Learning from networked examples in a k-partite graph[*]

Yuyi Wang[1], Jan Ramon[1], et Zheng-Chu Guo[2]

[1]KULeuven, Belgium
[2]University of Exeter, UK

June 4, 2013

Abstract

Many machine learning algorithms are based on the assumption that training examples are drawn independently. However, this assumption does not hold anymore when learning from a networked examples, i.e. examples sharing pieces of information (such as vertices or edges). We propose an efficient weighting method for learning from networked examples and show a sample error bound which is better than previous work.

Keywords: Learning theory, Networked examples, Non-independent sample, Sample error, Generalization bound.

1 Introduction

In supervised learning, a labeled training sample for learning takes the form $\mathbf{Z} = \{\mathbf{z}_i\}_{i=1}^m$ with $\mathbf{z}_i = (\mathbf{x}_i, y_i) \in (\mathcal{X} \times \mathcal{Y})$ where $\mathcal{X}$ is called the feature space or the input space, and $\mathcal{Y}$ is called the label space or the output space. A standard assumption is that the training examples $\mathbf{z}_i$ from $\mathbf{Z}$ are drawn independently and identically (i.i.d.) from a probability distribution ρ on $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$.

In this paper, we consider a setting where examples share part of their features. The i.i.d. assumption does not always hold in this setting. For instance, suppose that we are interested in predicting whether a given person likes a given movie. We could ask a set of persons to grade five of the movies they have seen in the past. Then, we want to predict for a new visitor (drawn from the same distribution as our training persons) whether he will like a newly introduced movie (having features drawn from the same distribution as the movies in the past). Our training examples (containing a person ID, a movie ID and a grade) are not all independent since each person graded several movies and all movies were graded by several persons. Still, we would like to get a generalization guarantee.

A naive method would be to ignore the problem and treat the training examples as independent examples. The result in [Jan04] showed that a larger number of possibly non-independent examples would not necessarily mean that a more accurate model can be constructed.

Another straightforward method is to first find a subset of the training examples which are independent and then learn from these. Though we can directly use existing results to bound the sample error of this approach, it is inherently difficult to find a sufficiently large independent set of training examples. Moreover, we will show that possibly not all available information is used and the solution is suboptimal.

In this paper, we propose a novel approach to learn from networked examples. In our method, we first compute nonnegative weights for all training examples. Using these weighted examples, we show that we can get better bounds of the sample error than the two methods above. It is an advantage that the weights of the examples are efficiently computable.

The remainder of this paper is organized as follows. We introduce networked training examples in Section 2. We review some basic concepts of statistical learning theory in Section 3. The related work is discussed in Section 4, and this section mainly gives the sample error bounds of learning from networked data if we treat the data as i.i.d.. In Section 5, we consider the above-mentioned method that first select a set of independent training examples. In Section 6, we propose our network learning method. We derive weighted inequalities in Section 6.2, and they are used in Section 6.3 to estimate the sample error of the ERM algorithms with networked training examples. Section 7 concludes

[*]A longer version of this paper appears at http://arxiv.org/abs/1306.0393 [arXiv:1306.0393].

this paper with a summary of our contributions and a discussion of future work.

2 Problem Statement

Before discussing our method, we first give a formal problem statement.

2.1 The network

In this paper, we use a k-partite hypergraph $G = (V, E, \mathcal{X}, \mathcal{Y}, \phi)$ to represent the network which induces all the training examples. The set of vertices V is partitioned into k disjoint sets $V^{(1)}, V^{(2)}, \ldots, V^{(k)}$, and each hyperedge $e \in E \subseteq V^{(1)} \times V^{(2)} \times \cdots \times V^{(k)}$ intersects every set of the partition in exact one vertex. The number of hyperedges is denoted by m, and the cardinality of a partition $V^{(i)}$ is denoted by n_i, i.e., $|E| = m$ and $|V^{(i)}| = n_i$. The j-th vertex of the partition $V^{(i)}$ is denoted by $v^{(i,j)}$ where $1 \leq j \leq n_i$. We denote the i-th component of an edge e as $e^{(i)}$, which is a vertex in $V^{(i)}$. Two edges e_a and e_b *overlap* if and only if there exists $1 \leq i \leq k$ such that $e_a^{(i)} = e_b^{(i)}$.

