Model Characterization with Inductive Orientation Vectors

Kerria Pang-Naylor[1] Eric Chen[1,2] George D. Montañez[1]

[1]AMISTAD Lab
Department of Computer Science
Harvey Mudd College
Claremont, CA, United States

[2]Department of Computer Science
Stanford University
Stanford, CA, United States

Motivation and Overview

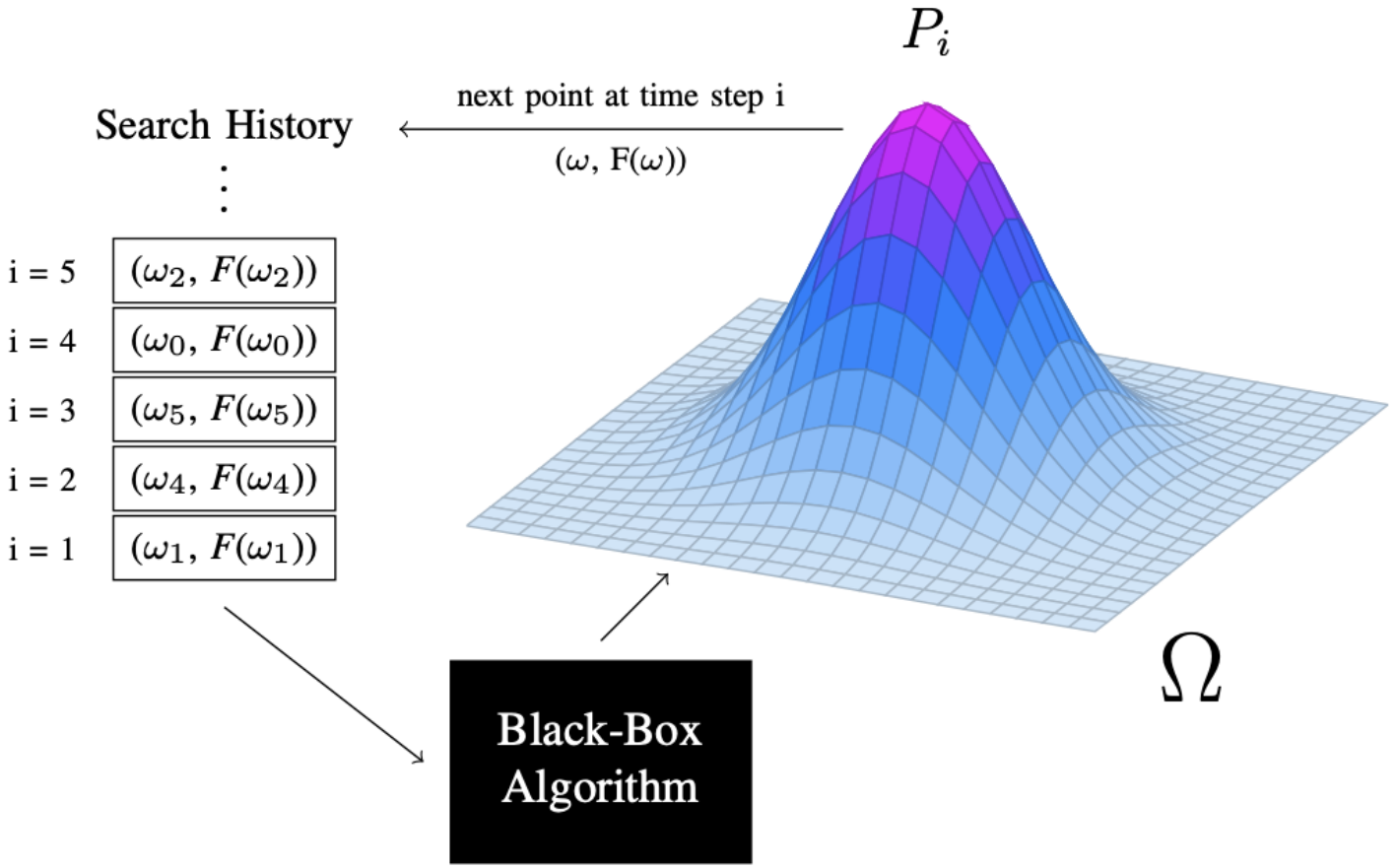
- Increasingly complex models $\rightarrow$ black-box evaluation techniques
- The **inductive orientation vector** allows us to calculate
  - Algorithmic Bias
  - Entropic Expressivity
  - Algorithmic Capacity
- Grounded in the **Algorithmic Search Framework** (ASF)
- We will focus on **classification models**

Overview

Algorithmic Search Framework (ASF)

(Montañez, 2017)

- Ω – search space
- $T \subseteq \Omega$ – target set
- F – external information resource
- $\mathcal{A}$ – search algorithm



ASF for Classification Models

Let's say we want to classify n datapoints with c categories.

