Learning from networked examples

Yuyi Wang
Department of Computer Science
KU Leuven
Heverlee 3001, Belgium

Jan Ramon
Department of Computer Science
KU Leuven
Heverlee 3001, Belgium

Zheng-Chu Guo
Department of Mathematics,
Zhejiang University
Hangzhou 310027, P.R. China

Abstract

Many machine learning algorithms are based on the assumption that training examples are drawn identically and independently. However, this assumption does not hold anymore when learning from a networked sample because two or more training examples may share some common objects, and hence share the features of these shared objects. We first show that the classic approach of ignoring this problem potentially can have a disastrous effect on the accuracy of statistics, and then consider alternatives. One of these is to only use independent examples, discarding other information. However, this is clearly suboptimal. We analyze sample error bounds in a networked setting, providing both improved and new results. Next, we propose an efficient weighting method which achieves a better sample error bound than those of previous methods. Our approach is based on novel concentration inequalities for networked variables.

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

1. Introduction

Recently, there has been an increasing interest in network-structured data, such as the data in social networks, economic networks, citation networks and chemical interaction networks. An important challenge is that data concerning related objects can’t be assumed to be independent. More precisely, an important assumption made by many approaches in the field of statistics and machine learning is that observations are drawn independently and identically (i.i.d.) from a fixed distribution. However, this assumption does not hold for observations extracted from the same network.

Many practical approaches to supervised learning in networks ignore (at least partially) the problem and learn models with classic machine learning techniques. While these work to some extent, they are not supported by a well-developed theory such as the one which
provides generalization guarantees for the i.i.d. case as a function of the number of training examples. As a consequence, one may miss opportunities to learn due to the numerous dependencies between the training examples.

In this paper, we make a step towards addressing this problem. Our contribution is fourfold.

• First, we introduce a framework for explicitly representing dependencies between examples. Such framework allows for encoding domain knowledge and can form a basis for studies preparing a machine learning effort on networked data.

• Second, we introduce a relaxed i.i.d. assumption for networked examples. Even though not valid in general, our relaxed independence assumption is elegant and significantly generalizes over classic i.i.d. assumptions.

• Third, we show concentration inequalities for networked examples and extend existing strategies for generalization guarantees to networked examples.

• Finally, we analyze several alternative strategies for learning from networked examples. In the process, we improve on an earlier result by Usunier et al. (2006). We propose a new example-weighting strategy, and show that is better than naive approaches which ignore the dependencies between examples. We illustrate this on power-law graphs, which are classic models capturing important properties of real-world graphs.

The remainder of this paper is structured as follows.

We introduce networked examples and our learning task in Section 2. We derive concentration inequalities with networked random variables and study the variance of statistics with networked random variables in Section 3. In Section 4, we review some basic concepts of learning theory and study several example weighting schemes and their effect on learning bounds. In Section 5, we discuss existing work considering similar tasks and slightly improve the concentration bounds of the U statistics. Section 6 concludes this paper with a summary and discussion of our contributions and a discussion of future work.

2. Problem statement

In this section, we introduce networked examples and a framework to represent them. The basic intuition is that a networked example combines a number of objects, each of which can carry some information (features). However, each of these objects can be shared with other examples, such that we get a network. The sharing of information also makes explicit the dependencies between the examples. In Section 2.1 and 2.2, we define more formally networked examples. Next, in Section 2.3, we introduce our independence assumptions and formalize the supervised learning problem we will consider. In Section 2.4, we show our setting is equivalent to the problem of learning conditional probability distributions, a fundamental task in learning probabilistic logical models.

2.1 Networked examples

Before we introduce our model for networked examples, we first give the definition of hypergraphs.
Definition 1 (hypergraph) A hypergraph $G$ is a pair $G = (V, E)$ where $V$ is the set of vertices, $E \subseteq 2^V$ is a set of hyperedges, each containing zero or more vertices. If $G$ is a hypergraph, we will denote with $V(G)$ its vertex set and with $E(G)$ its edge set.

We will use a hypergraph $G = (V, E)$ to represent a set of networked examples. The vertices $V = \{v_1, v_2, \ldots, v_m\}$ will represent objects, and the hyperedges $E = \{e_1, e_2, \ldots, e_n\}$ will represent examples grouping a number of objects. In what follows, we will often abuse terminology, identify vertices and objects and use the terms interchangeably. We will consider two variants of networked examples: general networked examples and tuple networked examples. Tuple networked examples are special cases of general networked examples. The following running examples, which will be used throughout this paper, illustrate both types of networked examples.

Example 1 (club) Consider a dataset of clubs, i.e., every club is an example. A club is related to one or more themes and has two or more members. In our network representation, we have nodes for persons and for themes. A club is represented with a hyperedge containing the persons who are member of the club and the themes to which the club is related. Clubs can share members and/or themes.

The following example has been discussed in (Usunier et al., 2006).

Example 2 (binary classification) In a binary classification problem, we want to predict the relationship between some objects. For example, given two persons, a question is whether they are friends or not. In network analysis, this task is also called link prediction. We would have a vertex set $V$ of persons and every hyperedge contains two persons.

For problems with tuple networked examples, we can use a more special type of hypergraphs which are $k$-partite.

Definition 2 ($k$-partite hypergraph) A hypergraph $G = (V, E)$ is called $k$-partite if the set of vertices $V$ can be partitioned into $k$ disjoint sets, i.e., $V = V^{(1)} \cup \ldots \cup V^{(k)}$, and every hyperedge meets each $V^{(i)}$ exactly once, i.e., $\forall e \in E, \forall i : |V^{(i)} \cap e| = 1$.

Example 3 (court) In court, lawsuits involves litigants (clients), lawyers and judges. We want to analyze the satisfaction of clients in their lawsuits. We can construct a network with a vertex set $V^{(1)}$ of litigants, a vertex set $V^{(2)}$ of lawyers and a set $V^{(3)}$ of judges. The hyperedges (training examples) of interest are then tuples $(\text{litigant}, \text{lawyer}, \text{judge}) \in V^{(1)} \times V^{(2)} \times V^{(3)}$.

Example 4 (movie rating) Consider the problem of predicting the rating by a person who watched a movie in a particular cinema. Every training example contains three objects: a person, a movie and a cinema. Figure 1a illustrates this setup.
2.2 Features and target values

We will denote with $\mathcal{O}$ the space of all possible objects. Objects can be represented with feature vectors or other suitable data structures. As usual in supervised learning, in our setting each example has a target value (e.g. a class label or regression value). We will denote with $\mathcal{Y}$ the space of all possible target values.

**Example 5 (club)** Clubs have several members and several themes. Each member may be represented by a feature vector containing demographic data and a set of interests. Each theme may be described by a the Wikipedia page on the theme. Suppose we want to model how active each club is. Then we could attach to each club as target value the frequency of its meetings.

**Example 6 (binary classification)** In the problem of predicting whether two persons are friends, every person has his/her own feature vector (e.g. interests, demographical info, education, . . .). The target value of an example is a boolean value to show whether they are friends.

**Example 7 (movie rating)** In the movie rating example, a movie can be described by a feature vector containing genre, actor popularity, etc. A person who watches movies can be described by a feature vector containing amongst others gender, age and nationality. The venue can be described by the number of small and large theatres. The target value of an example is the rating the person gave to the movie in the concerned venue.

When representing a dataset, we will use a labeled hypergraph where vertices are labeled with the description of the objects they represent and the hyperedges are labeled with the target values of the examples they represent.

**Definition 3 (labeled hypergraph)** A labeled hypergraph is a tuple $(G, \Sigma_V, \phi, \Sigma_E, \lambda)$ where $G$ is a hypergraph, $\Sigma_V$ is a vertex label alphabet, $\Sigma_E$ is an edge label alphabet, $\phi : V(G) \to \Sigma_V$ is a vertex labeling function and $\lambda : E(G) \to \Sigma_E$ is an edge labeling function.
The two labeling functions defined above are used to assign features to every object and to assign a target value to every example. Therefore, we will use $\Sigma_V = \mathcal{O}$ and $\Sigma_E = \mathcal{Y}$.

2.3 Independence assumption

Though networked examples are not independent, we still need to assume some weaker form of independence of the examples. If we wouldn’t make any assumption, the dependence between examples could be so strong that they perfectly correlate (and hence are all identical). In such situation, it is not possible to make statistics nor to learn.

This paper is a first step relaxing the classical i.i.d. assumption. Our assumptions are still not fully general and are not satisfied in all applications, but we believe they make a useful first step. The idea we adopt here is that we explicitly model information shared by several examples and in this way also explicitly model the dependencies between the examples. We don’t model the dependencies in detail and our analysis works for any possible dependency of the examples on the shared information.

We consider a labeled hypergraph $(G, \mathcal{O}, \phi, \mathcal{Y}, \lambda)$, where the labels assigned by $\phi$ and $\lambda$ are drawn randomly from a probability distribution $\rho$. We make the following assumptions:

- The objects (and their features) assigned to vertices are independent from the hyperedges in which they participate, i.e., there is a function $\rho_x : \mathcal{O} \to [0, 1]$ such that for every $q \in \mathcal{O}$ and $v \in V(G)$, $\rho(\phi(v) = q) = \rho_x(q) = \rho_x(\phi(v) = q \mid E(G))$.

- Moreover, all hyperedges (examples) get a target value drawn identically and independently, i.e., there is a function $\rho_{y|x} : \mathcal{Y} \times \mathbb{N}^\mathcal{O} \to [0, 1]$ such that for all $e \in E(G)$, $\rho(\lambda(e) = y \mid \phi|_e) = \rho_{y|x}(y, \phi|_e) = \rho(\lambda(e) = y \mid \phi, E(G))$ (where $\phi|_e$ is $\phi$ restricted to $e$, i.e. $\phi|_e = \{(v, \phi(v)) \mid v \in e\}$). Even if the hyperedges share vertices, still their target values are sampled i.i.d. from $\rho_{y|x}$ based on their (possibly identical) feature vector.

- One can choose freely which vertices participate in which hyperedges, and which hyperedges 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 objects for the vertices and the drawing of target values for the hyperedges.

From the above assumptions, we can infer that

$$\rho(\phi, \lambda) = \prod_{v \in V(G)} \rho_x(\phi(v)) \prod_{e \in E(G)} \rho_{y|x}(\lambda(e), \phi|_e).$$

Our analysis holds no matter what the distributions $\rho$ is, as long as the above assumptions are met.