For instance, in our movie rating example we would have a vertex set $V^{(1)}$ of movies, a vertex set $V^{(2)}$ of persons (watching movies) and a set $V^{(3)}$ of movie ratings. Hyperedges would be triple $(m, p, r) \in V^{(1)} \times V^{(2)} \times V^{(3)}$ of a movie, a person and the rating this person gave to that movie.

2.2 Features

Let $\mathcal{X} = \mathcal{X}^{(1)} \times \cdots \times \mathcal{X}^{(k)}$ be a k-dimensional compact metric space. Let $\phi : \bigcup_{1 \leq i \leq k} (V^{(i)} \mapsto \mathcal{X}^{(i)})$ be a function on the vertex set V assigning to every vertex $v^{(i,j)}$ in $V^{(i)}$ a feature $\phi(v^{(i,j)}) = x^{(i,j)}$ drawn independently and identically from a *fixed but unknown* distribution ρ_i. We will call $\phi(v^{(i,j)})$ a feature, even though it may be a compound object such as a vector. We also use the notation ϕ as a function on hyperedges. For any hyperedge e, we call $\phi(e) = [\phi(e^{(1)}), \phi(e^{(2)}), \ldots, \phi(e^{(k)})]$ the feature vector of e.

For instance, in our movie rating example, ϕ could assign to movies $m \in V^{(1)}$ pairs $(genre, length)$, to persons $p \in V^{(2)}$ a triple $(gender, age, nationality)$ and to a rating $r \in V^{(3)}$ a pair $(watching\_time, movie\_version)$. ϕ would therefore assign to every hyperedge a triple containing in total 8 values.

2.3 Examples

Every hyperedge e_i in E induces an example $\mathbf{z}_i = (\mathbf{x}_i, y_i) \in \mathcal{Z} = \mathcal{X} \times \mathcal{Y}$. The feature vector of this example is $\mathbf{x}_i = \phi(e_i)$. We will use $\mathbf{x}_i^{(j)}$ to denote the j-th component of the feature vector $\mathbf{x}_i$. If $e_i^{(j)} = v^{(j,l)}$ where $1 \leq l \leq n_j$, then the j-th component of the feature of the training example $\mathbf{z}_i$ is $x^{(j,l)}$. Thus, $\mathbf{x}_i^{(j)} = x^{(j,l)} = \phi(v^{(j,l)})$. We can see that if two hyperedges overlap, then the two corresponding examples are not independent (they share part of their features, and hence drawing the one example puts restrictions on the drawing of the other example). Given the features $\mathbf{x}_i$ of this example, the label y_i follows a fixed but unknown probability distribution $\rho_{y|\mathbf{x}}$. We can then write $\rho(\mathbf{x}, y) = \rho_{y|\mathbf{x}}(\mathbf{x}, y)\rho_{\mathbf{x}}(\mathbf{x})$. The training dataset derived from G is denoted by $\mathbf{Z} = \{\mathbf{z}_i | e_i \in E\}$, and it is called a *G-networked sample*. The size of the sample $\mathbf{Z}$ is the same as the number of hyperedges, so $|\mathbf{Z}| = m$.

2.4 Independence assumption

We make the following assumptions:

- As in the traditional form of PAC-learning, the feature of every vertex in the partitions V_i is drawn identically and independently from ρ_i.
- Especially, these features are independent from the edges in which they participate, i.e., $\rho_i(x^{(i,l)}) = \rho_i(x^{(i,l)} | E(G))$.
- Moreover, all hyperedges (examples) get a target value drawn identically and independently from $\rho_{y|\mathbf{x}}$. Even if the hyperedges share vertices, still there target value is sampled i.i.d. from $\rho_{y|\mathbf{x}}$ based on their (possibly identical) feature vector.
- One can choose freely which vertices participate in which hyperedges, and which edges belong to the training set and the test set, as long as this hyperedge and training set selection process is completely independent from the drawing of features for the vertices and the drawing of target values.

From the above assumptions, we can infer that $\rho_{\mathbf{x}}(\mathbf{x}) = \prod_{i=1}^{k} \rho_i(\mathbf{x}^{(i)})$. Our analysis of the sample error holds no matter what the distributions ρ_i and $\rho_{y|\mathbf{x}}$ are, as long as the above assumptions hold.

