\|_2 = \|\boldsymbol{\mu}_\mathcal{D} + \boldsymbol{\mu}_\mathcal{S} - \boldsymbol{\mu}_\mathcal{S} - \hat{\boldsymbol{\mu}}_\mathcal{S}\|_2 \tag{4.131}$$

$$\leq \|\boldsymbol{\mu}_\mathcal{D} - \boldsymbol{\mu}_\mathcal{S}\|_2 + \|\boldsymbol{\mu}_\mathcal{S} - \hat{\boldsymbol{\mu}}_\mathcal{S}\|_2 \ . \tag{4.132}$$

Here we are interested in bounding the latter norm in Step 4.132, since we can treat the former one as done in the proof of Theorem 23. Let us denote $\eta \doteq \boldsymbol{\mu}_\mathcal{S} - \hat{\boldsymbol{\mu}}_\mathcal{S} \in \mathbb{R}^d$ and similarly $\eta_j, \eta_j^y \in \mathbb{R}^d$ the difference of bag-wise and bag-wise, label-wise mean operators respectively. By Jensen's inequality and the fact that $p(j) \leq 1$ and $\pi_j \leq$

1, $\forall j$ we have:

$$\|\eta\|_2^2 = \left\| \sum_{j\in[n]} p(j)\eta_j \right\|_2^2 \tag{4.133}$$

$$= \left\| \sum_{j\in[n]} p(j) \left(\pi_j \eta_j^+ - (1-\pi_j)\eta_j^- \right) \right\|_2^2 \tag{4.134}$$

$$\leq \sum_{j\in[n]} p(j) \left\| \pi_j \eta_j^+ - (1-\pi_j)\eta_j^- \right\|_2^2 \tag{4.135}$$

$$\leq \sum_{j\in[n]} \left\| \pi_j \eta_j^+ - (1-\pi_j)\eta_j^- \right\|_2^2 \tag{4.136}$$

$$\leq \sum_{j\in[n]} \sum_{y\in\mathcal{Y}} \left\| y \cdot \eta_j^y \right\|_2^2 \tag{4.137}$$

$$= \sum_{j\in[n]} \sum_{y\in\mathcal{Y}} \sum_{k\in[d]} \left| \left(\eta_j^y\right)^k \right|^2 \tag{4.138}$$

$$= \left\| M - \hat{M} \right\|_F^2 \tag{4.139}$$

where $M, \hat{M}$ are defined in Theorem 36. Therefore, by Theorem 36 and taking $\varepsilon = 0$:

$$\|\boldsymbol{\mu}_{\mathcal{D}} - \hat{\boldsymbol{\mu}}_{\mathcal{S}}\|_2 \leq \|\boldsymbol{\mu}_{\mathcal{D}} - \boldsymbol{\mu}_{\mathcal{S}}\|_2 + \left\| M - \hat{M} \right\|_F \tag{4.140}$$

$$\leq \|\boldsymbol{\mu}_{\mathcal{D}} - \boldsymbol{\mu}_{\mathcal{S}}\|_2 + \sqrt{n/2} \cdot \frac{\left(\max_{j\neq j'} v_{jj'}\right)^2}{\min_j w_j^2} \|B^{\pm}\|_F \tag{4.141}$$

We bound:

$$\left\|B^{\pm}\right\|_F = \sqrt{\sum_{j,y} \left\| \boldsymbol{b}_j^y \right\|_2^2} \tag{4.142}$$

$$= \sqrt{\sum_{j,y} \left\| \frac{1}{m_j} \sum_{i\in\mathcal{S}_j:\ y_i=y} \boldsymbol{x}_i \right\|_2^2} \tag{4.143}$$

$$\leq \sqrt{\sum_{j,y} \frac{1}{m_j^2} \cdot m_j \sqrt{X}} \tag{4.144}$$

$$= \sqrt{2n\sqrt{X} \cdot \sum_j \frac{1}{m_j}} \tag{4.145}$$

and thus Equation 4.141 is upper bounded by:

$$\|\boldsymbol{\mu}_{\mathcal{D}} - \boldsymbol{\mu}_{\mathcal{S}}\|_2 + nX^{1/4}\frac{\left(\max_{j\neq j'} v_{jj'}\right)^2}{\min_j w_j^2} \cdot \sqrt{\sum_{j\in[n]} \frac{1}{m_j}} \ . \tag{4.146}$$

Therefore, under the assumptions of Theorem 23 it follows the excess risk is bounded with probability $1-\delta$ by:

$$\begin{aligned} R_{\mathcal{D},\ell}(\hat{\boldsymbol{\theta}}) - R_{\mathcal{D},\ell}(\boldsymbol{\theta}^\star) \leq \frac{\sqrt{2}+1}{2} \cdot \frac{XHL}{\sqrt{m}} + c(X,H) \cdot \sqrt{\frac{1}{m}\log\left(\frac{2}{\delta}\right)} \\ + 2|a|H\left(X\sqrt{\frac{d}{m}\log\left(\frac{2d}{\delta}\right)} + nX^{1/4}\frac{\left(\max_{j\neq j'} v_{jj'}\right)^2}{\min_j w_j^2}\sqrt{\sum_{j\in[n]}\frac{1}{m_j}}\right) \ . \end{aligned} \tag{4.147}$$

To obtain the Theorem notice:

$$\sum_{j\in[n]} \frac{1}{m_j} = \frac{n}{\mathrm{Harm}(m)} \ . \tag{4.148}$$

4.11.9 Proof of Theorem 46

We prove separately Equations 4.33 and 4.34.

4.11.9.1 Proof of Equation 4.33

Unless explicitly stated, all samples like $\mathcal{S}$ and $\mathcal{S}'$ are of size m. To make the reading of our expectations clear and simple, we shall write $\mathbb{E}_{\Sigma_m}$ for $\mathbb{E}_{\sigma\sim\Sigma_m}$, $\mathbb{E}_{\mathcal{S}}$ for $\mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{S}}$, $\mathbb{E}_{\mathcal{D}'_m}$ for $\mathbb{E}_{\mathcal{S}'\sim\mathcal{D}}$ and $\mathbb{E}_{\mathcal{D}_m}$ for $\mathbb{E}_{\mathcal{S}\sim\mathcal{D}}$. We now proceed to the proof, that follows the same main steps as that of Theorem 5 in Bartlett and Mendelson [2002]. For any $q\in[0,1]$, let us define the convex combination:

$$\ell(q, h(\boldsymbol{x})) \doteq q\ell(h(\boldsymbol{x})) + (1-q)\ell(-h(\boldsymbol{x})) \ . \tag{4.149}$$

It follows that:

$$\mathbb{E}_{\Sigma_\pi}\mathbb{E}_{\mathcal{S}}[\ell(\sigma(\boldsymbol{x})h(\boldsymbol{x}))] = \mathbb{E}_{\mathcal{S}}[\ell(\pi(\boldsymbol{x}), h(\boldsymbol{x}))] \ , \tag{4.150}$$

with $\pi(\boldsymbol{x})$ the label proportion of the bag to which $\boldsymbol{x}$ belongs in $\mathcal{S}$. We also have $\forall h$:

$$\mathbb{E}_{\mathcal{D}}[\ell(yh(\boldsymbol{x}))] \leq \mathbb{E}_{\mathcal{S}}[\ell(\pi(\boldsymbol{x}), h(\boldsymbol{x}))] + \phi(\mathcal{S}) \ , \tag{4.151}$$

with:

$$\phi(\mathcal{S}) \doteq \sup_g \left\{\mathbb{E}_{\mathcal{D}}[\ell(yg(\boldsymbol{x}))] - \mathbb{E}_{\mathcal{S}}[\ell(\pi(\boldsymbol{x}), g(\boldsymbol{x}))]\right\} \ . \tag{4.152}$$

Let us bound the deviations of $\phi(\mathcal{S})$ around its expectation on the sampling of $\mathcal{S}$, using McDiarmid's inequality. This departs from the well known result of Lemma 14 because we have extended the definition of $\phi(\mathcal{S})$ to account for the missing labels by $\mathbb{E}_{\Sigma_\pi}$. We need to upper bound the maximum difference for the supremum term computed over two samples $\mathcal{S}$ and $\mathcal{S}'$ of the same size, such that $\mathcal{S}'$ is $\mathcal{S}$ with one example replaced. We have:

$$\left|\phi(\mathcal{S}) - \phi(\mathcal{S}')\right| \leq \left|\mathbb{E}_{\mathcal{S}}[\ell(\pi(\boldsymbol{x}), g(\boldsymbol{x}))] - \mathbb{E}_{\mathcal{S}'}[\ell(\pi'(\boldsymbol{x}), g(\boldsymbol{x}))]\right| \ , \tag{4.153}$$

with $\boldsymbol{\pi}$ and $\boldsymbol{\pi}'$ denoting the corresponding label proportions in $\mathcal{S}$ and $\mathcal{S}'$. Let $\{\boldsymbol{x}_1\} = \mathcal{S}\backslash\mathcal{S}'$ and $\{\boldsymbol{x}_2\} = \mathcal{S}'\backslash\mathcal{S}$. Let $\boldsymbol{x}_1 \in \mathcal{S}_j$ and $\boldsymbol{x}_2 \in \mathcal{S}'_{j'}$ for some bags j and j'. The bound 4.153 depends only on bags j and j'. For any $\boldsymbol{x} \in (\mathcal{S}_j \cup \mathcal{S}_{j'})\backslash\{\boldsymbol{x}_1, \boldsymbol{x}_2\}$, with SPLS, due to Lemma 34, we have:

$$\ell(\pi(\boldsymbol{x}), g(\boldsymbol{x})) - \ell(\pi'(\boldsymbol{x}), g(\boldsymbol{x})) \leq \frac{|\ell(g(\boldsymbol{x})) - \ell(-g(\boldsymbol{x}))|}{m(\boldsymbol{x})} \tag{4.154}$$

$$= \frac{|g(\boldsymbol{x})|}{b_\phi m(\boldsymbol{x})} \tag{4.155}$$

$$\leq \frac{H}{b_\phi m(\boldsymbol{x})} \ , \tag{4.156}$$

where $m(\boldsymbol{x})$ is the size of the bag to which it belongs in $\mathcal{S}$, plus 1 iff it is bag j' and $j' \neq j$, minus 1 iff it is bag j and $j' \neq j$. Furthermore:

$$\ell(\pi(\boldsymbol{x}), g(\boldsymbol{x})) = \ell(|g(\boldsymbol{x})|) + \frac{1}{b_\phi} \cdot ((1 - \pi(\boldsymbol{x}) 1\{g(\boldsymbol{x}) > 0\} + \pi(\boldsymbol{x})(1 - 1\{g(\boldsymbol{x}) > 0\})) \cdot |g(\boldsymbol{x})| \tag{4.157}$$

$$\leq \ell(0) + \frac{1}{b_\phi}((1 - \pi(\boldsymbol{x}))1\{g(\boldsymbol{x}) > 0\} + \pi(\boldsymbol{x})(1 - 1\{g(\boldsymbol{x}) > 0\})) \cdot H \tag{4.158}$$

$$\leq \ell(0) + \frac{H}{b_\phi} \ , \forall \boldsymbol{x} \in \mathcal{S} \ . \tag{4.159}$$

Also, it comes from Lemma 34 that:

$$\ell(0) = \frac{1}{b_\phi}(0 \cdot \phi'^{-1}(0) - \phi(\phi'^{-1}(0))) \tag{4.160}$$

$$= \frac{-\phi(1/2)}{b_\phi} = 1 \ . \tag{4.161}$$

We obtain that:

$$|\phi(\mathcal{S}) - \phi(\mathcal{S}')| \leq \frac{1}{m}\left(1 + \frac{H}{b_\phi} + 1 + \frac{H}{b_\phi}\right) + \frac{1}{m}\sum_{\boldsymbol{x}\in(\mathcal{S}_j\cup\mathcal{S}_{j'})\setminus\{\boldsymbol{x}_1,\boldsymbol{x}_2\}} \frac{H}{b_\phi m(\boldsymbol{x})} \tag{4.162}$$

$$\leq \frac{Q_1}{m} \ , \tag{4.163}$$

where:

$$Q_1 \doteq 2\left(\frac{2H}{b_\phi} + 1\right) \ . \tag{4.164}$$

So McDiarmid's inequality yields that with probability $\leq \delta/2$ over the sampling of $\mathcal{S}$:

$$\phi(\mathcal{S}) \geq \mathbb{E}_{\mathcal{D}_m} \sup_g \{\mathbb{E}_{\mathcal{D}}[\ell(yg(\boldsymbol{x}))] - \mathbb{E}_{\mathcal{S}}[\ell(\pi(\boldsymbol{x}), g(\boldsymbol{x}))]\} + Q_1\sqrt{\frac{1}{2m}\log\frac{2}{\delta}} \tag{4.165}$$

We now upper bound the expectation in 4.165. Using the convexity of the supremum, we have:

$$\mathbb{E}_{\mathcal{D}_m} \sup_g \{\mathbb{E}_{\mathcal{D}}[\ell(yg(\boldsymbol{x}))] - \mathbb{E}_{\mathcal{S}}[\ell(\pi(\boldsymbol{x}), g(\boldsymbol{x}))]\} \tag{4.166}$$

$$= \mathbb{E}_{\mathcal{D}_m} \sup_g \left\{\mathbb{E}_{\mathcal{D}'_m}[\ell(yg(\boldsymbol{x}))] - \mathbb{E}_{\mathcal{S}}[\ell(\pi(\boldsymbol{x}), g(\boldsymbol{x}))]\right\} \tag{4.167}$$

$$\leq \mathbb{E}_{\mathcal{D}_m,\mathcal{D}'_m} \sup_g \{\mathbb{E}_{\mathcal{S}'}[\ell(yg(\boldsymbol{x}))] - \mathbb{E}_{\mathcal{S}}[\ell(\pi(\boldsymbol{x}), g(\boldsymbol{x}))]\} \ . \tag{4.168}$$

Consider any set $\mathcal{S} \sim \mathcal{D}_{2m}$, and let $\mathcal{I}'^2 \subseteq [2m]$ be a subset of m indices, picked uniformly at random among all $\binom{2m}{m}$ possible choices. For any $\mathcal{I} \subseteq [2m]$, let $\mathcal{S}(\mathcal{I})$ denote the subset of examples whose index matches $\mathcal{I}$, and for any $\boldsymbol{x} \in \mathcal{S}(\mathcal{I})$, let $\pi(\boldsymbol{x}|\mathcal{S}(\mathcal{I}))$ denote its bag proportion in $\mathcal{S}(\mathcal{I})$. For any $\mathcal{I}'^2_l$ indexed by $l \geq 1$ and any $\boldsymbol{x} \in \mathcal{S}$, let:

$$\pi^s_{|l}(\boldsymbol{x}) \doteq \begin{cases} \pi(\boldsymbol{x}|\mathcal{S}(\mathcal{I}'^2_l)) & \text{if} \quad \boldsymbol{x} \in \mathcal{S}(\mathcal{I}'^2_l) \\ \pi(\boldsymbol{x}|\mathcal{S}\setminus\mathcal{S}(\mathcal{I}'^2_l)) & \text{otherwise} \end{cases} \tag{4.169}$$

denote the label proportions induced by the split of $\mathcal{S}$ in two subsamples $\mathcal{S}(\mathcal{I}'^2_l)$ and $\mathcal{S}\setminus\mathcal{S}(\mathcal{I}'^2_l)$. Let:

$$\pi^\ell_{|l}(\boldsymbol{x}) \doteq \begin{cases} y & \text{if} \quad \boldsymbol{x} \in \mathcal{S}(\mathcal{I}'^2_l) \\ \pi(\boldsymbol{x}|\mathcal{S}\setminus\mathcal{S}(\mathcal{I}'^2_l)) & \text{otherwise} \end{cases} \ , \tag{4.170}$$

where y is the true label of $\boldsymbol{x}$. Let $\sigma_l(\boldsymbol{x}) \doteq 2 \times 1\{\boldsymbol{x} \in \mathcal{S}(\mathcal{I}'^2_l)\} - 1$. The Label Proportion Complexity $L(\mathcal{H})$ quantifies the discrepancy between these two estimators. When each bag in $\mathcal{S}$ has label proportion zero or one, each term factoring classifier h in Equation 4.32 is zero, so $L(\mathcal{H}) = 0$.

Lemma 51. *The following holds true:*

$$\mathbb{E}_{\mathcal{D}_m,\mathcal{D}'_m} \sup_g \{\mathbb{E}_{\mathcal{S}'}[\ell(yg(\boldsymbol{x}))] - \mathbb{E}_{\mathcal{S}}[\ell(\pi(\boldsymbol{x}), g(\boldsymbol{x}))]\} \tag{4.171}$$

$$\leq 2\mathbb{E}_{\mathcal{D}_m,\Sigma_m} \sup_h \{\mathbb{E}_{\mathcal{S}}[\sigma(\boldsymbol{x})\ell(\pi(\boldsymbol{x}), h(\boldsymbol{x}))]\} + L(\mathcal{H}) \ . \tag{4.172}$$

Proof For any $\sigma \in \Sigma_m$ and any sets $\mathcal{S} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_m\}$ and $\mathcal{S}' = \{\boldsymbol{x}'_1, \boldsymbol{x}'_2, ..., \boldsymbol{x}'_m\}$ of size m, denote:

$$\mathcal{S}_\sigma \doteq \{\boldsymbol{x}'_i \text{ iff } \sigma_i = 1, \boldsymbol{x}_i \text{ otherwise}\} \ , \tag{4.173}$$

$$\mathcal{S}_{\overline{\sigma}} \doteq \{\boldsymbol{x}'_i \text{ iff } \sigma_i = -1, \boldsymbol{x}_i \text{ otherwise}\} = (\mathcal{S} \cup \mathcal{S}')\backslash\mathcal{S}_\sigma \ . \tag{4.174}$$

and:

$$\pi_*(\boldsymbol{x}) \doteq \left\{ \begin{array}{ll} \pi_\sigma(\boldsymbol{x}) & \text{if} \quad \boldsymbol{x} \in \mathcal{S}_\sigma \ , \\ \pi_{\overline{\sigma}}(\boldsymbol{x}) & \text{otherwise} \end{array} \right. , \tag{4.175}$$

where: $\pi_\sigma(.)$ denote the label proportions in $\mathcal{S}_\sigma$ and $\pi_{\overline{\sigma}}(.)$ denote the label proportions in $\mathcal{S}_{\overline{\sigma}}$. Let $\pi(.)$ denote the label proportions in $\mathcal{S}$, $\pi'(.)$ denote the label proportions in $\mathcal{S}'$ (we know each bag to which each example in $\mathcal{S}'$ belongs to, so we can compute these estimators), We have:

$$\mathbb{E}_{\mathcal{D}_m,\mathcal{D}'_m} \sup_h \{\mathbb{E}_{\mathcal{S}'}[\ell(yh(\boldsymbol{x}))] - \mathbb{E}_{\mathcal{S}}[\ell(\pi(\boldsymbol{x}), h(\boldsymbol{x}))]\} \tag{4.176}$$

$$= \mathbb{E}_{\mathcal{D}_m,\mathcal{D}'_m} \sup_h \left\{ \mathbb{E}_{\mathcal{S}'}[\ell(\pi'(\boldsymbol{x}), h(\boldsymbol{x}))] - \mathbb{E}_{\mathcal{S}}[\ell(\pi(\boldsymbol{x}), h(\boldsymbol{x}))] - \frac{\Delta_1}{b_\phi} \right\} \tag{4.177}$$

$$= \mathbb{E}_{\mathcal{D}_m,\mathcal{D}'_m} \sup_h \left\{ \mathbb{E}_{\mathcal{S}_\sigma}[\sigma(\boldsymbol{x})\ell(\pi^l(\boldsymbol{x}), h(\boldsymbol{x}))] - \mathbb{E}_{\mathcal{S}_{\overline{\sigma}}}[\sigma(\boldsymbol{x})\ell(\pi^r(\boldsymbol{x}), h(\boldsymbol{x}))] - \frac{\Delta_1}{b_\phi} \right\} \ , \tag{4.178}$$

with:

$$\Delta_1 \doteq \mathbb{E}_{\mathcal{S}'}[((1 - \pi'(\boldsymbol{x}))1\{y = 1\} - \pi'(\boldsymbol{x})1\{y = -1\})h(\boldsymbol{x})] \ , \tag{4.179}$$

$$\pi^l(\boldsymbol{x}) \doteq \frac{1}{2}\left((1 + \sigma(\boldsymbol{x}))\pi'(\boldsymbol{x}) + (1 - \sigma(\boldsymbol{x}))\pi(\boldsymbol{x})\right) \ , \tag{4.180}$$

$$\pi^r(\boldsymbol{x}) \doteq \frac{1}{2}\left((1 + \sigma(\boldsymbol{x}))\pi(\boldsymbol{x}) + (1 - \sigma(\boldsymbol{x}))\pi'(\boldsymbol{x})\right) \ . \tag{4.181}$$

We also have from Lemma 34:

$$\mathbb{E}_{\mathcal{S}_\sigma}[\sigma(\boldsymbol{x})\ell(\pi^l(\boldsymbol{x}), h(\boldsymbol{x}))] = \mathbb{E}_{\mathcal{S}_\sigma}[\sigma(\boldsymbol{x})\ell(\pi_\sigma(\boldsymbol{x}), h(\boldsymbol{x}))] - \frac{\Delta_2}{b_\phi} \ , \tag{4.182}$$

$$\mathbb{E}_{\mathcal{S}_{\overline{\sigma}}}[\sigma(\boldsymbol{x})\ell(\pi^r(\boldsymbol{x}), h(\boldsymbol{x}))] = \mathbb{E}_{\mathcal{S}_{\overline{\sigma}}}[\sigma(\boldsymbol{x})\ell(\pi_{\overline{\sigma}}(\boldsymbol{x}), h(\boldsymbol{x}))] - \frac{\Delta_3}{b_\phi} \ , \tag{4.183}$$

with:

$$\Delta_2 \doteq \mathbb{E}_{\mathcal{S}_\sigma}[\sigma(\boldsymbol{x})(\pi^l(\boldsymbol{x}) - \pi_\sigma(\boldsymbol{x}))h(\boldsymbol{x})] \ , \tag{4.184}$$

$$\Delta_3 \doteq \mathbb{E}_{\mathcal{S}_{\overline{\sigma}}}[\sigma(\boldsymbol{x})(\pi^r(\boldsymbol{x}) - \pi_{\overline{\sigma}}(\boldsymbol{x}))h(\boldsymbol{x})] \ . \tag{4.185}$$

We also have:

$$\Delta_3 - \Delta_2 - \Delta_1 = \mathbb{E}_{\mathcal{S}'}[(\pi_*(\boldsymbol{x}) - 1\{y = 1\})h(\boldsymbol{x})] + \mathbb{E}_{\mathcal{S}}[(\pi(\boldsymbol{x}) - \pi_*(\boldsymbol{x}))h(\boldsymbol{x})] \tag{4.186}$$

$$\doteq \Delta_4 \ . \tag{4.187}$$

Putting Equations 4.178, 4.182, 4.183 and 4.187 altogether, we get, after introducing Rademacher variables:

$$\mathbb{E}_{\mathcal{D}_m, \mathcal{D}'_m, \Sigma_m} \sup_h \{\mathbb{E}_{\mathcal{S}'}[\ell(yh(\boldsymbol{x}))] - \mathbb{E}_{\mathcal{S}}[\ell(\pi(\boldsymbol{x}), h(\boldsymbol{x}))]\} \tag{4.188}$$

$$= \mathbb{E}_{\mathcal{D}_m, \mathcal{D}'_m, \Sigma_m} \sup_h \{\mathbb{E}_{\mathcal{S}_\sigma}[\sigma(\boldsymbol{x})\ell(\pi_\sigma(\boldsymbol{x}), h(\boldsymbol{x}))] - \mathbb{E}_{\mathcal{S}_{\overline{\sigma}}}[\sigma(\boldsymbol{x})\ell(\pi_{\overline{\sigma}}(\boldsymbol{x}), h(\boldsymbol{x}))] + \Delta_4\} \tag{4.189}$$

$$\leq \mathbb{E}_{\mathcal{D}_m, \mathcal{D}'_m, \Sigma_m} \sup_h \{\mathbb{E}_{\mathcal{S}_\sigma}[\sigma(\boldsymbol{x})\ell(\pi_\sigma(\boldsymbol{x}), h(\boldsymbol{x}))] - \mathbb{E}_{\mathcal{S}_{\overline{\sigma}}}[\sigma(\boldsymbol{x})\ell(\pi_{\overline{\sigma}}(\boldsymbol{x}), h(\boldsymbol{x}))]\}$$
$$+ \mathbb{E}_{\mathcal{D}_m, \mathcal{D}'_m, \Sigma_m} \sup_h \{\mathbb{E}_{\mathcal{S}'}[(\pi_*(\boldsymbol{x}) - 1\{y = 1\})h(\boldsymbol{x})] + \mathbb{E}_{\mathcal{S}}[(\pi(\boldsymbol{x}) - \pi_*(\boldsymbol{x}))h(\boldsymbol{x})]\} \tag{4.190}$$

$$= \mathbb{E}_{\mathcal{D}_m, \mathcal{D}'_m, \Sigma_m} \sup_h \left\{\mathbb{E}_{\mathcal{S}'}[\sigma(\boldsymbol{x})\ell(\pi'(\boldsymbol{x}), h(\boldsymbol{x}))] - \mathbb{E}_{\mathcal{S}}[\sigma(\boldsymbol{x})\ell(\pi(\boldsymbol{x}), h(\boldsymbol{x}))]\right\}$$
$$+ \mathbb{E}_{\mathcal{D}_m, \mathcal{D}'_m, \Sigma_m} \sup_h \{\mathbb{E}_{\mathcal{S}'}[(\pi_*(\boldsymbol{x}) - 1\{y = 1\})h(\boldsymbol{x})] + \mathbb{E}_{\mathcal{S}}[(\pi(\boldsymbol{x}) - \pi_*(\boldsymbol{x}))h(\boldsymbol{x})]\} \tag{4.191}$$

$$\leq 2\mathbb{E}_{\mathcal{D}_m, \Sigma_m} \sup_h \{\mathbb{E}_{\mathcal{S}}[\sigma(\boldsymbol{x})\ell(\pi(\boldsymbol{x}), h(\boldsymbol{x}))]\}$$
$$+ \mathbb{E}_{\mathcal{D}_m, \mathcal{D}'_m, \Sigma_m} \sup_h \{\mathbb{E}_{\mathcal{S}'}[(\pi_*(\boldsymbol{x}) - 1\{y = 1\})h(\boldsymbol{x})] + \mathbb{E}_{\mathcal{S}}[(\pi(\boldsymbol{x}) - \pi_*(\boldsymbol{x}))h(\boldsymbol{x})]\} \ . \tag{4.192}$$

Equation 4.191 holds because the distribution of the supremum is the same. We also have:

$$\mathbb{E}_{\mathcal{D}_m, \mathcal{D}'_m, \Sigma_m} \sup_h \{\mathbb{E}_{\mathcal{S}'}[(\pi_*(\boldsymbol{x}) - 1\{y = 1\})h(\boldsymbol{x})] + \mathbb{E}_{\mathcal{S}}[(\pi(\boldsymbol{x}) - \pi_*(\boldsymbol{x}))h(\boldsymbol{x})]\} \tag{4.193}$$

$$= \mathbb{E}_{\mathcal{D}_m, \mathcal{D}'_m, \Sigma_m} \sup_h \{\mathbb{E}_{\mathcal{S}}[(\pi(\boldsymbol{x}) - \pi_*(\boldsymbol{x}))h(\boldsymbol{x})] - \mathbb{E}_{\mathcal{S}'}[(1\{y = 1\} - \pi_*(\boldsymbol{x}))h(\boldsymbol{x})]\} \tag{4.194}$$

$$= \mathbb{E}_{\mathcal{D}_{2m}} \mathbb{E}_{\mathcal{I}_1^{/2}, \mathcal{I}_2^{/2}} \sup_h \mathbb{E}_{\mathcal{S}}[\sigma_1(\boldsymbol{x})(\pi^s_{|2}(\boldsymbol{x}) - \pi^\ell_{|1}(\boldsymbol{x}))h(\boldsymbol{x})] \tag{4.195}$$

$$= L(\mathcal{H}) \ . \tag{4.196}$$

Equation 4.195 holds because swapping the sample does not make any difference in the outer expectation, as each couple of swapped samples is generated with the same probability without swapping. Putting altogether 4.192 and 4.196 ends the proof of

Lemma 51. ■

We now bound the deviations of $\mathbb{E}_{\Sigma_m} \sup_h \{\mathbb{E}_{\mathcal{S}}[\sigma(\boldsymbol{x})\ell(\pi(\boldsymbol{x}), h(\boldsymbol{x}))]\}$ with respect to its expectation over the sampling of $\mathcal{S}$, $\mathbb{E}_{\mathcal{D}_m, \Sigma_m} \sup_h \{\mathbb{E}_{\mathcal{S}}[\sigma(\boldsymbol{x})\ell(\pi(\boldsymbol{x}), h(\boldsymbol{x}))]\}$. To do that, we use a third time McDiarmid's inequality and compute an upper bound for:

$$\left| \begin{array}{l} \mathbb{E}_{\Sigma_m} \sup_h \{\mathbb{E}_{\mathcal{S}_1}[\sigma(\boldsymbol{x})\ell(\pi(\boldsymbol{x}), h(\boldsymbol{x}))]\} \\ -\mathbb{E}_{\Sigma_m} \sup_h \{\mathbb{E}_{\mathcal{S}_2}[\sigma(\boldsymbol{x})\ell(\pi(\boldsymbol{x}), h(\boldsymbol{x}))]\} \end{array} \right| \tag{4.197}$$

$$\leq \mathbb{E}_{\Sigma_m} \left[\left| \begin{array}{l} \sup_h \{\mathbb{E}_{\mathcal{S}_1}[\sigma(\boldsymbol{x})\ell(\pi(\boldsymbol{x}), h(\boldsymbol{x}))]\} \\ -\sup_h \{\mathbb{E}_{\mathcal{S}_2}[\sigma(\boldsymbol{x})\ell(\pi(\boldsymbol{x}), h(\boldsymbol{x}))]\} \end{array} \right| \right] \tag{4.198}$$

$$\leq \max_{\Sigma_m} \left[\left| \begin{array}{l} \sup_h \{\mathbb{E}_{\mathcal{S}_1}[\sigma(\boldsymbol{x})\ell(\pi(\boldsymbol{x}), h(\boldsymbol{x}))]\} \\ -\sup_h \{\mathbb{E}_{\mathcal{S}_2}[\sigma(\boldsymbol{x})\ell(\pi(\boldsymbol{x}), h(\boldsymbol{x}))]\} \end{array} \right| \right] \leq \frac{Q_1}{m} \ , \tag{4.199}$$

where Q_1 is defined in Equation 4.164. Equation 4.198 holds because of the triangular inequality. Inequality 4.199 holds because $|\sigma(\cdot)| = 1$. So with probability $\leq \delta/2$ over the sampling of $\mathcal{S}$:

$$\begin{aligned} &\mathbb{E}_{\Sigma_m} \sup_h \{\mathbb{E}_{\mathcal{S}}[\sigma(\boldsymbol{x})\ell(\pi(\boldsymbol{x}), h(\boldsymbol{x}))]\} \\ &\leq \mathbb{E}_{\mathcal{D}_m, \Sigma_m} \sup_h \{\mathbb{E}_{\mathcal{S}}[\sigma(\boldsymbol{x})\ell(\pi(\boldsymbol{x}), h(\boldsymbol{x}))]\} - Q_1 \sqrt{\frac{1}{2m} \log \frac{2}{\delta}} \ , \end{aligned} \tag{4.200}$$

where Q_1 is defined via Equation 4.163. We obtain that with probability $> 1 - (\delta/2 +$

$\delta/2) = 1 - \delta$, the following holds $\forall h$:

$$
\begin{aligned}
&\mathbb{E}_{\mathcal{D}}[\ell(yh(\boldsymbol{x}))] \\
&\leq \mathbb{E}_{\mathcal{S}}[\ell(\pi(\boldsymbol{x}), h(\boldsymbol{x}))] + \phi(\mathcal{S}) \text{ (see 4.151 and 4.152)} && (4.201)\\
&\leq \mathbb{E}_{\mathcal{S}}[\ell(\pi(\boldsymbol{x}), h(\boldsymbol{x}))] + \mathbb{E}_{\mathcal{D}_m} \sup_g \{\mathbb{E}_{\mathcal{D}}[\ell(yg(\boldsymbol{x}))] - \mathbb{E}_{\mathcal{S}}[\ell(\pi(\boldsymbol{x}), g(\boldsymbol{x}))]\} \\
&\qquad + Q_1 \sqrt{\frac{1}{2m} \log \frac{2}{\delta}} \text{ (from 4.165)} && (4.202)\\
&\leq \mathbb{E}_{\mathcal{S}}[\ell(\pi(\boldsymbol{x}), h(\boldsymbol{x}))] + \mathbb{E}_{\mathcal{D}_m, \mathcal{D}'_m} \sup_g \{\mathbb{E}_{\mathcal{S}'}[\ell(yg(\boldsymbol{x}))] - \mathbb{E}_{\mathcal{S}}[\ell(\pi(\boldsymbol{x}), g(\boldsymbol{x}))]\} \\
&\qquad + Q_1 \sqrt{\frac{1}{2m} \log \frac{2}{\delta}} \text{ (from 4.168)} && (4.203)\\
&\leq \mathbb{E}_{\mathcal{S}}[\ell(\pi(\boldsymbol{x}), h(\boldsymbol{x}))] + 2\mathbb{E}_{\mathcal{D}_m, \Sigma_m} \sup_g \{\mathbb{E}_{\mathcal{S}}[\sigma(\boldsymbol{x})\ell(\pi(\boldsymbol{x}), g(\boldsymbol{x}))]\} + L(\mathcal{H}) \\
&\qquad + Q_1 \sqrt{\frac{1}{2m} \log \frac{2}{\delta}} \text{ (Lemma 51)} && (4.204)\\
&\leq \mathbb{E}_{\mathcal{S}}[\ell(\pi(\boldsymbol{x}), h(\boldsymbol{x}))] + 2\mathbb{E}_{\Sigma_m} \sup_h \{\mathbb{E}_{\mathcal{S}}[\sigma(\boldsymbol{x})\ell(\pi(\boldsymbol{x}), h(\boldsymbol{x}))]\} + L(\mathcal{H}) \\
&\qquad + 2Q_1 \sqrt{\frac{1}{2m} \log \frac{2}{\delta}} \text{ (from 4.200)} && (4.205)\\
&= \mathbb{E}_{\Sigma_\pi}\mathbb{E}_{\mathcal{S}}[\ell(\sigma(\boldsymbol{x})h(\boldsymbol{x}))] + 2R^b(\ell \circ \mathcal{H} \circ \mathcal{S}) + L(\mathcal{H}) \\
&\qquad + 4\left(\frac{2H}{b_\phi} + 1\right) \sqrt{\frac{1}{2m} \log \frac{2}{\delta}}\ , && (4.206)
\end{aligned}
$$

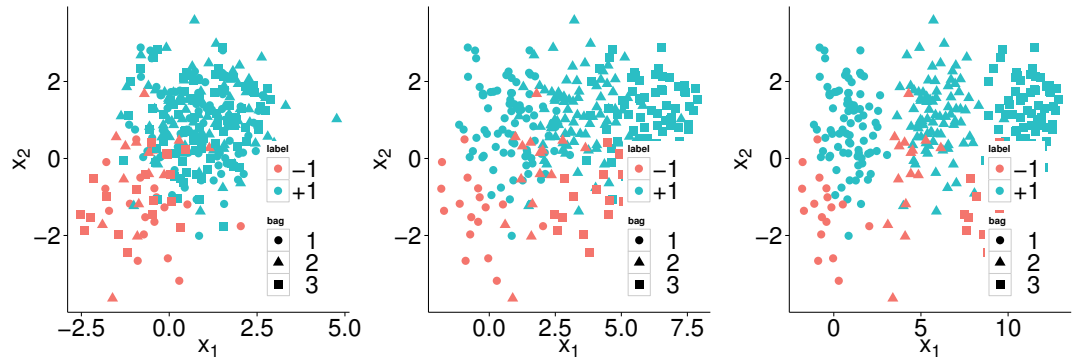
as claimed.

4.11.9.2 Proof of Equation 4.34

We have $\ell'(x) = -(1/b_\phi)(\phi^\star)'(-x) = -(1/b_\phi)(\phi')^{-1}(-x) \in [-1/b_\phi, 0]$. ℓ is therefore $1/b_\phi$-Lipschitz. So Theorem 4.12 in Ledoux and Talagrand [1991] brings:

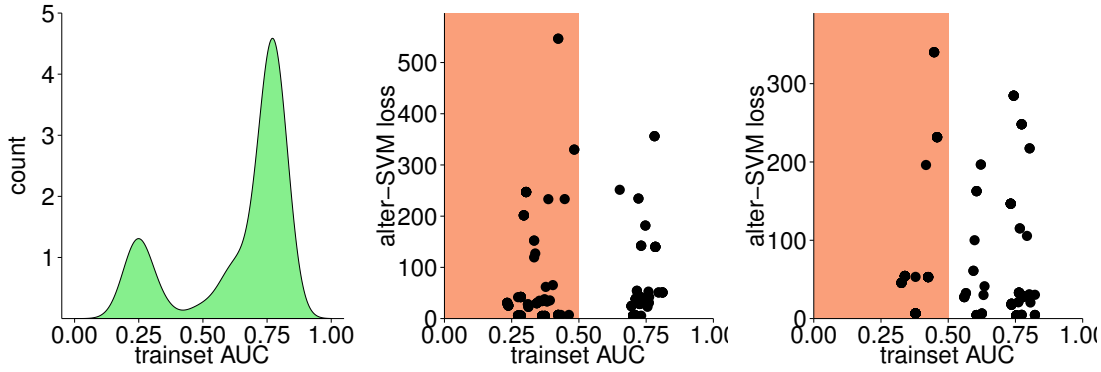
$$
\begin{aligned}
\mathcal{R}^b(F, \eta) &= \mathbb{E}_{\sigma \sim \Sigma_m} \sup_{h \in \mathcal{H}} \left\{ \mathbb{E}_{i \sim [m]}[\sigma_i \mathbb{E}_{\sigma' \sim \Sigma_\pi}[\ell(\sigma'_i h(\boldsymbol{x}_i) - \eta)]] \right\} && (4.207)\\
&\leq b_\phi \mathbb{E}_{\sigma \sim \Sigma_m} \sup_{h \in \mathcal{H}} \left\{ \mathbb{E}_{i \sim [m]}[\sigma_i \mathbb{E}_{\sigma' \sim \Sigma_\pi}[\sigma'_i h(\boldsymbol{x}_i) - \eta]] \right\} && (4.208)\\
&= b_\phi \mathbb{E}_{\sigma \sim \Sigma_m} \sup_{h \in \mathcal{H}} \left\{ \mathbb{E}_{i \sim [m]}[\sigma_i \mathbb{E}_{\sigma' \sim \Sigma_\pi}[\sigma'_i h(\boldsymbol{x}_i)]] \right\} && (4.209)\\
&= b_\phi \mathbb{E}_{\sigma \sim \Sigma_m} \sup_{h \in \mathcal{H}} \left\{ \mathbb{E}_{i \sim [m]}[\sigma_i (2\pi(\boldsymbol{x}_i) - 1) h(\boldsymbol{x}_i)] \right\}\ , && (4.210)
\end{aligned}
$$

as claimed.



(a) (b) (c)

Figure 4.2: Violation of homogeneity assumption



(a) (b) (c)

Figure 4.3: alter-∝SVM: empirical distribution of AUC (a), and relationship between loss and AUC in two different train spit (b)(c)

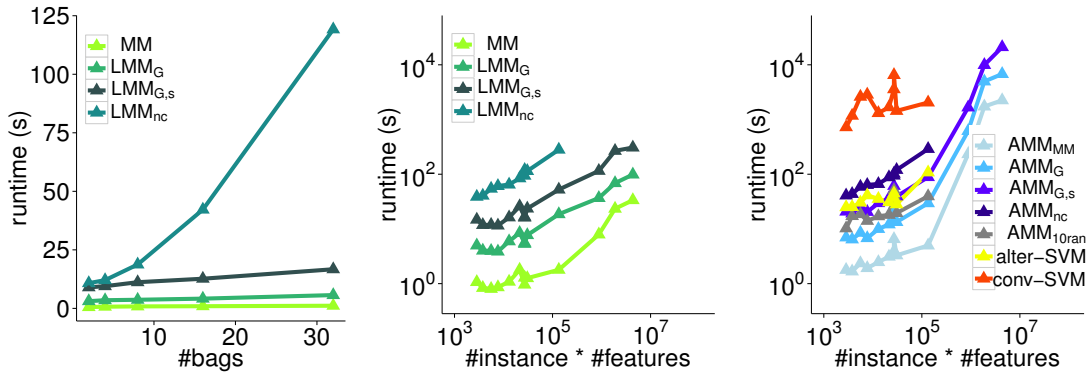
4.12 Appendix: additional experimental results

4.12.1 Simulated domain for violation of homogeneity assumption

The synthetic data generated for this test consists on 16 classification problems, each one formed by 16 bags of 100 two-dimensional normal samples. The distribution generating the first dataset satisfies the homogeneity assumption (Figure 4.2 (a)). Then, we gradually change the position of the class-conditional bag-conditional means on one linear direction (to the right on Figure 4.2 (b) and (c)), with different offsets for different bags. In Figure 4.2 we give a graphical explanation of the process on 3 bags.

4.12.2 Additional tests on alter-∝SVM Yu et al. [2013]

In our experiments, we observe that the AUC achieved by ∝SVM can be very high, but it is also often *below* 0.5; in those cases the algorithm outputs models which are



(a) (b) (c)

Figure 4.4: Learning runtime of LMM for bags number (a), and for domain size one-shot (b) and iterative methods (c)

worse than random and the average performance over the 5 test splits drops. We are able to reproduce the same behavior on the heart dataset provided by the authors in a demo for alter-∝SVM; this also proves our bag assignment for LLP simulation does not introduce the issue. In a first test, we randomly select 3/4 of the dataset, and randomly assign instances to 4 bags of fixed size 64, following Yu et al. [2013]. We repeat the training split 50 times with $C = C_p = 1$, as in the demo, and we measure AUC on the same training set. As expected, a consistent number of run (22%) ends up producing AUC smaller than 0.5. We display in Figure 4.3 (a) the AUC density profile, which shows a relevant mass around 0.25; notice also the two distribution modes look symmetric around 0.5.

In a second test, we investigate further measuring pairs of training set AUC and losses obtained by the same execution of the algorithm. In this case, we run over all parameters ranges defined in ∝SVM paper, and do not pick the model that minimizes the loss over the 10 random runs, but record losses of all. Figures 4.3 (b) and (c) show scatter plots relative to two chosen training set splits. We observe that loss minimization can lead both to high and low AUC, with few points close to 0.5. A possible explanation might be found in the inverted polarity of the learned linear classifier; inverted polarity in this contest means having a model which would achieve better performance classifying instances labels opposite to the ones predicted. We conclude that optimizing ∝SVM loss in some cases might be equivalent to train a max-margin separator of the unlabeled data, exploiting only weakly the information given by the label proportions. This would give a heuristic understanding of the frequent symmetrical behavior of the AUC.