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

**Example 8** 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.

As a special case, we will often consider a \( k \)-partite setting. We can see this is a special case as follows. Let \((G, \mathcal{O}, \phi, \mathcal{Y}, \lambda)\) be a labeled \( k \)-partite hypergraph with \( V = \bigcup_{i=1}^{k} V^{(i)} \) the vertex partition. Let \( \mathcal{O} = \mathcal{O}_1 \times \ldots \times \mathcal{O}_k \) be the cartesian product of \( k \) feature spaces. Then, writing \( \rho_{y|x} \) as \( \rho_{y|x}(y, \phi|_e) = f(y, \phi(e \cap V^{(1)}), \ldots, \phi(e \cap V^{(k)})) \) for some function \( f : \mathcal{Y} \times \mathcal{O}_1 \times \ldots \times \mathcal{O}_k \rightarrow [0,1] \) ensures that for vertices of \( V^{(i)} \) only the part of the features from \( \mathcal{O}_i \) is relevant. Learning \( \rho_{y|x}(y, \phi|_e) \) is then equivalent to assigning (independently) to every vertex of \( V^{(i)} \) a set of features from \( \mathcal{O}_i \) and learning the function \( f : \mathcal{O}_1 \times \ldots \times \mathcal{O}_k \rightarrow \mathcal{Y} \).

### 2.4 A relation to learning probabilistic logical models

The problem described above is also equivalent to one of the fundamental tasks faced when learning directed probabilistic models such as Probabilistic Relational Models (Friedman et al., 1999), Logical Bayesian Networks (Fierens et al., 2005), Relational Bayesian Networks (Jaeger, 1997) and other directed models in the field of Statistical Relational Learning (Getoor and Taskar, 2007), namely learning the conditional probability function of a dependency template.

For instance, a dependency template may state that the rating a person gives to a movie depends on the interests of the person, the genre of the movie, the production cost of the movie and the size of the cinema theatre. A classic logic-based notation for such template(s) is

\[
\begin{align*}
\text{rating}(M, P, V) & \mid \text{interest}(P) \\
\text{rating}(M, P, V) & \mid \text{genre}(M) \\
\text{rating}(M, P, V) & \mid \text{cost}(M) \\
\text{rating}(M, P, V) & \mid \text{size}(V)
\end{align*}
\]

The semantics of such template is that for every instantiation (also called grounding) of the template (i.e. substitution of the variable \( M \) with a movie, of \( P \) with a person and of \( V \) with a cinema) the corresponding conditional probability function describes the dependency of the random variable representing the rating on the random variables representing the interests, genre, cost and size.

When we consider all ground dependencies, we get a ground Bayesian network, as illustrated in Figure 1b.

A classic approach to learn such conditional probability function is to collect a training set where every examples consists of the features of a specific grounding and the corresponding target value. Then, a classifier is learned using a supervised learning algorithm, e.g. a decision tree learner. However, often the grounding of the template share random variables, and in fact the problem is equivalent to the problem of learning from networked examples introduced above.

### 3. Concentration inequalities

Let \( f : \mathcal{O}^n \times \mathcal{Y} \rightarrow \mathbb{R} \) be a function. We may want to estimate the expected value \( \mu_f = \mathbb{E}_{z \sim \rho}[f(z)] \), for instance because it gives us some parameter we can use to model the dependency of hyperedge labels on the features of the incident vertices.


One typically approximates $\mathbb{E}_{z \sim \rho}[f(z)]$ by averaging the value of $f$ over an independent sample of the distribution, in particular, given a set $Z = \{z_i\}_{i=1}^n$ of instances $z_i$ drawn i.i.d. from $\rho$, we could estimate

$$
\hat{\mu}_f = \frac{1}{n} \sum_{i=1}^{n} f(z_i)
$$

(1)

In a networked setting, we don’t have an independent sample but rather a set of vertices $V$ with independently drawn features and a fixed hyperedge structure $E$. In such a case, what is the optimal way to combine the observations to get a good estimate? As we will see, the answer depends on the criterion one chooses to measure the quality of the estimate. We therefore first define a measure for concentration.

### 3.1 Concentration measures

**Definition 4** Let $g : \mathbb{R} \to \mathbb{R}_+$ be a function satisfying that if $x_1 > x_2 \geq 0$ or $x_1 < x_2 \leq 0$ then $g(x_1) \geq g(x_2)$. Then, we define the $g$-concentration of a random variable $X$ which follows a distribution $P$ to be

$$
L_g(X) = \int g(t - \mathbb{E}[X]) P(X = t) dt.
$$

$L_g(X)$ can be seen as a loss function measuring how well $X$ can be approximated by its mean. This generalizes a number of classical cases:

**Example 9** If we choose in the above definition $g(t) = t^2$ we get the variance

$$
L_2(X) = \int \left(t - \mathbb{E}[X]\right)^2 P(X = t) dt = \text{var}(X).
$$

**Example 10** If we choose in the above definition the step function $g(t) = I(t \geq a)$ where $I$ is the indicator function and $a$ is a constant, we get the probability that $X$ exceeds a (one-sided) deviation $a$ from its mean

$$
L_{I(\cdot \geq a)}(X) = P(X - \mathbb{E}[X] \geq a).
$$

With $g(t) = I(|t| \geq a)$ we get the probability of a deviation on any of both (positive or negative) side exceeding $a$.

In many cases, among which the classic concentration bounds such as the Markov inequality and Hoeffding inequality, one is interested in the worst case behavior over some space of probability distributions $\mathcal{D}$.

**Definition 5** Let $\mathcal{D}$ be a space of probability distributions, and $g : \mathbb{R} \to \mathbb{R}_+$ be a function satisfying that if $x_1 > x_2 \geq 0$ or $x_1 < x_2 \leq 0$ then $g(x_1) \geq g(x_2)$. We define a $g$-concentration of the distribution space $\mathcal{D}$ to be

$$
L_g(\mathcal{D}) = \sup_{D \in \mathcal{D}, X \sim D} L_g(X).
$$

7
3.2 Simple concentration bounds

**Definition 6 (concentration bound)** A concentration (upper) bound for a space of probability distributions $D_q$ parameterized by $q$ and a concentration measure $g_r$ parameterized by $r$ is a function $b(q, r)$ such that

\[ \forall q, r : L_{g_r}(D_q) \leq b(q, r). \]

**Example 11 (Markov inequality)** Consider that space $D_{EA}$ of all probability distributions such that random variables $X$ distributed according to such probability distribution satisfying $P(X < 0) = 0$ and $\mathbb{E}[X] = q$. Then $b_{\text{Markov}}(q, a) = \frac{q}{a+q}$ is a concentration bound for $D_{EA}$ and $L_{I(\cdot \geq a)}$, called Markov’s inequality, as

\[ \forall q, a : L_{I(\cdot \geq a)}(D_{EA}) = \sup_{X : \mathbb{E}[X] = q} P(X - \mathbb{E}[X] \geq a) \leq \frac{q}{a + q} = b_{\text{Markov}}(q, a). \]

A lot of research considers how much a sum (or average) of random variables of particular type are concentrated around their mean. This is especially useful when estimating a value by averaging over a sample as in Equation (1).

In this paper, we will consider the concentration behavior of weighted sums, i.e. given some vector of random variables $X = (X_i)_{i=1}^n$ and a weight vector $w \in \mathbb{R}^n_{+}$, we consider the weighted sum $w^\top X$ and its concentration behavior as measured with $L_g(w^\top X)$.

**Definition 7** If $D$ is a space of probability distributions and let $w \in \mathbb{R}^n_{+}$ be a vector of $n$ positive real numbers, then $D^{w+\text{iid}}$ is the space of all probability distributions of random variables $Y$ which can be written as a weighted sums $Y = w^\top X$ with $X = (X_i)_{i=1}^n$ a vector of $n$ i.i.d. random variables distributed according to some probability distribution in $D$.

**Example 12 (Hoeffding inequality)** Consider the space $D_{\text{max};M}$ of all probability distributions which have zero mean, and which are 0 outside the interval $[-M, M]$. Let $1_n$ be the vector of length $n$ having in all entries a 1. We denote with $D^{1_n;\text{iid};\text{max};M}$ the space of probability distributions followed by random variables which are sums of $n$ identically and independently distributed random variables all following the same probability distribution in $D_{\text{max};M}$. Then,

\[ b_{\text{Hoeffding}}((q, M), \lambda) = \exp \left( \frac{-n\lambda^2}{2M^2} \right) \]

is a concentration bound for $D^{1_n;\text{iid};\text{max};M}$ and $L_{I(\cdot \geq a)}$. It is known as the Hoeffding inequality.

**Definition 8 (weighted-sum concentration bound)** A weighted-sum concentration bound for a space of probability distributions $D_q$ parameterized by $q$ and a concentration measure $g_r$ parameterized by $r$ is a function $b(q, w, r)$ such that

\[ \forall q, w, r : L_{g_r}(D_q^{w+\text{iid}}) \leq b(q, w, r). \]

The following is a weighted version of the Hoeffding bound which follows immediately from Hoeffding’s original result:
Example 13 Consider the space $D_{\text{max};M}$ of all probability distributions which have zero mean and which are 0 outside the interval $[-M, M]$. Then,

$$b_{\text{Hoeffding}}((q, M), w, \lambda) = \exp\left(-\frac{-\lambda^2}{2\sum_{i=1}^n w_i^2 M^2}\right)$$

is a concentration bound for $D_{\text{max};M}$ and $L_1(\geq a)$.

As we will see, in a networked setting choosing a weight vector $w$ different from the one with $w_i = 1/n$ for all $1 \leq i \leq n$ is sometimes desirable. \cite{Janson2004} shows the following results which can be used to bound the error on averaging a function over networked sample.

**Theorem 9** Let $X = \{X_i\}_{i=1}^n$ be $H$-networked random variables with mean $\mathbb{E}(X_i) = \mu$, variance $\sigma^2(X_i) = \sigma^2$ and satisfying $|X_i - \mu| \leq M$. Then for all $\epsilon > 0$,

$$\Pr\left(\frac{1}{n} \sum_{i=1}^n X_i - \mu \geq \epsilon\right) \leq \exp\left(-\frac{n\epsilon^2}{2\chi^*(H)M^2}\right),$$

$$\Pr\left(\frac{1}{n} \sum_{i=1}^n X_i - \mu \geq \epsilon\right) \leq \exp\left(-\frac{8n\epsilon^2}{25\chi^*(H)(\sigma^2 + M\epsilon/3)}\right),$$

where $\chi^*$ is the fractional hyperedge-chromatic number.

