The Algorithmic Search Framework

Hana Ahmed[†]
AMISTAD Lab
Harvey Mudd College
Claremont, CA, USA
hahmed@hmc.edu

Amani R. Maina-Kilaas[†]
AMISTAD Lab
Harvey Mudd College
Claremont, CA, USA
amainakilaas@hmc.edu

Shifa Somji
AMISTAD Lab
Harvey Mudd College
Claremont, CA, USA
ssomji@hmc.edu

Sarah Embry
AMISTAD Lab
Harvey Mudd College
Claremont, CA, USA
sembry@hmc.edu

Isabel Duan
AMISTAD Lab
Harvey Mudd College
Claremont, CA, USA
iduan@hmc.edu

Cynthia Hom
AMISTAD Lab
Harvey Mudd College
Claremont, CA, USA
chom@hmc.edu

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

[†]denotes equal authorship.

***Abstract*—Artificial intelligence and machine learning are generally viewed as distinct research areas with various conceptual divisions even within themselves. Consequently, much of the existing theory is focused on specific domains or tasks. Although often useful, this specificity limits its ability to apply to other areas—requiring researchers to prove distinct results for each task and approach. The algorithmic search framework is a theoretical framework that unifies artificial intelligence and machine learning under the single research discipline of *artificial learning*, enabling us to prove results that simultaneously apply to a wide range of problems such as reinforcement learning, particle swarm optimization, genetic algorithms, and constraint satisfaction problems. This unification can reveal universal principles and constraints on the learning process, and has already yielded many theoretical bounds regarding algorithm performance.**

***Index Terms*—artificial intelligence, machine learning, search, statistical learning theory**

I. Introduction

Imagine you are an adventurous scavenger, searching for treasure. You are currently exploring a region of small islands, several of which you know to contain great riches hidden by various legendary pirates. In your hand, you hold a map to the buried treasure of your seafaring great-grandfather. However, while this map will lead you directly to his treasure, it is useless in searching for treasure belonging to any other pirates. To find these other pirates' treasure, you consider the possibility of investigating islands at random. This process will lead you to discover the treasures of many different pirates, but also cause you to waste time on many empty islands. While contemplating this issue, you remember something your father once said: "Most of the pirates in this area preferred to hide their treasure on islands with lots of trees." If you follow this option, you may still waste time, but your search is more likely to succeed in finding the treasures of multiple pirates.

In this fictitious scenario, you considered the outcomes of following several search algorithms. The map is highly-biased towards a particular target, guaranteeing your great-grandfather's treasure but failing miserably at locating the treasure of any other pirate. A random search is not biased at all, and is equally as good or bad at discovering the target of any pirate's treasure. Lastly, following your father's advice is a decent strategy for finding the treasure of some pirates, but can direct you away from pirates that use other criteria for their island choices. *What can we infer from analyzing these search algorithms?* To the extent that an algorithm is biased towards some targets, it is biased against other targets, in equal and opposite measure. This intuitive concept is a general principle of algorithms. We call this the Conservation of Bias (Theorem 2), and it implies that no single algorithm will be good at locating every possible target.

This is but one theorem that can be proven within the algorithmic search framework (ASF). The framework has been used to show that fixed algorithms can only perform well on a small proportion of problems [1], that the proportion of favorable algorithm strategies is strictly limited for a fixed problem [1], and that all successful learning is a product of information-theoretic dependence [2].

This theoretical framework unifies artificial intelligence (AI) and machine learning (ML) under the single discipline of *artificial learning*, enabling us to prove results that simultaneously apply to a wide range of problems such as reinforcement learning, particle swarm optimization, genetic algorithms, and constraint satisfaction problems. This is desirable, as there are many new algorithms being developed within the realms of AI and ML. Researchers have been able to separately prove results for each of these algorithms, but this requires new—and perhaps redundant—proofs for each one. This task becomes challenging as new algorithms are developed frequently. However, in viewing learning problems as a type of algorithmic search, we can show that there are results which hold for all AI and ML algorithms. Results proven within the ASF apply to anything that can be reduced to a search problem. Note that this requirement is not particularly limiting, as for example, any problem that can be reduced into Vapnik's learning framework can also be reduced to the ASF

(see Appendix).

After reduction to the framework, the implications of the theorems can be interpreted within the context of the original learning problem. This process has been used to prove several novel theoretical results for transfer learning [3] and even discover constraints on the minimum structural complexity for DNA binding proteins based on the information-theoretic specificity of the tasks they accomplish [4]. The ASF also warrants additional merit due to how it provides a more intuitive way to think about complicated algorithms. Physical search is a common human experience and can help to introduce artificial learning in educational settings.

The rest of this paper introduces the framework, demonstrates how it can be used, and presents prior results that prove theoretical bounds regarding algorithm performance on learning problems.

II. Using the Framework

In order to use results from the ASF, one needs to reduce the learning problem to an algorithmic search problem. A learning algorithm is represented within the framework as a black-box sampling algorithm, as illustrated in Fig. 1. The algorithm uses information it has gained during the search to produce a probability distribution on the search space and iteratively sample from it ("non-iterative" learning algorithms can be viewed as a single iteration with exactly one sample). After each sample, the algorithm uses the additional information to update its probability distribution and repeat the process until reaching its target. We formalize the components of this process as follows:

A. Components of an Algorithmic Search Problem

The *search space* Ω is a finite, discrete set consisting of elements to be examined during the search. However, note that these requirements are neither unreasonable nor unduly limiting. The assumption that Ω is finite is justified by the restrictions of physical computer hardware operating within a finite amount of time. The requirement that Ω is discrete is also

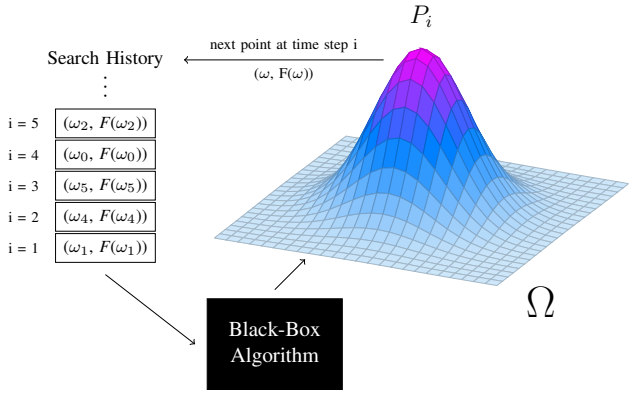


Fig. 1: A graphical representation of the black-box search process at iteration i. After the algorithm induces probability distribution P_i on search space Ω, it samples ω_j and learns new information $F(\omega_j)$. Figure reproduced from [5].

justified by consequence of physical computer hardware—computers only have finite precision representations of numbers and so our search spaces are effectively limited to be discrete, regardless of whether the "ideal" space is continuous. The ASF takes these physical limitations seriously, and works to account for them rather than ignore them.

Next, we let the *target set* $T \subseteq \Omega$ be a nonempty subset of the search space containing the elements we wish to find in the search. The set T can also be represented using a binary vector the size of Ω. In this vector, each indexed position evaluates to 1 if the corresponding element is in T and 0 otherwise. We call this one-to-one mapped binary vector of size $|\Omega|$ the *target function*, typically denoted $\mathbf{t}$.

Often, an *external information resource* F is used to evaluate the elements from the search space Ω. We choose to abstract F as a finite length binary string, which can represent anything from an objective function to a set of training data. Each specific problem domain defines the structure of this binary string and contains some form of interface that algorithms can "call" for information about elements in the search space. Calling this interface on the null element, which represents the lack of an element, would provide information about initialization of the search. The structure of this interface and the types of information it provides is intentionally left open, as it will vary depending on the problem domain.

B. Reducing to the Framework

Reducing a problem into the ASF consists of identifying and defining Ω, T, and F such that they fully encapsulate the problem at hand. We now provide a reduction for the machine learning problem of regression to demonstrate how this is done.

In simple cases, given a set of real-valued inputs, a learning algorithm uses *regression* to learn a function $g : \mathbb{R} \to \mathbb{R}$ from the inputs to real-valued outputs. The functions can also be more general, possibly taking multiple inputs and producing vector-valued outputs. Let D be a set of training data $D = \{(x_1, y_1), \ldots, (x_n, y_n)\}$ and V be a set of test data $V = \{(x_1, y_1), \ldots, (x_n, y_n)\}$, where $\mathcal{X}$ and $\mathcal{Y}$ are finite sets such that $x_i \in \mathcal{X}$ and $y_i \in \mathcal{Y}$. A learning algorithm $\mathcal{A}$ chooses a regression function $\hat{g}$ from a large, finite space $\mathcal{G}$ of possible regression functions according to loss function $\mathcal{L}$ and training data D. This means that $\hat{g} = \mathcal{A}(\mathcal{L}, D)$ is the function that produces the output values at data points in D and minimizes the loss function $\mathcal{L}$. We next let Ξ be an error functional operating on $\mathcal{L}$, regression function g, and test data V. Then the target set $T \subseteq \mathcal{G}$ is the set of all regression functions with acceptable performance, $T = \{g : g \in \Omega, \Xi(\mathcal{L}, g, V) \leq \epsilon\}$, where ϵ is some scalar threshold.

With this definition of a regression problem, we can reduce regression to a search problem within the ASF:

- $\Omega = \mathcal{G}$
- $T = \{g : g \in \Omega, \Xi(\mathcal{L}, g, V) \leq \epsilon\}$
- $F = \{D, \mathcal{L}\}$
- $F(\emptyset) = D$

- $F(\omega_i) = \mathcal{L}(g_i)$ where $g_i \in \mathcal{G}$ is the ith query in the search process

To summarize, we search through the set of all possible regression functions, looking for a function that has an error less than a designated threshold, using the training data and loss function as a guide. We initialize our search using information from the training data, and refine our search after each query by testing it with the loss function. Once we have appropriately defined the learning problem, we can use existing theorems to infer limits in terms of the original problem—this is the ASF's strength.

The ASF also interacts well with other theoretical frameworks. Vapnik's learning framework, a generalization that can be applied to machine learning problems such as density estimation and classification, can also be reduced into the ASF. This means that our results hold for any problem that can be represented within Vapnik's framework. A reduction of Vapnik's learning framework, as well as reductions to more artificial intelligence and machine learning problems, can be found in the Appendix.

III. Definitions

Before presenting the results obtained using the ASF, we will first introduce some of the necessary definitions.

Various metrics, such as loss functions and accuracy, are available to evaluate the performance of learning algorithms [6]. Within the framework we measure learning algorithms by their probability of success in finding a search target. A *probability-of-success metric* takes in a target set T and an information resource F and returns the likelihood of finding an element of T on a specific query. Note that using a probability-of-success metric does not mean discarding other metrics, since other metrics can often be integrated into the target set of the search problem.

One example of a probability-of-success metric is the *general probability of success*, which is a weighted average of the probability of success over each of the queries in the search's history.

Definition 1 (General Probability of Success)**.**

$$q_\alpha(T, F) = \mathbb{E}_{\tilde{P},H}\left[\sum_{i=1}^{|\tilde{P}|} \alpha_i P_i(w \in T)\Big|F\right] = P_\alpha(X \in T|F)$$

where $\tilde{P}$ is a sequence of probability distributions generated by the black box, P_i is the i^{th} probability distribution, α_i is the weight allocated to P_i, P_α is a valid probability distribution on the search space Ω, H is the search history, and T and F are the target set and information resource of the search problem.

Commonly, we use the *expected per-query probability of success*, which is a specific instance of Definition 1 that can be obtained by using a uniform average with $\alpha_i = \frac{1}{|\tilde{P}|}$.

Definition 2 (Expected Per-Query Probability of Success)**.**

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

where $\tilde{P}$ is a sequence of probability distributions generated by the black box, P_i is the i^{th} probability distribution, $\overline{P}$ is an appropriately averaged distribution, H is the search history, and T and F are the target set and information resource of the search problem.

The general probability of success and the expected per-query probability of success are both types of *decomposable* probability of success metrics. Any probability of success metric will hold for our results as long as it is *decomposable*.

Definition 3 (Decomposability)**.** *A probability-of-success metric ϕ is decomposable if and only if there exists a $\mathbf{P}_{\phi,F}$ such that*

$$\phi(T, F) = \mathbf{t}^\top \mathbf{P}_{\phi,F} = P_\phi(X \in T|F),$$

where $\mathbf{P}_{\phi,F}$ is the vector representation of the probability distribution (conditioned on F, and conditionally independent of T), induced on Ω during the course of the search.

Decomposable probability-of-success metrics are particularly useful because we can represent them as linear functions of a probability distribution.

Note that the use of decomposable probability-of-success metrics does not necessarily mean discarding the other metrics you may be familiar with—rather, other metrics can often be integrated into the search problem's target set in the form of

$$T_c = \{\omega \in \Omega \mid \omega \text{ meets criteria } c\}.$$

For example, one can search for regression functions such that the resulting mean squared error is less than a given ϵ, or a classification model such that accuracy is above a certain percentage on a specific data set. With a defined target set such as T_c, probability-of-success metrics can inherently represent other desired metrics.

Having defined metrics for the probability of success, we now explore what is required for an algorithm to be successful. Mitchell presents the concept of *inductive bias*, which he argues is required for a learning algorithm to generalize beyond training data [7].

