Bounding Generalization Error Through Bias and Capacity

Ramya Ramalingam
Dept. of Computer and Info. Science
University of Pennsylvania
Philadelphia, PA, USA
ramya23@seas.upenn.edu

Nicolas Espinosa Dice
AMISTAD Lab
Harvey Mudd College
Claremont, CA, USA
nespinosadice@hmc.edu

Megan L. Kaye
AMISTAD Lab
Harvey Mudd College
Claremont, CA, USA
mkaye@hmc.edu

George D. Montañez
AMISTAD Lab
Harvey Mudd College
Claremont, CA, USA
gmontanez@hmc.edu

***Abstract*—We derive generalization bounds on learning algorithms through algorithm capacity and a vector representation of inductive bias. Leveraging the algorithmic search framework, a formalism for casting machine learning as a type of search, we present a unified interpretation of the upper bounds of generalization error in terms of a vector representation of bias and the mutual information between the hypothesis and the dataset.**

***Index Terms*—generalization error; bias; algorithm capacity; inductive orientation;**

I. Introduction

Bias is a crucial component in the analysis of machine learning algorithms. Unbiased algorithms cannot outperform uniform random sampling in expectation [1]. Recent work has established an information-usage framework to quantify and bound the bias produced through data exploration [2], [3]. Xu and Raginsky [4] extended Russo and Zou's [2] work on bias due to data exploration to provide an information-theoretic perspective on the generalization capabilities of learning algorithms. Specifically, they developed generalization error bounds through the mutual information of the algorithm's inputs and outputs.

An alternative framework for understanding learning algorithms is the *algorithmic search framework* [5]. The algorithmic search framework is a formalism for casting machine learning as a type of search. It allows for a quantitative measure of algorithmic bias to be defined [1]. Its measure of algorithmic bias introduced a geometric representation of inductive bias [6], [7]. This representation is referred to as the *inductive orientation* vector of an algorithm, a vector that can be used to quantify algorithmic bias.

Additionally, Bashir et al. [7] used the algorithmic search framework to present an information-theoretic perspective on overfitting and underfitting in machine learning algorithms. Overfitting and underfitting are defined in terms of an *algorithm's capacity* via the information transferred from training datasets to models. Moreover, Bashir et al. defined algorithm capacity in terms of the same search framework and derived upper bounds on algorithm capacity, showing how high capacity models can overfit [7]. Segura Sandoval et al. [8] developed a tool for estimating algorithm capacity within the same framework.

The algorithmic search framework has also been used to relate bias and algorithm flexibility, defined in the framework as *entropic expressivity*, and establish the trade-off between the two quantities [6], [9]. In addition to being a unified framework applicable to different learning problems, the framework offers the pedagogical benefit of framing machine learning as search, which is often more easily understood and more intuitive than information theoretic approaches.

The algorithmic search framework is applicable to many machine learning problems. Specifically, Montañez [10] showed that Vapnik's generalized learning problem [11], which is applicable to classification, regression, and density estimation, can be reduced to an algorithmic search problem. Montañez also showed that classification, clustering, parameter estimation, and hyperparameter optimization problems can be reduced to an algorithmic search problem [10]. Consequently, the algorithmic search framework is a useful framework for understanding machine learning problems generally.

A. Contributions

The algorithmic search framework, while generally useful, lacks a method for deriving generalization bounds, a necessary component for understanding supervised learning algorithms. We close this gap by introducing generalization bounds into the algorithmic search framework for the first time. We do this by combining the work of Montañez [5], the bounds on an algorithm's capacity developed by Bashir et al. [7], and the bounds on the generalization error of learning algorithms developed by Xu, Raginsky, Russo and Zou [3], [4]. We derive the bounds in terms of algorithm capacity and the inductive orientation vector. We show that an algorithm's generalization error can be upper-bounded in terms of distributional algorithm capacity, which measures the mutual information between the hypothesis and the dataset, and the inductive orientation vector. Lastly, our paper helps unify information-theoretic results with a geometric representation of bias.

Section II reviews the algorithmic search framework. Section III defines algorithm capacity and the algorithm bias in terms of the inductive orientation vector. Section IV presents the theorems defining generalization error bounds in terms of algorithm capacity and algorithm bias. Finally, Section V

presents examples of the bounds being used in overfitting and underfitting cases.

II. The Search Framework

Before proceeding with the main results, we review the algorithmic search framework, which is used to construct definitions of algorithmic capacity, algorithmic bias, and the inductive orientation vector of an algorithm.

A. The Search Problem

Following Montañez, we cast machine learning problems as search problems using the algorithmic search framework [5]. Within the algorithmic search framework, search problems have a 3-tuple representation (Ω, T, F), where Ω is the finite search space that can be sampled,[1] T is the target set, and F is the external information resource. The target set T is the subset of elements in the search space that are being searched for. A target function that represents set T is defined as a $|T|$-hot vector of length $|\Omega|$ that specifies which elements of Ω are contained within T [1]. The external information resource F is defined to be a binary string that represents the initialization and querying information for the search. It guides the search process as it is used to evaluate queried points in $|\Omega|$ [1].

B. The Search Algorithm

An algorithmic search is a process that determines the querying of elements in Ω when provided with a search problem, a history of elements that have already been examined, and information resource evaluations [5]. The points that the search algorithm queries, along with the information resource evaluations, are indexed by time and added to the search history. The search is deemed successful if an element $\omega \in T$ is queried by the algorithm [5].

C. Measuring Performance

We measure a learning algorithm's performance by examining the expected per-query probability of success, as suggested in [5]. The expected per-query probability of success provides a measure of performance in relation to an algorithm's total probability of being successful, normalized. This normalization accounts for the fact that the number of sampling steps could vary with the algorithm that is used, affecting the total probability of success [1]. Furthermore, it also accounts for sampling procedures that could sample the same subset of points in the search space multiple times — such as in genetic algorithms [12], [13]. This measure naturally accommodates algorithms that have to weigh the benefits and drawbacks between exploration and exploitation.

