# The Futility of Bias-Free Learning and Search

George D. Montañez(✉), Jonathan Hayase, Julius Lauw, Dominique Macias, Akshay Trikha, and Julia Vendemiatti

AMISTAD Lab, Harvey Mudd College, Claremont, CA 91711, USA
{gmontanez,jhayase,julauw,dmacias,atrikha,jvendemiatti}@hmc.edu

**Abstract.** Building on the view of machine learning as search, we demonstrate the necessity of bias in learning, quantifying the role of bias (measured relative to a collection of possible datasets, or more generally, information resources) in increasing the probability of success. For a given degree of bias towards a fixed target, we show that the proportion of favorable information resources is strictly bounded from above. Furthermore, we demonstrate that bias is a conserved quantity, such that no algorithm can be favorably biased towards many distinct targets simultaneously. Thus bias encodes trade-offs. The probability of success for a task can also be measured geometrically, as the angle of agreement between what holds for the actual task and what is assumed by the algorithm, represented in its bias. Lastly, finding a favorably biasing distribution over a fixed set of information resources is provably difficult, unless the set of resources itself is already favorable with respect to the given task and algorithm.

**Keywords:** Machine learning · Inductive bias · Algorithmic search

## 1 Introduction

Imagine you are on a routine grocery shopping trip and plan to buy some bananas. You know that the store carries both good and bad bananas which you must search through. There are multiple ways you can go about your search. One way is to randomly pick any ten bananas available on the shelf, which can be regarded as a form of unbiased search. Alternatively, you could introduce some bias to your search by only picking those bananas that are neither underripe nor overripe. Based on your past experiences from eating bananas, there is a better chance that these bananas will taste better. The proportion of good bananas retrieved in your biased search is greater than the same proportion in an unbiased search; you used your prior knowledge about tasty bananas. This

Supported by the NSF under Grant No. 1659805, Harvey Mudd College, and the Walter Bradley Center for Natural and Artificial Intelligence.

J. Hayase, J. Lauw, D. Macias, A. Trikha and J. Vendemiatti—Denotes equal contribution.

J. Liu and J. Bailey (Eds.): AI 2019, LNAI 11919, pp. 277–288, 2019.
https://doi.org/10.1007/978-3-030-35288-2_23

common routine shows how bias enables us to conduct more successful searches based on prior knowledge of the search target.

Viewing these decision-making processes through the lens of machine learning, we analyze how algorithms tackle learning problems under the influence of bias. Will we be better off without the existence of bias in machine learning algorithms? Our goal in this paper is to formally characterize the direct relationship between the performance of machine learning algorithms and their underlying biases. Without bias, machine learning algorithms will not perform better than uniform random sampling, on average. Yet to the extent an algorithm is biased toward some target is the extent to which it is biased against all remaining targets. As a consequence, no algorithm can be biased towards all targets. Therefore, bias represents the trade-offs an algorithm makes in how to respond to data.

We approach this problem by analyzing the performance of search algorithms within the algorithmic search framework introduced by Montañez [5]. This framework applies to common machine learning tasks such as classification, regression, clustering, optimization, reinforcement learning, and the general machine learning problems considered in Vapnik's learning framework [6]. We derive results characterizing the role of bias in successful search, extending Famine of Forte results [5] for a fixed search target and varying information resources. Our results for bias-free search then directly apply to bias-free learning, showing the extent to which bias is necessary for successful learning and quantifying how difficult it is to find a distribution with favorable bias for a particular target.

We should note that while bias formally measures how much an algorithm's predisposition towards a fixed outcome causes it's performance to deviate from that of uniform random sampling, we also use that term to refer to the underlying predisposition itself and its causes, which are responsible for that deviance.

## 2 Related Work

Schaffer's seminal work [11] showed that generalization performance for classification problems is a conserved quantity, such that favorable performance on a particular subset of problems will always be offset and balanced by poor performance over the remaining problems. Similarly, we show that bias is also a conserved quantity for any set of information resources. While Schaffer studied the performance of a single algorithm over different learning classes, Wolpert and Macready's "No Free Lunch Theorems for Optimization" [13] established that all optimization algorithms have the same performance when uniformly averaged over all possible cost functions. They also provided a geometric intuition for this result by defining an inner product which measures the alignment between an algorithm and a given prior over problems. This shows that no algorithm can be simultaneously aligned with all possible priors. In the context of the search framework, we define the geometric divergence as a measure of alignment between a search algorithm and a target in order to bound the proportion of favorable search problems.