Janson inequalities can only bound the error when we give every variable the same weight which we will call the EQW method. They are not useful when we give different weights for different variables, and we will see that even for only equally weighted case, Janson inequalities can be improved.

### 3.3 Networked variance

In this section, we analyze the variance of the statistics on $k$-partite networks (tuple networked examples). This analysis will result in a convex quadratically constrained linear program which minimizes the variance of the worst case.

First, we define a decomposition of functions defined on the hyperedges. Let $[k] = \{i \in \mathbb{N} | 1 \leq i \leq k\}$. Given a $k$ dimensional vector space $\mathcal{X} = \times_{i=1}^k \mathcal{X}^{(i)}$ and an index set $S \subseteq [k]$, we define $\mathcal{X}^{(S)} := \times_{i \in S} \mathcal{X}^{(i)}$ (where the Cartesian product is taken in increasing order of $i$). For every vector $x \in \mathcal{X}$, $x^{(S)}$ is the projection of $x$ into the space $\mathcal{X}^{(S)}$.

Consider a distribution $\rho$ of the vector $x \in \mathcal{X}$ (the marginal distribution of $x^{(S)}$ is denoted by $\rho_S$) and any zero-mean function $f$ defined on this space, we define that

$$\mu_S(x^{(S)}) := \mathbb{E}_{x^{([k]-S)} \sim \rho_{[k]-S}}(f(x)|x^{(S)}) - \sum_{T \subset S} \mu_T(x^{(T)})$$

and when $S = \emptyset$, $\mu_S(x^{(S)}) := \mathbb{E} f(x) = 0$.

From the definition, we can easily see that $f(x) = \sum_{S \subseteq [k]} \mu_S(x^{(S)})$. The following are the properties of this decomposition:
Lemma 10 Every $\mu_S$ is zero-mean for every dimension, i.e., $E_{x(i) \sim \rho_i} \mu_S(x(S)) = 0$ for any $i \in S$.

Proof By induction, we can show that,
$$\mu_S(x(S)) = \sum_{T \subseteq S} (-1)^{|S-T|} E_{x([k]-T) \sim \rho[k]-T} (f(x)|x^T).$$

Then,
$$E_{x(i) \sim \rho_i} \mu_S(x(S)) = \sum_{T:T \subseteq S-\{i\}} (-1)^{|S-T|} E_{x([k]-T) \sim \rho[k]-T} (f(x)|x^T)$$
$$+ \sum_{T:i \in T \subseteq S} (-1)^{|S-T|} E_{x([k]-T \cup \{i\}) \sim \rho[k]-T \cup \{i\}} (f(x)|x^{T-i})$$
$$= \sum_{T:T \subseteq S-\{i\}} (-1)^{|S-T|} E_{x([k]-T) \sim \rho[k]-T} (f(x)|x^T)$$
$$+ \sum_{T:T \subseteq S \setminus \{i\}} (-1)^{|S-T|+1} E_{x([k]-T) \sim \rho[k]-T} (f(x)|x^T)$$
$$= \sum_{T:T \subseteq S-\{i\}} (-1)^{|S-T|} E_{x([k]-T) \sim \rho[k]-T} (f(x)|x^T)$$
$$- \sum_{T:T \subseteq S-\{i\}} (-1)^{|S-T|} E_{x([k]-T) \sim \rho[k]-T} (f(x)|x^T) = 0. \blacksquare$$

Lemma 11 Two functions $\mu_S$ and $\mu_T$ are orthogonal, i.e., $E_{x(S \cup T) \sim \rho_{S \cup T}} \mu_S(x(S)) \mu_T(x(T)) = 0$ for any $S \neq T$.

Proof Because $S \neq T$, there is an $i$ that $i \in T$ but $i \notin S$ (or $i \in S$ but $i \notin T$, but we only need to consider one case). Then,
$$E_{x(S \cup T) \sim \rho_{S \cup T}} \mu_S(x(S)) \mu_T(x(T)) = E_{x(S \cup T - \{i\}) \sim \rho_{S \cup T - \{i\}}} \mu_S(x(S')) E_{x(i) \sim \rho_i} \mu_T(x(T)).$$

We use Lemma 10 that $E_{x(i) \sim \rho_i} \mu_T(x(T)) = 0$, so $E_{x(S \cup T) \sim \rho_{S \cup T}} \mu_S(x(S)) \mu_T(x(T)) = 0. \blacksquare$

Lemma 12 The variance of the function $f$ is the sum of the variances of $\mu_S$ of all $S$, i.e.,
$$\sigma^2 = \sum_{S \subseteq [k]} \sigma^2_S$$
where $\sigma^2_S = E_{x(S) \sim \rho_S} \mu_S^2(x(S))$.

Proof By definition,
$$f(x) = \sum_{S \subseteq [k]} \mu_S(x(S)).$$

Then, $\sigma^2 = \sum_{S,T \subseteq [k]} \text{cov}(\mu_S(x(S)), \mu_T(x(T))).$ Using the result in Lemma 11 when $S \neq T$, $\text{cov}(\mu_S(x(S)), \mu_T(x(T))) = 0$, so $\sigma^2 = \sum_{S \subseteq [k]} \sigma^2_S. \blacksquare$
According to the definitions above, if two examples \( e_i \) and \( e_j \) share the vertices in \( V(S) \) (we write \( J_{i,j} = S \)), then their covariance is \( \sum_{T \subseteq S} \sigma_T^2 \). Thus, given a \( k \)-partite hypergraph, we can construct a covariance matrix \( \Sigma \) with \( \Sigma_{i,j} = \sum_{T \subseteq J_{i,j}} \sigma_T^2 \), the covariance of \( e_i \) and \( e_j \). If we weight every example in the \( k \)-partite hypergraph with \( w \), the variance is \( \frac{w^T \Sigma w}{|w|^2} \). For simplicity, we consider the normalized variance for each example, \( \sigma^2 = 1 \). We also consider a normalized weighting vector \( w' \) that \( |w'|^1 = 1 \).

One can rewrite \( w'^T \Sigma w' \) as \( \sum_{i \in [n]} \sum_{j \in [n]} w'_i w'_j \sum_{T \subseteq J_{i,j}} \sigma_T^2 \).

We do not know the values of \( \sigma_T^2 \), so we consider the worst case, and then minimize the worst variance by choosing \( w' \). Therefore, we get an optimization problem:

\[
\min_{w'} \max_{\{ \sigma_T^2 : T \subseteq [k] \}} \sum_{i \in [n]} \sum_{j \in [n]} \sum_{T \subseteq J_{i,j}} \sigma_T^2
\]

s.t. \( \sum_{T \subseteq [k]} \sigma_T^2 = 1 \)
\( \sum_{i \in [n]} w'_i = 1 \)
\( \forall i : w'_i \geq 0 \)

We can see that it is not harmful to force \( \sigma_U^2 \) to be 0 if \( |U| > 1 \). If we have \( \{ \sigma_T^2 : T \subseteq [k] \} \) that \( \sigma_U^2 > 0 \) for some \( U \) and \( |U| > 1 \), then we can find \( \{ \sigma'_T^2 : T \subseteq [k] \} \) that achieves a larger objective value for the optimization problem above. The \( \{ \sigma'_T^2 : T \subseteq [k] \} \) can be constructed in this way: i) \( \sigma_U^2 = 0 \), ii) \( \sigma_l^2 = \sigma_l^2 + \sigma_U^2 \) for some \( l \in U \), and iii) \( \sigma_T^2 = \sigma_T^2 \) for \( T \neq U \) and \( T \neq l \). We rewrite the optimization problem:

\[
\min_{w'} \max_{\{ \sigma_l^2 : l \in [k] \}} \sum_{i \in [n]} \sum_{j \in [n]} \sum_{l \in J_{i,j}} \sigma_l^2
\]

s.t. \( \sum_{l \in [k]} \sigma_l^2 = 1 \)
\( \sum_{i \in [n]} w'_i = 1 \)
\( \forall i : w'_i \geq 0 \)

For a fixed \( w' \), the inner part

\[
\max_{\{ \sigma_l^2 : l \in [k] \}} \sum_{i \in [n]} \sum_{j \in [n]} \sum_{l \in J_{i,j}} \sigma_l^2
\]

s.t. \( \sum_{l \in [k]} \sigma_l^2 = 1 \)
is a linear program with decision variables \( \{\sigma_i^2\} \), so it reaches the maximum value when \( \sigma_i^2 = 1 \) for some \( l \). Then, the inner part is equivalent to:

\[
\max_{l \in [k]} \sum_{i \in [n]} \sum_{j \in [n]} w_i' w_j' \delta_{l \in J_{i,j}}
\]

where \( \delta_l = 1 \) if \( I \) is true, and \( \delta_l = 0 \) if \( I \) is false.

Introducing a new decision variable \( t \), one can rewrite the whole optimization problem as follows:

\[
\min_{w',t} \quad t \\
\text{s.t.} \quad \forall l : \sum_{i \in [n]} \sum_{j \in [n]} w_i' w_j' \delta_{l \in J_{i,j}} \leq t \\
\quad \sum_{i \in [n]} w_i' = 1 \\
\quad \forall i : w_i' \geq 0.
\]

We let \( w_i = w_i' / t \) for every \( i \), then \( |w|_1 = \sum_{i \in [n]} w_i = 1 / t \). The final convex quadratically constrained linear program is

\[
\max_w \quad \sum_{i \in [n]} w_i \\
\text{s.t.} \quad \forall l : \sum_{i} \sum_{j} w_i w_j \delta_{l \in J_{i,j}} \leq \sum_{i} w_i \\
\quad \forall i : w_i \geq 0.
\]

### 3.4 Networked concentration bounds

A common key property used for proving basic exponential concentration inequalities is that all observations are independent. That is, if \( X_1, \ldots, X_n \) are independent random variables, then the moment-generating function satisfies

\[
\mathbb{E}\exp\left(\sum_{i=1}^n X_i\right) = \prod_{i=1}^n \mathbb{E}e^{X_i}.
\]

However, when considering networked random variables, the equality can not be used. Instead, we will show a similar property which can be used for networked variables. Before that, we define feasible weights of hypergraphs.

**Definition 13 (feasible weights)** Given a hypergraph \( H = (V, E) \), a feasible weights \( w \) is a nonnegative vector \( \{w_i\}_{i=1}^n \) defined on its hyperedges satisfying that for every vertex \( v \in V \), \( \sum_{i : v \in e_i} w_i \leq 1 \).