4.12.3 Scalability

Figure 4.1 (a) shows runtime of learning (including cross-validation) of MM and LMM with regard to the number of bags – which is the natural parameter of time

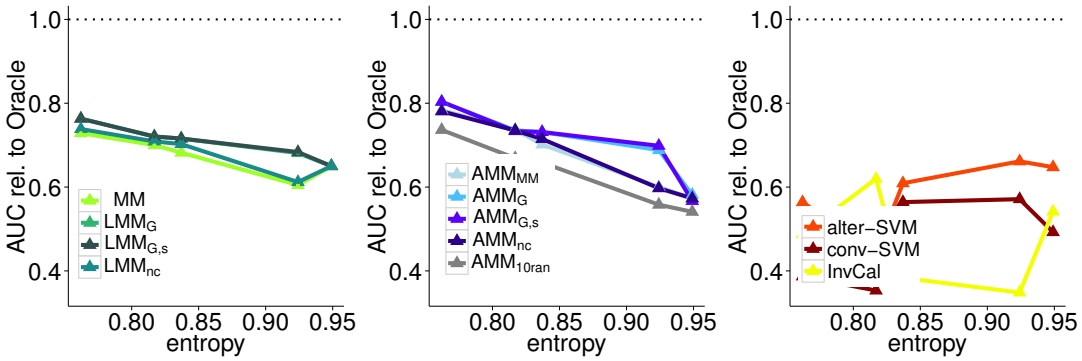
dataset	instances	feature
arrhythmia	452	297
australian	690	39
breastw	699	11
colic	368	83
german	1000	27
heart	270	14
ionosphere	351	37
vertebral column	620	9
vote	435	49
wine	178	16

Table 4.5: Small domains size

complexity for our Laplacian-based methods. Although the 3 layers of cross-validation of $\text{LMM}_{\text{G,s}}$, LMM_{nc} results the only method clearly not scalable. Figure 4.1 (b) presents how our one-shots algorithms scale on all small domains as a function of problem size. Runtime is averaged over the different bag assignments. The same plot is given in Figure 4.1 (c) for iterative algorithms, in particular AMM^{min} and (alter/conv)-$\propto$SVM. All curves are completed with measurements on bigger domains when available. Runtime of SVMs is not directly comparable with our methods. This is due to both (a) the implementation on different programming languages and (b) to the fact that the code provided implements kernel SVM, even for linear kernels, which is a big overhead in computation and memory access. Nevertheless, the high growth rate of conv-$\propto$SVM makes the algorithm not suitable for large datasets. Noticeably, even if alter-$\propto$SVM does not show such behavior, we are not able to run it on the bigger domains, since it requires several hours to run on a training set split with fixed parameters.

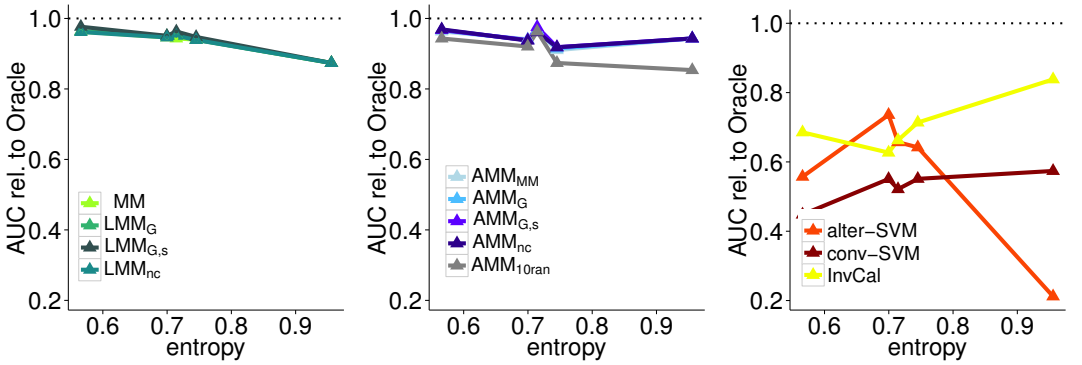
4.12.4 Full results on small domains

Finally we report details about all experiments run on the 10 small domains (Table 4.5). In the following Tables, columns show the number of bags generated through k-MEANS. Each cell contains average AUC over 5 test splits and standard deviation; runtime in second is in the separated column. Best performing algorithm and ones not worse than 0.1 AUC are bold faced. Comparisons are made in the respective top/bottom sub-tables, which group one-shot and iterative algorithms. We use $\uparrow$ to highlight runs which achieve AUC greater or equal than the Oracles.



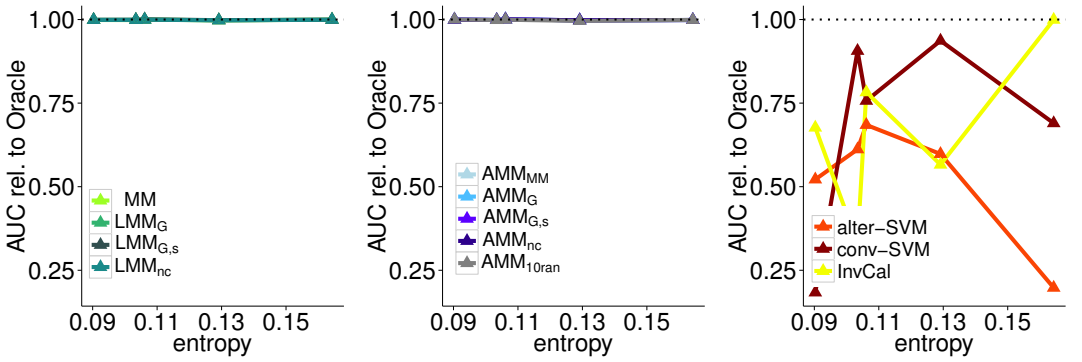
(a) (b) (c)

Figure 4.5: Relative AUC (w.r.t. Oracle) vs entropy on arrhythmia



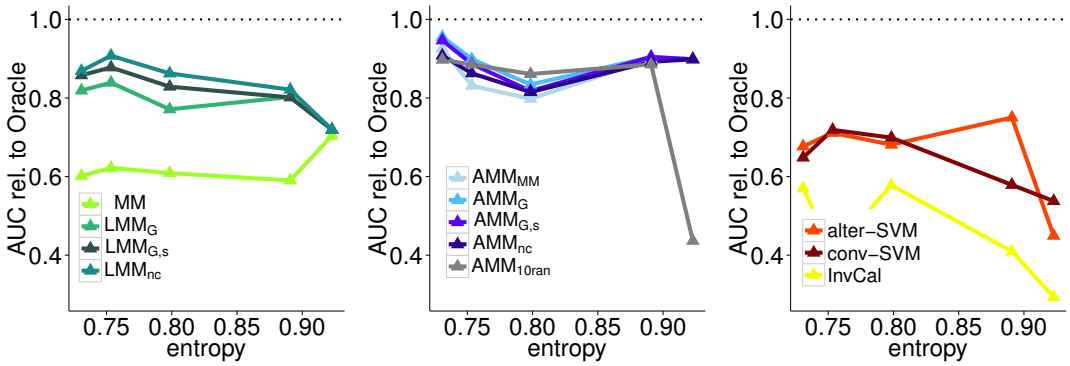
(a) (b) (c)

Figure 4.6: Relative AUC (w.r.t. Oracle) vs entropy on australian



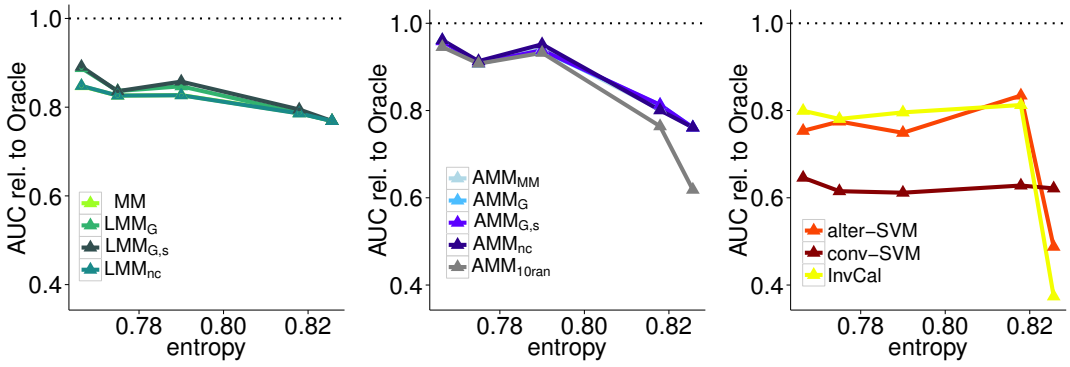
(a) (b) (c)

Figure 4.7: Relative AUC (w.r.t. Oracle) vs entropy on breastw



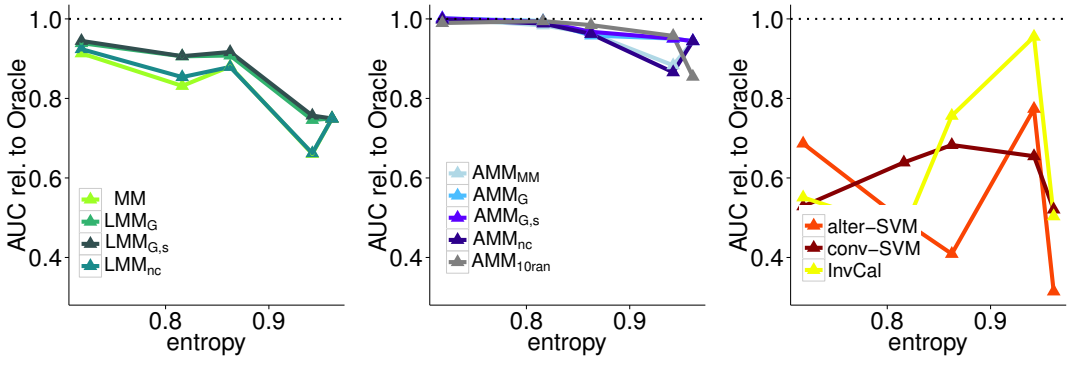
(a) (b) (c)

Figure 4.8: Relative AUC (w.r.t. Oracle) vs entropy on colic



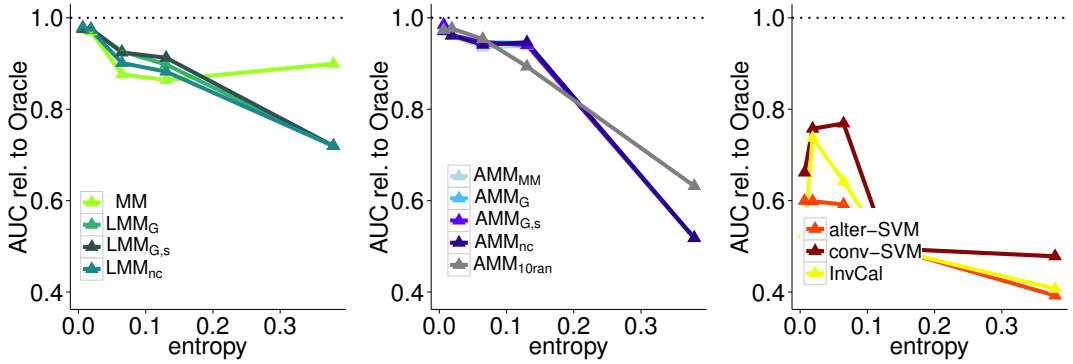
(a) (b) (c)

Figure 4.9: Relative AUC (w.r.t. Oracle) vs entropy on german



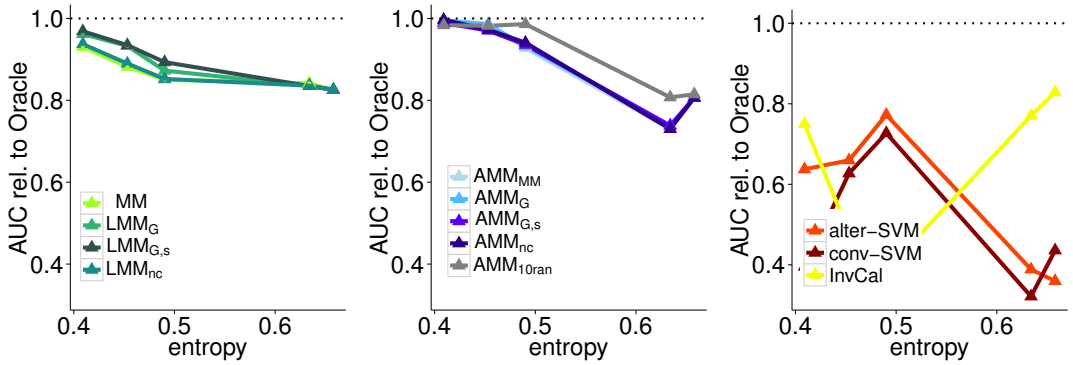
(a) (b) (c)

Figure 4.10: Relative AUC (w.r.t. Oracle) vs entropy on heart



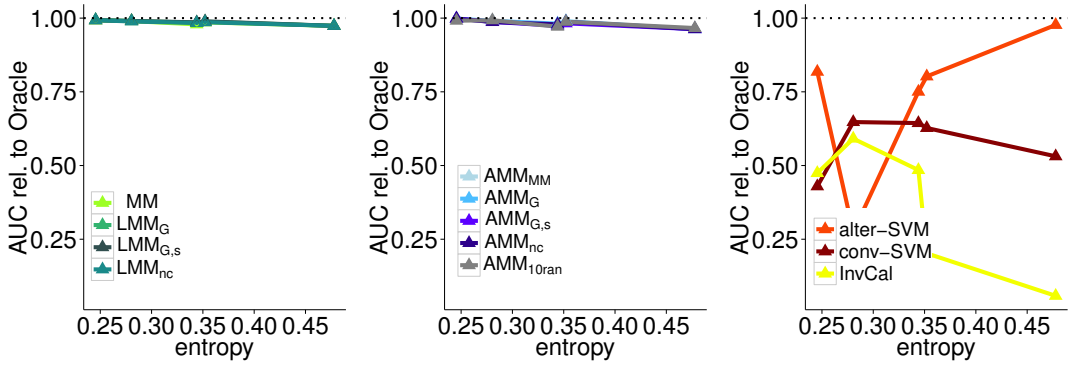
(a) (b) (c)

Figure 4.11: Relative AUC (w.r.t. Oracle) vs entropy on ionosphere



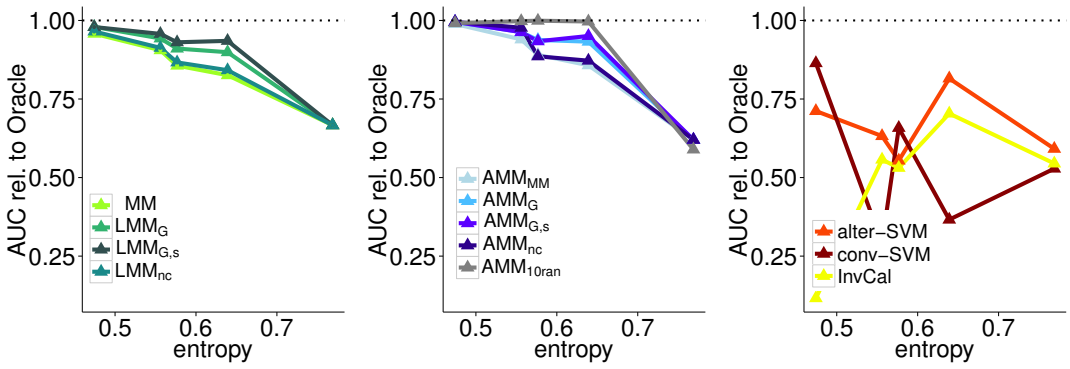
(a) (b) (c)

Figure 4.12: Relative AUC (w.r.t. Oracle) vs entropy on vertebral column



(a) (b) (c)

Figure 4.13: Relative AUC (w.r.t. Oracle) vs entropy on vote



(a) (b) (c)

Figure 4.14: Relative AUC (w.r.t. Oracle) vs entropy on wine

algorithm		**2 bags**		**4 bags**		**8 bags**		**16 bags**		**32 bags**	
	AUC	time(s)	AUC	time(s)	AUC	time(s)	AUC	time(s)	AUC	time(s)	
	EMM	**70.91 ± 6.81**	2	50.55 ± 7.54	2	50.31 ± 7.55	2	47.03 ± 6.60	2	52.34 ± 7.25	2
	MM	64.99 ± 2.99	2	60.48 ± 7.28	1	68.17 ± 5.95	2	70.01 ± 9.33	2	72.85 ± 9.49	2
	LMM_G	64.99 ± 2.99	18	68.10 ± 4.43	17	**71.53 ± 2.36**	20	**72.06 ± 7.62**	18	**76.29 ± 7.91**	20
	$LMM_{G,s}$	64.99 ± 2.99	49	**68.34 ± 3.95**	49	**71.53 ± 2.36**	54	**72.06 ± 7.62**	52	**76.29 ± 7.91**	57
	LMM_{nc}	64.99 ± 2.99	83	61.19 ± 7.53	83	70.21 ± 5.17	119	70.89 ± 9.86	267	73.82 ± 9.29	854
	InvCal	64.75 ± 3.04	17	66.12 ± 260	17	60.87 ± 3.54	17	44.46 ± 3.36	17	56.36 ± 5.26	17
AMM^{min}	AMM_{EMM}	59.54 ± 7.52	9	52.65 ± 3.10	8	63.46 ± 10.37	8	67.85 ± 9.56	8	75.65 ± 8.81	8
AMM^{min}	AMM_{MM}	57.29 ± 5.95	7	60.00 ± 7.96	4	70.12 ± 6.46	4	73.66 ± 8.86	5	78.36 ± 8.53	5
AMM^{min}	AMM_G	58.15 ± 6.83	31	68.80 ± 2.15	28	**73.08 ± 2.92**	30	74.54 ± 7.98	29	**80.32 ± 8.08**	30
AMM^{min}	$AMM_{G,s}$	56.67 ± 4.66	92	69.83 ± 2.69	84	**73.08 ± 2.92**	88	73.34 ± 7.62	88	**80.32 ± 8.08**	91
AMM^{min}	AMM_{nc}	57.29 ± 5.95	97	59.71 ± 8.39	90	71.43 ± 6.21	126	73.49 ± 8.95	274	78.04 ± 8.26	862
AMM^{min}	AMM_1	**65.80 ± 6.92**	5	**70.00 ± 5.89**	4	68.17 ± 7.19	4	69.93 ± 4.27	4	72.31 ± 5.02	5
AMM^{min}	AMM_{10ran}	54.09 ± 12.03	30	55.78 ± 17.36	32	66.38 ± 7.32	51	66.89 ± 6.75	51	73.61 ± 5.15	57
AMM^{max}	AMM_{EMM}	50.59 ± 5.97	41	59.32 ± 5.82	41	60.85 ± 5.43	37	60.38 ± 4.08	41	58.31 ± 8.40	40
AMM^{max}	AMM_{MM}	62.08 ± 9.46	45	46.86 ± 3.90	34	67.28 ± 8.92	33	74.04 ± 9.46	35	71.00 ± 7.65	38
AMM^{max}	AMM_G	62.08 ± 9.46	141	62.27 ± 8.14	128	65.78 ± 3.92	118	64.64 ± 10.26	121	73.07 ± 6.72	124
AMM^{max}	$AMM_{G,s}$	62.08 ± 9.46	414	63.13 ± 5.17	380	63.85 ± 7.00	346	65.49 ± 10.62	354	73.05 ± 6.70	374
AMM^{max}	AMM_{nc}	62.08 ± 9.46	206	55.57 ± 6.07	182	64.30 ± 6.24	207	**76.33 ± 3.96**	362	70.82 ± 4.23	965
AMM^{max}	AMM_1	60.53 ± 9.79	31	54.14 ± 13.28	34	67.45 ± 3.91	32	55.85 ± 8.96	35	61.26 ± 6.95	38
AMM^{max}	AMM_{10ran}	49.79 ± 8.14	307	55.37 ± 14.62	370	53.78 ± 5.13	301	60.62 ± 8.04	322	64.20 ± 2.84	338
SVM	alter-$\propto$	49.24 ± 3.92	96	57.10 ± 2.71	100	56.38 ± 2.73	104	35.31 ± 1.30	114	38.68 ± 6.10	125
SVM	conv-$\propto$	54.15 ± 2.22	2054	34.82 ± 3.20	2078	38.31 ± 8.24	2168	61.96 ± 1.10	1930	48.77 ± 5.73	2004
	Oracle	99.99 ± 0.02	2	99.98 ± 0.05	2	99.94 ± 0.13	2	100.00 ± 0.00	2	99.97 ± 0.07	2

Table 4.6: arrhythmia

algorithm		**2 bags**		**4 bags**		**8 bags**		**16 bags**		**32 bags**	
	AUC	time(s)	AUC	time(s)	AUC	time(s)	AUC	time(s)	AUC	time(s)	
	EMM	66.48 ± 3.16	<1	64.67 ± 4.22	<1	63.56 ± 4.00	<1	64.17 ± 4.80	<1	63.14 ± 5.41	<1
	MM	**81.08 ± 1.66**	<1	87.11 ± 2.68	<1	87.49 ± 2.86	1	87.36 ± 2.22	<1	89.53 ± 2.13	2
	LMM_G	**81.08 ± 1.66**	4	87.09 ± 2.82	4	**87.81 ± 3.16**	5	88.46 ± 2.50	6	89.69 ± 2.68	8
	$LMM_{G,s}$	**81.08 ± 1.66**	14	**87.81 ± 3.08**	15	**87.88 ± 3.21**	19	**89.18 ± 2.05**	20	**90.80 ± 2.53**	27
	LMM_{nc}	**81.08 ± 1.66**	57	87.02 ± 2.72	49	87.46 ± 3.03	57	88.06 ± 2.31	90	89.41 ± 2.41	217
	Invcal	19.67 ± 2.23	5	59.50 ± 5.86	5	68.00 ± 5.27	5	60.83 ± 3.17	5	51.81 ± 4.72	5
AMM^{min}	AMM_{EMM}	86.65 ± 2.06	4	**86.59 ± 3.08**	4	86.50 ± 4.11	4	89.51 ± 2.48	6	88.85 ± 4	6
AMM^{min}	AMM_{MM}	**87.54 ± 3.84**	3	84.35 ± 3.63	4	**86.99 ± 3.87**	4	89.43 ± 1.34	4	89.55 ± 3.18	5
AMM^{min}	AMM_G	**87.54 ± 3.84**	10	84.79 ± 3.17	13	86.78 ± 4.21	14	89.52 ± 2.18	14	89.88 ± 2.78	18
AMM^{min}	$AMM_{G,s}$	**87.54 ± 3.84**	30	85.12 ± 3.75	39	86.75 ± 4.19	43	**90.37 ± 1.67**	43	89.95 ± 2.80	54
AMM^{min}	AMM_{nc}	**87.54 ± 3.84**	63	85.10 ± 3.55	57	86.63 ± 4.02	66	89.00 ± 1.83	97	**90.11 ± 2.93**	227
AMM^{min}	AMM_1	72.60 ± 5.70	2	85.04 ± 2.53	3	**86.89 ± 3.73**	4	88.91 ± 2.32	4	88.98 ± 3.00	4
AMM^{min}	AMM_{10ran}	79.21 ± 5.07	27	80.97 ± 2.27	31	85.08 ± 3.30	34	89.19 ± 1.81	46	87.70 ± 2.68	47
AMM^{max}	AMM_{EMM}	80.09 ± 3.99	17	71.46 ± 1.85	16	73.41 ± 6.07	16	73.25 ± 3.33	18	81.73 ± 3.60	19
AMM^{max}	AMM_{MM}	86.83 ± 4.26	20	72.96 ± 2.30	15	70.25 ± 4.65	16	73.89 ± 5.77	18	75.91 ± 3.50	21
AMM^{max}	AMM_G	86.83 ± 4.26	61	73.32 ± 1.95	48	71.16 ± 4.94	51	73.57 ± 6.86	55	75.25 ± 3.18	63
AMM^{max}	$AMM_{G,s}$	86.83 ± 4.26	181	73.25 ± 2.03	143	71.19 ± 4.91	153	74.77 ± 6.85	163	75.25 ± 3.18	188
AMM^{max}	AMM_{nc}	86.83 ± 4.26	114	73.74 ± 2.48	92	70.36 ± 5.16	102	75.16 ± 5.71	138	76.44 ± 2.74	272
AMM^{max}	AMM_1	69.57 ± 3.99	15	73.12 ± 3.41	15	68.25 ± 2.80	16	71.02 ± 5.46	17	81.70 ± 3.02	19
AMM^{max}	AMM_{10ran}	77.82 ± 9.12	192	68.82 ± 4.73	138	73.58 ± 4.29	146	72.21 ± 9.35	164	74.16 ± 5.25	188
SVM	alter-$\propto$	53.26 ± 2.07	25	51.08 ± 2.35	27	50.90 ± 1.63	31	48.29 ± 4.51	38	41.66 ± 5.11	64
SVM	conv-$\propto$	77.80 ± 6.16	3924	66.14 ± 4.68	3790	57.94 ± 18.54	3244	61.37 ± 21.17	3327	63.73 ± 11.33	3603
	Oracle	92.81 ± 2.89	<1	92.68 ± 2.24	<1	92.44 ± 3.01	,1	92.61 ± 2.03	<1	92.99 ± 3.58	<1

Table 4.7: australian

algorithm		2 bags		4 bags		8 bags		16 bags		32 bags	
	AUC	time(s)	AUC	time(s)	AUC	time(s)	AUC	time(s)	AUC	time(s)	
	EMM	48.65 ± 7.54	<1	71.45 ± 16.59	<1	61.68 ± 7.47	<1	34.88 ± 12.33	<1	47.50 ± 22.77	<1
	MM	**99.42 ± 0.44**	2	**99.30 ± 0.39**	<1	**99.28 ± 0.25**	<1	**99.28 ± 0.37**	<1	**99.18 ± 0.47**	1
	LMM_G	**99.42 ± 0.44**	6	**99.33 ± 0.38**	3	**99.28 ± 0.25**	3	**99.35 ± 0.39**	3	**99.22 ± 0.46**	4
	$LMM_{G,s}$	**99.42 ± 0.44**	20	**99.34 ± 0.39**	10	**99.37 ± 0.24** ↑	11	**99.36 ± 0.38**	12	**99.23 ± 0.44**	15
	LMM_{nc}	**99.42 ± 0.44**	41	**99.29 ± 0.40**	39	**99.27 ± 0.25**	41	**99.30 ± 0.38**	59	**99.20 ± 0.47**	125
	Invcal	19.67 ± 2.23	5	59.50 ± 5.86	5	68 ± 5.27	5	60.83 ± 3.17	5	51.81 ± 4.72	5
AMM^{min}	AMM_{EMM}	**99.37 ± 0.42**	1	**99.33 ± 0.39**	1	99.17 ± 0.54	1	**99.34 ± 0.40**	2	99.29 ± 0.49	2
AMM^{min}	AMM_{MM}	**99.34 ± 0.46**	2	**99.30 ± 0.37**	1	**99.36 ± 0.27** ↑	2	99.29 ± 0.41	2	99.29 ± 0.48	2
AMM^{min}	AMM_G	**99.34 ± 0.46**	8	**99.30 ± 0.37** ↑	5	**99.36 ± 0.27** ↑	6	99.29 ± 0.41	7	99.30 ± 0.49	8
AMM^{min}	$AMM_{G,s}$	**99.34 ± 0.46**	23	**99.30 ± 0.37** ↑	16	**99.36 ± 0.27** ↑	19	99.29 ± 0.41	20	99.30 ± 0.49	25
AMM^{min}	AMM_{nc}	**99.34 ± 0.46**	43	**99.31 ± 0.35**	41	**99.36 ± 0.27** ↑	44	99.29 ± 0.41	62	99.29 ± 0.48	129
AMM^{min}	AMM_1	**99.35 ± 0.45**	<1	**99.32 ± 0.37**	1	99.20 ± 0.45	1	99.30 ± 0.42	1	**99.31 ± 0.48**	2
AMM^{min}	AMM_{10ran}	**99.36 ± 0.45**	8	99.11 ± 0.56	9	99.26 ± 0.35	11	99.28 ± 0.43	11	**99.32 ± 0.49** ↑	14
AMM^{max}	AMM_{EMM}	**99.42 ± 0.55**	6	99.02 ± 0.66	6	**99.32 ± 0.25** ↑	6	**99.43 ± 0.30** ↑	7	**99.40 ± 0.38** ↑	9
AMM^{max}	AMM_{MM}	99.01 ± 1.12	6	99.00 ± 0.64	6	**99.32 ± 0.35** ↑	6	**99.37 ± 0.38**	7	**99.39 ± 0.39** ↑	9
AMM^{max}	AMM_G	99.01 ± 1.12	20	98.99 ± 0.64	17	**99.33 ± 0.35** ↑	18	**99.37 ± 0.38**	21	**99.41 ± 0.39** ↑	27
AMM^{max}	$AMM_{G,s}$	99.01 ± 1.12	60	98.99 ± 0.64	52	99.19 ± 0.45	55	**99.37 ± 0.39**	63	**99.41 ± 0.39** ↑	82
AMM^{max}	AMM_{nc}	99.01 ± 1.12	55	98.99 ± 0.64	53	**99.32 ± 0.35** ↑	56	**99.37 ± 0.39**	76	**99.40 ± 0.38** ↑	148
AMM^{max}	AMM_1	99.09 ± 1.08	5	99.09 ± 0.46	5	**99.29 ± 0.26**	5	**99.37 ± 0.38**	6	**99.40 ± 0.38** ↑	8
AMM^{max}	AMM_{10ran}	98.97 ± 1.29	47	98.58 ± 0.75	48	**99.39 ± 0.27** ↑	52	**99.37 ± 0.38**	61	**99.36 ± 0.41** ↑	81
SVM	alter-∝	68.63 ± 17.63	24	93.24 ± 4.43	25	75.17 ± 7.19	33	90.11 ± 2.58	42	18.23 ± 5.67	82
SVM	conv-∝	**99.41 ± 0.48**	3346	56.33 ± 4.28	3043	77.71 ± 15.51	2800	32.90 ± 7.24	3036	67.21 ± 8.19	2037
	Oracle	99.48 ± 0.41	<1	99.53 ± 0.41	<1	99.31 ± 0.37	<1	99.43 ± 0.39	<1	99.32 ± 0.44	<1

Table 4.8: breastw

algorithm		2 bags		4 bags		8 bags		16 bags		32 bags	
	AUC	time(s)	AUC	time(s)	AUC	time(s)	AUC	time(s)	AUC	time(s)	
	EMM	60.69 ± 11.30	<1	51.83 ± 6.36	<1	52.99 ± 5.37	<1	53.83 ± 11.49	<1	52.95 ± 13.28	<1
	MM	**62.00 ± 6.44**	<1	70.48 ± 7.43	<1	67.13 ± 9.85	2	72.60 ± 9.35	1	72.05 ± 3.38	1
	LMM_G	**62.00 ± 6.44**	7	70.37 ± 7.47	6	72.15 ± 8.51	8	75.96 ± 10.38	8	75.47 ± 3.59	9
	$LMM_{G,s}$	**62.00 ± 6.44**	20	**72.10 ± 6.26**	20	**75.08 ± 7.14**	28	**78.54 ± 10.20**	26	**76.43 ± 3.10**	27
	LMM_{nc}	**62.00 ± 6.44**	31	70.45 ± 7.46	33	68.38 ± 9.69	52	74.04 ± 10.02	112	72.87 ± 3.20	345
	Invcal	38.73 ± 5.43	6	65.87 ± 6.70	6	59.30 ± 3.28	6	61.54 ± 4.17	6	59.53 ± 10.00	6
AMM^{min}	AMM_{EMM}	59.12 ± 8.86	3	56.23 ± 8.49	3	70.93 ± 10.31	3	**78.22 ± 6.00**	3	74.22 ± 6.35	4
AMM^{min}	AMM_{MM}	**77.44 ± 3.16**	2	78.84 ± 6.95	3	69.46 ± 6.44	4	71.93 ± 7.61	4	81.44 ± 5.18	4
AMM^{min}	AMM_G	**77.44 ± 3.16**	11	**79.41 ± 2.23**	12	72.62 ± 5.42	14	77.80 ± 8.11	14	**84.05 ± 2.33**	16
AMM^{min}	$AMM_{G,s}$	**77.44 ± 3.16**	34	**79.41 ± 2.23**	36	71.19 ± 5.38	41	76.71 ± 6.70	40	83.27 ± 3.14	47
AMM^{min}	AMM_{nc}	**77.44 ± 3.16**	36	78.33 ± 7.35	38	70.95 ± 4.69	57	74.67 ± 9.10	117	79.86 ± 4.87	352
AMM^{min}	AMM_1	38.69 ± 7.18	1	56.07 ± 14.68	2	**75.14 ± 4.78**	2	75.36 ± 5.64	3	77.51 ± 5.00	3
AMM^{min}	AMM_{10ran}	37.63 ± 4.19	10	77.75 ± 5.66	12	74.95 ± 5.64	15	76.59 ± 10.81	17	78.94 ± 4.17	23
AMM^{max}	AMM_{EMM}	50.94 ± 6.54	9	62.44 ± 9.94	9	57.53 ± 13.37	15	53.63 ± 14.71	17	67.63 ± 5.63	19
AMM^{max}	AMM_{MM}	43.05 ± 14.65	8	75.40 ± 4.64	9	63.72 ± 14.41	16	55.37 ± 10.19	18	69.49 ± 3.17	20
AMM^{max}	AMM_G	43.05 ± 14.65	28	78.19 ± 5.93	31	63.14 ± 7.53	51	61.32 ± 5.69	57	68.21 ± 9.35	62
AMM^{max}	$AMM_{G,s}$	43.05 ± 14.65	84	77.91 ± 6.36	91	62.57 ± 6.11	151	64.42 ± 10.77	168	69.47 ± 6.40	184
AMM^{max}	AMM_{nc}	42.92 ± 14.74	52	73.74 ± 7.21	57	60.39 ± 12.21	94	62.46 ± 15.13	162	68.63 ± 2.37	381
AMM^{max}	AMM_1	51.92 ± 19.91	7	59.89 ± 10.79	8	58.76 ± 12.16	14	62.31 ± 13.32	17	68.25 ± 6.42	18
AMM^{max}	AMM_{10ran}	56.39 ± 10.26	60	71.28 ± 8.76	68	65.01 ± 13.85	114	69.59 ± 9.96	139	74.40 ± 5.54	159
SVM	alter-∝	46.33 ± 2.73	18	50.82 ± 1.21	19	60.84 ± 5.51	23	62.20 ± 3.79	32	57.04 ± 10.10	49
SVM	conv-∝	25.27 ± 3.45	1438	35.96 ± 9.34	1460	50.31 ± 5.57	1439	35.46 ± 9.11	1423	50.13 ± 8.34	1427
	Oracle	86.19 ± 4.23	<1	87.80 ± 2.50	<1	87.05 ± 6.05	<1	86.53 ± 7.15	<1	87.97 ± 2.02	<1

Table 4.9: colic

algorithm		**2 bags**		**4 bags**		**8 bags**		**16 bags**		**32 bags**	
	AUC	time(s)	AUC	time(s)	AUC	time(s)	AUC	time(s)	AUC	time(s)	
	EMM	47.90 ± 4.51	<1	50.11 ± 5.17	<1	46.02 ± 5.88	<1	50.94 ± 1.61	<1	51.02 ± 2.55	<1
	MM	**61.07 ± 5.57**	<1	62.09 ± 4.00	<1	65.50 ± 6.54	2	65.61 ± 6.05	2	66.96 ± 4.56	2
	LMM_G	**61.07 ± 5.57**	4	62.14 ± 4.04	4	67.07 ± 6.36	6	66.43 ± 6.61	6	70.18 ± 4.76	7
	$LMM_{G,s}$	**61.07 ± 5.57**	11	62.75 ± 3.32	12	**67.91 ± 5.80**	16	**66.40 ± 6.90**	19	**70.43 ± 5.57**	21
	LMM_{nc}	**61.07 ± 5.57**	103	62.04 ± 4.00	87	65.47 ± 6.56	87	65.61 ± 6.06	113	67.01 ± 4.58	209
	Invcal	38.74 ± 5.43	6	**65.87 ± 6.70**	6	59.30 ± 3.28	6	61.53 ± 4.17	6	59.54 ± 10.00	6
AMM^{min}	AMM_{EMM}	53.89 ± 6.82	7	48.63 ± 8.71	7	53.24 ± 8.02	8	57.58 ± 3.44	9	63.64 ± 11.82	11
	AMM_{MM}	**60.45 ± 5.58**	5	63.33 ± 4.99	6	74.58 ± 4.76	6	72.43 ± 1.39	8	75.84 ± 5.24	7
	AMM_G	**60.45 ± 5.58**	17	**64.16 ± 6.99**	18	74.18 ± 4.34	21	72.08 ± 1.24	22	**75.94 ± 4.55**	24
	$AMM_{G,s}$	**60.45 ± 5.58**	52	**64.20 ± 7.24**	57	74.29 ± 4.50	57	72.18 ± 1.37	66	75.77 ± 4.44	74
	AMM_{nc}	**60.45 ± 5.58**	118	63.20 ± 6.09	101	**75.37 ± 4.42**	100	72.53 ± 1.25	130	**75.99 ± 5.26**	225
	AMM_1	37.08 ± 4.42	3	38.53 ± 2.97	3	41.89 ± 2.07	6	41.13 ± 2.58	9	47.09 ± 9.40	10
	AMM_{10ran}	49.12 ± 6.50	36	60.31 ± 5.57	38	73.82 ± 4.70	44	72.07 ± 3.22	54	74.73 ± 4.54	72
AMM^{max}	AMM_{EMM}	46.45 ± 3.30	18	46.31 ± 3.02	19	67.34 ± 13.42	19	72.41 ± 6.17	20	74.58 ± 4.63	22
	AMM_{MM}	52.47 ± 8.88	18	58.61 ± 12.19	18	65.14 ± 21.84	19	74.90 ± 4.86	20	74.88 ± 3.75	22
	AMM_G	52.47 ± 8.88	54	56.12 ± 12.25	53	74.93 ± 8.18	57	73.87 ± 4.55	60	75.43 ± 4.02	67
	$AMM_{G,s}$	52.47 ± 8.88	160	54.79 ± 11.61	158	74.84 ± 8.12	167	73.87 ± 4.55	180	75.40 ± 4.05	197
	AMM_{nc}	52.47 ± 8.88	154	49.24 ± 12.68	137	65.11 ± 21.84	137	74.89 ± 4.75	167	74.70 ± 3.71	269
	AMM_1	58.39 ± 13.20	17	61.04 ± 14.43	17	69.66 ± 16.93	17	**76.49 ± 3.29**	18	75.44 ± 3.65	20
	AMM_{10ran}	50.47 ± 9.69	168	56.78 ± 10.89	164	60.41 ± 15.48	160	61.62 ± 18.81	170	73.25 ± 6.97	191
SVM	alter-∝	49.36 ± 1.68	34	49.59 ± 1.58	37	48.43 ± 2.23	40	48.85 ± 1.55	47	51.05 ± 2.72	64
	conv-∝	29.70 ± 2.03	6031	**64.15 ± 5.43**	6343	63.01 ± 2.59	6362	62.01 ± 3.61	6765	63.17 ± 3.62	7004
	Oracle	79.43 ± 2.88	<1	78.95 ± 3.99	<1	79.18 ± 1.70	<1	79.42 ± 2.80	<1	79.02 ± 3.62	<1

Table 4.10: german

algorithm		**2 bags**		**4 bags**		**8 bags**		**16 bags**		**32 bags**	
	AUC	time(s)	AUC	time(s)	AUC	time(s)	AUC	time(s)	AUC	time(s)	
	EMM	51.82 ± 12.39	<1	50.43 ± 23.03	<1	55.09 ± 19.44	<1	49.55 ± 17.47	<1	63.49 ± 18.11	<1
	MM	**68.75 ± 6.09**	<1	60.24 ± 13.54	<1	80.35 ± 9.42	<1	76.11 ± 6.66	1	83.50 ± 6.22	1
	LMM_G	**68.75 ± 6.09**	3	68.04 ± 8.53	3	82.87 ± 6.16	4	**82.92 ± 1.28**	4	85.85 ± 3.84	6
	$LMM_{G,s}$	**68.75 ± 6.09**	9	**69.04 ± 6.52**	12	**83.68 ± 5.90**	13	**82.96 ± 1.79**	14	**86.36 ± 3.94**	17
	LMM_{nc}	**68.75 ± 6.09**	11	60.40 ± 14.18	12	80.24 ± 9.74	189	78.14 ± 4.98	42	84.47 ± 5.06	119
	Invcal	28.84 ± 4.96	4	70.58 ± 6.45	4	37.33 ± 10.31	4	44.96 ± 9.64	4	62.76 ± 15.05	4
AMM^{min}	AMM_{EMM}	60.50 ± 30.88	<1	63.36 ± 28.50	1	72.05 ± 19.17	1	80.87 ± 15.51	1	**91.63 ± 6.10** ↑	2
	AMM_{MM}	86.59 ± 6.14	1	80.57 ± 16.72	1	87.96 ± 4.50	2	90.04 ± 5.14	2	91.45 ± 5.70 ↑	2
	AMM_G	86.59 ± 6.14	5	86.70 ± 5.45	5	87.46 ± 2.67	6	**91.06 ± 2.87**	7	**91.55 ± 5.93** ↑	9
	$AMM_{G,s}$	86.59 ± 6.14	15	86.70 ± 5.45	16	88.31 ± 4.00	18	90.86 ± 2.81	21	**91.55 ± 5.93** ↑	27
	AMM_{nc}	86.59 ± 6.14	13	78.97 ± 16.78	14	87.82 ± 4.42	21	90.48 ± 3.53	45	91.25 ± 5.77	125
	AMM_1	**90.62 ± 5.82**	<1	89.19 ± 5.90	1	88.64 ± 3.21	1	90.78 ± 2.10	1	91.03 ± 5.82	1
	AMM_{10ran}	78.38 ± 30.44	5	87.32 ± 4.71	6	89.85 ± 2.31	7	**91.02 ± 2.49**	9	90.47 ± 6.39	14
AMM^{max}	AMM_{EMM}	85.74 ± 13.28	3	84.60 ± 10.87	4	84.60 ± 7.84	3	89.83 ± 2.72	5	71.65 ± 18.52	6
	AMM_{MM}	85.35 ± 11.06	4	82.43 ± 9.76	4	**90.49 ± 4.75**	4	89.92 ± 2.90		89.35 ± 6.98	7
	AMM_G	85.35 ± 11.06	13	87.18 ± 6.56	13	**90.49 ± 4.75**	13	89.58 ± 2.79	16	88.55 ± 9.71	23
	$AMM_{G,s}$	85.35 ± 11.06	39	**90.49 ± 5.05**	40	**90.58 ± 4.77**	40	89.58 ± 2.79	49	89.94 ± 6.63	67
	AMM_{nc}	85.35 ± 11.06	20	82.73 ± 9.23	21	89.84 ± 4.24	30	90.06 ± 3.20	54	89.54 ± 6.60	140
	AMM_1	72.77 ± 37.27	4	89.31 ± 3.99	3	89.68 ± 3.79	3	90.62 ± 3.18	5	87.97 ± 9.42	6
	AMM_{10ran}	89.96 ± 5.62	32	89.93 ± 5.02	31	88.03 ± 3.16	30	90.80 ± 3.61	38	89.61 ± 8.68	54
SVM	alter-∝	47.75 ± 17.58	15	59.72 ± 18.21	16	62.32 ± 12.83	20	58.49 ± 10.98	27	48.33 ± 12.77	47
	conv-∝	46.18 ± 43.41	1211	87.13 ± 5.30	1185	69.03 ± 23.18	1197	42.78 ± 23.51	1188	50.34 ± 15.75	1080
	Oracle	91.72 ± 3.95	<1	91.22 ± 4.09	<1	91.27 ± 2.88	<1	91.54 ± 2.76	<1	91.42 ± 5.46	<1