While No Free Lunch Theorems are widely recognized as landmark ideas in machine learning, McDermott claims that No Free Lunch results are often misinterpreted and are practically insignificant for many real-world problems [3]. This is because algorithms are commonly tailored to a specific subset of problems in the real world, but No Free Lunch requires that we consider the set of all problems that are closed under permutation. These arguments against the applicability of No Free Lunch results are less relevant to our work here, since we evaluate the proportion of successful problems instead of considering the mean performance over the set of all problems. Furthermore, our results hold for sets of problems that are not closed under permutation, as a generalization of No Free Lunch results.

In "The Famine of Forte: Few Search Problems Greatly Favor Your Algorithm," Montañez [5] reduces machine learning problems to search problems and develops a rigorous search framework to generalize No Free Lunch ideas. He strictly bounds the proportion of problems that are favorable for a fixed algorithm and shows that no single algorithm can perform well over a large fraction of search problems. Extending these results to fixed search targets, we show that there are also strict bounds on the proportion of favorable information resources, and that the bound relaxes with the introduction of bias.

Our notion of bias relates to ideas introduced by Mitchell [4]. According to Mitchell, a completely unbiased classification algorithm cannot generalize beyond training data. He argued that the ability of a learning algorithm to generalize depends on incorporating biases, which equates to making assumptions beyond strict consistency with training data. These biases may include prior knowledge of the domain, preferences for simplicity, restrictions on algorithm structure, and awareness of the algorithm's real-world application. We strengthen Mitchell's argument with a mathematical justification for the need for bias in improving learning performance.

Gülçehre and Bengio empirically support Mitchell's ideas by investigating the nature of training barriers affecting the generalization performance of black-box machine learning algorithms [2]. Using the Structured Multi-Layer Perceptron (SMLP) neural network architecture, they showed that pre-training the SMLP with hints based on prior knowledge of the task generalizes more efficiently as compared to an SMLP pre-trained with random initializers. Furthermore, Ulyanov et al. explore the success of deep convolutional networks applied to image generation and restoration [12]. By applying untrained convolutional networks to image reconstruction with competitive success to trained ones, they show that the impressive performance of these networks is not due to learning alone. They highlight the importance of inductive bias, which is built into the structure of these generator networks, in achieving this high level of success. In a similar vein, Runarsson and Yao establish that bias is an essential component in constrained evolutionary optimization search problems [10]. It is experimentally shown that carefully selecting an appropriate constraint handling method and applying a biasing penalty function enhances the probability of locating feasible solutions for evolutionary algorithms. Inspired by the results obtained from these experimental studies, we formulate a theoretical validation of the role of bias in generalization performance for learning problems.

# 3 The Search Framework

## 3.1 The Search Problem

We formulate machine learning problems as search problems using the algorithmic search framework [5]. Within the framework, a search problem is represented as a 3-tuple $(\Omega, T, F)$. The finite search space from which we can sample is $\Omega$. The subset of elements in the search space that we are searching for is the target set $T$. A target function that represents $T$ is an $|\Omega|$-length vector with entries having value 1 when the corresponding elements of $\Omega$ are in the target set and 0 otherwise. The external information resource $F$ is a binary string that provides initialization information for the search and evaluates points in $\Omega$, acting as an oracle that guides the search process.

## 3.2 The Search Algorithm

Given a search problem, a history of elements already examined, and information resource evaluations, an algorithmic search is a process that decides how to query elements of $\Omega$. As the search algorithm samples, it adds the record of points queried and information resource evaluations, indexed by time, to the search history. If the algorithm queries an element $\omega \in T$ at least once during the course of its search, we say that the search is successful. Figure 1 visualizes the search algorithm.

## 3.3 Measuring Performance

Within this search framework, we measure a learning algorithm's performance by examining the expected per-query probability of success. This measure is more effective than measuring an algorithm's total probability of success, since the number of sampling steps may vary depending on the algorithm used, inflating the total probability for algorithms that sample more. Furthermore, the per-query probability of success naturally accounts for sampling procedures that involve repeatedly sampling the same points in the search space, as is the case of genetic algorithms [1,9]. Thus, this measure effectively handles search algorithms that attempt to manage trade-offs between exploration and exploitation.

