No-Free-Lunch Theorem for ML in Isabelle/HOL

Michikazu Hirata

TPP2025,
December 4th, 2025

Today's Talk

Topic

A formalization of the no-free-lunch theorem for ML in Isabelle/HOL.

- Outline of the theorem
- Formalization in Isabelle/HOL

ML $\cdots$ Machine Learning

Reference

[1] *Understanding Machine Learning: From Theory to Algorithms*, Shai Shalev-Shwartz and Shai Ben-David, Cambridge University Press, 2014.

Source Code

AFP entry: *No-free-lunch theorem for machine learning*

The views expressed are my own and do not necessarily reflect the views of my employer.

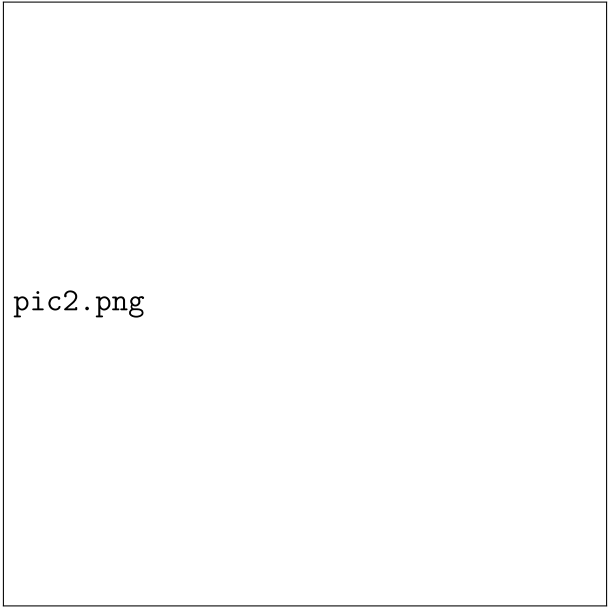
No Free Lunch

TANSTAAFL is a popular adage.

There ain't no such thing as a free lunch

Wikipedia: No such thing as a free lunch pic1_1.png

This image is generated by Gemini.



This image is generated by Gemini.

pic3.png

This image is generated by Gemini.

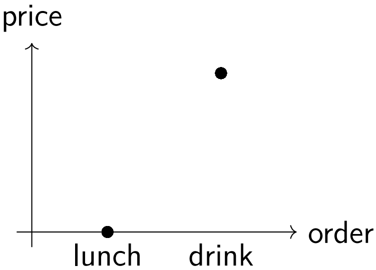
pic4.png

This image is generated by Gemini.

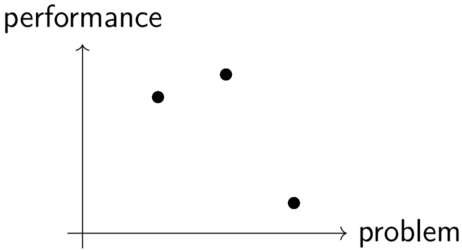
No-Free-Lunch Theorem

It is impossible to get something for nothing.

Free lunch restaurants



ML algorithms

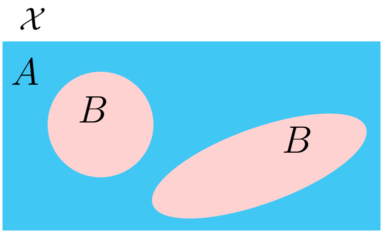


No-Free-Lunch theorem for ML (the version in [1])

No universal learner exists.
More precisely, for binary classification prediction tasks, for every learner, there exists a distribution on which it fails.

Binary Classification

Binary classification $\cdots$
the task of putting each $x \in \mathcal{X}$ into one of two categories.



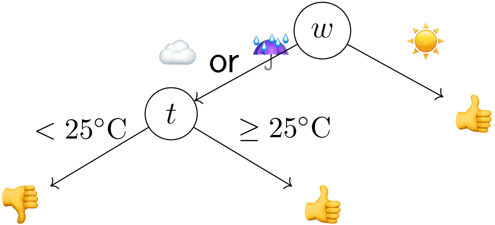
Example

Predict whether ice cream sells well or not from weather conditions.

