Introduction to the Theory of Machine Learning

To Trust Models or Not To Trust Models

Rachit Nimavat

IIIT Surat

June 27, 2025

Outline

Statistical View of Learning - I

We are given a *training set* $S = \{(x_1, y_1), \ldots, (x_m, y_m)\}$ of *m labeled data points*.

Statistical View of Learning - I

We are given a *training set* $S = \{(x_1, y_1), \ldots, (x_m, y_m)\}$ of m *labeled data points*.

$x \in \mathcal{X}$ is a datapoint
eg. a photo

$y \in \mathcal{Y}$ is the label
eg. Cat or Dog

Statistical View of Learning - I

We are given a *training set* $S = \{(x_1, y_1), \ldots, (x_m, y_m)\}$ of *m labeled data points*.

Let's focus on when $\mathcal{Y}$ is discrete: **Classification** (like classifying images)

Statistical View of Learning - I

We are given a *training set* $S = \{(x_1, y_1), \ldots, (x_m, y_m)\}$ of *m labeled data points*.
Let's focus on when $\mathcal{Y}$ is discrete: **Classification** (like classifying images)
If $\mathcal{Y}$ is continous: **Regression** (like predicting tomorrow's rainfall). We won't worry about it today.

Statistical View of Learning - I

We are given a *training set* $S = \{(x_1, y_1), \ldots, (x_m, y_m)\}$ of *m labeled data points*.

Let's focus on when $\mathcal{Y}$ is discrete: **Classification** (like classifying images)

If $\mathcal{Y}$ is continous: **Regression** (like predicting tomorrow's rainfall). We won't worry about it today.

*The regression, we need to worry about the rain though.

Statistical View of Learning - I

We are given a *training set* $S = \{(x_1, y_1), \ldots, (x_m, y_m)\}$ of *m labeled data points*.

Let's focus on when $\mathcal{Y}$ is discrete: **Classification** (like classifying images)

We *assume* that S is drawn i.i.d. from $\mathcal{D}$

Statistical View of Learning - I

We are given a *training set* $S = \{(x_1, y_1), \ldots, (x_m, y_m)\}$ of *m labeled data points*.
Let's focus on when $\mathcal{Y}$ is discrete: **Classification** (like classifying images)
We *assume* that S is drawn i.i.d. from $\mathcal{D}$
Our learning algorithm $\mathcal{A}$ chooses a hypothesis $h : \mathcal{X} \mapsto \mathcal{Y}$

Statistical View of Learning - I

- We are given a *training set* $S = \{(x_1, y_1), \dots, (x_m, y_m)\}$ of *m labeled data points*.
- Let's focus on when $\mathcal{Y}$ is discrete: **Classification** (like classifying images)
- We *assume* that S is drawn i.i.d. from $\mathcal{D}$
- Our learning algorithm $\mathcal{A}$ chooses a hypothesis $h : \mathcal{X} \mapsto \mathcal{Y}$ i.e on input x, the *prediction* of h is $h(x) = y$

Statistical View of Learning - I

We are given a *training set* $S = \{(x_1, y_1), \ldots, (x_m, y_m)\}$ of *m labeled data points*.

Let's focus on when $\mathcal{Y}$ is discrete: **Classification** (like classifying images)

We *assume* that S is drawn i.i.d. from $\mathcal{D}$

Our learning algorithm $\mathcal{A}$ chooses a hypothesis $h : \mathcal{X} \mapsto \mathcal{Y}$ i.e on input x, the *prediction* of h is $h(x) = y$

How well does h perform on unseen data?

Statistical View of Learning - I

We are given a *training set* $S = \{(x_1, y_1), \ldots, (x_m, y_m)\}$ of m *labeled data points*.

Let's focus on when $\mathcal{Y}$ is discrete: **Classification** (like classifying images)

We *assume* that S is drawn i.i.d. from $\mathcal{D}$

Our learning algorithm $\mathcal{A}$ chooses a hypothesis $h : \mathcal{X} \mapsto \mathcal{Y}$ i.e on input x, the *prediction* of h is $h(x) = y$

How well does h perform on unseen data?

$$\text{Test Error} \equiv \mathrm{err}_{\mathcal{D}}(h) = \mathbf{Pr}_{(x,y)\sim\mathcal{D}}\left[h(x) \neq y\right]$$

Statistical View of Learning - I

We are given a *training set* $S = \{(x_1, y_1), \ldots, (x_m, y_m)\}$ of *m labeled data points*.

Let's focus on when $\mathcal{Y}$ is discrete: **Classification** (like classifying images)

We *assume* that S is drawn i.i.d. from $\mathcal{D}$

Our learning algorithm $\mathcal{A}$ chooses a hypothesis $h : \mathcal{X} \mapsto \mathcal{Y}$ i.e on input x, the *prediction* of h is $h(x) = y$

How well does h perform on unseen data?

$$\text{Test Error} \equiv \mathrm{err}_{\mathcal{D}}(h) = \mathbf{Pr}_{(x,y)\sim\mathcal{D}}\left[h(x) \neq y\right]$$

We can measure:

$$\text{Empirical Error} \equiv \mathrm{err}_{S}(h) = \mathbf{Pr}_{(x,y)\sim S}\left[h(x) \neq y\right]$$

Statistical View of Learning - I

We are given a *training set* $S = \{(x_1, y_1), \ldots, (x_m, y_m)\}$ of *m labeled data points*.

