




Decomposable Probability-of-Success Metrics in Algorithmic Search

Tyler Sam¹, Jake Williams¹, Abel Tadesse², Huey Sun³ and George Montañez¹

¹Harvey Mudd College, California, U.S.A.

²Claremont McKenna College, California, U.S.A.

³Pomona College, California, U.S.A.

Keywords: Decomposable Probability-of-Success Metric, Machine Learning as Search, Algorithmic Search Framework.

Abstract: Prior work in machine learning has used a specific success metric, the expected per-query probability of success, to prove impossibility results within the algorithmic search framework. However, this success metric prevents us from applying these results to specific subfields of machine learning, e.g. transfer learning. We define decomposable metrics as a category of success metrics for search problems which can be expressed as a linear operation on a probability distribution to solve this issue. Using an arbitrary decomposable metric to measure the success of a search, we demonstrate theorems which bound success in various ways, generalizing several existing results in the literature.

1 INTRODUCTION

Analyzing the success of a machine learning algorithm on specific problems is often very difficult given all the different variables in each problem. One solution is to reduce machine learning to search since many machine learning tasks, such as classification, regression, and clustering, can be reduced to search problems (Montañez, 2017b). Through this reduction, one can apply concepts from information theory to derive impossibility results about machine learning. For example, any specific machine learning algorithm can only do well on a small subset of all possible problems. To compare the success of different algorithms, or the expected probability of finding a desired element, Montañez defined a metric of success that averaged the probability of success over all iterations of an algorithm (Montañez, 2017b). While this metric has many applications, it is not appropriate for cases where the probability of success for a given iteration of an algorithm is required. An example of this is transfer learning, where the probability of success at the final step of the algorithm is more relevant than the average probability of success.


Building on this work, we define decomposability


as a property of probability-of-success metrics and show that the expected per-query probability of success (Montañez, 2017b) and more general probability of success metrics are decomposable. We then show that the results previously proven for the expected per-query probability of success hold for all decomposable probability-of-success metrics. Under this generalization, we can prove results related to the probability of success for specific iterations of a search rather than just uniformly averaged over the entire search, giving the results much broader applicability.


2 RELATED WORK


Several decades ago, Mitchell proposed that classification could be viewed as search, and reduced the problem of learning generalizations to a search problem within a hypothesis space (Mitchell, 1980; Mitchell, 1982). Montañez subsequently expanded this idea into a formal search framework (Montañez, 2017b).


Montañez showed that for a given algorithm with a fixed information resource, favorable target sets, or the target sets on which the algorithm would perform better than uniform random sampling, are rare. He did this by proving that the proportion of b -bit favorable problems has an exponentially decaying restrictive bound (Montañez, 2017a). He further showed that this scarcity of favorable problems exists even for

^a <https://orcid.org/0000-0001-7974-3226>

^b <https://orcid.org/0000-0001-9714-1851>

^c <https://orcid.org/0000-0002-3337-9454>

^d <https://orcid.org/0000-0002-0949-3169>

^e <https://orcid.org/0000-0002-1333-4611>

small k -sized target sets.

Montañez et al. later defined bias, the degree to which an algorithm is predisposed to a fixed target, with respect to the expected per-query probability of success metric, and proved that there were a limited number of favorable information resources for a given bias (Montañez et al., 2019). Using the search framework, they proved that an algorithm cannot be favorably biased towards many distinct targets simultaneously.

As machine learning grew in prominence, researchers began to probe what was possible within machine learning. Valiant considered learnability of a task as the ability to generate a program for performing the task without explicit programming of the task (Valiant, 1984). By restricting the tasks to a specific context, Valiant demonstrated a set of tasks which were provably learnable.

Schaffer provided an early foundation to the idea of bounding universal performance of an algorithm (Schaffer, 1994). Schaffer analyzed generalization performance, the ability of a learner to classify objects outside of its training set, in a classification task. Using a baseline of uniform sampling from the classifiers, he showed that, over the set of all learning situations, a learner's generalization performance sums to zero, which makes generalization performance a conserved quantity.