The expected per-query probability of success is defined as

$$q(T, F) = \mathbb{E}_{\tilde{P},H}\left[\frac{1}{|\tilde{P}|}\sum_{i=1}^{|\tilde{P}|} P_i(\omega \in T) \;\middle|\; F\right], \qquad (1)$$

[1]The requirement that Ω be a finite search space is a mild limitation given that all algorithms run on physical computer hardware are limited by finite-precision numerical representations, with finite amounts of compute time. [10]

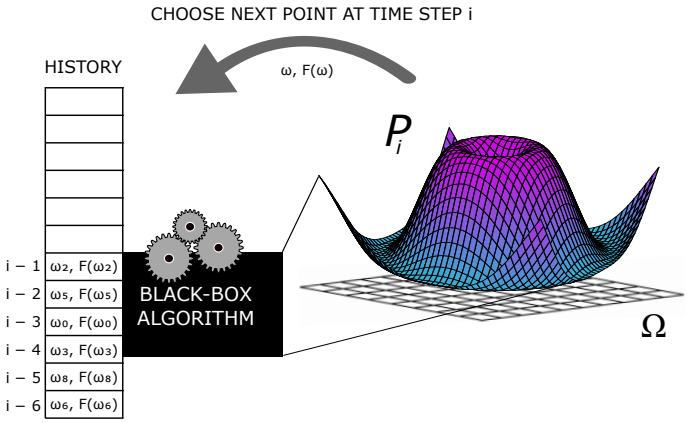

Fig. 1. A black-box search algorithm, as represented by Montañez [5]. First, the algorithm calculates a probability distribution P_i over the search space Ω at some time i. This distribution is computed using information from the algorithm's history. Then, using P_i, a new point is chosen and evaluated using F, the external information resource. The history is then updated at position i by adding the tuple $(\omega, F(\omega))$. The indices on the elements of ω in Fig. 1 correspond to sampled locations, not to time steps.

where $\tilde{P}$ is a sequence of probability distributions over the search space, where each timestep i produces a distribution $P_i \in \tilde{P}$, T is the target, F is the information resource, and H is the search history. The number of queries during a search is equal to the length of the probability distribution sequence, $|\tilde{P}|$ [5].

III. Preliminaries

Next, we present the definitions and quantitative measures necessary for the main results, beginning with the definition of the generalization error of a learning algorithm.

A. Generalization Error

Let $D = (Z_1, Z_2, \cdots, Z_n)$ be an input dataset of size n, with a probability distribution $\mathcal{D}$, and elements independently and identically drawn from a probability distribution $\mathcal{D}_\mathcal{Z}$ over an instance space $\mathcal{Z}$. Note that $\mathcal{D} = \mathcal{D}_\mathcal{Z}^{\otimes n}$. Using a stochastic map $\mathcal{P}_{G|D}$, let $\mathcal{A}$ be a learning algorithm that induces a distribution from which the hypothesis g is chosen out of a hypothesis space $\mathcal{G}$, when given an input D. Additionally, define a non-negative loss function $l : \mathcal{G} \times \mathcal{Z} \to \mathbb{R}_{\geq 0}$ which is used to calculate both empirical and population risk.

Definition 1 (Empirical Risk of a Hypothesis): For a specific training dataset D, the *empirical risk* $\hat{R}_D$ of a chosen hypothesis g is given by the average loss over all training set elements,

$$\hat{R}_D(g) = \frac{1}{n}\sum_{i=1}^{n} \ell(g, z_i). \qquad (2)$$

Definition 2 (Population Risk of a Hypothesis): The *population risk* of the hypothesis is the expected value of loss across all possible elements in the distribution $\mathcal{D}$, such that

$$R_\mathcal{D}(g) = \mathbb{E}_{\mathcal{D}_\mathcal{Z}}[\ell(g, Z)] = \int_\mathcal{Z} \ell(g, z) d\mathcal{D}_Z(z). \qquad (3)$$

Using the empirical and population risks of a hypothesis, we can define the generalization error of an algorithm.

Definition 3 (Generalization Error of an Algorithm): The *generalization error* of an algorithm $\mathcal{A}$ is defined as the expected difference between population risk and empirical risk, where expectation is taken over the joint distribution of datasets and hypotheses:

$$\text{gen}(\mathcal{D}, P_{G|D}) = \mathbb{E}[R_{\mathcal{D}}(G) - \hat{R}_D(G)]. \tag{4}$$

Algorithms with low generalization error allow us to tightly bound the population risk — which cannot directly be computed without knowledge of $\mathcal{D}$ — to empirical risk, which is typically what $\mathcal{A}$ uses to select a hypothesis from $\mathcal{G}$.

B. Algorithm Capacity

We will show that an algorithm's generalization error can be upper bounded in terms of distributional algorithm capacity, which measures the mutual information between the hypothesis G and the dataset D.

Definition 4 (Algorithm Capacity): Bashir et al. [7] define the *capacity $C_{\mathcal{A}}$ of an algorithm $\mathcal{A}$* to be the maximum amount of information that $\mathcal{A}$ can extract from a dataset $D \sim \mathcal{D}$, where $\mathcal{D}$ is an unknown distribution of that dataset, when selecting its output hypothesis g, namely,

$$\mathcal{C}_{\mathcal{A}} = \sup_{\mathcal{D}} I(G; D), \tag{5}$$

where G takes values in $\mathcal{G}$ and denotes the random variable representing the output of $\mathcal{A}$ with D as input.

Definition 5 (Distributional Algorithm Capacity): For a fixed distribution $\mathcal{D}$, Bashir et al. [7] define algorithm capacity relative to that particular distribution, called *distributional algorithm capacity*, to be

$$\mathcal{C}_{\mathcal{A},\mathcal{D}} = I(G; D), \tag{6}$$

for $D \sim \mathcal{D}$. This is the mutual information between an hypothesis G and the dataset D.