Table 4.11: heart

algorithm		2 bags		4 bags		8 bags		16 bags		32 bags	
	AUC	time(s)	AUC	time(s)	AUC	time(s)	AUC	time(s)	AUC	time(s)	
	EMM	44.28 ± 12.13	<1	51.86 ± 8.01	<1	50.69 ± 6.34	<1	44.60 ± 3.91	<1	48.91 ± 11.73	<1
	MM	**64.81 ± 8.82**	<1	77.74 ± 5.23	1	78.95 ± 7.36	1	86.76 ± 2.96	1	88.13 ± 4.16	2
	LMM_G	**64.81 ± 8.82**	5	80.80 ± 2.32	6	**83.46 ± 4.62**	5	87.12 ± 2.23	7	**88.24 ± 4.41**	7
	$LMM_{G,s}$	**64.81 ± 8.82**	14	**82.12 ± 2.50**	15	83.24 ± 4.84	15	**87.23 ± 1.57**	17	87.99 ± 4.58	21
	LMM_{nc}	**64.81 ± 8.82**	20	79.39 ± 2.12	22	81.18 ± 6.40	32	87.05 ± 2.48	68	**88.34 ± 4.32**	182
	Invcal	35.34 ± 8.76	5	44.78 ± 15.37	5	53.28 ± 9.02	5	53.52 ± 8.51	5	54.08 ± 9.53	5
AMM^{min}	AMM_{EMM}	56.77 ± 6.42	2	**85.07 ± 5.24**	2	86.04 ± 5.21	2	86.81 ± 3.81	2	86.71 ± 3.54	3
AMM^{min}	AMM_{MM}	46.67 ± 8.53	3	84.52 ± 4.60	2	84.23 ± 6.67	2	85.92 ± 4.48	3	87.77 ± 5.56	3
AMM^{min}	AMM_G	46.67 ± 8.53	10	85.05 ± 4.11	9	85.28 ± 6.19	9	85.97 ± 3.19	11	88.85 ± 5.15	12
AMM^{min}	$AMM_{G,s}$	46.67 ± 8.53	28	84.63 ± 3.80	26	85.28 ± 6.19	27	86.01 ± 4.37	30	88.85 ± 5.15	36
AMM^{min}	AMM_{nc}	46.67 ± 8.53	24	**85.16 ± 4.39**	26	84.77 ± 6.45	36	85.96 ± 4.50	72	87.57 ± 5.23	174
AMM^{min}	AMM_1	51.47 ± 13.46	1	83.65 ± 3.89	2	**87.51 ± 4.24**	2	86.76 ± 4.07	2	87.83 ± 5.05	2.11
AMM^{min}	AMM_{10ran}	56.92 ± 22.42	10	80.39 ± 6.36	11	85.89 ± 5.52	12	87.32 ± 3.17	13	87.81 ± 6.52	15
AMM^{max}	AMM_{EMM}	57.99 ± 8.96	10	76.31 ± 5.29	10	82.07 ± 4.47	11	86.99 ± 7.23	11	87.08 ± 5.86	12
AMM^{max}	AMM_{MM}	**74.57 ± 18.16**	10	75.32 ± 4.74	10	78.65 ± 7.93	11	88.84 ± 3.10	12	**90.01 ± 5.50**	13
AMM^{max}	AMM_G	**74.57 ± 18.16**	32	78.06 ± 5.11	33	83.24 ± 6.54	35	89.98 ± 3.08 ↑	38	88.41 ± 5.94	41
AMM^{max}	$AMM_{G,s}$	**74.57 ± 18.16**	96	79.21 ± 4.58	98	83.36 ± 6.61	104	**90.88 ± 3.11** ↑	112	88.41 ± 5.94	121
AMM^{max}	AMM_{nc}	**74.57 ± 18.16**	47	75.80 ± 5.14	50	80.22 ± 6.95	61	88.05 ± 2.47	99	89.19 ± 5.45	198
AMM^{max}	AMM_1	65.53 ± 17.30	10	77.29 ± 6.63	9	82.10 ± 7.95	10	85.45 ± 3.31	11	89.01 ± 7.02	12
AMM^{max}	AMM_{10ran}	65.05 ± 16.59	85	79.60 ± 6.56	82	78.56 ± 4.77	88	88.44 ± 3.22	94	89.37 ± 6.67	109
SVM	alter-∝	43.07 ± 6.05	22	44.58 ± 4.95	24	69.24 ± 4.99	27	67.72 ± 12.25	55	59.67 ± 7.01	49
SVM	conv-∝	36.67 ± 7.44	1316	44.55 ± 9.58	1280	57.84 ± 5.98	1788	65.93 ± 3.90	887	47.58 ± 11.29	1287
	Oracle	90.07 ± 5.04	<1	89.99 ± 4.23	<1	90.08 ± 5.50	<1	89.42 ± 6.34	<1	90.22 ± 5.17	<1

Table 4.12: ionosphere

algorithm		2 bags		4 bags		8 bags		16 bags		32 bags	
	AUC	time(s)	AUC	time(s)	AUC	time(s)	AUC	time(s)	AUC	time(s)	
	EMM	57.91 ± 22.04	<1	59.05 ± 10.46	<1	51.43 ± 17.22	<1	45.39 ± 23.81	<1	61.30 ± 17.86	<1
	MM	77.45 ± 6.14	<1	78.97 ± 3.54	<1	79.85 ± 4.14	<1	82.74 ± 2.11	1	87.45 ± 3.57	1
	LMM_G	77.45 ± 6.14	3	78.34 ± 2.82	3	81.93 ± 3.81	3	87.52 ± 2.71	5	90.43 ± 3.20	6
	$LMM_{G,s}$	77.45 ± 6.14	9	78.34 ± 2.82	8	**83.87 ± 3.63**	9	**87.71 ± 2.56**	13	**91.06 ± 3.00**	14
	LMM_{nc}	77.45 ± 6.14	31	78.43 ± 2.74	31	80.02 ± 4.02	35	83.50 ± 2.46	54	88.10 ± 3.57	122
	InvCal	33.74 ± 24.95	4	36.46 ± 5.27	4	72.54 ± 5.79	4	61.89 ± 6.25	4	59.91 ± 8.79	4
AMM^{min}	AMM_{EMM}	**81.07 ± 8.12**	2	78.56 ± 8.66	2	90.56 ± 3.44	2	92.08 ± 1.78	2	93.14 ± 2.04	3
AMM^{min}	AMM_{MM}	75.64 ± 5.02	2	68.54 ± 4.90	2	87.10 ± 4.16	2	**92.66 ± 1.99**	3	93.50 ± 1.93	3
AMM^{min}	AMM_G	75.64 ± 5.02	6	69.27 ± 5.69	7	87.57 ± 4.48	8	92.45 ± 1.89	10	93.59 ± 1.83	11
AMM^{min}	$AMM_{G,s}$	75.64 ± 5.02	19	69.27 ± 5.69	22	87.86 ± 4.62	23	91.04 ± 3.82	30	92.97 ± 1.58	32
AMM^{min}	AMM_{nc}	75.64 ± 5.02	34	68.49 ± 4.86	35	88.33 ± 5.17	39	91.26 ± 3.98	59	**93.70 ± 2.09**	127
AMM^{min}	AMM_1	74.49 ± 6.08	1	68.66 ± 4.92	1	90.60 ± 3.18	2	92.41 ± 1.58	2	92.95 ± 1.75	2
AMM^{min}	AMM_{10ran}	76.42 ± 4.80	12	75.75 ± 5.07	16	**92.59 ± 0.22**	18	92.15 ± 1.44	15	92.46 ± 1.79	19
AMM^{max}	AMM_{EMM}	76.02 ± 12.70	4	78.42 ± 14.14	5	87.87 ± 1.94	5	87.88 ± 3.29	6	90.71 ± 2.79	8
AMM^{max}	AMM_{MM}	75.31 ± 13.69	5	**87.22 ± 3.13**	5	87.43 ± 2.59	6	88.85 ± 2.39	6	90.29 ± 2.47	9
AMM^{max}	AMM_G	75.31 ± 13.69	15	73.91 ± 16.06	17	87.89 ± 1.97	17	87.98 ± 3.27	21	90.29 ± 2.47	28
AMM^{max}	$AMM_{G,s}$	75.31 ± 13.69	44	67.48 ± 16.70	50	87.89 ± 1.97	51	87.98 ± 3.27	63	90.18 ± 3.26	82
AMM^{max}	AMM_{nc}	75.31 ± 13.69	43	82.97 ± 8.05	45	87.85 ± 2.00	49	88.91 ± 2.41	70	90.29 ± 2.47	144
AMM^{max}	AMM_1	77.35 ± 13.61	4	70.14 ± 17.19	5	84.17 ± 2.66	5	89.12 ± 2.31	6	90.94 ± 3.06	8
AMM^{max}	AMM_{10ran}	72.39 ± 14.33	36	82.49 ± 9.32	47	87.44 ± 1.52	47	85.79 ± 4.54	50	90.87 ± 2.53	69
SVM	alter-∝	40.88 ± 5.80	21	30.17 ± 7.47	23	68.26 ± 6.40	26	58.84 ± 21.21	33	37.17 ± 17.48	48
SVM	conv-∝	77.72 ± 6.23	3624	72.28 ± 8.88	2292	36.21 ± 8.38	2328	45.01 ± 14.91	2481	70.49 ± 5.59	2306
	Oracle	93.80 ± 1.06	<1	93.83 ± 1.67	<1	93.89 ± 1.89	<1	93.83 ± 1.62	<1	94.00 ± 1.42	<1

Table 4.13: vertebral column

algorithm		**2 bags**		**4 bags**		**8 bags**		**16 bags**		**32 bags**	
	AUC	time(s)	AUC	time(s)	AUC	time(s)	AUC	time(s)	AUC	time(s)	
	EMM	54.32 ± 8.79	<1	45.47 ± 15.63	<1	46.88 ± 6.06	1	55.20 ± 18.03	1	53.93 ± 10.59	1
	MM	94.56 ± 2.04	1	95.37 ± 2.62	2	95.65 ± 0.85	2	**96.33 ± 1.19**	2	96.74 ± 1.50	2
	LMM_G	94.56 ± 2.04	7	95.93 ± 2.47	8	95.87 ± 1.12	8	**96.41 ± 1.51**	9	**96.94 ± 1.67**	10
	$LMM_{G,s}$	94.56 ± 2.04	20	**96.03 ± 2.42**	22	**96.00 ± 1.18**	23	**96.38 ± 1.99**	25	96.81 ± 2.09	28
	LMM_{nc}	94.56 ± 2.04	28	95.83 ± 2.34	31	95.71 ± 0.92	43	96.23 ± 1.58	85	96.81 ± 1.50	234
	Invcal	**94.85 ± 1.71**	4	73.10 ± 2.21	4	77.86 ± 4.92	4	26.74 ± 6.82	4	79.77 ± 6.25	4
AMM^{min}	AMM_{EMM}	93.67 ± 1.84	2	95.04 ± 3.01	2	**96.18 ± 0.78**	2	96.43 ± 1.31	2	96.94 ± 1.62	3
	AMM_{MM}	93.48 ± 2.31	2	95.12 ± 2.89	3	96.10 ± 0.82	3	96.15 ± 1.31	4	**97.30 ± 1.58**	4
	AMM_G	93.48 ± 2.31	10	**95.61 ± 1.90**	12	95.92 ± 1.02	11	96.41 ± 1.12	13	**97.36 ± 1.47**	15
	$AMM_{G,s}$	93.48 ± 2.31	29	94.87 ± 3.02	33	95.34 ± 0.98	35	96.11 ± 1.30	39	**97.36 ± 1.47**	46
	AMM_{nc}	93.48 ± 2.31	32	95.38 ± 2.38	35	95.81 ± 1.01	46	96.03 ± 1.48	89	**97.38** ± 1.45	238
	AMM_1	93.57 ± 1.99	2	94.32 ± 3.36	2	**96.25 ± 0.66**	2	96.17 ± 1.20	2	96.83 ± 1.42	2
	AMM_{10ran}	**93.84 ± 2.23**	11	94.59 ± 3.56	11	95.85 ± 0.97	12	**96.63 ± 1.32**	15	96.66 ± 1.70	18
AMM^{max}	AMM_{EMM}	91.68 ± 0.81	11	94.97 ± 2.24	12	94.94 ± 1	13	95.83 ± 1.36	14	96.60 ± 1.31	15
	AMM_{MM}	92.47 ± 0.38	12	93.43 ± 4.07	13	93.71 ± 1.34	14	95.40 ± 1.10	15	96.77 ± 1.31	17
	AMM_G	92.47 ± 0.38	40	94.34 ± 2.65	34	94.03 ± 0.81	43	95.65 ± 1.70	48	96.45 ± 1.52	53
	$AMM_{G,s}$	92.47 ± 0.38	124	94.22 ± 2.87	127	94.03 ± 0.81	132	96.01 ± 1.83	142	96.37 ± 1.39	160
	AMM_{nc}	92.47 ± 0.38	65	94.96 ± 3.48	66	94.07 ± 0.78	78	95.14 ± 1.18	124	96.74 ± 1.31	275
	AMM_1	91.60 ± 1.29	11	94.48 ± 2.14	12	94.34 ± 0.82	12	95.36 ± 1.56	13	96.54 ± 1.51	15
	AMM_{10ran}	90.49 ± 2.02	101	94.59 ± 2.85	103	94.19 ± 0.73	104	95.73 ± 1.83	112	96.21 ± 1.67	128
SVM	alter-∝	51.58 ± 3.27	19	62.74 ± 4.27	21	60.88 ± 3.50	25	63.01 ± 9.51	33	41.87 ± 7.12	57
	conv-∝	5.63 ± 2.03	1848	47.22 ± 4.92	1807	19.62 ± 5.91	1855	57.54 ± 11.22	1598	46.27 ± 9.48	1281
	Oracle	97.11 ± 1.31	<1	97.43 ± 2.25	<1	97.06 ± 0.87	<1	97.33 ± 1.38	<1	97.52 ± 1.49	<1

Table 4.14: vote (feature *physician-fee-freeze* was removed to make the problem more difficult)

algorithm		**2 bags**		**4 bags**		**8 bags**		**16 bags**		**32 bags**	
	AUC	time(s)	AUC	time(s)	AUC	time(s)	AUC	time(s)	AUC	time(s)	
	EMM	**70.38 ± 20.39**	<1	56.72 ± 29.85	<1	55.42 ± 20.70	<1	65.82 ± 21.45	<1	46.85 ± 16.71	<1
	MM	66.45 ± 5.42	1	82.41 ± 6.76	1	85.28 ± 4.80	1	90.35 ± 3.73	1	95.57 ± 2.45	1
	LMM_G	66.45 ± 5.42	4	89.72 ± 3.73	5	90.69 ± 5.30	5	94.09 ± 3.45	5	**97.74 ± 0.67**	6
	$LMM_{G,s}$	66.45 ± 4.412	13	**93.32 ± 2.94**	13	**92.68 ± 6.06**	14	**95.53 ± 2.40**	15	**97.69 ± 0.90**	19
	LMM_{nc}	66.45 ± 5.42	9	84.00 ± 5.48	11	86.30 ± 4.18	18	91.10 ± 4.52	40	96.28 ± 2.06	116
	Invcal	58.96 ± 5.77	6	81.38 ± 4.59	6	55.18 ± 9.59	6	63.07 ± 12.61	6	71.01 ± 18.19	6
AMM^{min}	AMM_{EMM}	80.27 ± 18.08	1	90.33 ± 8.87	1	91.46 ± 10.59	1	88.97 ± 6.26	1	88.34 ± 22.79	2
	AMM_{MM}	61.84 ± 9.20	2	85.56 ± 7.20	1	88.70 ± 8.31	2	93.78 ± 9.12	2	98.66 ± 1.11	2
	AMM_G	61.84 ± 9.20	6	93.06 ± 7.88	7	93.42 ± 8.24	7	96.09 ± 8.18	7	**99.33 ± 1.01**	9
	$AMM_{G,s}$	61.84 ± 9.20	17	94.87 ± 5.68	18	93.00 ± 8.95	20	96.09 ± 8.18	21	**99.33 ± 1.01**	27
	AMM_{nc}	61.84 ± 9.20	10	87.03 ± 3.93	13	88.23 ± 7.90	20	97.49 ± 5.06	43	**99.33 ± 1.01**	119
	AMM_1	82.21 ± 11.39	<1	94.12 ± 6.34	1	99.60 ± 0.60	1	96.03 ± 7.57	1	97.03 ± 3.66	1
	AMM_{10ran}	58.75 ± 31.30	4	**99.47 ± 0.68**	5	99.52 ± 0.45	6	**99.59 ± 0.54**	7	98.95 ± 1.66	10
AMM^{max}	AMM_{EMM}	74.23 ± 32.62	3	85.52 ± 17.48	4	99.67 ± 0.74	5	98.09 ± 3.09	6	92.00 ± 11.55	7
	AMM_{MM}	88.23 ± 18.56	5	97.60 ± 2.40	4	87.42 ± 27.76	6	99.42 ± 0.79	7	98.61 ± 1.69	8
	AMM_G	88.23 ± 18.56	15	88.41 ± 20	15	**100.00 ± 0.00** ↑	19	**99.63 ± 0.66**	20	98.61 ± 1.69	25
	$AMM_{G,s}$	88.23 ± 18.56	44	79.11 ± 23.90	44	**100.00 ± 0.00** ↑	56	**99.63 ± 0.66**	59	98.61 ± 1.69	75
	AMM_{nc}	88.23 ± 18.56	19	85.44 ± 19.04	21	86.17 ± 27.19	32	99.36 ± 0.74	56	98.61 ± 1.69	135
	AMM_1	75.24 ± 21.10	3	80.45 ± 10.01	4	91.83 ± 14.63	5	91.79 ± 9.05	5	88.01 ± 9.78	7
	AMM_{10ran}	**97.54 ± 1.55**	30	96.80 ± 3.94	32	99.46 ± 0.82	41	99.21 ± 0.79	47	98.54 ± 1.66	58
SVM	alter-∝	52.68 ± 2.54	14	36.53 ± 10.97	16	65.54 ± 2.26	19	29.15 ± 9.60	32	86.22 ± 11.93	44
	conv-∝	54.31 ± 4.63	831	70.23 ± 6.58	794	52.88 ± 13.86	840	55.60 ± 11.29	659	11.58 ± 7.84	495
	Oracle	99.69 ± 0.52	<1	99.80 ± 0.44	<1	99.60 ± 0.43	<1	99.80 ± 0.44	<1	99.78 ± 0.33	<1

Table 4.15: wine

4.13 References

To the best of our knowledge, the setting was originally introduced by Kuck and de Freitas [2005], inspired by previous work on multiple instance learning with threshold functions for Computer Vision [Kück et al., 2004]. The seminal work of Kuck and de Freitas [2005] proposes a hierarchical graphical model that generates labels consistent with the proportions. Training is performed in a fully Bayesian fashion by MCMC sampling. Some scaling issues on following this approach are highlighted by Quadrianto et al. [2009]. A similar but simpler graphical model is learned via EM algorithm in Wager et al. [2015].

Somehow independently, Chen et al. [2006] and its follower Musicant et al. [2007] offer a variety of standard machine learning methods such as Bayesian networks, ensemble trees, KNNs, SVMs and neural networks designed to generate self-consistent labels. Along this line, Hernández and Inza [2011]; Hernández-González et al. [2013] rely respectively on naive Bayes and structure learning of Bayesian networks with missing data. Other methods jointly exploit the proportions and geometrical assumptions on the data, by mixing classification with clustering [Chen et al., 2009; Stolpe and Morik, 2011; Cui et al., 2016]. An adaptation of latent discriminant analysis to LLP is given in Pérez-Ortiz et al. [2016].

It seems that the important contribution of Quadrianto et al. [2009] was initially conceived with no connection with previous work until N. De Freitas and A. Smola discussed the topic "while walking along the beach in Australia"[2] at a Machine Learning Summer School in 2008. This piece of work is central to the Thesis, since it is the first published instance of Loss Factorization for weakly supervised learning, with the related estimation of the mean operator. The paper also shows empirically that the Mean Map algorithm outperforms multiple strong baselines as well as the approach of Kuck and de Freitas [2005].

There are several SVM-based algorithms for LLP. Rüping [2010] propose to optimize an inverted calibration function [Platt, 1999] by margin maximization, with the objective of matching bag-wise average prediction with the given proportions. Instead, Yu et al. [2013] follow a more traditional formulation of SVMs, by augmenting the search space with the latent labels. Two alternative approximations are proposed: a convex relaxation, with better performance but hard to scale, and a more practical alternating optimization similar to AMM. Yu et al. [2013] show how their implementation outperform most of other known methods at date. The success of Yu et al. [2013] have inspired several SVM-based solutions [Wang et al., 2015; Ni et al., 2015; Qi et al., 2016; Chen et al., 2016].

Deep neural networks have recently been used for LLP. Fan et al. [2014] solve a density estimation problem with a restricted Boltzmann machines to learn a generative model. Li and Taylor [2015] formulate an alternating algorithm similar to Yu et al. [2013] and our AMM, but utilizing a convolutional neural network as the core model. A rather original insight is contained in the work of Kotzias et al. [2015], with a specific application on sentiment analysis *of sentences* instead of documents. In a

[2]Excerpt from a workshop talk by Nando de Freitas at NIPS'15, Montreal.

first phase, a convolutional network is trained for predicting document sentiments, which is the level of aggregated supervision that is available; in a second phase, a sentence embedding is obtained from the network by removing the top layer; finally, the architecture is retrained with a loss penalizing mistakes at document level predictions with an additional manifold regularization. The sentence embedding step is essential for constructing a suitable feature representation for the problem and, importantly, it is learned from label proportions only.

Prior work on theory of LLP belongs to two main categories. The first is about estimation guarantee of either the mean operator [Quadrianto et al., 2009] in RKHS or the resulting classifier [Altun and Smola, 2006; Quadrianto et al., 2009]. Those have already been discussed above. The second regards generalization bounds on predicting the bag proportions. Yu et al. [2014b] introduce the framework of *empirical proportion risk estimation*, showing that it is possible to bound generalization error of predicting bag proportions; in turn, the result is used to bound the more interesting error on label prediction by the use of distributional assumptions on the bag assignment process.

LLP has been studied in conjunction with *domain adaptation* in Ardehaly and Culotta [2016]. A model is learned from label proportions on the source domain and used to generate label proportions for a target domain; a self-training process is repeated to adapt the model from source to target.

Finally, extension of the LLP setting have been recently proposed, where the supervision is weaker but still represented by proportional quantities: *learning from positive-unlabeled proportions* [Hernández-González et al., 2015, 2016], *ballpark learning* [Hope and Shahaf, 2016], *i.e.* learning from bounds on the label proportions, *learning from histograms and order statistics* [Bhowmik et al., 2015]. The last setting is also linked with the broad topic of *data imputation* with the help of auxiliary aggregated information [Park and Ghosh, 2012, 2013] and with the *ecological inference*, presented next.

4.13.1 Ecological inference

We conclude this literature review with a mention to a related yet fundamentally different learning scenario, which originated outside the Machine Learning community. The problem deals with recovering information from aggregate data and thus share the intent of learning from label proportions. Although, in contrast with the topic of this Chapter, the objective is not to classify individual observations, but "disaggregate" variables from an higher level group representation into constituent parts.

The iconic application is inferring electorate behavior: given turnout results for several parties and proportions of some population strata, *e.g.* percentages of ethnic groups, for many geographical regions such as counties, the aim is to recover contingency tables for parties $\times$ groups for all those counties. In the language of probability the problem is isomorphic to the following: given two random variables and their respective marginal distributions — conditioned to another variable —, reconstruct their conditional joint distribution. Let us simplify the problem in that the feature

vector x and the label y are both scalar binary variables. As in LLP, the the problem has another dimension, the one of bags $j \in [n]$ which often stands for a geographical descriptor. The ecological inference problem is then summarized as: assuming to know the marginals — conditioned to j — $p(x|j)$ and $p(y|j)$, can we infer the joint distribution $p(x, y|j)$?

The ecological inference arises in a diversity of applied fields such as Econometrics [Cross and Manski, 2002; Cho and Manski, 2008], Sociology and Political Science [King, 1997; King et al., 2004] and Epidemiology [Wakefield and Shaddick, 2006], with a long history [Robinson, 1950]; interestingly, the Empirical Software Engineering community has also explored the idea [Posnett et al., 2011].

As with weak supervision, the problem is fundamentally under-determined and any solution can only provide either loose deterministic bounds [Duncan and Davis, 1953; Cross and Manski, 2002; Cho and Manski, 2008] or needs to enforce additional assumptions and prior knowledge on the data domain [King, 1997]. A decade ago, the problem has witnessed a period of renaissance along with the publication of a diversity of methods from the second family, mostly inspired by several models and related distributional assumptions collected in the book of King et al. [2004]. In contrast, Judge et al. [2004] follow the road of a minimal subset of assumptions and frame the inference as an optimization problem. The approach favors one solution according to some information-theoretic solution, *e.g.* the Cressie-Read power divergence, intended as an entropic measure of the joint distribution. Along this last line of work, we have co-authored a new solution based on *optimal transport* [Villani, 2008] between marginal distributions as presented in Muzellec et al. [2017].

On the one hand, LLP is a strictly more ambitious problem than ecological inference. In fact, in learning from label proportions — the marginals for y — the prediction happens at the level of each individual. In contrast, the goal of ecological inference is to "disaggregate" data into another, finer-grain level of representation, the one of the bags. From this point of view, a solution for LLP can be used to infer the unknown joint distributions. On the other hand, LLP is assuming a much richer representation of the data as a starting point. Individual features vectors, *e.g.* the individual demographics of voters, are not known and not necessary for solving ecological inference. Yet it has been shown that such information can be beneficially integrated into learning methods for ecological inference [Flaxman et al., 2015].

Chapter 5

Learning with noisy labels I: theory for linear models

The Chapter specializes our approach to the problem of learning a classifier under asymmetric label noise. The above theory and algorithms are reviewed to adapt to this particular case of weak supervision. A novel estimator for the label sufficient statistic is defined by its property of unbiasedness. We also characterize the whole family of linear-odd losses by an approximate criterion of label noise robustness. Simple experiments with simulated noise corroborate both theoretical and algorithmic statements.

5.1 Motivation

Large datasets used in training modern machine learning models are often affected by label noise. The problem is pervasive for a simple reason: manual expert-labeling of each instance at a large scale is not feasible and so researchers and practitioners often resort to cheap but imperfect surrogates. Two such popular surrogates are crowdsourcing using non-expert labelers and (especially for images) the use of a search engine to query instances by a keyword, where it is assumed that the keyword is a valid label for what is collected from the query [Fergus et al., 2010; Schroff et al., 2011; Divvala et al., 2014; Krause et al., 2016]. Both approaches offer the possibility to scale the acquisition of training labels. However, they invariably result in the introduction of label noise, which may adversely affect model training. On top of those issues, label noisy is intrinsic and virtually impossible to eliminate from many data domains, because of inherent ambiguity and task misspecification [Misra et al., 2016].

In the Thesis we adopt the nomenclature of Table 5.1. We focus on label noise that is *asymmetric*, *i.e.* we work in the class-conditional noise setting of Natarajan et al. [2013] . This is clearly a simplification of real-world information corruption, which may also be *instance dependent*. Little is known about learning under such noise, with few exceptions [Xiao et al., 2015; Ghosh et al., 2015; Menon et al., 2016].

name	assumption	example of recent study
symmetric	$p(\tilde{y}\|y,\boldsymbol{x}) = p(\tilde{y})$	van Rooyen et al. [2015]
asymmetric	$p(\tilde{y}\|y,\boldsymbol{x}) = p(\tilde{y}\|y)$	Natarajan et al. [2013]
instance dependent	no assumption	Menon et al. [2016]

Table 5.1: Label noise types considered in literature.

5.2 Learning setting

In learning with asymmetric noisy labels (ALN), $\widetilde{\mathcal{S}}$ is a set of examples drawn from a distribution $\widetilde{\mathcal{D}}$, which samples from $\mathcal{D}$ and flips labels at random. Each example $(\boldsymbol{x}_i, \tilde{y}_i)$ is $(\boldsymbol{x}_i, -y_i)$ with probability at most $1/2$ or it is $(\boldsymbol{x}_i, y_i)$ otherwise. The *noise rates* are label dependent by $(p_+, p_-) \in [0, 1/2)^2$ respectively for positive and negative examples. The marginal feature distribution is not affected by noise. In sum, examples are drawn from the distribution:

$$p(\boldsymbol{x}, \tilde{y}) = p_+ \cdot p(y = 1|\boldsymbol{x})p(\boldsymbol{x}) + p_- \cdot p(y = -1|\boldsymbol{x})p(\boldsymbol{x}), \tag{5.1}$$

with $p_+ = p(\tilde{y} = -1|y = 1)$ and $p_- = p(\tilde{y} = 1|y = -1)$.

5.3 Estimating the sufficient statistic and μSGD

The schema of the solution for this learning problem follows closely what we did in the Chapter 4 for LLP and implements our two-step procedure: estimate the sufficient statistic and solve ERM. The estimation step turns out to be much simpler in this scenario. Although, notice that we work under the assumption of knowing the noise rates (p_+, p_-). Towards a more realistic treatment, Chapter 6 will deal with the estimation of those quantities as well.

We construct an estimator for the mean operator that is unbiased with respect to the noisy distribution. The result builds on Natarajan et al. [2013, Lemma 1] that provides a recipe for unbiased estimators *of losses*. Instead of estimating the whole ℓ, as consequence of the Factorization Theorem 18 we act on the sufficient statistic.

Theorem 52. *The estimator defined as:*

$$\hat{\boldsymbol{\mu}}_{\mathcal{S}} \doteq \mathbb{E}_{\mathcal{S}}\left[\frac{y - (p_- - p_+)}{1 - p_- - p_+}\boldsymbol{x}\right] \tag{5.2}$$

is unbiased, that is, its expectation over the noise distribution $\widetilde{\mathcal{D}}$ is the population mean operator:

$$\mathbb{E}_{\widetilde{\mathcal{D}}}\left[\hat{\boldsymbol{\mu}}_{\widetilde{\mathcal{S}}}\right] = \boldsymbol{\mu}_{\mathcal{D}} \ . \tag{5.3}$$

The corresponding risk estimator $\hat{R}_{\mathcal{S},\ell}(\boldsymbol{\theta}) \doteq \frac{1}{2}R_{\mathcal{S}_{2x},\ell}(\boldsymbol{\theta}) + a\langle\boldsymbol{\theta}, \hat{\boldsymbol{\mu}}_{\mathcal{S}}\rangle$ is also unbiased.

Proof in 5.7.1. We have thus obtained a good candidate as input for any algorithm

Algorithm 8: μSGD for asymmetric noisy label

Input: $\widetilde{\mathcal{S}}$, ℓ is a-LOL, (p_+, p_-), $\lambda > 0$, $T > 0$
$\mathcal{S}_{2x} \leftarrow \{(\boldsymbol{x}_i, \sigma), i \in [m], \forall \sigma \in \mathcal{Y}\}$
$\hat{\boldsymbol{\mu}}_{\widetilde{\mathcal{S}}} \leftarrow \mathbb{E}_{\widetilde{\mathcal{S}}}\left[\frac{y-(p_- - p_+)}{1-p_- - p_+}\boldsymbol{x}\right]$
$\hat{\boldsymbol{\theta}} \leftarrow \mu\text{SGD}(\mathcal{S}_{2x}, \hat{\boldsymbol{\mu}}_{\widetilde{\mathcal{S}}}, \lambda, T)$
Output: $\hat{\boldsymbol{\theta}}$

Algorithm 9: μSGD with L_2 regularization

Input: $\mathcal{S}_{2x}$, $\boldsymbol{\mu}$, ℓ is a-LOL, $\lambda > 0$, $T > 0$
$\boldsymbol{\theta}^0 \leftarrow \mathbf{0}$
For any $t = 1, \ldots, T$:
  Pick $i \in [|\mathcal{S}_{2x}|]$ uniformly at random
  $\eta \leftarrow 1/(\lambda t)$
  Pick any $\boldsymbol{v} \in \partial\ell(y_i\langle\boldsymbol{\theta}^t, \boldsymbol{x}_i\rangle)$
  $\boldsymbol{\theta}^{t+1} \leftarrow (1 - \eta\lambda)\boldsymbol{\theta}^t - \eta(\boldsymbol{v} + a\boldsymbol{\mu}/2)$
  $\boldsymbol{\theta}^{t+1} \leftarrow \min\left\{\boldsymbol{\theta}^{t+1}, \boldsymbol{\theta}^{t+1}/(\sqrt{\lambda} \cdot \|\boldsymbol{\theta}^{t+1}\|_2)\right\}$
Output: $\boldsymbol{\theta}^T$

implementing our two-step approach, like μSGD (Algorithm 9). But there is more. On one hand, the estimators of Natarajan et al. [2013] may not be convex even when ℓ is so, but this is never the case with LOLS. In fact, $\ell(x) - \ell(-x) = 2ax$ may be seen as an alternative sufficient condition to Natarajan et al. [2013, Lemma 4] for convexity, without asking for differentiability.

The estimator of Equation 5.2 is all we need for presenting a complete version of the weakly supervised version of SGD given in Algorithm 8. We also restate Algorithm 9 for the sake of readability. This is another example of implementation of the Meta Algorithm 1, this time for ALN.

5.4 Generalization bounds

We now prove that *any* algorithm minimizing LOLS that uses the estimator in Equation 5.2 has a non-trivial generalization bound. We further assume that ℓ is Lipschitz.

Theorem 53. *Consider the setting of Theorem 23, except that here* $\hat{\boldsymbol{\theta}} = \operatorname{argmin}_{\boldsymbol{\theta}\in\mathcal{H}} \hat{R}_{\widetilde{\mathcal{S}},\ell}(\boldsymbol{\theta})$. *Then for any* $\delta > 0$, *with probability at least* $1 - \delta$*:*

$$R_{\mathcal{D},\ell}(\hat{\boldsymbol{\theta}}) - \inf_{\boldsymbol{\theta}\in\mathcal{H}} R_{\mathcal{D},\ell}(\boldsymbol{\theta}) \leq \frac{\sqrt{2}+1}{2}\frac{XHL}{\sqrt{m}} + c(X,H)\sqrt{\frac{1}{m}\log\left(\frac{2}{\delta}\right)} + \frac{2|a|XH}{1-p_- - p_+}\sqrt{\frac{d}{m}\log\left(\frac{2d}{\delta}\right)}\,. \tag{5.4}$$

Proof in 5.7.2. The complexity term is tighter than prior work. Natarajan et al. [2013, Theorem 3] prove a factor of $2L/(1-p_- - p_+)$ that may even be unbounded due to noise, while our estimate shows a constant of about $1.2 < 2$ and it is noise free. In fact, LOLS are such that noise affects only the linear component of the bound, as a direct effect of Factorization. Although we are not aware of any other such results, this is intuitive: the Rademacher complexity $\mathcal{R}(\mathcal{H} \circ \mathcal{S})$ is computed regardless of sample labels and therefore is unchanged by label noise[1]. Furthermore, depending on the loss, the effect of (limited) noise on generalization may also be negligible since $c(X, H)$ could be very large for losses like strongly convex.

The next Theorem comes in pair with Theorem 53: it holds regardless of algorithm and (linear-odd) loss of choice. In particular, we demonstrate that every learner enjoys a distribution-dependent property of robustness against ALN. No estimate of $\boldsymbol{\mu}$ is involved and hence the Theorem also applies to any naïve supervised learner *on* $\widetilde{\mathcal{S}}$ via linear-odd losses. We first recall a strong notion of robustness from Manwani and Sastry [2013]; van Rooyen et al. [2015].

Definition 54. *Let $(\boldsymbol{\theta}^\star, \tilde{\boldsymbol{\theta}}^\star)$ respectively be the minimizers of $(R_{\mathcal{D},\ell}(\boldsymbol{\theta}), R_{\widetilde{\mathcal{D}},\ell}(\boldsymbol{\theta}))$ in $\mathcal{H}$. ℓ is said ALN robust if for any $\mathcal{D}, \widetilde{\mathcal{D}}$:*

$$R_{\mathcal{D},\ell}(\boldsymbol{\theta}^\star) = R_{\mathcal{D},\ell}(\tilde{\boldsymbol{\theta}}^\star) \ . \tag{5.5}$$

The influential work of Long and Servedio [2010], together with van Rooyen et al. [2015], settles the question of whether commonly used loss function can be robust to label noise, in particular in the milder *symmetric* noise scenario. Rather negatively, the literature concludes that no *convex potentials*[2] is immune to such noise. Instead, the already cited unhinged loss $\ell(x) = 1 - x$, a linear loss, satisfies Equation 5.5 [van Rooyen et al., 2015].

To study the LOLS under ASL, we take a more pragmatic point of view and extend Definition 5.5 by the weaker formulation of ϵ-robustness.

Definition 55. *In the context of Definition 54, ℓ is said ϵ-ALN robust if for any $\mathcal{D}, \widetilde{\mathcal{D}}$:*

$$R_{\widetilde{\mathcal{D}},\ell}(\boldsymbol{\theta}^\star) - R_{\widetilde{\mathcal{D}},\ell}(\tilde{\boldsymbol{\theta}}^\star) \leq \epsilon \ . \tag{5.6}$$

The distance of the two minimizers is measured by ℓ-risk under expected label noise. 0-ALN robust losses are also ALN robust: in fact if $R_{\widetilde{\mathcal{D}},\ell}(\boldsymbol{\theta}^\star) = R_{\widetilde{\mathcal{D}},\ell}(\tilde{\boldsymbol{\theta}}^\star)$ then $\boldsymbol{\theta}^\star \in \operatorname{argmin}_{\boldsymbol{\theta}} R_{\widetilde{\mathcal{D}},\ell}(\boldsymbol{\theta})$. And hence if $R_{\widetilde{\mathcal{D}},\ell}(\boldsymbol{\theta})$ has a unique global minimum, that is $\boldsymbol{\theta}^\star$. More generally we have the following result.

Theorem 56. *Assume $\{\boldsymbol{\theta} \in \mathcal{H} : ||\boldsymbol{\theta}||_2 \leq H\}$. Then every a-LOL is ϵ-ALN. That is:*

$$R_{\widetilde{\mathcal{D}},\ell}(\boldsymbol{\theta}^\star) - R_{\widetilde{\mathcal{D}},\ell}(\tilde{\boldsymbol{\theta}}^\star) \leq 4|a|H \max\{p_+, p_-\}\|\boldsymbol{\mu}_{\mathcal{D}}\|_2 \tag{5.7}$$

[1] Strictly speaking, this is true only considering the upper bound $L \cdot \mathcal{R}(\mathcal{H} \circ \mathcal{S})$ for Lipschitz losses, without dependency on ℓ. In contrast, $\mathcal{R}(\ell \circ \mathcal{H} \circ \mathcal{S})$ would depend on the sample labels.

[2] Namely, losses $\ell \in C^1$, convex, such that $\ell(0) < 0$ and $\ell(x) \to 0$ for $x \to \infty$. Many convex potentials are LOLS but not all. An example is e^{-x}.

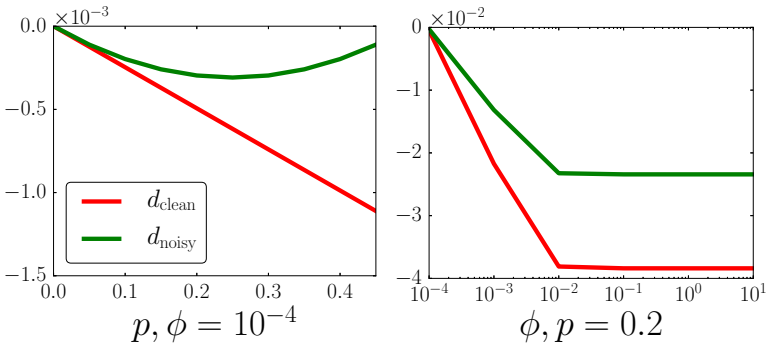


Figure 5.1: Behavior of Theorem 56 on synthetic data. Definition of the axes within the text.

Proof in 5.7.3. The reader might question the bound for the fact that the quantity on the right-hand side can change by rescaling $\boldsymbol{\mu}_{\mathcal{D}}$ by X, *i.e.* the max L_2 norm of observations in the space $\mathcal{X}$. Although, such transformation would affect ℓ-risks on the left-hand side as well, balancing the effect. With this in mind, we formulate the result without making explicit dependency on X. Unlike Theorem 53, this bound holds *in expectation* over the noisy risk $R_{\tilde{\mathcal{D}},\ell}$. Its shape depends on the population mean operator, a *distribution-dependent* quantity, which empirical approximation is readily available by Equation 5.2. There are two immediate corollaries.

Corollary 57. *If* $\|\boldsymbol{\mu}_{\mathcal{D}}\|_2 = 0$ *then every* LOL *is ALN for any* $\widetilde{\mathcal{D}}$.

Corollary 58. *Suppose additionally that* ℓ *is once differentiable and* γ*-strongly convex. Then:*

$$\left\|\boldsymbol{\theta}^\star - \tilde{\boldsymbol{\theta}}^\star\right\|_2^2 \leq \frac{2\epsilon}{\gamma} \ . \tag{5.8}$$

Proofs in 5.7.4. When $\|\boldsymbol{\mu}_{\mathcal{D}}\|_2 = 0$, we obtain optimality for all LOLS. Notice that this is not necessarily a condition detrimental to learning. In fact, as remarked in Lemma 28, $\|\boldsymbol{\mu}_{\mathcal{D}}\|_2 = 0$ does not implies $\mathrm{Cov}_{\mathcal{D}}[\boldsymbol{x}, y] = 0$. The second corollary goes further, limiting the minimizers' distance when losses are differentiable and strongly convex. But even more generally, under some compactness assumptions on the domain of ℓ, Theorem 56 tells us more: in the case $R_{\tilde{\mathcal{D}},\ell}(\boldsymbol{\theta})$ has a unique global minimizer, the smaller $\|\boldsymbol{\mu}_{\mathcal{D}}\|_2$, the closer the minimizer *on noisy data* $\tilde{\boldsymbol{\theta}}^\star$ will be to the minimizer *on clean data* $\boldsymbol{\theta}^\star$. Therefore, assuming an efficient algorithm that computes a model not far from the global optimum $\tilde{\boldsymbol{\theta}}^\star$, that will be not far from $\boldsymbol{\theta}^\star$ either. This is true in spite of the presence of local minima and/or saddle points.

Let us compare the significance of Theorem 56 with what is known in literature. Long and Servedio [2010] prove that no convex loss is noise robust, that is, 0-ALN robust. This is not a contradiction. To show the negative statement, the authors craft

a case of $\mathcal{D}$ breaking all such losses. In fact that choice of $\mathcal{D}$ does not meet optimality in our bound, because:

$$\|\boldsymbol{\mu}_{\mathcal{D}}\|_2 = \frac{1}{4}(18\gamma^2 + 6\gamma + 1) > 0 \ , \tag{5.9}$$

with $\gamma \in (0, 1/6)$. The worst-case result of Long and Servedio [2010], like any extreme-case argument, should be handled with care. It does not give the big picture for all data we may encounter in the real world, but only the most pessimistic. We present such a global view which appears better than expected: learning from noisy data does not necessarily reduce convex losses to a singleton [van Rooyen et al., 2015] but depends on the data at hand — via the mean operator — for a broad set of losses.

We also compare our ϵ-ALN robustness with a robustness bound proposed by Ghosh et al. [2015]. In the same setting as Definition 54:

$$R_{\mathcal{D},\ell}(\tilde{\boldsymbol{\theta}}^*) \leq \frac{1}{1 - 2\max(p_-, p_+)} R_{\mathcal{D},\ell}(\boldsymbol{\theta}^*) \, . \tag{5.10}$$

While relating the (non-noisy) ℓ-risks, this bound is not data dependent and may not be informative for high noise rates.

Finally, Manwani and Sastry [2013] give a sufficient condition for loss function to be ALN robust (Definition 54):

$$\ell(x) + \ell(-x) = \text{const} \ \Rightarrow \ \text{0-ALN} \tag{5.11}$$

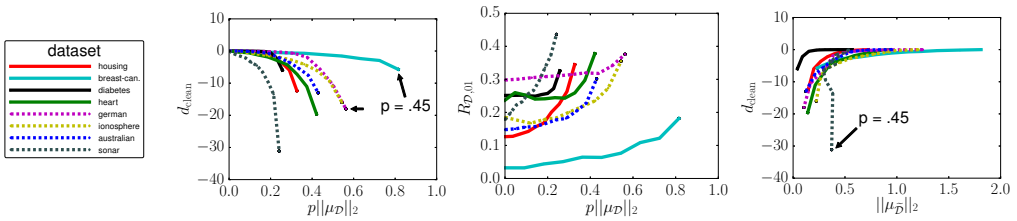
Quite surprisingly, Loss Factorization also marries two opposite views in one formula:

$$\ell(x) = \frac{1}{2}(\underbrace{\ell(x) + \ell(-x)}_{=\text{const} \ \Rightarrow \ \text{0-ALN}} + \underbrace{\ell(x) - \ell(-x)}_{=ax \ \Rightarrow \ \epsilon\text{-ALN}}) \tag{5.12}$$

As a by product, we confirm the peculiar role of unhinged loss, being the only function — modulo linear transformation — that satisfies both conditions.