Wolpert and Macready demonstrated that the historical performance of a deterministic optimization algorithm provides no *a priori* justification whatsoever for its continued use over any other alternative going forward (Wolpert and Macready, 1997), implying that there is no utility in rationally choosing a thus-far better algorithm over choosing the opposite. Furthermore, just as there does not exist a single algorithm that performs better than random on all possible optimization problems, they proved that there also does not exist an optimization problem on which all algorithms perform better than average.

Continuing the application of prior knowledge to learning and optimization, Gülçehre and Bengio showed that the worse-than-chance performance of certain machine learning algorithms can be improved through learning with hints, namely, guidance using a curriculum (Gülçehre and Bengio, 2016). So, while Wolpert's results might make certain tasks seem futile and infeasible, Gülçehre's empirical results show that there exist some alternate means through which we can use prior knowledge to attain better results in both learning and optimization. Dembski and Marks measured the contributions of such prior knowledge using active information (Dembski and Marks II, 2009) and proved the difficulty of finding a good search al-

gorithm for a fixed problem (Dembski and Marks II, 2010), through their concept of a *search for a search (S4S)*. Eventually, their work expanded into a formal general theory of search, characterizing the information costs associated with success (Dembski et al., 2013), which served as an inspiration for later developments in machine learning (Montañez, 2017b).

Others have worked towards meaningful bounds on algorithmic success through different approaches. Sterkenburg approached this concept from the perspective of Putnam, who originally claimed that a universal learning machine is impossible through the use of a diagonalization argument (Sterkenburg, 2019). Sterkenburg follows up on this claim, attempting to find a universal inductive rule by exploring a measure which cannot be diagonalized. Even when attempting to evade Putnam's original diagonalization, Sterkenburg is able to apply a new diagonalization that reinforces Putnam's original claim of the impossibility of a universal learning machine.

There has also been work on proving learning bounds for specific problems. Kumagai and Kanamori analyzed the theoretical bounds of parameter transfer algorithms and self-taught learning (Kumagai and Kanamori, 2019). By looking at the local stability, or the degree to which a feature is affected by shifting parameters, they developed a definition for parameter transfer learnability, which describes the probability of effective transfer.

2.1 Distinctions from Prior Work

The expected per-query probability of success metric previously defined in the algorithmic search framework (Montañez, 2017b) tells us, for a given information resource, algorithm, and target set, how often (in expectation) our algorithm will successfully locate elements of the target set. While this metric is useful when making general claims about the performance of an algorithm or the favorability of an algorithm and information resource to the target set, it lacks the specificity to make claims about similar performance and favorability on a per-iteration basis. This trade-off calls for a more general metric that can be used to make both general and specific (per iteration) claims. For instance, in transfer learning tasks, the performance and favorability of the last pre-transfer iteration is more relevant than the overall expected per-query probability of success. The general probability of success, which we will define as a particular decomposable probability-of-success metric, is a tool through which we can make claims at specific and relevant steps.

3 BACKGROUND

In this section, we will present definitions for the main framework that we will use throughout this paper.

3.1 The Search Framework

Montañez describes a framework which formalizes search problems in order to analyze search and learning algorithms (Montañez, 2017a). To that end, he casts various machine learning algorithms, such as regression, classification, and clustering, into this search framework. This ML-as-search framework is valuable because it provides a structure to understand and reason about different machine learning problems within the same formalism. For example, we can understand regression as a search through a space of possible regression functions, and parameter estimation as a search through possible vectors for a black-box process (Montañez, 2017b). Therefore, we can apply results about search to any machine learning problem we can cast into the search framework. Furthermore, this framework lets us more easily analyze the necessary factors for success in a machine learning problem by viewing it as a search problem.

There are three components to a search problem. The first is the finite discrete search space, Ω , which is the set of elements to be examined. (Finiteness and discreteness follows from finite precision numerical representation, so the loss of generality is not great.) Next is the target set, T , which is a nonempty subset of the search space that we are trying to find. Finally, we have an external information resource, F , which provides an evaluation of elements of the search space. Typically, there is a tight relationship between the target set and the external information resource, as the resource is expected to lead to or describe the target set in some way, such as the target set being elements which meet a certain threshold under the external information resource.