Definition 4 (Inductive Bias (Mitchell))**.** *In the case of a machine learning algorithm, Inductive Bias is any basis for choosing one generalization over another, other than strict consistency with the observed training instances.*

It is important to note that this is distinct from prejudicial bias, which is problematic and should generally be removed from algorithms. Inductive bias, on the other hand, is necessary. In the ASF, we use a mathematical definition of *bias* inspired by Definition 4, which represents the algorithm's divergence from a random search, based on its implicit assumptions.

Definition 5 (Bias)**.** *Bias is the difference between the expected performance of a search algorithm, over a fixed target T (with corresponding target function $\mathbf{t}$) and distribution over information resources $\mathcal{D}$, and the baseline probability of success of a uniform random sample.*

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

The inductive bias of an algorithm is captured by its *inductive orientation vector*, which represents the learning algorithm's probability distribution over its search space.

Definition 6 (Inductive Orientation)**.** *Let $\mathbf{P}_{\phi,F}$ be the vector representation of the weighted-averaged probability distribution (conditioned on an information resource F) induced by a search algorithm on the search space Ω, namely,*

$$\mathbf{P}_{\phi,F} := \mathbb{E}_{\tilde{P},H}\left[\sum_{i=1}^{|\tilde{P}|} \alpha_i P_i \Big| F\right].$$

Highly biased algorithms allow us to have more successful searches for a particular problem. However, algorithms with too much bias have difficulty changing their behavior in response to new information. What highly biased algorithms lack is expressivity.

Definition 7 (Entropic Expressivity)**.** *Given a distribution $\mathcal{D}$ over information resources, we define the entropic expressivity of a search algorithm as the information-theoretic entropy of the averaged strategy distributions over $\mathcal{D}$, namely,*

$$H(\mathbf{P}_{\phi,\mathcal{D}}) = H(\mathbb{E}_{\mathcal{D}}\left[\mathbf{P}_{\phi,F}\right]) = H(\mathcal{U}) - D_{KL}(\mathbf{P}_{\phi,\mathcal{D}} \,\|\, \mathcal{U})$$

where $F \sim \mathcal{D}$ and the quantity $D_{KL}(\mathbf{P}_{\phi,\mathcal{D}} \,\|\, \mathcal{U})$ is the Kullback-Leibler divergence between distribution $\mathbf{P}_{\phi,\mathcal{D}}$ and the uniform distribution $\mathcal{U}$, both being distributions over search space Ω.

As we alter the properties of various search problems, we show that there are limitations to bias and expressivity, and as a result, there are several implications to how well an algorithm can perform over various problems. We present results about these limitations by examining the implications of varying the target set, varying the information resource, varying both the target set and information resource, and lastly, fixing both the target set and information resource.

IV. Results

A. Varying the Target Set

We start by considering the implications of varying the target set, which is the set of elements we wish our search algorithm to find. For example, in an object detection algorithm, this would be akin to varying the set of objects we'd like to identify.

First we can investigate what happens when we vary the ratio between the size of the target set and the size of the search space.

Theorem 1 (Bias Upper Bound)**.** *Let $\tau_k = \{\mathbf{t} \mid \mathbf{t} \in \{0,1\}^{|\Omega|}, \|\mathbf{t}\| = \sqrt{k}\}$ be the set of all $|\Omega|$-length k-hot vectors and let $\mathcal{B}$ be a finite set of information resources. Then,*

$$\sup_{\mathbf{t}\in\tau_k} \text{Bias}(\mathcal{B}, \mathbf{t}) \leq \left(\frac{p-1}{p}\right) \inf_{\mathbf{t}\in\tau_k} \text{Bias}(\mathcal{B}, \mathbf{t}) \quad \textit{where} \quad p = \frac{k}{|\Omega|}.$$

We see here that the ratio p between the size of the target set and the size of the search space determines the relationship between the upper and lower bounds on the values that bias can take over all possible target sets of size k. Specifically, as the size of the target set increases relative to the size of the search space Ω, the algorithm is more likely to perform better on a greater amount of target sets. Thus, the upper bound on the bias decreases as the ratio between the size of the target set and the size of the search space increases. Similarly, the lower bound on the bias increases as the ratio between the size of the target set and the size of the search space increases. Together, these imply that the range of values that bias can take gets smaller as the size of the target set approaches the size of the search space.

We can also examine what happens over target sets of the same size. In particular, we show that an algorithm that is biased towards any particular target is biased against other targets in equal and opposite measure.

Theorem 2 (Conservation of Bias)**.** *Let $\mathcal{D}$ be a distribution over a set of information resources and let $\tau_k = \{\mathbf{t} | \mathbf{t} \in \{0,1\}^{|\Omega|}, \|\mathbf{t}\| = \sqrt{k}\}$ be the set of all target functions $\mathbf{t}$ that are $|\Omega|$-length k-hot vectors. Then for any fixed algorithm $\mathcal{A}$,*

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

In other words, for any given distribution over information functions, the sum of the bias values over all possible target sets of a given size k within a search space is zero. This means bias is conserved over the set of all target sets, so no single algorithm will be good at outputting every target set.

Finally, we find that, when evaluated over all permutations of target sets and information resources, any two algorithms will have the same overall performance.

Theorem 3 (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 set of target sets τ closed under permutation, any set of information resources $\mathcal{B}$, and decomposable probability-of-success metric ϕ,*

$$\sum_{T\in\tau}\sum_{F\in\mathcal{B}} \phi_{\mathcal{A}_1}(T, F) = \sum_{T\in\tau}\sum_{F\in\mathcal{B}} \phi_{\mathcal{A}_2}(T, F). \tag{1}$$

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

Overall, by varying the target set, we see that we cannot have one algorithm that is good at finding every target set,

nor can we have an algorithm that performs better on average over all permutations of target sets and information resources than another algorithm.

B. Varying the Information Resource

Instead of varying the target set, we now consider the implications of varying the information resource. In practice, this would be equivalent to choosing a different set of training data to train a learning model, such as a deep neural network.

This next theorem states that for any decomposable probability-of-success metric, the bound on the algorithm's probability of success improves monotonically with the amount of information about the target set in the information resource.

Theorem 4 (Learning Under Dependence)**.** *Define*

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

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

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

Then,

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

where $I_\Omega = -\log k/|\Omega|$, $D(P_T \| \mathcal{U}_T)$ is the Kullback-Leibler 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

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

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

Upper-bounding the value of the probability of success shows 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. This means that favorable information resources are important, and so we ask the question: *are favorable information resources unlikely?* Unfortunately, yes.

We generalize the Improbability of Favorable Information Resources [5], restating the bound for any decomposable probability-of-success metric.

Theorem 5 (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 $\mathbf{t}$, and let $\phi(T,F)$ be a decomposable probability of success for algorithm $\mathcal{A}$ on search problem (Ω, T, F). Then, for any $q_{\min} \in [0,1]$,*

$$\Pr(\phi(T,F) \geq q_{\min}) \leq \frac{p + \text{Bias}(\mathcal{D}, \mathbf{t})}{q_{\min}} \quad \textit{where} \quad p = \frac{k}{|\Omega|}.$$

The size of the target set T is usually small relative to the size of the search space Ω, which means that p is also usually small. So, the probability that a search problem with an information resource drawn from $\mathcal{D}$ is favorable is bounded above by a small value. The bound tightens as we increase $q_{\min}$, our minimum threshold of success. However, introducing bias relaxes the bound.

To help understand this theorem, we contextualize it with an example. Consider a natural language processing task for understanding social media text. To train a model, one needs a large corpus of text, which serves as the information resource. Not all corpora will be equally helpful for this task. Intuitively, we know that other social media text would be the best training data, while text from modern books would be less helpful, and poems from the renaissance period would be near useless. This captures the core idea of this theorem, which states that in order to have a good chance of success, we need to sample from a distribution that is highly biased, meaning it places more weight on information resources that are well aligned with the target. In this example, when the set of possible training data extends all text, we need to sample according to a distribution that favors similar text like other social media over distant text like poetry. If the sampling is not biased, then the chance of obtaining good performance is very low. This is because the probability of a random training set being favorable for the given target is low.

The connection between bias and favorable resources is emphasized by the next theorem, demonstrating 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.

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 per-query probability of success. Then,

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

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

Following Theorem 6, if the algorithm does not induce bias aligned with $\mathbf{t}$ 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 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* $\text{Bias}(\mathcal{D}, \mathbf{t}) = 0$*, 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 [7] and Montañez's formal proof for the need for bias for multi-class classification [1]. This result shows that bias is necessary for any machine learning or search problem to have better than random chance performance.

The next theorem shows that the proportion of target sets for which our algorithm is highly biased is small, given that p is typically small relative to q_{min}.

Theorem 8 (Famine of Applicable Targets)**.** *Let $\mathcal{D}$ be a distribution over a finite set of information resources. Define*

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

$$\tau_{q_{\text{min}}} = \{T \mid T \in \tau_k, \text{Bias}(\mathcal{D}, \mathbf{t}) \geq q_{\text{min}}\}$$

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

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

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

A high value of $\text{Bias}(\mathcal{D}, \mathbf{t})$ implies that the algorithm, given $\mathcal{D}$, places a large amount of mass on $\mathbf{t}$ and a small amount of mass on other target functions. Consequently, our algorithm is acceptably biased toward fewer target sets as we increase our minimum threshold of bias. In other words, no algorithm can be strongly biased towards many targets simultaneously.

We can also prove a similar bound on the proportion of favorable targets by looking at the probability of success as opposed to bias.

Theorem 9 (Famine of Favorable Targets)**.** *For fixed $k \in \mathbb{N}$, fixed information resource F, and decomposable probability-of-success metric ϕ, define*

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

$$\tau_{q_{\text{min}}} = \{T \mid T \in \tau_k, \phi(T, F) \geq q_{\text{min}}\}$$

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

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

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

Lauw et al. [8] explores the inverse relation between algorithmic bias and expressivity for learning algorithms. Bias and expressivity are two aspects of a model's performance, where bias refers to a model's preference towards a specific target and expressivity refers to our algorithm's flexibility over all the elements. Their main theorem establishes a trade-off between an algorithm's bias and its expressivity. This shows that the specialization and the flexibility of a learning algorithm are inversely related, so a high preference towards a specific target reduces the potential flexibility of our algorithm and vice versa.

Let $\overline{P}_{\mathcal{D}}$ be the vector representation of the averaged probability distribution (conditioned on $\mathcal{D}$) induced on Ω during the course of the search.

Theorem 10 (Bias-Expressivity Trade-off)**.** *Given a distribution over information resources $\mathcal{D}$ and a fixed target function $\mathbf{t} \in \{0,1\}^{|\Omega|}$, entropic expressivity is bounded above in terms of bias,*

$$H(\overline{P}_{\mathcal{D}}) \leq \log_2 |\Omega| - 2\text{Bias}(\mathcal{D}, \mathbf{t})^2.$$

Additionally, bias is bounded above in terms of entropic expressivity,

$$\text{Bias}(\mathcal{D}, \mathbf{t}) \leq \sqrt{\frac{1}{2}(\log_2 |\Omega| - H(\overline{P}_{\mathcal{D}}))} = \sqrt{\frac{1}{2} D_{KL}(\overline{P}_{\mathcal{D}} \| \mathcal{U})}.$$

We give a corollary bound allowing us to bound bias as a function of the expected entropy of induced strategy distributions, rather than the entropic expressivity.

Corollary 1 (Bias Bound Under Expected Expressivity)**.**

$$\begin{aligned}\text{Bias}(\mathcal{D}, \mathbf{t}) &\leq \sqrt{\frac{1}{2}(\log_2 |\Omega| - \mathbb{E}_{\mathcal{D}}[H(\overline{P}_F)])} \\ &= \sqrt{\mathbb{E}_{\mathcal{D}}\left[\frac{1}{2} D_{KL}(\overline{P}_F \| \mathcal{U})\right]}.\end{aligned}$$

Lastly, the following theorem shows that entropic expressivity is bounded above and below with respect to the level of bias on a fixed target.

Theorem 11 (Expressivity Bounded By Bias)**.** *Given a fixed k-hot target function $\mathbf{t}$ and a distribution over information resources $\mathcal{D}$, the entropic expressivity of a search algorithm can be bounded in terms of $\epsilon := \text{Bias}(\mathcal{D}, \mathbf{t})$, by*

$$\begin{aligned}H(\overline{P}_{\mathcal{D}}) \in \Bigg[& H(p+\epsilon), \Big((p+\epsilon)\log_2\left(\frac{k}{p+\epsilon}\right) + \\ & (1-(p+\epsilon))\log_2\left(\frac{|\Omega|-k}{1-(p+\epsilon)}\right)\Big)\Bigg].\end{aligned}$$

C. Varying the Search Problem

A specific choice of target set $T \subset \Omega$ and information resource F defines a search problem. Instead of varying T and F separately, we can vary them together, allowing us to bound the proportion of favorable search problems.

Theorem 12 (Famine of Forte)**.** *Define*

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

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

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

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 expected per-query probability of success for algorithm $\mathcal{A}$ on problem (Ω, T, F). Then for any $m \in \mathbb{N}$,

$$\frac{|R_{q_{\min}}|}{|R|} \leq \frac{p}{q_{\min}} \qquad \textit{and} \qquad \lim_{m\to\infty} \frac{|R_{q_{\min}}|}{|R|} \leq \frac{p}{q_{\min}}$$

where $p = k/|\Omega|$.

For problems with sparse target sets (small p), favorable search problems are rare if we desire a strong probability of success (high $q_{\min}$). The greater the desired probability of success, the smaller the proportion of favorable search problems will be.

Instead of a probability-of-success metric, we can use an active information transform to measure the improvement in a search relative to uniform random sampling.