5.5 Experiments

We begin by building a toy planar dataset to probe the behavior of Theorem 56. It is made of four observations: $(0, 1)$ and $(\phi/3, 1/3)$ three times, with the first example the only negative, repeated 5 times. We consider this the distribution $\mathcal{D}$ so as to calculate $\|\boldsymbol{\mu}_{\mathcal{D}}\|_2 = \phi^2/4$. We fix $p_+, p_- = 0.2 = p$ and control ϕ to measure the discrepancy $d_{\text{noisy}} \doteq R_{\widetilde{\mathcal{D}},\ell}(\boldsymbol{\theta}^\star) - R_{\widetilde{\mathcal{D}},\ell}(\tilde{\boldsymbol{\theta}}^\star)$, its counterpart d_{clean} computed on $\mathcal{D}$, and how the two minimizers "differ in sign" by $d_{\text{models}} \doteq \langle \boldsymbol{\theta}^\star, \tilde{\boldsymbol{\theta}}^\star \rangle / \|\boldsymbol{\theta}^\star\|_2 \|\tilde{\boldsymbol{\theta}}^\star\|_2$. The same simulation is run varying the noise rates with constant $\phi = 10^{-4}$. We learn with square loss and $\lambda = 10^{-6}$. Results are in Figure 5.1. The closer the parameters to 0, the smaller $d_{\text{clean}} - d_{\text{noisy}}$, while they are equal when each parameter is individually 0. d_{models} is negligible, which is good news for the 01-risk on sight.



(a) $p\|\mu_{\mathcal{D}}\|_2$ *vs.* d_{clean} (b) $p\|\mu_{\mathcal{D}}\|_2$ *vs.* $R_{\mathcal{D},01}$ (c) $\|\mu_{\widetilde{\mathcal{D}}}\|_2$ *vs.* d_{clean}

Figure 5.2: How mean operator and noise rate condition risks. $d_{\text{clean}} \doteq R_{\mathcal{D},\ell}(\boldsymbol{\theta}^\star) - R_{\mathcal{D},\ell}(\tilde{\boldsymbol{\theta}}^\star)$.

The next results are obtained by Algorithm 8 on UCI data. We learn with logistic loss, without model intercept and set $\lambda = 10^{-6}$ and $T = 4 \cdot 2m$ (4 epochs). We measure d_{clean} and $R_{\mathcal{D},01}$, injecting symmetric label noise $p \in [0, 0.45)$ and averaging over 25 runs. Again, we consider *the whole distribution* so as to play with the ingredients of Theorem 56. Figure 5.2a confirms how the combined effect of $p\|\boldsymbol{\mu}_{\mathcal{D}}\|_2$ can explain most variation of d_{clean}. While this is not strictly implied by Theorem 56 that only involves d_{noisy}, the observed behavior is expected. A similar picture is given in Figure 5.2b which displays the true risk $R_{\mathcal{D},01}$ computed on the minimizer $\tilde{\boldsymbol{\theta}}^\star$ of $\widetilde{\mathcal{S}}$. From 5.2a and 5.2b we also deduce that large $\|\boldsymbol{\mu}_{\mathcal{D}}\|_2$ is a good proxy for generalization with linear classifiers; see the relative difference between points at the same level of noise. Finally, we have also monitored $\boldsymbol{\mu}_{\widetilde{\mathcal{D}}}$. Figure 5.2c shows that large $\|\boldsymbol{\mu}_{\widetilde{\mathcal{D}}}\|_2$ indicates small d_{clean} as well. This remark can be useful in practice, when the norm can be estimated from $\widetilde{\mathcal{S}}$, as opposite to p and $\boldsymbol{\mu}_{\mathcal{D}}$, and used to anticipate the effect of noise on the task at hand.

We conclude with a systematic study of hold-out error of μSGD. The same datasets are now split in $1/5$ test and $4/5$ training sets once at random. In contrast with the previous experimental setting we perform cross-validation of $\lambda \in 10^{\{-3,\dots,+3\}}$ on 5-folds in the training set. We compare with vanilla SGD run on corrupted sample $\widetilde{\mathcal{S}}$ and measure the gain from estimating $\hat{\boldsymbol{\mu}}_{\widetilde{\mathcal{S}}}$. The other parameters l, T, λ are the same for both algorithms; the learning rate η is untouched from Shalev-Shwartz et al. [2011] and not tuned for μSGD. The only differences are in input and gradient update. Table 5.2 reports test error for SGD and its difference with μSGD, for a range of values of (p_-, p_+). μSGD beats systematically SGD with large noise rates, and yet performs in pair under low or null noise. Interestingly, in the presence of very intense noise $p_+ \approx .5$, μSGD still learns sensible models and improves up to 55% relatively to the error of SGD, which is often doomed to random guess.

5.6 Discussion

In this Chapter we have shown that both theoretical analysis and algorithms are easily adaptable to the case of noisy labels, granted that we have access of a good estimator of the label sufficient statistic. In this context, we have also derived a

$(p_-, p_+) \rightarrow$	(.00, .00)		(.20, .00)		(.20, .10)		(.20, .20)		(.20, .30)		(.20, .40)		(.20, .49)	
dataset	SGD	μSGD	SGD	μSGD	SGD	μSGD	SGD	μSGD	SGD	μSGD	SGD	μSGD	SGD	μSGD
australian	0.13	+.01	0.15	**−.01**	0.14	±.00	0.14	+.01	0.16	**−.01**	0.26	**−.09**	0.45	**−.25**
breast-can.	0.02	+.01	0.03	±.00	0.03	±.00	0.03	±.00	0.05	**−.01**	0.11	**−.06**	0.17	**−.08**
diabetes	0.28	**−.03**	0.29	**−.03**	0.29	**−.03**	0.27	**−.02**	0.28	**−.02**	0.39	**−.13**	0.59	**−.22**
german	0.27	**−.02**	0.26	±.00	0.27	**−.02**	0.29	**−.02**	0.31	**−.01**	0.31	±.00	0.31	±.00
heart	0.15	+.01	0.17	**−.01**	0.16	±.00	0.17	±.00	0.18	**−.01**	0.26	**−.08**	0.35	**−.15**
housing	0.17	**−.03**	0.23	**−.05**	0.22	**−.04**	0.20	**−.02**	0.22	**−.03**	0.34	**−.12**	0.41	**−.13**
ionosphere	0.14	+05	0.19	**−.05**	0.20	**−.05**	0.20	**−.03**	0.21	**−.03**	0.35	**−.13**	0.54	**−.29**
sonar	0.27	±.00	0.29	+.02	0.29	+.01	0.34	**−.04**	0.36	**−.03**	0.43	**−.10**	0.45	**−.05**

Table 5.2: Test error for SGD and μSGD over 25 trials of artificially corrupted datasets.

general property of label noisy robustness for linear-odd losses. The next Chapter discusses some practical open issues, in particular the restriction to linear models and the strong hypothesis of known noise rates.

5.7 Appendix: proofs

5.7.1 Proof of Theorem 52

Natarajan et al. [2013, Lemma 1] provide an unbiased estimator for a loss $\ell(x)$ computed on x of the form:

$$\hat{\ell}(y\langle \boldsymbol{\theta}, \boldsymbol{x}_i \rangle) \doteq \frac{(1 - p_{-y}) \cdot \ell(\langle \boldsymbol{\theta}, \boldsymbol{x}_i \rangle) + p_y \cdot \ell(-\langle \boldsymbol{\theta}, \boldsymbol{x}_i \rangle)}{1 - p_- - p_+} \ . \tag{5.13}$$

We apply the Lemma for estimating the mean operator instead. We are allowed to do so by the very result of the Factorization Theorem, since the noise corruption has only an effect on the linear-odd term of the loss. The estimator of the sufficient statistic of a single example $y\boldsymbol{x}$ is:

$$\hat{\boldsymbol{z}} \doteq \frac{1 - p_{-y} + p_y}{1 - p_- - p_+} y\boldsymbol{x} \tag{5.14}$$

$$= \frac{1 - (p_- - p_+)y}{1 - p_- - p_+} y\boldsymbol{x} \tag{5.15}$$

$$= \frac{y - (p_- - p_+)}{1 - p_- - p_+} \boldsymbol{x} \ , \tag{5.16}$$

and its average, *i.e.* the mean operator estimator, is:

$$\hat{\boldsymbol{\mu}}_{\mathcal{S}} \doteq \mathbb{E}_{\mathcal{S}} \left[\frac{y - (p_- + p_+)}{1 - p_- - p_+} \boldsymbol{x} \right] \ , \tag{5.17}$$

such that in expectation over the noisy distribution it holds $\mathbb{E}_{\widetilde{\mathcal{D}}}[\hat{\boldsymbol{z}}] = \boldsymbol{\mu}_{\mathcal{D}}$. Moreover, the corresponding risk estimator $\hat{R}$ enjoys the same unbiasedness property. In fact:

$$\hat{R}_{\widetilde{\mathcal{D}},\ell}(\boldsymbol{\theta}) = \frac{1}{2} R_{\mathcal{D}_{2x},\ell}(\boldsymbol{\theta}) + \mathbb{E}_{\widetilde{\mathcal{D}}} \left[a\langle \boldsymbol{\theta}, \hat{\boldsymbol{z}} \rangle \right] \tag{5.18}$$

$$= \frac{1}{2} R_{\mathcal{D}_{2x},\ell}(\boldsymbol{\theta}) + a\langle \boldsymbol{\theta}, \hat{\boldsymbol{\mu}}_{\widetilde{\mathcal{D}}} \rangle \tag{5.19}$$

$$= \frac{1}{2} R_{\mathcal{D}_{2x},\ell}(\boldsymbol{\theta}) + a\langle \boldsymbol{\theta}, \boldsymbol{\mu}_{\mathcal{D}} \rangle \tag{5.20}$$

$$= R_{\mathcal{D},\ell}(\boldsymbol{\theta}) \ , \tag{5.21}$$

by using the independence from labels, hence from noise, of $R_{\mathcal{D}_{2x},\ell}$.

5.7.2 Proof of Theorem 53

This is a version of Theorem 23 applied to ALN. The two results differ in three elements. First, we consider the generalization property of a minimizer $\hat{\boldsymbol{\theta}}$ that is learned on the corrupted sample $\widetilde{\mathcal{S}}$. Second, the minimizer is computed on the basis of the unbiased estimator of $\hat{\boldsymbol{\mu}}_{\widetilde{\mathcal{S}}}$ and not barely by the average $\boldsymbol{\mu}_{\widetilde{\mathcal{S}}}$. Third, as a consequence, Lemma 27 is not valid in this scenario. Therefore, we first prove

a version of the bound for the mean operator norm discrepancy while considering label noise.

Lemma 59. *Suppose* $\mathbb{R}^d \supseteq \mathcal{X} = \{\boldsymbol{x} : \|\boldsymbol{x}\|_2 \leq X < \infty\}$ *be the observations space. Let* $\widetilde{\mathcal{S}}$ *is a learning sample affected by ALN with noise rates* $(p_+, p_-) \in [0, 1/2)^2$*. Then for any* $\delta > 0$ *with probability at least* $1 - \delta$*:*

$$\left\|\hat{\boldsymbol{\mu}}_{\widetilde{\mathcal{D}}} - \hat{\boldsymbol{\mu}}_{\widetilde{\mathcal{S}}}\right\|_2 \leq \frac{X}{1 - p_- - p_+} \cdot \sqrt{\frac{d}{m} \log\left(\frac{d}{\delta}\right)} \,. \tag{5.22}$$

Proof. Let $\widetilde{\mathcal{S}}$ and $\widetilde{\mathcal{S}}'$ be two learning samples from the corrupted distribution $\widetilde{\mathcal{D}}$ that differ for only one example $(\boldsymbol{x}_i, \tilde{y}_i) \neq (\boldsymbol{x}_{i'}, \tilde{y}_{i'})$. Let first consider the one-dimensional case. We refer to the k-dimensional component of $\boldsymbol{\mu}$ with $\boldsymbol{\mu}^k$. For any $\widetilde{\mathcal{S}}, \widetilde{\mathcal{S}}'$ and any $k \in [d]$ it holds:

$$\begin{aligned}
\left|\hat{\boldsymbol{\mu}}^k_{\widetilde{\mathcal{S}}} - \hat{\boldsymbol{\mu}}^k_{\widetilde{\mathcal{S}}'}\right| &= \frac{1}{m}\left|\left(\frac{\tilde{y}_i - (p_- - p_+)}{1 - p_- - p_+}\right)\boldsymbol{x}^k_i - \left(\frac{\tilde{y}_{i'} - (p_- - p_+)}{1 - p_- - p_+}\right)\boldsymbol{x}^k_{i'}\right| && (5.23)\\
&= \frac{1}{m}\left|\frac{\tilde{y}_i \boldsymbol{x}^k_i}{1 - p_- - p_+} - \frac{\tilde{y}_{i'}\boldsymbol{x}^k_{i'}}{1 - p_- - p_+}\right| && (5.24)\\
&\leq \frac{X}{m(1 - p_- - p_+)}\left|\tilde{y}_i - \tilde{y}_{i'}\right| && (5.25)\\
&\leq \frac{2X}{m(1 - p_- - p_+)} \,. && (5.26)
\end{aligned}$$

This satisfies the bounded difference condition of McDiarmid's inequality, which let us write for any $k \in [d]$ and any $\epsilon > 0$ that:

$$\mathbb{P}\left(\left|\hat{\boldsymbol{\mu}}^k_{\mathcal{D}} - \hat{\boldsymbol{\mu}}^k_{\mathcal{S}}\right| \geq \epsilon\right) \leq \exp\left(-(1 - p_- - p_+)^2 \frac{m\epsilon^2}{2X^2}\right) \tag{5.27}$$

and the multi-dimensional case, by union bound:

$$\mathbb{P}\left(\exists k \in [d] : \left|\hat{\boldsymbol{\mu}}^k_{\mathcal{D}} - \hat{\boldsymbol{\mu}}^k_{\mathcal{S}}\right| \geq \epsilon\right) \leq d \exp\left(-(1 - p_- - p_+)^2 \frac{m\epsilon^2}{2X^2}\right) \,. \tag{5.28}$$

Then by negation:

$$\mathbb{P}\left(\forall k \in [d] : \left|\hat{\boldsymbol{\mu}}^k_{\mathcal{D}} - \hat{\boldsymbol{\mu}}^k_{\mathcal{S}}\right| \leq \epsilon\right) \geq 1 - d \exp\left(-(1 - p_- - p_+)^2 \frac{m\epsilon^2}{2X^2}\right) \,, \tag{5.29}$$

which implies that for any $\delta > 0$ with probability $1 - \delta$:

$$\frac{X}{(1 - p_- - p_+)}\sqrt{\frac{2}{m}\log\left(\frac{d}{\delta}\right)} \geq \left\|\hat{\boldsymbol{\mu}}_{\mathcal{D}} - \hat{\boldsymbol{\mu}}_{\mathcal{S}}\right\|_\infty \geq d^{-1/2}\left\|\boldsymbol{\mu}_{\mathcal{D}} - \boldsymbol{\mu}_{\mathcal{S}}\right\|_2 \,. \tag{5.30}$$

This concludes the proof. □

The proof of Theorem 53 follows the structure of Theorem 23's and elements of Theorem 3 of Natarajan et al. [2013]'s. Let $\hat{\boldsymbol{\theta}} = \operatorname{argmin}_{\boldsymbol{\theta}\in\mathcal{H}} \hat{R}_{\widetilde{\mathcal{D}},\ell}(\boldsymbol{\theta})$ and $\boldsymbol{\theta}^\star = \operatorname{argmin}_{\boldsymbol{\theta}\in\mathcal{H}} R_{\mathcal{D},\ell}(\boldsymbol{\theta})$. We have:

$$
\begin{aligned}
R_{\mathcal{D},\ell}(\hat{\boldsymbol{\theta}}) - R_{\mathcal{D},\ell}(\boldsymbol{\theta}^\star) &= \hat{R}_{\widetilde{\mathcal{D}},\ell}(\hat{\boldsymbol{\theta}}) - \hat{R}_{\widetilde{\mathcal{D}},\ell}(\boldsymbol{\theta}^\star) && (5.31)\\
&= \frac{1}{2} R_{\mathcal{D}_{2x},\ell}(\hat{\boldsymbol{\theta}}) + a\langle \hat{\boldsymbol{\theta}}, \hat{\boldsymbol{\mu}}_{\widetilde{\mathcal{D}}}\rangle - \frac{1}{2} R_{\mathcal{D}_{2x},\ell}(\boldsymbol{\theta}^\star) - a\langle \boldsymbol{\theta}^\star, \hat{\boldsymbol{\mu}}_{\widetilde{\mathcal{D}}}\rangle && (5.32)\\
&= \frac{1}{2}\left(R_{\mathcal{D}_{2x},\ell}(\hat{\boldsymbol{\theta}}) - R_{\mathcal{D}_{2x},\ell}(\boldsymbol{\theta}^\star)\right) + a\langle \hat{\boldsymbol{\theta}} - \boldsymbol{\theta}^\star, \hat{\boldsymbol{\mu}}_{\widetilde{\mathcal{D}}}\rangle && (5.33)\\
&= \frac{1}{2}\left(R_{\mathcal{S}_{2x},\ell}(\hat{\boldsymbol{\theta}}) - R_{\mathcal{S}_{2x},\ell}(\boldsymbol{\theta}^\star)\right) + a\langle \hat{\boldsymbol{\theta}} - \boldsymbol{\theta}^\star, \hat{\boldsymbol{\mu}}_{\widetilde{\mathcal{D}}}\rangle && (5.34)\\
&\quad + \frac{1}{2}\Big(R_{\mathcal{D}_{2x},\ell}(\hat{\boldsymbol{\theta}}) - R_{\mathcal{S}_{2x},\ell}(\hat{\boldsymbol{\theta}}) - R_{\mathcal{D}_{2x},\ell}(\boldsymbol{\theta}^\star) + R_{\mathcal{S}_{2x},\ell}(\boldsymbol{\theta}^\star)\Big) \quad \Big\} A_1 \ . && (5.35)
\end{aligned}
$$

Step 5.31 is due to unbiasedness. Again, rename Line 5.35 as A_1, which this time is bounded directly by Theorem 22. Next, we proceed as within the proof of Theorem 22 but now exploiting the fact that $\frac{1}{2} R_{\mathcal{S}_{2x},\ell}(\boldsymbol{\theta}) = \hat{R}_{\widetilde{\mathcal{S}},\ell}(\boldsymbol{\theta}) - a\langle \boldsymbol{\theta}, \hat{\boldsymbol{\mu}}_{\widetilde{\mathcal{D}}}\rangle$:

$$
R_{\mathcal{D},\ell}(\hat{\boldsymbol{\theta}}) - R_{\mathcal{D},\ell}(\boldsymbol{\theta}^\star) \leq \underbrace{\hat{R}_{\widetilde{\mathcal{S}},\ell}(\hat{\boldsymbol{\theta}}) - \hat{R}_{\widetilde{\mathcal{S}},\ell}(\boldsymbol{\theta}^\star)}_{A_2} + \underbrace{a\langle \hat{\boldsymbol{\theta}} - \boldsymbol{\theta}^\star, \hat{\boldsymbol{\mu}}_{\widetilde{\mathcal{D}}} - \hat{\boldsymbol{\mu}}_{\widetilde{\mathcal{S}}}\rangle}_{A_3} + A_1 \ . \tag{5.36}
$$

A_2 is never more than 0 because $\hat{\boldsymbol{\theta}}$ is the minimizer of $\hat{R}_{\widetilde{\mathcal{S}},\ell}(\boldsymbol{\theta})$. From the Cauchy-Schwarz inequality and bounded models it holds true:

$$
A_3 \leq |a| \left\| \hat{\boldsymbol{\theta}} - \boldsymbol{\theta}^\star \right\|_2 \cdot \left\| \hat{\boldsymbol{\mu}}_{\widetilde{\mathcal{D}}} - \hat{\boldsymbol{\mu}}_{\mathcal{S}} \right\|_2 \leq 2|a| H \left\| \hat{\boldsymbol{\mu}}_{\widetilde{\mathcal{D}}} - \hat{\boldsymbol{\mu}}_{\widetilde{\mathcal{S}}} \right\|_2 \ , \tag{5.37}
$$

for which we can call Lemma 59. Finally, by a union bound we get that for any $\delta > 0$ with probability $1 - \delta$:

$$
\begin{aligned}
R_{\mathcal{D},\ell}(\hat{\boldsymbol{\theta}}) - R_{\mathcal{D},\ell}(\boldsymbol{\theta}^\star) &\leq \frac{\sqrt{2}+1}{2} \cdot \frac{XHL}{\sqrt{m}} \\
&+ c(X,H)\sqrt{\frac{1}{m}\log\left(\frac{2}{\delta}\right)} + \frac{2|a|XH}{1-p_+-p_-}\sqrt{\frac{d}{m}\log\left(\frac{2d}{\delta}\right)} \ . && (5.38)
\end{aligned}
$$

5.7.3 Proof of Theorem 56

The proof draws ideas from Manwani and Sastry [2013]. Let us first assume symmetric noise, *i.e.* $p_+ = p_- = p$. For any $\boldsymbol{\theta}$ we have:

$$
\begin{aligned}
R_{\widetilde{\mathcal{D}},\ell}(\boldsymbol{\theta}^\star) - R_{\widetilde{\mathcal{D}},\ell}(\boldsymbol{\theta}) &= (1-p)\left(R_{\mathcal{D},\ell}(\boldsymbol{\theta}^\star) - R_{\mathcal{D},\ell}(\boldsymbol{\theta})\right) \\
&\quad + p\left(R_{\mathcal{D},\ell}(\boldsymbol{\theta}^\star) - R_{\mathcal{D},\ell}(\boldsymbol{\theta}) + 2a\langle \boldsymbol{\theta}^\star - \boldsymbol{\theta}, \boldsymbol{\mu}_{\mathcal{D}}\rangle\right) && (5.39)\\
&\leq \left(R_{\mathcal{D},\ell}(\boldsymbol{\theta}^\star) - R_{\mathcal{D},\ell}(\boldsymbol{\theta})\right) + 4|a|Hp\|\boldsymbol{\mu}_{\mathcal{D}}\|_2 && (5.40)\\
&\leq 4|a|Hp\|\boldsymbol{\mu}_{\mathcal{D}}\|_2 \ . && (5.41)
\end{aligned}
$$

We are working with LOLS, which are such that $\ell(x) = \ell(-x) + 2ax$ and therefore we can take Step 5.39. Step 5.40 follows from Cauchy-Schwartz inequality and bounded models. Step 5.41 is true because $\boldsymbol{\theta}^\star$ is the minimizer of $R_{\mathcal{D},\ell}(\boldsymbol{\theta})$. We have obtained a bound for any $\boldsymbol{\theta}$ and so for the supremum with regard to $\boldsymbol{\theta}$. Therefore:

$$\sup_{\boldsymbol{\theta}\in\mathcal{H}} \left(R_{\widetilde{\mathcal{D}},\ell}(\boldsymbol{\theta}^\star) - R_{\widetilde{\mathcal{D}},\ell}(\boldsymbol{\theta}) \right) = R_{\widetilde{\mathcal{D}},\ell}(\boldsymbol{\theta}^\star) - R_{\widetilde{\mathcal{D}},\ell}(\tilde{\boldsymbol{\theta}}) \ . \tag{5.42}$$

To lift the discussion to *asymmetric* label noise, risks have to be split into losses for negative and positive examples. Let $R_{\mathcal{D}^+,\ell}$ be the risk computed over the distribution of the positive examples $\mathcal{D}^+$ and $R_{\mathcal{D}^-,\ell}$ the one of the negatives, and denote the mean operators $\boldsymbol{\mu}_{\mathcal{D}^+}, \boldsymbol{\mu}_{\mathcal{D}^-}$ accordingly. Also, define the probability of positive and negative labels in $\mathcal{D}$ as $\pi_\pm = \mathbb{P}(y = \pm 1)$. The same manipulations for the symmetric case let us write:

$$\begin{aligned}
R_{\widetilde{\mathcal{D}},\ell}(\boldsymbol{\theta}^\star) - R_{\widetilde{\mathcal{D}},\ell}(\boldsymbol{\theta}) &= \pi_- \left(R_{\mathcal{D}^-,\ell}(\boldsymbol{\theta}^\star) - R_{\mathcal{D}^-,\ell}(\boldsymbol{\theta}) \right) + \pi_+ \left(R_{\mathcal{D}^+,\ell}(\boldsymbol{\theta}^\star) - R_{\mathcal{D}^+,\ell}(\boldsymbol{\theta}) \right) \\
&\quad + 2ap_-\pi_- \langle \boldsymbol{\theta}^\star - \boldsymbol{\theta}, \boldsymbol{\mu}_{\mathcal{D}^-} \rangle + 2ap_+\pi_+ \langle \boldsymbol{\theta}^\star - \boldsymbol{\theta}, \boldsymbol{\mu}_{\mathcal{D}^+} \rangle && (5.43) \\
&\leq \left(R_{\mathcal{D},\ell}(\boldsymbol{\theta}^\star) - R_{\mathcal{D},\ell}(\boldsymbol{\theta}) \right) + 2a \langle \boldsymbol{\theta}^\star - \boldsymbol{\theta}, p_-\boldsymbol{\mu}_{\mathcal{D}^-} + p_+\boldsymbol{\mu}_{\mathcal{D}^+} \rangle && (5.44) \\
&\leq 4|a|H \cdot \| p_-\pi_-\boldsymbol{\mu}_{\mathcal{D}^-} + p_+\pi_+\boldsymbol{\mu}_{\mathcal{D}^+} \|_2 && (5.45) \\
&\leq 4|a|H \max(p_-, p_+) \cdot \| \pi_-\boldsymbol{\mu}_{\mathcal{D}^-} + \pi_+\boldsymbol{\mu}_{\mathcal{D}^+} \|_2 && (5.46) \\
&= 4|a|H \max(p_-, p_+) \cdot \| \boldsymbol{\mu}_{\mathcal{D}} \|_2 \ . && (5.47)
\end{aligned}$$

Then, we conclude the proof by the same argument for the symmetric case.

5.7.4 Proof of Corollaries 57 and 58

The first corollary is immediate by using the additional assumption. For the second, if ℓ is once differentiable and γ-strongly convex in the $\boldsymbol{\theta}$ argument, so is the risk $R_{\tilde{\mathcal{D}},\ell}$ by composition with linear functions. Notice also that $\nabla R_{\widetilde{\mathcal{D}},\ell}(\tilde{\boldsymbol{\theta}}^\star) = 0$ because $\tilde{\boldsymbol{\theta}}^\star$ is the minimizer. Therefore:

$$\begin{aligned}
\epsilon &\geq R_{\widetilde{\mathcal{D}},\ell}(\boldsymbol{\theta}^\star) - R_{\widetilde{\mathcal{D}},\ell}(\tilde{\boldsymbol{\theta}}^\star) && (5.48) \\
&\geq \left\langle \nabla R_{\widetilde{\mathcal{D}},\ell}(\tilde{\boldsymbol{\theta}}^\star), \boldsymbol{\theta}^\star - \tilde{\boldsymbol{\theta}}^\star \right\rangle + \frac{\gamma}{2} \left\| \boldsymbol{\theta}^\star - \tilde{\boldsymbol{\theta}}^\star \right\|_2^2 && (5.49) \\
&\geq \frac{\gamma}{2} \left\| \boldsymbol{\theta}^\star - \tilde{\boldsymbol{\theta}}^\star \right\|_2^2 \ , && (5.50) \\
& && (5.51)
\end{aligned}$$

which means that:

$$\left\| \boldsymbol{\theta}^\star - \tilde{\boldsymbol{\theta}}^\star \right\|_2^2 \leq \frac{2\epsilon}{\gamma} \ . \tag{5.52}$$

5.8 References

Learning with noisy labels has been widely investigated in literature; we point any interested reader to Frénay and Verleysen [2014] for a comprehensive survey.

From the theoretical standpoint label noise has been studied in two different regimes, with vastly different conclusions. In the case of low-capacity (typically linear) models, even mild symmetric label noise can produce solutions that are akin to random guessing [Long and Servedio, 2010]. On the other hand, the Bayes-optimal classifier remains unchanged under symmetric [Natarajan et al., 2013; Menon et al., 2015] and even instance dependent label noise [Menon et al., 2016] implying that high-capacity models are robust to essentially any level of such noise, given a sufficient number of samples. A caveat with the latter is that label noise adversely affects the number of samples needed for learning [van Rooyen, 2015, Chapter 3].

In practice, and with more relevance for the mentioned low capacity models, a common strategy for learning with noisy labels is to engineer a loss function that is more suited to handle noise. Suppose one wishes to minimize a loss ℓ on clean data. When the level of noise is known *a priori* — as we have assumed in this Chapter —, the cited Natarajan et al. [2013] provide the general form of a *noise corrected* loss $\hat{\ell}$ such that minimization of $\hat{\ell}$ on noisy data is equivalent (by unbiasedness) to minimization of ℓ on clean data. The algorithms proposed in this Chapter can be seen as an application of the method of Natarajan et al. [2013] for correcting linear-odd losses. In the idealized case of symmetric label noise, for certain ℓ one in fact does not need to know the noise rate: Manwani and Sastry [2013]; Ghosh et al. [2015] give a sufficient condition for which ℓ is robust, and several examples of such robust non-convex losses, while van Rooyen et al. [2015] show that the (convex) unhinged loss is its own noise-corrected loss. Other proposed non-convex losses are in Masnadi-Shirazi and Vasconcelos [2009]; Masnadi-Shirazi et al. [2010]; Ding and Vishwanathan [2010].

Chapter 6

Learning with noisy labels II: deep neural networks, multi-class, noise estimation

In this Chapter, we greatly extend the discussion on label noise from Chapter 5. We propose three important improvements to the idealized learning scenario presented so far. First and foremost, the hypothesis space considered here is the one of deep neural networks. Second, it follows naturally from this choice that we study multi-class classification. Last but not least, we relax the assumption concerning the knowledge of the noisy rates and formulate an estimator for multi-class asymmetric noise; importantly, this last step let us formulate an end-to-end framework. The Chapter is accompanied by a extensive experimental analysis on MNIST, IMDB, CIFAR-10, CIFAR-100, other than on a large scale dataset of clothing images with a variety of convolutional and recurrent architectures.

6.1 Motivation

The goal of this Chapter is to effectively train deep neural networks with modern architectures under label noise. In Chapter 5 we discussed one strand of research for combating label noise based on the idea of loss correction. This broad line of work is strongly inspired by a theoretical analysis of this learning setting and is mostly developed within the Machine Learning community. In Chapter 5 we actually interpret this approach within the framework proposed by the Thesis, that is, sufficient statistic estimation followed by learning with linear-odd losses. This interpretation is grounded on the assumption of working with linear (or kernel) models.

Another strand of work is on *ad-hoc deep architectures* tailored to tolerate label noise, primarily developed in Computer Vision [Mnih and Hinton, 2012; Reed et al., 2015; Sukhbaatar et al., 2015; Xiao et al., 2015]. While some such approaches have shown good experimental performance on specific domains, they lack a solid theoretical framework, and often need large a amount of clean labels to obtain acceptable results — in particular, for pre-training or validating hyper-parameters [Xiao et al., 2015; Krause et al., 2016; Reed et al., 2015].

We unify the above research streams by introducing two alternative procedures for loss correction. In doing so, this Chapter slightly departs from the main approach proposed in the Thesis, with the aim of laying a bridge between those two areas.

Both procedures amount to simple linear algebra provided that we know a stochastic matrix T summarizing the probability of one class being flipped into another under noise. The first procedure, a multi-class extension of Natarajan et al. [2013] applied to neural networks, is called "*backward*" correction as it multiplies the loss by T^{-1}. The second, inspired by Sukhbaatar et al. [2015], is named "*forward*" correction as it multiplies the network predictions by T. We prove that both procedures enjoy formal guarantees of robustness with regard to the clean data distribution. Since we only operate on the loss function, the approach is architecture independent and not tied to a particular application domain, other than viable for any loss (even non-linear-odd).

In real applications practitioners may be able to obtain a good estimate of T by polishing a subset of the available training set [Xiao et al., 2015] — something undoubtedly useful and often necessary for tuning hyper-parameters and testing the model anyway. Nevertheless, we take a further step extending the noise estimator of Menon et al. [2015] to the multi-class setting. Incidentally, we also prove that, when the network only non-linearity is ReLU, the Hessian of the loss is not affected by noise.

A clear motivation of this Chapter is to push for a practical application of our formal work on label noise. We apply our loss corrections to image recognition on MNIST, CIFAR-10, CIFAR-100 and sentiment analysis on IMDB; we simulate corruption by artificially injecting noise on the training labels. In order to show that no architectural choice is the secret ingredient of our robustification recipe, we experiment with a variety of network modules: convolutions and pooling [LeCun et al., 1998], dropout [Srivastava et al., 2014], batch normalization [Ioffe and Szegedy, 2015], word embedding and residual units [He et al., 2016a,b]. Additional tests on LSTM [Hochreiter and Schmidhuber, 1997] confirms that the procedures can be seamlessly applied to recurrent neural networks as well. Comparisons with non-corrected losses and several known methods confirm robustness of our two procedures, with the forward correction dominating the backward. Unsurprisingly, the noise estimator is the bottleneck in obtaining near-perfect robustness, yet in most experiments our approach is often the best compared to prior work. Finally, we experiment with the $1M$ clothing images dataset of Xiao et al. [2015], establishing a new state of the art.

6.2 Learning setting

In supervised c-class classification, we have a feature space $\mathcal{X} \subseteq \mathbb{R}^d$ and a label space $\mathcal{Y} = \{\boldsymbol{e}^i : i \in [c]\}$. Note that each $\boldsymbol{y}$ only has one non-zero value at the coordinate corresponding to the underlying label.

A n-layer neural network comprises a transformation $\boldsymbol{h}\colon \mathcal{X} \to \mathbb{R}^c$, where $\boldsymbol{h} = (\boldsymbol{h}^{(n)} \circ \boldsymbol{h}^{(n-1)} \circ \cdots \circ \boldsymbol{h}^{(1)})$ is the composition of a number of intermediate transforma-

tions, the layers, defined by:

$$(\forall i \in [n-1])\, \boldsymbol{h}^{(i)}(\boldsymbol{z}) = \sigma(W^{(i)}\boldsymbol{z} + \boldsymbol{b}^{(i)}) \ ,$$
$$\boldsymbol{h}^{(n)}(\boldsymbol{z}) = W^{(i)}\boldsymbol{z} + \boldsymbol{b}^{(i)} \ .$$

where $W^{(i)} \in \mathbb{R}^{d^{(i)} \times d^{(i-1)}}$ and $\boldsymbol{b}^{(i)} \in \mathbb{R}^{d^{(i)}}$ are parameters to be estimated[1], and σ is any activation function that acts *coordinate-wise*, such as the rectified linear unit (ReLU) $\sigma(\boldsymbol{x})_i = \max(0, \boldsymbol{x}_i)$. Observe that the final layer applies a *linear* projection, unlike all preceding layers. To simplify notation, we write:

$$(\forall i \in [n])\, \boldsymbol{x}^{(i)} \doteq \boldsymbol{h}^{(i)}(\boldsymbol{x}^{(i-1)}),$$

with the base case $\boldsymbol{x}^{(0)} \doteq \boldsymbol{x}$, so that *e.g.* $\boldsymbol{x}^{(1)}$ is exactly the representation in the first layer. Without loss of generality, we assume all layers to be *fully connected*, or dense; for example, convolutions can be represented by dense layers with shared sparse weights. The coordinates of $\boldsymbol{h}(\boldsymbol{x})$ represent the relative weights that the model assigns to each class $i \in [c]$ to be predicted. The predicted label is thus simply $\boldsymbol{y}(\boldsymbol{x}) = \arg\max_{i \in [c]} \boldsymbol{h}_i(\boldsymbol{x})$.

In the training phase, the output of the final layer is contrasted with the true label $\boldsymbol{y}$ via two steps. First, $\boldsymbol{h}(\cdot)$ passes through the *softmax* function $e^{\boldsymbol{h}_i(\boldsymbol{x})} / \sum_{k=1}^{c} e^{\boldsymbol{h}_k(\boldsymbol{x})}$. The softmax output may be interpreted as the vector of class-wise probabilities living in the simplex Δ_{c-1}, and hence we may denote it by $p(\boldsymbol{y}|\boldsymbol{x})$. Next, we measure the discrepancy between label $\boldsymbol{y} = \boldsymbol{e}^i$ and network's output by a loss function $\ell\colon \mathcal{Y} \times [0,1]^c \to \mathbb{R}$, given for example by the *cross-entropy*:

$$\ell(\boldsymbol{e}^i, \boldsymbol{h}(\boldsymbol{x})) = -(\boldsymbol{e}^i)^\top \log\ p(\boldsymbol{y}|\boldsymbol{x}) = -\log\ p(\boldsymbol{y}|\boldsymbol{x})_i \ . \tag{6.1}$$

With some abuse of notation, we also define the loss in a vector form computed on every possible label:

$$\boldsymbol{\ell}(\boldsymbol{h}(\boldsymbol{x})) = \Big(\ell(\boldsymbol{e}^1, \boldsymbol{h}(\boldsymbol{x})), \ldots, \ell(\boldsymbol{e}^c, \boldsymbol{h}(\boldsymbol{x}))\Big)^\top \in \mathbb{R}^c \ . \tag{6.2}$$

Cross-entropy is essentially a multi-class version of logistic loss when we encode the labels for each class in $\{0,1\}$ instead of $\{\pm 1\}$. In this Chapter, formal results hold under very mild conditions on generic loss functions. We do not assume to work with linear-odd losses, unless specified as in Theorem 64. Also, notice that here a loss ℓ with a single argument *is not* a margin loss (Definition 2), but it is a notational shortcut to denote losses in vector form in the multi-class setting.

We update the setting of asymmetric label noise of Chapter 5 to multi-class. Each label $\boldsymbol{y}$ in the training set is flipped to $\tilde{\boldsymbol{y}} \in \mathcal{Y}$ with probability $p(\tilde{\boldsymbol{y}}|\boldsymbol{y})$. Denote by $T \in [0,1]^{c \times c}$ the noise transition matrix specifying the probability of one label being flipped to another, so that $\forall i, j \ \ T_{ij} = p(\tilde{\boldsymbol{y}} = \boldsymbol{e}^j | \boldsymbol{y} = \boldsymbol{e}^i)$. The matrix is row-stochastic and not necessarily symmetric across the classes. We aim to modify a loss ℓ so as to

[1]Here, $d^{(0)} = d$, the original feature dimensionality, and $d^{(n)} = c$, the label dimensionality.

make it robust to label noise; in fact, this is possible if T is known.

6.3 Loss correction procedures

6.3.1 The backward correction

We can build an *unbiased estimator* of the loss function in the same sense of Theorem 52. The corrected loss *under expected label noise* equals the original one computed on clean data. This property is stated in the next Theorem, a multi-class generalization of the already cited Lemma 1 Natarajan et al. [2013]. The Theorem is also a particular instance of the more abstract [van Rooyen, 2015, Theorem 3.2].

Theorem 60. *Suppose that the noise matrix T is non-singular. Given a loss $\boldsymbol{\ell}$, define the* backward *corrected loss as:*

$$\boldsymbol{\ell}^{\leftarrow}(\boldsymbol{h}(\boldsymbol{x})) = T^{-1}\boldsymbol{\ell}(\boldsymbol{h}(\boldsymbol{x})) \ . \tag{6.3}$$

Then, the loss correction is unbiased, i.e., *its expectation under label noise is exactly the loss:*

$$(\forall \boldsymbol{y} = \boldsymbol{e}^i) \quad \mathbb{E}_{p(\tilde{\boldsymbol{y}}|\boldsymbol{y})} \, \boldsymbol{\ell}^{\leftarrow}(\boldsymbol{h}(\boldsymbol{x}))_i = \boldsymbol{\ell}(\boldsymbol{h}(\boldsymbol{x}))_i \ , \tag{6.4}$$

and therefore the minimizers are the same:

$$\operatorname*{argmin}_{\boldsymbol{h}} \mathbb{E}_{\tilde{D}} \ \ell^{\leftarrow}(\boldsymbol{y}, \boldsymbol{h}(\boldsymbol{x})) = \operatorname*{argmin}_{\boldsymbol{h}} \mathbb{E}_{D} \ \ell(\boldsymbol{y}, \boldsymbol{h}(\boldsymbol{x})) \ . \tag{6.5}$$

Proof in 6.7.1. The corrected loss is effectively a linear combination of the loss values for each observable label, which coefficients are due to the probability that T^{-1} attributes to each possible true label $\boldsymbol{y}$, given the observed one $\tilde{\boldsymbol{y}}$. Intuitively, we are "going one step back" in the noise process described by the Markov chain T. The corrected loss is differentiable — although not always non-negative — and can be minimized with any off-the-shelf algorithm for back-propagation. Although in practice T would be invertible almost surely, its condition number may be problematic. A simple solution is to mix T with the identity matrix before inversion; this may be seen as taking a more conservative noise-free prior.

6.3.2 The forward correction

Alternatively, we can correct the model predictions. Following Sukhbaatar et al. [2015], we start by observing that a neural network learned with no loss correction would result in a predictor for noisy labels $p(\tilde{\boldsymbol{y}}|\boldsymbol{x})$. We can make explicit the depen-

dency on T. For instance, with cross-entropy we have:

$$\ell(\boldsymbol{e}^i, \boldsymbol{h}(\boldsymbol{x})) = -\log\ p(\tilde{\boldsymbol{y}} = \boldsymbol{e}^i|\boldsymbol{x}) \tag{6.6}$$

$$= -\log \sum_{j\in[c]} p(\tilde{\boldsymbol{y}} = \boldsymbol{e}^i|\boldsymbol{y} = \boldsymbol{e}^j)\ p(\boldsymbol{y} = \boldsymbol{e}^j|\boldsymbol{x}) \tag{6.7}$$

$$= -\log \sum_{j\in[c]} T_{ji}\ p(\boldsymbol{y} = \boldsymbol{e}^j|\boldsymbol{x})\ , \tag{6.8}$$

or in matrix form $\boldsymbol{\ell}(\boldsymbol{h}(\boldsymbol{x})) = -\log\ T^\top p(\boldsymbol{y}|\boldsymbol{x})$. This loss compares the noisy label $\tilde{\boldsymbol{y}}$ to averaged noisy prediction corrupted by T. We call this procedure "forward" correction.

In order to analyze its behavior, we first need to recall definition and properties of a new family of losses, named *proper composite* [Reid and Williamson, 2010, Section 4]. This is an additional requirement with respect to properness of Definition 1[2]. Many losses are said to be *composite*, in the sense that they can be expressed by the aid of an *link function*.

Definition 61. *A loss $\ell_{\boldsymbol{\psi}}$ is* composite *with* link function $\boldsymbol{\psi} : \Delta_{c-1} \to \mathbb{R}^c$*, invertible, if it can be written as:*

$$\ell_{\boldsymbol{\psi}}(\boldsymbol{y}, \boldsymbol{h}(\boldsymbol{x})) = \boldsymbol{\ell}(\boldsymbol{y}, \boldsymbol{\psi}^{-1}(\boldsymbol{h}(\boldsymbol{x})))\ . \tag{6.9}$$

Cross-entropy and square are examples of proper composite losses. In the case of cross-entropy, the softmax is the *inverse* link function. When composite losses are also *proper*, their minimizer assumes the particular shape of the link function applied to the class probability:

$$\operatorname*{argmin}_{\boldsymbol{h}} \mathbb{E}_{\mathcal{D}}\ \ell_{\boldsymbol{\psi}}(\boldsymbol{y}, \boldsymbol{h}(\boldsymbol{x})) = \boldsymbol{\psi}(p(\boldsymbol{y}|\boldsymbol{x}))\ . \tag{6.10}$$

An intriguing robustness property holds for forward correction of proper composite losses.