Within the framework, we have an iterative algorithm which searches for elements of the target set, shown in Figure 1. The algorithm is a black-box that has access to a search history and produces a probability distribution over the search space. At each step, the algorithm samples over the search space using the probability distribution, evaluates that element using the information resource, adds the result to the search history, and determines the next probability distribution through its own internal rules and logic. The abstraction of finding the next probability distribution as a black-box algorithm allows the search framework to work with all types of search problems.

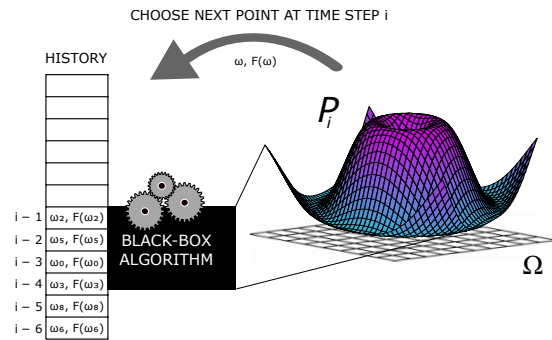


Figure 1: Black-box search algorithm. We iteratively populate the history with samples from a distribution that is determined by the black-box at each iteration, using the history (Montañez, 2017b).

3.2 Expected Per-query Probability of Success

In order to compare search algorithms, Montañez defined the expected per-query probability of success,

$$q(t, f) = \mathbb{E}_{\bar{P}, H} \left[\frac{1}{|\bar{P}|} \sum_{i=1}^{|\bar{P}|} P_i(w \in t) \mid f \right] = \bar{P}(X \in t \mid f) \quad (3.1)$$

where \bar{P} is the sequence of probability distributions generated by the black box, H is the search history, and t and f are the target set and information resource of the search problem, respectively (Montañez, 2017a). This metric of success is particularly useful because it can be shown that $q(t, f) = \mathbf{t}^\top \bar{\mathbf{P}}_f$, where $\bar{\mathbf{P}}_f$ is the average of the vector representation of the probability distribution from the search algorithm at each step, conditioned on an information resource f .

Measuring success using the expected per-query probability of success, Montañez demonstrated bounds on the success of any search algorithm (Montañez, 2017a). The Famine of Forte states that for a given algorithm, the proportion of target set-information resource pairs yielding a success level above a given threshold is inversely related to the threshold. Thus, the greater the threshold for success, the fewer problems you can be successful on, regardless of the algorithm. The expected per-query probability of success can also be used to prove a version of the No Free Lunch theorem, demonstrating that all algorithms perform the same averaged over all target sets and information resources, as is done in Theorem 1 of the current manuscript.

3.3 Bias

Using the search framework, Montañez defined a measure of bias between a distribution over information resources and a fixed target (Montañez et al.,

2019). For a distribution \mathcal{D} over a collection of possible information resources \mathcal{F} , with $F \sim \mathcal{D}$, and a fixed k -hot¹ target \mathbf{t} , the bias between the distribution and the target is defined as

$$\text{Bias}(\mathcal{D}, \mathbf{t}) = \mathbb{E}_{\mathcal{D}}[\mathbf{t}^\top \bar{\mathbf{P}}_F] - \frac{k}{|\Omega|} \quad (3.2)$$

$$= \mathbf{t}^\top \mathbb{E}_{\mathcal{D}}[\bar{\mathbf{P}}_F] - \frac{\|\mathbf{t}\|^2}{|\Omega|} \quad (3.3)$$

$$= \mathbf{t}^\top \int_{\mathcal{F}} \bar{\mathbf{P}}_f \mathcal{D}(f) df - \frac{\|\mathbf{t}\|^2}{|\Omega|}. \quad (3.4)$$

Recall from above that $\bar{\mathbf{P}}_f$ was the averaged probability distribution over Ω from a search.

The bias term measures the performance of an algorithm in expectation (over a given distribution of information resources) compared to uniform sampling. Mathematically, this is computed by taking the difference between the expected value of the average performance of an algorithm and the performance of uniform sampling. The distribution \mathcal{D} captures what information resources (e.g., datasets) one is likely to encounter.