Definition 8 (Active Information)**.** *Let $\mathcal{U}$ be a uniform distribution of an unassisted search and let ϕ be the nonuniform measure for an assisted search [9]. If $\mathcal{U}(\mathrm{T})$ and $\phi(T)$ denote the probability over the target set $T \in \Omega$ and q is the probability of success, then the active information of the search is defined as*

$$I_+(\phi \mid \mathcal{U}) := \log_2 \frac{\phi(T)}{\mathcal{U}(T)} = \log_2 \frac{q}{p}.$$

Active information quantifies the effectiveness of assisted search against blind search—any search without active information will perform as well as blind search.

We generalize the Conservation of Active Information of Expectations [1] which allows us to bound the proportion of favorable problems as a function of the desired bits of information improvement instead of a desired probability of success.

Corollary 2 (The Conservation of Active Information of Expectations)**.** *Define $I_{\phi(T,F)} = -\log_2 p/\phi(T, F)$ to be a version of active information of expectations for decomposable metrics and let*

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

and

$$R_b = \{(T, F) \mid T \in \tau_k, F \in \mathcal{B}_m, I_{\phi(T,F)} \geq b\}.$$

Then for any $m \in \mathbb{N}$

$$\frac{|R_b|}{|R|} \leq 2^{-b}.$$

From this result, we have that finding a search problem for which an algorithm effectively reduces a search space by b bits requires at least b bits, so the information is conserved in this context.

D. Varying the Search Strategy

Alternatively, we can also fix the search problem. For a given problem, we can investigate the likelihood of performing well on it. Recall that over the course of a search, an algorithm induces a sequence of probability distributions $\tilde{P}$ over the search space Ω. Furthermore, the probability of success over this search is mathematically equivalent to a single query from an appropriately averaged distribution $P_\phi(\cdot \mid F)$. Therefore, each $P_\phi(\cdot \mid F)$ demarks an equivalence class of search algorithms mapping to the same averaged distribution. We refer to these classes as *search strategies*. Not only are favorable problems rare, so are favorable strategies.

Theorem 13 (Famine of Favorable Strategies)**.** *For any fixed search problem (Ω, t, f), set of probability mass functions $\mathcal{P} = \{P : P \in \mathbb{R}^{|\Omega|}, \sum_j P_j = 1\}$, and a fixed threshold $q_{min} \in [0, 1]$,*

$$\frac{\mu(\mathcal{G}_{t,q_{min}})}{\mu(\mathcal{P})} \leq \frac{p}{q_{min}},$$

where $\mathcal{G}_{t,q_{min}} = \{P : P \in \mathcal{P}, t^\top P \geq q_{min}\}$ and μ is Lebesgue measure. Furthermore, the proportion of possible search strategies giving at least b bits of active information of expectations is no greater than 2^{-b}.

Thus, for search problems with sparse target sets, favorable search strategies are rare if we desire a strong probability of success. Whether you fix the algorithm and try to match a problem to it, or fix the problem and try to match a strategy, both are provably difficult.

V. Applications of the Framework

Now that we have described the ASF and given several theorems, we will show how the framework can be applied to specific learning problems to gain insights.

add results that come from direct application of existing theorems

A. Classification Algorithms

We begin with classification algorithms. First, we must reduce the classification problem to a search problem.

Input values of a classification problem are given in the form of real-valued variables (often data points with categorical class labels). These data points are contained in a training set D and have corresponding class labels. We also have a test data set V. Using a loss function $\mathcal{L}$, a learning algorithm $\mathcal{A}$ chooses from a finite set $\mathcal{G}$ of classification hypotheses, which are possible ways of labeling V. The classification problem can be reduced to the ASF as such:

- $\Omega = \mathcal{G}$
- $T = \{g : g \in \Omega, \Xi(\mathcal{L}, g, V) \leq \epsilon\}$
- $F = \{\mathcal{V}, \mathcal{L}\}$
- $F(\emptyset) = \mathcal{D}$
- $F(h) = \mathcal{L}(\omega_i)$ where ω_i is the ith query in the search process

Having reduced the problem of classification to the ASF, we can now develop an information-theoretic perspective of the problem and prove new results. Here, we present one of

the main theorems of Bashir et al. [10], developed within the context of the ASF.

First we must introduce the concept of algorithmic capacity. The capacity $C_{\mathcal{A}}$ of an algorithm is the maximum amount of information that $\mathcal{A}$ can extract from a dataset $D \sim \mathcal{D}$. For a fixed distribution $\mathcal{D}$, we can define the capacity relative to that particular distribution.

Definition 9 (Distributional Algorithm Capacity)**.** *For $D \sim \mathcal{D}$,*

$$C_{\mathcal{A},\mathcal{D}} = I(G; D).$$

When viewed in the light of the ASF, we can then upper bound the distributional algorithm capacity using bias.

Theorem 14 (Distributional Capacity Upper Bound)**.**

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

where $\overline{\mathbf{P}}_F$ is the expected average conditional distribution on the search space given F.

This theorem tells us that the maximum amount of information that a classification algorithm can extract decreases with respect to the algorithm's bias. This is an example of how by examining a specific class of algorithms within the framework, we can derive specific theorems.

B. Transfer Learning

We can also apply the ASF to transfer learning. In a transfer learning problem, one takes insight learned from one machine learning problem and applies it to another. We call these the source problem and recipient problem, respectively. The source problem has target set T_S and information resource F_S, and the recipient problem has target set T_R and information resource F_R.

We are interested in the learned knowledge that is passed over from the source, and so we denote this information as L, in the form of a binary string. Then F_{R+L} is the information resource of the recipient problem combined with the learned information from the source problem. With this notation, we can use the framework and define a probability of success metric, ϕ_{TL}, for the recipient problem after transfer learning. Furthermore, we can bound this probability of success using information theoretic quantities [3].

Theorem 15 (Transfer Learning under Dependence)**.** *Define*

$$\phi_{TL} \coloneqq \mathbb{E}_{T_R, F_{R+L}}\left[\phi(T_R, F_{R+L})\right] = Pr(\omega \in T_R; \mathcal{A})$$

as the probability of success for transfer learning. Then,

$$\phi_{TL} \leq \frac{I(F_S; T_R) + I(F_R; T_R) + D(P_{T_R} \| \mathcal{U}_{T_R}) + 1}{I_\Omega}$$

where $I_\Omega = -\log |T_R|/|\Omega|$, ($T_R$ being of fixed size), $D(P_{T_R} \| \mathcal{U}_{T_R})$ is the Kullback-Leibler divergence between the marginal distribution on T_R and the uniform distribution on T_R, and $I(F; T)$ is the mutual information.

With this, we can even bound the difference in performance between the problem with and without transfer learning [3].

Theorem 16 (Success Difference from Distribution Divergence)**.** *Given the performance of a search algorithm on the recipient problem in the transfer learning case, ϕ_{TL}, and without the learning resource, ϕ_{NoTL}, we can upperbound the absolute difference as*

$$|\phi_{TL} - \phi_{NoTL}| \leq |T| \sqrt{\frac{1}{2} D_{KL}(\mathbf{P}_{TL} \| \mathbf{P}_{NoTL})},$$

where $\mathbf{P}_{TL}$ and $\mathbf{P}_{NoTL}$ are probability distributions on the search space induced during the search by the algorithm with and without transfer learning, respectively.

This shows that unless using the learning resource significantly changes the resulting distribution over the search space, the change in performance from transfer learning will be minimal. This is another example of how we can prove results by reducing specific learning problems into the ASF.

VI. Future Directions

There are both applied and theoretical future directions for the ASF. Applied future work explores the application of the ASF to existing learning algorithms. Recent work by Bekerman et al. empirically estimates the inductive orientation vectors of various algorithms [11]. The inductive orientation vector (Definition 6) represents the probability distribution over the search space and captures aspects of the algorithm's inductive bias, which determines how it interacts with data. When clustering is performed on these vectors, it notably groups the tree-based algorithms (Random Forest, Decision Tree, Adaboost, Gradient Boosting), indicating that their implementation similarly affects their vectors. The results of Bekerman et al. further suggest that changing the hyperparameters of an algorithm may not fundamentally change the algorithm. Future applied work can explore the uses of inductive orientation vectors or otherwise investigate the real-world counterparts of the ASF.

Theoretical future work seeks to represent concepts from learning algorithms within the framework and prove mathematical results for them. An important recent contribution is the upper-bounding of generalization error within the ASF. Generalization error measures how accurately an algorithm can predict values for unseen data, and is a necessary component for understanding supervised learning algorithms. Ramalingam et al. showed that generalization error can be bounded above in terms of distributional algorithm capacity (Definition 9) and the inductive orientation vector . Future theoretical work can continue proving results for generalization or similarly bound other learning algorithm quantities.

cite unpublished paper

VII. Conclusion

The results presented provide not only a unified way of bounding the performance of artificial learning algorithms, but also a more intuitive way of understanding how these algorithms work.

As mentioned earlier, artificial intelligence and machine learning are typically viewed as separate research areas, such that researchers must prove results independently for each new

algorithm. However, since we prove the results above within our unified algorithmic search framework, our results are applicable to all artificial learning algorithms that are reducible into this framework. That is, if we reduce a new algorithm, then all the results provided above will hold true. This is powerful since artificial intelligence and machine learning are rapidly growing fields in which researchers are continuously developing new algorithms and building upon previous ones. As such, it is useful to have a search framework with which we can reason about the bounds on the performance of these new algorithms without needing to develop new methods for each new model.

The algorithmic search framework also provides a more intuitive way to understand how artificial learning algorithms work. Search is inherently a physical phenomenon, one that is familiar to people from an early age. As children, we learn to crawl and search for toys and other objects, using the information we have from past exploration of our environment. Since physical search is such a common human experience, reducing artificial learning algorithms into our algorithmic search framework provides a more intuitive way to understand and reason about otherwise much more complex and abstract algorithms.

look into future work with ASF and put that in conclusion

REFERENCES

[1] G. D. Montanez, "The famine of forte: few search problems greatly favor your algorithm," in *2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. IEEE, 2017, pp. 477–482.

[2] ——, "Why machine learning works," 2017.

[3] J. Williams, A. Tadesse, T. Sam, H. Sun, and G. D. Montañez, "Limits of Transfer Learning," *The Sixth International Conference on Machine Learning, Optimization, and Data Science (LOD 2020)*, 2020. [Online]. Available: https://arxiv.org/abs/2006.12694

[4] G. D. Montañez, L. Sanders, and H. C. Deshong, "Minimal Complexity Requirements for Proteins and Other Combinatorial Recognition Systems," in *Proceedings of the 13th International Joint Conference on Biomedical Engineering Systems and Technologies (BIOSTEC 2020) - Volume 3: BIOINFORMATICS, Valletta, Malta, February 24-26, 2020*, E. D. Maria, A. L. N. Fred, and H. Gamboa, Eds. SCITEPRESS, 2020, pp. 71–78. [Online]. Available: https://doi.org/10.5220/0008856800710078

[5] G. D. Montañez, J. Hayase, J. Lauw, D. Macias, A. Trikha, and J. Vendemiatti, "The Futility of Bias-Free Learning and Search," in *32nd Australasian Joint Conference on Artificial Intelligence*. Springer, 2019, pp. 277–288.

[6] T. Sam, J. Williams, A. Tadesse, H. Sun, and G. D. Montañez, "Decomposable Probability-of-Success Metrics in Algorithmic Search," 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. 785–792. [Online]. Available: https://doi.org/10.5220/0009098807850792

[7] T. M. Mitchell, *The need for biases in learning generalizations*. Department of Computer Science, Laboratory for Computer Science Research ..., 1980.

[8] 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

[9] W. Dembski and R. II Marks, "The search for a search: Measuring the information cost of higher level search." *JACIII*, vol. 14, pp. 475–486, 07 2010.

[10] 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. [Online]. Available: http://arxiv.org/abs/2010.06076

[11] 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.

[12] V. Vapnik, "An overview of statistical learning theory," *IEEE Transactions on Neural Networks*, vol. 10, no. 5, pp. 988–999, 1999.

[13] R. M. Fano, "Transmission of information: A statistical theory of communications," *American Journal of Physics*, vol. 29, no. 11, pp. 793–794, 1961.

[14] T. M. Cover and J. A. Thomas, *Elements of information theory*. John Wiley & Sons, 2012.

APPENDIX

A. Additional Reductions

We next present several reductions to the search framework.

1) Vapnik's Learning Framework.: Vapnik-Chervonenkis theory is a statistical framework for understanding particular learning problems, and can be applied to regression, density estimation, and classification. For any particular learning problem, we have a space Z of elements, where each element z describes an input-output pair $(\mathbf{x}, y)$. Given a supervisor which for every input $\mathbf{x}$ returns the corresponding output y , we wish to find a function $f(\mathbf{x})$ that best estimates the labels for these elements. In other words, we wish to minimize some loss function $Q_\alpha(z)$ for some element z in the space Z and some parameter α in the hyperparameter space Λ.

In Vapnik's Framework, we consider the situation where we want to minimize the expected value of the loss, the risk functional $R(\alpha) = \int Q_\alpha(z)dP(z)$, where $P(z)$ is the probability measure defined on space Z. However, $P(z)$ is unknown and we only have an i.i.d. sample $z_1, \ldots, z_\ell$ of elements sampled from $P(z)$.