- Ω: All possible c^n labelings
  - e.g., $c = 2$ and $n = 5 \rightarrow$
    $\Omega = \{(0,0,0,0,0), (0,0,0,0,1) \cdots (1,0,1,1,0) \cdots\}$
- $T \subset \Omega$: "Accurate enough" labelings
- F: training data + loss function
- $\mathcal{A}$: algorithm choice + optimization strategy (e.g., logistic regression + SGD)

The Inductive Orientation Vector

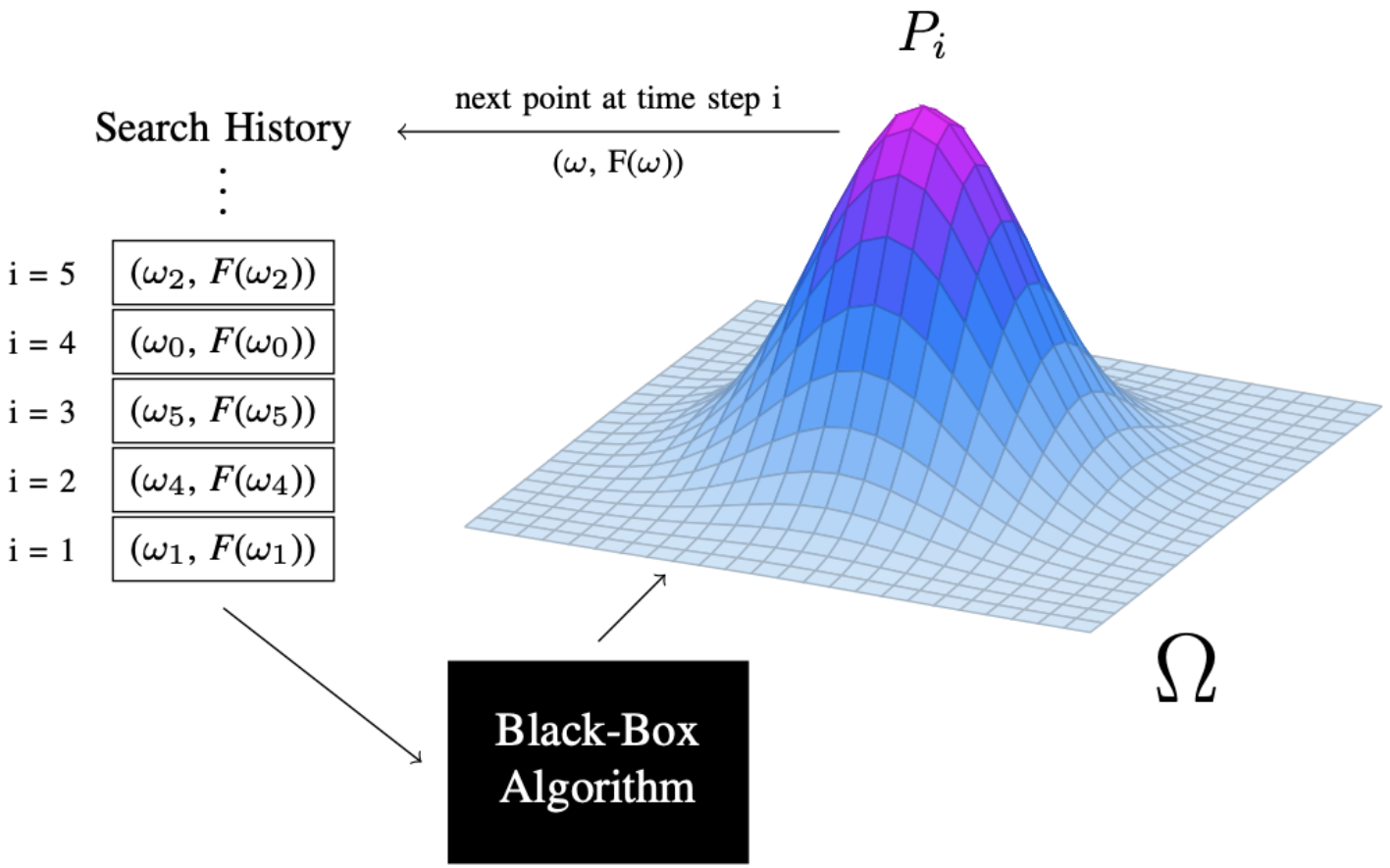
(Bekerman et. al., 2022)

- The **inductive orientation vector** ($\overline{P}_{\mathcal{D}}$) of an algorithm is a combination of its **inductive bias** and **training data**.
- $\mathcal{D}$ represents a data-generating distribution of F

Definition (Inductive orientation, $\overline{P}_{\mathcal{D}}$)

An algorithm's **inductive orientation** is its expected distribution over the search space Ω for $F \sim \mathcal{D}$.

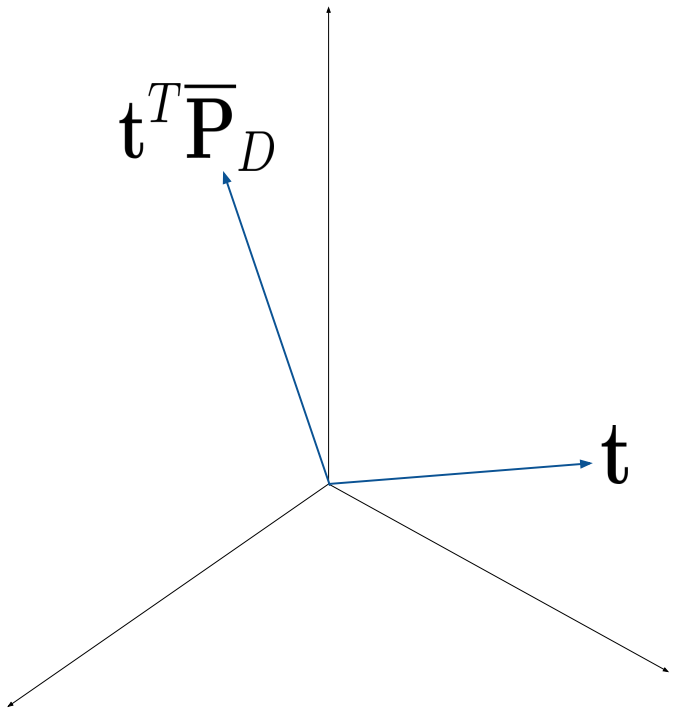
The Inductive Orientation Vector (cont.)