For a non-mathematical example of the effect of bias, suppose we are searching for parking space within a parking lot. If we randomly choose parking spaces to check, we are searching without bias. However, if we consider the location of the parking spaces, we may find that parking spaces furthest from the entrance are usually free, and could find an open parking space with a higher probability. Here, the information resource telling us the distance of each parking space from the entrance and our belief that parking spaces further from the entrance tend to be open creates a distribution over possible parking spaces, favoring those that are further away for being checked first.

4 PRELIMINARIES

In this section, we introduce a new property of success metrics called decomposability, which allows us to generalize concepts of success and bias. We provide a number of preliminary lemmata, with full proofs given in the Appendix (available online (Sam et al., 2020)).

4.1 Decomposability

We now give a formal definition for a decomposable probability-of-success metric, which will be used throughout the rest of the paper.

¹ k -hot vectors are binary vectors of length $|\Omega|$ with exactly k ones.

Definition 4.1. A probability-of-success metric ϕ is **decomposable** if and only if there exists a $\mathbf{P}_{\phi, f}$ such that

$$\phi(t, f) = \mathbf{t}^\top \mathbf{P}_{\phi, f} = P_{\phi}(X \in t | f), \quad (4.1)$$

where $\mathbf{P}_{\phi, f}$ is not a function of t , being conditionally independent of t given f .

As we stated previously, what makes the expected per-query probability of success particularly useful is that it can be represented as a linear function of a probability distribution. This definition allows us to reference any probability-of-success metric having this property.

As a first example, we show that the expected per-query probability of success is a decomposable probability-of-success metric.

Lemma 4.2 (Decomposability of the Expected Per-Query Probability of Success). *The expected per-query probability of success is decomposable, namely,*

$$q(t, f) = \mathbf{t}^\top \bar{\mathbf{P}}_f. \quad (4.2)$$

Our goal is to show that the theorems proved for the expected per-query probability of success hold for all decomposable metrics. Showing that the expected per-query probability of success is decomposable suggests that these theorems may be generalizable to any metrics sharing that property.

4.1.1 The General Probability of Success

While the expected per-query probability of success averages the probability of success over each of the queries in a search history, we may care more about a specific query in the search history, e.g., the final query of a sequence. Thus, we can generalize the expected per-query probability of success by replacing the averaging with an arbitrary distribution α over the probability distributions in the search history. We define the General Probability of Success as

$$q_{\alpha}(t, f) = \mathbb{E}_{\bar{P}, H} \left[\sum_{i=1}^{|\bar{P}|} \alpha_i P_i(w \in t) \middle| f \right] = P_{\alpha}(X \in t | f) \quad (4.3)$$

where P_{α} is a valid probability distribution on the search space Ω and α_i is the weight allocated to the i th probability distribution in our sequence. This formula allows us to consider a wide variety of success metrics as being instances of the general probability of success metric. For example, the expected per-query probability of success is equivalent to setting $\mathbf{P}_{\alpha, f} = \bar{\mathbf{P}}_f$, with $\alpha_i = 1/|\bar{P}|$. Similarly, a metric of success which only cares about the final query can be represented by letting $\mathbf{P}_{\alpha, f} = \bar{\mathbf{P}}_{n, f}$ where n is the length of the sequence of queries, and $\bar{\mathbf{P}}_{n, f}$ is the average of the distributions from the n th iteration of our search.

It should be noted that α within the expectation will be random, being defined over the random number of steps within \bar{P} . Our operative definition of the α distribution, however, will allow us to generate the corresponding distribution for the needed number of steps, such as when we place all mass on the n th iteration of the search. With a slight abuse of notation, we thus let α signify both the process by which the distribution is generated as well as the particular distribution produced for a given number of steps.

As the general probability of success provides a layer of abstraction above the expected per-query probability of success, if we prove that results about the expected per-query probability of success also hold for the general probability of success, we gain a more powerful tool set. To do so, we must first demonstrate that the general probability of success is a decomposable probability-of-success metric.