Theorem 62. *Suppose that the noise matrix T is non-singular. Given a proper composite loss $\boldsymbol{\ell}_{\boldsymbol{\psi}}$, define the* forward *loss correction as:*

$$\boldsymbol{\ell}^{\rightarrow}_{\boldsymbol{\psi}}(\boldsymbol{h}(\boldsymbol{x})) = \boldsymbol{\ell}(T^\top \boldsymbol{\psi}^{-1}(\boldsymbol{h}(\boldsymbol{x})))\ . \tag{6.11}$$

Then, the minimizer of the corrected loss under the noisy distribution is the same as the minimizer of the original loss under the clean distribution:

$$\boldsymbol{\psi}(p(\boldsymbol{y}|\boldsymbol{x})) = \operatorname*{argmin}_{\boldsymbol{h}}\ \ \mathbb{E}_{\tilde{\mathcal{D}}}\ \ell^{\rightarrow}_{\boldsymbol{\psi}}(\boldsymbol{y}, \boldsymbol{h}(\boldsymbol{x})) \tag{6.12}$$

$$= \operatorname*{argmin}_{\boldsymbol{h}}\ \ \mathbb{E}_{D}\ \ell_{\boldsymbol{\psi}}(\boldsymbol{y}, \boldsymbol{h}(\boldsymbol{x}))\ . \tag{6.13}$$

[2]Symmetric proper losses of Chapter 4 are also proper composite, with link function equal to the derivative of the generator ϕ. See Nock and Nielsen [2009] and Reid and Williamson [2010] for details.

Algorithm 10: Robust two-stage training

Input: the noisy training set $\widetilde{\mathcal{S}}$, any loss ℓ
If T is unknown:
  Train a network $\boldsymbol{h}(\boldsymbol{x})$ on $\widetilde{\mathcal{S}}$ with loss ℓ
  Obtain an unlabeled sample $\mathcal{S}_X$
  Estimate $\hat{T}$ by Equations (6.15)-(6.16) on $\mathcal{S}_X$
Train the network $\boldsymbol{h}(\boldsymbol{x})$ on $\widetilde{\mathcal{S}}$ with loss $\ell^{\leftarrow}$ or $\ell^{\rightarrow}$
Output: $\boldsymbol{h}(\cdot)$

Proof in 6.7.2. The result expresses a weaker property with respect to unbiasedness of Theorem 60. Robustness applies to the minimizer only: the model learned by forward correction is the minimizer over the *clean* distribution. Yet, Theorem 62 guarantees noise independence without explicitly inverting the noise process, but it does it "behind the scenes" by a "de-noising" link function. This turns out to be an important factor in practice, as shown in Section 6.5 experimentally and discussed in Section 6.6.

6.3.3 Estimating the noise rates

A clear limitation of the above procedures is that they require knowing T. In most applications, the matrix T would be unknown and needs to be estimated. We present here an extension of the noise estimator of Menon et al. [2015]; Liu and Tao [2016] to the multi-class settings. The estimator is derived under two assumptions.

Theorem 63. *Assume $p(\boldsymbol{x}, \boldsymbol{y})$ is such that:*

(i) there exist "perfect examples" of each of class $j \in [c]$, in the sense that:

$$(\exists \bar{\boldsymbol{x}}^j \in \mathcal{X}): \quad p(\bar{\boldsymbol{x}}^j) > 0 \wedge p(\boldsymbol{y} = \boldsymbol{e}^j | \bar{\boldsymbol{x}}^j) = 1.$$

(ii) given sufficiently many corrupted samples, $\boldsymbol{h}$ is rich enough to model $p(\tilde{\boldsymbol{y}}|\boldsymbol{x})$ accurately.

It follows that:

$$\forall i, j \in [c], \;\; T_{ij} = p(\tilde{\boldsymbol{y}} = \boldsymbol{e}^j | \bar{\boldsymbol{x}}^i) \; . \tag{6.14}$$

Proof in 6.7.3. Rather surprisingly, Theorem 63 tells us that we can estimate each component of matrix T *just based on noisy class probability estimates*, that is, the output of the softmax of a network trained with noisy labels. In particular, let $\mathcal{S}_X$ be any set of features vectors. This can be taken from $\mathcal{S}$ itself, but not necessarily: we do not require this sample to have *any* label at all and therefore whenever more unlabeled samples are easy to obtain from the same distributions; they could be used in place

loss	correction	$\mathbb{E}_{\tilde{\mathcal{D}}}$	Hessian of $\mathbb{E}_{\tilde{\mathcal{D}}}$
ℓ	-	no guarantee	unchanged
$\ell^{\leftarrow}$	$T^{-1}\cdot$	unbiased estimator of ℓ	unchanged
$\ell^{\rightarrow}$	$T\cdot$	same minimizer of ℓ	no guarantee

Table 6.1: Qualitative comparison of loss corrections.

of $\mathcal{S}$. We can approximate T with two steps:

$$\tilde{\boldsymbol{x}}^i = \operatorname*{argmax}_{\boldsymbol{x}\in\mathcal{S}_X} \; p(\tilde{\boldsymbol{y}} = \boldsymbol{e}^i|\boldsymbol{x}) \tag{6.15}$$

$$\hat{T}_{ij} = p(\tilde{\boldsymbol{y}} = \boldsymbol{e}^j|\tilde{\boldsymbol{x}}^i) \; . \tag{6.16}$$

In practice, assumption (1) of Theorem 63 might hold true when $\mathcal{S}_X$ is large enough. Assumption (2) is more difficult to justify: we require the network to perfectly model the probability of *the noisy labels*. Although, in the experiments we can often recover T close to the ground truth and find that small estimation errors have a mild, not catastrophic effect on the quality of the correction.

Algorithm 10 summarizes the end-to-end approach. If we know T, for example by cleaning manually a subset of training data, we can train with $\ell^{\leftarrow}$ or $\ell^{\rightarrow}$. Otherwise, we first have to train the network with ℓ on noisy data, and obtain from it estimates of $p(\tilde{\boldsymbol{y}}|\boldsymbol{x})$ for each class via the output of the softmax. After training $\hat{T}$ is computable in $O(c^2 \cdot |\mathcal{S}_X|)$. Finally, we re-train with the corrected loss, while potentially utilizing the first network to help initializing the second one.

6.4 Noise free Hessians via ReLU

We now present a result of independent interest in the context of label noise. The ReLU activation function appears to be a good fit for an architecture in our noise model, since is brings the particular convenience that the Hessian of the loss *does not depend on noise*, and hence the local curvature is left unchanged. At the same time, we are assured that backward correction by T — or any arbitrarily bad estimator of the matrix — has no impact on those second order properties of the loss — something that does not hold for the forward correction though. We stress the fact that other activation functions like the sigmoid do not share this guarantee. The proof in 6.7.4 makes use of the Factorization Theorem 18.

Theorem 64. *Assume ℓ is* LOL *and all network activation functions are ReLUs. Then, the Hessian of ℓ does not change under noise. Moreover, the Hessians of $\ell^{\leftarrow}$ and ℓ are the same for any T.*

Theorem 64 does not provide any assurance on minima: indeed, stationary points may change location due to label noise. What it *does* guarantee is that the convergence rate of first-order methods is the same: the loss curvature cannot blow up or flat out and instead it is the same point by point in the model space. The Theorem advocates

for use of ReLU networks, in line with the recent theoretical breakthrough allowing for deep learning with no local minima [Kawaguchi, 2016]. Table 6.1 summarizes the properties of loss correction.

6.5 Experiments

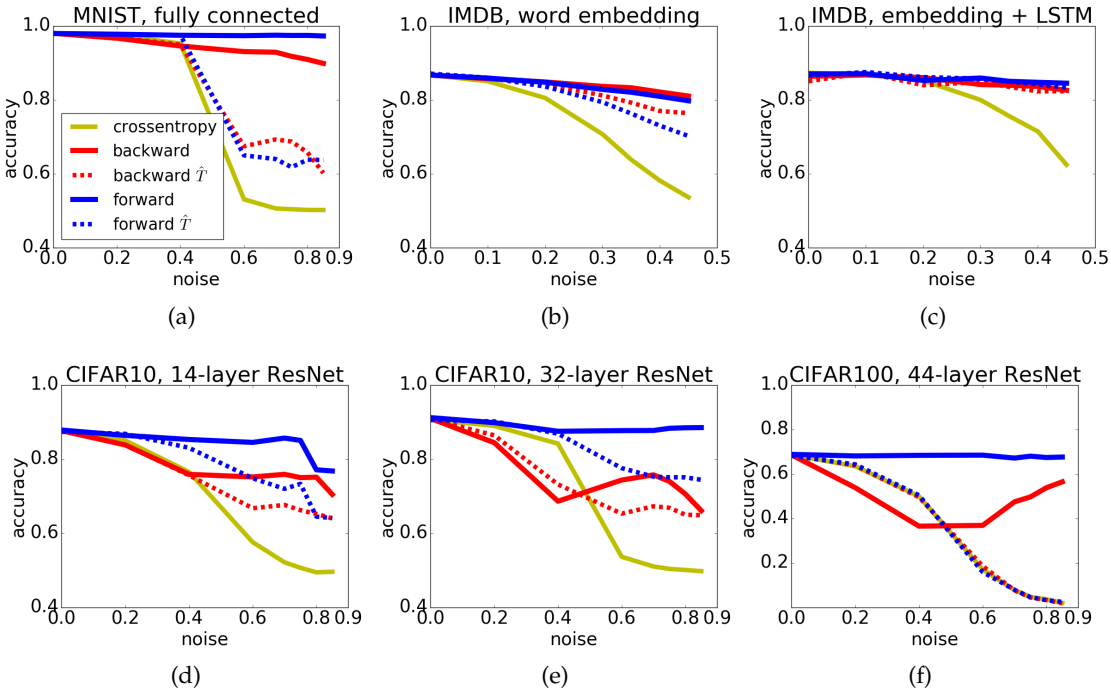


Figure 6.1: Comparison of cross-entropy with its corrections, with known or estimated T.

We now test the theory on various deep neural networks trained on MNIST [LeCun et al., 1998], IMDB [Maas et al., 2011], CIFAR-10, CIFAR-100 [Krizhevsky and Hinton, 2009] and Clothing1M [Xiao et al., 2015] so as to stress that our approach is independent on both architecture and data domain.

6.5.1 Loss corrections with T known or estimated

We artificially corrupt labels by a parametric matrix T. The rational is to mimic some of the structure of real mistakes for similar classes, *e.g.* CAT $\rightarrow$ DOG. Transitions are parameterized by $N \in [0, 1]$ such that ground truth and wrong class have probability respectively of $1 - N, N$. An example of T used for MNIST with $N = 0.7$ is on the

left:

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & .3 & 0 & 0 & 0 & 0 & .7 & 0 & 0 \\ 0 & 0 & 0 & .3 & 0 & 0 & 0 & 0 & .7 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & .3 & .7 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & .7 & .3 & 0 & 0 & 0 \\ 0 & .7 & 0 & 0 & 0 & 0 & 0 & .3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \begin{bmatrix} 1 & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & 1 & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & .33 & \epsilon & \epsilon & \epsilon & \epsilon & .67 & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & .35 & \epsilon & \epsilon & \epsilon & \epsilon & .65 & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & 1 & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & .29 & .71 & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & .73 & .26 & \epsilon & \epsilon & \epsilon \\ \epsilon & .75 & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & .25 & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & 1 & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & 1 \end{bmatrix} \tag{6.17}$$

Common to all experiments is what follows. The loss ℓ chosen for comparison is cross-entropy. 10% of training data is hold out for validation. *The loss* is evaluated on it during training: with the corrected losses we can validated *on noisy data*, which is advantageous over other approaches that measure noisy validation accuracy instead. The available standard test sets are used for testing. We use ReLUs for all networks and initialize weights prior to ReLUs as in He et al. [2015], otherwise by uniform sampling in $[-0.05, 0.05]$. The mini-batch size is 128. The estimator of T from noisy labels is applied to $\mathcal{S}_X$ being training and validation sets together; in fact, preliminary experiments highlighted that a $\mathcal{S}_X$ of large size sensibly improves the approximation of T; after estimation, we row-normalize the matrix. Following Menon et al. [2015], we take a α-percentile in place of the argmax of Equation 6.15, and we find $\alpha = 97\%$ to work well for most experiments. Although, such estimator performs very poorly with CIFAR100, possibly due the small number of images per class, and we find it is better to run the argmax instead.

Fully connected network on MNIST. In the first set of experiments we consider MNIST. Pixels are normalized in $[0, 1]$. Noise flips some of the similar digits: $1 \to 7, 2 \to 7, 3 \to 8, 5 \leftrightarrow 6$; see Equation (6.17, left). We train an architecture with two dense hidden layers of size 128, with probability 0.5 of dropout. AdaGrad [Duchi et al., 2011] is run for 40 epochs with initial learning rate 0.01 and $\delta = 10^{-6}$. We repeat each experiment 5 times to account for noise and weight initialization. It is clear from Figure 6.1a that, although the model is somewhat robust to mild noise, high level of corruption has a disrupting effect on ℓ. Instead, our losses do not witness a drastic drop. With $\hat{T}$ estimated performance lays in between, yet it is significantly better than with no correction. An example of $\hat{T}$ is in Equation (6.17, right), with $\epsilon < 10^{-6}$.

Word embedding and LSTM on IMDB. We keep only the top 5000 most frequent words in the corpus. Each review is either truncated or padded to 400-word long. To simulate asymmetric noise in this binary problem, we keep constant noise for the transition $0 \to 1$ at 5%, while $1 \to 0$ is parameterized as above; 0/1 are the two review's sentiments. We trained two models inspired by the baselines of Dai and Le [2015]. The first maps words into 50-dimensional embedding, before passing through ReLUs; dropout with probability 0.8 is applied to the embedding output as in Yarin and Ghahramani [2016]. In the second model the embedding has dimension 256 and it is followed by an LSTM with 512 units and by a last 512-dimensional hidden layer with 0.5 dropout. AdaGrad is run for 50 epochs with the same setup as above; results are averages over 5 runs. Figures 6.1b-6.1c display an outcome similar to what previously observed on MNIST, in spite of difference in dataset, number of classes,

architecture and structure of T. Noticeably, our approach is effective on recurrent networks as well. Correcting with $\hat{T}$ is in line with the true T here; we believe this is because estimation is easier on this binary problem.

Residual networks on CIFAR-10 and CIFAR-100. For both datasets we perform per-pixel mean subtraction and data augmentation as in He et al. [2016a], by horizontal random flips and 32×32 random crops after padding with 4 pixels on each side. T for CIFAR-10 is described by: TRUCK $\rightarrow$ AUTOMOBILE, BIRD $\rightarrow$ AIRPLANE, DEER $\rightarrow$ HORSE, CAT $\leftrightarrow$ DOG. In CIFAR-100, the 100 classes are grouped into 20 5-size super-classes, *e.g.* AQUATIC mammals contain BEAVER, DOLPHIN, OTTER, SEAL and WHALE. Within super-classes, the noise flips each class into the next, circularly.

For this last set of experiments we use deep residual networks (ResNet), the CIFAR10/100 architectures from He et al. [2016a]. In short, residual blocks implements a non-linear operation $F(\boldsymbol{x})$ in parallel with an identity shortcut, so as to sum the input with the output of the same block: $\boldsymbol{x} \rightarrow \boldsymbol{x} + F(\boldsymbol{x})$. Conceptually, this allows gradients to propagate more freely and helps convergence in deep architectures. F is implemented as cascade of twice batch normalization $\rightarrow$ ReLU $\rightarrow 3 \times 3$ convolution, following the "pre-activation" recommendation of He et al. [2016b]. Striding is used instead of max pooling; average pooling over channels is placed just before the output layer. We point to He et al. [2016a] for a detailed description. Here we experiment with ResNets of depth 14 and 32 (CIFAR10) and 44 (CIFAR100). By common practice [Huang et al., 2016], we run SGD with 0.9 momentum and learning rate 0.01, and divide it by 10 after 40 and 80 epoch (120 in total) for CIFAR-10 and after 80 and 120 (150 in total) for CIFAR-100; weight decay is set to 10^{-4}. Training deep ResNets is more time consuming and therefore experiments are run only once. Since we use shallower networks than the ones in He et al. [2016a], performance is not necessarily comparable with the original work. Figures 6.1d-6.1e-6.1f forward correction does not show any significant drop at all. Except with the shallowest ResNet, backward correction does not work so well in the low noise regime and it is possibly affected by high variance. Finally, notice how the noise estimation seems particularly difficult on CIFAR100.

6.5.2 Comparing with other loss functions

We now compare with other methods. Data, architectures and artificial noise are the same as above. Additionally, we test the case of symmetric noise: N is the probability of label flip that is spread uniformly among all the other classes. We select methods prescribing changes in the loss function, similarly to ours: unhinged [van Rooyen et al., 2015], sigmoid [Ghosh et al., 2015], Savage [Masnadi-Shirazi and Vasconcelos, 2009] and soft and hard bootstrapping [Reed et al., 2015]; hyper-parameters of the last two methods are set in accordance with their paper.

Unhinged loss is unbounded and cannot be used alone. In the original work L_2 regularization is applied to address the problem, when training non-parametric kernel models. We tried to regularize every layer with little success: learning either does not converge (too little regularization) or converge to very poor solutions (too

much). On preliminary experiments sigmoid loss ran into the opposite issue, namely premature saturation: the loss reaches a plateau too quickly, a well-known problem with sigmoidal activation functions [Glorot and Bengio, 2010]. To make those losses usable for comparison, we stack a layer of batch normalization right before the loss function. Essentially, the network output is normalized around 0 and thus operates in a bounded and non-saturated area of the loss; note that this is never required for linear or kernel models.

Table 6.3 presents the empirical analysis. We list the key findings.

(a) In the absence of artificial noise (first column for each dataset), all losses reach similar accuracy with a spread of 2 points; exceptions are some instances of unhinged and sigmoid, and Savage on CIFAR100 that makes learning impossible. Additionally, with IMDB there are cases († in Table 6.3) of loss correction with noise estimation that perform slightly better than assuming no noise: clearly, the estimator is able to recover the *natural* noise in the sentiment reviews.

(b) With low asymmetric noise (second column) results differ between simple architecture/tasks (datasets on the left) and deep networks/more difficult problems (right): in the former case, the two corrections behave similarly and are not statistically far from the competitors; in the latter case, forward correction with known T is unbeaten, with no clear winner among the remaining ones.

(c) With asymmetric noise (last two columns) the two loss corrections with known T are overall the best performing, confirming the practical implications of their formal guarantees; forward is usually the best.

(d) If we exclude CIFAR100, the noise estimation accounts for average accuracy drops between 0 (IMBD with LSTM model) and 27 points (MNIST); nevertheless, our performance is better than any other methods in many occasions.

(e) In the experiment on CIFAR100 we obtain essentially perfect noise robustness with the ideal forward correction. The noise estimation works well except in the very last column, yet it guarantees again better accuracy over competing methods. We discuss this issue in Section 6.6.

6.5.3 Experiments on Clothing1M

Finally, we test on Clothing1M [Xiao et al., 2015], consisting of 1M images with noisy labels, with additional $50k, 14k, 10k$ of clean data respectively for training, validation and testing; we refer to those sets by their size. We aim to classify images within 14 classes, *e.g.* t-shirt, suit, vest. In the original work two AlexNets [Krizhevsky et al., 2012] are trained together via EM; the networks are pre-trained with ImageNet. Two practical tricks are fundamental: a first learning phase with the clean $50k$ to help EM (#1 in Table 6.2) and a second phase with the mix of $50k$ bootstrapped to $500k$ and $1M$ (#3). Data augmentation is also applied, same as in Section 6.5.1 for CIFAR10.

We learn a 50-layer ResNet pre-trained on ImageNet — the bottleneck architecture of He et al. [2016a] — with SGD with learning rate 10^{-3} and 10^{-4} for 5 epochs each. and 0.9 momentum. Batch size is 32. When we train with $50k$ we use weight decay $5 \cdot 10^{-2}$ and data augmentation, while with $1M$ we use only weight decay of 10^{-3}.

Clothing1M					
#	model	loss	init	training	accuracy
1	AlexNet	cross-.	ImageNet	$50k$	72.63
2	AlexNet [Sukhbaatar et al., 2015]	cross-.	#1	$1M, 50k$	76.22
3	AlexNet [Xiao et al., 2015]	cross-.	#1	$1M, 50k$	78.24
4	50-ResNet	cross-	ImageNet	$1M$	68.94
5	50-ResNet	backward	ImageNet	$1M$	69.13
6	50-ResNet	forward	ImageNet	$1M$	69.84
7	50-ResNet	cross-.	ImageNet	$50k$	75.19
8	50-ResNet	cross-.	#6	$50k$	**80.38**

Table 6.2: Results on the top section are from Xiao et al. [2015]. In #2, #3 the clean $50k$ are bootstrapped to $500k$. Best result #8 is obtained by fine tuning a net trained with forward correction.

The ResNet gives an uplift of about 2.5 points by training with $50k$ only (#7 vs. #1). However, the large amount of noisy images is essential to compete with #3. Instead of estimating the matrix T by (6.15)-(6.16), we exploit the curated labels of $50k$ and their noisy versions in $1M$. Forward and backward corrections confirm to work better than cross-entropy (#6, #5 vs. #4), yet cannot reach the state of the art without the additional clean data. Thus, we fine-tune the networks with $50k$, with the same learning parameters as in #7; because of time constraints we only fine-tune #6. The new state of the art is #8 that outperforms Xiao et al. [2015] of more than 2 points, which is achieved without time consuming bootstrapping of the $50k$.

6.6 Discussion

We have proposed two methods for training deep neural networks with noisy labels that boils down to two loss corrections based on modeling the noise by a row-stochastic matrix T. Test accuracy is consistently only a few percents away from training cross-entropy *on clean data*, while corruption often worsen performance of cross-entropy by 40 points or more. Forward correction often outperforms the backward one. The explicit inversion of T — sometimes with high condition number — may be the root of the problem; indeed, backward correction is a linear combination of loss values for each possible label, and their coefficients may be far by orders of magnitude, which intuitively makes optimization hard. Instead, forward correction projects model predictions into another distribution in the probability simplex.

The quality of noise estimation is evidently the key factor for successfully obtaining robustness. Estimation works fairly well in most experiments with a median drop of only 10 points of accuracy with respect to using the true T. The only exception is the very last column of tests on CIFAR100, where estimation destroys most of the gain from correcting the loss. We believe that the combination of high noise and limited number of images per class (500) is detrimental to the proposed estimator. This is confirmed by the sensitivity of the hyper-parameter α. In fact, the same value

used in other experiments led to $\hat{T}$ with no resemblance to the ground truth, leading the learning process to very poor solutions.

We attempt two explanations. First, the sample size may be too small for estimating T with 100 classes — it is 10 times smaller per class than on CIFAR10. Second, we assumption (2) of Theorem 63 may be the bottleneck: cross-entropy achieves less than 70 accuracy on clean data versus $\sim$ 90 with CIFAR10; this is an indication that the model may well be misspecified for CIFAR100 and hence the estimator of $p(\tilde{\boldsymbol{y}}|\boldsymbol{x})$ may be not good enough.

Future work shall improve the estimation phase by incorporating knowledge of the noise structure, for example assuming low rank T. Improvements on this direction may also enlarge the applicability of our approach to multi-class with thousands of classes. It remains an open question whether more realistic *instance*-dependent noise may be included in our approach [Xiao et al., 2015; Menon et al., 2016]. Finally, we anticipate the application of our approach as a seamless tool for pre-training models with noisy data from the Web, in the spirit of Krause et al. [2016].

	MNIST, fully connected				CIFAR10, 14-layer ResNet			
	NO NOISE	SYMM. $N = 0.2$	ASYMM. $N = 0.2$	ASYMM. $N = 0.6$	NO NOISE	SYMM. $N = 0.2$	ASYMM. $N = 0.2$	ASYMM. $N = 0.6$
cross-entropy	97.9 ± 0.0	$\mathbf{96.9 \pm 0.1}$	97.5 ± 0.0	53.0 ± 0.6	87.8	83.7	85.0	57.6
unhinged (BN)	97.6 ± 0.0	$\mathbf{96.9 \pm 0.1}$	97.0 ± 0.1	$\mathbf{71.2 \pm 1.0}$	86.9	**84.1**	83.8	52.1
sigmoid (BN)	97.2 ± 0.1	93.1 ± 0.2	96.7 ± 0.1	$\mathbf{71.4 \pm 1.3}$	76.0	66.6	71.8	57.0
Savage	97.3 ± 0.0	$\mathbf{96.9 \pm 0.0}$	97.0 ± 0.1	51.3 ± 0.4	80.1	77.4	76.0	50.5
bootstrap soft	97.9 ± 0.0	$\mathbf{96.9 \pm 0.0}$	97.5 ± 0.0	53.0 ± 0.4	87.7	**84.3**	84.6	57.8
bootstrap hard	97.9 ± 0.0	96.8 ± 0.0	97.4 ± 0.0	55.0 ± 1.3	87.3	83.6	84.7	58.3
backward	97.9 ± 0.0	96.3 ± 0.1	96.6 ± 1.1	$\mathbf{93.0 \pm 0.9}^\star$	87.6	81.5	83.8	$\mathbf{75.2}^\star$
backward $\hat{T}$	97.9 ± 0.0	$\mathbf{96.9 \pm 0.0}$	96.7 ± 0.1	67.4 ± 1.5	87.7	80.4	83.8	66.7
forward	97.9 ± 0.0	$\mathbf{97.3 \pm 0.0}^\star$	$\mathbf{97.7 \pm 0.0}$	$\mathbf{97.3 \pm 0.0}^\star$	87.8	$\mathbf{85.6}^\star$	86.3	$\mathbf{84.5}^\star$
forward $\hat{T}$	97.9 ± 0.0	$\mathbf{96.9 \pm 0.0}$	$\mathbf{97.7 \pm 0.0}$	64.9 ± 4.4	87.4	83.4	**87.0**	**74.8**
	IMBD, word embedding				CIFAR10, 32-layer ResNet			
	NO NOISE	SYMM. $N = 0.1$	ASYMM. $N = 0.1$	ASYMM. $N = 0.4$	NO NOISE	SYMM. $N = 0.2$	ASYMM. $N = 0.2$	ASYMM. $N = 0.6$
cross-entropy	86.7 ± 0.0	84.6 ± 0.1	85.0 ± 0.2	58.1 ± 0.5	90.1	86.6	89.0	53.6
unhinged (BN)	83.3 ± 0.0	76.9 ± 0.5	80.6 ± 0.3	72.9 ± 0.4	90.2	86.5	87.1	60.0
sigmoid (BN)	84.3 ± 0.0	80.2 ± 0.3	81.7 ± 0.5	72.8 ± 0.6	81.6	69.6	79.1	61.8
Savage	86.5 ± 0.0	84.3 ± 0.4	85.2 ± 0.3	58.3 ± 1.0	88.3	86.2	86.3	53.5
bootstrap soft	86.7 ± 0.0	84.5 ± 0.1	85.1 ± 0.1	57.8 ± 0.7	90.9	86.9	88.6	53.1
bootstrap hard	86.7 ± 0.0	84.6 ± 0.3	85.1 ± 0.3	59.0 ± 0.6	90.4	86.4	88.6	54.7
backward	86.7 ± 0.0	$\mathbf{85.3 \pm 0.3}^\star$	$\mathbf{85.7 \pm 0.1}$	$\mathbf{82.1 \pm 0.1}^\star$	90.1	83.0	84.4	74.3
backward $\hat{T}$	$87.0 \pm 0.0^\dagger$	$\mathbf{85.1 \pm 0.4}$	$\mathbf{85.8 \pm 0.2}$	$\mathbf{77.0 \pm 1.4}$	90.8	86.9	86.4	66.7
forward	86.7 ± 0.0	$\mathbf{85.3 \pm 0.2}^\star$	$\mathbf{85.9 \pm 0.1}$	$\mathbf{80.9 \pm 1.3}^\star$	91.2	**87.7**	**89.9**	$\mathbf{87.6}^\star$
forward $\hat{T}$	$87.0 \pm 0.0^\dagger$	$\mathbf{85.2 \pm 0.3}$	$\mathbf{85.9 \pm 0.2}$	73.0 ± 1.2	90.5	**87.9**	**90.1**	**77.6**
	IMBD, word embedding + LSTM				CIFAR100, 44-layer ResNet			
	NO NOISE	SYMM. $N = 0.1$	ASYMM. $N = 0.1$	ASYMM. $N = 0.4$	NO NOISE	SYMM. $N = 0.2$	ASYMM. $N = 0.2$	ASYMM. $N = 0.6$
cross-entropy	87.8 ± 0.4	$\mathbf{85.2 \pm 0.5}$	86.8 ± 0.4	71.4 ± 1.3	68.5	57.9	63.5	17.1
unhinged (BN)	84.3 ± 4.4	69.7 ± 15.9	85.2 ± 1.2	59.4 ± 12.9	50.9	47.5	48.0	14.5
sigmoid (BN)	87.7 ± 0.5	77.6 ± 13.6	$\mathbf{86.3 \pm 3.1}$	$\mathbf{70.0 \pm 14.6}$	58.2	47.6	55.6	16.4
Savage	87.4 ± 0.3	$\mathbf{85.1 \pm 0.6}$	$\mathbf{87.2 \pm 0.3}$	70.4 ± 3.8	1.4	2.0	1.8	1.6
bootstrap soft	87.1 ± 0.6	$\mathbf{83.5 \pm 2.5}$	$\mathbf{86.1 \pm 1.2}$	69.0 ± 5.3	67.9	57.8	**63.8**	16.3
bootstrap hard	86.5 ± 0.5	$\mathbf{84.3 \pm 1.0}$	86.7 ± 0.4	71.8 ± 3.3	68.5	57.3	**63.9**	17.0
backward	87.6 ± 0.2	$\mathbf{84.3 \pm 0.9}$	86.7 ± 0.5	$\mathbf{83.6 \pm 1.4}$	68.5	55.1	53.8	$\mathbf{36.8}^\star$
backward $\hat{T}$	87.2 ± 0.7	$\mathbf{82.8 \pm 2.7}$	$\mathbf{87.3 \pm 0.1}$	$\mathbf{82.3 \pm 1.7}$	68.6	51.7	63.8	**18.5**
forward	87.5 ± 0.2	$\mathbf{85.0 \pm 0.2}$	87.0 ± 0.4	$\mathbf{84.7 \pm 0.6}^\star$	68.8	$\mathbf{64.0}^\star$	$\mathbf{68.1}^\star$	$\mathbf{68.4}^\star$
forward $\hat{T}$	$87.8 \pm 1.5^\dagger$	$\mathbf{84.1 \pm 1.0}$	$\mathbf{87.5 \pm 0.2}$	$\mathbf{84.2 \pm 0.9}$	68.1	**58.6**	**64.2**	15.9

Table 6.3: Mean accuracy (over 5 runs) are bold faced when statistically different from the others, by means of passing a *Welch's t-test* with p-value $< 5\%$; in case the highest accuracy is due to $\ell^{\leftarrow}$ or $\ell^{\rightarrow}$ with the ground truth T, we denote those by * and highlight the next highest accuracy as well. For experiments without any variance information, the same rule is applied but bold face is given to the all accuracies in a range of 0.5 points from the highest. Notice that the meaning of N depends on symmetric vs. asymmetric noise and on number of classes (see Section 6.5.1). On the first columns with no injected noise, † indicates when the noise estimation recovers some *natural* noise and beats "loss correction" with $T = I$.

6.7 Appendix: proofs

6.7.1 Proof of Theorem 60

Simply:

$$\mathbb{E}_{p(\tilde{\boldsymbol{y}}|\cdot)}\, \boldsymbol{\ell}^{\leftarrow}(\boldsymbol{h}(\boldsymbol{x})) = T\, \boldsymbol{\ell}^{\leftarrow}(\boldsymbol{h}(\boldsymbol{x})) = T\, T^{-1}\boldsymbol{\ell}(\boldsymbol{h}(\boldsymbol{x})) = \boldsymbol{\ell}(\boldsymbol{h}(\boldsymbol{x}))\ . \tag{6.18}$$

The second statement follows from $\ell(\boldsymbol{e}^i, \boldsymbol{h}(\boldsymbol{x})) = (\boldsymbol{e}^i)^\top \boldsymbol{\ell}(\boldsymbol{h}(\boldsymbol{x}))$. Therefore, the minimizers are the same.

6.7.2 Proof of Theorem 62

First notice that:

$$\ell^{\rightarrow}_{\boldsymbol{\psi}}(\boldsymbol{y}, \boldsymbol{h}(\boldsymbol{x})) = \ell(\boldsymbol{y}, T^\top \boldsymbol{\psi}^{-1}(\boldsymbol{h}(\boldsymbol{x}))) = \ell_{\boldsymbol{\phi}}(\boldsymbol{y}, \boldsymbol{h}(\boldsymbol{x})) \tag{6.19}$$

where we denote $\boldsymbol{\phi}^{-1} = \boldsymbol{\psi}^{-1} \circ T^\top$, or equivalently $\boldsymbol{\phi} = (T^{-1})^\top \circ \boldsymbol{\psi}$ by rule of inverse of composition. $\boldsymbol{\phi}$ is invertible by composition of invertible functions, its domain is $[0, 1]$ as of $\boldsymbol{\psi}$ and its codomain is $\mathbb{R}$ because of composition of T^{-1} with $\boldsymbol{\psi}$. The last loss in Equation 6.19 is therefore proper composite with link $\boldsymbol{\phi}$. Finally, from Equation 6.10, the loss minimizer over the noisy distribution is:

$$\operatorname*{argmin}_{\boldsymbol{h}} \mathbb{E}_{\tilde{\mathcal{D}}}\, \ell_{\boldsymbol{\phi}}(\boldsymbol{y}, \boldsymbol{h}(\boldsymbol{x})) = \boldsymbol{\phi}(p(\tilde{\boldsymbol{y}}|\boldsymbol{x})) \tag{6.20}$$

$$= \boldsymbol{\psi}((T^{-1})^\top p(\tilde{\boldsymbol{y}}|\boldsymbol{x})) = \boldsymbol{\psi}(p(\boldsymbol{y}|\boldsymbol{x}))\ , \tag{6.21}$$

that proves the Theorem.

6.7.3 Proof of Theorem 63

For any $j \in [c]$ and any $\boldsymbol{x} \in \mathcal{X}$, we have:

$$p(\tilde{\boldsymbol{y}} = \boldsymbol{e}^j|\boldsymbol{x}) = \sum_{k\in[c]} p(\tilde{\boldsymbol{y}} = \boldsymbol{e}^j|\boldsymbol{y} = \boldsymbol{e}^k)\ p(\boldsymbol{y} = \boldsymbol{e}^k|\boldsymbol{x}) \tag{6.22}$$

$$= \sum_{k\in[c]} T_{kj}\ p(\boldsymbol{y} = \boldsymbol{e}^k|\boldsymbol{x})\ . \tag{6.23}$$

When $\boldsymbol{x} = \bar{\boldsymbol{x}}^i$, we have $p(\boldsymbol{y} = \boldsymbol{e}^k|\bar{\boldsymbol{x}}^i) = 0$ for $k \neq i$.

6.7.4 Proof of Theorem 64

We give the proof for cross-entropy for simplicity. One can use the Factorization Theorem 18 to generalize the result to all LOLS. When $\boldsymbol{y} = \boldsymbol{e}^i$ the loss is:

$$\ell(\boldsymbol{e}^i, \boldsymbol{h}(\boldsymbol{x})) = -\log\ p(\boldsymbol{y}|\boldsymbol{x})_i \tag{6.24}$$

$$= -\log\ \frac{e^{W_{i\cdot}^{(n)}\boldsymbol{x}^{(n-1)}+\boldsymbol{b}_i^{(n)}}}{\sum_{k=1}^{c} e^{W_{k\cdot}^{(n)}\boldsymbol{x}^{(n-1)}+\boldsymbol{b}_k^{(n)}}} \tag{6.25}$$

$$= -W_{i\cdot}^{(n)}\boldsymbol{x}^{(n-1)} + \boldsymbol{b}_i^{(n)} + \log\ \sum_{k=1}^{c} e^{W_{k\cdot}^{(n)}\boldsymbol{x}^{(n-1)}+\boldsymbol{b}_k^{(n)}}\ . \tag{6.26}$$

The only dependence on the true class $\boldsymbol{e}^i$ above are the first two terms. The log-partition is *independent* of the precise class i. Evidently, the noise affects the loss only through $W_{\cdot i}^{(n)}$ and $\boldsymbol{b}_i^{(n)}$: those are the *only* terms in which $\ell(\boldsymbol{y}, \boldsymbol{h}(\boldsymbol{x}))$ and $\ell(\tilde{\boldsymbol{y}}, \boldsymbol{h}(\boldsymbol{x}))$ may differ. Therefore we can rewrite the backward corrected loss as:

$$\ell^{\leftarrow}(\boldsymbol{e}^j, \boldsymbol{h}(\boldsymbol{x})) = \left(T^{-1}\boldsymbol{\ell}(\boldsymbol{h}(\boldsymbol{x}))\right)_j \tag{6.27}$$

$$= -\left(T^{-1}W^{(n)}\right)_{j\cdot}\boldsymbol{x}^{(n-1)} - \left(T^{-1}\boldsymbol{b}^{(n)}\right)_j + \log\ \sum_{k=1}^{c} e^{W_{k\cdot}^{(n)}\boldsymbol{x}^{(n-1)}+\boldsymbol{b}_k^{(n)}}\ . \tag{6.28}$$

In fact, note that T^{-1} does not affect the log-partition function. To see this, let $A(\boldsymbol{x}) = \log\ (\sum_{k=1}^{c} e^{W_{k\cdot}^{(n)}\boldsymbol{x}^{(n-1)}+\boldsymbol{b}_k^{(n)}})$, with the (vector) log-partition being $A(\boldsymbol{x})\mathbf{1}$. It follows that its correction is $T^{-1}A(\boldsymbol{x})\mathbf{1} = A(\boldsymbol{x})\mathbf{1}$, by left-multiplication of T and because $T\mathbf{1} = \mathbf{1}$ since T is row-stochastic. Thus $\ell^{\leftarrow}(\boldsymbol{e}^j, \boldsymbol{h}\,(\boldsymbol{x})) = B(\boldsymbol{x}) + A(\boldsymbol{x})$, where $B(\boldsymbol{x}) = -(T^{-1}W^{(n)})_{j\cdot}\boldsymbol{x}^{(n-1)} - (T^{-1}\boldsymbol{b}^{(n)})_j$ is a *piece-wise linear* function of the *model parameters*, and the log-partition $A(\boldsymbol{x})$ is non-linear because of the loss and the architecture but *does not depend on noise*. Since the composition of piece-wise linear function is piece-wise linear, the Hessian of $B(\boldsymbol{x})$ vanishes, and therefore the Hessian of $\ell^{\leftarrow}$ is noise independent for any T. The same holds for ℓ (no correction) by taking $T = I$ and hence the Hessians are the same.

6.8 References

We review relevant literature on Deep Learning. Several works have attempted to deal with noisy labels of late, especially in Computer Vision. This is often achieved by formulating noise-aware models. Mnih and Hinton [2012] build a noise model for binary classification of aerial image patches, which can handle omission and wrong location of training labels. Xiao et al. [2015] construct a more sophisticated mix of symmetric, asymmetric and instance-dependent noise; two networks are learned by EM as model for classifier and noise type. Reed et al. [2015] augment the objective

similarly to entropy regularization. In practice, it is often the case that a small set of clean labels is needed in order either to pre-train or fine-tune the model [Xiao et al., 2015; Krause et al., 2016; Reed et al., 2015].

The work of Sukhbaatar et al. [2015] deserves a particular mention. The method augments the architecture by adding a linear layer on top of the network. Once learned, this layer plays the role of our matrix T. The insight is that, at training time, the effect of the noise is captured by the linear layer, emulating the corruption of the model predictions; at test time, the linear layer must be removed to obtain clean predictions. However, learning this architecture appears problematic; heuristics are necessary, such as trace regularization and a fixed updating schedule for the linear layer. We sidestep those issues by decoupling the two phases: we first estimate T and then learn with loss correction.

Recent work has provided methods to estimate label flip probabilities directly from noisy samples. (Note that the matrix is fully specified by 2 noise rates in binary classification.) If one has access to clean samples, estimation is reliable Kearns [1998]. With only noisy samples, one can obtain *bounds* on these rates [Laird, 1988, Algorithm 5.8], one can obtain T under some assumptions on the generating distribution [Scott et al., 2013; Sanderson and C. Scott, 2014; Liu and Tao, 2016; Menon et al., 2015; Ramaswamy et al., 2016]. Typically, it is required that the generating distribution is "weakly separable": that is, such that for each class, there exists some "perfect" instance, *i.e.* one that is classified with probability equal to one. Proposed estimators involve either the use of kernel mean embedding [Ramaswamy et al., 2016], or post-processing the output of a standard class-probability estimator such as logistic regression using order statistics on the range of scores [Liu and Tao, 2016; Menon et al., 2015] or the slope of the induced ROC curve [Sanderson and C. Scott, 2014]. A common limitation is the focus on the case of binary labels, with the exception of Sanderson and C. Scott [2014].

We are not aware of any other attempt at either applying the noise-corrected loss approach of Natarajan et al. [2013] to neural networks, nor on combining those losses with the above noise rate estimators. Our work sits precisely in this intersection. Note that, even though in principle loss correction should not be necessary for high-capacity models like deep neural networks, owing to aforementioned theoretical results, in practice such correction may offset the sub-optimality of these models arising from training on finite samples.

Chapter 7

Learning from vertically distributed data without entity matching

This last Chapter deals with a rather particular learning problem. Our goal is to learn classifiers combining features from two different spaces. To a first approximation, we can consider the data as vertically (feature-wise) partitioned, hence distributed. But the scenario is actually more challenging. We do not even know how to match features vectors from one dataset to the other; their identity has been lost, for instance due to anonymization. Therefore, to some extent, the weak supervision derives from the fact that we do not exactly know how to match "parts" of features vectors, while we do assume to know all the relative labels. Once again, we design a solution within the two-step framework by first recovering the missing information via estimation of sufficient statistics. This approach allow us to bypass a more traditional solution, combinatorial in nature, that resorts to entity matching. While conceptually similar to the rest of the Thesis, the peculiar constraints for this learning problem push for a different workaround manipulating loss functions. We require the notion of Rademacher observations, aggregated statistics from which we learn instead of examples. This way, entity resolution can be bypassed to carry out supervised learning almost as accurate as if its solution were known.

7.1 Motivation

Consider the following data fusion scenario: two datasets/peers contain the same real-world entities described using partially shared features, for example banking and insurance company records of the same customer base. Our goal is to learn a classifier in the cross product space of the two domains, in the hard case in which no shared ID is available – *e.g.* due to anonymization.

A main motivation towards this objective comes from the reported experience that combining features from different sources leads to better predictive power. For instance, insurance and banking data together can improve fraud detection; shopping

	Peer 1				Peer 2		
		shared				shared	
	x_1	x_3	c		x_2	x_3	c
e_1	1	1	1	e'_1	-1	1	1
e_2	-1	1	1	e'_2	1	1	1

Table 7.1: A simple case of vertical partitioning with $p = 2$ peers, two shared variables x_3 and c (the class to predict). This toy example has binary description features and a binary shared feature, but this restriction does not need to hold in the general case. For example, each shared feature can be any categorical/ordinal feature, like "post-code", "age-band", etc.

records well complement medical history for estimating risk of disease [Tsui et al., 2003]; joining heterogeneous data helps prediction in genomics [Lanckriet et al., 2004; Yamanishi et al., 2004]; security agencies integrate various sources for terrorism intelligence [Sweeney, 2005; Christen, 2006; Sproull et al., 2015]

Typical data fusion methods however rely on a known map between entities [Bleiholder and Naumann, 2008], *i.e.* peers have partially different views of the *same* examples. Instead, we assume the datasets do not share a common ID, as shown in Figure 7.2; that is, for example, the case when data collection is performed independently by each peer, or when sources are deliberately anonymized. Thus, we can think the data as vertically partitioned (**VP**).