The Inductive Orientation Vector (cont.)

(Bekerman et. al., 2022)

- We can represent both T and $\overline{P}_{\mathcal{D}}$ as $|\Omega|$-length vectors
  - T becomes $|T|$-hot encoding vector $\mathbf{t}$
- The **alignment** ($\mathbf{t}^T\overline{\mathbf{P}}_{\mathcal{D}}$) of these vectors represents the search algorithm's expected per query **probability of success**



Algorithmic Bias

(Montañez, 2017)

Algorithmic Bias measures how successful the algorithm is compared to uniform random sampling.

Definition (Bias($\mathcal{D}, \mathbf{t}$))

Let p be the per-query probability of success under uniform random sampling. Then, for target $\mathbf{t}$,

$$\text{Bias}(\mathcal{D}, \mathbf{t}) = \mathbf{t}^T \overline{\mathbf{P}}_{\mathcal{D}} - p.$$

Entropic Expressivity

(Montañez, 2017)

Entropic Expressivity measures how responsive an algorithm *can be* to changes in training data and stochasticity.

- measures how uniform the expected $\overline{\mathbf{P}}_{\mathcal{D}}$ distribution is.
- Can be calculated with **Shannon Entropy** $H(\overline{\mathbf{P}}_{\mathcal{D}})$ or **KL divergence** $D_{\mathbf{KL}}(\overline{\mathbf{P}}_{\mathcal{D}}||\mathcal{U})$

Definition (Entropic Expressivity, $H(\overline{\mathbf{P}}_{\mathcal{D}})$)

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

Algorithmic Capacity
(Bashir et. al., 2020)

Algorithmic Capacity isolates the part of entropic expressivity responding to *just* changes in training data

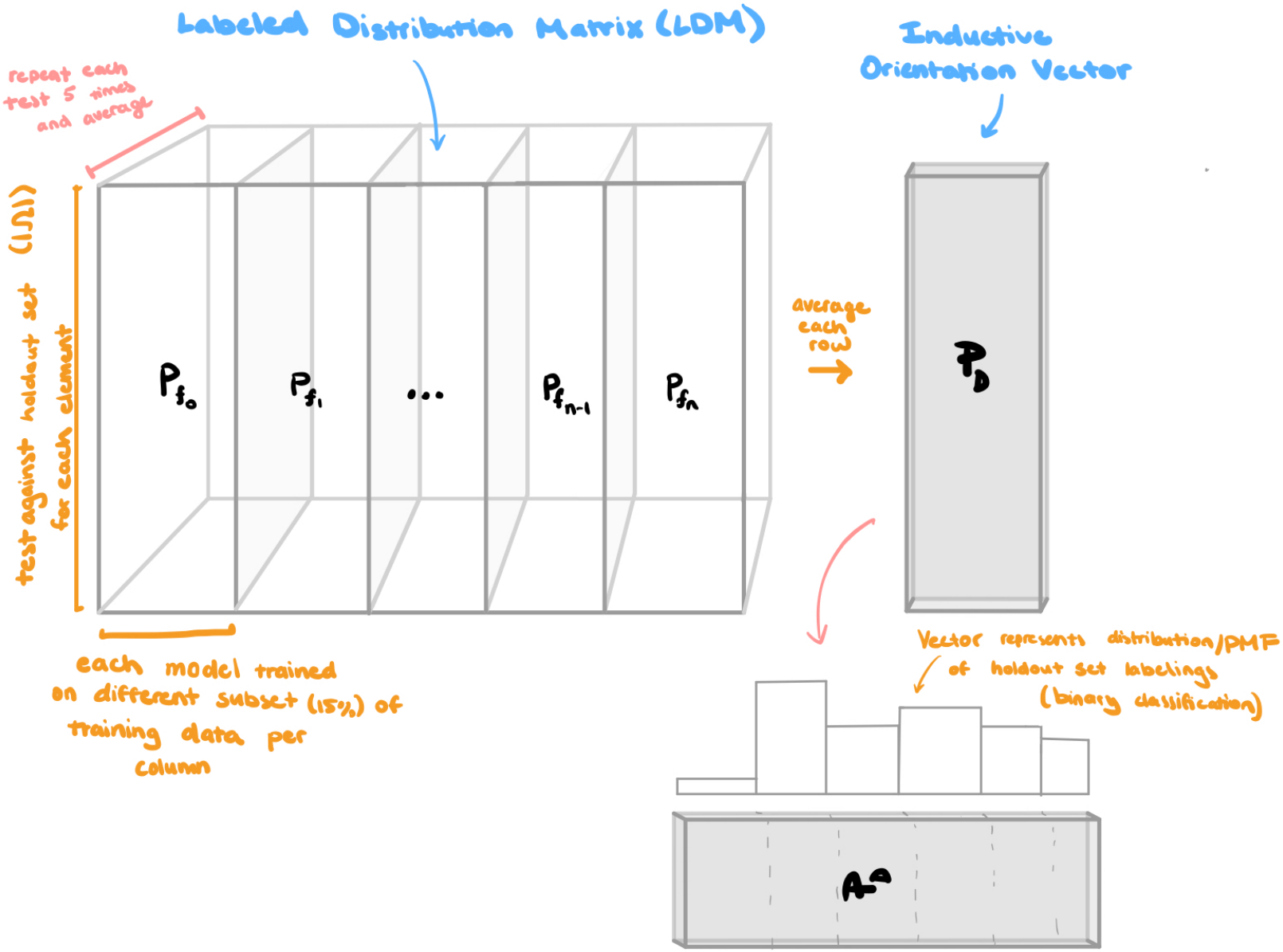
- "Remove" entropy within individual training sets $\overline{\mathbf{P}}_F$