Lemma 4.3 (Decomposability of the General Probability of Success Metric). *The general probability of success is decomposable, namely,*

$$q_\alpha(t, f) = \mathbf{t}^\top \mathbf{P}_{\alpha, f}. \quad (4.4)$$

These lemmata allow us to apply later theorems about decomposable metrics to these two useful metrics. Given a metric of interest, performing a similar proof of decomposability will allow for the application of the subsequent theorems.

Lemma 4.4 (Decomposability closed under expectation). *Given a set $S = \{\phi_i\}$ of decomposable probability-of-success metrics and a distribution \mathcal{D} over S , it holds that*

$$\phi'(t, f) = \mathbb{E}_{\mathcal{D}}[\phi(t, f)] \quad (4.5)$$

is also a decomposable probability-of-success metric.

Lemma 4.4 gives us an easy way to construct a new decomposable metric from a set of known decomposable metrics. Note that not every success metric is decomposable; we can create non-decomposable success metrics by taking non-convex combinations of decomposable probability-of-success metrics.

4.2 Generalization of Bias

Our definition of decomposability allows us to redefine bias in terms of any decomposable metric, $\phi(T, F)$. We replace $\bar{\mathbf{P}}_F$ with $\mathbf{P}_{\phi, F}$ and obtain

$$\text{Bias}_\phi(\mathcal{D}, \mathbf{t}) = \mathbb{E}_{\mathcal{D}}[\mathbf{t}^\top \mathbf{P}_{\phi, F}] - \frac{k}{|\Omega|}. \quad (4.6)$$

$$= \mathbf{t}^\top \mathbb{E}_{\mathcal{D}}[\mathbf{P}_{\phi, F}] - \frac{\|\mathbf{t}\|^2}{|\Omega|} \quad (4.7)$$

$$= \mathbf{t}^\top \int_{\mathcal{F}} \mathbf{P}_{\phi, f} \mathcal{D}(f) df - \frac{\|\mathbf{t}\|^2}{|\Omega|}. \quad (4.8)$$

Because $\phi(t, f)$ is decomposable, it is equal to $\mathbf{t}^\top \mathbf{P}_{\phi, f}$. This makes results about the bias particularly interesting, since they relate directly to any probability-of-success metric we create, so long as the metric is decomposable.

5 RESULTS

Montañez proved a number of results and bounds on the success of machine learning algorithms relative to the expected per-query probability of success, along with its corresponding definition of bias (Montañez, 2017b; Montañez et al., 2019). We now generalize these to apply to any decomposable probability-of-success metric, with full proofs given in the Appendix (available online (Sam et al., 2020)).

5.1 No Free Lunch for Search

First, we prove a version of the No Free Lunch Theorems for any decomposable probability-of-success metric within the search framework.

Theorem 1 (No Free Lunch for Search and Machine Learning). *For any pair of search/learning algorithms $\mathcal{A}_1, \mathcal{A}_2$ operating on discrete finite search space Ω , any closed under permutation set of target sets τ , any set of information resources \mathcal{B} , and decomposable probability-of-success metric ϕ ,*

$$\sum_{t \in \tau} \sum_{f \in \mathcal{B}} \phi_{\mathcal{A}_1}(t, f) = \sum_{t \in \tau} \sum_{f \in \mathcal{B}} \phi_{\mathcal{A}_2}(t, f). \quad (5.1)$$

This means that performance, in terms of our decomposable probability-of-success metric, is conserved in the sense that increased performance of one algorithm over another on some information resource-target pair comes at the cost of a loss in performance elsewhere.

5.2 The Fraction of Favorable Targets

Montañez proved that for a fixed information resource, a given algorithm \mathcal{A} will perform favorably relative to uniform random sampling on only a few target sets, under the expected per-query probability of success (Montañez, 2017b). We generalize this result with a decomposable probability-of-success metric and define a version of active information of expectations for decomposable metrics $I_{\phi(t, f)} := -\log_2 \frac{p}{\phi(t, f)}$. This transforms the ratio of success probabilities into bits where $p = |t|/|\Omega|$, the per-query probability of success for uniform random sampling with replacement. $I_{\phi(t, f)}$ denotes the advantage \mathcal{A} has over uniform random sampling with replacement, in bits.