It is possible that the empirical distribution of the training and/or test set deviate from ρ, but we will show that we can bound the extent to which this is possible based on the assumptions.

In our movie rating example, it may or may not be realistic that these assumptions hold. In particular, if ratings are obtained from visitors of a cinema, then probably some visitors will already have a preference and will not choose movies randomly. On the other hand, if ratings are obtained during an experiment or movie contest where a number of participants or jury members are asked to watch a specific list of movies, one could randomize the movies to increase fairness, and in this way our assumptions would be satisfied.

3 Preliminaries

In this section, we review some basic concepts of statistical learning theory when the training sample $\mathbf{Z}$ is i.i.d.. These concepts will be used in following sections.

3.1 Learning task

The main goal of supervised learning is to learn a function $f : \mathcal{X} \mapsto \mathcal{Y}$ from training examples $\mathbf{Z}$ to predict a label y of an unseen point $\mathbf{x}$. For convenience, we assume $\mathcal{Y} = \mathbb{R}$. We define a loss function $L : \mathcal{Y} \times \mathcal{Y} \mapsto \mathbb{R}_+$ to measure the prediction errors. The function value $L(f(\mathbf{x}), y)$ denotes the local error suffered from the use of f to produce y from $\mathbf{x}$. We average the local error over all pairs $(\mathbf{x}, y)$ by integrating over $\mathcal{Z}$ with respect to ρ. A natural idea is to find the minimizer of the *expected risk*

$$\mathcal{E}^L(f) = \int_{\mathcal{Z}} L(f(\mathbf{x}), y)\rho(\mathbf{x}, y)\mathrm{d}(\mathbf{x}, y).$$

Then the target function we want to learn is defined as

$$f^L_\rho = \arg\min \mathcal{E}^L(f),$$

where the minimization is taken over the set of all measurable functions. Unfortunately, the probability distribution ρ is unknown, f^L_ρ can not be computed directly. If every example in $\mathbf{Z}$ is independent from each other, by the law of large numbers, as the sample size m tends to infinity, the *empirical risk*

$$\mathcal{E}^L_{\mathbf{Z}}(f) = \frac{1}{m}\sum_{i=1}^{m} L(f(\mathbf{x}_i), y_i)$$

converges to the expected risk $\mathcal{E}^L(f)$. Then we may get a good candidate $f^L_{\mathbf{Z}}$ to approximate the target function f^L_ρ, where

$$f^L_{\mathbf{Z}} = \arg\min \mathcal{E}^L_{\mathbf{Z}}(f).$$

3.2 Empirical risk minimization principle

In order to avoid over-fitting, we will not take the minimization of the empirical risk over all the measurable functions. The main idea of the empirical risk minimization principle is to find the minimizer in a properly selected hypothesis space $\mathcal{H}$, i.e.,

$$f^L_{\mathbf{Z},\mathcal{H}} = \arg\min_{f\in\mathcal{H}} \mathcal{E}^L_{\mathbf{Z}}(f).$$

The hypothesis space $\mathcal{H}$ is usually chosen as a subset of $\mathcal{C}(\mathcal{X})$ which is the Banach space of continuous functions on a compact metric space $\mathcal{X}$ with the norm $\|f\|_\infty = \sup_{\mathbf{x}\in\mathcal{X}} |f(\mathbf{x})|$.

The performance of the ERM approach is evaluated in terms of the *excess risk*

$$\mathcal{E}^L(f^L_{\mathbf{Z},\mathcal{H}}) - \mathcal{E}^L(f^L_\rho).$$

If we define

$$f^L_{\mathcal{H}} = \arg\min_{f\in\mathcal{H}} \mathcal{E}^L(f),$$

then the excess risk can be decomposed as

$$\mathcal{E}^L(f^L_{\mathbf{Z},\mathcal{H}}) - \mathcal{E}^L(f^L_\rho) = [\mathcal{E}^L(f^L_{\mathbf{Z},\mathcal{H}}) - \mathcal{E}^L(f^L_{\mathcal{H}})] + [\mathcal{E}^L(f^L_{\mathcal{H}}) - \mathcal{E}^L(f^L_\rho)].$$

We call the first part $\mathcal{E}^L(f^L_{\mathbf{Z}}) - \mathcal{E}^L(f^L_{\mathcal{H}})$ the *sample error*, the second part $\mathcal{E}^L(f^L_{\mathcal{H}}) - \mathcal{E}^L(f^L_\rho)$ the *approximation error*. The approximation error is independent of the sample and it is well studied in [CZ07]. In this paper, we concentrate on the sample error.