C. Inductive Orientation

Definition 6 (Inductive Orientation): Let F be an external information resource, such as a dataset, and let

$$\overline{\mathbf{P}}_F := \mathbb{E}_{\tilde{P},H}\left[\frac{1}{|\tilde{P}|}\sum_{i=1}^{|\tilde{P}|}\mathbf{P}_i \;\middle|\; F\right]. \tag{7}$$

That is, vector $\overline{\mathbf{P}}_F$ is the expected average conditional distribution on the search space given F. Bashir et al. [7] define the *inductive orientation* of an algorithm (relative to $\mathcal{D}$) to be

$$\overline{\mathbf{P}}_{\mathcal{D}} = \mathbb{E}_{F\sim\mathcal{D}}[\overline{\mathbf{P}}_F]. \tag{8}$$

D. Algorithmic Bias

Definition 7 (Algorithmic Bias): Following Montañez et al. [1], let $\mathcal{D}$ be a distribution over a space of information resources $\mathcal{F}$ and let $F \sim \mathcal{D}$. For a given $\mathcal{D}$ and a fixed k-hot target function $\mathbf{t}$,

$$\text{Bias}(\mathcal{D}, \mathbf{t}) = \mathbb{E}_{\mathcal{D}}[\mathbf{t}^\top \overline{\mathbf{P}}_F] - \frac{k}{|\Omega|}. \tag{9}$$

The algorithmic bias is the deviation in expected per-query probability of success of an algorithm from that of a uniform random sampler, relative to a distribution over information resources and a fixed target.

The inductive orientation vector is a vector representation of inductive bias, as $\overline{\mathbf{P}}_F$ is the geometric representation of biases relative to our external information resource F. Thus, $\overline{\mathbf{P}}_{\mathcal{D}}$ represents the expected value of the difference between different regions' probability masses placed by the algorithm.

Using the definition of the inductive orientation vector, Bashir et al. [7] showed that algorithmic bias can be defined in terms of the inductive orientation of an algorithm, such that

$$\text{Bias}(\mathcal{D}, \mathbf{t}) = \mathbf{t}^\top(\overline{\mathbf{P}}_{\mathcal{D}} - \mathbf{P}_{\mathcal{U}}), \tag{10}$$

where $\mathbf{P}_{\mathcal{U}} = \mathbf{1} \cdot |\mathcal{G}|^{-1}$, the inductive orientation vector for a uniform random sampler.

E. Entropic Expressivity

Definition 8 (Entropic Expressivity): Given a distribution over information resources $\mathcal{D}$, Lauw et al. [6] define the *entropic expressivity* of an algorithm to be the Shannon entropy of its inductive orientation vector,

$$H(\overline{\mathbf{P}}_{\mathcal{D}}) = H(\mathcal{U}) - D_{KL}(\overline{\mathbf{P}}_{\mathcal{D}} \,||\, \mathcal{U}), \tag{11}$$

where $D_{KL}(\overline{\mathbf{P}}_{\mathcal{D}} \,||\, \mathcal{U})$ is the Kullback-Leibler (KL) divergence between distribution $\overline{\mathbf{P}}_{\mathcal{D}}$ and the uniform distribution $\mathcal{U}$, and both are distributions over the search space Ω. Note, in the special case of the uniform distribution, the cross-entropy $H(\overline{\mathbf{P}}_{\mathcal{D}}, \mathcal{U}) = \mathbb{E}_{\overline{\mathbf{P}}_{\mathcal{D}}}[-\log_2 \mathcal{U}(X)] = H(\mathcal{U})$.

IV. Main Results

In this section, we derive generalization bounds in terms of algorithm capacity and the inductive orientation vector.

A. Generalization Bounds Through Algorithmic Capacity

We first derive a generalization error bound under distributional algorithm capacity for loss functions that are σ-subgaussian.

Theorem 1 (Generalization Error Bound Under Distributional Algorithm Capacity): When $D \sim \mathcal{D}$ and $\ell(g, D)$ is σ-subgaussian under $\mathcal{D}$ for all $g \in \mathcal{G}$, then

$$|\text{gen}(\mathcal{D}, P_{G|D})| \leq \sqrt{\frac{2\sigma^2}{n} \cdot C_{\mathcal{A},\mathcal{D}}}, \tag{12}$$

where $C_{\mathcal{A},\mathcal{D}}$ is the distributional algorithm capacity, defined in [7].

Recall that a variable is said to be σ-subgaussian if, for all $\sigma > 0$, the following inequality is satisfied

$$\mathbb{E}\left[e^{\lambda(X-\mathbb{E}[X])}\right] \leq e^{\frac{\lambda^2\sigma^2}{2}} \tag{13}$$

for all $\lambda \in \mathbb{R}$. A random variable which is bound in the interval (a, b) is subgaussian with $\sigma = \frac{b-a}{2}$. Thus, common loss functions such as the 0-1 loss function and the hinge loss function, which are used for support vector machines in multi-class classification problems, are always subgaussian. This is regardless of the hypothesis and training dataset distribution they are defined by.

Next, we provide a result for the generalization error in terms of the inductive orientation vector of an algorithm.

Corollary 1 (Generalization Error Bound Under Entropic Expressivity):

$$|\text{gen}(\mathcal{D}, P_{G|D})| \leq \sqrt{\frac{2\sigma^2}{n}\left(H(\overline{\mathbf{P}}_{\mathcal{D}}) - \mathbb{E}_{\mathcal{D}}[H(\overline{\mathbf{P}}_F)]\right)}, \tag{14}$$

where $H(\overline{\mathbf{P}}_{\mathcal{D}})$ and $\mathbb{E}_{\mathcal{D}}[H(\overline{\mathbf{P}}_F)]$ are the entropic expressivity and the expected entropic expressivity of $\mathcal{A}$, respectively. $\overline{\mathbf{P}}_F$ is the expected average conditional distribution on the search space given an external information resource F, and $\overline{\mathbf{P}}_{\mathcal{D}}$ is the inductive orientation of the algorithm.

We see that Theorem 1 relates the generalization error to the algorithm capacity, which is defined in terms of the mutual information between the hypothesis and the dataset. Corollary 1 relates the generalization error to the inductive orientation vector. The generalization error can also be interpreted in terms of algorithmic bias, which is shown below.