Theorem 2 (The Fraction of Favorable Targets). *Let $\tau = \{t \mid t \subseteq \Omega\}$, $\tau_b = \{t \mid \emptyset \neq t \subseteq \Omega, I_{\phi(t,f)} \geq b\}$ for decomposable probability-of-success metric ϕ . Then for $b \geq 3$,*

$$\frac{|\tau_b|}{|\tau|} \leq 2^{-b}. \quad (5.2)$$

Thus, the scarcity of b -bit favorable targets still holds under for any decomposable probability-of-success metric.

5.3 The Famine of Favorable Targets

Following up on the previous result, we can show a similar bound in terms of the success of a given algorithm, for targets of a fixed size.

Theorem 3 (The Famine of Favorable Targets). *For fixed $k \in \mathbb{N}$, fixed information resource f , and decomposable probability-of-success metric ϕ , define*

$$\tau = \{T \mid T \subseteq \Omega, |T| = k\}, \text{ and}$$

$$\tau_{q_{\min}} = \{T \mid T \subseteq \Omega, |T| = k, \phi(T, f) \geq q_{\min}\}.$$

Then,

$$\frac{|\tau_{q_{\min}}|}{|\tau|} \leq \frac{p}{q_{\min}} \quad (5.3)$$

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

Here, we compare success not against uniform sampling but against a fixed constant q_{\min} . This theorem thus upper bounds the number of targets for which the probability of success of the search is greater than q_{\min} .

5.4 Famine of Forte

We generalize the Famine of Forte (Montañez, 2017b), showing a bound that holds in the k -sparse case using any decomposable probability-of-success metric.

Theorem 4 (The Famine of Forte). *Define*

$$\tau_k = \{T \mid T \subseteq \Omega, |T| = k \in \mathbb{N}\}$$

and let \mathcal{B}_m denote any set of binary strings, such that the strings are of length m or less. Let

$$R = \{(T, F) \mid T \in \tau_k, F \in \mathcal{B}_m\}, \text{ and}$$

$$R_{q_{\min}} = \{(T, F) \mid T \in \tau_k, F \in \mathcal{B}_m, \phi(T, F) \geq q_{\min}\},$$

where $\phi(T, F)$ is the decomposable probability-of-success metric for algorithm \mathcal{A} on problem (Ω, T, F) . Then for any $m \in \mathbb{N}$,

$$\frac{|R_{q_{\min}}|}{|R|} \leq \frac{p}{q_{\min}}. \quad (5.4)$$

This demonstrates that for any decomposable metric there is an upper bound on the proportion of problems an algorithm is successful on. Here, we measure success as being above a certain threshold with respect to a decomposable metric, and the upper bound is inversely related to this threshold.

5.5 Learning Under Dependence

While the previous theorems highlight cases where an algorithm is unlikely to succeed, we now consider the conditions that make an algorithm likely to succeed. To begin, we consider how the target and information resource can influence an algorithm's success by generalizing the Learning Under Dependence theorem (Montañez, 2017a).

Theorem 5 (Learning Under Dependence). *Define $\tau_k = \{T \mid T \subseteq \Omega, |T| = k \in \mathbb{N}\}$ and let \mathcal{B}_m denote any set of binary strings (information resources), such that the strings are of length m or less. Define q as the expected decomposable probability of success under the joint distribution on $T \in \tau_k$ and $F \in \mathcal{B}_m$ for any fixed algorithm \mathcal{A} , such that $q := \mathbb{E}_{T,F}[\phi(T, F)]$, namely,*

$$q = \mathbb{E}_{T,F} [P_{\phi}(\omega \in T|F)] = \Pr(\omega \in T; \mathcal{A}).$$

Then,

$$q \leq \frac{I(T; F) + D(P_T \| \mathcal{U}_T) + 1}{I_{\Omega}} \quad (5.5)$$

where $I_{\Omega} = -\log k/|\Omega|$, $D(P_T \| \mathcal{U}_T)$ is the Kullback-Liebler divergence between the marginal distribution on T and the uniform distribution on T , and $I(T; F)$ is the mutual information. Alternatively, we can write

$$\Pr(\omega \in T; \mathcal{A}) \leq \frac{H(\mathcal{U}_T) - H(T|F) + 1}{I_{\Omega}} \quad (5.6)$$

where $H(\mathcal{U}_T) = \log \binom{|\Omega|}{k}$.