Definition (Algorithmic Capacity, $C_{\mathcal{A},\mathcal{D}}$)

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

Experimental Implementation – LDM

(Segura et. al., 2020)

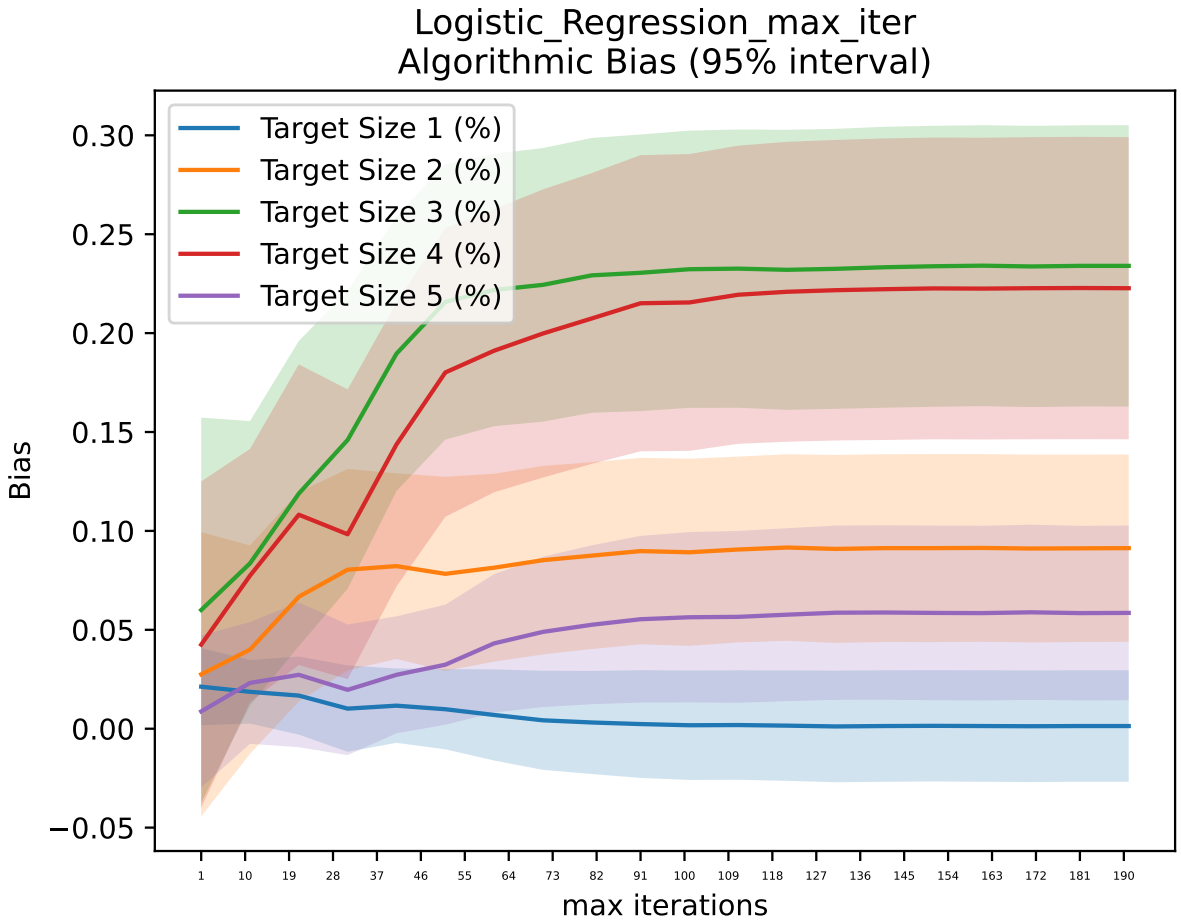


Experimental Implementation (cont.)

- 80:20 train/test ratio to train binary classification algorithms
- Tested bias, expressivity, and capacity over range of **hyperparameters** for each algorithm
- Holdout set of size 5, inductive orientation vectors have size 2^5
- Ran metrics with 100 repetitions on 10 datasets and 9 different algorithm-hyperparameter combinations
- Focused on classic, **interpretable** algorithms for grounding