The expected per-query probability of success is defined as

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

where $\tilde{P}$ is a sequence of probability distributions over the search space (where each timestep $i$ produces a distribution $P_i$), $T$ 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, $|\tilde{P}|$.

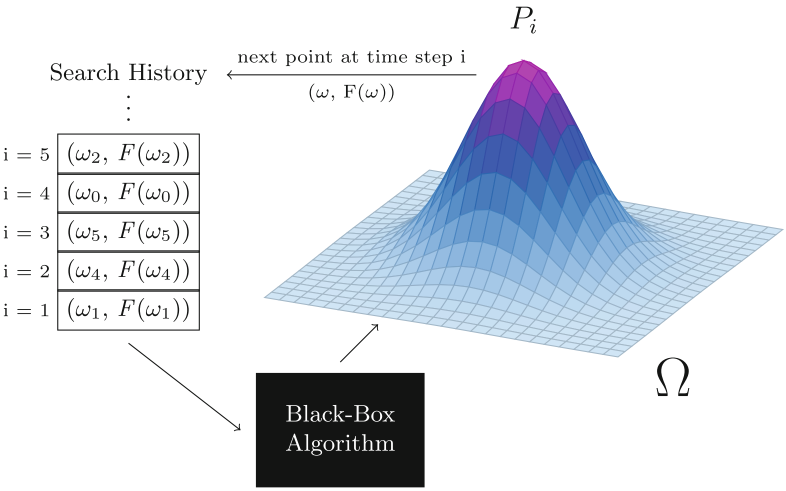

$P_i$
Search History
next point at time step i
$(\omega,\ \mathrm{F}(\omega))$
i = 5 $(\omega_2,\ F(\omega_2))$
i = 4 $(\omega_0,\ F(\omega_0))$
i = 3 $(\omega_5,\ F(\omega_5))$
i = 2 $(\omega_4,\ F(\omega_4))$
i = 1 $(\omega_1,\ F(\omega_1))$
$\Omega$
Black-Box Algorithm

**Fig. 1.** As a black-box optimization algorithm samples from $\Omega$, it produces an associated probability distribution $P_i$ based on the search history. When a sample $\omega_k$ corresponding to location $k$ in $\Omega$ is evaluated using the external information resource $F$, the tuple $(\omega_k,\ F(\omega_k))$ is added to the search history.

## 4 Main Results

We present and explain our main results in this section. Note that full proofs for the following results can be found in the Appendix (available online, on arXiv [7]). We proceed by defining our measures of bias and target divergence, then show conservation results of bias and give bounds on the probability of successful search and the proportion of favorable search problems given a fixed target.

**Definition 1** *(Bias for a distribution over information resources and a fixed target)*. *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 $\boldsymbol{t}$,*

$$\begin{aligned}\mathrm{Bias}(\mathcal{D}, \boldsymbol{t}) &= \mathbb{E}_{\mathcal{D}}\left[\boldsymbol{t}^\top \overline{P}_F\right] - \frac{k}{|\Omega|}\\ &= \boldsymbol{t}^\top \mathbb{E}_{\mathcal{D}}\left[\overline{P}_F\right] - \frac{\|\boldsymbol{t}\|^2}{|\Omega|}\\ &= \boldsymbol{t}^\top \int_{\mathcal{F}} \overline{P}_f \mathcal{D}(f)\,\mathrm{d}f - \frac{\|\boldsymbol{t}\|^2}{|\Omega|}\end{aligned}$$

*where $\overline{P}_f$ is the vector representation of the averaged probability distribution (conditioned on $f$) induced on $\Omega$ during the course of the search, which can be shown to imply $q(t, f) = \boldsymbol{t}^\top \overline{P}_f$.*

**Definition 2** *(Bias for a finite set of information resources and a fixed target)*. *Let $\mathcal{U}[\mathcal{B}]$ denote a uniform distribution over a finite set of information resources*