The search problem then becomes the question: *what is the loss function $Q_\alpha(z)$ for this problem that will allow us to choose a function $f(\mathbf{x})$ that will best minimize the risk functional $R(\alpha)$?* Note that in practice, $P(z)$ is unknown. Thus, instead of the idealized $R(\alpha)$, we instead minimize the empirical risk [12], $R_{emp}(\alpha) = \frac{1}{\ell}\Sigma_{i=1}^{\ell} Q_\alpha(z_i)$, to help us find α.

Assuming Λ is finite, and choosing $\epsilon \in \mathbb{R}_{\geq 0}$, we can represent this framework in terms of the ASF as follows:

- $\Omega = \Lambda$
- $T = \{\alpha : R(\alpha) - \text{argmin}_{\alpha' \in \Lambda} R(\alpha') < \epsilon\}$
- $F = \{z_1, \ldots, z_\ell\}$
- $F(\emptyset) = \{z_1, \ldots, z_\ell\}$
- $F(\alpha) = R_{emp}(\alpha)$

The target set T uses $R(\alpha) - \text{argmin}_{\alpha' \in \Lambda} R(\alpha') < \epsilon$ to account for cases where the model is imperfect or there are limited representations available. Since $P(z)$ is unknown, we choose $F(\alpha) = R_{emp}(\alpha)$, which is constructed using the training set. Following Vapnik, this is based on the empirical risk minimization induction principle (ERM principle). Therefore, any finite problem represented in Vapnik's learning framework can also be represented in our search framework. In the following sections, we will present reductions of regression,

density estimation, and classification problems to Vapnik's learning framework.

Pattern Recognition. Pattern recognition or classification algorithms detect categorizations among points in a dataset. The problem of pattern recognition can be modeled in Vapnik's learning framework as follows: let $\mathbf{x}$ be a random input vector, $y = \{0, 1\}$, and $f(\mathbf{x}, \alpha), \alpha \in \Lambda$ be a set of indicator functions. And finally, consider the loss function:

$$L(y, f(\mathbf{x}, \alpha)) = \begin{cases} 0 & \text{if } y = f(\mathbf{x}, \alpha) \\ 1 & \text{if } y \neq f(\mathbf{x}, \alpha) \end{cases}$$

The risk functional

$$R(\alpha) = \int Q_\alpha(z) dP(z) = \int L(y, f(\mathbf{x}, \alpha)) dP(\mathbf{x}, y)$$

therefore represents the probability of a classification error. The goal of the learning problem is to find the function that minimizes the probability of classification errors when the underlying $P(\mathbf{x}, y)$ is unknown but the data is given.

Regression. Given a set of inputs in the form of real-valued variables, a learning algorithm uses *regression* to learn a function $f : \mathbb{R} \to \mathbb{R}$ from the inputs to a real-valued output. In addition to directly reducing regression to the ASF as in Section II, regression can be modeled in Vapnik's learning framework as follows: let $\mathbf{x}$ be a random input vector, y be a real number, and $f(\mathbf{x}, \alpha), \alpha \in \Lambda$ be a set of real functions. Finally, if $f(\mathbf{x}, \alpha) \in L_2$, then consider the following loss function:

$$L(y, f(\mathbf{x}, \alpha)) = (y - f(\mathbf{x}, \alpha))^2$$

The goal of the learning problem is to find the function that minimizes the corresponding risk functional when $P(\mathbf{x}, y)$ is unknown but the data are given.

Density Estimation. Density estimation algorithms recreate a probability density function, given a data set of k-length vectors $\{\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_n}\}$. The known element and label pairs $(\mathbf{x}, y) \in Z$ can then be constructed $(\mathbf{x_i}, F(\mathbf{x_i}))$ where

$$F(\mathbf{x}) = \frac{1}{n} \sum_{j=1}^{n} \theta(x_1 - x_{j,1}) \theta(x_2 - x_{j,2}) \ldots \theta(x_k - x_{j,k})$$

and $\theta(x)$ is defined

$$\theta(x) = \begin{cases} 1 & \text{if } x > 0 \\ 0 & \text{if } x \leq 0 \end{cases}$$

Effectively, $F(\mathbf{x})$ counts proportion of the vectors $\mathbf{x_j}$ in our data set, which for every entry in $\mathbf{x_j}$ is less than the corresponding entry in $\mathbf{x}$.

The problem of recreating the probability density function then becomes the regression problem on our data set to find the cumulative distribution function $p(\mathbf{x}, \alpha), \alpha \in \Lambda$.

2) Learning Algorithms.: Supervised learning aims to approximate a true signal relating features X to responses Y, which can be viewed either as a function $f : X \to Y$ or a distribution $P(Y \mid X)$. A *learning algorithm* is our general term for a machine learning approach that takes in data to produce models or hypotheses, and is equipped with a hypothesis space $\mathcal{G}$ containing guesses for the target function or distribution.

With this, a learning algorithm $\mathcal{A}$ can be viewed as a stochastic map $P_{G|D}$ that takes as input a training set D of size n, namely $D = (Z_1, \ldots, Z_n)$ whose elements belong to an instance space $\mathcal{Z}$, and outputs a hypothesis $g \in \mathcal{G}$. Note that each instance Z_i in the dataset could also represent a pair $(\boldsymbol{x}_i, y_i)$, where $\boldsymbol{x}_i$ is a feature vector and y_i is the corresponding label or response. For future results, let G denote the random variable representing the output of $\mathcal{A}$ with input D.

Having presented our definition of a learning algorithm, we now reduce it to the ASF.

Our search space Ω is the hypothesis space $\mathcal{G}$ of $\mathcal{A}$. When called on the empty set, the external information resource returns a training dataset D of size n; when evaluating a specific element of the search space, it returns $F(h)$, which is a non-negative loss function. The target set T is the set of all hypotheses $g \in \mathcal{G}$ that achieve low population risk, namely,

$$R_D(g) = \mathbb{E}[\ell(g, \mathcal{Z})] = \int_{\mathcal{Z}} \ell(g, z) \mathcal{D}(dz) < \epsilon$$

for some fixed scalar $\epsilon > 0$ for any data-generating distribution $\mathcal{D}$. However, since $\mathcal{D}$ is unknown, we can compute the empirical risk of g on dataset D as

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

Finally, note that the history of the sampled elements corresponds to hypotheses considered while $\mathcal{A}$ is trained, for example using a method such as stochastic gradient descent. To summarize, the following represents a learning algorithm as search:

- $\Omega = \mathcal{G}$
- $T = \{g \in \mathcal{G} \mid R_{\mathcal{D}}(g)\}$
- $F(\emptyset) = D$
- $F(h) = \ell : \mathcal{G} \times \mathcal{Z} \to \mathbb{R}_{\geq 0}$

3) Clustering as Search.: Given a sample of points in a vector space, the output of a clustering problem is an assignment of these input points to a fixed set of distinct groupings, or *clusters*. *Soft-clustering* algorithms can assign a single point to multiple clusters, whereas a *hard-clustering* algorithm maps each point to a single cluster. We assume a finite number of points, and that the number of clusters does not exceed the number of points.

Let $\mathcal{A}$ be a clustering algorithm. Any clustering of N points to K clusters is represented by an $N \times K$ matrix W, where entry W_{ij} represents the membership of point i in cluster j. $\mathcal{A}$ chooses from a space $\mathcal{G}$ of all possible assignment matrices

W, according to a loss function $\mathcal{L}$, and the true assignment matrix is represented as W^*. Assume both $\mathcal{A}$ and the set of N points are both fixed and given. We cast the clustering problem to the ASF as such:

- $\Omega = \mathcal{G}$
- $T = \{W : W \in \Omega, \mathcal{L}(W, W^*) \leq \epsilon\}$
- $F =$ a binary-coded vector representation of all N points
- $F(\emptyset) =$ the entire set of points
- $F(h) = W$

Here, we assume that the external information source's algorithm acts as a single-query algorithm; thus, it will produce at most one W matrix.

4) Parameter Estimation as Search.: Given a vector θ, the output of a parameter estimation problem is a data set D. A parameter estimation algorithm $\mathcal{A}$ uses D to search a finite space of possible vectors θ for the true parameter vector. We cast parameter estimation as follows:

- $\Omega = \mathcal{G}$
- $T = \{\theta' : \theta' \in \Omega, \mathcal{L}(\theta', \theta) \leq \epsilon\}$, where $\mathcal{L}$ is a loss function
- $F = D$
- $F(\emptyset) =$ the given vectors
- $F(h) =$ the correct parameter vector θ

5) Hyperparameter Optimization as Search.: The goal of hyperparameter optimization is to choose the optimal hyperparameters for a given learning algorithm. This lends itself well to the search framework. Let Λ be a discretized hyperparameter space, and let $\phi(\lambda)$ be a metric that measures how well an algorithm performs using the hyperparameter configuration λ. Let $\phi(\lambda^*)$ represent the best performance possible using hyperparameter configurations within Λ. We can now represent hyperparameter optimization in the ASF as follows:

- $\Omega = \Lambda$
- $T = \{\lambda : \lambda \in \Lambda, |\phi(\lambda) - \phi(\lambda^*)| < \epsilon\}$, where $\epsilon > 0$
- $F = \{\phi(\lambda_1), \ldots, \phi(\lambda_{|\Omega|})\}$
- $F(\emptyset) = \emptyset$
- $F(h) = \phi(\lambda_i)$

In cases of sequential hyperparameter optimization, the algorithm would first sample elements from the hyperparameter search space and use F to evaluate them. This information would then be added to the algorithm's history, which then informs which elements are sampled next.

B. Proofs

Theorem 2 (Conservation of Bias)**.** *Let $\mathcal{D}$ be a distribution over a set of information resources and let $\tau_k = \{\mathbf{t}|\mathbf{t} \in \{0,1\}^{|\Omega|}, \|\mathbf{t}\| = \sqrt{k}\}$ be the set of all target functions $\mathbf{t}$ that are $|\Omega|$-length k-hot vectors. Then for any fixed algorithm $\mathcal{A}$,*

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

Proof.

$$\begin{aligned}
\sum_{t \in \tau_k} \text{Bias}(\mathcal{D}, \mathbf{t}) &= \sum_{t \in \tau_k} \mathbb{E}_{\mathcal{D}}[\mathbf{t}^\top P] - p \\
&= \sum_{t \in \tau_k} \mathbb{E}_{\mathcal{D}}[\mathbf{t}^\top P] - \sum_{t \in \tau_k} \frac{k}{n} \\
&= \mathbb{E}_{\mathcal{D}}\left[\sum_{t \in \tau_k} \mathbf{t}^\top P\right] - \sum_{t \in \tau_k} \frac{k}{n} \\
&= \mathbb{E}_{\mathcal{D}}\left[\binom{n-1}{k-1}\mathbf{1}^\top P\right] - \binom{n}{k}\frac{k}{n} \\
&= \mathbb{E}_{\mathcal{D}}\left[\binom{n-1}{k-1}\right] - \binom{n-1}{k-1} \\
&= 0.
\end{aligned}$$

□

Lemma 1 (Maximum probability mass over a target set)**.** *Let $\tau_k = \{\mathbf{t}|\mathbf{t} \in \{0,1\}^{|\Omega|}, \|\mathbf{t}\| = \sqrt{k}\}$ be the set of all $|\Omega|$-length k-hot vectors. Given an arbitrary probability distribution P,*

$$\sup_{\mathbf{t} \in \tau_k} \mathbf{t}^\top P \leq 1 - \left(\frac{1-p}{p}\right) \inf_{\mathbf{t} \in \tau_k} \mathbf{t}^\top P$$

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

Proof. We proceed by contradiction. Suppose that

$$\sup_{\mathbf{t} \in \tau_k} \mathbf{t}^\top P > 1 - \left(\frac{1-p}{p}\right) \inf_{\mathbf{t} \in \tau_k} \mathbf{t}^\top P.$$

Then, there exists some target function $\mathbf{t} \in \tau_k$ such that

$$\mathbf{t}^\top P > \left(\frac{1-p}{p}\right) \inf_{\mathbf{t} \in \tau_k} \mathbf{t}^\top P.$$

By Lemma 2 there exists a k-sized subset of the complementary target set with total probability mass q such that

$$\begin{aligned}
q &\leq \frac{k}{|\Omega| - k}\left(\mathbf{s}^\top P\right) \\
&< \frac{k}{|\Omega| - k}\left(\left(\frac{1-p}{p}\right) \inf_{\mathbf{t} \in \tau_k} \mathbf{t}^\top P\right) \\
&= \frac{k}{|\Omega| - k}\left(\left(\frac{|\Omega| - k}{k}\right) \inf_{\mathbf{t} \in \tau_k} \mathbf{t}^\top P\right) \\
&= \inf_{\mathbf{t} \in \tau_k} \mathbf{t}^\top P.
\end{aligned}$$

Thus, we can always find a target set with total probability mass strictly less than $\inf_{\mathbf{t} \in \tau_k} \mathbf{t}^\top P$, which is a contradiction. Therefore, we have proven that

$$\sup_{\mathbf{t} \in \tau_k} \mathbf{t}^\top P \leq 1 - \left(\frac{1-p}{p}\right) \inf_{\mathbf{t} \in \tau_k} \mathbf{t}^\top P.$$

□

Lemma 2 (Existence of subset with at most uniform mass)**.** *Given an n-sized subset of S of the sample space of an arbitrary probability distribution with total probability mass*

M_S, there exists a k-sized proper subset $R \subset S$ with total probability mass M_R such that

$$M_R \leq \frac{k}{n} M_S.$$

Proof. We proceed by induction on the size k.