The value of q defined here represents the expected single-query probability of success of an algorithm relative to a randomly selected target and information resource, distributed according to some joint distribution. The probability of success for a single query (marginalized over information resources) is equivalent to the expectation of the conditional probability of success, conditioned on the random information resource. Upper-bounding this value states that regardless of the choice of decomposable probability-of-success metric, the probability of success depends on the amount of information regarding the target contained within the information resource, as measured by the mutual information.

5.6 Famine of Favorable Information Resources

We now demonstrate the effect of the general bias term defined earlier on the probability of a success of an algorithm. We begin with a generalization of the Famine of Favorable Information Resources (Montañez et al., 2019).

Theorem 6 (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 \mathbf{t} . Define*

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

where $\phi(t, f)$ is an arbitrary decomposable probability-of-success metric for algorithm \mathcal{A} on search problem (Ω, t, f) , and $q_{\min} \in (0, 1]$ represents the minimally acceptable probability of success. Then,

$$\frac{|\mathcal{B}_{q_{\min}}|}{|\mathcal{B}|} \leq \frac{p + \text{Bias}_{\phi}(\mathcal{B}, \mathbf{t})}{q_{\min}} \quad (5.7)$$

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

This result demonstrates the mathematical effect of bias, of which we have previously provided one hypothetical example (car parking). In particular, we can show that the bias of our expected information resources towards the target will upper bound the probability of a given information resource leading to a successful search.

5.7 Futility of Bias-Free Search

We can also use our definition of bias to generalize the Futility of Bias-Free Search (Montañez, 2017b), which demonstrates the inability of an algorithm to perform better than uniform random sampling without bias, defined with respect to the expected per-query probability of success. Our generalization proves that the theorem holds for bias defined with respect to any decomposable probability-of-success metric.

Theorem 7 (Futility of Bias-Free Search). *For any fixed algorithm \mathcal{A} , fixed target $t \subseteq \Omega$ with corresponding target function \mathbf{t} , and distribution over information resources \mathcal{D} , if $\text{Bias}_{\phi}(\mathcal{D}, \mathbf{t}) = 0$, then*

$$\Pr(\omega \in t; \mathcal{A}) = p \quad (5.8)$$

where $\Pr(\omega \in t; \mathcal{A})$ represents the expected decomposable 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 demonstrates that, regardless of how we measure the success of an algorithm with respect to a decomposable metric, it cannot perform better than uniform random sampling without bias.

5.8 Famine of Favorable Biasing Distributions

Montañez proved that the percentage of minimally favorable distributions (biased over some threshold towards some specific target) is inversely proportional to the threshold value and directly proportional to the bias between the information resource and target function (Montañez, 2017b). We will show that this scarcity of favorable biasing distributions holds, in general, for bias under any decomposable probability-of-success metric.

Theorem 8 (Famine of Favorable Biasing Distributions). *Given a fixed target function \mathbf{t} , a finite set of information resources \mathcal{B} , a distribution over information resources \mathcal{D} , 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,*

$$\frac{\mu(\mathcal{G}_{\mathbf{t}, q_{\min}})}{\mu(\mathcal{P})} \leq \frac{p + \text{Bias}_{\phi}(\mathcal{B}, \mathbf{t})}{q_{\min}} \quad (5.9)$$

where $\mathcal{G}_{\mathbf{t}, q_{\min}} = \{\mathcal{D} \mid \mathcal{D} \in \mathcal{P}, \text{Bias}_{\phi}(\mathcal{D}, \mathbf{t}) \geq q_{\min}\}$, $p = \frac{k}{|\Omega|}$, and μ is Lebesgue measure.