Corollary 2 (Generalization Error Bound Under Algorithmic Bias): For a classification problem in the algorithmic framework with fixed target function $\mathbf{t}$,

$$|\text{gen}(\mathcal{D}, P_{G|D})| \leq \sqrt{\frac{2\sigma^2}{n}B}, \tag{15}$$

where

$$B := \log_2|\mathcal{G}| - 2\text{Bias}(\mathcal{D}, \mathbf{t})^2 - \mathbb{E}_{\mathcal{D}}[H(\overline{\mathbf{P}}_F)]. \tag{16}$$

Additionally, we present a bound on the generalization error using KL divergence.

Corollary 3 (Generalization Error Bound Under KL Divergence):

$$|\text{gen}(\mathcal{D}, P_{G|D})| \leq \sqrt{\frac{2\sigma^2}{n}\left(\sup_{\mathcal{D}}\left[D_{\text{KL}}(P_{G|D}||P_G)\right]\right)}. \tag{17}$$

B. Generalization Bounds For Additional Distributions

We now introduce a new random variable $\hat{L}$, which is a tuple of empirical risk values for all hypotheses $g \in \mathcal{G}$, such that

$$\hat{L} = (\hat{\ell}_1, \hat{\ell}_2, \ldots, \hat{\ell}_m) \tag{18}$$

$$= (\hat{R}_D(g_1), \hat{R}_D(g_2), \ldots, \hat{R}_D(g_m)), \tag{19}$$

where $m = |\mathcal{G}|$ is the total number of hypotheses in our hypothesis space. Notice that $\hat{L}$ is a function of $D \sim \mathcal{D}$, so

$$I(G; \hat{L}) \leq I(G; D) \tag{20}$$

$$= C_{\mathcal{A},\mathcal{D}} \tag{21}$$

by the Data-Processing Inequality [14]. We use $\hat{L}$ to upper bound generalization error for other distributions of the loss function.

Theorem 2 (Generalization Bound Under Algorithmic Capacity for σ_i Distributions): If the input dataset D is drawn from a distribution $\mathcal{D}$, such that $\ell(g_i, D)$ is σ_i-subgaussian under $\mathcal{D}$ for all $1 \leq i \leq m$, then

$$|\text{gen}(\mathcal{D}, P_{G|D})| \leq \sqrt{\frac{\mathbb{E}[\sigma_G^2]}{n} \cdot 2C_{\mathcal{A},\mathcal{D}}}. \tag{22}$$

Notice that Theorem 1 can be obtained from Theorem 2 by setting $\sigma_i = \sigma$ for all $1 \leq i \leq m$. Using the upper bounds on $C_{\mathcal{A},\mathcal{D}}$ that were proved in [7] and referred to in Corollaries 1, 2, and 3, we may obtain similar corollaries for Theorem 2.

Corollary 4 (Generalization Bound Under Entropic Expressivity for σ_i Distributions):

$$|\text{gen}(\mathcal{D}, P_{G|D})| \leq \sqrt{2\left(H(\overline{\mathbf{P}}_{\mathcal{D}}) - \mathbb{E}_{\mathcal{D}}[H(\overline{\mathbf{P}}_F)]\right)}\sqrt{\frac{\mathbb{E}[\sigma_G^2]}{n}}. \tag{23}$$

Corollary 5 (Generalization Bound Under KL Divergence for σ_i Distributions):

$$|\text{gen}(\mathcal{D}, P_{G|D})| \leq \sqrt{2\sup_{\mathcal{D}}\left(D_{\text{KL}}(P_{G|D}||P_G)\right)}\sqrt{\frac{\mathbb{E}[\sigma_G^2]}{n}}. \tag{24}$$

Theorem 3 (Generalization Bound for Sub-exponential Distributions): If the input dataset D is drawn from a distribution $\mathcal{D}$, such that $\ell(g, D)$ is sub-exponential with parameters (σ, b) under $\mathcal{D}$ for all $g \in \mathcal{G}$, then

$$\text{gen}(\mathcal{D}, P_{G|D}) \leq b \cdot C_{\mathcal{A},\mathcal{D}} + \frac{\sigma^2}{2nb}. \tag{25}$$

Recall that a variable X is said to be *sub-exponential* with parameters (σ, b), where $b > 0$, if (13) is satisfied for all $|\lambda| < \frac{1}{b}$. By definition, all subgaussian variables are sub-exponential, so the 0-1 and hinge loss functions are examples of sub-exponential loss functions.

V. Examples

We now consider applications of the generalization bounds proved above.

A. Underfitting Case

To demonstrate the effect of an underfitting algorithm, let us define $\mathcal{A}$, which induces a fixed distribution over the hypotheses in $\mathcal{G}$. Upon receiving a dataset $D = (Z_1, Z_2, \cdots, Z_n)$, where $Z_i \in \mathcal{Z}$, it does not update its distribution but selects a hypothesis based on the same fixed distribution as before. Since the input training dataset has no effect on the distribution over hypotheses, i.e., $P(G|D) = P(G)$, the mutual information transferred between any given dataset and the algorithm is

zero. That is, $I(G;D) = 0$ for all datasets D. By definition of algorithmic capacity and distributional algorithmic capacity,

$$\mathcal{C}_{\mathcal{A},\mathcal{D}} = I(G;D) = 0, \text{ and} \tag{26}$$

$$\mathcal{C}_{\mathcal{A}} = \sup_{\mathcal{D}} I(G;D) = 0. \tag{27}$$

Since the distribution of G does not change over time, the time-indexed capacity $\mathcal{C}^i_{\mathcal{A}} = 0$ for all timesteps i.