Base Case: When $k = 1$, there exists an element with total probability mass at most $\frac{M_S}{n}$, since for any element in S that has probability mass greater than the uniform mass $\frac{M_S}{n}$, there exists an element with mass strictly less than $\frac{M_S}{n}$ by the law of total probability. This establishes our base case.

Inductive Hypothesis: Suppose that a k-sized subset $R_k \subset S$ exists with total probability mass M_{R_k} such that $M_{R_k} \leq \frac{k}{n} M_S$.

Induction Step: We show that there exists a subset $R_{k+1} \subset S$ of size $k+1$ with total probability mass $M_{R_{k+1}}$ such that $M_{R_{k+1}} \leq \frac{k+1}{n} M_S$.

First, let $M_{R_k} = \frac{k}{n} M_S - s$, where $s \geq 0$ represents the slack between M_{R_k} and $\frac{k}{n} M_S$. Then, the total probability mass on ${R_k}^{\mathrm{c}} := S \setminus R_k$ is

$$M_{{R_k}^{\mathrm{c}}} = M_S - M_{R_k} = M_S - \frac{k}{n} M_S + s.$$

Given that $M_{{R_k}^{\mathrm{c}}}$ is the total probability mass on set ${R_k}^{\mathrm{c}}$, either each of the $n-k$ elements in ${R_k}^{\mathrm{c}}$ has a uniform mass of $M_{{R_k}^{\mathrm{c}}}/(n-k)$, or they do not. If the probability mass is uniformly distributed, let e be an element with mass exactly $M_{{R_k}^{\mathrm{c}}}/(n-k)$. Otherwise, for any element e' with mass greater than $M_{{R_k}^{\mathrm{c}}}/(n-k)$, by the law of total probability there exists an element $e \in {R_k}^{\mathrm{c}}$ with mass less than $M_{{R_k}^{\mathrm{c}}}/(n-k)$. Thus, in either case there exists an element $e \in {R_k}^{\mathrm{c}}$ with mass at most $M_{{R_k}^{\mathrm{c}}}/(n-k)$.

Then, the set $R_{k+1} = R_k \cup \{e\}$ has total probability mass

$$\begin{aligned}
M_{R_{k+1}} &\leq M_{R_k} + \frac{M_{{R_k}^{\mathrm{c}}}}{n-k} \\
&= \frac{k}{n} M_S - s + \frac{M_S - \frac{k}{n} M_S + s}{n-k} \\
&= \frac{k M_S (n-k) + n(M_S - \frac{k}{n} M_S + s)}{n(n-k)} - s \\
&= \frac{knM_S - k^2 M_S + nM_S - kM_S + ns}{n(n-k)} - s \\
&= \frac{(n-k)(kM_S + M_S) + ns}{n(n-k)} - s \\
&= \frac{k+1}{n} M_S + \frac{s}{n-k} - s \\
&= \frac{k+1}{n} M_S + \frac{s(1+k-n)}{n-k} \\
&\leq \frac{k+1}{n} M_S
\end{aligned}$$

where the final inequality comes from the fact that $k < n$. Thus, if a k-sized subset $R_k \in S$ exists such that $M_{R_k} \leq \frac{k}{n} M_S$, a $k+1$-sized subset $R_{k+1} \in S$ exists such that $M_{R_{k+1}} \leq \frac{k+1}{n} M_S$.

Since the base case holds true for $k = 1$ and the inductive hypothesis implies that this rule holds for $k+1$, we can always find a k-sized subset $R_k \in S$ such that

$$M_{R_k} \leq \frac{k}{n} M_S.$$

□

Theorem 1 (Bias Upper Bound)**.** *Let $\tau_k = \{\mathbf{t} \mid \mathbf{t} \in \{0,1\}^{|\Omega|}, \|\mathbf{t}\| = \sqrt{k}\}$ be the set of all $|\Omega|$-length k-hot vectors and let $\mathcal{B}$ be a finite set of information resources. Then,*

$$\sup_{\mathbf{t} \in \tau_k} \mathrm{Bias}(\mathcal{B}, \mathbf{t}) \leq \left(\frac{p-1}{p} \right) \inf_{\mathbf{t} \in \tau_k} \mathrm{Bias}(\mathcal{B}, \mathbf{t}) \quad \textit{where} \quad p = \frac{k}{|\Omega|}.$$

Proof. First, define

$$m := \inf_{\mathbf{t} \in \tau_k} \mathbb{E}_{\mathcal{U}[\mathcal{B}]}[\mathbf{t}^\top \overline{P}_F] = \inf_{\mathbf{t} \in \tau_k} \mathrm{Bias}(\mathcal{B}, \mathbf{t}) + p$$

and

$$M := \sup_{\mathbf{t} \in \tau_k} \mathbb{E}_{\mathcal{U}[\mathcal{B}]}[\mathbf{t}^\top \overline{P}_F] = \sup_{\mathbf{t} \in \tau_k} \mathrm{Bias}(\mathcal{B}, \mathbf{t}) + p.$$

By Lemma 1,

$$M \leq 1 - \left(\frac{1-p}{p} \right) m.$$

Substituting the values of m and M,

$$\begin{aligned}
\sup_{\mathbf{t} \in \tau_k} \mathrm{Bias}\left(\mathcal{B}, \mathbf{t}\right) &\leq 1 - p - \left(\frac{1-p}{p} \right) \\
&\quad \left(\inf_{\mathbf{t} \in \tau_k} \mathrm{Bias}\left(\mathcal{B}, \mathbf{t}\right) + p \right) \\
&= \left(\frac{p-1}{p} \right) \inf_{\mathbf{t} \in \tau_k} \mathrm{Bias}\left(\mathcal{B}, \mathbf{t}\right).
\end{aligned}$$

□

Theorem 3 (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 set of target sets τ closed under permutation, any set of information resources $\mathcal{B}$, and decomposable probability-of-success metric ϕ,*

$$\sum_{T \in \tau} \sum_{F \in \mathcal{B}} \phi_{\mathcal{A}_1}(T, F) = \sum_{T \in \tau} \sum_{F \in \mathcal{B}} \phi_{\mathcal{A}_2}(T, F). \tag{1}$$

Proof. Note that the closed under permutation condition implies $\sum_{\mathbf{t}\in\tau}\mathbf{t} = [c, c, \ldots, c] = \mathbf{1}\cdot c$ for some constant c.

$$\begin{aligned}
\sum_{t\in\tau}\sum_{f\in\mathcal{B}}\phi_{\mathcal{A}_1}(t,f) &= \sum_{\mathbf{t}\in\tau}\sum_{f\in\mathcal{B}}\mathbf{P}^{\top}_{\phi,f,\mathcal{A}_1}\mathbf{t}\\
&= \sum_{f\in\mathcal{B}}\mathbf{P}^{\top}_{\phi,f,\mathcal{A}_1}\sum_{\mathbf{t}\in\tau}\mathbf{t}\\
&= \sum_{f\in\mathcal{B}}\mathbf{P}^{\top}_{\phi,f,\mathcal{A}_1}\mathbf{1}\cdot c\\
&= c\sum_{f\in\mathcal{B}}\mathbf{P}^{\top}_{\phi,f,\mathcal{A}_1}\mathbf{1}\\
&= c\sum_{f\in\mathcal{B}}1\\
&= c\sum_{f\in\mathcal{B}}\mathbf{P}^{\top}_{\phi,f,\mathcal{A}_2}\mathbf{1}\\
&= \sum_{\mathbf{t}\in\tau}\sum_{f\in\mathcal{B}}\mathbf{P}^{\top}_{\phi,f,\mathcal{A}_2}\mathbf{t}\\
&= \sum_{t\in\tau}\sum_{f\in\mathcal{B}}\phi_{\mathcal{A}_2}(t,f)
\end{aligned}$$

where the first and final equalities follow from the definition of decomposability. □

Theorem 4 (Learning Under Dependence)**.** *Define*

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

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

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

Then,

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

where $I_\Omega = -\log k/|\Omega|$, $D(P_T\|\mathcal{U}_T)$ is the Kullback-Leibler 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

$$\Pr(\omega\in T;\mathcal{A}) \le \frac{H(\mathcal{U}_T) - H(T\mid F) + 1}{I_\Omega}$$

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

Proof. This proof loosely follows that of Fano's Inequality [13], being a reversed generalization of it. Let $Z = \mathbb{1}(\mathrm{X}\in T)$. Using the chain rule for entropy to expand $H(Z,T|\mathrm{X})$ in two different ways, we get

$$\begin{aligned}
H(Z,T|\mathrm{X}) &= H(Z|T,\mathrm{X}) + H(T|\mathrm{X})\\
&= H(T|Z,\mathrm{X}) + H(Z|\mathrm{X}).
\end{aligned}$$

By definition, $H(Z|T,\mathrm{X}) = 0$, and by the data processing inequality $H(T|F)\le H(T|\mathrm{X})$. Thus,

$$H(T|F) \le H(T|Z,\mathrm{X}) + H(Z|\mathrm{X}).$$

Define $P_g = \Pr(\mathrm{X}\in T;\mathcal{A}) = P(Z=1)$. Then,

$$\begin{aligned}
H(T|Z,\mathrm{X}) &= (1-P_g)H(T|Z=0,\mathrm{X}) + P_g H(T|Z=1,\mathrm{X})\\
&\le (1-P_g)\log\binom{|\Omega|}{k} + P_g\log\binom{|\Omega|-1}{k-1}\\
&= \log\binom{|\Omega|}{k} - P_g\log\frac{|\Omega|}{k}.
\end{aligned}$$

We let $H(\mathcal{U}_T) = \log(\frac{|\Omega|}{k})$, being the entropy of the uniform distribution over k-sparse target sets in Ω. Therefore,

$$H(T|F) \le H(\mathcal{U}_T) - P_g\log\frac{|\Omega|}{k} + H(Z|\mathrm{X}).$$

Using the definitions of conditional entropy and I_Ω, we get

$$H(T) - I(T;F) \le H(\mathcal{U}_T) - P_g I_\Omega + H(Z|\mathrm{X}),$$

which implies

$$\begin{aligned}
P_g I_\Omega &\le I(T;F) + H(\mathcal{U}_T) - H(T) + H(Z|\mathrm{X})\\
&= I(T;F) + D(P_T\,\|\,\mathcal{U}_T) + H(Z|\mathrm{X}).
\end{aligned}$$

Examining $H(Z|\mathrm{X})$, we see it captures how much entropy of Z is due to the randomness of T. We upper-bound this by its maximum value of 1 and obtain

$$\Pr(\mathrm{X}\in T;\mathcal{A}) \le \frac{I(T;F) + D(P_T\,\|\,\mathcal{U}_T) + 1}{I_\Omega},$$

and substitute q for $\Pr(\mathrm{X}\in T;\mathcal{A})$ to obtain the first result, noting that $q = \mathbb{E}_{T,F}\left[P_\phi(\omega\in T|F)\right]$ specifies a proper probability distribution by the linearity and boundedness of the expectation. To obtain the second form, use the definitions $I(T;F) = H(T) - H(T|F)$ and $D(P_T\,\|\,\mathcal{U}_T) = H(\mathcal{U}_T) - H(T)$. □

Theorem 5 (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 $\mathbf{t}$, and let $\phi(T,F)$ be a decomposable probability of success for algorithm $\mathcal{A}$ on search problem (Ω,T,F). Then, for any $q_{\min}\in[0,1]$,*

$$\Pr(\phi(T,F)\ge q_{\min}) \le \frac{p + \mathrm{Bias}(\mathcal{D},\mathbf{t})}{q_{\min}} \quad \textit{where} \quad p = \frac{k}{|\Omega|}.$$

Proof. We seek to bound the probability of achieving a successful search on target function $\mathbf{t}$ with information resource F. By Definition 3, it follows that

$$\begin{aligned}
\Pr(\phi(T,F)\ge q_{\min}) &= \Pr(P_\phi(\omega\in T|F)\ge q_{\min})\\
&= \Pr\left(\mathbf{t}^{\top}\mathbf{P}_{\phi,F}\ge q_{\min}\right)
\end{aligned}$$

where $\omega\in T$ means the target function $\mathbf{t}$ evaluated at ω is one, and $\mathbf{P}_{\phi,F}$ represents the $|\Omega|$-length probability vector defined by $P_\phi(\cdot|F)$. Applying Markov's Inequality,

$$\begin{aligned}
\Pr(\phi(T,F)\ge q_{\min}) &\le \frac{1}{q_{\min}}\mathbb{E}_{\mathcal{D}}[\mathbf{t}^{\top}\mathbf{P}_{\phi,F}]\\
&= \frac{p + \mathrm{Bias}(\mathcal{D},\mathbf{t})}{q_{\min}}.
\end{aligned}$$

□

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 per-query probability of success. Then,

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

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

Proof. We seek to bound the proportion of successful search problems for which $\phi(t, f) \geq q_{\min}$ for any threshold $q_{\min} \in (0, 1]$. Let $F \sim \mathcal{U}[\mathcal{B}]$. Then,

$$\begin{aligned}\frac{|\mathcal{B}_{q_{\min}}|}{|\mathcal{B}|} &= \frac{1}{|\mathcal{B}|} \sum_{f \in \mathcal{B}} \mathbb{1}_{\phi(t,f) \geq q_{\min}} \\ &= \mathbb{E}_{\mathcal{U}[\mathcal{B}]}[\mathbb{1}_{\phi(t,F) \geq q_{\min}}] \\ &= \Pr(\phi(t, F) \geq q_{\min}).\end{aligned}$$

By decomposability, we have

$$\frac{|\mathcal{B}_{q_{\min}}|}{|\mathcal{B}|} = \Pr(\mathbf{t}^\top \mathbf{P}_{\phi,F} \geq q_{\min}).$$

Applying Markov's Inequality and by the definition of $\text{Bias}(\mathcal{B}, \mathbf{t})$, we obtain

$$\begin{aligned}\frac{|\mathcal{B}_{q_{\min}}|}{|\mathcal{B}|} &\leq \frac{\mathbb{E}_{\mathcal{U}[\mathcal{B}]}[\mathbf{t}^\top \mathbf{P}_{\phi,F}]}{q_{\min}} \\ &= \frac{p + \text{Bias}(\mathcal{B}, \mathbf{t})}{q_{\min}}.\end{aligned}$$

□

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 $\text{Bias}(\mathcal{D}, \mathbf{t}) = 0$, 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.