**Theorem 14** Given \( H \)-networked random variables \( \{X_i\}_{i=1}^n \), if \( w \) is a feasible weight vector of the hypergraph \( H \), then

\[
\mathbb{E}\exp\left(\sum_{i=1}^n w_i X_i\right) \leq \prod_{i=1}^n (\mathbb{E}e^{X_i})^{w_i}.
\]
Before we prove it, we give a lemma which will be used in the proof. The lemma says that the weighted geometric mean function is concave.

Lemma 15 If $\beta = [\beta_1, \ldots, \beta_k] \in \mathbb{R}^k$ such that $\sum_{i=1}^k \beta_i \leq 1$ and $t = [t_1, \ldots, t_k] \in \mathbb{R}_+^k$, then the function $g(t) = \prod_{i=1}^k t_i^{\beta_i}$ is concave.

Proof We prove by showing that its Hessian matrix $\nabla^2 g(t)$ is negative semidefinite. $\nabla^2 g(t)$ is given by

$$
\frac{\partial^2 g(t)}{\partial t_i^2} = \frac{\beta_i (\beta_i - 1) g(t)}{t_i^2}, \quad \frac{\partial^2 g(t)}{\partial t_i \partial t_j} = \frac{\beta_i \beta_j g(t)}{t_i t_j},
$$

and can be expressed as

$$
\nabla^2 g(t) = (qq^T - \text{diag}(\beta_1/t_1^2, \ldots, \beta_n/t_n^2)) g(t)
$$

where $q = [q_1, \ldots, q_k]$ and $q_i = \beta_i/t_i$. We must show that $\nabla^2 g(t) \preceq 0$, i.e., that

$$
u^T \nabla^2 g(t) u = \left( \sum_{i=0}^k \beta_i u_i/t_i \right)^2 - \sum_{i=0}^k \beta_i u_i^2/t_i^2 \leq 0
$$

for all $u \in \mathbb{R}^k$. We let $\beta_0 = 1 - \sum_{i=1}^k \beta_i, u_0 = 0$ and $t_0$ be any positive number. Because $g(t) \geq 0$ for all $t$, we only need to prove

$$
\left( \sum_{i=0}^k \beta_i u_i/t_i \right)^2 - \sum_{i=0}^k \beta_i u_i^2/t_i^2 \leq 0.
$$

This follows from the fact that the square function is convex, and $\sum_{i=0}^k \beta_i u_i/t_i$ is a convex combination of $u_i/t_i$. \(\blacksquare\)

Now, we can prove Theorem 14.

Proof of Theorem 14. First, note that $X$ is determined by the features of vertices in the hypergraph, i.e., $o_1, \ldots, o_m$, then we rewrite the left hand side as

$$
\mathbb{E} \exp \left( \sum_{i=1}^N w_i X_i \right) = \mathbb{E}_{o_1, \ldots, o_m} \prod_{i=1}^N e^{w_i X_i}.
$$

Now, we focus on the expectation of $o_1$ and rewrite it as a conditional expectation,

$$
\mathbb{E} \exp \left( \sum_{i=1}^N w_i X_i \right) = \mathbb{E}_{o_1 \ldots, o_m} \left[ \mathbb{E}_{o_1} \prod_{i=1}^N e^{w_i X_i | o_2, \ldots, o_m} \right].
$$

We define $A = \{i | v_1 \in e_i\}$ and $B = \{i | v_1 \notin e_i\}$, i.e., the for all $i \in B$, $X_i$ is independent of $o_1$.

$$
\mathbb{E} \exp \left( \sum_{i=1}^N w_i X_i \right) = \mathbb{E}_{o_2, \ldots, o_m} \left[ \prod_{j \in B} e^{w_i X_i | o_1} \prod_{i \in A} e^{w_i X_i | o_2, \ldots, o_m} \right].
$$
By the definition of the feasible weights, \( \sum_{i \in A} w_i \leq 1 \). Using Lemma \[15\] to \( \mathbb{E}_{o_1} \prod_{i \in A} e^{w_i X_i} | o_2, \ldots, o_m \), we let \( t = [e^{X_i}]_{o_2, \ldots, o_m} \), \( \beta = [w_i]_{i \in A} \) and \( g(t) = \prod_{i \in A} e^{w_i X_i} | o_2, \ldots, o_m \). Lemma \[15\] tells us that \( g(t) \) is a concave function, and then

\[
\mathbb{E}_{o_1} g(t) \leq g(\mathbb{E}_o t)
\]

which follows the Jensen’s inequality. Notice that

\[
\mathbb{E}_{o_1} t = \mathbb{E}_{o_1} [e^{X_i}]_{o_2, \ldots, o_m} = [\mathbb{E}_{o_1} e^{X_i}]_{o_2, \ldots, o_m}.
\]

We can write

\[
\mathbb{E} \exp \left( \sum_{i=1}^{N} w_i X_i \right) \leq \mathbb{E}_{o_2, \ldots, o_m} \left[ \prod_{i \in B} e^{w_i X_i} \prod_{i \in A} (\mathbb{E}_{o_1} e^{X_i})^{w_i} | o_2, \ldots, o_m \right].
\]

We repeat the steps above for \( o_2, \ldots, o_m \). We can get

\[
\mathbb{E} \exp \left( \sum_{i=1}^{N} w_i X_i \right) \leq \prod_{i=1}^{N} \left( \mathbb{E} e^{X_i} \right)^{w_i}.
\]

Using Theorem \[14\] we are able to obtain exponential concentration inequalities of networked variables, and they will be used for bounding the sample error in Section 4. Necessary lemmas and their proofs can be found in Appendix A.

The following inequalities are analogues of the Bennett, Bernstein, Hoeffding inequalities and Chernoff bounds.

**Theorem 16** Let \( X = \{X_i\}_{i=1}^{n} \) be \( H \)-networked random variables with mean \( \mathbb{E}(X_i) = \mu \), variance \( \sigma^2(X_i) = \sigma^2 \), and satisfying \( |X_i - \mu| \leq M \). Let \( w \) be a feasible weight vector of \( H \) and \( |w| = \sum_{i=1}^{n} w_i \), then for all \( \epsilon > 0 \),

\[
\begin{align*}
\text{Pr} \left( \frac{1}{|w|} \sum_{i=1}^{n} w_i X_i - \mu \geq \epsilon \right) & \leq \exp \left( -\frac{|w| \epsilon}{2M} \log \left( 1 + \frac{M \epsilon}{\sigma^2} \right) \right), \\
\text{Pr} \left( \frac{1}{|w|} \sum_{i=1}^{N} w_i X_i - \mu \geq \epsilon \right) & \leq \exp \left( -\frac{|w| \epsilon^2}{2(\sigma^2 + \frac{3}{4} M \epsilon)} \right), \\
\text{Pr} \left( \frac{1}{|w|} \sum_{i=1}^{N} w_i X_i - \mu \geq \epsilon \right) & \leq \exp \left( -\frac{|w| \epsilon^2}{2M^2} \right).
\end{align*}
\]

**Proof** Apply Lemma \[27\] (see Appendix A) to the function \( X_i' = X_i/s \) which satisfies \( |X_i' - \mathbb{E}X_i'| \leq M/s, \sigma^2(X_i') = \sigma^2/s^2 \).

From the inequalities above, one can see that it is a better exponential bound if we can maximize \( |w| \). Given the hypergraph \( H = (V, E) \), this can be achieved by solving a linear
Learning from networked examples

\[
\begin{align*}
\text{max}_w \sum_{i=1}^{n} w_i \\
\text{s.t.} \quad \forall i : w_i \geq 0 \\
\quad \forall v \in V : \sum_{i : v \in e_i} w_i \leq 1
\end{align*}
\]

We will call the optimal value of the linear program the \(s\)-value [Wang et al., 2013]. That is, the \(s\)-value is defined as the maximum of the sum of the weights.

**Definition 17 (s-value)** Given a hypergraph \(H\), the \(s\)-value

\[s(H) = \max_w \sum_{e_i \in E(H)} w_i\]

where \(w\) is a feasible weight on \(E(H)\).

3.5 An analysis on large network models

In this section, we analyze what is the effect of several example weighting schemes in the context of two popular large-network models: the Barabási-Albert model and the Erdős-Rényi model. The first one, describing graphs with a powerlaw degree distribution, applies to many real-world applications. The latter model has been studied from a theoretical point of view in more depth [Bollobás, 1998].

3.5.1 Barabási-Albert model

In Barabási-Albert graphs, the degree distribution follows a powerlaw

\[p(d) = cd^{-\gamma}\]

where \(c\) is a constant satisfying \(\sum_{d=1}^{N} cd^{-\gamma} = 1\), i.e.,

\[c = \frac{1}{\sum_{d=1}^{N} d^{-\gamma}}\]

The most common value of \(\gamma\) which follows from the standard preferential attachment model is \(\gamma = 3\).

Consider a bipartite Barabási-Albert graph \(G = (V^{(1)} \cup V^{(2)}, E)\) with \(|V^{(1)}| = |V^{(2)}| = N\), if the random variables defined on the edges fully depend on the features of vertices in \(V^{(1)}\), for a vertex of degree \(d\) in \(V^{(1)}\), the variance of the sum of the random variables attached to the incident edges is

\[d^2 \sigma^2.\]

1. If we consider a standard Barabási-Albert graph, the range of \(d\) is from 1 to \(N - 1\). Here we make the range of \(d\) from 1 to \(N\) because of the bipartite Barabási-Albert graph model we will consider.
2. This bipartite model generates a graph as follows: 1) at time \(t_0\), the initial graph is \(K_{m,m}\), 2) at every time \(t_i\) \((N - m > i > 0)\), we add a new vertex to \(V^{(1)}\) and \(V^{(2)}\), and we add \(m\) edges between the new vertex in \(V^{(1)}\) \((V^{(2)}\) and the vertices in \(V^{(2)}\) \((V^{(1)}\) according to the preferential attachment principle.
The variance of the sum of all the random variables is

\[
N \sum_{d=1}^{N} p(d)d^2 \sigma^2 = Nc\sigma^2 \sum_{d=1}^{N} d^{-\gamma+2} \approx \begin{cases} 
\frac{N^{3-\gamma-1}}{3-\gamma}N\sigma^2 & \text{if } \gamma < 3 \\
N(\log N + O(1))\sigma^2 & \text{if } \gamma = 3 \\
A_{\gamma}N\sigma^2 & \text{if } \gamma > 3
\end{cases}
\]

where the approximations come from an integral representation given by Euler and when \( N \) is large enough and \( \gamma > 3 \), this value converges to a constant \( A_{\gamma} > 1 \). The average degree can be written as \( \sum_{d=1}^{N} d^{\gamma} \). Let \( \bar{d} \) (when \( N \) is large enough, \( \bar{d} = 2m \)) be the average degree of the graph, then \( c = \frac{\bar{d}}{\sum_{d=1}^{N} d^{-\gamma+1}} \). The number of random variables is \( \bar{d}N \), then we give every random variable the same weight \( \frac{1}{\bar{d}N} \). Therefore, the variance for the EQW method is approximately

\[
\begin{cases} 
\frac{N^{2-\gamma-1}}{(3-\gamma)d^{\gamma}}c\sigma^2 & \text{if } \gamma < 3 \\
\log N + O(1)\sigma^2 & \text{if } \gamma = 3 \\
A_{\gamma}\sigma^2 & \text{if } \gamma > 3
\end{cases}
\]