Another challenge about the ERM approach is how to choose a proper hypothesis space. Intuitively, a small hypothesis space brings a large approximation error, while large hypothesis space results in over-fitting. Hence the hypothesis space must be chosen to be not too large or too small. It is closely related to the bias-variance problem (see, e.g., Section 1.5 of [CZ07]). In learning theory, the complexity of the hypothesis space is usually measured in terms of covering number, entropy number, VC-dimension, etc. In this paper, we will use the covering numbers defined below to measure the capacity of our hypothesis space $\mathcal{H}$.

In this paper, we focus on the ERM approach associated with the least square loss function, that is $L(f(\mathbf{x}), y) = (f(\mathbf{x}) - y)^2$. Note that our analysis can easily be extended to general loss functions case.

3.3 Estimating the sample error

For the sake of conciseness, we denote $f^L_{\mathbf{Z},\mathcal{H}}$, $f^L_{\mathcal{H}}$, $\mathcal{E}^L_{\mathbf{Z}}(f)$ and $\mathcal{E}^L(f)$ as $f_{\mathbf{Z}}$, $f_{\mathcal{H}}$, $\mathcal{E}_{\mathbf{Z}}(f)$ and $\mathcal{E}(f)$ respectively. Now we are in a position to estimate the sample error $\mathcal{E}_{\mathcal{H}}(f_{\mathbf{Z}})$. The definition of $f_{\mathbf{Z}}$ tells us that

$\mathcal{E}_{\mathbf{Z}}(f_{\mathbf{Z}}) - \mathcal{E}_{\mathbf{Z}}(f_{\mathcal{H}}) \leq 0$, therefore the sample error can be decomposed as

$$\begin{aligned}\mathcal{E}_{\mathcal{H}}(f_{\mathbf{Z}}) &= \mathcal{E}(f_{\mathbf{Z}}) - \mathcal{E}(f_{\mathcal{H}}) = [\mathcal{E}(f_{\mathbf{Z}}) - \mathcal{E}_{\mathbf{Z}}(f_{\mathbf{Z}})] \\ &+[\mathcal{E}_{\mathbf{Z}}(f_{\mathbf{Z}}) - \mathcal{E}_{\mathbf{Z}}(f_{\mathcal{H}})] + [\mathcal{E}_{\mathbf{Z}}(f_{\mathcal{H}}) - \mathcal{E}(f_{\mathcal{H}})] \\ &\leq [\mathcal{E}(f_{\mathbf{Z}}) - \mathcal{E}_{\mathbf{Z}}(f_{\mathbf{Z}})] + [\mathcal{E}_{\mathbf{Z}}(f_{\mathcal{H}}) - \mathcal{E}(f_{\mathcal{H}})].\end{aligned}$$

Before stating the results, we first introduce some notations and definitions.

Definition 1. *Let S be a metric space and $\tau > 0$. We define the covering number $\mathcal{N}(S, \tau)$ to be the minimal $\ell \in \mathbb{N}$ such that there exists ℓ disks in S with radius τ covering S. When S is compact, this number is finite.*

Definition 2. *Let $M > 0$ and ρ be a probability distribution on $\mathcal{Z}$. We say that a set $\mathcal{H}$ of functions from $\mathcal{X}$ to $\mathcal{Y}$ is M-bounded when*

$$\sup_{f \in \mathcal{H}} |f(\mathbf{x}) - y| \leq M$$

holds almost everywhere on $\mathcal{Z}$.

To bound the sample error, the Bernstein inequality is used [**?**].