*$\mathcal{B}$. For a random quantity $F \sim \mathcal{U}[\mathcal{B}]$, the averaged $|\Omega|$-length simplex vector $\overline{P}_F$, and a fixed $k$-hot target function $\boldsymbol{t}$,*

$$\begin{aligned}\mathrm{Bias}(\mathcal{B}, \boldsymbol{t}) &= \mathbb{E}_{\mathcal{U}[\mathcal{B}]}[\boldsymbol{t}^\top \overline{P}_F] - \frac{k}{|\Omega|} \\ &= \boldsymbol{t}^\top \mathbb{E}_{\mathcal{U}[\mathcal{B}]}[\overline{P}_F] - \frac{k}{|\Omega|} \\ &= \boldsymbol{t}^\top \left( \frac{1}{|\mathcal{B}|} \sum_{f \in \mathcal{B}} \overline{P}_f \right) - \frac{\|\boldsymbol{t}\|^2}{|\Omega|}.\end{aligned}$$

We define bias as the difference between average performance of a search algorithm on a fixed target over a set of information resources and the baseline search performance for the case of uniform random sampling. Definition 1 is a generalized form of Definition 2, characterizing the alignment between a target function and a distribution over information resources instead of a fixed set.

**Definition 3** *(Target Divergence)***.** *The measure of similarity between a fixed target function $\boldsymbol{t}$ and the expected value of the averaged $|\Omega|$-length simplex vector $\overline{P}_F$, where $F \sim \mathcal{D}$, is defined as*

$$\theta = \arccos \left( \frac{\boldsymbol{t}^\top \mathbb{E}_{\mathcal{D}}[\overline{P}_F]}{\|\boldsymbol{t}\| \|\mathbb{E}_{\mathcal{D}}[\overline{P}_F]\|} \right)$$

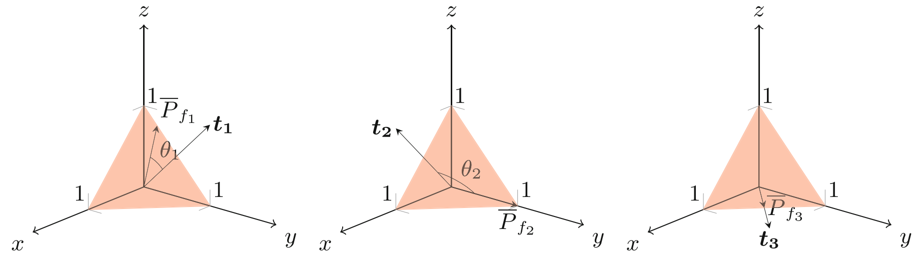
Similar to Wolpert and Macready's geometric interpretation of the No Free Lunch theorems [13], we can evaluate how far a target function $\boldsymbol{t}$ deviates from the averaged probability simplex vector $\overline{P}_f$ for a given search problem. We use cosine similarity to measure the level of similarity between $\boldsymbol{t}$ and $\overline{P}_f$. Geometrically, the target divergence is the angle between the target vector and the averaged $|\Omega|$-length simplex vector. Figure 2 depicts the target divergence for various levels of alignments between $\boldsymbol{t}$ and $\overline{P}_f$.

**Theorem 1 (Improbability of Favorable Information Resources).** *Let $\mathcal{D}$ be a distribution over a set of information resources $\mathcal{F}$, let $F$ be a random variable such that $F \sim \mathcal{D}$, let $t \subseteq \Omega$ be an arbitrary fixed $k$-sized target set with corresponding target function $\boldsymbol{t}$, and let $q(t, F)$ be the expected per-query probability of success for algorithm $\mathcal{A}$ on search problem $(\Omega, t, F)$. Then, for any $q_{\min} \in [0, 1]$,*

$$\Pr(q(t, F) \geq q_{\min}) \leq \frac{p + \mathrm{Bias}(\mathcal{D}, \boldsymbol{t})}{q_{\min}}$$

*where $p = \frac{k}{|\Omega|}$.*

Since the size of the target set $t$ is usually small relative to the size of the search space $\Omega$, $p$ is also typically small. Following the above results, we see that the probability that a search problem (with information resource drawn from $\mathcal{D}$) is favorable is bounded by a small value. This bound tightens as we increase our minimum threshold of success, $q_{\min}$. Notably, our bound relaxes with the introduction of bias.