Now, we consider the weighting method. Wang et al. (2013) showed that if we use the weighting method described in Section 3.4, the variance

\[
\sigma^2_s \leq \frac{\sigma^2_s}{s}.
\]

We numerically compute the \( s \) values for some bipartite Barabási-Albert graphs, and the results are shown in Fig. 2a. One can see that, the \( s \) value is almost linear to the size of the network when \( m \) is fixed, then we write that \( s = q_mN \) where \( q_m \) is a constant. From our numerical result, \( q_1 = 0.603, q_2 = 0.835 \) and \( q_3 = 0.984 \). Besides, when \( m \) is large enough (\( m \geq 4 \)), \( s = N \) (i.e., \( q_m = 1 \)) with very high probability. Thus, the variance for the EQW method is upper bounded by \( \frac{\sigma^2_{EQW}}{q_mN} \).

Comparing the EQW method with the weighting method, we can see that when \( \gamma < 3 \), \( \frac{\sigma^2_{EQW}}{\sigma^2_s} \) is at least of the order of \( N^{3-\gamma} \), when \( \gamma > 3 \), \( \frac{\sigma^2_{EQW}}{\sigma^2_s} \) is lower bounded by a constant (which is usually greater than one) only depends on \( \gamma \). The theoretically most well-known value \( \gamma = 3 \) is a critical value for the weighting method making a real difference, \( \frac{\sigma^2_{EQW}}{\sigma^2_s} \) is at least of the order of \( \log N \). We see that when \( \gamma \leq 3 \), \( \frac{\sigma^2_{EQW}}{\sigma^2_s} \) goes infinity when \( N \) approaches infinity.

3.5.2 Erdős-Rényi model

Consider a bipartite Erdős-Rényi graph \( G = (V^{(1)} \cup V^{(2)}, E) \) with \( |V^{(1)}| = |V^{(2)}| = N \). The degree distribution in a Erdős-Rényi graph follows a

\[
p(d) = \binom{N}{d} p^d(1-p)^{N-d}.
\]

3. In this bipartite graph model, every vertex in \( V^{(1)} \) is adjacent to a vertex in \( V^{(2)} \) with a fixed probability \( p \).
(a) The $s$ values of Barabási-Albert graphs

(b) The $s$ values of Erdős-Rényi graphs

Figure 2: The $s$ values of different networks
Similar to the analysis in the above section, the variance of the sum of all the random variables is
\[ N \sum_{d=1}^{N} p(d) d^2 \sigma^2 = N \sigma^2 \sum_{d=1}^{N} d \binom{N}{d} p^d (1 - p)^{N-d} d^2 = N^2 \sigma^2 (p^2 N + p - p^2). \]

The number of random variables is \( pN^2 \). Therefore, the variance for the EQW method is
\[ \frac{N^2 \sigma^2 (p^2 N + p - p^2)}{p^2 N^4} = \left( \frac{1}{N} + \frac{1 - p}{N^2 p} \right) \sigma^2. \]

Now, we consider the weighting method. Remember that the variance of the weighting method is bounded by \( \sigma^2 s \).

Again, our numerical result in Fig. 2b shows that when \( pN \) is a constant, the \( s \) value is linear to the size of the network. From our numerical result, \( q_{2/N} = 0.782 \) and \( q_{4/N} = 0.977 \). Then, \( \sigma^2 \) is bounded by \( \frac{\sigma^2 s}{q_p N} \) where \( q_p \) is a constant.

In many real-world networks, the average degree is rather low. Especially in that range, the weighting approach performs slightly better than the EQW method, but never worse.

One conclusion is that because many real-world applications show power-law behavior with \( 2 \leq \gamma \leq 4 \), the applications may benefit from the use of an appropriate weighting scheme.

### 4. Learning theory

In this section, we first review some basic concepts of statistical learning theory, and then discuss the learning theory for networked examples of different strategies. Table 1 summarizes the several notations introduced in this section.

| Symb  | Definition                                      | Symb  | Definition                                      |
|-------|------------------------------------------------|-------|------------------------------------------------|
| \( \rho \) | distribution over examples                      | \( Z \) | training examples                               |
| \( \mathcal{F} \) | space of all measurable functions               | \( \mathcal{H} \) | hypothesis space                               |
| \( \mathcal{X} \) | input space                                     | \( \mathcal{Y} \) | output space                                    |
| \( Z \) | \( \mathcal{X} \times \mathcal{Y} \)           | \( f \) | any function \( \mathcal{X} \mapsto \mathcal{Y} \) |
| \( L(\cdot, \cdot) \) | loss function                                    | \( \mathcal{E}(\cdot) \) | expected risk w.r.t square loss                  |
| \( \mathcal{E}_Z(\cdot) \) | empirical risk w.r.t square loss                 | \( f_{\rho, \mathcal{H}} \) | minimizer of \( \mathcal{E}(\cdot) \) in \( \mathcal{H} \) |
| \( f_{\rho, \mathcal{F}} \) | global minimizer of \( \mathcal{E}(\cdot) \)    | \( f_{Z, \mathcal{H}} \) | minimizer of \( \mathcal{E}_Z(\cdot) \) in \( \mathcal{H} \) |
| \( \mathcal{C}(\cdot) \) | a Banach space of continuous functions          | \( \mathcal{E}_S(\cdot) \) | sample error                                    |
| \( \mathcal{E}_A(\cdot) \) | approximation error                             | \( \mathcal{N}(\cdot, \cdot) \) | covering number                                 |

Table 1: Notations introduced in Section 4

#### 4.1 Expected risk and empirical risk

The main goal of supervised learning is to learn a function \( f : \mathcal{X} \mapsto \mathcal{Y} \) from a set of training examples \( Z = \{ z_i \}_{i=1}^{n} \) with \( z_i = (x_i, y_i) \sim \rho \), and to predict labels for unseen examples. We define a loss function \( L : \mathcal{Y} \times \mathcal{Y} \mapsto \mathbb{R}_+ \). The value \( L(f(x), y) \) denotes the expected local
error suffered from the use of $f$ to predict $y$ from $x$. In this paper, we use the square loss function, that is $L(f(x), y) = (f(x) - y)^2$. Note that our analysis can easily be extended to general loss functions case. We can measure the predictive ability of a learned model $f$ approximating $\rho$ by averaging the local error over all pairs $(x, y)$ by integrating over $\mathcal{Z}$ with respect to $\rho$. More precisely, we define the expected risk as

$$\mathcal{E}(f) = \int_{\mathcal{Z}} (f(x) - y)^2 \rho(x, y) dx dy.$$  

A natural idea is to find the minimizer $f_{\rho, F}$ of $\mathcal{E}(f)$ over all functions, i.e.,

$$f_{\rho, F} = \arg\min \mathcal{E}(f),$$

where the minimization is taken over the set of all measurable functions $\mathcal{F}$. Unfortunately, the probability distribution $\rho$ is unknown, $f_{\rho, F}$ can not be computed directly. If every example in $\mathcal{Z}$ were independent from each other, by the law of large numbers, as the sample size $N$ tends to infinity, the empirical risk

$$\mathcal{E}_Z(f) = \frac{1}{N} \sum_{i=1}^{N} (f(x_i) - y_i)^2$$

converges to the expected risk $\mathcal{E}(f)$. Then we may get a good candidate $f_{Z, F}$ to approximate the target function $f_{\rho, F}$, where

$$f_{Z, F} = \arg\min \mathcal{E}_Z(f).$$

4.2 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 (ERM) principle (Shawe-Taylor et al., 1998) is to find the minimizer in a properly selected hypothesis space $\mathcal{H}$, i.e.,

$$f_{Z, H} = \arg\min_{f \in \mathcal{H}} \mathcal{E}_Z(f).$$

The performance of the ERM approach is commonly measured in terms of the excess risk

$$\mathcal{E}(f_{Z, H}) - \mathcal{E}(f_{\rho, F}).$$

If we define

$$f_{\rho, H} = \arg\min_{f \in \mathcal{H}} \mathcal{E}(f),$$

then the excess risk can be decomposed as

$$\mathcal{E}(f_{Z, H}) - \mathcal{E}(f_{\rho, F}) = [\mathcal{E}(f_{Z, H}) - \mathcal{E}(f_{\rho, H})] + [\mathcal{E}(f_{\rho, H}) - \mathcal{E}(f_{\rho, F})].$$

We call the first part the sample error $\mathcal{E}_s(\mathcal{Z}) := \mathcal{E}(f_{Z, H}) - \mathcal{E}(f_{\rho, H})$, the second part the approximation error $\mathcal{E}_A(\mathcal{H}) := \mathcal{E}(f_{\rho, H}) - \mathcal{E}(f_{\rho, F})$.

The approximation error is independent of the sample and it is studied in (Cucker and Zhou, 2007). It is an interesting question 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. In this paper, we concentrate on the sample error.

The complexity of the hypothesis space is usually measured in terms of covering number (Zhou 2002), entropy number (Tsuda 1999), VC-dimension (Vapnik et al., 1994), etc. As an illustration of our approach, in this paper, we will use the covering numbers defined below to measure the capacity of our hypothesis space $H$, and the hypothesis space $H$ will be chosen as a subset of $C(X)$ which is a Banach space of continuous functions on a compact metric space $X$ with the norm $\|f\|_\infty = \sup_{x \in X} |f(x)|$. However, our approach can be applied using other hypothesis space measures as well.

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

**Definition 18 (Covering number)** Let $H$ be a metric space and $\tau > 0$. We define the covering number $N(H, \tau)$ to be the minimal $\ell \in \mathbb{N}$ such that there exists $\ell$ disks in $H$ with radius $\tau$ covering $H$. When $H$ is compact, this number is finite.