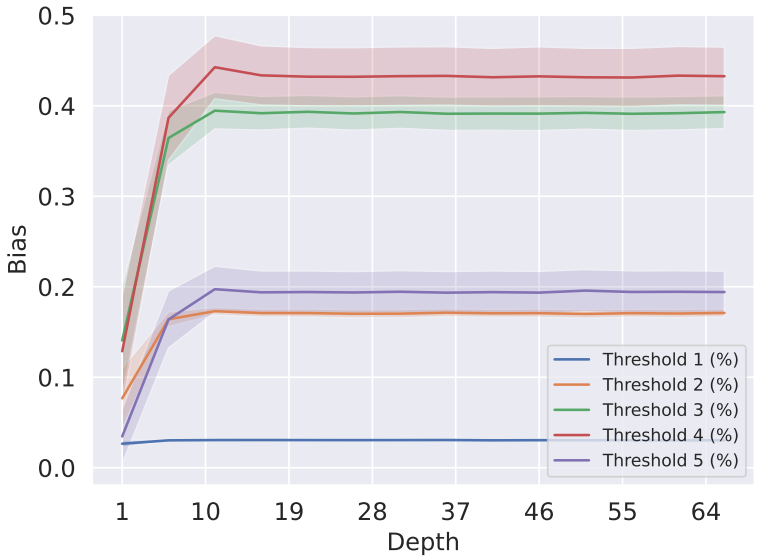
Experiments – Algorithmic Bias

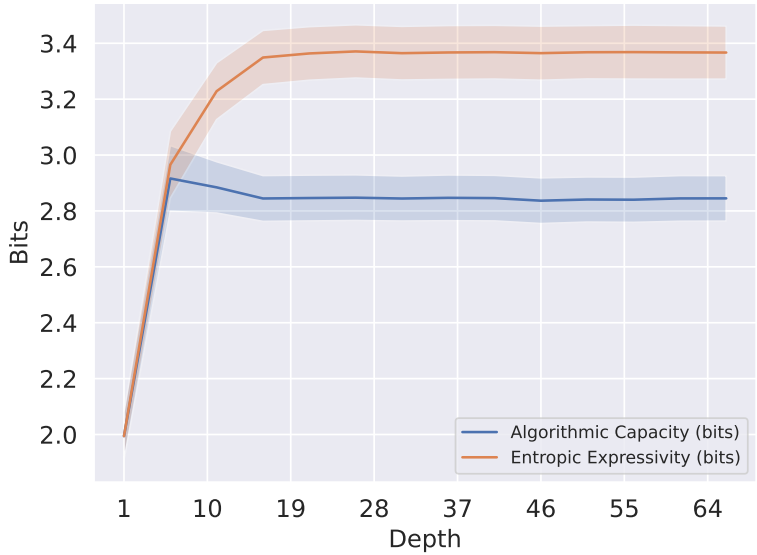
- Tested **different accuracy thresholds** for target sets
- "Target size" refers to the minimum number of labels the model must get correct (1 to 5)
- Bias is greatest for threshold sizes of 3 and 4 across all models



Experiments – Explainable Patterns in Decision Trees

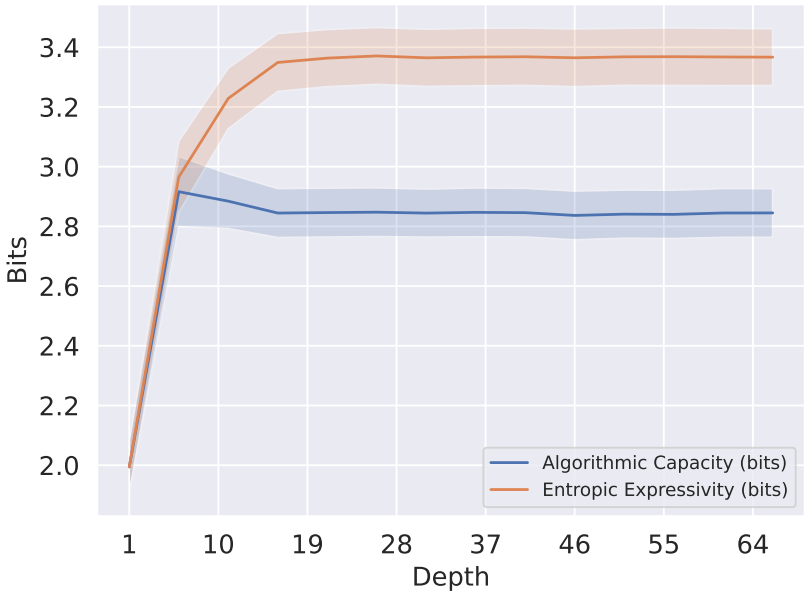
- As maximum depth increases, sensitivity to changes in training data and stochasticity increases
  - "memorizing" training data (but also noise)

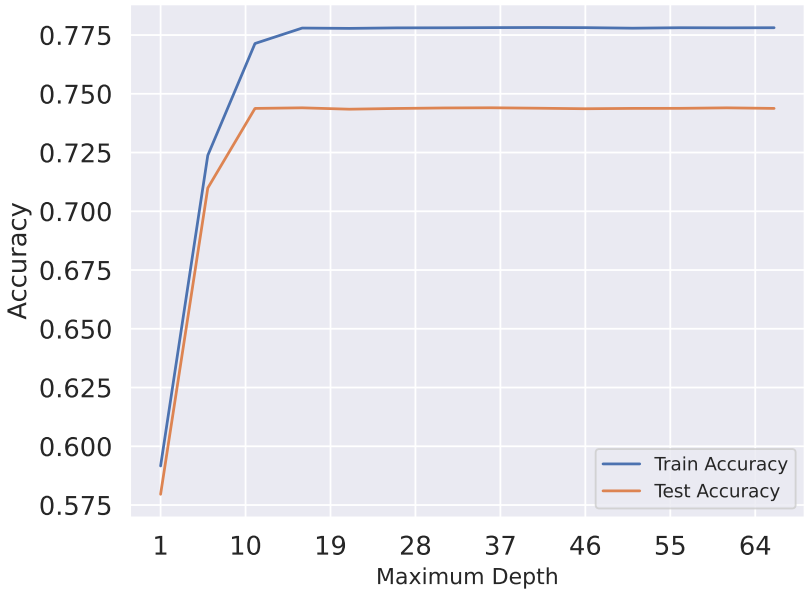




Experiments – Decision Tree Depth and Overfitting

Decision Trees exhibited **near 1 correlation coefficient** between test/train accuracy difference and Expressivity/Capacity difference.