(a) $\overline{P}_{f_1} = [0, 0.2, 0.8]^\top$, $\boldsymbol{t_1} = [0, 1, 1]^\top$, and $\theta_1 \approx 31°$. While all of the probability mass in $\overline{P}_{f_1}$ lies on the target set $\boldsymbol{t_1}$, the target divergence takes value greater than 0° because $\overline{P}_{f_1}$ is not uniform.

(b) $\overline{P}_{f_2} = [0, 1, 0]^\top$, $\boldsymbol{t_2} = [1, 0, 1]^\top$, and $\theta_2 = 90°$. Since none of the non-zero probability mass in $\overline{P}_{f_2}$ aligns with their corresponding target elements in the target set $\boldsymbol{t_2}$, the target divergence is maximized at 90°.

(c) $\overline{P}_{f_3} = [0.5, 0.5, 0]^\top$, $\boldsymbol{t_3} = [1, 1, 0]^\top$, and $\theta_3 = 0°$. Since $\overline{P}_{f_3}$ places all of its probability mass uniformly on the target set, the target divergence is minimized at 0°.

**Fig. 2.** These examples visualize the target divergence for various possible combinations of target functions and simplex vectors. (b) demonstrates minimum alignment, while (c) demonstrates maximum alignment.

**Corollary 1 (Probability of Success Under Bias-Free Search).** *When* $\text{Bias}(\mathcal{D}, \boldsymbol{t}) = 0$*,*

$$\Pr(q(t, F) \geq q_{\min}) \leq \frac{p}{q_{\min}}$$

Directly following Theorem 1, if the algorithm does not induce bias on $\boldsymbol{t}$ given a distribution over a set of information resources, the probability of successful search based on an information resource sampled from $\mathcal{D}$ cannot be any higher than that of uniform random sampling divided by the minimum performance that we specify. This bound matches that of the original Famine of Forte [5].

**Corollary 2 (Geometric Divergence).**

$$\Pr(q(t, F) \geq q_{\min}) \leq \frac{\sqrt{k}\cos(\theta)}{q_{\min}} = \frac{\|\boldsymbol{t}\|\cos(\theta)}{q_{\min}}$$

This result shows that greater geometric alignment between the target vector and expected distribution over the search space loosens the upper bound on the probability of successful search. Connecting this to our other results, the geometric alignment can be viewed as another interpretation of the bias the algorithm places on the target set.

**Theorem 2 (Conservation of Bias).** *Let* $\mathcal{D}$ *be a distribution over a set of information resources and let* $\tau_k = \{\boldsymbol{t} | \boldsymbol{t} \in \{0, 1\}^{|\Omega|}, ||\boldsymbol{t}|| = \sqrt{k}\}$ *be the set of all*

*$|\Omega|$-length k-hot vectors. Then for any fixed algorithm $\mathcal{A}$,*

$$\sum_{\boldsymbol{t} \in \tau_k} \text{Bias}(\mathcal{D}, \boldsymbol{t}) = 0$$

Since bias is a conserved quantity, an algorithm that is biased towards any particular target is equally biased against other targets, as is the case in Schaffer's conservation law for generalization performance [11]. This conservation property holds regardless of the algorithm or the distribution over information resources. Positive dependence between targets and information resources is the grounds for all successful machine learning [6], and this conservation result is another manifestation of this general property of learning.

**Theorem 3 (Famine of Favorable Information Resources).** *Let $\mathcal{B}$ be a finite set of information resources and let $t \subseteq \Omega$ be an arbitrary fixed k-size target set with corresponding target function $\boldsymbol{t}$. Define*

$$\mathcal{B}_{q_{\min}} = \{f \mid f \in \mathcal{B}, q(t, f) \geq q_{\min}\},$$

*where $q(t, f)$ is the expected per-query probability of success for algorithm $\mathcal{A}$ on search problem $(\Omega, t, f)$ and $q_{\min} \in [0, 1]$ represents the minimally acceptable per-query probability of success. Then,*

$$\frac{|\mathcal{B}_{q_{\min}}|}{|\mathcal{B}|} \leq \frac{p + \text{Bias}(\mathcal{B}, \boldsymbol{t})}{q_{\min}}$$

*where $p = \frac{k}{|\Omega|}$.*

This theorem shows us that unless our set of information resources is biased towards our target, only a small proportion of information resources will yield a high probability of search success. In most practical cases, $p$ is small enough that uniform random sampling is not considered a plausible strategy, since we typically have small targets embedded in very large search spaces. Thus the bound is typically very constraining. The set of information resources will be overwhelmingly unhelpful unless we restrict the given information resources to be positively biased towards the specified target.