**Definition 19 (M-bounded functions)** Let $M > 0$ and $\rho$ be a probability distribution on $\mathcal{Z}$. We say that a set $H$ of functions from $X$ to $Y$ is M-bounded when

$$\Pr_{(x,y) \sim \rho} \left( \sup_{f \in H} |f(x) - y| \leq M \right) = 1.$$  

The following result can be found in (Cucker and Zhou, 2007).

**Theorem 20** Let $H$ be a compact and convex subset of $C(X)$. If $H$ is M-bounded, then for all $\epsilon > 0$,

$$\Pr \left( \mathcal{E}_S(Z) \geq \epsilon \right) \leq N(H, \frac{\epsilon}{\sqrt{12M}}) \exp \left( -\frac{N\epsilon}{300M^4} \right).$$

4.3 Learning theory for networked examples

In this subsection, we provide statistical learning theory for learning from networked examples. We consider three methods, and they have different upper sample error bounds which are related to different important parameters of hypergraphs. The first two methods are straightforward, but from the upper bound point of view, they waste the information provided by the networked examples. The third method reaches a better sample error bound via solving the linear program discussed in Section 3.4.

4.3.1 The EQW method

Let us now consider the EQW learning strategy 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 19 above to bound the sample error of EQW:
Learning from networked examples

Theorem 21 Let $\mathcal{H}$ be a compact and convex subset of $C(X)$, and $Z$ be an $H$-networked sample. If $\mathcal{H}$ is $M$-bounded, then for all $\epsilon > 0$,

$$\Pr \left( \mathcal{E}_S(Z) \leq \epsilon \right) \geq 1 - \mathcal{N} \left( \mathcal{H}, \frac{\epsilon}{12M} \right) \exp \left( - \frac{3N\epsilon}{1400\chi^*(\Gamma)M^4} \right).$$

The result above show that the bound of the sample error not only relies on the sample size but also the fractional hyperedge-chromatic number $\chi^*$. That is, a larger sample may result in a poorer sample error bound since $\chi^*$ can also become larger.

4.3.2 The IND method

A straightforward idea to learn from an $H$-networked sample $Z$ is to find a (maximal) subset $Z_I \subseteq Z$ of training examples which correspond to a matching in $H$. Due to our assumptions, such set will be an i.i.d. sample. We can then perform algorithms on $Z_I$ for learning. We can define the empirical risk

$$\mathcal{E}_{Z_I}(f) = \frac{1}{|Z_I|} \sum_{z_i \in Z_I} (f(x_i) - y_i)^2,$$

and the function we obtain by ERM principle is

$$f_{Z_I, \mathcal{H}} = \arg \min_{f \in \mathcal{H}} \mathcal{E}_{Z_I}(f).$$

We call this method the IND method. To bound the sample error of this method, we can directly use Theorem 20.

The key step of the IND method is to find a large $Z_I$. The larger $|Z_I|$ is, the more accurate $f_{Z_I}$ we can expect. To find a large $Z_I$ is equivalent to to find a large matching in $H$. For any hypergraph $H$, it holds that (see, e.g., Diestel (2010)),

$$\frac{n}{\chi^*(H)} \leq \alpha(H),$$

where $\alpha(H)$ is the matching number of the hypergraph $H$. Therefore, if we can find a maximum matching of the hypergraph $H$, then the sample error bound of the IND method is better than that of the EQW method discussed earlier (This is debatable, see Section 4.4 for a detailed discussion). However, given a positive integer $n$, it is in general an NP-complete problem to decide whether there is a matching in $H$ of size greater than $n$ (Garey and Johnson, 1979). Moreover, this problem is also an APX-complete problem (Uriel et al., 1991), so we would not expect an efficient algorithm to achieve a good approximation in practice.

4.3.3 The weighting method

The IND method can not be used in practice since it is difficult to find a large independent set of networked examples, and we will see that it may waste some information provided by networked examples. We propose another method based on the inequalities in Section 3.4 and this method actually computes the linear relaxation of the maximum matching.
The $s$-value can be considered as a linear program relaxation of the maximum matching problem \cite{Lovasz1975,Chan2012}, so it always holds that $s(H) \geq \alpha(H)$. We call the weight vector $w^*$ which makes the linear program maximum the optimal weighting.

For an $H$-networked sample $Z$, we denote the weighted sample $Z_s = \{(z_i, w_i)\}$ where $[w_1, \ldots, w_n]$ is an optimal weighting. Now we can define a new empirical risk on the weighted sample $Z_s$ that
\[
E_{Z_s}(f) = \frac{1}{s} \sum_{i=1}^{n} w_i(f(x_i) - y_i)^2.
\]

Later, we will see that the empirical risk $E_{Z_s}$ converges to the expected risk
\[
E(f) = \int (f(x) - y)^2 \rho(x,y) dxdy
\]
as $s$ tends to infinity for fixed $f$.

We consider the ERM approach associated with $Z_s$. As discussed in Section 4.2, the ERM approach aims to find a minimizer of the empirical risk in a proper hypothesis space $H$ to approximate the target function, i.e.,
\[
 f_{Z_s,H} = \arg \min_{f \in H} E_{Z_s}(f).
\]

Then the performance of the ERM approach is measure by the excess risk
\[
E(f_{Z_s,H}) - E(f_{\rho,F}).
\]

Recall the definition $f_H = \arg \min_{f \in H} E(f)$, the excess risk can be divided into two parts (sample error and approximation error) as follows
\[
E(f_{Z_s,H}) - E(f_{\rho,F}) = [E(f_{Z_s,H}) - E(f_H)] + [E(f_H) - E(f_{\rho,F})].
\]

Notice that the approximation error $E(f_H) - E(f_{\rho,F})$ is independent of sample $Z_s$, and the approximation error vanishes if $f_{\rho,F} \in H$.

In this section, we focus on the sample error $E_S(f_{Z_s,H}) := E(f_{Z_s,H}) - E(f_{\rho,H})$. To this end, we use the probability inequalities with $s$-value to estimate the sample error $E_S(f_{Z_s,H})$.

The definition of $f_{Z_s,H}$ tells us that $E_{Z_s}(f_{Z_s,H}) - E_{Z_s}(f_{\rho,H}) \leq 0$, then the sample error can be decomposed as
\[
E_S(f_{Z_s,H}) = E(f_{Z_s,H}) - E(f_{\rho,H})
= \left[ E(f_{Z_s,H}) - E_{Z_s}(f_{Z_s,H}) \right] + \left[ E_{Z_s}(f_{Z_s,H}) - E_{Z_s}(f_{\rho,H}) \right] + \left[ E_{Z_s}(f_{\rho,H}) - E(f_{\rho,H}) \right].
\]

The following is the main result of this paper.

**Theorem 22** Let $H$ be a compact and convex subset of $C(X)$. If $H$ is a $M$-bounded, then for all $\epsilon > 0$, \[
\Pr \left( E_S(f_{Z_s}) \leq \epsilon \right) \geq 1 - \mathcal{N} \left( \mathcal{H}, \frac{\epsilon}{12M} \right) \exp \left( -\frac{s\epsilon}{300M^2} \right).
\]
Proof Consider the function set
\[ G = \{(f(x) - y)^2 - (f_H(x) - y)^2 : f \in H\}. \]
For any function \( g \in G \), we have \( E(g) = E_H(f) \geq 0 \). Since \( H \) is M-bounded, which implies that \( -M^2 \leq g(z) \leq M^2 \) and \( |g - E(g)| \leq B = 2M^2 \) hold almost everywhere. One can easily see that
\[ g(z) = (f(x) - f_{\#H}(x))(f(x) - y) + (f_{\#H}(x) - y), \quad z = (x, y) \in Z. \]
If follows that \( |g(z)| \leq 2M|f(x) - f_{\#H}(x)| \) and \( E(g^2) \leq 4M^2 \int_X (f(x) - f_{\#H}(x))^2 \rho(x)dx. \)
Together with Lemma 32 (see Appendix B) this implies that \( E(g^2) \leq 4M^2 \mathcal{E}_H(f) = cE(g) \) with \( c = 4M^2 \). Hence, all the conditions in Lemma 31 (see Appendix B) hold true and we have the following conclusion from the identity \( E_{Z_t}(g) = E_{H, Z}(f) \):
for every \( \epsilon > 0 \) and \( 0 < \alpha \leq 1 \), with probability at least
\[ 1 - N(G, \alpha \epsilon) \exp \left\{ -\frac{\alpha^2 \epsilon^2}{8M^2 + \frac{4M^2}{3}} \right\} \]
there holds
\[ \sup_{f \in H} \frac{\mathcal{E}_S(f) - \mathcal{E}_{H, Z}(f)}{\sqrt{\mathcal{E}_S(f)} + \epsilon} \geq 4\sqrt{\alpha} \epsilon. \]
that means for all \( f \in H \), \( \mathcal{E}_S(f) \leq \mathcal{E}_{H, Z}(f) + 4\sqrt{\alpha} \epsilon \sqrt{\mathcal{E}_S(f)} + \epsilon \). Now we take \( \alpha = \sqrt{2}/8 \) and \( f = f_{Z_t} \). The definition of \( f_{Z_t} \) tells us that \( \mathcal{E}_{H, Z}(f_{Z_t}) \leq 0 \), it follows that
\[ \mathcal{E}_H(f_z) \leq \sqrt{2}/\epsilon \sqrt{\mathcal{E}_S(f_{Z_t})} + \epsilon. \]
The above quadratic equation with respect to \( \sqrt{\mathcal{E}_H(f_{Z_t})} \) implies \( \mathcal{E}_H(f_{Z_t}) \leq \epsilon \). Finally, the inequality \( \|g_1 - g_2\|_{C(Z)} = \|f_1(x) - f_2(x)(f_1(x) - y) + (f_2(x) - y)\|_{C(Z)} \leq 2M\|f_1 - f_2\|_{C(X)} \),
tells us that
\[ N(G, \alpha \epsilon) \leq N(H, \frac{\alpha \epsilon}{2M}). \]
This completes our proof by taking \( \alpha = \sqrt{2}/8 \).

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 method, which is initially proposed to solve ill-posed phenomenon 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 3.4.