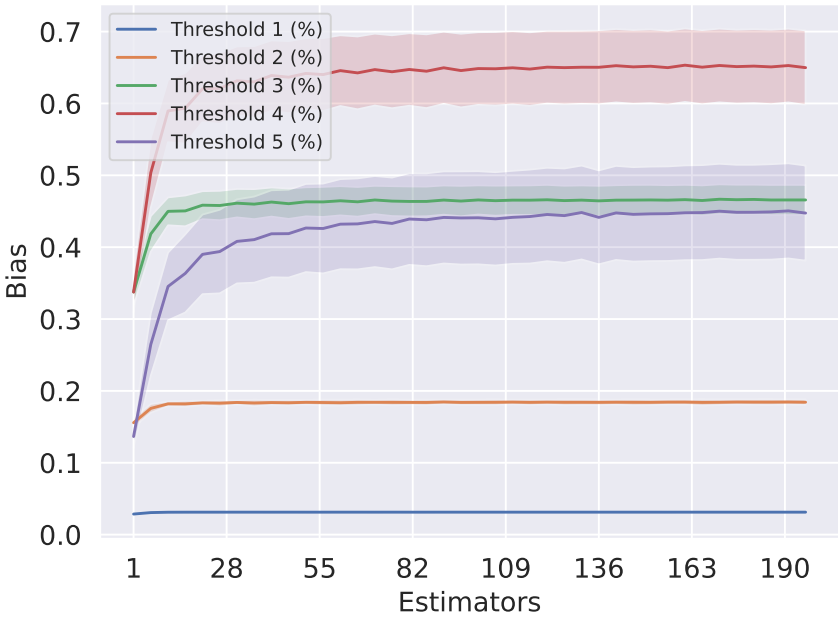


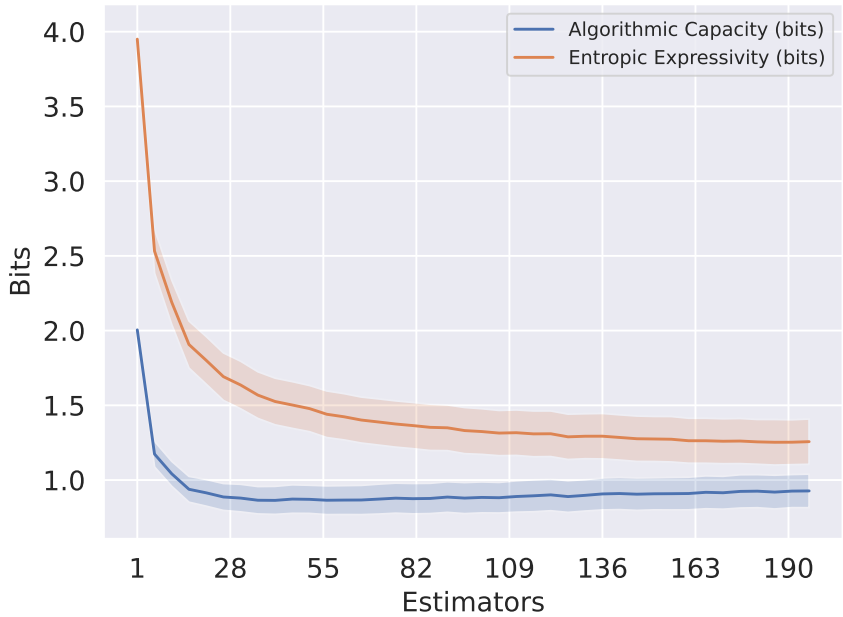
Connection between effect of **stochasticity** in training $\mathbb{E}_{\mathcal{D}}[H(\overline{\mathbf{P}}_F)]$ and **overfitting** vulnerability.

Experiments – Explainable Patterns in Random Forests

Increasing the number of estimators leads to:

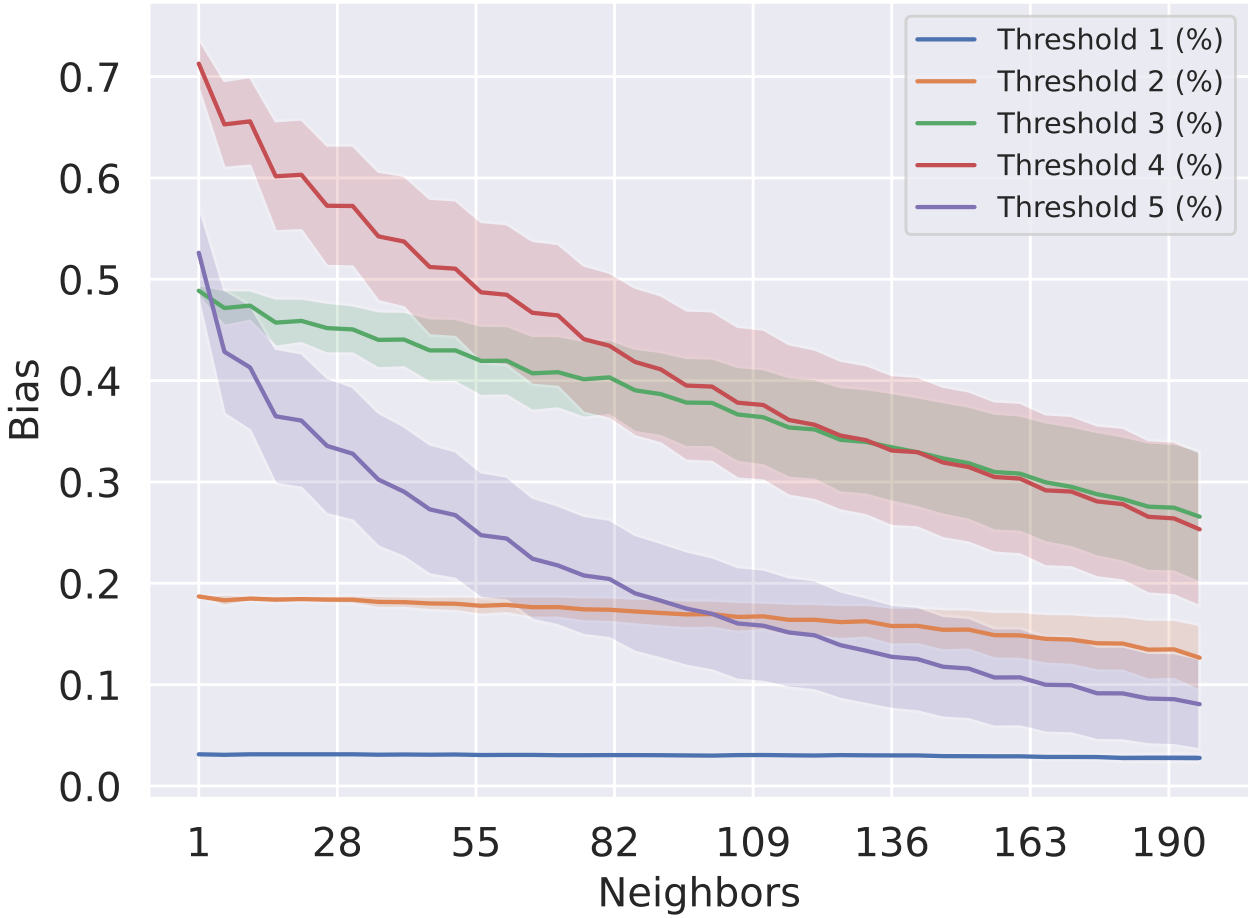
- Increasing algorithmic bias
- Decreasing entropic expressivity
  - reduction in the spread of the model's predictions
- Decreasing stochasticity