Learning from massively distributed data collections and multiple information sources has become a pivotal problem, yet it faces critical challenges, among which is the fact that it relies on reconstructing consistent examples from diverse features distributed between different data handling *peers*. Exhaustive search to solve this problem is simply not scalable, nor communication efficient, and sometimes not even accurate [Estrada et al., 2010; Zhang et al., 2015].

7.1.1 Entity resolution

Entity resolution (ER), or entity matching [Christen, 2012], would be the traditional approach for reconciling entities with no shared ID. It approximates a JOIN operation, assuming that some of the attributes are shared, such as *age-band*, *gender*, *post-code* (etc.), and hence can be used as "weak IDs". Most techniques for ER are based on similarity functions and thresholding: candidate entities are selected as matches when their similarity is above a threshold. Both components can be tuned on some ground truth matches and effectively enhanced with learning techniques [Bilenko and Mooney, 2003; Christen, 2012]. Performance metrics of ER encompass lots of different parameters, including generality, accuracy, soundness, scalability, parallelizability [Rastogi et al., 2011].

The standard pipeline for learning with ER is depicted in Figure 7.1 (left): (1) entities are matched based on similarity and heuristics, (2) they are merged in one unique database and (3) a model is learned on the joint data. Here, we deliberately

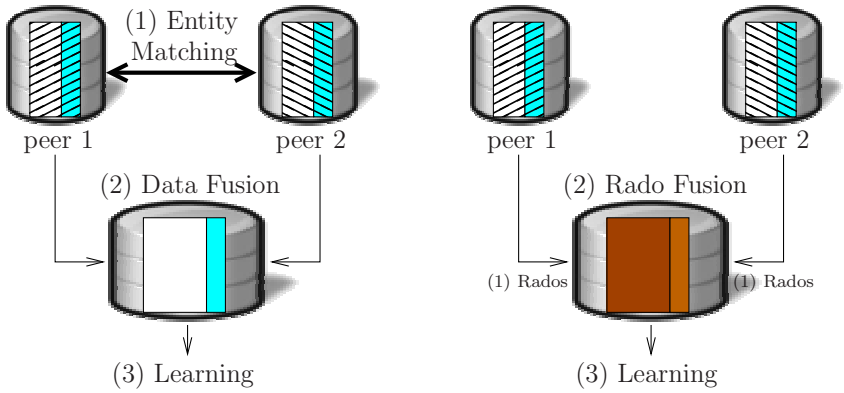


Figure 7.1: Learning on top of ER (left) or with rados (right).

do not consider common issues, such as *conflicts* and *heterogeneity* [Bleiholder and Naumann, 2008].

From a high level view, ER integrates data as a pre-process for other tasks. When it comes to learning from ER'ed data, small changes in ER can have large impact on evaluating classifiers, even for simple classifiers as linear models. To see this, suppose we are in the toy example of Table 7.1. Here, all shared variables have the same values, so entity matching has two potential solutions; notice that one of the shared variable is class c. One, say ER1, is matching e_1 with e'_1 and e_2 with e'_2. We denote the examples obtained by

$$e_{11} = ((1,-1,1),1), \;\; e_{22} = ((-1,1,1),1) \; .$$

The other solution, say ER2, is matching e_1 with e'_2 and e_2 with e'_1. We denote the examples obtained by

$$e_{12} = ((1,1,1),1)), \;\; e_{21} = ((-1,-1,1),1) \; .$$

Consider linear classifier $\boldsymbol{\theta} = (1,1,1) \in \mathbb{R}^3$; the predicted class is given by the sign of its inner product with an observation, $\text{sign}\langle \boldsymbol{\theta}, \boldsymbol{z} \rangle$. While $\boldsymbol{\theta}$ classifies perfectly on $\{e_{11}, e_{22}\}$ (zero error), it classifies no better than random on $\{e_{12}, e_{21}\}$ (error 50%). The simple demonstration shows how the approach of entity matching is not only computationally expensive but potentially detrimental to the final learning performance. We advocate for an entirely different solution addressing both issues.

7.2 Learning setting

In this Chapter we work again on the setting of binary classification with linear models $\boldsymbol{\theta}$. We refer to $\mathcal{S} \doteq \{(\boldsymbol{x}_i, y_i), i \in [m]\}$ as the *total* learning sample. We have p (sub)samples, $\mathcal{S}^j$ of size m_j, $j \in [p]$ for some $p > 1$. Each one is defined in its own

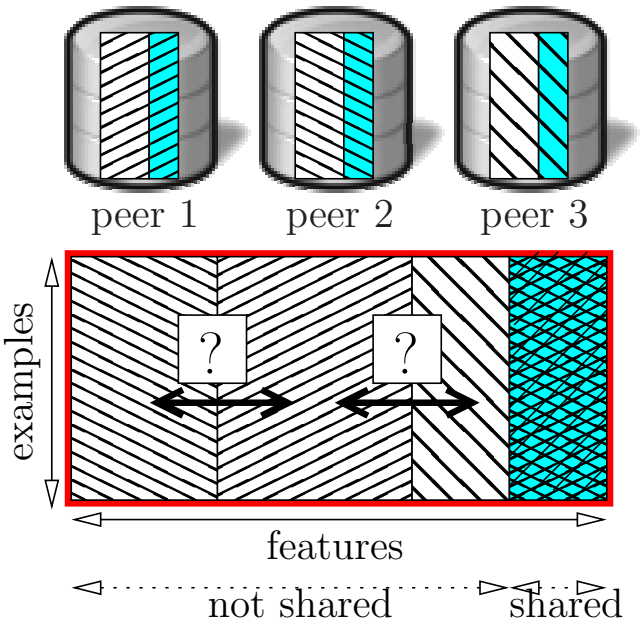


Figure 7.2: Schematic view of our setting, with $p = 3$ peers (best viewed in color). Some features (cyan) are described in each peer and one these shared features is a class. Non-shared features are split among peers. A so-called *total* sample $\mathcal{S}$ is figured by the red rectangle. In the case of vertical partition (**VP**) all peers see different views of the same examples, but do not know who is who among their datasets ("?"). Hence, each bit of the total sample is seen by one peer.

feature space

$$\mathcal{X}^j \doteq \times_{k=1}^{d_j} \mathcal{X}_{j_k}, j_k \in [d], \forall k \ . \tag{7.1}$$

To get a simple case of this framework, shown in Figure 7.2, one may see each $\mathcal{S}^j \doteq \{(\boldsymbol{x}_i^j, y_i^j), i \in [m_j]\}$ handled by a *peer* P^j. We denote this setting by the work "distributed" not necessarily meaning that the data is split on different computing machines or locations — although this is also possible —, but to stress the fact that the partial views of the dataset cannot be simply joined together. To avoid confusion, we reserve the word *entity* for a generic record in a dataset, the object of matching, and *attributes* or *features* to its fields. In the following, subscripts i will refer to an example or entity, while superscripts j to a peer.

We rely on the following assumption:

(**A**7) The class $\mathcal{Y}$ and a subset of features $\mathcal{J} \subseteq \{\mathcal{X}_k\}_{k=1}^d$ are shared by all peers. Each other feature is exclusive to one peer.

Hence, there exists $\dim(\mathcal{J}) + 1$ columns that represent the same set of attributes among peers, and one of them is the class. Each of the dimensions of $\mathcal{J}$ is in all $\mathcal{X}^j$s. This is a realistic assumption for the features in $\mathcal{J}$: in the (**VP**) setting the domain is

vertically partitioned for the non-shared features, implying $m_j = m_{j'} = m, \forall j, j' \in [p]$. In this case, there exists an (unknown) one-to-one mapping between the peers' rows. The shared labels may be harder to justify, since they are the attribute we aim to predict. However, we argue in Section 7.6 that if at least one peer has classes than all peers can get their labels as well *without entity resolution*, by methods for LLP from Chapter 4.

7.3 Rademacher observations

In order to elaborate our solution to the learning problem of this Chapter, we need to recall elements of the theory on *Rademacher observations* (rados). The notion is introduced in Nock et al. [2015], which we have co-authored. In the standard binary classification setting, a Rademacher observation is a simple transformation of the examples in $\mathcal{S}$.

Definition 65. *Let $\boldsymbol{\sigma} \in \Sigma_m = \{-1, 1\}^m$. A* Rademacher observation, *or* rado, *is a vector in $\mathbb{R}^d$ defined as:*

$$\boldsymbol{\pi}_{\boldsymbol{\sigma}} \doteq \sum_{i=1}^{m} 1\{y_i = \sigma_i\} \; y_i \boldsymbol{x}_i \; . \tag{7.2}$$

We also term any element $y\boldsymbol{x}$ as an *edge* vector. Rados are effectively sums of edge vectors restricted to subsample of $\mathcal{S}$. From $\mathcal{S}$, we can obtain 2^m rados. For any $\Sigma'_m \subseteq \Sigma_m \doteq \{-1, 1\}^m$, we let $\mathcal{R}_{\mathcal{S},\Sigma'_m} \doteq \{\boldsymbol{\pi}_{\boldsymbol{\sigma}} : \boldsymbol{\pi}_{\boldsymbol{\sigma}} \in \Sigma'_m\}$ which denotes the set of rados that can be crafted from Σ'_m using $\mathcal{S}$. One rado in $\mathcal{R}_{\mathcal{S},\Sigma_m}$ is a non-normalized version of the mean operator $\boldsymbol{\pi} = \sum_{i=1}^{m} y_i \cdot \boldsymbol{x}_i$, with $\boldsymbol{\sigma} = \boldsymbol{y}$. A comment on notation in this Chapter: for maintaining coherence with published work, we use the greek letter $\boldsymbol{\pi}$ for rados and the letter $\mathcal{R}$ for sets of rados. Since there is no mention of neither label proportions (Chapter 4) nor Rademacher complexity, this choice should not be a source of confusion.

One reason behind Rademacher observations is the existence of multiple equivalences between loss functions computed on examples and loss functions computed on rados. The idea was originally introduced by mapping the standard logistic loss on examples to a particular exponential loss on rados [Nock et al., 2015, Lemma 2].

Theorem 66. *The following holds true for any $\boldsymbol{\theta}$ and $\mathcal{S}$:*

$$\mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{S}} \log\left(1 + e^{-y\langle\boldsymbol{\theta},\boldsymbol{x}\rangle}\right) = \log(2) + \frac{1}{m}\log\left(\mathbb{E}_{\pi\sim\mathcal{R}_{\mathcal{S},\Sigma_m}} \; e^{-\langle\boldsymbol{\theta},\boldsymbol{\pi}\rangle}\right) \tag{7.3}$$

By virtue of the Theorem, in order to compute and hence optimize the right-hand side of 7.3, we don't need to know any individual examples. Instead, we can train a model on rados. Importantly, the minimizers of the two sides of the Equation are the same. Therefore, no post-processing of the model is required. This should remind of the Factorization Theorem 18.

There is an caveat though: the equivalence strictly holds true only for the whole, exponentially-sized set $\mathcal{R}_{\mathcal{S},\Sigma_m}$, undermining any direct application. Nevertheless, we can get interesting generalization bounds based on Equation 7.3 when we compute the right-hand side by uniform sampling of rados belonging in $\mathcal{R}_{\mathcal{S},\Sigma_m}$ and by limiting ourselves to a set of linear size [Nock et al., 2015, Theorem 3].

Instead of concentrating on the losses of Theorem 66, we formulate an equivalence based on square loss. Such another relationship between the two worlds, examples and rados, suggests the existence of a wide theory underpinning those aggregated statistics. For a more complete view on the topic, the interested reader can consult Nock [2015].

We consider the empirical risk of an L_2 regularized square loss $\ell(x) = (1-x)^2$. Here Γ is a symmetric positive definite matrix.

$$\mathbb{E}_{\mathcal{S}} \ (1 - y\langle \boldsymbol{\theta}, \boldsymbol{x} \rangle)^2 + \boldsymbol{\theta}^\top \Gamma \boldsymbol{\theta} \ . \tag{7.4}$$

The loss admits a simple closed form solution:

$$\boldsymbol{\theta}^\star \doteq \arg\min_{\boldsymbol{\theta}} \ \mathbb{E}_{\mathcal{S}} \ (1 - y\langle \boldsymbol{\theta}, \boldsymbol{x} \rangle)^2 + \boldsymbol{\theta}^\top \Gamma \boldsymbol{\theta} = \left(XX^\top + m \cdot \Gamma \right)^{-1} \boldsymbol{\pi} \ , \tag{7.5}$$

where $X \doteq [\boldsymbol{x}_1 | \boldsymbol{x}_2 | \cdots | \boldsymbol{x}_m]$, and so, $XX^\top = \sum_i \boldsymbol{x}_i \boldsymbol{x}_i^\top$. Remark that the computation of $\boldsymbol{\theta}^\star$ involves $\boldsymbol{\pi}$, the rado corresponding with the mean operator; labels do not appear anywhere else in the formula in light of label sufficiency. Let us define the following M-loss over rados via its empirical risk.

Definition 67. *The empirical risk associated with the M-loss over $\mathcal{R}_{\mathcal{S},\Sigma'_m}$ of classifier $\boldsymbol{\theta}$ is:*

$$\frac{1}{2} \mathbb{V}_{\Sigma'_m} \ \langle \boldsymbol{\theta}, \boldsymbol{\pi}_\sigma \rangle - \mathbb{E}_{\Sigma'_m} \ \langle \boldsymbol{\theta}, \boldsymbol{\pi}_\sigma \rangle \ , \tag{7.6}$$

where expectation and variance are computed with respect to uniform sampling of $\boldsymbol{\sigma}$ in Σ'_m.

Equation (7.6) resembles a Markowitz mean-variance criterion [Markowitz, 1952] — hence the loss name —, with no coefficient for the risk aversion. What this means is that a good classifier trained on rados should have large "return" and small "risk", where the risk is the variance of its predictions and the return is its inner product with the expected rado. M-loss and squared loss are linked via the next Theorem.

Theorem 68. *The following holds true for any $\boldsymbol{\theta}$ and $\mathcal{S}$:*

$$\mathbb{E}_{(x,y)\sim\mathcal{S}} \ (1 - y\langle \boldsymbol{\theta}, \boldsymbol{x} \rangle)^2 = 1 + \frac{2}{m} \mathbb{V}_{\sigma\sim\Sigma_m} \ \langle \boldsymbol{\theta}, \boldsymbol{\pi}_\sigma \rangle - \frac{4}{m} \mathbb{E}_{\sigma\sim\Sigma_m} \ \langle \boldsymbol{\theta}, \boldsymbol{\pi}_\sigma \rangle \ . \tag{7.7}$$

The Proof in 7.7.1 simplifies the long derivation in Nock [2015]. The above equivalence still holds if both risks are regularized by $\boldsymbol{\theta}^\top \Gamma \boldsymbol{\theta}$, for any symmetric positive semi-definite Γ. Hence, minimizing the L_2 regularized square loss over examples is equivalent to minimizing a regularized version of the M-loss, over the complete set of all rados.

Rados can also be thought as sufficient statistics, similarly to the mean operator.

Lemma 69. *The function $\mathcal{S} \mapsto \mathcal{R}_{\mathcal{S},\Sigma_m}$ that produces the whole set of rados $\mathcal{R}_{\mathcal{S},\Sigma_m}$ is a sufficient statistic for variables $(\boldsymbol{x}, y)$ with regard to the hypothesis space of linear classifiers $\mathcal{H}$ and square loss, in the sense that, for any two sample $\mathcal{S}$ and $\mathcal{S}'$ and for any $\boldsymbol{\theta} \in \mathcal{H}$ we have:*

$$\begin{aligned} &\mathbb{E}_{\mathcal{S}}(1 - y\langle \boldsymbol{\theta}, \boldsymbol{x}\rangle)^2 - \mathbb{E}_{\mathcal{S}'}(1 - y\langle \boldsymbol{\theta}, \boldsymbol{x}\rangle)^2 \ \textit{ does not depend on } \ (\boldsymbol{x}, y) \\ &\qquad \iff \mathcal{R}_{\mathcal{S},\Sigma_m} = \mathcal{R}_{\mathcal{S}',\Sigma_m} \ . \end{aligned} \tag{7.8}$$

Proof in 7.7.2. As we have noticed already, $\mathcal{R}_{\mathcal{S},\Sigma_m}$ contains the mean operator and hence sufficiency for the label variable is not a surprise. Although, the claim is about all examples in $\mathcal{S}$, not just the labels. Yet, the Lemma is in some sense trivial. In fact, the whole set of rados $\mathcal{R}_{\mathcal{S},\Sigma_m}$ contains also all "the sums" of single edge vectors $y_i \boldsymbol{x}_i$ — when there is only one i such that σ_i in the definition of rados is equal to y_i. The knowledge of all edge vectors would be sufficient for all $\mathcal{S}$ even without the use of rados, in case we use any margin loss with linear classifiers. Therefore, expressing the empirical risk with rados requires many more elements to achieve statistical sufficiency, *i.e.* the whole $\mathcal{R}_{\mathcal{S},\Sigma_m}$.

Therefore, why using rados instead of examples? The main reason is that rados are invariant to the selection of different solutions for entity resolution. For example, consider again Table 7.1. Since all classes are positive, computing a rado is just summing observations. Let $\boldsymbol{\pi}_{ij,kl}$ be the rado that sums those of examples e_{ij} and e_{kl}. Then, surprisingly, regardless of the solution to ER, this rado is the *same*:

$$\begin{aligned} (\text{E1}) \ \boldsymbol{\pi}_{11,22} &= (1, -1, 1) + (-1, 1, 1) && (7.9) \\ &= (0, 0, 2) && (7.10) \\ &= (1, 1, 1) + (-1, -1, 1) = && (7.11) \\ &= \boldsymbol{\pi}_{12,21} \ (\text{E2}) \ . && (7.12) \end{aligned}$$

This always holds in the (**VP**) setting: there a set of rado, potentially of exponential size, that matches the set that could be built *knowing* the true entity resolution. Next, we give the algorithm that builds these rados and it is communication efficient and easily parallelizable.

7.4 Building and learning from block rados

In our distributed setting, we extend the definition of rados in the following way. We let $\boldsymbol{s} \in \mathcal{J}$ denote a *signature*, a vector of shared attributes.

Definition 70. *The rado of signature $\boldsymbol{s}$ and peer P^j is:*

$$\boldsymbol{\pi}^j_{(\boldsymbol{s},y)} \doteq \sum_{i=1}^{m} 1\{\text{proj}_{\mathcal{J}}(\boldsymbol{x}^j_i) = \boldsymbol{s} \wedge y^j_i = y\} \ y^j_i \boldsymbol{x}^j_i \ , \tag{7.13}$$

where $\mathrm{proj}_{\mathcal{I}}(\boldsymbol{z})$ *denotes the restriction of a vector* $\boldsymbol{z}$ *to* $\mathcal{I}$.

In short, $\boldsymbol{\pi}^j_{(\boldsymbol{s},y)}$ sums edge vectors local to P^j whose examples match signature $\boldsymbol{s}$ and class y. Intuitively, we can conceptualize those rados and expressing statistics *locally* sufficient for the examples sharing the same signature $\boldsymbol{s}$ in the data of P^j. Let $\mathcal{F}(\boldsymbol{z}) \subseteq \mathcal{X}$ be the set of features of $\boldsymbol{z}$. We also define, for any $\mathcal{F}' \supseteq \mathcal{F}(\boldsymbol{z})$, $\mathrm{lift}_{\mathcal{F}'}(\boldsymbol{z})$ to be the vector $\boldsymbol{z}'$ described using $\mathcal{F}'$ such that $\mathrm{proj}_{\mathcal{F}(\boldsymbol{z})}(\boldsymbol{z}') = \boldsymbol{z}$ and $\mathrm{proj}_{\mathcal{F}'\setminus\mathcal{F}(\boldsymbol{z})}(\boldsymbol{z}') = \boldsymbol{0}$. While $\mathrm{proj}_{\mathcal{F}}(\boldsymbol{z})$ removes coordinates of $\boldsymbol{z}$, $\mathrm{lift}_{\mathcal{F}'}(\boldsymbol{z})$ "completes" the coordinates of $\boldsymbol{z}$ with zeroes.

By analogy with entity resolution [Whang et al., 2009], we define *block rados* as rados, lifted to $\mathcal{X}$, that are the (weighted) sums of examples matching a particular signature and class in all peers.

Definition 71. *For any* $\boldsymbol{s} \in \mathcal{J}$, $y \in \{-1, 1\}$, *let* $m_{(\boldsymbol{s},y)}$ *be the number of examples matching signature* $(\boldsymbol{s}, y)$. *Then a* basic block *(BB) rado for* $(\boldsymbol{s}, y)$ *is:*

$$\boldsymbol{\pi}_{(\boldsymbol{s},y)} \doteq \sum_{j=1}^{p} \mathrm{lift}_{\mathcal{X}}(\boldsymbol{\pi}^j_{(\boldsymbol{s},y)}) - m_{(\boldsymbol{s},y)}(p-1) \cdot \mathrm{lift}_{\mathcal{X}}(y \cdot \boldsymbol{s}) \ . \tag{7.14}$$

We need to subtract the second term to take into account that $\boldsymbol{s}$ has already been summed $m_{(\boldsymbol{s},y)}$ times by each peer. Let:

$$\mathcal{J}_* \doteq \{(\boldsymbol{s}, y) \in \mathcal{J} \times \{-1, 1\} : \exists j \in [p], \boldsymbol{\pi}^j_{(\boldsymbol{s},y)} \neq \boldsymbol{0}\} \ . \tag{7.15}$$

This latter set, which can easily be computed from all peers, has cardinal $m_* \doteq |\mathcal{J}_*| \leq m$, and even $m_* \ll m$ when few features are shared. We let:

$$\mathcal{R}_{\mathrm{B}} \doteq \{\boldsymbol{\pi}_{\boldsymbol{v}_i}, \forall i \in [m_*]\} \tag{7.16}$$

denote the ordered set of each BB rado, each coordinate of $\boldsymbol{v} = (\boldsymbol{s}, y)$ being in one-one correspondence with an element of $\mathcal{J}_*$. A superset of $\mathcal{R}_{\mathrm{B}}$ is interesting, that considers all sums of vectors from $\mathcal{R}_{\mathrm{B}}$:

$$\mathcal{R}_* \doteq \left\{ \sum_{i \in \mathcal{U}} \boldsymbol{\pi}_{\boldsymbol{v}_i}, \forall \mathcal{U} \subseteq [m_*] \right\} \ . \tag{7.17}$$

We call $\mathcal{R}_*$ the set of *block rados*. Notice that we may have $|\mathcal{R}_*| = \Omega(2^{\sum_j |\mathcal{S}^j|})$. It is therefore intractable in general to *explicitly* compute $\mathcal{R}_*$. However, $|\mathcal{R}_{\mathrm{B}}| = O(\sum_j |\mathcal{S}^j|)$ and to compute it, we just need the set of $\boldsymbol{\pi}^j_{(\boldsymbol{s},y)}$, hence a communication complexity that can be much smaller than $\sum_j |\mathcal{S}^j|$.

This set has exponential size. A possibility is to randomly subsample the set, along with proving good uniform convergence bounds for the M-loss — this can be done in the same way as in [Nock et al., 2015]. However, in the case of the square loss, greed pays twice: learning from all rados in $\mathcal{R}_*$ may be both computationally cheap and accurate.

Algorithm 11: RadoCraft($\mathsf{P}^1, \mathsf{P}^2, ..., \mathsf{P}^p$)

Input: Peers $\mathsf{P}^1, \mathsf{P}^2, ..., \mathsf{P}^p$
$\mathcal{R}_{\text{B}} \leftarrow \emptyset$
for $\boldsymbol{s} \in \mathcal{J}$, $y \in \{\pm 1\}$
  Let $\boldsymbol{\pi}_{(\boldsymbol{s},y)} \leftarrow \mathbf{0} \in \mathbb{R}^d$
  for $j \in [p]$
    $\boldsymbol{\pi}_{(\boldsymbol{s},y)} \leftarrow \boldsymbol{\pi}_{(\boldsymbol{s},y)} + \text{lift}_{\mathcal{X}}(\textsc{craft}(\boldsymbol{s}, y) \rightsquigarrow \mathsf{P}^j)$
  $\mathcal{R}_{\text{B}} \leftarrow \mathcal{R}_{\text{B}} \cup \{\boldsymbol{\pi}_{(\boldsymbol{s},y)}\}$
Output: $\mathcal{R}_{\text{B}}$

7.4.1 Computation and optimality of block rados

We do not have access to all rados because we do not assume to know the entity matching function. Yet, we are going to show a first result which is, in a sense, *stronger*: $\mathcal{R}_*$ always belongs to $\mathcal{R}_{\mathcal{S},\Sigma_m}$. Therefore $\mathcal{R}_*$ —potentially exponential-sized— gives us a set of rados that would have been built *from $\mathcal{S}$, had we known the perfect solution to entity matching*. So, even without carrying out entity matching, we have access to a potentially huge set of "ideal" rados which we can use to learn $\boldsymbol{\theta}$ via the minimization of the empirical risk on M-loss.

Theorem 72. *In setting (**VP**), for any $p \geq 2$, any $\mathcal{S}$, any $\mathcal{J}$. Let $\mathcal{R}_{\text{B}}$ be the output of Algorithm 11 and let $\mathcal{R}_*$ its superset by Equation (7.17). Then, $\mathcal{R}_* \subseteq \mathcal{R}_{\mathcal{S},\Sigma_m}$.*

Proof in 7.7.3. Furthermore, there exists a simple algorithm to build $\mathcal{R}_{\text{B}}$. Algorithm 11 summarizes the protocol. Each peer P^j crafts rados upon request of a particular signature and label; "$\textsc{craft}(\boldsymbol{s}, y) \rightsquigarrow$" symbolizes a message sent, expecting $\boldsymbol{\pi}^j_{(\boldsymbol{s},y)}$ in return. The computation of each rado for each peer can easily be performed in parallel. Algorithm 11 always provides the basis for the set $\mathcal{R}_*$ of the "ideal" rados.

7.4.2 Learning from all block rados

The questions that remain are how we minimize the regularized M-loss and, more importantly, what subset of rados from $\mathcal{R}_*$ we shall use. As already discussed, we choose "greediness" against randomization: instead of picking a (small) random subset of $\mathcal{R}_*$, we want to use them *all* because we know that all of them are "ideal" or close to being so via Theorems 72. Recall that $|\mathcal{R}_*|$ may be of exponential size (in m, d, $|\mathcal{J}_*|$, etc.). We now show that if we consider all of $\mathcal{R}_*$, the optimal model learned from rados has an analytic expression which depends *only* on the rados of $\mathcal{R}_{\text{B}}$. In short, it is even *faster* to compute than $\hat{\boldsymbol{\theta}}$ from $\mathcal{S}$ in Equation (7.5), and can be directly computed from the output of Algorithm 11.

Theorem 73. *Let $\hat{\boldsymbol{\theta}}$ be the minimizer of Equation (7.6). Then:*

$$\hat{\boldsymbol{\theta}} = \left(BB^\top + \dim_{\text{c}}(B) \cdot \Gamma\right)^{-1} B\mathbf{1} \ , \tag{7.18}$$

Algorithm 12: DRL($\mathsf{P}^1, \mathsf{P}^2, ..., \mathsf{P}^p; \Gamma$)

Input: Peers $\mathsf{P}^1, \mathsf{P}^2, ..., \mathsf{P}^p$, Γ, $\gamma > 0$
$B \leftarrow \text{Column}(\text{RadoCraft}(\mathsf{P}^1, \mathsf{P}^2, ..., \mathsf{P}^p))$
$\boldsymbol{\theta} \leftarrow \left(BB^\top + \gamma \cdot \Gamma\right)^{-1} B\mathbf{1}$
Output: $\boldsymbol{\theta}$

where B stacks in columns the rados of $\mathcal{R}_{\text{B}}$, and $\dim_{\text{c}}(B)$ is the number of columns of B.

Proof in 7.7.4. When $m_* = m$, each element of $\mathcal{R}_{\text{B}}$ is in fact an example, and we retrieve Equation (7.5). One consequence of Theorem 73 is the following convergence property which we state informally: in the (**VP**) setting, for any $\varepsilon \geq 0$, there exists a minimal size for $\mathcal{J}_*$ such that $\hat{\boldsymbol{\theta}}$ will be ε-close to $\boldsymbol{\theta}^\star$, where the closeness can be measured by $\|\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}^\star\|_2$ or $|\cos(\hat{\boldsymbol{\theta}}, \boldsymbol{\theta}^\star)|$. The statement of Distributed Rado-Learn (DRL) is in Algorithm 12. "column(.)" takes a set of vectors and put them in column in a matrix.

7.4.3 A more realistic setting

What happens if we drop the assumption of data being vertically partitioned (**VP**)? Or equivalently, what if examples are not shared by *all* peers ? This is a much more realistic scenario. Since there is no shared ID — and the data may have been anonymized — we are not even in a situation where we can guarantee that a specific client of the bank *is*, or *is not*, a client of the insurance company. Thus, there may be significant unknown data "to reconstruct" the total sample $\mathcal{S}$, but we do not know which specific examples have missing features.

In this most general setting (**G**), it is possible to show that a very simple transformation of the rados, involving only the shared features, has the same properties so far described and for which Theorem 72 holds *in expectation*. We leave the details to Appendix 7.8 to avoid the heavier notation. However, in the next Section, we provide an experimental validation of our approach in both the settings.

7.5 Experiments

We evaluate the leverage that DRL provides compared to the peers, that would learn using only their local dataset. Each peer P^j estimates learns through a 10-folds stratified cross-validation minimization of regularized squared loss (Equation 7.5), where γ is also locally optimized through a ten-folds CV in set $\{.01, 1.0, 100.0\}$. DRL solves Equation 7.18 where $\mathcal{R}_{\text{B}}$ is built using RadoCraft, with the set of all peers as input.

We have carried out a very simple optimization of the regularization matrix of DRL as a diagonal matrix which weights differently the shared features:

$$\Gamma = \text{diag}(\text{lift}_{\mathcal{X}}(\text{proj}_{\mathcal{J}}(\mathbf{1}))) + \gamma \cdot \text{diag}(\text{lift}_{\mathcal{X}}(\text{proj}_{\mathcal{X}\setminus\mathcal{J}}(\mathbf{1}))), \quad \text{for} \quad \gamma \in \mathcal{G} \ . \tag{7.19}$$

γ is optimized by a 10-folds CV on $\mathcal{J}_*$, defined in Equation 7.15. CV is performed on *rados* as follows: first, $\mathcal{R}_{\text{B}}$ is split in 10 folds, $\mathcal{R}_{\text{B},\ell}$, for $\ell = 1, 2, ..., 10$. Then, we repeat for $\ell = 1, 2, ..., 10$ (and then average) the following CV routine:

1. DRL is trained using $\mathcal{R}_{\text{B}} \backslash \mathcal{R}_{\text{B},\ell}$
2. DRL's solution is evaluated on "test rados" by computing the average M-loss

The expression of Γ for rados exploits the idea that the estimations related to a shared feature can be much more accurate than for another, non shared feature.

7.5.1 Domain generation

We ran experiments on a dozen UCI domains [Bache and Lichman, 2013]. Only two are fully detailed here, the rest are presented in Appendix 7.9. They are *mice* $(m \times d) \approx (1K \times 70)$ and *musk* $\approx (6.5K \times 166)$. For each domain, we have varied (i) the number of peers p, (ii) the number of shared features $\dim(\mathcal{J})$, and (iii) the number b of numeric modalities ("bins") each shared feature was reduced to (it controls the size of $\mathcal{J}_*$). The training sample is split among peers, each keeping record of $\mathcal{I}$ and its own features (non shared features are evenly partitioned among peers). Finally, for some $p_s \in [0, 1]$, each peer P^j selects a proportion p_s of its examples index and for each of them, another peer $\mathsf{P}^{j'}$, chosen at random, gets the example as well (on its own set of features $\mathcal{X}^{j'}$). This policy simulate scenario (**G**). When $p_s = 0$, this is setting (**VP**). We then run *all* algorithms for *each* value $p, \dim(\mathcal{J}), b, p_s$. As we shall see, b appears to have a relatively small influence compared to the other factors, so we mainly report results combining various values for $p, \dim(\mathcal{J})$ and p_s, for the range of values of $p, \dim(\mathcal{J})$ specified in Table 7.6, and for $p_s \in \{0.0, 0.2\}$. We have chosen $b = 4$ for all domains, except when it is not possible (if for example all features are boolean), in which case we pick $b = 2$.

7.5.2 Metric

We used two metrics. The first:

$$\Delta \doteq p_{\text{err}}(\text{DRL}) - \min_j p_{\text{err}}(\mathsf{P}^j) \ \in [-1, 1] \ , \tag{7.20}$$

is the test error for DRL minus that of the *optimal* peer *in hindsight* (since we consider the peer's test error). when $\Delta < 0$, DRL beats *all* peers. For example, Table 7.3 (left) provides the results obtained on UCI domain mice. We see that for almost all combinations of p and $\dim(\mathcal{J})$, DRL beats all peers.

The smallest test error obtained for a peer among all runs for each domain: this is an indication of the room of improvement for DRL, and it also shows that in general, at least some (and in fact most) peers were always very significantly better than random guessing, a safe-check that DRL is not just beating unbiased coins.

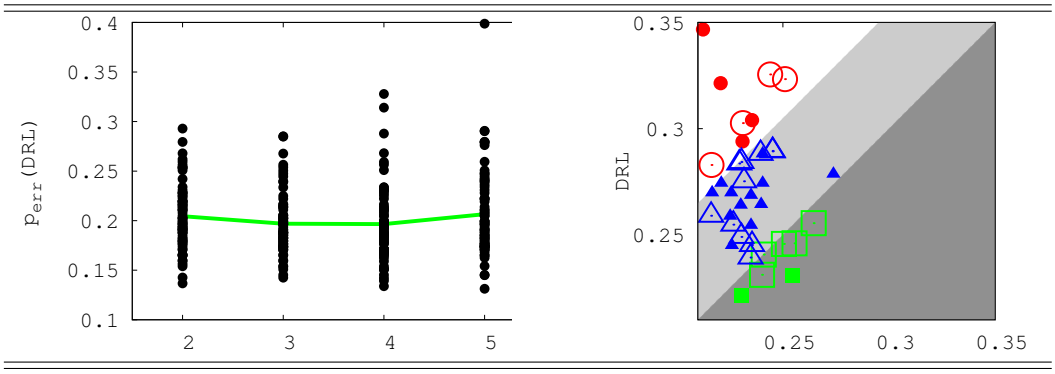


Table 7.2: *Left*: test error of DRL on domain ionosphere, as a function of the number of bins, aggregating all values of the number of peers p and number of shared features $\dim(\mathcal{J})$ used; the green line denotes the average values. *Right*: scatter plot of the test error of DRL (y) vs that of the Oracle (learning using the complete entity-resolved domain). Points in the dark grey area (green) denote better performances of DRL; points in the light grey area (blue) denote better performances of the Oracle (but not statistically better). Points in the white area (red) denote *statistically* better performances of the Oracle (filled points:$p_s = 0.2$; empty points:$p_s = 0.8$).

To evaluate the statistical significance, we compute:

$$\mathrm{q} \doteq \text{ proportion of peers } \textit{statistically} \text{ beaten by } DRL \ . \tag{7.21}$$

To compute the test, we use the powerful Benjamini-Hochberg procedure [Benjamini and Hochberg, 1995] on top of paired t-tests with $q^* = p\text{-val} = 0.05$. $q = 0.8$ surface helps see when DRL *statistically beats all peers*. For example, Table 7.3 (right) displays that DRL does not always *statistically* beat all peers when $\Delta < 0$, yet it manages to statistically beat all of them in approximately one third to one half of the total tests, which implies that, on this domain, there is a significant chance that DRL improves on the peers, regardless of their number and the number of shared features.

7.5.3 Results

All domains display that there exists regimes $(p, \dim(\mathcal{J}))$ for which DRL improves on all peers, in some cases significantly. Sometimes, the improvement is sparse (phishing, creditcard), but sometimes it is quite spectacular and in fact (almost) systematic (page, ionosphere, steelplates). Domain steelplate's case is interesting, since the so-called *Oracle*, *i.e.* the learner that learns from the complete training fold *before* it is split among peers — and therefore knows the solution to entity matching —, has for this domain almost optimal error, but local peers are in fact very far from this optimum. This indicates that many features, properly combined, are necessary to attain the best performances. DRL's performances are close to the Oracle, which accounts for the huge gap in classification compared to peers — sometimes, DRL's test error is smaller than that of the *best* peer by more than 20% —, and so it seems

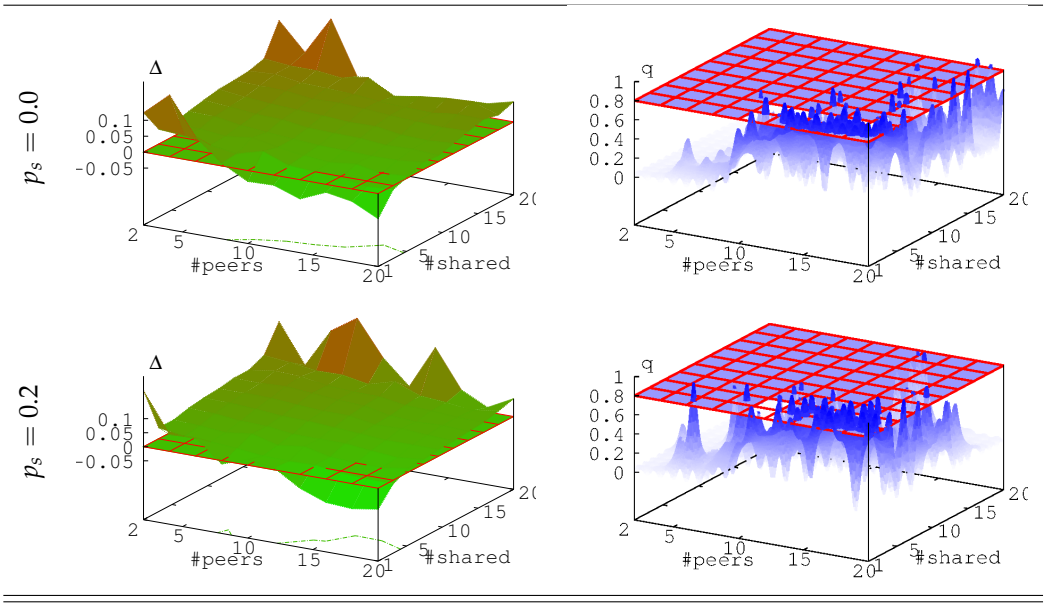


Table 7.3: Results on domain mice: fig/rado of $\Delta \doteq p_{\text{err}}(\text{DRL}) - \min_j p_{\text{err}}(\text{P}^j)$ (left) and $q =$ prop. peers *simultaneously* beaten by DRL (right) as a function of the number of peers p and the number of shared features $\dim(\mathcal{J})$. On mice, $\min_j p_{\text{err}}(\text{P}^j) = 0.30$. Top: proportion of shared examples $p_s = 0.0$ (**VP**); bottom: proportion of shared examples $p_s = 0.2$ (**G**). The isoline on the left fig/rado is $\Delta = 0$.

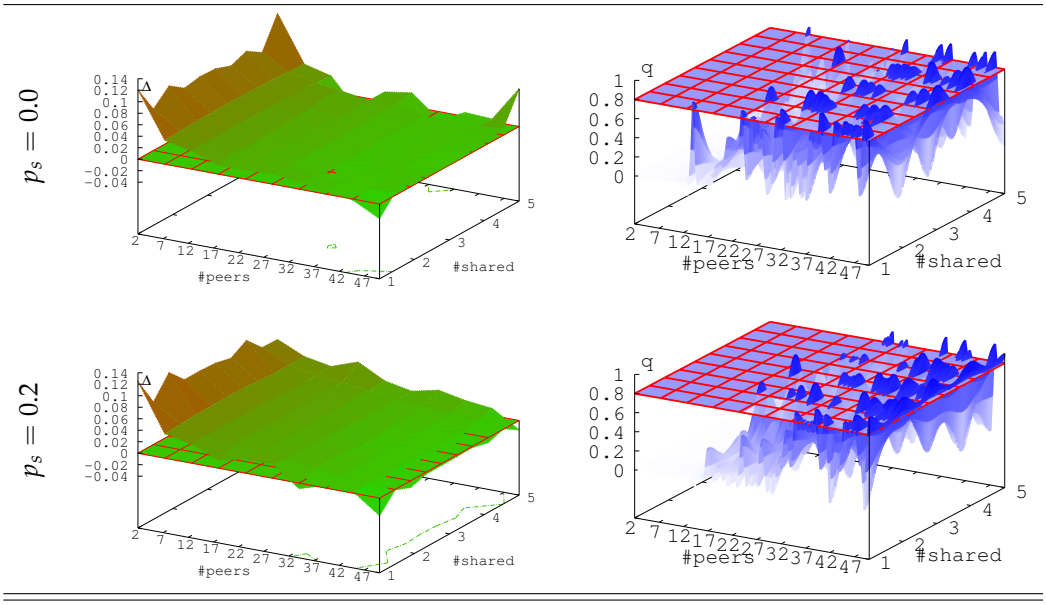


Table 7.4: Results on domain musk, using the same convention as Table 7.3. On musk, $\min_j p_{\text{err}}(\text{P}^j) = 0.25$

that DRL indeed successfully bypasses entity matching to learn a classifier that almost matches the Oracle's performances, and therefore represents a very significant leverage of each peer's data.

For more detailed results, Table 7.2 (left) displays that binning indeed does not affect significantly DRL on average, which is also good news, since it means that there is no restriction on the shared features for DRL to perform well: shared features can be binary, or categorical with any number of modalities. Table 7.7 displays that while the CV tuning of Γ offers leverage to DRL (*vs* $\Gamma = \mathrm{Id}_d$) in general (firmteacher), there are some (rare) domains (mice) on which relying on the simplest $\Gamma = \mathrm{Id}_d$ improves upon the results of CV. This, we believe, comes from the fact that CV as we have carried out is certainly not optimal because one rado can aggregate any number of examples. Finally, Table 7.2 (right) drills down a bit more into the performances of DRL with respect to those of the Oracle on a domain for which DRL obtains somehow "median" performances among all domains, sonar. The Oracle (10-folds CV from the *total* ER'ed $\mathcal{S}$) is *idealistic* since in general we do not know the solution to ER, yet it gives clues on how close DRL may be from the "grail". Interestingly, DRL comes frequently under the statistical significance radar ($\alpha = 0.05$). In notable cases (more frequent as p_s increases), DRL beats Oracle — but not significantly. Aside from theory, these are good news as DRL does not assume ER'ed data, and uses an amount of data which can be $\sim p^2$ times *smaller* than Oracle.

7.6 Discussion and references

We remark that our framework is not formally comparable with ER, since the two address different problems. On the one hand, ER has a much broader applicability than the problem we are interested in here; learning on distributed datasets is less general than ER: in fact, we show a solution that bypasses ER. On the other hand, *learning-based* ER [Bilenko and Mooney, 2003] as well as manifold alignment techniques [Lafon et al., 2006] are viable only knowing some ground truth matches — which are not required for working with rados. From another perspective, in concert with the *open issues* in Getoor and Machanavajjhala [2012], we study ER as component of a pipeline for classification, and highlight how matching is not necessary for the purpose of learning.