4.4 An improved sample error bound for the EQW method

The EQW method discussed in Section 4.3.1 can be considered as a weighting method that every example is weighted with a same value, i.e., \( w_1 = \ldots = w_n \). We rewrite the empirical risk of the EQW method as follows:
\[ \mathcal{E}_{Z}(f) = \frac{1}{n} \sum_{i=1}^{n} (f(x_i) - y_i)^2 = \frac{1}{w_1 n} \sum_{i=1}^{n} w_1 (f(x_i) - y_i)^2. \]
If the weight \( w_1 \) is a feasible weight, i.e., \( 0 \leq w \leq \frac{1}{\omega(H)} \) where \( \omega(H) := \max_{v \in V(H)} |\{e : v \in e\}| \), we can use the inequality in Section 3.4 to analyze the EQW method.
Theorem 23 Let $X = \{X_i\}$ be $H$-networked random variables with mean $\mathbb{E}(X_i) = \mu$, variance $\sigma^2(X_i) = \sigma^2$, and satisfying $|X_i - \mu| \leq M$. Then for all $\epsilon > 0$,

\[
\Pr \left( \frac{1}{n} \sum_{i=1}^{n} X_i - \mu \geq \epsilon \right) \leq \exp \left( -\frac{n\epsilon}{2\omega(H)M} \log \left( 1 + \frac{M\epsilon}{\sigma^2} \right) \right),
\]

\[
\Pr \left( \frac{1}{n} \sum_{i=1}^{n} X_i - \mu \geq \epsilon \right) \leq \exp \left( -\frac{n\epsilon^2}{2\omega(H)(\sigma^2 + \frac{1}{3}M\epsilon)} \right),
\]

\[
\Pr \left( \frac{1}{n} \sum_{i=1}^{n} X_i - \mu \geq \epsilon \right) \leq \exp \left( -\frac{n\epsilon^2}{2\omega(H)M^2} \right).
\]

Proof We rewrite the left hand side

\[
\Pr \left( \frac{1}{n} \sum_{i=1}^{n} X_i - \mu \geq \epsilon \right) = \Pr \left( \frac{\omega(H)}{n} \sum_{i=1}^{n} \frac{1}{\omega(H)} X_i - \mu \geq \epsilon \right).
\]

Then, we can use the Theorem 16.

Using these inequalities, we obtain a better sample error bound for the EQW method since $\omega(H) \leq \chi^*(\Gamma)$.

Theorem 24 Let $\mathcal{H}$ be a compact and convex subset of $\mathcal{C}(X)$, and $Z$ be an $H$-networked sample. If $\mathcal{H}$ is $M$-bounded, then for all $\epsilon > 0$,

\[
\Pr \left( \mathcal{E}_S(Z) \geq \epsilon \right) \leq \mathcal{N} \left( \mathcal{H}, \frac{\epsilon}{12M} \right) \exp \left( -\frac{N\epsilon}{300\omega(H)M^4} \right).
\]

A direct conclusion from the theorem above is that, we can not claim that the sample error bound of the IND method is better than that of the EQW method because it is possible that $\alpha(H) < \omega(H)$. However, $\frac{\alpha(H)}{\omega(H)} \leq s(H)$ always holds, then the s-value method still can reach better sample error bound than the EQW method.

5. Related Work

In this section, we discuss the related work.

5.1 Hypothesis test

In [Wang et al., 2011], the authors consider a similar setting of networked examples. They use the dependency graph to represent the examples and their relations. While we assume a worst case over all possible dependencies, and allow to model explicitly causes of dependencies (represented with vertices which can be incident with more than two edges), this work assumes a bounded covariance between pairs of examples connected with an edge (excluding possible higher-order interactions). While we use our model to show learning guarantees, Wang et al. (2011) shows corrections for the bias (induced by the dependencies between examples) on statistical hypothesis tests. It seems plausible that both models can be applied for both the learning guarantee and statistical testing tasks.
5.2 Concentration inequalities for U statistics

Hoeffding (1948) gave concentration inequalities for U statistics. Using the result in Section 4.4, we can improve the concentration inequalities. As an example, we only consider one-sample U statistics here.

**Definition 25 (One-sample U statistics)** Let \( s_1, s_2, \ldots, s_n \) be independent random variables. For \( n \geq r \) consider a random variable of the form

\[
U = \frac{1}{n^{(r)}} \sum_{n,r} X_i(s_{i_1}, \ldots, s_{i_r})
\]

where \( n^{(r)} = n(n-1) \cdots (n-r+1) \) and the sum \( \sum_{n,r} \) is taken over all \( r \)-tuples \( i_1, \ldots, i_r \) of distinct positive integers not exceeding \( n \). The random variable \( U \) is called a one-sample U statistics.

If the function \( X_i \) is bounded, \( a \leq X_i \leq b \), Hoeffding (1948) showed that for any \( \epsilon > 0 \),

\[
\Pr (U - \mu \geq \epsilon) \leq \exp\left(-\frac{2[n/r] \epsilon^2}{(b-a)^2}\right)
\]

where \( \mu \) is the expected value of \( X_i \).

Arcones (1995) showed a Bernstein-type bound that for any \( \epsilon > 0 \),

\[
\Pr (U - \mu \geq \epsilon) \leq \exp\left(-\frac{2[n/r] \epsilon^2}{\sigma^2 + M\lambda}ight)
\]

However, a corollary of our result shows that the operator \( \lfloor \cdot \rfloor \) is not necessary, i.e.,

\[
\Pr (U - \mu \geq \epsilon) \leq \exp\left(-\frac{2ne^2}{r(b-a)^2}\right)
\]

and

\[
\Pr (U - \mu \geq \epsilon) \leq \exp\left(-\frac{2n/\epsilon^2}{\sigma^2 + M\lambda}\right)
\]

To prove these inequalities, we just let \( N = n^{(r)} \) and \( \omega(H) = r(n-1)(n-2) \cdots (n-r+1) \) in Theorem 23.

5.3 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 in time series analysis. For example, in (Guo and Shi, 2011), 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 \( \alpha \)-mixing or \( \beta \)-mixing conditions. In (Sun and Wu, 2010), regularized least square regression with dependent samples is considered under the assumption that the training sample satisfies some mixing conditions. In (Modha and Masry, 1996), the authors established a Bernstein type
inequality is presented for stationary exponentially $\alpha$-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 (Bradley, 2005) and references therein for more details about the mixing conditions.

6. Conclusions

In this paper, we introduce a problem that learning from network data. We first show that it may result in poor prediction if we ignore the dependency relationship between the examples. We analyze the IND method that performing learning algorithm on selected independent examples. 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, and the weights are efficiently computable. To assess learning algorithms on these weighted examples, we derived some statistical inequalities. Using these inequalities, we can prove the generalization bounds and estimate the sample error.

In the future, we want to consider settings where we do not make such strong independence assumptions 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.

Acknowledgments

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

Miguel A. Arcones. A bernstein-type inequality for u-statistics and u-processes. *Statistics & probability letters*, 22(3):239–247, 1995.

Bla Bollobs. *Random graphs*. Springer New York, 1998.

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

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

Felipe Cucker and Ding-Xuan Zhou. *Learning theory: an approximation theory viewpoint*. Cambridge University Press, 2007.
Reinhard Diestel. *Graph theory.* Springer-Verlag, 2010.

Daan Fierens, Hendrik Blockeel, Maurice Bruynooghe, and Jan Ramon. Logical bayesian networks and their relation to other probabilistic logical models. In *Inductive Logic Programming,* pages 121–135. Springer Berlin Heidelberg, 2005.

Nir Friedman, Lise Getoor, Daphne Koller, and Avi Pfeffer. Learning probabilistic relational models. In *IJCAI,* volume 99, pages 1300–1309, 1999.

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

Lise Getoor and eds. Taskar, Ben. *Introduction to statistical relational learning,* 2007.

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

Wassily Hoeffding. A class of statistics with asymptotically normal distributions. *Annals of Statistics,* 19:293–325, 1948.

Manfred. Jaeger. Relational bayesian networks. In *Proceedings of the Thirteenth conference on Uncertainty in artificial intelligence,* pages 266–273. Morgan Kaufmann Publishers, 1997.

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

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

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

John Shawe-Taylor, Peter L. Bartlett, Robert C. Williamson, and Martin Anthony. Structural risk minimization over data dependent hierarchies. *Information Theory, IEEE Transactions on,* 44(5):1926–1940, September 1998.

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

Koji Tsuda. Optimal hyperplane classifier based on entropy number bound. volume 1.4, pages 419–424, 1999.

Feige Uriel, Shafi Goldwasser, László Lovász, Shmuel Safra, and Mario Szegedy. Approximating clique is almost NP-complete. In *In Foundations of Computer Science, 1991. Proceedings., 32nd Annual Symposium on,* pages 2–12, 1991.

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.
Wang et al.

Vladimir Vapnik, Esther Levin, and Yann Le Cun. Measuring the VC-dimension of a learning machine. *Neural Computation*, 6(5):851–876, 1994.

Tao Wang, Jennifer Neville, Brian Gallagher, and Tina Eliassi-Rad. Correcting bias in statistical tests for network classifier evaluation. In *Proceedings of ECML/PKDD*, volume 6913 of *LNCS*, pages 506–521, 2011.

Yuyi Wang, Jan Ramon, and Thomas Fannes. An efficiently computable subgraph pattern support measure: counting independent observations. *Data Mining and Knowledge Discovery*, 27(3):444–477, 2013.

Ding-Xuan Zhou. The covering number in learning theory. *Journal of Complexity*, 18(3):739–767, 2002.

**Appendix A**

In this part, we give proofs for the lemmas and the theorems in Section 4.3.3. The main ideas were borrowed from (Cucker and Zhou, 2007).

**Lemma 26** Let $X = \{X_i\}$ be $H$-networked random variables with mean $E(X_i) = \mu$, variance $\sigma^2(X_i) = \sigma^2$, and satisfying $|X_i - E(X_i)| \leq M$. Let $w = \{w_i\}$ be a feasible weight of $H$ and $|w| = \sum_i w_i$, then for all $\epsilon > 0$,

$$
\Pr \left( \sum_i w_i(X_i - \mu) \geq \epsilon \right) \leq \exp \left( -\frac{|w|\sigma^2}{M^2} h \left( \frac{M\epsilon}{|w|\sigma^2} \right) \right)
$$

where $h$ is given by $h(a) = (1 + a) \log(1 + a) - a$.