This result shows that the more bias there is between our set of information resources \mathcal{B} and the target function \mathbf{t} , the easier it is to find a minimally favorable distribution, and the higher the threshold for what qualifies as a minimally favorable distribution, the harder our search becomes. Thus, unless we want to suppose that we begin with a set of information resources already favorable towards our fixed target, finding a highly favorable distribution is difficult.

6 CONCLUSION

Casting machine learning problems as search provides a common formalism within which to prove bounds and impossibility results for a wide variety of learning algorithms and tasks. In this paper, we introduce a property of probability-of-success metrics called decomposability, and show that the expected per-query probability of success and general probability of success are decomposable. To demonstrate the value of this property, we prove that a number of existing algorithmic search framework results continue to hold for all decomposable probability-of-success metrics. These results provide a number

of useful insights: we show that algorithmic performance is conserved with respect to all decomposable probability-of-success metrics, favorable targets are scarce no matter your decomposable probability-of-success metric, and that without the generalized bias defined here, an algorithm will not perform better than uniform random sampling.

The goal of this work is to offer additional machinery within the search framework, allowing for more general application. To that end, we can develop decomposable probability-of-success metrics for problems concerned with the state of an algorithm at specific steps, and leverage existing results as a foundation for additional insight into those problems. One application is transfer learning. To do so, we use a decomposable probability-of-success metric that utilizes only the state of the algorithm at the last step to represent the information learned from a source problem.

ACKNOWLEDGEMENTS

This work was supported by the Walter Bradley Center for Natural and Artificial Intelligence. We thank Dr. Robert J. Marks II (Baylor University) for providing support and feedback. We also thank Harvey Mudd College's Department of Computer Science for their continued resources and support.

REFERENCES

- Dembski, W. A., Ewert, W., and Marks II, R. J. (2013). A general theory of information cost incurred by successful search. In *Biological Information: New Perspectives*, pages 26–63. World Scientific.
- Dembski, W. A. and Marks II, R. J. (2009). Conservation of information in search: measuring the cost of success. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 39(5):1051–1061.
- Dembski, W. A. and Marks II, R. J. (2010). The search for a search: Measuring the information cost of higher level search. *JACIII*, 14(5):475–486.
- Gülgehre, Ç. and Bengio, Y. (2016). Knowledge matters: Importance of prior information for optimization. *The Journal of Machine Learning Research*, 17(1):226–257.
- Kumagai, W. and Kanamori, T. (2019). Risk bound of transfer learning using parametric feature mapping and its application to sparse coding. *Machine Learning*, 108:1975–2008.
- Mitchell, T. M. (1980). The need for biases in learning generalizations. Technical report, Computer Science Department, Rutgers University, New Brunswick, MA.
- Mitchell, T. M. (1982). Generalization as Search. *Artificial intelligence*, 18(2):203–226.
- Montañez, G. D. (2017a). The Famine of Forte: Few Search Problems Greatly Favor Your Algorithm. In *Systems, Man, and Cybernetics (SMC), 2017 IEEE International Conference on*, pages 477–482. IEEE.
- Montañez, G. D. (2017b). *Why Machine Learning Works*. PhD thesis, Carnegie Mellon University.
- Montañez, G. D., Hayase, J., Lauw, J., Macias, D., Trikha, A., and Vendemiatti, J. (2019). The Futility of Bias-Free Learning and Search. In *32nd Australasian Joint Conference on Artificial Intelligence*, pages 277–288. Springer.
- Sam, T., Williams, J., Tadesse, A., Sun, H., and Montanez, G. (2020). Decomposable probability-of-success metrics in algorithmic search. *CoRR*, abs/2001.00742.
- Schaffer, C. (1994). A Conservation Law for Generalization Performance. *Machine Learning Proceedings 1994*, 1:259–265.
- Sterkenburg, T. F. (2019). Putnam's Diagonal Argument and the Impossibility of a Universal Learning Machine. *Erkenntnis*, 84(3):633–656.
- Valiant, L. (1984). A Theory of the Learnable. *Communications of the ACM*, 27:1134–1142.
- Wolpert, D. H. and Macready, W. G. (1997). No free lunch theorems for optimization. *IEEE Trans. Evolutionary Computation*, 1:67–82.