In spite of those considerations, we can still draw comparisons with methods that learn on top of data merged through ER (Table 7.5). In both settings, no ID is shared between datasets but some attributes must be so, in order to allow entities comparison for matching or for building rados. Obviously, entity matching does not require the labels to be one of those shared attributes, while this is a fundamental hypothesis of our approach. Although, it is not as restrictive as it might seem at first: if just one peer has labels, then *all* can obtain labels on their own data, via learning from label proportions (Chapter 4): the label handling peer computes the label proportions per each block; the "bags" are defined by examples matching a particular signature. Proportions are then shared among all other peers, which can

metric	ER + learning	Algorithms 11+12
Hp: shared IDs	no	no
Hp: shared variables	necessary	necessary
Hp: shared labels	no	may be relaxed
Fusion / Rados crafting	$O(m^2/m^\star T_{sim})$	$O(m)$
Communication	$m \times d$	$m^\star \times d$
Learning problem	$m \times d$	$m^\star \times d$
Privacy	complex	many guarantees

Table 7.5: Multiple metrics of comparison between learning on top of ER and our approach. Time complexity are estimated for 2 peers in the (**VP**) scenario, assuming all blocks of equal size. "Hp" is short for hypothesis. See Section 7.6 for details.

train a classifier with them so as to estimate a label for each observation.

To discuss time complexity, let us consider a simplified problem with only 2 peers with m examples each in the (**VP**) scenario. In terms of complexity of fusion, if we assume that examples are uniformly distributed in the blocks, each block has size $m/m^\star$ ($m^\star$ is the number of blocks). DRL builds each block rado in time $O(m/m^\star)$, with total cost linear in m. ER takes $O((m/m^\star)^2 \cdot T_{sim})$ to match entities in each of the $m^\star$ blocks, where T_{sim} is the cost of evaluation any similarity function; learning-based methods spend additional time for training; advanced blocking strategies can reduce the average complexity [Bilenko et al., 2006; Whang et al., 2009; Whang and Garcia-Molina, 2012].

Most literature on distributed learning is concerned with limiting communication and designing optimal strategies for merging models [Balcan et al., 2012; Liu and Ihler, 2014]; beside that, previous works focus on horizontal split by observations, with few exceptions [Liu and Ihler, 2012]. In contrast, we exploit what is sufficient to merge *about the data*. The communication protocol is extremely simple. Once rados are crafted locally, they are sent to a central learner in one shot. By Theorem 73, only d-dimensional $m^\star$ block rados are needed. *Data is not accessed anymore* and learning takes place centrally. Moreover, rados help with data compression, being $m^\star \times d$, $m^\star \ll m$ the problem size. ER needs to transfer and learn from all entities, for a total size of $m \times d$.

Learning on data described by different feature sets is the topic of multiple view learning and co-training [Blum and Mitchell, 1998; Sindhwani et al., 2005]. To the best of our knowledge, co-training with unknown matches has not been addressed before. Brefeld et al. [2006] present a multi-view distributed algorithm with co-regularization; although it requires matches for all unlabeled examples.

In settings with multiple data providers, privacy can be crucial [Balcan et al., 2012]. The peers have to trade off model enhancements and information leaks. A learner receives rados to train the model; this can be done by one of the peers, or by a third party — paralleling multi-party ER scenarios [Christen, 2006]. The only information sent through the channel consists of rados, while examples, with their individual sensitive features, are never shared. Computational complexity results

on reconstruct-ability of examples have been proven in Nock et al. [2015], along with NP-HARD characterizations, and protection in the sense of differential privacy [Dwork, 2011; Dwork and Roth, 2013]. Regarding ER, since matching has the potential of de-anonymizing the entities, privacy is usually a very relevant issue to address [Christen, 2006]. However, solutions are not straightforward, as proven by the vast amount of research on the topic [Vatsalan et al., 2013]. Techniques based on partial share of attributes, anonymization or hashing can severely impair the process.

The key message of the Chapter is that entity matching addresses a very general and difficult problem but, in the comparatively restricted context of supervised learning from distributed datasets, accurate learning evading the pitfalls of entity matching is possible with Rademacher observations. Rados have another advantage: they offer a cheap, easily parallelizable material which somehow "compresses" examples while allowing accurate learning. They also offer a readily available solution for guaranteed private exchange of data in a distributed setting. Finally, some domains display that there is significant room space for improvement of how cross-validation of optimized parameters is performed. This interesting problem comes in part from the fact that statistical properties of cross-validation on rados are *not* the same as when carried out on examples; see the recent study of Nock [2016].

7.7 Appendix: proofs

7.7.1 Proof of Theorem 68

First, we remark that $\mathbb{E}_{\Sigma_m}[\langle \boldsymbol{\theta}, \boldsymbol{\pi}_\sigma \rangle] = \langle \boldsymbol{\theta}, \mathbb{E}_{\Sigma_m}[\boldsymbol{\pi}_\sigma] \rangle = (1/2) \cdot \langle \boldsymbol{\theta}, \boldsymbol{\pi} \rangle$, since each example participates to half of the 2^m rados. Letting $v \doteq 2^{m+2} \cdot \mathbb{V}_{\Sigma_m}[\langle \boldsymbol{\theta}, \boldsymbol{\pi}_\sigma \rangle]$, we also have:

$$v = 4 \cdot \sum_{\sigma \in \Sigma_m} \left(\langle \boldsymbol{\theta}, \boldsymbol{\pi}_\sigma \rangle - \frac{1}{2} \cdot \langle \boldsymbol{\theta}, \boldsymbol{\pi} \rangle \right)^2 \tag{7.22}$$

$$= \sum_{\sigma \in \Sigma_m} \left(\sum_i \sigma_i \langle \boldsymbol{\theta}, \boldsymbol{x}_i \rangle \right)^2 \tag{7.23}$$

$$= \sum_{\sigma \in \Sigma_m} \left[\sum_{i=1}^{m} \langle \boldsymbol{\theta}, \boldsymbol{x}_i \rangle^2 + \sum_{i=1}^{e} m \sum_{i' \neq i} \sigma_i \sigma_{i'} \langle \boldsymbol{\theta}, \boldsymbol{x}_i \rangle \langle \boldsymbol{\theta}, \boldsymbol{x}_{i'} \rangle \right] \tag{7.24}$$

$$= 2^m \cdot \sum_{i=1}^{m} \langle \boldsymbol{\theta}, \boldsymbol{x}_i \rangle^2 + \sum_{i=1}^{m} \sum_{i' \neq i} v_{ii'} \cdot \langle \boldsymbol{\theta}, \boldsymbol{x}_i \rangle \boldsymbol{\theta}^\top \boldsymbol{x}_{i'} \ , \tag{7.25}$$

with $v_{ii'} \doteq \sum_{\sigma \in \Sigma_m} \sigma_i \sigma_{i'}$. For any $i \neq i'$, $\sigma_i \sigma_{i'}$ takes exactly the same number of times value $+1$ and value -1, and so $v_{ii'} = 0, \forall i \neq i'$. We get from Equation 7.25:

$$\mathbb{V}_{\Sigma_m}[\langle \boldsymbol{\theta}, \boldsymbol{\pi}_\sigma \rangle] = (1/4) \cdot \sum_{i=1}^{m} \langle \boldsymbol{\theta}, \boldsymbol{x}_i \rangle^2 = (1/4) \cdot \sum_{i=1}^{m} (y_i \langle \boldsymbol{\theta}, \boldsymbol{x}_i \rangle)^2 \ . \tag{7.26}$$

Finally,

$$1 - \frac{2}{m} \cdot \sum_{i=1}^{m} y_i \langle \boldsymbol{\theta}, \boldsymbol{x}_i \rangle + \frac{1}{m} \cdot \sum_{i=1}^{m} (y_i \langle \boldsymbol{\theta}, \boldsymbol{x}_i \rangle)^2 = \frac{1}{m} \cdot \sum_i (1 - y_i \langle \boldsymbol{\theta}, \boldsymbol{x}_i \rangle)_2^2 \ . \tag{7.27}$$

7.7.2 Proof of Lemma 69

As with the sufficiency of mean operator, we need to show the double implication that defines sufficiency for $(\boldsymbol{x}, y)$. This is trivial by applying Theorem 68.

7.7.3 Proof of Theorem 72

We give a proof sketch. Once three simple facts are established in the (**VP**) setting, the Theorem follows. (a) The ground truth entity matching exists. (b) Any BB rado for pair $(\boldsymbol{s}, y)$ would be obtained as a rado summing the contributions of all examples in $\mathcal{S}$ matching the corresponding signature $\boldsymbol{s}$ and class y. (c) We obtain $\mathcal{R}_{\text{B}} \subseteq \mathcal{R}_{\mathcal{S},\Sigma_m}$, from which follows the Theorem's statement with Equation (7.18) and the fact that any sum of a subset of rados in $\mathcal{R}_{\text{B}}$ would also be in $\mathcal{R}_{\mathcal{S},\Sigma_m}$ since an example cannot match two distinct couples (signature, class).

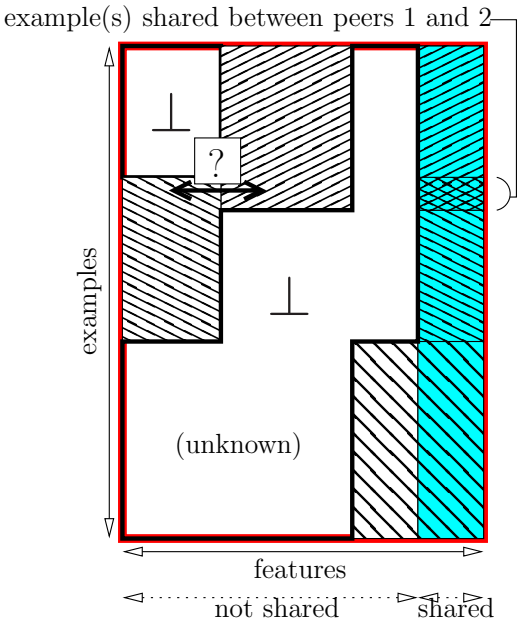


Figure 7.3: Schematic view of our setting, with $p = 3$ peers. Some features (cyan) are described in each peer and one of these shared features is a class. Non-shared features are split among peers. The *total* sample $\mathcal{S}$ is figured by the red rectangle. In the more general setting (**G**), it is not known whether one example, viewed by a peer, also exists in other peers' datasets. In this case, there may be a lot of missing data ($\perp$), but it is not known of which examples.

7.7.4 Proof of Theorem 73

The proof uses the following trick: consider any sample $\mathcal{S}'$ such that its edge vectors match the basic block rados. Remark that $XX^\top = \sum_i (y_i \boldsymbol{x}_i)(y_i \boldsymbol{x}_i)^\top$ in Equation (7.5) depends only on edge vectors. Thus, since $\boldsymbol{\pi} = B\mathbf{1}$, the optimal square loss classifier on $\mathcal{S}'$ is $\hat{\boldsymbol{\theta}}$ in Equation (7.18), which, through Theorem 68, is also the optimal classifier of the empirical risk associated with M-loss.

7.8 Appendix: extension to the more general setting

We extend here the algorithms from Section 7.4 to setting (**G**). Figure (7.3) depicts this more challenging learning setting. We do not assume that the peers handle the same examples, and therefore $m \geq m_j, \forall j \in [p]$. However, hypothesis (7**A**) still holds, that is, the peers share the same set of features.

The definition of rados itself needs to be upgraded first. In the (**VP**) setting we could avoid an obvious complication that we face here: in the computation of the basic block rados, we need to rescale the signature in order to take into account differences in the number of examples per block for each peer. First, we redefine the rados, by a projection onto the set of features that are not shared among peers.

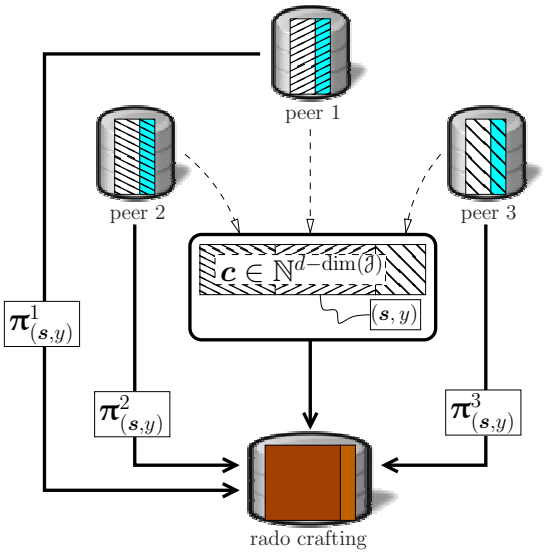


Figure 7.4: Communication for one BB rado, with $(\boldsymbol{s}, y) \in \mathcal{J}_*$. Counter $\boldsymbol{c}$ is defined in Algorithm RadoCraft (see text).

Definition 74. *For any $\boldsymbol{s} \in \mathcal{J}$, $y \in \{-1, 1\}$ and P^j, a rado is:*

$$\boldsymbol{\pi}^j_{(\boldsymbol{s},y)} \doteq \mathrm{proj}_{\mathcal{X}^j \setminus \mathcal{J}} \left(\sum_{i=1}^{m_j} 1_{\mathrm{proj}_{\mathcal{J}}(\boldsymbol{x}^j_i) = \boldsymbol{s} \wedge y^j_i = y} y^j_i \cdot \boldsymbol{x}^j_i \right) \ . \tag{7.28}$$

The definition of *u-basic block* rados follows.

Definition 75. *For any $\boldsymbol{s} \in \mathcal{J}$, $y \in \{-1, 1\}$, $u \in \mathbb{R}$, the* u-basic block *(BB) rado for pair $(\boldsymbol{s}, y)$ is:*

$$\boldsymbol{\pi}_{(\boldsymbol{s},y,u)} \doteq u \cdot \mathrm{lift}_{\mathcal{X}}(y \cdot \boldsymbol{s}) + \sum_{j=1}^{p} \mathrm{lift}_{\mathcal{X}}(\boldsymbol{\pi}^j_{(\boldsymbol{s},y)}) \ . \tag{7.29}$$

The set of block rados is upgraded accordingly. Recall that $\mathcal{J}_* \doteq \{(\boldsymbol{s}, y) \in \mathcal{J} \times \{-1, 1\} : \exists j \in [p], \boldsymbol{\pi}^j_{(\boldsymbol{s},y)} \neq \mathbf{0}\}$ and that $m_* \doteq |\mathcal{J}_*|$. Then, for any $\boldsymbol{u} \in \mathbb{R}^{m_*}$, we let:

$$\mathcal{R}^{\boldsymbol{u}}_{\mathrm{B}} \doteq \{\boldsymbol{\pi}^{u_i}_{\boldsymbol{v}_i}, \forall i \in [m_*]\} \tag{7.30}$$

denote the set of each u_i-BB rado, each coordinate of $\boldsymbol{u}$ being in one-one correspondence with an element of $\mathcal{J}_*$ (represented by $\boldsymbol{v}_i$). Finally, the set of $\boldsymbol{u}$*-block rados* is:

$$\mathcal{R}^{\boldsymbol{u}}_* \doteq \left\{ \sum_{i \in \mathcal{U}} \boldsymbol{\pi}^{u_i}_{\boldsymbol{v}_i}, \forall \mathcal{U} \subseteq [m_*] \right\} \ . \tag{7.31}$$

We state (without proof) that in the general settings (**G**) there exists $\boldsymbol{u} \in \mathbb{R}^{m_*}$ such that $\mathcal{R}^{\boldsymbol{u}}_*$ belongs to $\mathcal{R}_{\mathcal{S},\Sigma_m}$ in expectation. This is now obviously more difficult

Algorithm 13: RadoCraft($\mathsf{P}^1, \mathsf{P}^2, ..., \mathsf{P}^p$)

Input Peers $\mathsf{P}^1, \mathsf{P}^2, ..., \mathsf{P}^p$
$\mathcal{R}^{\boldsymbol{u}}_{\text{B}} \leftarrow \emptyset$
for $\boldsymbol{s} \in \mathcal{J}$, $y \in \{\pm 1\}$
  $\boldsymbol{\pi}_{(\boldsymbol{s},y)} \leftarrow \mathbf{0} \in \mathbb{R}^d$, $\boldsymbol{c} \leftarrow \mathbf{0} \in \mathbb{N}^{d-\dim(\mathcal{J})}$;
  for $j \in [p]$
    $\boldsymbol{\pi}_{(\boldsymbol{s},y)} \leftarrow \boldsymbol{\pi}_{(\boldsymbol{s},y)} + \text{lift}_{\mathcal{X}}(\textsc{craft}(\boldsymbol{s}, y) \rightsquigarrow \mathsf{P}^j)$
  $u \leftarrow (\mathbf{1}^\top \boldsymbol{c})\,(d - \dim(\mathcal{J}))^{-1}$
  $\mathcal{R}^{\boldsymbol{u}}_{\text{B}} \leftarrow \mathcal{R}^{\boldsymbol{u}}_{\text{B}} \cup (u \cdot \text{lift}_{\mathcal{X}}(y \cdot \boldsymbol{s}) + \boldsymbol{\pi}_{(\boldsymbol{s},y)})$
Return $\mathcal{R}^{\boldsymbol{u}}_{\text{B}}$

to tackle than in Theorem 72, since there may be a large amount of missing data ($\perp$ in Figure 7.2) and there would be no one-one correspondence between the peers' examples in general. Yet, there is a interesting property which can be shown in the following (**R**)andomized model: each peer's features remain fixed but there exists a fixed $\boldsymbol{\eta} \in [0,1]^m$ such that example i has probability η_i to be seen by a peer. Let $\overline{\mathcal{S}}$ denote the "expected" sample, where each example is weighted by its probability. For any signature $\boldsymbol{s}$ and class y, $\mathbb{E}[\boldsymbol{\pi}_{(\boldsymbol{s},y)}]$ denotes the expected rado put in $\mathcal{R}^{\boldsymbol{u}}_{\text{B}}$ in Algorithm 13.

Theorem 76. *Under (**R**),* $\forall (\boldsymbol{s}, y) \in \mathcal{J}_*$, $\mathbb{E}[\boldsymbol{\pi}_{(\boldsymbol{s},y)}] \in \mathcal{R}_{\overline{\mathcal{S}},\Sigma_m}$.

Therefore, under setting (**G**), if examples are "seen" independently at random by peers, the expected output of Algorithm 13 still meets the guarantees of Theorem 72 with respect to the expected sample. The fact that $\mathcal{R}^{\boldsymbol{u}}_{\text{B}} \subseteq \mathcal{R}_{\mathcal{S},\Sigma_m}$ from Theorem 72 is also a consequence of Theorem 76 for $\boldsymbol{\eta} = \mathbf{1}$.

Algorithm 13 is a variation of the original Algorithm 11 which takes care of the computation of $\boldsymbol{u}$. Specifically, P^j does the following:

- it computes and return $\boldsymbol{\pi}^j_{(\boldsymbol{s},y)}$; let C_j be the number of examples that are counted in the sum in Equation (7.28);
- it updates counter vector $\boldsymbol{c}$: for each feature $k \notin \mathcal{J}$ it possesses in its local dataset, it does $c_k \leftarrow c_k + C_j$;

Letting $\boldsymbol{v}_i \doteq (\boldsymbol{s}, y) \in \mathcal{J}_*$, the corresponding value of u_i is given by:

$$u_i \doteq (\mathbf{1}^\top \boldsymbol{c})(d - \dim(\mathcal{J}))^{-1}\ , \tag{7.32}$$

which is guaranteed to be non-zero since $\boldsymbol{v}_i \in \mathcal{J}_*$.

7.9 Appendix: additional experimental results

domain	m	d	$\min_j p_{\text{err}}(\mathsf{P}^j)$	p	$\dim(\mathcal{J})$	results
Wine	178	12	0.07	$\{2,3,...,8\}$	$\{1,2,3,4\}$	Table 7.15
Sonar	208	60	0.29	$\{2,3,...,16\}$	$\{1,2,...,20\}$	Table 7.9
Ionosphere	351	33	0.20	$\{2,3,...,9\}$	$\{1,2,...,9\}$	Table 7.11
Mice	1 080	77	0.30	$\{2,3,...,20\}$	$\{1,2,...,20\}$	Table 7.3
Winered	1 599	11	0.26	$\{2,3,...,7\}$	$\{1,2,3,4\}$	Table 7.12
Steelplates	1 941	33	0.16	$\{2,3,...,14\}$	$\{1,2,...,5\}$	Table 7.18
Statlog	4 435	36	0.05	$\{2,3,...,30\}$	$\{1,2,...,5\}$	Table 7.17
Winewhite	4 898	11	0.32	$\{2,3,...,7\}$	$\{1,2,3,4\}$	Table 7.13
Page	5 473	10	0.21	$\{2,3,...,6\}$	$\{1,2,3,4\}$	Table 7.8
Musk	6 598	166	0.25	$\{2,3,...,50\}$	$\{1,2,3,5\}$	Table 7.4
Firmteacher	10 800	16	0.26	$\{2,3,...,7\}$	$\{1,2,...,7\}$	Table 7.10
Phishing	11 055	30	0.11	$\{2,3,4,5\}$	$\{1,2,3,4\}$	Table 7.14
Credit card	14 599	23	0.32	$\{2,3,...,18\}$	$\{1,2,...,5\}$	Table 7.16

Table 7.6: UCI domains used in our experiments Bache and Lichman [2013], with for each the indication of the total number of features (d), examples (m) and the error of the optimal peer in hindsight obtained in our experiments, $\min_j p_{\text{err}}(\mathsf{P}^j)$. Two of the right columns present, for each domain, the range of values for the number of peers (p) and the number of shared features ($\dim(\mathcal{J})$) considered. Experiments are performed considering *all* possible combinations of values of p and $\dim(\mathcal{J})$ within the allocated sets. The rightmost column points to the Table collecting specific results for each domain.

I

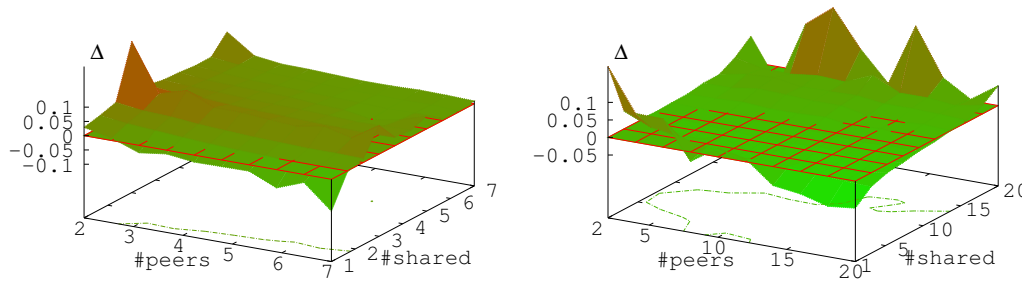


Table 7.7: Results of the dummy regularized DRL ($\Gamma = \mathrm{Id}_d$) on domains firmteacher (left) and mice (right), following the convention of Table 7.11 ($p_s = 0.2$).

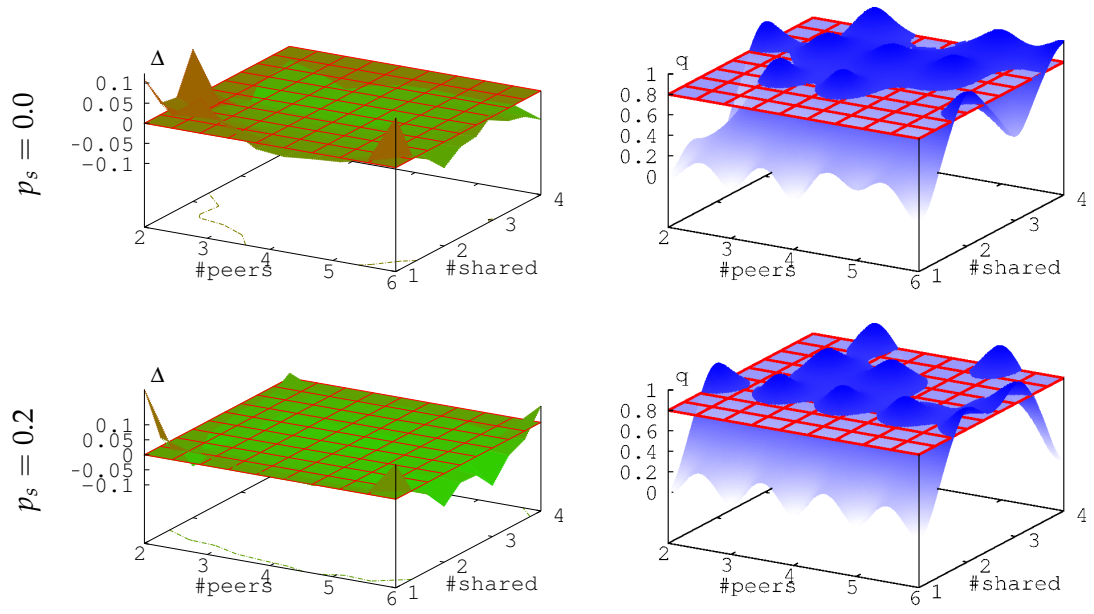


Table 7.8: Results on domain sonar, using the same convention as Table 7.3.

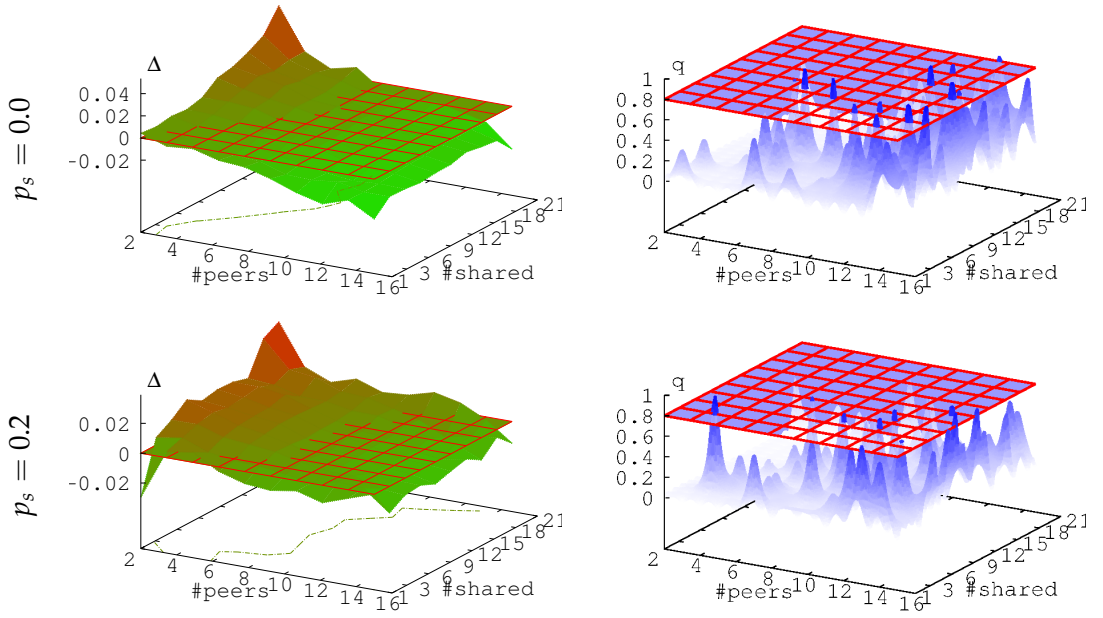


Table 7.9: Results on domain sonar, using the same convention as Table 7.3.

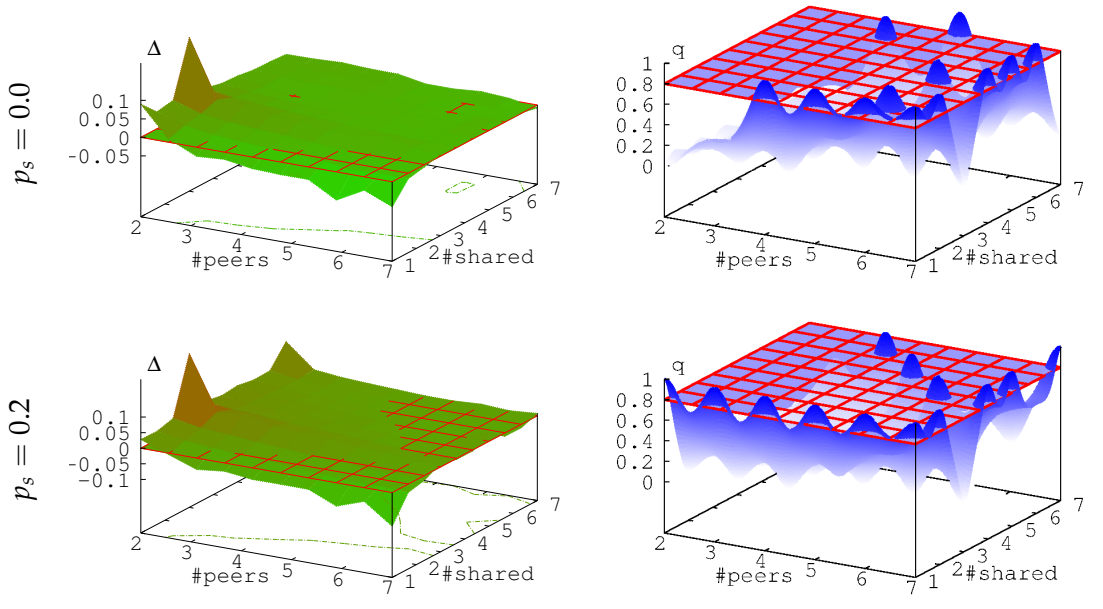


Table 7.10: Results on domain firmteacher, using the same convention as Table 7.3.

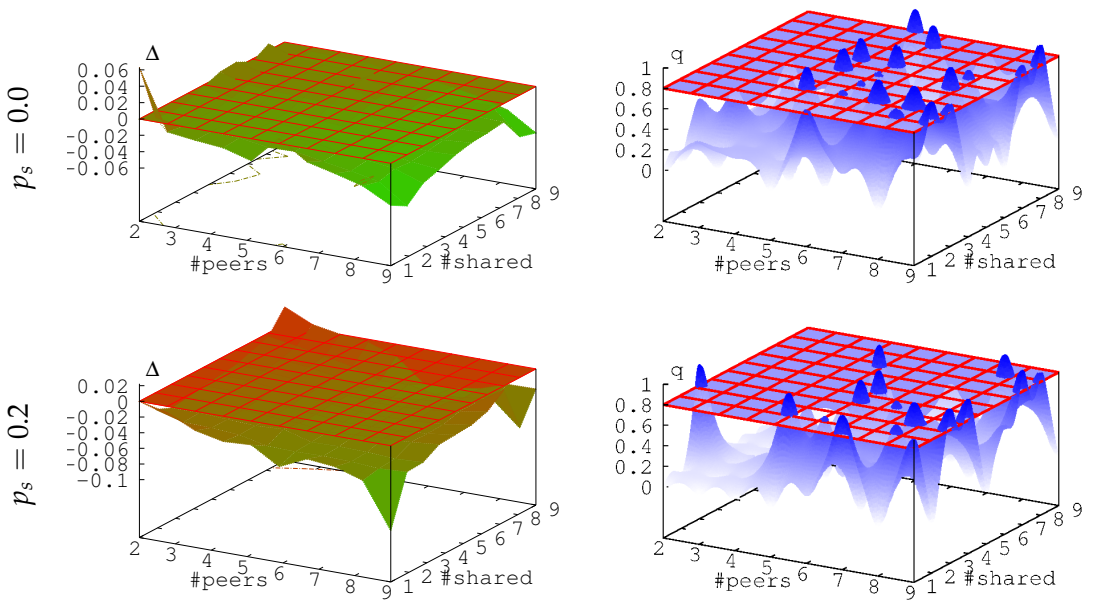


Table 7.11: Results on domain ionosphere, using the same convention as Table 7.3.

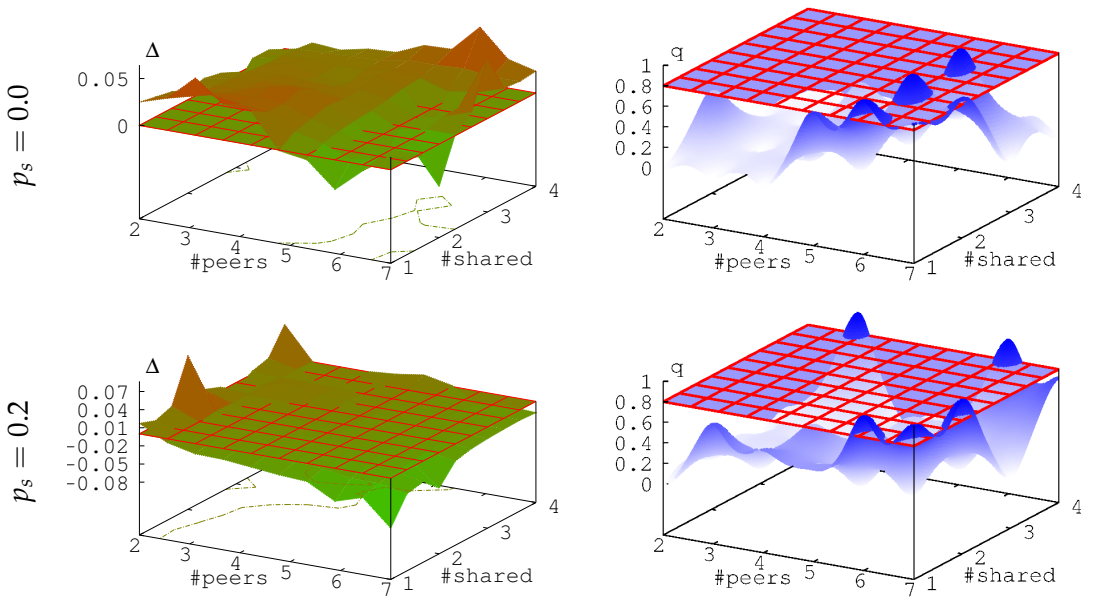


Table 7.12: Results on domain winered, using the same convention as Table 7.3.

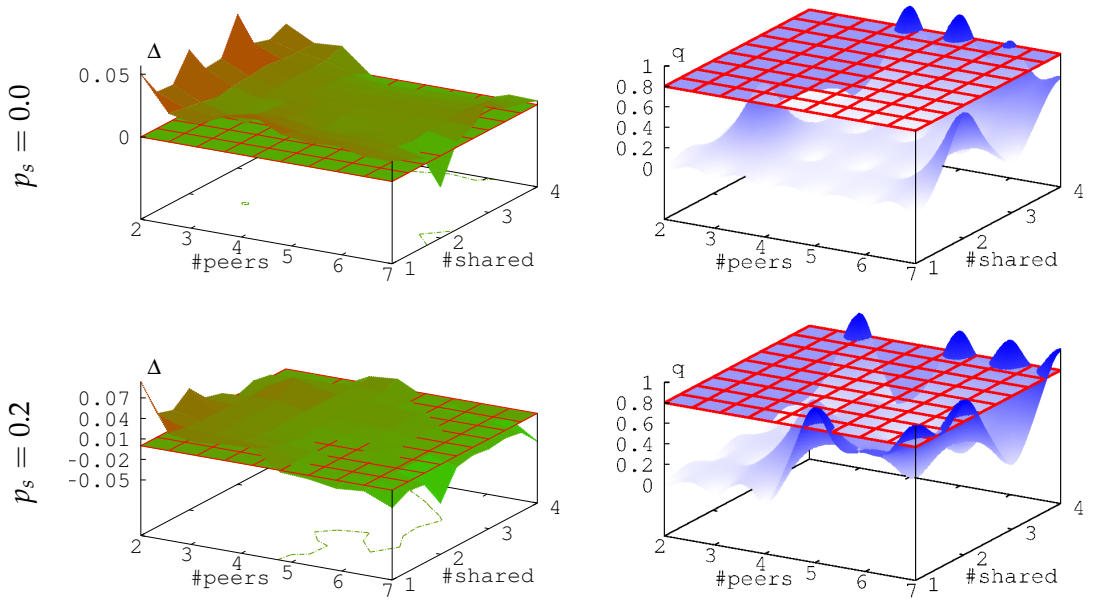


Table 7.13: Results on domain winewhite, using the same convention as Table 7.3.

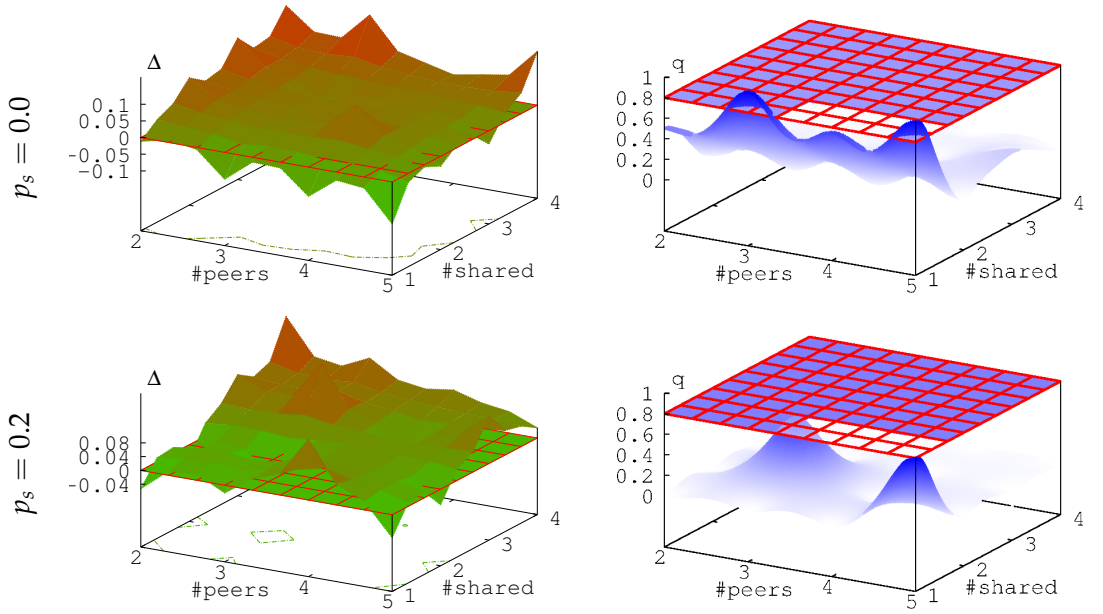


Table 7.14: Results on domain phishing, using the same convention as Table 7.3.

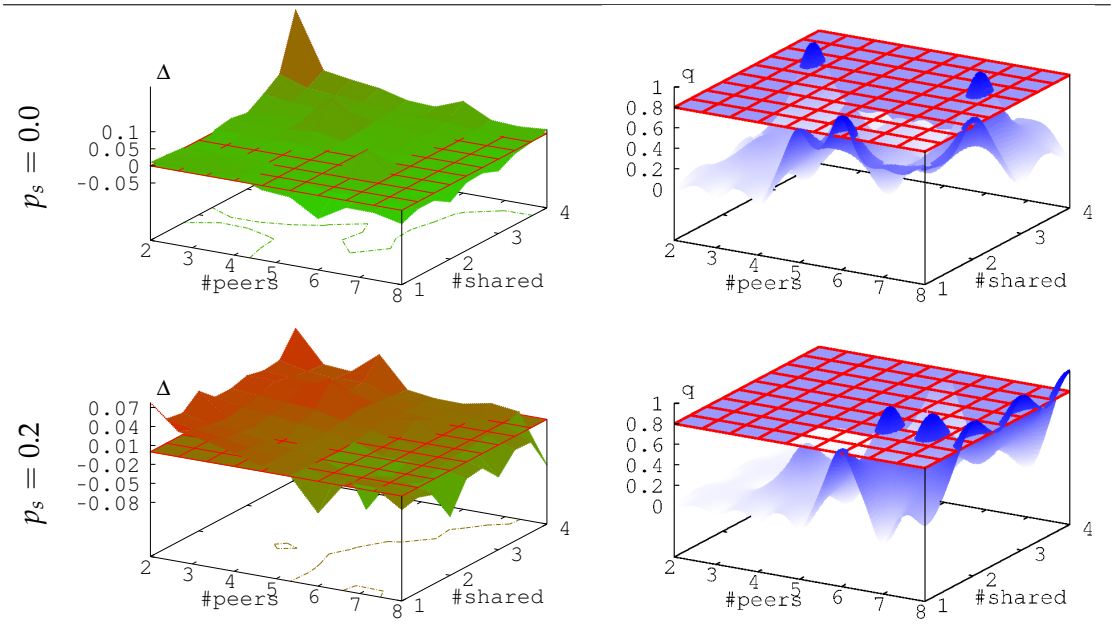


Table 7.15: Results on domain wine, using the same convention as Table 7.3.

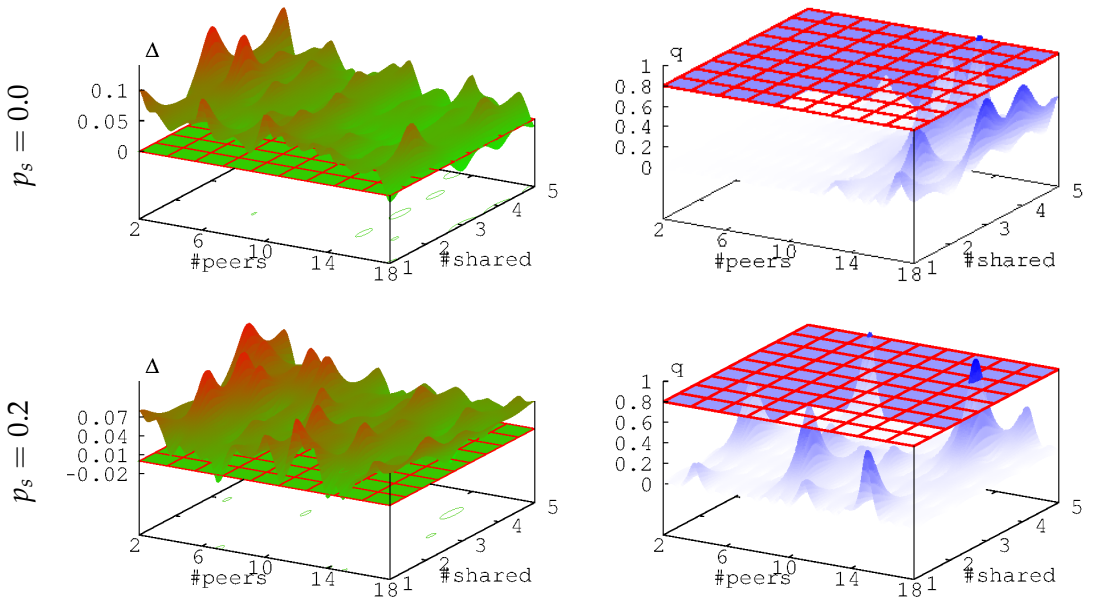


Table 7.16: Results on domain creditcard, using the same convention as Table 7.3.

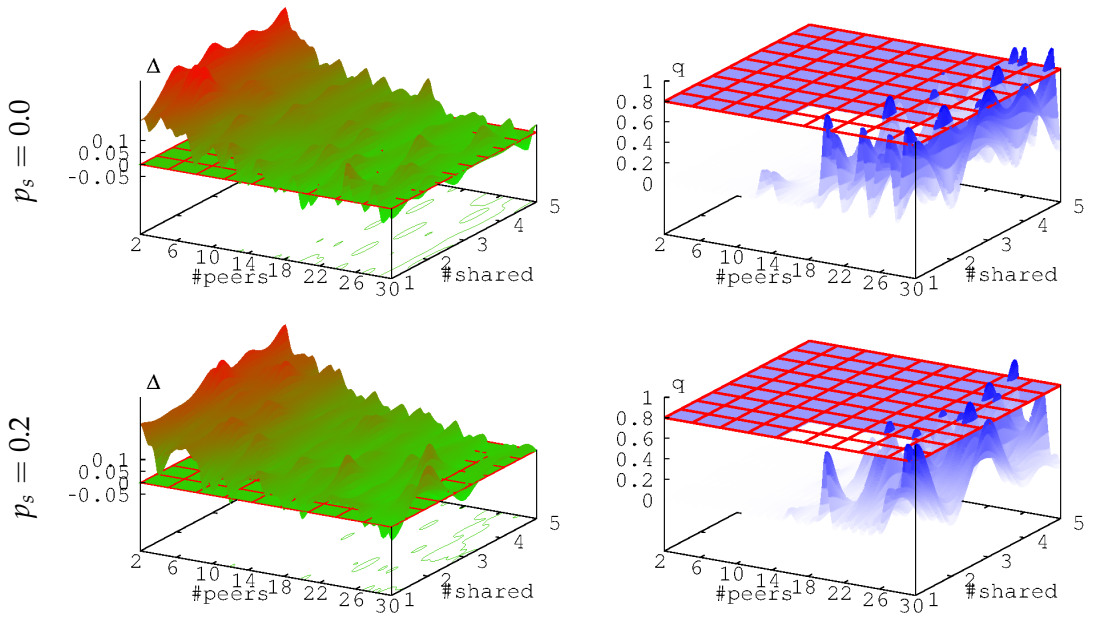


Table 7.17: Results on domain statlog, using the same convention as Table 7.3.