Theorem 1. *Let $\mathbf{Z}$ be an i.i.d sample and ξ be a function defined on the space $\mathcal{Z}$ with mean $\mathbf{E}(\xi) = \mu$, variance $\sigma^2(\xi) = \sigma^2$ and satisfying $|\xi(\mathbf{z}) - \mu| \leq M$ for almost all $\mathbf{z} \in \mathcal{Z}$. Then for all $\epsilon > 0$,*

$$\Pr\left(\frac{1}{m}\sum_i \xi(\mathbf{z}_i) - \mu \geq \epsilon\right) \leq \exp\left(-\frac{m\epsilon^2}{2(\sigma^2 + \frac{1}{3}M\epsilon)}\right).$$

We estimate the sample error by the above concentration inequality. In this paper, we omit the details of the proof and directly quote the following result from [CZ07].

Theorem 2. *Let $\mathcal{H}$ be a compact and convex subset of $\mathcal{C}(\mathcal{X})$. If $\mathcal{H}$ is M-bounded, then for all $\epsilon > 0$,*

$$\Pr\left(\mathcal{E}_{\mathcal{H}}(f_{\mathbf{Z}}) \geq \epsilon\right) \leq \mathcal{N}\Big(\mathcal{H}, \frac{\epsilon}{12M}\Big) \exp\Big(-\frac{m\epsilon}{300M^4}\Big).$$

4 Related work

In this section, we discuss the related work.

4.1 Dependency graphs

As described in [Jan04], a *dependency graph* can be used to represent the relationship between the training examples in $\mathbf{Z}$. The vertices of the dependency graph $\mathit{\Gamma}$ are the hyperedges in G, that is, $V(\mathit{\Gamma}) = E(G)$. Thus, the vertices in the dependency graph also represent training examples in $\mathbf{Z}$. Two vertices are adjacent if the corresponding two hyperedges overlap, i.e., if two hyperedges e_a and e_b in $E(G)$ satisfy that there exists j such that $e_a^{(j)} = e_b^{(j)}$, then the two vertices e_a and e_b are adjacent in $\mathit{\Gamma}$ and the induced examples $\mathbf{z}_a$ and $\mathbf{z}_b$ are not independent.

4.2 The chromatic-number bound

In [Jan04], the author shows an inequality which can be used to bound the error on averaging a function over networked sample.

Theorem 3. *Let $\mathbf{Z}$ be a G-networked sample and ξ be a function defined on the space $\mathcal{Z}$ with mean $\mathbf{E}(\xi) = \mu$, and satisfying $|\xi(\mathbf{z}) - \mu| \leq M$ for almost all $\mathbf{z} \in \mathcal{Z}$. Then for all $\epsilon > 0$,*

$$\Pr\left(\frac{1}{m}\sum_i \xi(\mathbf{z}_i) - \mu \geq \epsilon\right) \leq \exp\left(-\frac{8m\epsilon^2}{25\chi^*(\mathit{\Gamma})(\sigma^2 + M\epsilon/3)}\right),$$

where $\chi^*(\mathit{\Gamma})$ is the fractional chromatic number of the dependency graph.

Let us now consider a learning strategy we call EQW (EQual Weight) and which learns from a set of networked examples in the same way as if they were i.i.d. (i.e. without weighting them as a function of the network structure). We can use Theorem 3 above to bound the sample error of EQW:

Theorem 4. *Let $\mathcal{H}$ be a compact and convex subset of $\mathcal{C}(\mathcal{X})$, and $\mathbf{Z}$ be a G-networked sample. If $\mathcal{H}$ is M-bounded, then for all $\epsilon > 0$,*

$$\Pr\left(\mathcal{E}_{\mathcal{H}}(f_{\mathbf{Z}}) \geq \epsilon\right) \leq \mathcal{N}\Big(\mathcal{H}, \frac{\epsilon}{12M}\Big) \exp\Big(-\frac{3m\epsilon}{1400\chi^*(\mathit{\Gamma})M^4}\Big).$$

The result above shows that the bound of the sample error does not only rely on the sample size but also the fractional chromatic number of the dependency graph. That is, a larger sample may result in a poorer sample error bound since $\chi^*(\mathit{\Gamma})$ can also become larger.