This is canonically an underfitting algorithm by the definition of underfitting proposed in [7], which states that an algorithm underfits at iteration i if

$$\mathcal{C}^i_{\mathcal{A}} < \mathbb{E}[C_D] \tag{28}$$

where $C_D = \min(C_{D,M}, C'_D)$. $C_{D,M}$ measures the length of the shortest program which correctly maps every input in D to the correct output, and C'_D measures the number of bits required to memorize D. For any non-empty dataset D both these values must be positive, and so their minimum is also positive. Therefore,

$$\mathcal{C}^i_{\mathcal{A}} = 0 < \mathbb{E}[C_D] \tag{29}$$

at all timesteps i, so $\mathcal{A}$ underfits throughout all iterations of the algorithm.

Using our theorems for generalization error, all of which involve an upper bound with $\mathcal{C}_{\mathcal{A},\mathcal{D}}$ as a factor, we see that for any distribution $\mathcal{D}$ across datasets and any loss function satisfying the required criteria,

$$|\text{gen}(\mathcal{D}, P_{G|D})| \leq 0, \tag{30}$$

which implies that

$$|\text{gen}(\mathcal{D}, P_{G|D})| = 0. \tag{31}$$

While this may seem promising, notice that in minimizing the generalization error, the algorithm has neglected any information possibly provided by D. That is, $\mathcal{A}$ has no control over empirical risk, since it selects a hypothesis without taking into consideration loss over D, and thus no control over population risk despite generalization error being zero.

B. Overfitting Case

To demonstrate the effect of an overfitting algorithm, let us define $\mathcal{A}$ for a binary classification task, i.e., each training example is a k-element vector which is classified as 0 or 1. Upon receiving a dataset $D = (Z_1, Z_2, \ldots, Z_n)$, where $\mathcal{Z}$ is the instance space with probability distribution $\mathcal{D}_{\mathcal{Z}}$, and $Z_i \in \mathcal{Z}$. Algorithm $\mathcal{A}$ selects the hypothesis $G \in \mathcal{G}$ which minimizes absolute error, with a predetermined ordering across all hypotheses to break ties. Since we assume a finite hypothesis set, $\mathcal{A}$ places all probability mass on a single hypothesis once it is given a training dataset.

Consider any fixed dataset D. The mutual information transferred between the dataset and hypothesis random variables is defined as

$$I(G;D) = H(G) - H(G|D). \tag{32}$$

Assuming a uniform distribution across hypotheses prior to conditioning on D, the entropy of G is:

$$H(G) = \log_2 |\mathcal{G}|. \tag{33}$$

Since D induces a distribution with no uncertainty and all probability mass lies on a single hypothesis,

$$H(G|D) = 0. \tag{34}$$

Therefore, for any distribution across datasets $\mathcal{D}$,

$$\begin{aligned} \mathcal{C}_{\mathcal{A},\mathcal{D}} &= I(G;D) && (35)\\ &= H(G) - H(G|D) && (36)\\ &= \log_2 |\mathcal{G}|. && (37) \end{aligned}$$

In order to overfit, the algorithmic capacity $\mathcal{C}_{\mathcal{A},\mathcal{D}}$ must be greater than $\mathbb{E}[C_D]$, where $C_D = \min(C_{D,M}, C'_D)$. Let C'_D measure the number of bits required to memorize D without compression. Assuming each feature in a single training example requires one bit to store its value, the entire n-element dataset would require at most $n(k+1)$ bits to memorize — that is, to store feature vectors along with their corresponding binary classifications — without compression. Thus,

$$\begin{aligned} \mathbb{E}[C_D] &\leq \mathbb{E}[C'_D] && (38)\\ &\leq n(k+1). && (39) \end{aligned}$$

In our example, let $|\mathcal{G}| = 2^{100}$, and because we are seeking to demonstrate overfitting, let $n = 10$ and $k = 5$. Then,

$$\log_2 |\mathcal{G}| = 100 > 60 = n(k+1), \tag{40}$$

so

$$\mathcal{C}_{\mathcal{A},\mathcal{D}} > \mathbb{E}[C_D] \tag{41}$$

for all datasets D. Thus, $\mathcal{A}$ is an overfitting algorithm for these specific values.

Since this is a binary classification task, we use a 0-1 loss function ℓ. A random variable bounded within $[a,b]$ is $\frac{(b-a)}{2}$-subgaussian. Therefore, $\ell(g,D)$ is σ-subgaussian under $\mathcal{D}$ for all $g \in \mathcal{G}$, where $\sigma = \frac{1}{2}$. By Theorem 1, we can bound the algorithm's generalization error,

$$\begin{aligned} |\text{gen}(\mathcal{D}, P_{G|D})| &\leq \sqrt{\frac{2\sigma^2}{n} \cdot C_{\mathcal{A},\mathcal{D}}} && (42)\\ &= \sqrt{\frac{2}{10} \cdot \frac{1}{4} \cdot 100} && (43)\\ &= \sqrt{5}. && (44) \end{aligned}$$

Notice that since loss is bounded above by one, we already know that the generalization error must be bounded above by one. Thus, in this sort of extreme overfitting situation, these bounds do not give us any useful information about how low generalization error must be. This aligns with how the algorithm works: the algorithm prioritizes minimization of empirical risk by always selecting the hypothesis which minimizes loss for the given training dataset. However, in doing so, we risk a higher generalization error, which in turn affects our uncertainty of the overall population risk.

C. Standard Case

We consider an algorithm similar to that used in the overfitting example: a binary classification algorithm which selects a hypothesis which minimizes absolute training error. However, we change the size of the training dataset as well as the number of features of each training example, and add a component of domain knowledge, which allows us to disregard certain hypotheses.

Suppose that $n = 10^6$ and $k = 30$, where n is the number of examples and k is the number of features of each example. The maximum possible number of vectors in this input space is 2^{30}, so the maximum possible number of hypotheses is $2^{2^{30}}$ since each vector has a binary classification. However, suppose that by using domain-specific knowledge, the algorithm can pare down the number of feasible hypotheses to $2^{2^{19}}$. For example, the algorithm may do this by eliminating illogical predictions using domain knowledge about the co-occurrence of certain feature values. Consequently, $|\mathcal{G}| = 2^{2^{19}}$.