Proof. Let $\mathcal{F}$ be the space of possible information resources. Then,

$$\begin{aligned}\Pr(\omega \in t; \mathcal{A}) &= \int_{\mathcal{F}} \Pr(\omega \in t, f; \mathcal{A}) \mathrm{d}f \\ &= \int_{\mathcal{F}} \Pr(\omega \in t, f; \mathcal{A}) \Pr(f) \mathrm{d}f.\end{aligned}$$

Since we are considering the per-query probability of success for algorithm $\mathcal{A}$ on t using information resource f, we have

$$\Pr(\omega \in t \mid f : \mathcal{A}) = P_\phi(\omega \in t \mid f).$$

Also note that $\Pr(f) = \mathcal{D}(f)$ by the fact that $F \sim \mathcal{D}$. Making these substitutions, we obtain

$$\begin{aligned}\Pr(\omega \in T; \mathcal{A}) &= \int_{\mathcal{F}} \Pr(\omega \in t, f; \mathcal{A}) \mathcal{D}(f) \mathrm{d}f \\ &= \mathbb{E}_{\mathcal{D}}\left[P_\phi(\omega \in T \mid F)\right] \\ &= \mathbb{E}_{\mathcal{D}}\left[\mathbf{t}^\top \mathbf{P}_{\phi,F}\right] \\ &= \text{Bias}(\mathcal{D}, \mathbf{t}) + p \\ &= p.\end{aligned}$$

□

Theorem 8 (Famine of Applicable Targets)**.** *Let $\mathcal{D}$ be a distribution over a finite set of information resources. Define*

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

$$\tau_{q_{\min}} = \{T \mid T \in \tau_k, \text{Bias}(\mathcal{D}, \mathbf{t}) \geq q_{\min}\}$$

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

Proof. First, note that the size of τ_k is equivalent to the number of k-sized subsets of a $|\Omega|$-size set, $\binom{|\Omega|}{k}$. The size of $\tau_{q_{\min}}$ is the number of target sets in τ_k for which $\text{Bias}(\mathcal{D}, \mathbf{t}) \geq q_{\min}$. Let $F \sim \mathcal{D}$ and $T \sim \mathcal{U}[\tau_k]$. Then,

$$\begin{aligned}|\tau_{q_{\min}}| &= \sum_{\mathbf{t} \in \tau_k} \mathbb{1}_{\text{Bias}(\mathcal{D},\mathbf{t}) \geq q_{\min}} \\ &= \binom{|\Omega|}{k} \sum_{t \in \tau_k} \binom{|\Omega|}{k}^{-1} \mathbb{1}_{\text{Bias}(\mathcal{D},\mathbf{t}) \geq q_{\min}} \\ &= \binom{|\Omega|}{k} \mathbb{E}_{\mathcal{U}[\tau_k]}[\mathbb{1}_{\text{Bias}(\mathcal{D},T) \geq q_{\min}}] \\ &= \binom{|\Omega|}{k} \Pr(\text{Bias}(\mathcal{D}, T) \geq q_{\min}) \\ &= \binom{|\Omega|}{k} \Pr(p + \text{Bias}(\mathcal{D}, T) \geq p + q_{\min}) \\ &= \binom{|\Omega|}{k} \Pr(\mathbb{E}_{\mathcal{D}}[T^\top \overline{P}_F] \geq p + q_{\min}).\end{aligned}$$

Applying Markov's Inequality,

$$\begin{aligned}
|\tau_{q_{\min}}| &\leq \frac{\binom{|\Omega|}{k}\mathbb{E}_{\mathcal{U}[\tau_k]}[\mathbb{E}_{\mathcal{D}}[T^\top \overline{P}_F]]}{p+q_{\min}} \\
&= \frac{\binom{|\Omega|}{k}\sum_{\mathbf{t}\in\tau_k}\binom{|\Omega|}{k}^{-1}\mathbb{E}_{\mathcal{D}}[\mathbf{t}^\top \overline{P}_F]}{p+q_{\min}} \\
&= \frac{\mathbb{E}_{\mathcal{D}}[\overline{P}_F^\top \sum_{\mathbf{t}\in\tau_k}\mathbf{t}]}{p+q_{\min}} \\
&= \frac{\mathbb{E}_{\mathcal{D}}[\overline{P}_F^\top \mathbb{1}\binom{|\Omega|-1}{k-1}]}{p+q_{\min}} \\
&= \frac{\binom{|\Omega|-1}{k-1}\mathbb{E}_{\mathcal{D}}[\overline{P}_F^\top \mathbb{1}]}{p+q_{\min}} \\
&= \frac{\binom{|\Omega|-1}{k-1}}{p+q_{\min}}.
\end{aligned}$$

Thus,

$$\begin{aligned}
\frac{|\tau_{q_{\min}}|}{|\tau_k|} &\leq \frac{\binom{|\Omega|-1}{k-1}}{\binom{|\Omega|}{k}(p+q_{\min})} \\
&= \frac{\binom{|\Omega|-1}{k-1}}{\frac{|\Omega|}{k}\binom{|\Omega|-1}{k-1}(p+q_{\min})} \\
&= \frac{p}{p+q_{\min}} \\
&\leq \frac{p}{q_{\min}}.
\end{aligned}$$

□

Theorem 9 (Famine of Favorable Targets)**.** *For fixed $k \in \mathbb{N}$, fixed information resource F, and decomposable probability-of-success metric ϕ, define*

$$\tau_k = \{T \mid T \subseteq \Omega, |T| = k\}$$
$$\tau_{q_{\min}} = \{T \mid T \in \tau_k, \phi(T,F) \geq q_{\min}\}$$

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

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

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

Proof. Let $\mathcal{S} = \{\mathbf{s} : \mathbf{s} \in \{0,1\}^{|\Omega|}, \|\mathbf{s}\| = \sqrt{k}\}$. For brevity, we will allow $\mathbf{s}$ to also denote its corresponding target set, letting the context make clear whether the target set or target function is meant. Then,

$$\begin{aligned}
\frac{|\tau_{q_{\min}}|}{|\tau|} &= \frac{\sum_{\mathbf{s}\in\mathcal{S}}\mathbb{1}_{\phi(\mathbf{s},F)\geq q_{\min}}}{\binom{|\Omega|}{k}} \\
&= \binom{|\Omega|}{k}^{-1}\sum_{\mathbf{s}\in\mathcal{S}}\mathbb{1}_{\phi(\mathbf{s},F)\geq q_{\min}} \\
&= \mathbb{E}_{\mathcal{U}[\mathcal{S}]}\left[\mathbb{1}_{\phi(\mathbf{S},F)\geq q_{\min}}\right] \\
&= \Pr(\phi(\mathbf{S},F) \geq q_{\min}) \\
&\leq \frac{\mathbb{E}_{\mathcal{U}[\mathcal{S}]}[\phi(\mathbf{S},F)]}{q_{\min}},
\end{aligned}$$

where the final step follows from Markov's inequality. By decomposability of ϕ and linearity of expectation, we have

$$\begin{aligned}
\frac{\mathbb{E}_{\mathcal{U}[\mathcal{S}]}[\phi(\mathbf{S},F)]}{q_{\min}} &= \frac{\mathbb{E}_{\mathcal{U}[\mathcal{S}]}[\mathbf{S}^\top \mathbf{P}_{\phi,F}]}{q_{\min}} \\
&= \frac{\mathbf{P}_{\phi,F}^\top \mathbb{E}_{\mathcal{U}[\mathcal{S}]}[\mathbf{S}]}{q_{\min}} \\
&= \frac{\mathbf{P}_{\phi,f}^\top \mathbf{1}\left[\binom{|\Omega|}{k}^{-1}\binom{|\Omega|-1}{k-1}\right]}{q_{\min}} \\
&= \frac{k}{|\Omega|}\frac{\mathbf{P}_{\phi,f}^\top \mathbf{1}}{q_{\min}} \\
&= \frac{p}{q_{\min}}.
\end{aligned}$$

□

Theorem 10 (Bias-Expressivity Trade-off)**.** *Given a distribution over information resources $\mathcal{D}$ and a fixed target function $\mathbf{t} \in \{0,1\}^{|\Omega|}$, entropic expressivity is bounded above in terms of bias,*

$$H(\overline{P}_{\mathcal{D}}) \leq \log_2|\Omega| - 2\mathrm{Bias}(\mathcal{D},\mathbf{t})^2.$$

Additionally, bias is bounded above in terms of entropic expressivity,

$$\mathrm{Bias}(\mathcal{D},\mathbf{t}) \leq \sqrt{\frac{1}{2}(\log_2|\Omega| - H(\overline{P}_{\mathcal{D}}))} = \sqrt{\frac{1}{2}D_{KL}(\overline{P}_{\mathcal{D}}\|\mathcal{U})}.$$

Proof. Let $\omega \in t$ denote the measurable event that ω is an element of target set $t \subseteq \Omega$, and let Σ be the sigma algebra of measurable events. First, note that

$$\begin{aligned}
\mathrm{Bias}(\mathcal{D},t)^2 &= |\mathrm{Bias}(\mathcal{D},t)|^2 \\
&= |\mathbf{t}^\top \mathbb{E}_{\mathcal{D}}[\overline{P}_F] - p|^2 \\
&= |\mathbf{t}^\top \overline{P}_{\mathcal{D}} - p|^2 \\
&= |\overline{P}_{\mathcal{D}}(\omega \in t) - p|^2 \\
&\leq \frac{1}{2}D_{\mathrm{KL}}(\overline{P}_{\mathcal{D}} \,||\, \mathcal{U}) \\
&= \frac{1}{2}(H(\mathcal{U}) - H(\overline{P}_{\mathcal{D}})) \\
&= \frac{1}{2}(\log_2|\Omega| - H(\mathbb{E}_{\mathcal{D}}[\overline{P}_F]))
\end{aligned}$$

where the inequality is an application of Pinsker's Inequality. The quantity $D_{\mathrm{KL}}(\overline{P}_{\mathcal{D}} \;||\; \mathcal{U})$ is the Kullback-Leibler divergence between distributions $\overline{P}_{\mathcal{D}}$ and $\mathcal{U}$, which are distributions on search space Ω.

Thus,

$$H(\mathbb{E}_{\mathcal{D}}[\overline{P}_F]) \leq \log_2|\Omega| - 2\mathrm{Bias}(\mathcal{D},\mathbf{t})^2$$

and

$$\begin{aligned}
\mathrm{Bias}(\mathcal{D},t) &\leq \sqrt{\frac{1}{2}(\log_2|\Omega| - H(\overline{P}_{\mathcal{D}}))} \\
&= \sqrt{\frac{1}{2}D_{\mathrm{KL}}(\overline{P}_{\mathcal{D}} \,||\, \mathcal{U})} \\
&= \sqrt{\frac{1}{2}(\log_2|\Omega| - H(\mathbb{E}_{\mathcal{D}}[\overline{P}_F]))}.
\end{aligned}$$

□

Corollary 1 (Bias Bound Under Expected Expressivity)**.**

$$\begin{aligned}\text{Bias}(\mathcal{D}, \mathbf{t}) &\leq \sqrt{\frac{1}{2}(\log_2 |\Omega| - \mathbb{E}_{\mathcal{D}}[H(\overline{P}_F)])} \\ &= \sqrt{\mathbb{E}_{\mathcal{D}}\left[\frac{1}{2} D_{KL}(\overline{P}_F \| \mathcal{U})\right]}.\end{aligned}$$

Proof. By the concavity of the entropy function and Jensen's Inequality, we obtain

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

Thus, an upper bound of bias is

$$\begin{aligned}\text{Bias}(\mathcal{D}, t) &\leq \sqrt{\frac{1}{2} D_{KL}(\overline{P} \,\|\, \mathcal{U})} \\ &= \sqrt{\frac{1}{2}(\log_2 |\Omega| - H(\mathbb{E}_{\mathcal{D}}[\overline{P}_F]))} \\ &\leq \sqrt{\frac{1}{2}(\log_2 |\Omega| - \mathbb{E}_{\mathcal{D}}[H([\overline{P}_F])])} \\ &= \sqrt{\mathbb{E}_{\mathcal{D}}\left[\frac{1}{2} D_{KL}(\overline{P}_F \,\|\, \mathcal{U})\right]},\end{aligned}$$

where the final equality follows from the linearity of expectation and the definition of KL-divergence. □

Theorem 11 (Expressivity Bounded By Bias)**.** *Given a fixed k-hot target function* $\mathbf{t}$ *and a distribution over information resources* $\mathcal{D}$*, the entropic expressivity of a search algorithm can be bounded in terms of* $\epsilon := \text{Bias}(\mathcal{D}, \mathbf{t})$*, by*

$$\begin{aligned}H(\overline{P}_{\mathcal{D}}) \in \Bigg[& H(p+\epsilon), \Bigg((p+\epsilon) \log_2 \left(\frac{k}{p+\epsilon} \right) + \\ & (1-(p+\epsilon)) \log_2 \left(\frac{|\Omega| - k}{1-(p+\epsilon)} \right) \Bigg) \Bigg].\end{aligned}$$

Proof. Following Definition 7, the expressivity of a search algorithm varies solely with respect to $D_{\text{KL}}(\overline{P}_{\mathcal{D}} \,||\, \mathcal{U})$ since we always consider the same search space and thus $H(\mathcal{U})$ is a constant value. We obtain a lower bound of the expressivity by maximizing the value of $D_{\text{KL}}(\overline{P}_{\mathcal{D}} \,||\, \mathcal{U})$ and an upper bound by minimizing this term.