4.3 Mixing conditions

There is also some literature on learning from a sequence of examples where examples closeby in the sequence are dependent. In the community of machine learning, mixing conditions are usually used to quantify the dependence of sample points and are usually used in time series analysis. For example, in [GS11], the learning performance of a regularized classification algorithm using a non-i.i.d. sample is investigated, where the independence restriction is relaxed to so-called α-mixing or β-mixing conditions. In [SW10], regularized least square regression with dependent samples is considered under the assumption that the training sample satisfies some mixing conditions. In [MM96], the authors established a Bernstein type inequality is presented for stationary exponentially α-mixing processes, which is based on the effective number (less than the sample size). Our Bernstein type inequalities for dependent network data too assigns weights to examples. However, the assumptions for the the training sample are different, and the main techniques are distinct. Moreover, in practice, it is not easy to check whether the training sample satisfies the mixing conditions. Our networked training examples certainly do not satisfy any of these mixing conditions. We refer interested readers to [Bra05] and references therein for more details about the mixing conditions.

5 Selecting an independent subset of training examples

A straightforward idea to learn from a G-networked sample $\mathbf{Z}$ is to find a subset $\mathbf{Z}_I \subseteq \mathbf{Z}$ of training examples which correspond to non-overlapping hyperedges. Due to our assumptions, such set will be an i.i.d. sample. We can then perform algorithms on $\mathbf{Z}_I$ for learning. We call this method the IND method. To bound the sample error of this method, we can directly use the result in Section 3.

The key step of the IND method is to find a large $\mathbf{Z}_I$. The larger $|\mathbf{Z}_I|$ is, the higher will be the expected accuracy of $f_{\mathbf{Z}_I}$. If two hyperedges e_a and e_b in G do not share any vertex, i.e., $e_a^{(i)} \neq e_b^{(i)}$ for all $1 \leq i \leq k$, the two induced examples $\mathbf{z}_a$ and $\mathbf{z}_b$ are independent. Therefore, finding a subset $\mathbf{Z}_I$ from the training dataset $\mathbf{Z}$, is equivalent to finding an independent set in the dependency graph Γ, and is also equivalent to finding a hypergraph matching in G.

For any dependency graph Γ, it holds that (see, e.g., [Die10]),

$$\frac{m}{\chi^*(\Gamma)} \leq \alpha(\Gamma)$$

where α is the independence number. If we can find a maximum independent set of the dependency graph Γ, then the bound of the IND method will be better than that of the EQW method.

However, It is NP-hard to find a maximum independent set in Γ or equivalently to find a maximum matching in G when $k \geq 3$ [GJ79]. Therefore, the IND method is not effective in practice since it is difficult to find a large independent set of networked examples.

6 A weighting method

In this section, we propose a computationally efficient method based on a weighting strategy. It allows for a better bound of the sample error than the IND and EQW methods.

6.1 Feasible weighting

Given a hypergraph G, we weight every hyperedge e_i with a nonnegative value w_i. We use the notation w_F to denote the sum $\sum_{i \in F} w_i$ over a set of indices $F \subseteq \{1, \ldots, n\}$ of hyperedges, and $\eta(v)$ to denote the set of indices of hyperedges incident on a vertex $v \in V$. We say that $\mathbf{w} = [w_1, \ldots, w_n]$ is a *feasible weighting* of a hypergraph G if for all i it holds that $w_i \geq 0$ and for all $v \in V$ it holds that $w_{\eta(v)} \leq 1$.

For a hypergraph G, its s-value is defined as follows:

$$\mathsf{s}(G) = \max_{\mathbf{w}} \left\{ \sum_{e_i \in E} w_i : \mathbf{w} \text{ is a feasible weighting for } G \right\}$$

Notice that these constraints form a linear program on $\mathbf{w}$. We call a $\mathbf{w}$ which makes the linear program maximal an *optimal weighting*. There exist efficient methods to solve the linear program formed by the above-mentioned constraints and hence compute an optimal weighting and the s-value, e.g., interior point methods [BV04]. An optimal weighting can be considered as a fractional maximum hypergraph matching [Lov75, CL12]. One can show that the value $\mathsf{s}(G)$ is always greater than or equal to the size of a maximum hypergraph matching in any hypergraph G.

For a G-networked sample $\mathbf{Z}$, we denote the weighted sample $\mathbf{Z}_{\mathsf{s}} = \{(\mathbf{x}_i, y_i, w_i)\}$ where $[w_1, \ldots, w_n]$ is an optimal weighting. Now we can define a new empirical risk on the weighted sample $\mathbf{Z}_\mathsf{s}$ by

$$\mathcal{E}_\mathsf{s}(f) = \frac{1}{\mathsf{s}} \sum_{i=1}^{n} w_i (f(\mathbf{x}_i) - y_i)^2.$$

In the following, we will show the sample error bound of an ERM approach with $\mathbf{Z}_\mathsf{s}$.