As in the preceding algorithm, a 0-1 loss function ℓ is used, and $\ell(g, D)$ is σ-subgaussian under $\mathcal{D}$ for all $g \in \mathcal{G}$, where $\sigma = \frac{1}{2}$.

The mutual information transferred between the dataset and hypothesis random variables is

$$I(G; D) = H(G) - H(G|D). \tag{45}$$

We again assume a uniform distribution across feasible hypotheses prior to conditioning on D, so the entropy of G is given by

$$H(G) = \log_2 |\mathcal{G}|. \tag{46}$$

and

$$H(G|D) = 0. \tag{47}$$

Therefore, for any distribution across training datasets $\mathcal{D}$,

$$\mathcal{C}_{\mathcal{A},\mathcal{D}} = \log_2 |\mathcal{G}|. \tag{48}$$

By Theorem 1, the algorithm's generalization error can therefore be bound,

$$|\text{gen}(\mathcal{D}, P_{G|D})| \leq \sqrt{\frac{2\sigma^2}{n} \cdot C_{\mathcal{A},\mathcal{D}}} \tag{49}$$

$$= \sqrt{\frac{2}{10^6} \cdot \frac{1}{2^2} \cdot 2^{19}} \tag{50}$$

$$= \sqrt{\frac{2^{12}}{5^6}} \tag{51}$$

$$= 0.512. \tag{52}$$

VI. Conclusion

Using the information-theoretic bounds of Xu and Raginsky [4] and Russo and Zou [3], we introduce generalization bounds into the algorithmic search framework for the first time. We relate the generalization error of learning algorithms to algorithm capacity, inductive orientation, entropic expressivity, and KL divergence under subgaussian and sub-exponential loss functions.

We employ the bounds in examples of overfitting and underfitting. The underfitting example yields intuitive results. Through analysis of the generalization bounds, we can see that the generalization error is upper bounded by zero, implying that it is equal to zero. In the extreme overfitting example, the resulting bound is not any tighter than the general upper bound on loss. However, for more representative algorithms, which are not clearly underfitting nor overfitting, our results can be used to obtain non-trivial bounds on generalization error.

The addition of generalization bounds to the algorithmic search framework furthers its promise as a formalism for understanding machine learning problems. Future work in the algorithmic search framework may include analyzing generalization bounds under additional distributions and loss functions. With ongoing research into methods of estimating algorithm capacity [15] and inductive orientation empirically [8], [16], the algorithmic search framework may also become a useful tool for applied researchers. Future work may include leveraging algorithms for estimation of inductive orientation [16] to empirically study the efficacy of the given bounds. Furthermore, problems such as constraint satisfaction may be reduced into an algorithmic search problem, increasing the scope of the framework. Finally, additional information-theoretic approaches for understanding machine learning algorithms have arisen, leading to improved generalization bounds [17]–[20]. Following the approach of our paper, these could be used to derive stronger generalization bounds within the search framework as well.

References

[1] G. D. Montañez, J. Hayase, J. Lauw, D. Macias, A. Trikha, and J. Vendemiatti, "The Futility of Bias-Free Learning and Search," in *Australasian Joint Conference on Artificial Intelligence*. Springer, 2019, pp. 277–288.

[2] D. Russo and J. Zou, "Controlling Bias in Adaptive Data Analysis Using Information Theory," in *Artificial Intelligence and Statistics*. PMLR, 2016, pp. 1232–1240.

[3] ——, "How Much Does Your Data Exploration Overfit? Controlling Bias via Information Usage," *IEEE Transactions on Information Theory*, vol. 66, no. 1, pp. 302–323, 2020.

[4] A. Xu and M. Raginsky, "Information-theoretic analysis of generalization capability of learning algorithms," in *Advances in Neural Information Processing Systems*, 2017, pp. 2524–2533.

[5] G. D. Montañez, "The Famine of Forte: Few Search Problems Greatly Favor Your Algorithm," in *Systems, Man, and Cybernetics (SMC), 2017 IEEE International Conference on*. IEEE, 2017, pp. 477–482.

[6] J. Lauw, D. Macias, A. Trikha, J. Vendemiatti, and G. D. Montañez, "The Bias-Expressivity Trade-off," in *Proceedings of the 12th International Conference on Agents and Artificial Intelligence, ICAART 2020, Volume 2, Valletta, Malta, February 22-24, 2020*, A. P. Rocha, L. Steels, and H. J. van den Herik, Eds. SCITEPRESS, 2020, pp. 141–150. [Online]. Available: https://doi.org/10.5220/0008959201410150

[7] D. Bashir, G. D. Montañez, S. Sehra, P. Sandoval Segura, and J. Lauw, "An Information-Theoretic Perspective on Overfitting and Underfitting," *Australasian Joint Conference on Artificial Intelligence (AJCAI 2020)*, 2020.

[8] P. Sandoval Segura, J. Lauw, D. Bashir, K. Shah, S. Sehra, D. Macias, and G. D. Montañez, "The Labeling Distribution Matrix (LDM): A Tool for Estimating Machine Learning Algorithm Capacity," in *Proceedings of the 12th International Conference on Agents and Artificial Intelligence, ICAART 2020, Volume 2, Valletta, Malta, February 22-24, 2020*, A. P. Rocha, L. Steels, and H. J. van den Herik, Eds. SCITEPRESS, 2020, pp. 980–986. [Online]. Available: https://doi.org/10.5220/0009178209800986