First, we show that $H(p+\epsilon)$ is a lower bound of expressivity by constructing a distribution that deviates the most from a uniform distribution over Ω. By the definition of $\text{Bias}(\mathcal{D}, \mathbf{t})$, we place $(p+\epsilon)$ probability mass on the target set t and $1-(p+\epsilon)$ probability mass on the remaining $(n-k)$ elements of Ω. We distribute the probability mass such that all of the $(p+\epsilon)$ probability mass of the target set is concentrated on a single element and all of the $1-(p+\epsilon)$ probability mass of the complement of the target set is concentrated on a single element. In this constructed distribution where $D_{\text{KL}}(\overline{P}_{\mathcal{D}} \,||\, \mathcal{U})$ is maximized, the value of expressivity is

$$\begin{aligned}H(\overline{P}_{\mathcal{D}}) =& -\sum_{\omega \in \Omega} \overline{P}_{\mathcal{D}}(\omega) \log_2 \overline{P}_{\mathcal{D}}(\omega) \\ =& -(p+\epsilon) \log_2 (p+\epsilon) \\ & -(1-(p+\epsilon)) \log_2 (1-(p+\epsilon)) \\ =& H(p+\epsilon)\end{aligned}$$

where the $H(p+\epsilon)$ is the entropy of a Bernoulli distribution with parameter $(p+\epsilon)$. The entropy of this constructed distribution gives a lower bound on expressivity,

$$H(\overline{P}_{\mathcal{D}}) \geq H(p+\epsilon).$$

Now, we show that

$$(p+\epsilon) \log_2 \left(\frac{k}{p+\epsilon} \right) + (1-(p+\epsilon)) \log_2 \left(\frac{|\Omega| - k}{1-(p+\epsilon)} \right)$$

is an upper bound of expressivity by constructing a distribution that deviates the least from a uniform distribution over Ω. In this case, we uniformly distribute $\frac{1}{|\Omega|}$ probability mass over the entire search space, Ω. Then, to account for the ϵ level of bias, we add $\frac{\epsilon}{k}$ probability mass to elements of the target set and we remove $\frac{\epsilon}{n-k}$ probability mass to elements of the complement of the target set. In this constructed distribution where $D_{\text{KL}}(\overline{P}_{\mathcal{D}} \,||\, \mathcal{U})$ is minimized, the value of expressivity is

$$\begin{aligned}H(\overline{P}_{\mathcal{D}}) =& -\sum_{\omega \in \Omega} \overline{P}_{\mathcal{D}}(\omega) \log_2 \overline{P}_{\mathcal{D}}(\omega) \\ =& -\sum_{\omega \in t} \left(\frac{1}{|\Omega|} + \frac{\epsilon}{k} \right) \log_2 \left(\frac{1}{|\Omega|} + \frac{\epsilon}{k} \right) \\ & -\sum_{\omega \in t^c} \left(\frac{1}{|\Omega|} - \frac{\epsilon}{|\Omega| - k} \right) \log_2 \left(\frac{1}{|\Omega|} - \frac{\epsilon}{|\Omega| - k} \right) \\ =& -\sum_{\omega \in t} \left(\frac{p+\epsilon}{k} \right) \log_2 \left(\frac{p+\epsilon}{k} \right) \\ & -\sum_{\omega \in t^c} \left(\frac{1-(p+\epsilon)}{|\Omega| - k} \right) \log_2 \left(\frac{1-(p+\epsilon)}{|\Omega| - k} \right) \\ =& -k \left(\frac{p+\epsilon}{k} \right) \log_2 \left(\frac{p+\epsilon}{k} \right) \\ & -(|\Omega| - k) \left(\frac{1-(p+\epsilon)}{|\Omega| - k} \right) \log_2 \left(\frac{1-(p+\epsilon)}{|\Omega| - k} \right) \\ =& (p+\epsilon) \log_2 \left(\frac{k}{p+\epsilon} \right) \\ & + (1-(p+\epsilon)) \log_2 \left(\frac{|\Omega| - k}{1-(p+\epsilon)} \right).\end{aligned}$$

The entropy on this constructed distribution gives an upper bound on expressivity,

$$\begin{aligned}H(\overline{P}_{\mathcal{D}}) \leq & (p+\epsilon) \log_2 \left(\frac{k}{p+\epsilon} \right) \\ & + (1-(p+\epsilon)) \log_2 \left(\frac{|\Omega| - k}{1-(p+\epsilon)} \right).\end{aligned}$$

These two bounds give us a range of possible values of expressivity given a fixed level of bias, namely

$$H(\overline{P}_{\mathcal{D}}) \in \Bigg[H(p+\epsilon), \Bigg((p+\epsilon)\log_2\left(\frac{k}{p+\epsilon}\right) + (1-(p+\epsilon))\log_2\left(\frac{|\Omega|-k}{1-(p+\epsilon)}\right)\Bigg)\Bigg].$$

□

Lemma 3 (Expected Per Query Performance From Expected Distribution)**.** *This lemma has been proven by Montanez [1] and is directly drawn from [1]. Let T be a target set, $q(T,F)$ be the expected per-query probability of success for an algorithm, and ν be the conditional joint measure induced by that algorithm over finite sequences of probability distributions and search histories, conditioned on external information resource F. Denote a probability distribution sequence by $\tilde{P}$ and a search history by h. Let $\mathcal{U}[\tilde{P}]$ denote a uniform distribution on elements of $\tilde{P}$ and define $\overline{P}(x|F) = \int \mathbb{E}_{P\sim\mathcal{U}[\tilde{P}]}[P(x)]d\nu(\tilde{P},h|F)$. Then,*

$$q(T,F) = \overline{P}(X \in T|F)$$

Lemma 4. *(Maximum Number of Satisfying Vectors) Given an integer $1 \leq k \leq n$, a set $\mathcal{S} = \{\mathbf{s} : \mathbf{s} \in \{0,1\}^n, \|\mathbf{s}\| = \sqrt{k}\}$ of all n-length k-hot binary vectors, a set $\mathcal{P} = \{P : P \in \mathbb{R}^n, \sum_j P_j = 1\}$ of discrete n-dimensional simplex vectors, and a fixed scalar threshold $\epsilon \in [0,1]$, then for any fixed $P \in \mathcal{P}$,*

$$\sum_{\mathbf{s}\in\mathcal{S}} \mathbb{1}_{\mathbf{s}^\top P \geq \epsilon} \leq \frac{1}{\epsilon}\binom{n-1}{k-1}$$

where $\mathbf{s}^\top P$ denotes the vector dot product between $\mathbf{s}$ and P.

Proof. For $\epsilon = 0$, the bound holds trivially. For $\epsilon > 0$, let S be a random quantity that takes values $\mathbf{s}$ uniformly in the set $\mathcal{S}$. Then, for any fixed $P \in \mathcal{P}$,

$$\begin{aligned}\sum_{\mathbf{s}\in\mathcal{S}} \mathbb{1}_{\mathbf{s}^\top P \geq \epsilon} &= \binom{n}{k}\mathbb{E}\left[\mathbb{1}_{S^\top P \geq \epsilon}\right] \\ &= \binom{n}{k}\Pr\left(S^\top P \geq \epsilon\right).\end{aligned}$$

Let $\mathbb{1}$ denotes the all ones vector. Under a uniform distribution on random quantity S and because P does not change with respect to $\mathbf{s}$, we have

$$\begin{aligned}\mathbb{E}\left[S^\top P\right] &= \binom{n}{k}^{-1}\sum_{\mathbf{s}\in\mathcal{S}} \mathbf{s}^\top P \\ &= P^\top \binom{n}{k}^{-1}\sum_{\mathbf{s}\in\mathcal{S}} \mathbf{s} \\ &= P^\top \frac{\mathbb{1}\binom{n-1}{k-1}}{\binom{n}{k}} \\ &= P^\top \frac{\mathbb{1}\binom{n-1}{k-1}}{\frac{n}{k}\binom{n-1}{k-1}} \\ &= \frac{k}{n}P^\top\mathbb{1} \\ &= \frac{k}{n}\end{aligned}$$

since P must sum to 1.

Noting that $S^\top P \geq 0$, we use Markov's inequality to get

$$\begin{aligned}\sum_{\mathbf{s}\in\mathcal{S}} \mathbb{1}_{\mathbf{s}^\top P \geq \epsilon} &= \binom{n}{k}\Pr\left(S^\top P \geq \epsilon\right) \\ &\leq \binom{n}{k}\frac{1}{\epsilon}\mathbb{E}\left[S^\top P\right] \\ &= \binom{n}{k}\frac{1}{\epsilon}\frac{k}{n} \\ &= \frac{1}{\epsilon}\binom{n-1}{k-1}.\end{aligned}$$

□

Theorem 12 (Famine of Forte)**.** *Define*

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

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

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

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 expected per-query probability of success for algorithm $\mathcal{A}$ on problem (Ω,T,F). Then for any $m \in \mathbb{N}$,

$$\frac{|R_{q_{\min}}|}{|R|} \leq \frac{p}{q_{\min}} \qquad \textit{and} \qquad \lim_{m\to\infty}\frac{|R_{q_{\min}}|}{|R|} \leq \frac{p}{q_{\min}}$$

where $p = k/|\Omega|$.

Proof. We begin by defining a set $\mathcal{S}$ of all $|\Omega|$-length target functions with exactly k ones, namely, $\mathcal{S} = \{\mathbf{s} : \mathbf{s} \in \{0,1\}^{|\Omega|}, \|\mathbf{s}\| = \sqrt{k}\}$. For each of these, we have $|\mathcal{B}_m|$ external information resources. The total number of search problems is therefore

$$\binom{|\Omega|}{k}|\mathcal{B}_m|.$$

We seek to bound the proportion of possible search problems for which $\phi(\mathbf{s}, F) \geq q_{\min}$ for any threshold $q_{\min} \in (0, 1]$. Thus,

$$\begin{aligned}\frac{|R_{q_{\min}}|}{|R|} &\leq \frac{|\mathcal{B}_m| \sup_F \left[\sum_{\mathbf{s}\in\mathcal{S}} \mathbb{1}_{\phi(\mathbf{s},F)\geq q_{\min}}\right]}{|\mathcal{B}_m|\binom{|\Omega|}{k}} \\ &= \binom{|\Omega|}{k}^{-1} \sum_{\mathbf{s}\in\mathcal{S}} \mathbb{1}_{\phi(\mathbf{s},F^*)\geq q_{\min}},\end{aligned}$$

where $F^* \in \mathcal{B}_m$ denotes the arg sup of the expression. Therefore,

$$\begin{aligned}\frac{|R_{q_{\min}}|}{|R|} &\leq \binom{|\Omega|}{k}^{-1} \sum_{\mathbf{s}\in\mathcal{S}} \mathbb{1}_{\phi(\mathbf{s},F^*)\geq q_{\min}} \\ &= \binom{|\Omega|}{k}^{-1} \sum_{\mathbf{s}\in\mathcal{S}} \mathbb{1}_{\mathbf{s}^\top \mathbf{P}_{\phi,F^*} \geq q_{\min}}\end{aligned}$$

where the equality follows decomposability of $\phi(\mathbf{s}, F^*)$ and $\mathbf{P}_{\phi,F^*}$ represents the $|\Omega|$-length probability vector defined by $P_\phi(\cdot|F^*)$. By Lemma 4, we have

$$\begin{aligned}\binom{|\Omega|}{k}^{-1} \sum_{\mathbf{s}\in\mathcal{S}} \mathbb{1}_{\mathbf{s}^\top \overline{P}_{F^*} \geq q_{\min}} &\leq \binom{|\Omega|}{k}^{-1} \left[\frac{1}{q_{\min}} \binom{|\Omega|-1}{k-1}\right] \\ &= \frac{k}{|\Omega|}\frac{1}{q_{\min}} \\ &= p/q_{\min}\end{aligned}$$

proving the result for finite external information resources. To extend to infinite external information resources, let $A_m = \{f : f \in \{0,1\}^\ell, \ell \in \mathbb{N}, \ell \leq m\}$ and define

$$\begin{aligned}a_m &:= \frac{|A_m| \sup_{F\in A_m} \left[\sum_{\mathbf{s}\in\mathcal{S}} \mathbb{1}_{\phi(\mathbf{s},F)\geq q_{\min}}\right]}{|A_m|\binom{|\Omega|}{k}}, \\ b_m &:= \frac{|\mathcal{B}_m| \sup_{F\in \mathcal{B}_m} \left[\sum_{\mathbf{s}\in\mathcal{S}} \mathbb{1}_{\phi(\mathbf{s},F)\geq q_{\min}}\right]}{|\mathcal{B}_m|\binom{|\Omega|}{k}}.\end{aligned}$$