Experiments – Explainable Patterns in KNNs

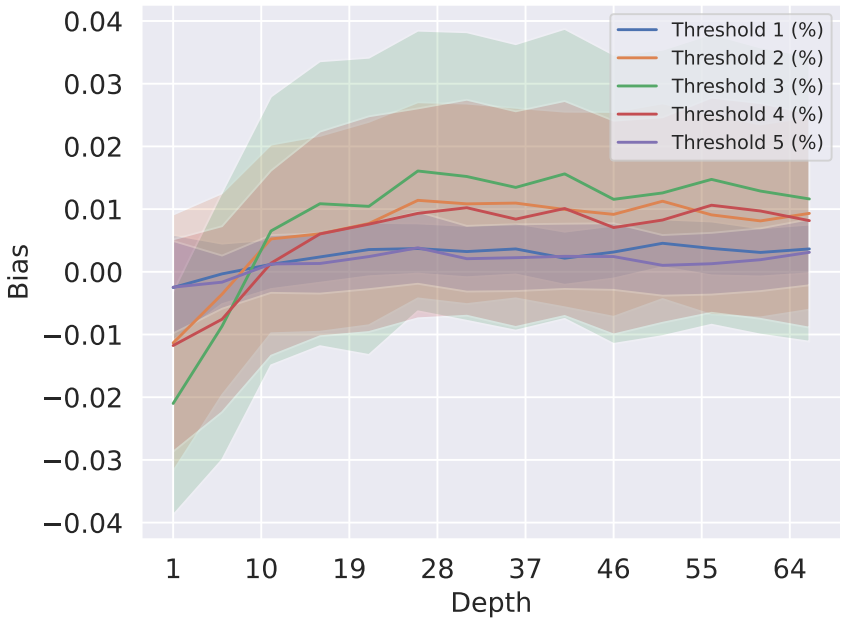
- Even-odd performance pattern
- Worse bias as k grows (majority voting)

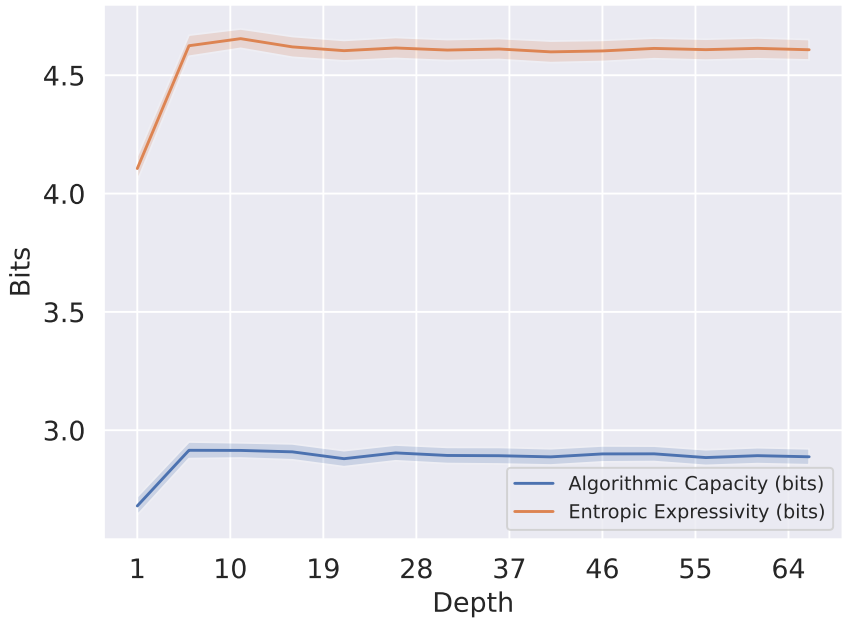


- Expressivity/capacity depends on balance of dataset

Experiments – Bias-Expressivity Tradeoff

- Montañez et. al. 2020 proved expressivity and algorithmic bias bound each other
  - “Effectiveness versus flexibility”