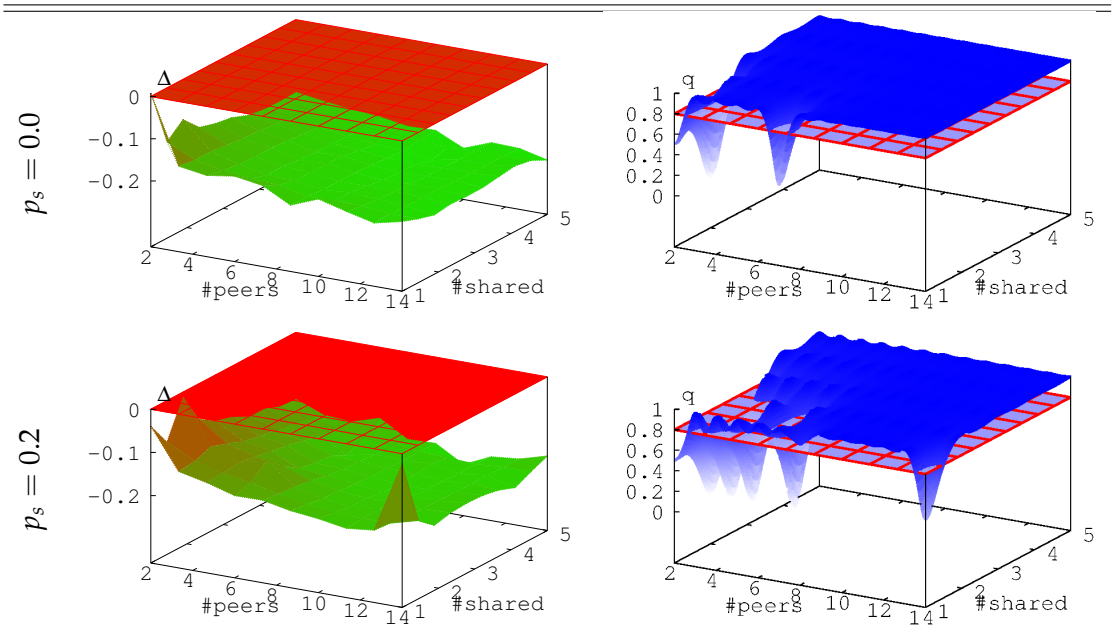


Table 7.18: Results on domain steelplates, using the same convention as Table 7.3.

Chapter 8

Conclusion

Research is seldom a linear path that proceeds from an open problem to its resolution and this Thesis is no exception. The presentation has not strictly followed the chronological order — by publication date of each respective piece of work.

This research journey originated from reasoning on how a classifier may be trained efficiently from label proportions and to do so without resorting to restrictive previous assumptions. The empirical success of Patrini et al. [2014] brought to question whether the proposed framework was viable at the larger extent of weakly supervised problems. A more abstract and mature view was later published in Patrini et al. [2016a] from which, in retrospective, we could have subsumed the case of LLP itself. The content of Chapter 3 and 4 is taken from those publications.

The scenario of learning with noisy labels was analyzed in Patrini et al. [2016a] to reinforce that our approach is also suited for solving the more common and well studied learning problem (Chapter 5). Patrini et al. [2017] deepens the experimental analysis bringing us closer to mainstream deep learning applications (Chapter 6) and also constitutes the major empirical effort of the Thesis in showing the practical outcome of our work on real world problems – for examples in Vision and Language.

At first sight, the problem of learning from vertically distributed datasets elaborated in Patrini et al. [2016b] has little relation to our framework. The peculiar setting was inspired discussing business needs of some of our collaborators involved in building start-up products. Yet we could carve a solution to this demanding application out of our conceptual framework (Chapter 7).

It is therefore evident that the core contribution of this Thesis is methodological. Most of the algorithmic content has revolved around the idea of casting problems with weak supervision into the two-step approach of sufficient statistic estimation followed by standard "fully" supervised learning. Different assumptions and procedures are required for the estimation of the sufficient statistic and we have detailed how to operate in several settings, as recalled above. The effectiveness of this *modus operandi*, by virtue of abstraction and adaptivity to several learning scenarios, suggests that our insight should open new ways for solving other challenging non-standard Machine Learning problems. In fact, the framework already unifies a growing body of literature [Quadrianto et al., 2009; Gao et al., 2016; Raghunathan et al., 2016].

There are similarities with many principles of traditional Software Engineering.

Decoupling is essential for analyzing a complex problem and attacking it via a *divide and conquer* strategy. We have treated the "label issue" in a modular fashion, separated from the learning process. We do not reinvent the wheel and instead we resort to well known optimization methods: gradient descent/L-BFGS (Algorithm 5), SGD and proximal algorithms (Algorithms 9 and 3), back-propagation (Algorithm 10) and the closed form solution for square loss by matrix inversion (Algorithm 12).

Ultimately, we should never be solving a more general problem than the one we are interested in the first place – *i.e.* estimating the latent variable is not a necessary step [Vapnik, 1998; Joulin and Bach, 2012]. We have given another important example in the introduction of Chapter 7, where we have motivated the need of bypassing the expensive and error prone step of entity resolution. This is precisely the glorified role of sufficient statistics, borrowed from statistical modeling and here fitted into Learning Theory; in the Thesis, we have entitled the mean operator and Rademacher observations of such power. We may also think of those statistics as compressing information that allows us to save computational time for learning.

Those insights challenge the literature which is dominated by problem specific solutions, where either loss functions or optimization algorithms are re-designed to handle the lack of supervision. We believe that our proposal takes the right direction of rethinking Machine Learning, that is historically "more akin to a craft that an engineering discipline" [Williamson, 2009]. We have also interpreted our framework as a family of learning reductions [Beygelzimer et al., 2005, 2015].

Theoretical arguments constitute a large section of the Thesis. The Factorization Theorem 18 underpins many of them. We have seen how problem agnostic generalization bounds (Theorem 23) can be easily tailored to LLP and ALN (Theorems 43 and 53). The shape of linear-odd losses, and in particular of symmetric proper losses in Chapter 4, has allowed further manipulation. We have formulated data dependent finite sample bounds on learning linear models (Theorems 26) and we have characterized a form of distribution dependent noise robustness (Theorem 56).

Aside from those results, Theorem 46 expresses generalization bounds that take into account both a more suited definition of Rademacher complexity and a novel complexity measure of the problem, related to the variance of the label proportions; these ideas are new. Surprisingly, Loss Factorization was also the key ingredient for proving that the loss curvature of ReLU networks is immune to label noise (Theorem 64).

More empirical investigation is required to confirm the practical implications of the algorithms for LLP. The success we have obtained on UCI domains — with synthetized bags — is promising for real world applications, where we believe that the relaxed assumption **(A1)**, Chapter 4, may be well justified. In some cases we have obtained surprisingly good performance with AMM (Algorithm 6). The Algorithm — which admittedly does not fit into the two-step framework, but it takes advantage of it via LMM (Algorithm 5) — should be effective in practice, at least in scenarios where linear models can bring good predictive power, in the vein of Mohammady and Culotta [2014]; Ardehaly and Culotta [2015]. In fact, we believe that LLP will have the largest impact in the future on this class of prediction problems, where

features and labels represent people sensitive attributes and behavior.

It is well known that personal attributes such as the one recorded in social networks are highly predictive of sensitive traits [Kosinski et al., 2013]. In turn, such private variables, *e.g.* electoral behavior, sexual preference, likelihood of contracting a disease or committing a crime, are often publicly available given by aggregate as recorded by governments and polling institutions, *e.g.* the census. This can be thought as a scenario for LLP. High predictive performance can be particularly problematic in this context. Owners of large assets of personal data would be able to infer ever more sensitive attributes of their own users or customers, information that has potentially never been shared or recorded elsewhere. Researchers working on LLP have discussed such potentially severe implications [Quadrianto et al., 2009; Yu et al., 2014b]. Yet we are not aware of any experimental evaluation on real world datasets; more research is needed on this front.

Most of the results presented, both from the algorithmics and the theory, is supported by the Factorization Theorem 18 — or the intimately related Theorem 68. It tells us that losses decompose in a way that we can isolate the *contribution of supervision* into a sufficient statistic. An intriguing open question is whether Factorization could help to identify what really matters in learning that is instead *completely unsupervised* [Sutskever et al., 2015].

Bibliography

Altun, Y. and Smola, A. J., Unifying divergence minimization and statistical inference via convex duality. In *COLT*, 2006. (cited on pages 29, 39, and 103)

Andrews, S.; Tsochantaridis, I.; and Hofmann, T., Support vector machines for multiple-instance learning. In *NIPS*, 2002. (cited on page 18)

Ardehaly, E. M. and Culotta, A., Inferring latent attributes of twitter users with label regularization. In *HLT*, 2015. (cited on pages 18, 52, and 166)

Ardehaly, E. M. and Culotta, A., Domain adaptation for learning from label proportions using self-training. In *IJCAI*, 2016. (cited on page 103)

Auer, P.; Cesa-Bianchi, N.; and Fischer, P., Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 47, 2-3 (2002), 235–256. (cited on page 18)

Bach, F.; Jenatton, R.; Mairal, J.; and Obozinski, G., Optimization with sparsity-inducing penalties. *Foundations and Trends® in Machine Learning*, 4, 1 (2012), 1–106. (cited on page 31)

Bache, K. and Lichman, M., 2013. UCI machine learning repository. http://archive.ics.uci.edu/ml. (cited on pages 65, 147, and 157)

Balcan, M. F.; Blum, A.; Fine, S.; and Mansour, Y., Distributed learning, communication complexity and privacy. *arXiv:1204.3514*, (2012). (cited on page 151)

Bartlett, P. L.; I, M. I. J.; and McAuliffe, J. D., Convexity, classification, and risk bounds. *Journal of the American Statistical Association*, 101, 473 (2006), 138–156. (cited on page 13)

Bartlett, P. L. and Mendelson, S., Rademacher and gaussian complexities: Risk bounds and structural results. *JMLR*, 3 (2002). (cited on pages 10, 49, and 83)

Belkin, M.; Niyogi, P.; and Sindhwani, V., Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *JMLR*, 7 (2006), 2399–2434. (cited on pages 17, 56, and 57)

Bengio, Y.; Delalleau, O.; and Roux, N. L., Label propagation and quadratic criterion. *Semi-supervised learning*, 10 (2006). (cited on page 17)

Benjamini, Y. and Hochberg, Y., Controlling the false discovery rate: A practical and powerful approach to multiple testing. *Journal of the Royal Stat. Society. Series B*, 57, 1 (1995), 289–300. (cited on page 148)

Bernstein, G. and Sheldon, D., Consistently estimating markov chains with noisy aggregate data. In *AISTATS*, 2016. (cited on page 18)

Beygelzimer, A.; Dani, V.; Hayes, T.; Langford, J.; and Zadrozny, B., Error limiting reductions between classification tasks. In *ICML*, 2005. (cited on pages 50 and 166)

Beygelzimer, A.; III, H. D.; Langford, J.; and Mineiro, P., Learning reductions that really work. *arXiv:1502.02704*, (2015). (cited on pages 50 and 166)

Bhowmik, A.; Ghosh, J.; and Koyejo, O., Generalized linear models for aggregated data. In *AISTATS*, 2015. (cited on pages 51 and 103)

Bilenko, M.; Basu, S.; and Mooney, R. J., Integrating constraints and metric learning in semi-supervised clustering. In *ICML*, 2004. (cited on page 16)

Bilenko, M.; Kamath, B.; and Mooney, R. J., Adaptive blocking: Learning to scale up record linkage. In *ICDM*, 2006. (cited on page 151)

Bilenko, M. and Mooney, R. J., Adaptive duplicate detection using learnable string similarity measures. In *KDD*, 2003. (cited on pages 138 and 150)

Bleiholder, J. and Naumann, F., Data fusion. *ACM Computing Surveys (CSUR)*, 41, 1 (2008), 1. (cited on pages 138 and 139)

Blum, A. and Chawla, S., Learning from labeled and unlabeled data using graph mincuts. In *ICML*, 2001. (cited on page 18)

Blum, A. and Mitchell, T., Combining labeled and unlabeled data with co-training. In *COLT*, 1998. (cited on pages 18 and 151)

Brefeld, U.; Gärtner, T.; Scheffer, T.; and Wrobel, S., Efficient co-regularised least squares regression. In *ICML*, 2006. (cited on page 151)

Chang, M. W.; Ratinov, L.; and Roth, D., Guiding semi-supervision with constraint-driven learning. In *ACL*, 2007. (cited on page 18)

Chapelle, O.; Schölkopf, B.; and Zien, A., 2006. *Semi-supervised learning*. MIT press Cambridge. (cited on pages 15 and 17)

Chen, B. C.; Chen, L.; Ramakrishnan, R.; and Musicant, D. R., Learning from aggregate views. In *ICDE*, 2006. (cited on pages 51 and 102)

Chen, S.; Liu, B.; Qian, M.; and Zhang, C., Kernel k-means based framework for aggregate outputs classification. In *ICDMW*, 2009. (cited on pages 52 and 102)

Chen, Z.; Qi, Z.; Wang, B.; Cui, L.; Meng, F.; and Shi, Y., Learning with label proportions based on nonparallel support vector machines. *Knowledge-Based Systems*, (2016). (cited on page 102)

Cho, W.-K.-T. and Manski, C.-F., Cross level/ecological inference. *Oxford Handbook of Political Methodology*, (2008), 547–569. (cited on page 104)

Christen, P., Privacy-preserving data linkage and geocoding: Current approaches and research directions. In *ICDMW*, 2006. (cited on pages 138, 151, and 152)

Christen, P., 2012. *Data Matching Concepts and Techniques for Record Linkage, Entity Resolution, and Duplicate Detection*. Springer Data-Centric Systems and Applications. (cited on page 138)

Cour, T.; Sapp, B.; and Taskar, B., Learning from partial labels. *JMLR*, (2011), 1501–1536. (cited on page 15)

Cross, P.-J. and Manski, C.-F., Regressions, short and long. *Econometrica*, 70, 1 (2002), 357–368. (cited on page 104)

Cui, L.; Qi, Z.; and Meng, F., A proportion learning algorithms with density peaks. *Procedia Computer Science*, 91 (2016), 841–846. (cited on page 102)

Dai, A. M. and Le, Q. V., Semi-supervised sequence learning. In *NIPS*, 2015. (cited on page 127)

de Sa, V. R., Spectral clustering with two views. In *ICML Workshop on Learning with Multiple Views*, 2005. (cited on page 18)

Dietterich, T. G.; Lathrop, R. H.; and Lozano-Pérez, T., Solving the multiple instance problem with axis-parallel rectangles. *Artificial Intelligence*, 89 (1997), 31–71. (cited on page 15)

Ding, N. and Vishwanathan, S. V. N., t-logistic regression. In *NIPS*, 2010. (cited on pages 17 and 118)

Divvala, S.; Farhadi, A.; and Guestrin, C., Learning everything about anything: Webly-supervised visual concept learning. In *CVPR*, 2014. (cited on page 105)

du Plessis, M. C.; Niu, G.; and Sugiyama, M., Convex formulation for learning from positive and unlabeled data. In *ICML*, 2015. (cited on pages viii, 15, 26, 47, and 48)

Duchi, J.; Hazan, E.; and Singer, Y., Adaptive subgradient methods for online learning and stochastic optimization. *JMLR*, 12 (2011), 2121–2159. (cited on page 127)

Duncan, O. D. and Davis, B., An alternative to ecological correlation. *American sociological review*, (1953), 665–666. (cited on page 104)

Dwork, C., 2011. Differential privacy. In *Encyclopedia of Cryptography and Security*, 338–340. Springer. (cited on page 152)

Dwork, C. and Roth, A., The algorithmic foundations of differential privacy. *Theoretical Computer Science*, 9, 3-4 (2013), 211–407. (cited on page 152)

Estrada, T.; Armen, R.; and Taufer, M., Automatic selection of near-native protein-ligand conformations using a hierarchical clustering and volunteer computing. In *International Conference on Bioinformatics and Computational Biology*, 2010. (cited on page 138)

Fan, K.; Zhang, H.; Yan, S.; Wang, L.; Zhang, W.; and Feng, J., Learning a generative classifier from label proportions. *NeuroComputing*, 139 (2014), 47–55. (cited on page 102)

Fergus, R.; Fei-Fei, L.; Perona, P.; and Zisserman, A., Learning object categories from internet image searches. *Proceedings of the IEEE*, 98, 8 (2010), 1453–1466. (cited on page 105)

Flaxman, S.-R.; Wang, Y.-X.; and Smola, A.-J., Who supported obama in 2012?: Ecological inference through distribution regression. In *KDD*, 2015. (cited on page 104)

Foulds, J. and Frank, E., A review of multi-instance learning assumptions. *The Knowledge Engineering Review*, 25, 01 (2010), 1–25. (cited on page 16)

Frénay, B. and Verleysen, M., Classification in the Presence of Label Noise: A Survey. *IEEE Transactions on Neural Networks and Learning Systems*, 25, 5 (May 2014), 845–869. (cited on page 118)

Ganchev, K.; Graça, J.; Gillenwater, J.; and Taskar, B., Posterior regularization for structured latent variable models. *JMLR*, 11 (2010), 2001–2049. (cited on page 18)

Gao, W.; Wang, L.; Li, Y. F.; and Zhou, Z. H., Risk minimization in the presence of label noise. In *AAAI*, 2016. (cited on pages 29, 49, and 165)

Garcia-Garcia, D. and Williamson, R. C., Degrees of supervision. In *NIPS Workshops*, 2011. (cited on page 18)

Getoor, L. and Machanavajjhala, A., Entity resolution: theory, practice & open challenges. *Proceedings of the VLDB Endowment*, 5, 12 (2012), 2018–2019. (cited on page 150)

Ghosh, A.; Manwani, N.; and Sastry, P. S., Making risk minimization tolerant to label noise. *Neurocomputing*, 160 (2015), 93–107. (cited on pages 17, 105, 110, 118, and 128)

Glorot, X. and Bengio, Y., Understanding the difficulty of training deep feedforward neural networks. In *AISTATS*, 2010. (cited on page 129)

Grandvalet, Y. and Bengio, Y., Semi-supervised learning by entropy minimization. In *NIPS*, 2004. (cited on page 17)

He, K.; Zhang, X.; Ren, S.; and Sun, J., Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *ICCV*, 2015. (cited on page 127)

He, K.; Zhang., X.; Ren, S.; and Sun, J., Deep residual learning for image recognition. In *CVPR*, 2016a. (cited on pages 120, 128, and 129)

He, K.; Zhang, X.; Ren, S.; and Sun, J., Identity mappings in deep residual networks. In *ECCV*, 2016b. (cited on pages 120 and 128)

Hernández, J. and Inza, I., Learning naive bayes models for multiple-instance learning with label proportions. In *Conference of the Spanish Association for Artificial Intelligence*, Springer, 2011. (cited on page 102)

Hernández-González, J.; Inza, I.; Crisol-Ortíz, L.; Guembe, M. A.; Iñarra, M. J.; and Lozano, J. A., Fitting the data from embryo implantation prediction: Learning from label proportions. *Statistical methods in medical research*, (2016). (cited on page 51)

Hernandez-Gonzalez, J.; Inza, I.; and Lozano, J., 2016. Weak supervision and other non-standard classification problems: a taxonomy. In *PRL*. Elsevier. (cited on page 16)

Hernández-González, J.; Inza, I.; and Lozano, J. A., Learning bayesian network classifiers from label proportions. *Pattern Recognition*, 46, 12 (2013), 3425–3440. (cited on page 102)

Hernández-González, J.; Inza, I.; and Lozano, J. A., A novel weakly supervised problem: Learning from positive-unlabeled proportions. In *Conference of the Spanish Association for Artificial Intelligence*, Springer, 2015. (cited on page 103)

Hernández-González, J.; Inza, I.; and Lozano, J. A., Learning from proportions of positive and unlabeled examples. *International Journal of Intelligent Systems*, (2016). (cited on page 103)

Hochreiter, S. and Schmidhuber, J., Long short-term memory. *Neural computation*, 9, 8 (1997), 1735–1780. (cited on page 120)

Hope, T. and Shahaf, D., Ballpark learning: Estimating labels from rough group comparisons. In *ECML-PKDD16*, 2016. (cited on page 103)

Horn, R. A. and Johnson, C. R., 2012. *Matrix analysis*. Cambridge university press. (cited on pages 70 and 71)

Huang, G.; Sun, Y.; Liu, Z.; Sedra, D.; and Weinberger, K., Deep networks with stochastic depth. In *ECCV*, 2016. (cited on page 128)

Ioffe, S. and Szegedy, C., Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, 2015. (cited on page 120)

Jaakkola, T. S. and Jordan, M. I., Bayesian parameter estimation via variational methods. *Statistics and Computing*, 10, 1 (2000), 25–37. (cited on page 49)

Janzing, D.; Peters, J.; Sgouritsa, E.; Zhang, K.; Mooij, J. M.; and Schölkopf, B., On causal and anticausal learning. In *ICML*, 2012. (cited on page 17)

Joachims, T., Transductive inference for text classification using support vector machines. In *ICML*, 1999. (cited on page 17)

Joulin, A. and Bach, F. R., A convex relaxation for weakly supervised classifiers. In *ICML*, 2012. (cited on pages 18 and 166)

Judge, G. G.; Miller, D. J.; and Cho, W. K. T., 2004. An information theoretic approach to ecological estimation and inference. In *Ecological inference: New methodological strategies*, 162–187. Cambridge University Press. (cited on page 104)

Kakade, S. M.; Sridharan, K.; and Tewari, A., On the complexity of linear prediction: Risk bounds, margin bounds, and regularization. In *NIPS*, 2009. (cited on pages 10 and 49)

Kawaguchi, K., Deep learning without poor local minima. In *NIPS*, 2016. (cited on page 126)

Kearns, M., Efficient noise-tolerant learning from statistical queries. *Journal of the ACM*, 45 (1998). (cited on page 135)

Kearns, M. J. and Mansour, Y., On the boosting ability of top-down decision tree learning algorithms. In *STOC*, 1996. (cited on page 53)

King, G., 1997. *A solution to the ecological inference problem: reconstructing individual behavior from aggregate data*. Princeton University Press. (cited on page 104)

King, G.; Tanner, M.-A.; and Rosen, O., 2004. *Ecological inference: New methodological strategies*. Cambridge University Press. (cited on page 104)

Kingma, D. P.; Mohamed, S.; Rezende, D. J.; and Welling, M., Semi-supervised learning with deep generative models. In *NIPS*, 2014. (cited on page 18)

Kosinski, M.; Stillwell, D.; and Graepel, T., Private traits and attributes are predictable from digital records of human behavior. *PNAS*, 110, 15 (2013), 5802–5805. (cited on page 167)

Kotzias, D.; Denil, M.; Freitas, N. D.; and Smyth, P., From group to individual labels using deep features. In *KDD*, 2015. (cited on pages 51 and 102)

Krause, J.; Sapp, B.; Howard, A.; Zhou, H.; Toshev, A.; Duerig, T.; Philbin, J.; and Fei-Fei, L., The unreasonable effectiveness of noisy data for fine-grained recognition. In *ECCV*, 2016. (cited on pages 105, 119, 131, and 135)

Krizhevsky, A. and Hinton, G., Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009. (cited on page 126)

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E., Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012. (cited on page 129)

Kück, H.; Carbonetto, P.; and de Freitas, N., A constrained semi-supervised learning approach to data association. In *ECCV*, 2004. (cited on page 102)

Kuck, H. and de Freitas, N., Learning about individuals from group statistics. In *UAI*, 2005. (cited on pages 15, 16, 18, 52, and 102)

Lafon, S.; Keller, Y.; and Coifman, R. R., Data fusion and multicue data matching by diffusion maps. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28, 11 (2006), 1784–1797. (cited on page 150)

Lai, K. T.; Yu, F.; Chen, M. S.; and Chang, S. F., Video event detection by inferring temporal instance labels. In *CVPR*, 2014. (cited on page 52)

Laird, P. D., 1988. *Learning from Good and Bad Data*. Kluwer Academic Publishers, Norwell, MA, USA. (cited on page 135)

Lanckriet, G. R. G.; Bie, T. D.; Cristianini, N.; Jordan, M. I.; and Noble, W. S., A statistical framework for genomic data fusion. *Bioinformatics*, 20, 16 (2004), 2626–2635. (cited on page 138)

Lawrence, N. D. and Jordan, M. I., Semi-supervised learning via gaussian processes. In *NIPS*, 2004. (cited on page 18)

LeCun, Y.; Bottou, L.; Bengio, Y.; and Haffner, P., Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86, 11 (1998), 2278–2324. (cited on pages 120 and 126)

Ledoux, M. and Talagrand, M., 1991. *Probability in Banach Spaces*. Springer Verlag. (cited on page 89)

Lehmann, E. L. and Casella, G., 1998. *Theory of point estimation*, vol. 31. Springer Science & Business Media. (cited on page 15)

Li, F. and Taylor, G., Alter-cnn: An approach to learning from label proportions with application to ice-water classification. In *NIPS Workshop on Learning and privacy with Incomplete Data and Weak Supervision*, 2015. (cited on pages 52 and 102)

Li, Y.-F.; Tsang, I. W.; Kwok, J. T.; and Zhou, Z.-H., Convex and scalable weakly labeled svms. *JMLR*, (2013), 2151–2188. (cited on page 18)

Liang, P.; Jordan, M. I.; and Klein, D., Learning from measurements in exponential families. In *ICML*, 2009. (cited on pages 18 and 52)

Liu, L.-P. and Dietterich, T. G., Learnability of the superset label learning problem. In *ICML*, 2014. (cited on page 15)

Liu, Q. and Ihler, A., Distributed parameter estimation via pseudo-likelihood. In *ICML*, 2012. (cited on page 151)

Liu, Q. and Ihler, A. T., Distributed estimation, information loss and exponential families. In *NIPS*, 2014. (cited on page 151)

Liu, T. and Tao, D., Classification with noisy labels by importance reweighting. *IEEE Transactions on pattern analysis and machine intelligence*, 38, 3 (2016), 447–461. (cited on pages 124 and 135)

Long, P. M. and Servedio, R. A., Random classification noise defeats all convex potential boosters. *Machine learning*, 78, 3 (2010), 287–304. (cited on pages 108, 109, 110, and 118)

Ma, F.; Shi, Y.; Wang, B.; and Chen, Z., Research on the classification of commercial banks? fund clients based on learning with label proportions. *Procedia Computer Science*, 91 (2016), 988–994. (cited on page 51)

Maas, A. L.; Daly, R. E.; Pham, P. T.; Huang, D.; Ng, A. Y.; and Potts, C., Learning word vectors for sentiment analysis. In *ACL*, 2011. (cited on page 126)

Mann, G. S. and McCallum, A., Generalized expectation criteria for semi-supervised learning of conditional random fields. In *ACL*, 2008. (cited on page 18)

Mann, G. S. and McCallum, A., Generalized expectation criteria for semi-supervised learning with weakly labeled data. *JMLR*, (2010), 955–984. (cited on page 18)

Manwani, N. and Sastry, P. S., Noise tolerance under risk minimization. *IEEE Transactions on Cybernetics*, 43, 3 (2013), 1146–1151. (cited on pages 108, 110, 116, and 118)

Markowitz, H., Portfolio selection. *Journal of Finance*, 6 (1952), 77–91. (cited on page 142)

Masnadi-Shirazi, H.; Mahadevan, V.; and Vasconcelos, N., On the design of robust classifiers for computer vision. In *CVPR*, 2010. (cited on pages 17 and 118)

Masnadi-Shirazi, H. and Vasconcelos, N., On the design of loss functions for classification: theory, robustness to outliers, and savageboost. In *NIPS*, 2009. (cited on pages 118 and 128)

McDiarmid, C., Concentration. In *Probabilistic Methods for Algorithmic Discrete Mathematics*, 1–54, Springer Verlag, 1998. (cited on pages 11 and 19)

Menon, A.; Rooyen, B. V.; Ong, C. S.; and Williamson, B., Learning from corrupted binary labels via class-probability estimation. In *ICML*, 2015. (cited on pages 16, 118, 120, 124, 127, and 135)

Menon, A.; van Rooyen, B.; and Natarajan, N., Learning from binary labels with instance-dependent corruption. *arXiv preprint arXiv:1605.00751*, (2016). (cited on pages 105, 106, 118, and 131)

Mintz, M.; Bills, S.; Snow, R.; and Jurafsky, D., Distant supervision for relation extraction without labeled data. In *ACL*, 2009. (cited on page 15)

Misra, I.; Zitnick, C. L.; Mitchell, M.; and Girshick, R., Seeing through the human reporting bias: Visual classifiers from noisy human-centric labels. In *CVPR*, 2016. (cited on page 105)

Mnih, V. and Hinton, G. E., Learning to label aerial images from noisy data. In *ICML*, 2012. (cited on pages 119 and 134)

Mohammady, E. and Culotta, A., Using county demographics to infer attributes of twitter users. In *ACL*, 2014. (cited on pages 52 and 166)

Musicant, D. J.; Christensen, J. M.; and Olson, J. F., Supervised learning by training on aggregate outputs. In *ICDM*, 2007. (cited on pages 51 and 102)

Muzellec, B.; Nock, R.; Patrini, G.; and Nielsen, F., Tsallis regularized optimal transport and ecological inference. In *AAAI*, 2017. (cited on page 104)

Natarajan, N.; Dhillon, I. S.; Ravikumar, P. K.; and Tewari, A., Learning with noisy labels. In *NIPS*, 2013. (cited on pages 15, 17, 105, 106, 107, 108, 114, 116, 118, 120, 122, and 135)

Ni, T.; Chung, F.-L.; and Wang, S., Support vector machine with manifold regularization and partially labeling privacy protection. *Information Sciences*, 294 (2015), 390–407. (cited on page 102)

Nigam, K. and Ghani, R., Analyzing the effectiveness and applicability of co-training. In *CIKM*, 2000. (cited on page 18)

Nock, R., Learning games and Rademacher observations losses. *CoRR*, abs/1512.05244 (2015). (cited on page 142)

Nock, R., On regularizing rademacher observation losses. In *NIPS*, 2016. (cited on page 152)

Nock, R. and Nielsen, F., Bregman divergences and surrogates for learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31 (2009), 2048–2059. (cited on pages 26, 53, and 123)

Nock, R.; Patrini, G.; and Friedman, A., Rademacher Observations, private data and boosting. In *ICML*, 2015. (cited on pages 4, 141, 142, 144, and 152)

Park, Y. and Ghosh, J., A probabilistic imputation framework for predictive analysis using variably aggregated, multi-source healthcare data. In *SIGHIT International Health Informatics Symposium*, 2012. (cited on page 103)

Park, Y. and Ghosh, J., Cudia: Probabilistic cross-level imputation using individual auxiliary information. *ACM Transactions on Intelligent Systems and Technology*, 4, 4 (2013), 66. (cited on page 103)

Patrini, G.; Nielsen, F.; Nock, R.; and Carioni, M., Loss factorization, weakly supervised learning and label noise robustness. In *ICML*, 2016a. (cited on pages 4, 49, and 165)

Patrini, G.; Nock, R.; Hardy, S.; and Caetano, T., Fast learning from distributed datasets without entity matching. In *IJCAI*, 2016b. (cited on pages 4 and 165)

Patrini, G.; Nock, R.; Rivera, P.; and Caetano, T., (Almost) no label no cry. In *NIPS*, 2014. (cited on pages 4, 29, 53, and 165)

Patrini, G.; Rozza, A.; Menon, A.; Nock, R.; and Qu, L., Making neural networks robust to label noise: a loss correction approach. In *Submitted to CVPR*, 2017. (cited on pages 4 and 165)

Pérez-Ortiz, M.; Gutiérrez, P. A.; Carbonero-Ruz, M.; and Hervás-Martínez, C., Learning from label proportions via an iterative weighting scheme and discriminant analysis. In *Conference of the Spanish Association for Artificial Intelligence*, 79–88, Springer, 2016. (cited on page 102)

Platt, J., Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in large margin classifiers*, 10, 3 (1999), 61–74. (cited on pages 10 and 102)

Posnett, D.; Filkov, V.; and Devanbu, P., Ecological inference in empirical software engineering. In *International Conference on Automated Software Engineering*, 2011. (cited on page 104)

Qi, Z.; Wang, B.; Meng, F.; and Niu, L., Learning with label proportions via npsvm. *IEEE Transactions on Cybernetics*, (2016). (cited on page 102)

Quadrianto, N.; Smola, A. J.; Caetano, T. S.; and Le, Q. V., Estimating labels from label proportions. *JMLR*, 10 (2009), 2349–2374. (cited on pages viii, 23, 29, 49, 51, 52, 53, 55, 63, 66, 77, 102, 103, 165, and 167)

Raghunathan, A.; Frostig, R.; Duchi, J.; and Liang, P., Estimation from indirect supervision with linear moments. In *ICML*, 2016. (cited on pages viii, 15, 18, 49, and 165)

Ramaswamy, H. G.; Scott, C.; and Tewari, A., Mixture proportion estimation via kernel embedding of distributions. In *ICML*, 2016. (cited on page 135)

Rasmus, A.; Berglund, M.; Honkala, M.; Valpola, H.; and Raiko, T., Semi-supervised learning with ladder networks. In *NIPS*, 2015. (cited on page 17)

Rastogi, V.; Dalvi, N.-N.; and Garofalakis, M.-N., Large-scale collective entity matching. *VLDB*, 4, 4 (2011), 208–218. (cited on page 138)

Reed, S.; Lee, H.; Anguelov, D.; Szegedy, C.; Erhan, D.; and Rabinovich, A., Training deep neural networks on noisy labels with bootstrapping. *ICLR Workshops*, (2015). (cited on pages 119, 128, 134, and 135)

Reid, M. D. and Williamson, R. C., Composite binary losses. *JMLR*, 11 (2010), 2387–2422. (cited on pages 10, 13, and 123)

Robinson, W.-S., Ecological correlations and the behavior of individuals. *American Sociological Review*, 15, 3 (1950), 351–357. (cited on page 104)

Rüping, S., Svm classifier estimation from group probabilities. In *ICML*, 2010. (cited on pages 51, 63, and 102)

Samdani, R.; Chang, M.-W.; and Roth, D., Unified expectation maximization. In *HLT*, 2012. (cited on page 18)

Sanderson, T. and C. Scott, C., Class proportion estimation with application to multiclass anomaly rejection. In *AISTATS*, 2014. (cited on page 135)

Schölkopf, B. and Smola, A. J., 2002. *Learning with kernels: Support vector machines, regularization, optimization, and beyond*. MIT press. (cited on page 43)

Schroff, F.; Criminisi, A.; and Zisserman, A., Harvesting image databases from the web. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33, 4 (2011), 754–766. (cited on page 105)

Scott, C.; Blanchard, G.; and Handy, G., Classification with asymmetric label noise : Consistency and maximal denoising. In *COLT*, 2013. (cited on page 135)

Seeger, M., Learning with labeled and unlabeled data. Technical report, Institute for ANC, Edinburgh, UK, 2000. (cited on page 18)

Shalev-Shwartz, S. and Ben-David, S., 2014. *Understanding machine learning: From theory to algorithms*. Cambridge University Press. (cited on pages 7, 10, and 19)

Shalev-Shwartz, S.; Singer, Y.; Srebro, N.; and Cotter, A., Pegasos: Primal estimated sub-gradient solver for svm. *Mathematical programming*, 127, 1 (2011), 3–30. (cited on pages 31 and 111)

Sheldon, D. R. and Dietterich, T. G., Collective graphical models. In *NIPS*, 2011. (cited on page 18)

Shi, J. and Malik, J., Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22 (2000), 888–905. (cited on page 58)

Sindhwani, V.; Niyogi, P.; and Belkin, M., A co-regularized approach to semi-supervised learning with multiple views. In *ICML Workshop on Learning with Multiple Views*, 2005. (cited on page 151)

Smola, A.; Gretton, A.; Song, L.; and Schölkopf, B., A Hilbert space embedding for distributions. In *Algorithmic Learning Theory*, 13–31, Springer, 2007. (cited on page 49)

Song, L.; Huang, J.; Smola, A. J.; and Fukumizu, K., Hilbert space embeddings of conditional distributions with applications to dynamical systems. In *ICML*, 2009. (cited on page 49)

Sproull, R. F.; DuMouchel, W. H.; Kearns, M.; Lampson, B. W.; Landau, S.; Leiter, M. E.; Parker, E. R.; and Weinberger, P. J., 2015. Bulk collection of signal intelligence: technical options. In *Committee on Responding to Section 5(d) of Presidential Policy Directive 28: The Feasibility of Software to Provide Alternatives to Bulk Signals Intelligence Collection*. National Academy Press. (cited on page 138)

Srivastava, N.; Hinton, G. E.; Krizhevsky, A.; Sutskever, I.; and Salakhutdinov, R., Dropout: a simple way to prevent neural networks from overfitting. *JMLR*, 15, 1 (2014), 1929–1958. (cited on page 120)

Stempfel, G. and Ralaivola, L., 2009. Learning SVMs from sloppily labeled data. In *ICANN*, 884–893. Springer. (cited on page 17)

Stolpe, M. and Morik, K., Learning from label proportions by optimizing cluster model selection. In *ECML-PKDD*, 2011. (cited on pages 51 and 102)

Sukhbaatar, S.; Bruna, J.; Paluri, M.; Bourdev, L.; and Fergus, R., Training convolutional networks with noisy labels. In *ICLR Workshops*, 2015. (cited on pages 119, 120, 122, 130, and 135)

Surdeanu, M.; Tibshirani, J.; Nallapati, R.; and Manning, C. D., Multi-instance multi-label learning for relation extraction. In *EMNLP-CoNLL*, 2012. (cited on page 15)

Sutskever, I.; Jozefowicz, R.; Gregor, K.; Rezende, D.; Lillicrap, T.; and Vinyals, O., Towards principled unsupervised learning. *arXiv preprint arXiv:1511.06440*, (2015). (cited on page 167)

Sutton, R. S. and Barto, A. G., 1998. *Reinforcement learning: An introduction*, vol. 1. MIT press Cambridge. (cited on page 18)

Sweeney, L., Privacy-enhanced linking. *ACM SIGKDD Explorations Newsletter*, 7, 2 (2005), 72–75. (cited on page 138)

Szummer, M. and Jaakkola, T. S., Information regularization with partially labeled data. In *NIPS*, 2002. (cited on page 17)

Tsui, F.; Espino, J. U.; Dato, V. M.; Gesteland, P. H.; Hutman, J.; and Wagner, M. M., Technical description of rods: a real-time public health surveillance system. *Journal of the American Medical Informatics Association*, 10, 5 (2003), 399–408. (cited on page 138)

van Rooyen, B., 2015. *Machine Learning via Transitions*. Ph.D. thesis, The Australian National University. (cited on pages 118 and 122)

van Rooyen, B.; Menon, A. K.; and Williamson, R. C., Learning with symmetric label noise: The importance of being unhinged. In *NIPS*, 2015. (cited on pages 17, 26, 28, 106, 108, 110, 118, and 128)

Vapnik, V., 1998. *Statistical Learning Theory*. John Wiley. (cited on pages 12, 23, and 166)

Vatsalan, D.; Christen, P.; and Verykios, V. S., A taxonomy of privacy-preserving record linkage techniques. *Information Systems*, 38, 6 (2013), 946–969. (cited on page 152)

Villani, C., 2008. *Optimal transport: old and new*, vol. 338. Springer Science & Business Media. (cited on page 104)

Von Luxburg, U., A tutorial on spectral clustering. *Statistics and computing*, 17, 4 (2007), 395–416. (cited on page 56)

Wager, S.; Blocker, A.; and Cardin, N., Weakly supervised clustering: Learning fine-grained signals from coarse labels. *The Annals of Applied Statistics*, 9, 2 (2015), 801–820. (cited on pages 52 and 102)

Wakefield, J. and Shaddick, G., Health-exposure modeling and the ecological fallacy. *Biostatistics*, 7, 3 (2006), 438–455. (cited on page 104)

Wang, B.; Chen, Z.; and Qi, Z., Linear twin svm for learning from label proportions. In *International Conference on Web Intelligence and Intelligent Agent Technology*, 2015. (cited on page 102)

Weston, J.; Ratle, R.; Mobahi, H.; and Collobert, R., 2012. Deep learning via semi-supervised embedding. In *Neural Networks: Tricks of the Trade*, 639–655. Springer. (cited on page 17)

Whang, S. E. and Garcia-Molina, H., Joint entity resolution. In *ICDE*, 2012. (cited on page 151)

Whang, S.-E.; Menestrina, D.; Koutrika, G.; Theobald, M.; and Garcia-Molina, H., Entity resolution with iterative blocking. In *SIGMOD*, 219–232, 2009. (cited on pages 144 and 151)

Williamson, R., Reconceiving machine learning. Technical report, NICTA, 2009. (cited on page 166)

Xiao, T.; Xia, T.; Yang, T.; Huang, C.; and Wang, X., Learning from massive noisy labeled data for image classification. In *CVPR*, 2015. (cited on pages 105, 119, 120, 126, 129, 130, 131, 134, and 135)

Yamanishi, Y.; Vert, J. P.; and Kanehisa, K., Protein network inference from multiple genomic data: a supervised approach. *Bioinformatics*, 20, suppl 1 (2004), i363–i370. (cited on page 138)

Yang, Z.; Cohen, W.; and Salakhutdinov, R., Revisiting semi-supervised learning with graph embeddings. In *ICML*, 2016. (cited on page 17)

Yarin, G. and Ghahramani, Z., A theoretically grounded application of dropout in recurrent neural networks. In *NIPS*, 2016. (cited on page 127)

Yu, F. X.; Cao, L.; Merler, M.; Codella, N.; Chen, T.; Smith, J. R.; and Chang, S.-F., Modeling attributes from category-attribute proportions. In *International Conference on Multimedia*, 2014a. (cited on page 52)

Yu, F. X.; Kumar, S.; Jebara, T.; and Chang, S. F., On learning with label proportions. *CoRR*, abs/1402.5902 (2014). (cited on pages 51, 61, 65, 103, and 167)

Yu, F. X.; Liu, D.; Kumar, S.; Jebara, T.; and Chang, S. F., $\propto$SVM for Learning with Label Proportions. In *ICML*, 2013. (cited on pages viii, 18, 61, 63, 64, 65, 90, 91, and 102)

Zantedeschi, V.; Emonet, R.; and Sebban, M., beta-risk: a new surrogate risk for learning from weakly labeled data. In *NIPS*, 2016. (cited on pages 18 and 49)

Zhang, B.; Estrada, T.; Cicotti, P.; Balaji, P.; and Taufer, M., Accurate scoring of drug conformations at the extreme scale. In *International Symposium on Cluster, Cloud and Grid Computing*, 2015. (cited on page 138)

Zhou, Z.-H. and Xu, J.-M., On the relation between multi-instance learning and semi-supervised learning. In *ICML*, 2007. (cited on page 16)

Zhu, X. and Ghahramani, Z., Learning from labeled and unlabeled data with label propagation. Technical report, CMU-CALD-02-107, 2002. (cited on page 17)