**Corollary 3 (Proportion of Successful Problems Under Bias-Free Search).** *When* $\text{Bias}(\mathcal{B}, \boldsymbol{t}) = 0$*,*

$$\frac{|\mathcal{B}_{q_{\min}}|}{|\mathcal{B}|} \leq \frac{p}{q_{\min}}$$

Directly following Theorem 3, if the algorithm does not induce bias on $\boldsymbol{t}$ given a set of information resources, the proportion of successful search problems cannot be any higher than the single-query success probability of uniform random sampling divided by the minimum specified performance.

**Theorem 4 (Futility of Bias-Free Search).** *For any fixed algorithm $\mathcal{A}$, fixed target $t \subseteq \Omega$ with corresponding target function $\boldsymbol{t}$, and distribution over information resources $\mathcal{D}$, if* $\mathrm{Bias}(\mathcal{D}, \boldsymbol{t}) = 0$*, then*

$$\Pr(\omega \in t; \mathcal{A}) = p$$

*where* $\Pr(\omega \in t; \mathcal{A})$ *represents the per-query probability of successfully sampling an element of $t$ using $\mathcal{A}$, marginalized over information resources $F \sim \mathcal{D}$, and $p$ is the single-query probability of success under uniform random sampling.*

This result shows that without bias, an algorithm can perform no better than uniform random sampling. This is a generalization of Mitchell's idea of the futility of removing biases for binary classification [4] and Montañez's formal proof for the need for bias for multi-class classification [6]. This result shows that bias is necessary for any machine learning or search algorithm to have better than random chance performance, of those representable in our framework.

**Theorem 5 (Famine of Applicable Targets).** *Let $\mathcal{D}$ be a distribution over a finite set of information resources. Define*

$$\begin{aligned} \tau_k &= \{t \mid t \subseteq \Omega, |t| = k\} \\ \tau_{q_{\min}} &= \{t \mid t \in \tau_k, \mathrm{Bias}(\mathcal{D}, \boldsymbol{t}) \geq q_{\min}\} \end{aligned}$$

*where $\boldsymbol{t}$ is the target function corresponding to the target set $t$. Then,*

$$\frac{|\tau_{q_{\min}}|}{|\tau_k|} \leq \frac{p}{p + q_{\min}} \leq \frac{p}{q_{\min}}$$

*where $p = \frac{k}{|\Omega|}$.*

This theorem shows that the proportion of target sets for which an algorithm is highly biased is small, given that $p$ is small relative to $q_{\min}$. A high value of $\mathrm{Bias}(\mathcal{D}, \boldsymbol{t})$ implies that the algorithm, given $\mathcal{D}$, places a large amount of mass on $\boldsymbol{t}$ and a small amount of mass on other target functions. Consequently, an algorithm is acceptably biased toward fewer target sets as we increase the minimum threshold of bias.

**Theorem 6 (Famine of Favorable Biasing Distributions).** *Given a fixed target function $\boldsymbol{t}$, a finite set of information resources $\mathcal{B}$, and a set $\mathcal{P} = \{\mathcal{D} \mid \mathcal{D} \in \mathbb{R}^{|\mathcal{B}|}, \sum_{f \in \mathcal{B}} \mathcal{D}(f) = 1\}$ of all discrete $|\mathcal{B}|$-dimensional simplex vectors,*

$$\frac{\mu(\mathcal{G}_{\boldsymbol{t}, q_{\min}})}{\mu(\mathcal{P})} \leq \frac{p + \mathrm{Bias}(\mathcal{B}, \boldsymbol{t})}{q_{\min}}$$

*where $\mathcal{G}_{\boldsymbol{t}, q_{\min}} = \{\mathcal{D} \mid \mathcal{D} \in \mathcal{P}, \mathrm{Bias}(\mathcal{D}, \boldsymbol{t}) \geq q_{\min}\}$ and $\mu$ is Lebesgue measure.*

We see that the proportion of distributions over $\mathcal{B}$ for which an algorithm is acceptably biased towards a fixed target function $\boldsymbol{t}$ decreases as we increase the minimum acceptable level of bias, $q_{\min}$. Additionally, the greater the amount of bias induced by an algorithm given a set of information resources on a fixed target, the higher the probability of identifying a suitable distribution that achieves successful search. However, unless the set is already filled with favorable elements, finding a minimally favorable distribution over that set is difficult.