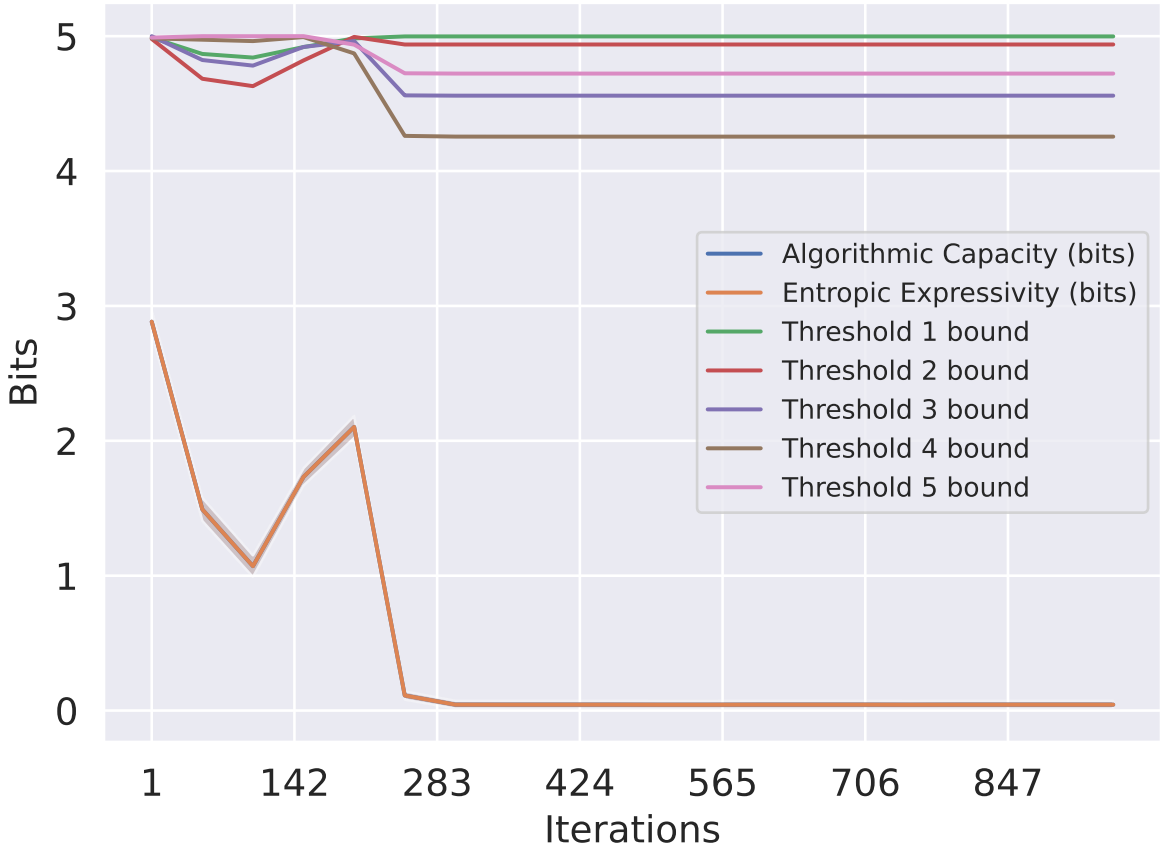




Random forest (random dataset).

Bias-Expressivity Tradeoff – Direct Bounds

Montañez et. al. 2020 also derived direct upper bounds for expressivity based on algorithmic bias.



C-support SVC entropic expressivity and upper bounds (Shopper's Intention).

Comparisons to Existing Work

Inductive orientation vectors $\rightarrow$ understandable model-theoretic metrics generalizeable to entire trained model behavior.

- Rote performance metrics: accuracy, precision/recall, and rouge
- Local estimation/explanation methods:
  - LIME, TREPAN
  - SHapley Additive exPlanations
- Upper bounds for capacity (VC dimension estimation and Rademacher complexity) and mutual information

Conclusion and Future Work

- Computed estimations of the **inductive orientation vector** $\rightarrow$ real values for information-theoretic quantities
  - New model-agnostic metrics
  - Hyperparameter/algorithm analysis
- Current focus is estimating metrics on **explainable models** for corroboration
- Future work includes applications on more larger, more complex algorithms
  - Weak point: computational intensity
- Analysis on higher vs. lower variance datasets (data distributions)

References

George Montañez (2017)
The Famine of Forte: Few Search Problems Greatly Favor Your Algorithm
IEEE SMC 2017, 477 – 482.

Bashir S., Montanez G., Sehra S., Segura P., Lauw J. (2020)
An Information-Theoretic Perspective on Overfitting and Underfitting
AJCAI 2020

Bekerman S., Chen E., Lily L., and Montanez G. (2022)
Vectorization of Bias in Machine Learning Algorithms
ICAART 2022

Segura P., Lauw J., Bashir D., Sehra S., Macias D., and Montanez G. (2022)
The Labeling Distribution Matrix (LDM)
ICAART 2020

Extra: Inductive Orientation Vector Formula

$$\overline{\mathbf{P}}_{\mathcal{D}} = \mathbb{E}_{\mathcal{D}}\left[\overline{\mathbf{P}}_{F}\right] = \mathbb{E}_{\mathcal{D}}\left[\mathbb{E}_{\tilde{P},H}\left[\frac{1}{|\tilde{P}|}\sum_{i=1}^{|\tilde{P}|} P_i \middle| F\right]\right]$$