We have shown that $a_m \leq p/q_{\min}$ for each $m \in \mathbb{N}$. Thus,

$$\begin{aligned}\limsup_{m\to\infty} \frac{|A_m| \sup_{F\in A_m} \left[\sum_{\mathbf{s}\in\mathcal{S}} \mathbb{1}_{\phi(\mathbf{s},F)\geq q_{\min}}\right]}{|A_m|\binom{|\Omega|}{k}} &= \limsup_{m\to\infty} a_m \\ &\leq \sup_m a_m \\ &\leq p/q_{\min}.\end{aligned}$$

Next, we use the monotone convergence theorem to show the limit exists. First,

$$\lim_{m\to\infty} a_m = \lim_{m\to\infty} \frac{\sup_{F\in A_m} \left[\sum_{\mathbf{s}} \mathbb{1}_{\phi(\mathbf{s},F)\geq q_{\min}}\right]}{\binom{|\Omega|}{k}}$$

By construction, the successive A_m are nested with increasing m, so the sequence of suprema (and numerator) are increasing, though not necessarily strictly increasing. The denominator is not dependent on m, so $\{a_m\}$ is an increasing sequence. Because it is also bounded above by $p/q_{\min}$, the limit exists by monotone convergence. Thus,

$$\lim_{m\to\infty} a_m = \limsup_{m\to\infty} a_m \leq p/q_{\min}.$$

Lastly,

$$\begin{aligned}\lim_{m\to\infty} b_m &= \lim_{m\to\infty} \frac{|\mathcal{B}_m| \sup_{F\in \mathcal{B}_m} \left[\sum_{\mathbf{s}\in\mathcal{S}} \mathbb{1}_{\phi(\mathbf{s},F)\geq q_{\min}}\right]}{|\mathcal{B}_m|\binom{|\Omega|}{k}} \\ &= \lim_{m\to\infty} \frac{\sup_{F\in \mathcal{B}_m} \left[\sum_{\mathbf{s}\in\mathcal{S}} \mathbb{1}_{\phi(\mathbf{s},F)\geq q_{\min}}\right]}{\binom{|\Omega|}{k}} \\ &\leq \lim_{m\to\infty} \frac{\sup_{F\in A_m} \left[\sum_{\mathbf{s}\in\mathcal{S}} \mathbb{1}_{\phi(\mathbf{s},F)\geq q_{\min}}\right]}{\binom{|\Omega|}{k}} \\ &= \lim_{m\to\infty} a_m \\ &\leq p/q_{\min}.\end{aligned}$$

□

Corollary 2 (The Conservation of Active Information of Expectations)**.** *Define $I_{\phi(T,F)} = -\log_2 p/\phi(T,F)$ to be a version of active information of expectations for decomposable metrics and let*

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

and

$$R_b = \{(T, F) \mid T \in \tau_k, F \in \mathcal{B}_m, I_{\phi(T,F)} \geq b\}.$$

Then for any $m \in \mathbb{N}$

$$\frac{|R_b|}{|R|} \leq 2^{-b}.$$

Proof. The proof follows from the definition of active information of expectations and Theorem 12. Note,

$$b \leq -\log_2\left(\frac{p}{\phi(T,F)}\right)$$

implies

$$\phi(T, F) \geq p2^b.$$

Since $I_{\phi(T,F)} \geq b$ implies $\phi(T, F) \geq p2^b$, the set of problems for which $I_{\phi(T,F)} \geq b$ can be no bigger than the set for which $\phi(T, F) \geq p2^b$. By Theorem 12, the proportion of problems for which $\phi(T, F)$ is at least $p2^b$ is no greater than $p/(p2^b)$. Thus,

$$\frac{|R_b|}{|R|} \leq \frac{1}{2^b}.$$

□

Lemma 5 (Maximum Proportion of Satisfying Strategies)**.** *Given an integer $1 \leq k \leq n$, a set $\mathcal{S} = \{\mathbf{s} : \mathbf{s} \in \{0,1\}^n, \|\mathbf{s}\| = \sqrt{k}\}$ of all n-length k-hot binary vectors, a set $\mathcal{P} = \{P : P \in \mathbb{R}^n, \sum_j P_j = 1\}$ of discrete n-dimensional simplex vectors, and a fixed scalar threshold $\epsilon \in [0, 1]$, then*

$$\max_{\mathbf{s}\in\mathcal{S}} \frac{\mu(\mathcal{G}_{\mathbf{s},\epsilon})}{\mu(\mathcal{P})} \leq \frac{1}{\epsilon}\frac{k}{n}$$

where $\mathcal{G}_{\mathbf{s},\epsilon} = \{P : P \in \mathcal{P}, \mathbf{s}^\top P \geq \epsilon\}$ and μ is Lebesgue measure.

Proof. For $\epsilon = 0$, the bound holds trivially. For $\epsilon > 0$, we first notice that the $\mu(\mathcal{P})^{-1}$ term can be viewed as a uniform

density over the region of the simplex $\mathcal{P}$, so that the integral becomes an expectation with respect to this distribution, where P is drawn uniformly from $\mathcal{P}$. Thus, for any $\mathbf{s} \in \mathcal{S}$,

$$\begin{aligned}\frac{\mu(\mathcal{G}_{\mathbf{s},\epsilon})}{\mu(\mathcal{P})} &= \int_{\mathcal{P}} \frac{1}{\mu(\mathcal{P})} \left[\mathbb{1}_{\mathbf{s}^\top P \geq \epsilon}\right] d\mu(P) \\ &= \mathbb{E}_{P\sim\mathcal{U}(\mathcal{P})}[\mathbb{1}_{\mathbf{s}^\top P \geq \epsilon}] \\ &= \Pr(\mathbf{s}^\top P \geq \epsilon) \\ &\leq \frac{1}{\epsilon}\mathbb{E}_{P\sim\mathcal{U}(\mathcal{P})}\left[\mathbf{s}^\top P\right],\end{aligned}$$

where the final line follows from Markov's inequality. Since the symmetric Dirichlet distribution in n dimensions with $\alpha = 1$ gives the uniform distribution over the simplex, we get

$$\begin{aligned}\mathbb{E}_{P\sim\mathcal{U}(\mathcal{P})}[P] &= \mathbb{E}_{P\sim\text{Dir}(\alpha=1)}[P] \\ &= \left(\frac{\alpha}{\sum_{i=1}^n \alpha}\right)\mathbf{1} \\ &= \left(\frac{1}{n}\right)\mathbf{1},\end{aligned}$$

where $\mathbf{1}$ denotes the all ones vector. We have

$$\begin{aligned}\frac{1}{\epsilon}\mathbb{E}_{P\sim\mathcal{U}(\mathcal{P})}\left[\mathbf{s}^\top P\right] &= \frac{1}{\epsilon}\mathbf{s}^\top\mathbb{E}_{P\sim\mathcal{U}(\mathcal{P})}[P] \\ &= \frac{1}{\epsilon}\mathbf{s}^\top\left(\frac{1}{n}\right)\mathbf{1} \\ &= \frac{1}{\epsilon}\frac{k}{n}.\end{aligned}$$

□

Theorem 13 (Famine of Favorable Strategies)**.** *For any fixed search problem (Ω, t, f), set of probability mass functions $\mathcal{P} = \{P : P \in \mathbb{R}^{|\Omega|}, \sum_j P_j = 1\}$, and a fixed threshold $q_{min} \in [0,1]$,*

$$\frac{\mu(\mathcal{G}_{t,q_{min}})}{\mu(\mathcal{P})} \leq \frac{p}{q_{min}},$$

where $\mathcal{G}_{t,q_{min}} = \{P : P \in \mathcal{P}, t^\top P \geq q_{min}\}$ and μ is Lebesgue measure. Furthermore, the proportion of possible search strategies giving at least b bits of active information of expectations is no greater than 2^{-b}.

Proof. Applying Lemma 5, with $\mathbf{s} = t$, $\epsilon = q_{\min}, k = |t|, n = |\Omega|$, and $p = \frac{|t|}{|\Omega|}$, yields the first result, while following the same steps as Corollary 2 gives the second (noting that by Lemma 5 each strategy is equivalent to a corresponding $\phi(t, f)$). □

Theorem 14 (Distributional Capacity Upper Bound)**.**

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

where $\overline{\mathbf{P}}_F$ is the expected average conditional distribution on the search space given F.

Proof. Combining the Distributional Capacity as Entropic Expressivity (Theorem 3 from [10]) and Theorem 10 yields

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

□

Lemma 6. *If $I(F_{R_0+L}; T_R) \leq I(F_{R_0}; T_R)$ then*

$$I(F_{R_0+L}; T_R) \leq I(F_S; T_R) + I(F_{R_0}; T_R).$$

Proof.

$$\begin{aligned}I(F_{R_0+L}; T_R) &\leq I(F_{R_0}; T_R) \\ &= I(L; T_R \,|\, F_{R_0}) + I(F_{R_0}; T_R) \\ &= H(L \,|\, F_{R_0}) - H(L \,|\, F_{R_0}, T_R) + I(F_{R_0}; T_R) \\ &= H(L \,|\, F_{R_0}) - H(L \,|\, T_R) + I(F_{R_0}; T_R) \\ &\leq H(L) - H(L \,|\, T_R) + I(F_{R_0}; T_R) \\ &= I(L; T_R) + I(F_{R_0}; T_R) \\ &\leq I(F_S; T_R) + I(F_{R_0}; T_R)\end{aligned}$$

where the first equality follows from application of the chain rule for mutual information, the second and fourth equalities follow from the definition of mutual information, the third equality follows from the conditional independence assumption, and the final equality follows by application of the Data Processing Inequality [14]. □

Theorem 15 (Transfer Learning under Dependence)**.** *Define*

$$\phi_{TL} \coloneqq \mathbb{E}_{T_R, F_{R+L}}\left[\phi(T_R, F_{R+L})\right] = \textit{Pr}(\omega \in T_R; \mathcal{A})$$

as the probability of success for transfer learning. Then,

$$\phi_{TL} \leq \frac{I(F_S; T_R) + I(F_R; T_R) + D(P_{T_R} \| \mathcal{U}_{T_R}) + 1}{I_\Omega}$$

where $I_\Omega = -\log|T_R|/|\Omega|$, ($T_R$ being of fixed size), $D(P_{T_R}\|\mathcal{U}_{T_R})$ is the Kullback-Leibler divergence between the marginal distribution on T_R and the uniform distribution on T_R, and $I(F;T)$ is the mutual information.

Proof. By d-separation of the graphical model structure in Fig. 2 and the Data Processing Inequality [14], we have that $I(F_{R_o+L}; T_R) \leq I(F_{R_o}, L; T_R)$. Applying the result from Lemma 6 to the Theorem 4 (Learning Under Dependence), we obtain

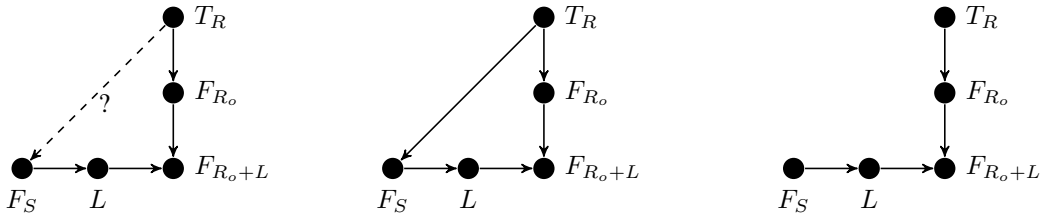
$$\begin{aligned}\phi_{TL} &\leq \frac{I(F_{R+L}; T_R) + D(P_{T_R}\|\mathcal{U}_{T_R}) + 1}{I_\Omega} \\ &\leq \frac{I(F_S; T_R) + I(F_R; T_R) + D(P_{T_R}\|\mathcal{U}_{T_R}) + 1}{I_\Omega}.\end{aligned}$$

□

Theorem 16 (Success Difference from Distribution Divergence)**.** *Given the performance of a search algorithm on the recipient problem in the transfer learning case, ϕ_{TL}, and without the learning resource, ϕ_{NoTL}, we can upperbound the absolute difference as*

$$|\phi_{TL} - \phi_{NoTL}| \leq |T|\sqrt{\frac{1}{2}D_{KL}(\mathbf{P}_{TL}\|\mathbf{P}_{NoTL})},$$

where $\mathbf{P}_{TL}$ and $\mathbf{P}_{NoTL}$ are probability distributions on the search space induced during the search by the algorithm with and without transfer learning, respectively.



(a) General Case (b) Case 1 (c) Case 2

Fig. 2: Dependence Structure for Transfer Learning.

Proof.

$$\begin{aligned}
|\phi_{TL} - \phi_{NoTL}| &= |\mathbf{t}^\top(\mathbf{P}_{TL} - \mathbf{P}_{NoTL})| \\
&= |\sum_{\omega} \mathbb{1}_{\omega \in T}(\mathbf{P}_{TL}(\omega) - \mathbf{P}_{NoTL}(\omega))| \\
&\leq |T| \sup_{w \in T} |\mathbf{P}_{TL}(\omega) - \mathbf{P}_{NoTL}(\omega)| \\
&\leq |T| \sqrt{\frac{1}{2} D_{KL}(\mathbf{P}_{TL} || \mathbf{P}_{NoTL})}
\end{aligned}$$

where the first equality follows form the definition of decomposable probability of success metrics and the final inequality follows by application of Pinsker's Inequality. □