**Theorem 7 (Bias Over Distributions).** *Given a finite set of information resources $\mathcal{B}$, a fixed target function $\boldsymbol{t}$, and a set $\mathcal{P} = \{\mathcal{D} \mid \mathcal{D} \in \mathbb{R}^{|\mathcal{B}|}, \sum_{f \in \mathcal{B}} \mathcal{D}(f) = 1\}$ of discrete $|\mathcal{B}|$-dimensional simplex vectors,*

$$\int_{\mathcal{P}} \mathrm{Bias}(\mathcal{D}, \boldsymbol{t})\, \mathrm{d}\mathcal{D} = C \cdot \mathrm{Bias}(\mathcal{B}, \boldsymbol{t})$$

*where $C = \int_{\mathcal{P}} \mathrm{d}\mathcal{D}$ is the uniform measure of set $\mathcal{P}$. For an unbiased set $\mathcal{B}$,*

$$\int_{\mathcal{P}} \mathrm{Bias}(\mathcal{D}, \boldsymbol{t})\, \mathrm{d}\mathcal{D} = 0$$

This theorem states that the total bias on a fixed target function over all possible distributions is proportional to the bias induced by the algorithm given $\mathcal{B}$. When there is no bias over a set of information resources, the total bias over all distributions sums to 0. It follows that any distribution over $\mathcal{D}$ for which the algorithm places positive bias on $\boldsymbol{t}$ is offset by one or more for which the algorithm places negative bias on $\boldsymbol{t}$.

**Corollary 4 (Conservation of Bias Over Distributions).** *Let $\tau_k = \{\boldsymbol{t} | \boldsymbol{t} \in \{0,1\}^{|\Omega|}, ||\boldsymbol{t}|| = \sqrt{k}\}$ be the set of all $|\Omega|$-length k-hot vectors. Then,*

$$\sum_{\boldsymbol{t} \in \tau_k} \int_{\mathcal{P}} \mathrm{Bias}(\mathcal{D}, \boldsymbol{t})\, \mathrm{d}\mathcal{D} = 0$$

Here we see that the total bias over all distributions and all $k$-size target sets sums to zero, even if beginning with a set of information resources that is positively biased towards a particular target, as implied by the previous Theorem 7.

# 5 Examples

## 5.1 Genetic Algorithms

Genetic algorithms are optimization methods inspired by evolutionary processes [9]. We can represent genetic algorithms in our search framework as follows:

- $\mathcal{A}$ - a genetic algorithm, with standard variation (mutation, crossover, etc.) operators.
- $\Omega$ - space of possible configurations (genotypes).
- $T$ - set of all configurations which perform well on some task.
- $F$ - a fitness function which can evaluate a configuration's fitness.
- $(\Omega, T, F)$ - genetic algorithm task.

Given any genetic algorithm that is unbiased towards a particular small target when averaged over a set of fitness functions (as in No Free Lunch scenarios), the proportion of highly favorable fitness functions in that set must also be small, which we state as a corollary following directly from Corollary 3.

**Corollary 5 (Famine of Favorable Fitness Functions).** *For any fixed target $t \subseteq \Omega$ and fixed genetic algorithm unbiased relative to a finite set of fitness functions $\mathcal{B}$, the proportion of fitness functions in $\mathcal{B}$ with expected per-query probability of success at least $q_{min}$ is no greater than $|t|/(q_{min}|\Omega|)$.*

### 5.2 Binary Classification

We can cast binary classification as a search problem, as follows [5]:

- $\mathcal{A}$ - classification algorithm, such as a decision tree learner.
- $\Omega$ - space of possible binary labelings over an instance space.
- $t \subseteq \Omega$ - set of all hypotheses with less than 10% classification error.
- $F$ - set of training examples, where $F(\emptyset)$ is the full set of training data and $F(c)$ is the loss on training data for hypothesis $c$.
- $(\Omega, t, F)$ - binary classification learning task.