6.2 Exponential inequalities

In Section 3, the Bernstein inequality is used to estimate the sample error. A key property used for proving the Bernstein inequality is that all observations are independent. That is, if $\xi_1, \ldots, \xi_m$ are independent random variables, then

$$\mathbf{E} \exp\left(\sum_{i=1}^{m} \xi_i\right) = \prod_{i=1}^{m} \mathbf{E} e^{\xi_i}.$$

However, when learning from networked training examples, the equality can not be used.

The following inequality is an analogue to the Bernstein inequality. The inequality will be used later to estimate the sample error for a networked sample.

Theorem 5. *Let* $\mathbf{Z}$ *be a* G*-networked sample and* ξ *be a function defined on the space* $\mathcal{Z}$ *with mean* $\mathbf{E}(\xi) = \mu$*, variance* $\sigma^2(\xi) = \sigma^2$*, and satisfying* $|\xi(\mathbf{z}) - \mu| \leq M$ *for almost all* $\mathbf{z} \in \mathcal{Z}$*. If* $\mathbf{w}$ *is an optimal weighting of* G*, then for all* $\epsilon > 0$*,*

$$\Pr\left(\frac{1}{\mathsf{s}} \sum_i w_i \xi(\mathbf{z}_i) - \mu \geq \epsilon\right) \leq \exp\left(-\frac{\mathsf{s}\epsilon^2}{2(\sigma^2 + \frac{1}{3}M\epsilon)}\right).$$

6.3 An ERM approach with $\mathbf{Z}_\mathsf{s}$

In this section, we consider the ERM approach associated with $\mathbf{Z}_\mathsf{s}$. As discussed in section 3.2, the ERM approach aims to find a minimizer of the empirical risk in a proper hypothesis space $\mathcal{H}$ to approximate the target function, i.e.,

$$f_{\mathbf{Z}_\mathsf{s}} = \arg\min_{f \in \mathcal{H}} \mathcal{E}_\mathsf{s}(f).$$

Recall the empirical risk with $\mathbf{Z}_\mathsf{s}$ takes the form $\mathcal{E}_\mathsf{s}(f) = \frac{1}{\mathsf{s}} \sum_{i=1}^{n} w_i (f(\mathbf{x}_i) - y_i)^2$, and the expected risk $\mathcal{E}(f) = \int (f(\mathbf{x}) - y)^2 \rho(\mathbf{x}, y) \mathrm{d}(\mathbf{x}, y)$. As mentioned in section 3.2, the target function is the minimizer of the expected risk $\mathcal{E}(f)$, one can easily see that the target function takes the form [CZ07]

$$f_\rho(\mathbf{x}) = \int y \rho_{y|\mathbf{x}}(\mathbf{x}, y) \mathrm{d}(\mathbf{x}, y).$$

Then the performance of the ERM approach is measured by the excess risk

$$\mathcal{E}(f_{\mathbf{Z}_\mathsf{s}}) - \mathcal{E}(f_\rho).$$

Recall the definition $f_\mathcal{H} = \arg\min_{f \in \mathcal{H}} \mathcal{E}(f)$, the excess risk can be divided into two parts (sample error and approximation error) as follows

$$\mathcal{E}(f_{\mathbf{Z}_\mathsf{s}}) - \mathcal{E}(f_\rho) = [\mathcal{E}(f_{\mathbf{Z}_\mathsf{s}}) - \mathcal{E}(f_\mathcal{H})] + [\mathcal{E}(f_\mathcal{H}) - \mathcal{E}(f_\rho)].$$

Notice that the approximation error $\mathcal{E}(f_\mathcal{H}) - \mathcal{E}(f_\rho)$ is independent of the sample $\mathbf{Z}_\mathsf{s}$, and the approximation error vanishes if $f_\rho \in \mathcal{H}$.

In this section, we focus on the sample error $\mathcal{E}_\mathcal{H}(f_{\mathbf{Z}_\mathsf{s}}) := \mathcal{E}(f_{\mathbf{Z}_\mathsf{s}}) - \mathcal{E}(f_\mathcal{H})$.