Domain set: $\mathcal{X} = \mathrm{Weather} \times \mathrm{Temp}$

e.g. (☀️, $20^\circ\mathrm{C}$) $\in \mathcal{X}$

Categories: group 👍 and group 👎



Setting for Binary Classification Prediction Tasks

Setting

Domain set: $\mathcal{X}$

Label set: $\mathbb{B} = \{0, 1\}$

Training data: $S = ((x_1, b_1), \dots, (x_n, b_n))$, where $x_i \in \mathcal{X}$ and $b_i \in \mathbb{B}$

Learning algorithm: A, where $A(S) : \mathcal{X} \to \mathbb{B}$ is a predictor.

Example

Predict whether ice cream sells well or not from weather conditions.

Domain set: $\mathrm{Weather} \times \mathrm{Temp}$

Training data: $S = (((☀️, 20°\mathrm{C}), 👍), ((☔, 22°\mathrm{C}), 👎), \dots)$

$A(S)$ might return the decision tree in the previous page. (e.g. $A(S)(☔, 27) = 👍$)

The Method to Analyze Learning Algorithms

Assume that each training data is generated from some probability distribution.
i.e.,

$$(x, b) \sim \mathcal{D},$$
$$S \sim \mathcal{D}^m, \text{ where } \mathcal{D} \text{ is a probability distribution on } \mathcal{X} \times \mathbb{B}.$$

Error of predictor

$\mathcal{L}_{\mathcal{D}}(h) = \Pr_{(x,b)\sim\mathcal{D}}[h(x) \neq b]$ for $h : \mathcal{X} \to \mathbb{B}$.

Bias of training data

is taken into account by assuming $S \sim \mathcal{D}^m$.

A *universal learning algorithm* A may has the following property. (cf. PAC learnability)

For all $\varepsilon > 0$, $\delta > 0$, there exists M s.t. for all $m \geq M$, for all $\mathcal{D}$,

$$\Pr_{S\sim\mathcal{D}^m}[\mathcal{L}_{\mathcal{D}}(A(S)) \leq \varepsilon] \geq 1 - \delta$$

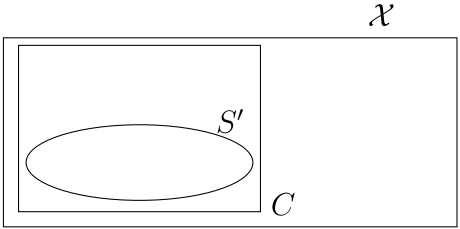
No-Free-Lunch Theorem

No-Free-Lunch Theorem for ML

Let A be a learning algorighm over a domain $\mathcal{X}$ and $m < |\mathcal{X}|/2$.
Then, there exists a distribution $\mathcal{D}$ on $\mathcal{X} \times \mathbb{B}$ s.t.

- there exists $f : \mathcal{X} \to \mathbb{B}$ s.t. $\mathcal{L}_{\mathcal{D}}(f) = 0$, and
- $\Pr_{S \sim \mathcal{D}^m} [\mathcal{L}_{\mathcal{D}}(A(S)) > 1/8] \geq 1/7$

Intuitition:
Let $C \subseteq \mathcal{X}$ s.t. $|C| = 2m$.
Then, any learning algorithm has information only half of the instances in C.

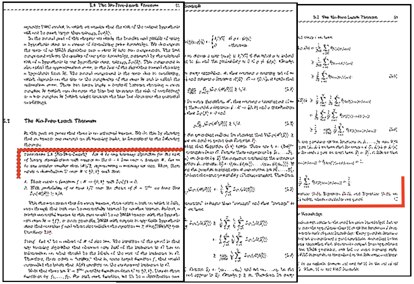


Proof Overview

1. Define $2^{|C|}$ uniform discrete distributions on $C \times \mathbb{B}$.
2. Show $\max_{\mathcal{D}} \mathbb{E}_{S \sim \mathcal{D}^m}[\mathcal{L}_{\mathcal{D}}(A(S))] \geq 1/4$.
3. Apply the Markov inequality.