[9] G. D. Montañez, D. Bashir, and J. Lauw, "Trading Bias for Expressivity in Artificial Learning," in *Agents and Artificial Intelligence*, A. P. Rocha, L. Steels, and J. van den Herik, Eds. Cham: Springer International Publishing, 2021, pp. 332–353.
[10] G. D. Montanez, "Why Machine Learning Works," *URL https://www.cs.hmc.edu/~montanez/montanez_dissertation.pdf*, 2017.
[11] V. N. Vapnik, "An Overview of Statistical Learning Theory," *IEEE transactions on neural networks*, vol. 10, no. 5, pp. 988–999, 1999.
[12] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, 1989.
[13] C. Reeves and J. E. Rowe, *Genetic Algorithms: Principles and Perspectives: A Guide to GA Theory*. Springer Science & Business Media, 2002, vol. 20.
[14] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. John Wiley & Sons, 1991.
[15] K. Rong, A. Khant, D. Flores, and G. D. Montañez, "The Label Recorder Method: Testing the Memorization Capacity of Machine Learning Models," in *The Seventh International Conference on Machine Learning, Optimization, and Data Science (LOD 2021)*, 2021.
[16] S. Bekerman, E. Chen, L. Lin, and G. Montañez, "Vectorization of bias in machine learning algorithms," in *Proceedings of the 14th International Conference on Agents and Artificial Intelligence - Volume 2: ICAART*, INSTICC. SciTePress, 2022, pp. 354–365.
[17] T. Steinke and L. Zakynthinou, "Reasoning about generalization via conditional mutual information," in *Conference on Learning Theory*. PMLR, 2020, pp. 3437–3452.
[18] M. Haghifam, J. Negrea, A. Khisti, D. M. Roy, and G. K. Dziugaite, "Sharpened generalization bounds based on conditional mutual information and an application to noisy, iterative algorithms," *Advances in Neural Information Processing Systems*, vol. 33, pp. 9925–9935, 2020.
[19] Y. Bu, S. Zou, and V. V. Veeravalli, "Tightening mutual information-based bounds on generalization error," *IEEE Journal on Selected Areas in Information Theory*, vol. 1, no. 1, pp. 121–130, 2020.
[20] G. Neu, "Information-theoretic generalization bounds for stochastic gradient descent," in *Conference on Learning Theory*. PMLR, 2021, pp. 3526–3545.

APPENDIX

Theorem 1 (Generalization Error Bound Under Distributional Algorithm Capacity): If the input dataset D is drawn from a distribution $\mathcal{D}$, such that $\ell(g, D)$ is σ-subgaussian under $\mathcal{D}$ for all $g \in \mathcal{G}$, then

$$|\text{gen}(\mathcal{D}, P_{G|D})| \leq \sqrt{\frac{2\sigma^2}{n} \cdot C_{\mathcal{A},\mathcal{D}}}, \tag{53}$$

where $C_{\mathcal{A},\mathcal{D}}$ is the distributional algorithm capacity, defined in [7].

Proof: By Theorem 1 in [4], if $\ell(g, D)$ is σ-subgaussian under $\mathcal{D}$ for all $g \in \mathcal{G}$, then

$$|\text{gen}(\mathcal{D}, P_{G|D})| \leq \sqrt{\frac{2\sigma^2}{n} \cdot I(D;G)}, \tag{54}$$

where $I(D;G)$ is the input-output mutual information of algorithm $\mathcal{A}$. As discussed in [7], $C_{\mathcal{A},\mathcal{D}} = I(G;D) = I(D;G)$ for a fixed distribution $\mathcal{D}$. Therefore, a direct substitution into (54) yields the desired inequality,

$$|\text{gen}(\mathcal{D}, P_{G|D})| \leq \sqrt{\frac{2\sigma^2}{n} \cdot C_{\mathcal{A},\mathcal{D}}}. \tag{55}$$

■

Corollary 1:

$$|\text{gen}(\mathcal{D}, P_{G|D})| \leq \sqrt{\frac{2\sigma^2}{n} \left(H(\overline{\mathbf{P}}_{\mathcal{D}}) - \mathbb{E}_{\mathcal{D}}[H(\overline{\mathbf{P}}_F)]\right)}, \tag{56}$$

where $H(\overline{\mathbf{P}}_{\mathcal{D}})$ and $\mathbb{E}_{\mathcal{D}}[H(\overline{\mathbf{P}}_F)]$ are the entropic expressivity and the expected entropic expressivity of $\mathcal{A}$, respectively.

Proof: Theorem 3 in [7] shows that

$$C_{\mathcal{A},\mathcal{D}} = H(\overline{\mathbf{P}}_{\mathcal{D}}) - \mathbb{E}_{\mathcal{D}}[H(\overline{\mathbf{P}}_F)]. \tag{57}$$

Applying (57) to (54) yields the desired inequality. ■

Corollary 2: For a classification problem in the algorithmic framework with fixed target function $\mathbf{t}$,

$$|\text{gen}(\mathcal{D}, P_{G|D})| \leq \sqrt{\frac{2\sigma^2}{n} B}, \tag{58}$$

where

$$B := \log_2 |\mathcal{G}| - 2\text{Bias}(\mathcal{D}, \mathbf{t})^2 - \mathbb{E}_{\mathcal{D}}[H(\overline{\mathbf{P}}_F)]. \tag{59}$$

Proof: Theorem 4 in [7] states that

$$C_{\mathcal{A},\mathcal{D}} \leq \log_2 |\mathcal{G}| - 2\text{Bias}(\mathcal{D}, \mathbf{t})^2 - \mathbb{E}_{\mathcal{D}}[H(\overline{\mathbf{P}}_F)]. \tag{60}$$

Applying (60) to Theorem 1 yields the desired inequality. ■

Corollary 3:

$$|\text{gen}(\mathcal{D}, P_{G|D})| \leq \sqrt{\frac{2\sigma^2}{n} \left(\sup_{\mathcal{D}} \left[D_{\text{KL}}(P_{G|D}||P_G)\right]\right)} \tag{61}$$

Proof: The proof follows directly from Theorem 5 in [7], which states that

$$C_{\mathcal{A},\mathcal{D}} \leq \sup_{\mathcal{D}} \left[D_{\text{KL}}(P_{G|D}||P_G)\right]. \tag{62}$$

Applying (62) to Theorem 1 yields the desired inequality. ■

Theorem 2 (Generalization Bound Under Distributional Algorithmic Capacity): If the input dataset D is drawn from a distribution $\mathcal{D}$, such that $\ell(g_i, D)$ is σ_i-subgaussian under $\mathcal{D}$ for all $1 \leq i \leq m$, then

$$|\text{gen}(\mathcal{D}, P_{G|D})| \leq \sqrt{\frac{\mathbb{E}[\sigma_G^2]}{n} \cdot 2C_{\mathcal{A},\mathcal{D}}}. \tag{63}$$