**Proof** Without loss of generality, we assume $\mu = 0$. Let $c$ be an arbitrary positive constant which will be determined later. Then

$$
I := \Pr \left( \sum_{i=1}^N w_i X_i \geq \epsilon \right) = \Pr \left( \exp \left( c \sum_{i=1}^N w_i X_i \right) \geq e^{c\epsilon} \right).
$$

By Markov’s inequality and Theorem 14, we have

$$
I \leq e^{-c\epsilon} \mathbb{E} \left( \exp \left( c \sum_{i=1}^N w_i X_i \right) \right) \leq e^{-c\epsilon} \prod_i \left( \mathbb{E} e^{cX_i} \right)^{w_i}.
$$

Since $|X_i| \leq M$ and $\mu = 0$, we have

$$
\mathbb{E} e^{cX_i} = 1 + \sum_{p=2}^{+\infty} \frac{c^p \mathbb{E} X_i^p}{p!} \leq 1 + \sum_{p=2}^{+\infty} \frac{c^p M^{p-2} \sigma^2}{p!}
$$

from the Taylor expansion for exponential functions. Using $1 + a \leq e^a$, it follows that

$$
\mathbb{E} e^{cX_i} \leq \exp \left( \sum_{p=2}^{+\infty} \frac{c^p M^{p-2} \sigma^2}{p!} \right) = \exp \left( \frac{e^{cM} - 1 - cM \sigma^2}{M^2 \sigma^2} \right)
$$

28
and therefore

\[ I \leq \exp \left( -c \epsilon + \frac{e^{cM} - 1 - cM}{M^2} |w|\sigma^2 \right). \]

Now choose the constant \( c \) to be the minimizer of the bound on the right hand side above:

\[ c = \frac{1}{M} \log(1 + \frac{M\epsilon}{|w|\sigma^2}). \]

That is, \( e^{cM} - 1 = \frac{M\epsilon}{|w|\sigma^2} \). With this choice,

\[ I \leq \exp \left( -\frac{|w|\sigma^2}{M^2} h \left( \frac{M\epsilon}{|w|\sigma^2} \right) \right). \]

This proves the desired inequality.

Lemma 27 Let \( X = \{X_i\} \) be \( H \)-networked random variables with mean \( \mathbb{E}(X_i) = \mu \), variance \( \sigma^2(X_i) = \sigma^2 \), and satisfying \( |X_i - \mathbb{E}(X_i)| \leq M \). Let \( w = \{w_i\} \) be a feasible weight of \( H \) and \( |w| = \sum w_i \), then for all \( \epsilon > 0 \),

\[
\begin{align*}
\Pr \left( \sum_{i=1}^{N} w_i(X_i - \mu) \geq \epsilon \right) &\leq \exp \left( -\frac{\epsilon}{2M} \log(1 + \frac{M\epsilon}{\sigma^2}) \right), \\
\Pr \left( \sum_{i=1}^{N} w_i(X_i - \mu) \geq \epsilon \right) &\leq \exp \left( -\frac{\epsilon^2}{2(\sigma^2 + \frac{1}{3}M\epsilon)} \right), \\
\Pr \left( \sum_{i=1}^{N} w_i(X_i - \mu) \geq \epsilon \right) &\leq \exp \left( -\frac{\epsilon^2}{2M^2} \right).
\end{align*}
\]

Proof The first inequality follows from Lemma 26 and the inequality

\[ h(a) \geq \frac{a}{2} \log(1 + a), \ \forall a \geq 0. \quad (2) \]

The second inequality follows from Lemma 26 and the inequality

\[ h(a) \geq \frac{3a^2}{6 + 2a}, \ \forall a \geq 0. \]

To prove the third inequality, we use Theorem 14. As the exponential function is convex and \( -M \leq X_i \leq M \) almost surely, there holds

\[ e^{cX_i} \leq \frac{e^{cM} - (-cM)}{2cM} e^{cM} + \frac{cM - cX_i}{2cM} e^{-cM}. \]
almost everywhere. It follows from \( E(X_i) = 0 \) and the Taylor expansion for the exponential function that
\[
E e^{cX_i} \leq e^{-cM} + e^{cM} = \sum_{p=0}^{+\infty} \frac{(-cM)^p}{p!} + \sum_{p=0}^{+\infty} \frac{(cM)^p}{p!} = \sum_{p=0}^{+\infty} \frac{(cM)^{2p}}{(2p)!} = \sum_{p=0}^{+\infty} \left( \frac{cM}{2} \right)^{2p} (2p)! = e^{(cM)^2/2}.
\]
This, together with Theorem \[9\] implies \( I \leq \exp\left(-c\epsilon + s(cM)^2/2\right) \). Choose \( c = \epsilon/(sM^2) \). Then,
\[
I \leq \exp\left(-\frac{\epsilon^2}{2sM^2}\right).
\]

Appendix B

In this part we assemble some lemmas, which were used to establish the sample error bounds for the ERM algorithm associated with the networked training sample. The main ideas were borrowed from \cite{Cucker:2007}.

Denote the defect function \( D_Z\mathcal{E}(f) = \mathcal{E}(f) - \mathcal{E}_Z(f) \). Then the following lemma follows directly from the third inequality in Theorem \[16\] by taking \( X_i = -(f(x) - y)^2 \) satisfying \(|X_i| \leq M^2 \) when \( f \) is M-bounded.

Lemma 28 Let \( M > 0 \) and \( f : \mathcal{X} \mapsto \mathcal{Y} \) be \( M \)-bounded. Then for all \( \epsilon > 0 \),
\[
\Pr \left( D_{Z_i}(f) \geq -\epsilon \right) \geq 1 - \exp \left( -\frac{s\epsilon^2}{2M^4} \right).
\]

Lemma 29 Let \( \mathcal{H} \) be a compact \( M \)-bounded subset of \( C(\mathcal{X}) \). Then, for all \( \epsilon > 0 \),
\[
\Pr \left( \sup_{f \in \mathcal{H}} D_{Z_i}(f) \leq \epsilon \right) \geq 1 - \mathcal{N}(\mathcal{H}, \frac{\epsilon}{8M}) \exp \left( -\frac{s\epsilon^2}{8M^4} \right).
\]

Proof Let \( \{f_j\}_{j=1}^\ell \subset \mathcal{H} \) with \( \ell = \mathcal{N}(\mathcal{H}, \frac{\epsilon}{4M}) \) such that \( \mathcal{H} \) is covered by disks \( D_j \) centered at \( f_j \) with radius \( \frac{\epsilon}{4M} \). Let \( U \) be a full measure set on which \( \sup_{f \in \mathcal{H}} |f(x) - y| \leq M \). Then for all \( Z \in U^n \) and for all \( f \in D_j \), there holds
\[
|D_{Z_i}(f) - D_{Z_i}(f_j)| \leq 4M\|f - f_j\|_\infty \leq 4M\frac{\epsilon}{4M} = \epsilon.
\]
Consequently,
\[
\sup_{f \in D_j} D_{Z_i}(f) \geq 2\epsilon \Rightarrow D_{Z_i}(f_j) \geq \epsilon.
\]
Then we conclude that, for \( j = 1, \cdots, \ell \),
\[
\Pr \left\{ \sup_{f \in D_j} D_{Z_i}(f) \geq 2\epsilon \right\} \leq \Pr \left\{ D_{Z_i}(f_j) \geq 2\epsilon \right\} \leq \exp \left\{ -\frac{s\epsilon^2}{2M^4} \right\}
\]
Learning from networked examples

the last inequality follows by taking $X_i = -(f(x) - y)^2$ on $Z$. In addition, one can easily see that

$$\sup_{f \in \mathcal{H}} \mathcal{D}_Z(f) \geq \epsilon \Leftrightarrow \exists j \leq \ell \text{ s.t. } \sup_{f \in \mathcal{D}_j} \mathcal{D}_Z(f) \geq \epsilon$$

and the fact that the probability of a union of events is bounded by the sum of the probabilities of these events. Hence

$$\Pr \left( \sup_{f \in \mathcal{H}} \mathcal{D}_Z(f) \geq \epsilon \right) \leq \sum_{j=1}^{\ell} \Pr \left( \sup_{f \in \mathcal{D}_j} \mathcal{D}_Z(f) \geq \epsilon \right) \leq \ell \exp \left( -\frac{s\epsilon^2}{8M^4} \right).$$

This completes our proof.

Lemma 30 Suppose a random variable $X_i$ satisfies $\mathbb{E}(X_i) = \mu \geq 0$, an $|X_i - \mu| \leq B$ almost everywhere. If $\mathbb{E}(X_i^2) \leq c\mathbb{E}(X_i)$, then for every $\epsilon > 0$ and $0 < \alpha \leq 1$, there holds

$$\Pr_Z \left\{ \frac{\mu - \frac{1}{\epsilon} \sum_{i=1}^{N} w_i X_i}{\sqrt{\mu + \epsilon}} > \alpha \sqrt{\epsilon} \right\} \leq \exp \left\{ -\frac{\alpha^2 \epsilon}{2c + \frac{4}{3} B} \right\}.$$  

Proof The lemma follows directly from the second statement of Theorem 16.

Lemma 31 can also be extended to families of functions as follows.

Lemma 31 Let $\mathcal{G}$ be a set of functions on $Z$ and $c > 0$ such that, for each $g \in \mathcal{G}$, $\mathbb{E}(g) \geq 0$, $\mathbb{E}(g^2) \leq c\mathbb{E}(g)$ and $|g - \mathbb{E}(g)| \leq B$ almost everywhere. Then for every $\epsilon > 0$ and $0 < \alpha \leq 1$, we have

$$\Pr_Z \left\{ \sup_{g \in \mathcal{G}} \frac{\mathbb{E}(g) - \mathbb{E}_Z(g)}{\sqrt{\mathbb{E}(g) + \epsilon}} > 4\alpha \sqrt{\epsilon} \right\} \leq \mathcal{N}(\mathcal{G}, \alpha \epsilon) \exp \left\{ -\frac{\alpha^2 \epsilon}{2c + \frac{4}{3} B} \right\}.$$  

Proof The proof of this lemma is similar to that of Lemma 29. Details for the proof can also be found in Chapter 3 of Cucker and Zhou (2007).

Denote $\mathcal{L}_2^\rho(\mathcal{X})$ as a Banach space with the norm $\|f\|_{\mathcal{L}_2^\rho(\mathcal{X})} = \left( \int_{\mathcal{X}} |f(x)|^2 d\rho(x) \right)^{\frac{1}{2}}$.

Lemma 32 Cucker and Zhou (2007) Let $\mathcal{H}$ be a convex subset of $C(\mathcal{X})$ such that $f_\mathcal{H}$ exists. Then $f_\mathcal{H}$ is unique as an element in $\mathcal{L}_2^\rho(\mathcal{X})$ and for all $f \in \mathcal{H}$,

$$\int_{\mathcal{X}} (f_\mathcal{H}(x) - f(x))^2 \rho(x) dx \leq E_\mathcal{H}(f).$$

In particular, if $\rho(x)$ is not degenerate then $f_\mathcal{H}$ is unique in $\mathcal{H}$.