In the book

2 pages of proof



In Isabelle/HOL

435 lines of proof

```
theorem no_free_lunch_ML:
  fixes X :: "'a measure" and m :: nat
    and A :: "(nat ⇒ 'a × bool) ⇒ 'a ⇒ bool"
  assumes X1:"finite (space X) ⟹ 2 * m < card (space X)"
    and X2[measurable]:"⋀x. x ∈ space X ⟹ {x} ∈ sets X"
    and m[arith]:"0 < m"
    and A[measurable]: "(λ(s,x). A s x) ∈ (PiM {..<m} (λi. X
                                              →M count_space
  shows "∃𝒟 :: ('a × bool) measure. sets 𝒟 = sets (X ⊗M c
                                         prob_space 𝒟 ∧
                       (∃f. f ∈ X →M count_space (UNIV :: bool set) ∧
                       𝒫(s in Pi M {..<m} (λi. 𝒟). 𝒫((x, y) in 𝒟. A s
proof -
```

Formalization in Isabelle/HOL

The type of learning algorithms in Isabelle/HOL.

$$A :: \underbrace{nat}_{\star\ \text{the size of data}} \Rightarrow \underbrace{(nat \Rightarrow {'a} \times bool)}_{S \in (\mathcal{X} \times \mathbb{B})^n} \Rightarrow \underbrace{{'a} \Rightarrow bool}_{\text{predictor}}$$

$\star$ is typically used to determine the number of loop executions.
We omit $\star$ because we do not need the concrete definitions of algorithms.

theorem *no-free-lunch-ML*:
fixes X :: $'a$ *measure* **and** m :: *nat*
and $A :: (nat \Rightarrow {'a} \times bool) \Rightarrow {'a} \Rightarrow bool$
assumes $0 < m$ **and** $finite\ (space\ X) \Longrightarrow 2 * m \le card\ (space\ X)$
and $\bigwedge x.\ x \in space\ X \Longrightarrow \{x\} \in sets\ X$ ← used to construct discrete distributions
and $(\lambda(s,x).\ A\ s\ x) \in (\Pi_M\ i{\in}\{..{<}m\}.\ X \bigotimes_M \mathbb{B}) \bigotimes_M X \rightarrow_M \mathbb{B}$ ← measurability
shows $\exists \mathcal{D} :: ({'a} \times bool)\ measure.\ sets\ \mathcal{D} = sets\ (X \bigotimes_M \mathbb{B}) \wedge prob\text{-}space\ \mathcal{D} \wedge$
$(\exists f.\ f \in X \rightarrow_M \mathbb{B} \wedge \mathcal{P}((x, y)\ in\ \mathcal{D}.\ f\ x \neq y) = 0) \wedge$
$\mathcal{P}(s\ in\ \Pi_M\ i{\in}\{..{<}m\}.\ \mathcal{D}.\ \mathcal{P}((x, y)\ in\ \mathcal{D}.\ A\ s\ x \neq y) > 1\ /\ 8) \ge 1\ /\ 7$

Lemma for the Proof of No-Free-Lunch Theorem

The following is used in the proof of the theorem.

Lemma

$$A = \{x_0, y_0, x_1, y_1, \ldots, x_n, y_n\}, \quad \forall i \le n,\ f(x_i) + f(y_i) = k \implies \sum_{x \in A} f(x) = k * \frac{|A|}{2}$$

lemma *sum-of-const-pairs*:
  fixes *k* :: *real*
  assumes *finite A*
    and *fst* ' *B* ∪ *snd* ' *B* = *A* **and** *fst* ' *B* ∩ *snd* ' *B* = {}
    and *inj-on fst B* **and** *inj-on snd B*
    and ⋀*x y*. (*x*,*y*) ∈ *B* ⟹ *f x* + *f y* = *k*
  shows (∑*x*∈*A*. *f x*) = *k* ∗ *real* (*card A*) / *2*

Note:
$B = \{(x_0, y_0), \ldots, (x_n, y_n)\}$

Shown by induction on A.

Conclusion

No-Free-Lunch Theorem for ML in Isabelle/HOL

- Outline of the theorem
- Formalization in Isabelle/HOL
  - How to denote learning algorithms in Isabelle/HOL
  - A lemma used in the proof