Proof: For each $i \in \{1, 2, \ldots, m\}$, we define $\ell_i = \mathbb{E}[\hat{\ell}_i]$. Notice that ℓ_i is simply the population risk of hypothesis g_i. That is,

$$L = (\ell_1, \ell_2, \ldots, \ell_m) \tag{64}$$
$$= (R_{\mathcal{D}}(g_1), R_{\mathcal{D}}(g_2), \ldots, R_{\mathcal{D}}(g_m)). \tag{65}$$

Consequently,

$$|\text{gen}(\mathcal{D}, P_{G|D})| = |\mathbb{E}[R_{\mathcal{D}}(G) - \hat{R}_D(G)]| \tag{66}$$
$$= |\mathbb{E}[\hat{\ell}_G - \ell_G]|. \tag{67}$$

Since $\ell(g_i, D)$ is σ_i-subgaussian under $\mathcal{D}$, it follows that

$$\hat{\ell}_i = \hat{R}_D(g_i) \tag{68}$$
$$= \frac{1}{n} \sum_{j=1}^{n} \ell(g_i, z_j) \tag{69}$$

is $\frac{\sigma}{\sqrt{n}}$-subgaussian under $\mathcal{D}$, since $\mathbb{E}[\hat{\ell}_i] = \frac{1}{n}\mathbb{E}[\ell(g_i, D)]$. Therefore, $\hat{\ell}_i - \ell_i$ is $\frac{\sigma_i}{\sqrt{n}}$-subgaussian for all $1 \leq i \leq m$,

since loss is a non-negative function, so $P(\hat{\ell}_i - \ell_i \leq \hat{\ell}_i) = 1$. Consequently, we can apply Proposition 8 in [3], so

$$|\mathbb{E}[\hat{\ell}_G - \ell_G]| \leq \sqrt{\mathbb{E}\left[\frac{\sigma_G^2}{n}\right]}\sqrt{2I(G;\hat{L})} \tag{70}$$

$$= \sqrt{\frac{\mathbb{E}[\sigma_G^2]}{n}}\sqrt{2I(G;\hat{L})} \tag{71}$$

$$\leq \sqrt{\frac{\mathbb{E}[\sigma_G^2]}{n}}\sqrt{2C_{\mathcal{A},\mathcal{D}}}, \tag{72}$$

using (21) for the final inequality. This gives us the desired bound,

$$|\text{gen}(\mathcal{D}, P_{G|D})| \leq \sqrt{\frac{\mathbb{E}[\sigma_G^2]}{n}}\sqrt{2C_{\mathcal{A},\mathcal{D}}}. \tag{73}$$

■

Notice that Theorem 1 can be obtained from Theorem 2 by setting $\sigma_i = \sigma$ for all $1 \leq i \leq m$. Using the upper-bounds on $C_{\mathcal{A},\mathcal{D}}$ that were proved in [7] and referred to in Corollaries 1, 2, and 3, we may obtain similar corollaries for Theorem 2.

Corollary 4:

$$|\text{gen}(\mathcal{D}, P_{G|D})| \leq \sqrt{2\left(H(\overline{\mathbf{P}}_{\mathcal{D}}) - \mathbb{E}_{\mathcal{D}}[H(\overline{\mathbf{P}}_F)]\right)}\sqrt{\frac{\mathbb{E}[\sigma_G^2]}{n}}. \tag{74}$$

Corollary 5:

$$|\text{gen}(\mathcal{D}, P_{G|D})| \leq \sqrt{2\sup_{\mathcal{D}}\left(D_{\text{KL}}(P_{G|D}||P_G)\right)}\sqrt{\frac{\mathbb{E}[\sigma_G^2]}{n}}. \tag{75}$$

Theorem 3: If the elements of the input dataset D are drawn from a distribution $\mathcal{D}$, such that $\ell(g, D)$ is sub-exponential with parameters (σ, b) under $\mathcal{D}$ for all $g \in \mathcal{G}$, then

$$\text{gen}(\mathcal{D}, P_{G|D}) \leq b \cdot C_{\mathcal{A},\mathcal{D}} + \frac{\sigma^2}{2nb}. \tag{76}$$

Proof: We use $\hat{L}$ and L as they are defined in Equations (18) and (64), respectively. If $\ell(g_i, D)$ is sub-exponential with parameters (σ, b), then $\hat{\ell}_i$ is sub-exponential with parameters $(\frac{\sigma}{\sqrt{n}}, b)$, and accordingly $\hat{\ell}_i - \ell_i$ is sub-exponential with parameters $(\frac{\sigma}{\sqrt{n}}, b)$, since loss is a non-negative function, so $P(\hat{\ell}_i - \ell_i \leq \hat{\ell}_i) = 1$. Consequently, we can apply Proposition 9 from [3], so

$$\mathbb{E}[\hat{\ell}_G - \ell_G] \leq b \cdot I(G;\hat{L}) + \frac{(\sigma/\sqrt{n})^2}{2b} \tag{77}$$

$$= b \cdot I(G;\hat{L}) + \frac{\sigma^2}{2nb} \tag{78}$$

$$\leq b \cdot C_{\mathcal{A},\mathcal{D}} + \frac{\sigma^2}{2nb}. \tag{79}$$

The left hand side may be equated to generalization error, so

$$\text{gen}(\mathcal{D}, P_{G|D}) \leq b \cdot C_{\mathcal{A},\mathcal{D}} + \frac{\sigma^2}{2nb}. \tag{80}$$

If $b < 1$, Proposition 9 promises a tighter bound,

$$\text{gen}(\mathcal{D}, P_{G|D}) \leq \sqrt{b} \cdot C_{\mathcal{A},\mathcal{D}} + \frac{\sigma^2}{2n\sqrt{b}}. \tag{81}$$

The corollaries due to upper-bounding $C_{\mathcal{A},\mathcal{D}}$ apply here as well. ■