In our example, let $|\Omega| = 2^{100}$. Assume the size of our target set is $|t| = 2^{10}$, the set of training examples $F$ is drawn from a distribution $\mathcal{D}$, and that the minimum performance $q_{\min}$ we want to achieve is 0.5. Then, by Corollary 1, if our algorithm (relative to $\mathcal{D}$) does not place any bias on the target set,

$$\Pr\left(q(t, F) \geq \frac{1}{2}\right) \leq \frac{p}{q_{\min}} = \frac{\frac{2^{10}}{2^{100}}}{\frac{1}{2}} = 2^{-89}.$$

Thus, the probability that we will have selected a dataset that results in at least our desired level of performance is upper bounded by $2^{-89}$. Notice that if we raised the minimum threshold, then the probability would decrease—favorable datasets would become more unlikely.

To perform better than uniform random sampling, we would need to introduce bias into the algorithm. For example, predetermined information or assumptions about the target set could be used to determine which hypotheses are more plausible. The principle of Occam's razor [8] is often used, which is the assumption that the elements in the target set are likely the "simpler" elements, by some definition of simplicity. Relating this to our formal definition of bias, if we introduce correct assumptions into the algorithm, then the expected alignment of the target set and the induced probability distribution over the search space increases accordingly.

## 6 Conclusion

We build on the algorithmic search framework and extend Famine of Forte results to search problems with fixed targets and varying information resources.

Our notion of bias quantifies the extent to which an algorithm is predisposed to a particular fixed target. We show that bias towards any target necessarily implies bias against the other remaining targets, underscoring the fact that no universally applicable form of bias can exist. Furthermore, one cannot perform better than uniform random sampling without introducing a predisposition in the algorithm towards a desired target—unbiased algorithms are useless. Few information resources can be greatly favorable towards any fixed target, unless the algorithm is already predisposed to the target no matter the information resource given. Thus, in machine learning as elsewhere, biases are needed for better than chance performance. Biases must also be correct, since the effectiveness of any bias depends on how well it aligns with the given target actually being sought.

## References

1. Goldberg, D.: Genetic Algorithms in Search Optimization and Machine Learning. Addison-Wesley Longman Publishing Company, Boston (1999)
2. Gülçehre, Ç., Bengio, Y.: Knowledge matters: importance of prior information for optimization. J. Mach. Learn. Res. **17**(8), 1–32 (2016)
3. McDermott, J.: When and why metaheuristics researchers can ignore "no free lunch" theorems. Metaheuristics, March 2019. https://doi.org/10.1007/s42257-019-00002-6
4. Mitchell, T.D.: The need for biases in learning generalizations. CBM-TR-117. Rutgers University (1980)
5. Montañez, G.D.: The famine of forte: few search problems greatly favor your algorithm. In: 2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC), pp. 477–482. IEEE (2017)
6. Montañez, G.D.: Why machine learning works. Dissertation, pp. 52–59. Carnegie Mellon University (2017)
7. Montañez, G.D., Hayase, J., Lauw, J., Macias, D., Trikha, A., Vendemiatti, J.: The futility of bias-free learning and search. arXiv e-prints arXiv:1907.06010, July 2019
8. Rasmussen, C.E., Ghahramani, Z.: Occam's Razor. In: Proceedings of the 13th International Conference on Neural Information Processing Systems, NIPS 2000, pp. 276–282. MIT Press, Cambridge, MA, USA (2000)
9. Reeves, C., Rowe, J.E.: Genetic Algorithms: Principles and Perspectives: A Guide to GA Theory, vol. 20. Springer, Heidelberg (2002). https://doi.org/10.1007/b101880
10. Runarsson, T., Yao, X.: Search biases in constrained evolutionary optimization. IEEE Trans. Syst. Man Cybern. Part C Appl. Rev. **35**, 233–243 (2005). https://doi.org/10.1109/TSMCC.2004.841906
11. Schaffer, C.: A conservation law for generalization performance. In: Machine Learning Proceedings 1994, pp. 259–265. Elsevier (1994)
12. Ulyanov, D., Vedaldi, A., Lempitsky, V.: Deep image prior. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 9446–9454 (2018)
13. Wolpert, D.H., Macready, W.G.: No free lunch theorems for optimization. Trans. Evol. Comput. **1**(1), 67–82 (1997). https://doi.org/10.1109/4235.585893