The following is our main result.

Theorem 6. *Let* $\mathcal{H}$ *be a compact and convex subset of* $\mathcal{C}(\mathcal{X})$*. If* $\mathcal{H}$ *is a* M*-bounded, then for all* $\epsilon > 0$*,*

$$\Pr\left(\mathcal{E}_\mathcal{H}(f_{\mathbf{Z}_\mathsf{s}}) \geq \epsilon\right) \leq \mathcal{N}\left(\mathcal{H}, \frac{\epsilon}{12M}\right) \exp\left(-\frac{\mathsf{s}\epsilon}{300M^4}\right).$$

Remark: In this paper, we mainly consider the ERM algorithm associated with networked samples to avoid over-fitting. Another way to deal with over-fitting is regularization, which is initially proposed to solve ill-posed phenomena induced in inverse problems, e.g., ill-conditioned matrix inversion problems. Similar results can also be obtained for the regularization algorithms by using the probability inequalities in section 6.2.

7 Conclusions

In this paper, we introduce the problem of learning from networked data. We first show that this may result in a poor sample error bound if we ignore the dependency relationship between the examples. We then analyze a method where first a set of i.i.d. examples is selected. Existing theoretical results can be directly used for this method, but it is difficult to find a large set of independent examples. We propose a novel method which is a weighting strategy with efficiently computable weights. To assess learning algorithms on these weighted examples, we show a Bernstein-type statistical inequality. Using this inequalitiy, we can estimate the sample error. We show that this bound is better than existing alternatives.

In future, we want to consider settings where we do not make the strong independence assumption that the occurrences of the hyperedges are independent of the features of the vertices. A first step in this direction would be to develop a measure to assess the strength of the dependency of the hyperedges on the features of the vertices and its influence on the learning task at hand.

Acknowledgements

The first author and the second author are supported by ERC Starting Grant 240186 "MiGraNT: Mining Graphs and Networks: a Theory-based approach". The third author is supported by the EPSRC under grant EP/J001384/1.

References

[Ben62] George Bennett. Probability inequalities for the sum of independent random variables. *Journal of the American Statistical Association*, 57.297:33–45, 1962.

[Bra05] Richard C. Bradley. Basic properties of strong mixing conditions, a survey and some open questions. *Probability Surveys*, 2:107–144, 2005. 5

[BV04] Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge University Press, 2004. 5

[CL12] YukHei Chan and LapChi Lau. On linear and semidefinite programming relaxations for hypergraph matching. *Mathematical Programming*, 135(1-2):123–148, 2012. 5

[CZ07] Felipe Cucker and Ding-Xuan Zhou. *Learning theory: an approximation theory viewpoint*. Cambridge University Press, 2007. 3, 4, 6

[Die10] Reinhard Diestel. *Graph theory*. Springer-Verlag, 2010. 5

[GJ79] Michael R. Garey and David S. Johnson. *Computers and intractibility, a guide to the theory of NP-Completeness*. W. H. Freeman Company, 1979. 5

[GS11] Zheng-Chu Guo and Lei Shi. Classification with non-iid sampling. *Mathematical and Computer Modelling*, 54.5:1347–1364, 2011. 5

[Hoe63] Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American statistical association*, 58.301:13–30, 1963.

[Jan04] Svante Janson. Large deviations for sums of partly dependent random variables. *Random Structures & Algorithms*, 24.3:234–248, 2004. 1, 4

[Lov75] László Lovász. On the ratio of optimal integral and fractional covers. *Discrete mathematics*, 13.4:383–390, 1975. 5

[MM96] Dharmendra S. Modha and Elias Masry. Minimum complexity regression estimation with weakly dependent observations. *Information Theory, IEEE Transactions on*, 42.6:2133–2145, 1996. 5

[SW10] Hongwei Sun and Qiang Wu. Regularized least square regression with dependent samples. *Advances in Computational Mathematics*, 32.2:175–189, 2010. 5

[UrAG06] Nicolas Usunier, Massih reza Amini, and Patrick Gallinari. Generalization error bounds for classifiers trained with interdependent data. In *Advances in Neural Information Processing Systems 18 (NIPS 2005)*, pages 1369–1376. MIT Press, 2006.