Let's focus on when $\mathcal{Y}$ is discrete: **Classification** (like classifying images)

We *assume* that S is drawn i.i.d. from $\mathcal{D}$

Our learning algorithm $\mathcal{A}$ chooses a hypothesis $h : \mathcal{X} \mapsto \mathcal{Y}$ i.e on input x, the *prediction* of h is $h(x) = y$

How well does h perform on unseen data?

$$\text{Test Error} \equiv \mathrm{err}_{\mathcal{D}}(h) = \mathbf{Pr}_{(x,y)\sim\mathcal{D}}\left[h(x) \neq y\right]$$

We can measure:

$$\text{Empirical Error} \equiv \mathrm{err}_{S}(h) = \mathbf{Pr}_{(x,y)\sim S}\left[h(x) \neq y\right]$$

Does low Empirical Error imply low Test Error?

Statistical View of Learning - II

What we can see: $\mathrm{err}_S(h)$

Error on training set

We can calculate it

S: trainset of labeled real-world observations
h: a hypothesis that predicts a label for a (single) datapoint

Statistical View of Learning - II

What we can see: $\mathrm{err}_S(h)$

Error on training set

We can calculate it

Empirical Risk Minimizer (ERM)
minimizes error on training set

$$\mathrm{err}_S(h) = \mathbf{Pr}_{(x,y)\sim S}\left[h(x) \neq y\right]$$

S: trainset of labeled real-world observations
h: a hypothesis that predicts a label for a (single) datapoint
$\mathcal{H}$: library of hypotheses to choose from

Statistical View of Learning - II

What we can see: $\text{err}_S(h)$

Error on training set

We can calculate it

Empirical Risk Minimizer (ERM) minimizes error on training set

$$\text{err}_S(h) = \mathbf{Pr}_{(x,y)\sim S}\left[h(x) \neq y\right]$$

$$\text{ERM}(S) \equiv \underset{h' \in \mathcal{H}}{\arg\min}\, \text{err}_S(h')$$

S: trainset of labeled real-world observations
h: a hypothesis that predicts a label for a (single) datapoint
$\mathcal{H}$: library of hypotheses to choose from

Statistical View of Learning - II

What we can see: $\mathrm{err}_S(h)$

Error on training set

We can calculate it

Empirical Risk Minimizer (ERM) minimizes error on training set

$$\mathrm{err}_S(h) = \mathbf{Pr}_{(x,y)\sim S}\left[h(x) \neq y\right]$$

$$\mathrm{ERM}(S) \equiv \operatorname*{arg\,min}_{h' \in \mathcal{H}} \mathrm{err}_S(h')$$

S: trainset of labeled real-world observations
h: a hypothesis that predicts a label for a (single) datapoint
$\mathcal{H}$: library of hypotheses to choose from

True Error $\mathrm{err}_{\mathcal{D}}(h)$

Expected error on new data

We can't calculate it directly

$\mathcal{D}$: unknown real-world we are sampling from

Statistical View of Learning - II

What we can see: $\text{err}_S(h)$

Error on training set

We can calculate it

Empirical Risk Minimizer (ERM) minimizes error on training set

$$\text{err}_S(h) = \mathbf{Pr}_{(x,y)\sim S}\left[h(x) \neq y\right]$$

$$\text{ERM}(S) \equiv \underset{h' \in \mathcal{H}}{\arg\min}\, \text{err}_S(h')$$

S: trainset of labeled real-world observations
h: a hypothesis that predicts a label for a (single) datapoint
$\mathcal{H}$: library of hypotheses to choose from

True Error $\text{err}_{\mathcal{D}}(h)$

Expected error on new data

We can't calculate it directly

This is what we really care about!

$$\text{err}_{\mathcal{D}}(h) = \mathbf{Pr}_{(x,y)\sim \mathcal{D}}\left[h(x) \neq y\right]$$

$\mathcal{D}$: unknown real-world we are sampling from

Statistical View of Learning - III

$\mathrm{err}_S(h)$ is a random variable: depends on our "luck" while drawing S

Statistical View of Learning - III

$\mathrm{err}_S(h)$ is a random variable: depends on our "luck" while drawing S
$\mathrm{err}_{\mathcal{D}}(h)$ is a fixed, (but unkown) value

Statistical View of Learning - III

$\mathrm{err}_S(h)$ is a random variable: depends on our “luck” while drawing S

$\mathrm{err}_{\mathcal{D}}(h)$ is a fixed, (but unkown) value

Can’t deterministically relate training error and test error...

Statistical View of Learning - III

$\mathrm{err}_S(h)$ is a random variable: depends on our "luck" while drawing S
$\mathrm{err}_{\mathcal{D}}(h)$ is a fixed, (but unkown) value
Can't deterministically relate training error and test error...
Aim for a **probabilistic guarantee**!

Statistical View of Learning - III

$\text{err}_S(h)$ is a random variable: depends on our "luck" while drawing S
$\text{err}_{\mathcal{D}}(h)$ is a fixed, (but unkown) value
Can't deterministically relate training error and test error...
Aim for a **probabilistic guarantee**!

PAC Definition (Informal)

For any desired accuracy ($\epsilon > 0$) and confidence ($1 - \delta < 1$), we want an algorithm $\mathcal{A}$ that finds a hypothesis h satisfying:

Statistical View of Learning - III

$\text{err}_S(h)$ is a random variable: depends on our "luck" while drawing S
$\text{err}_{\mathcal{D}}(h)$ is a fixed, (but unkown) value
Can't deterministically relate training error and test error...
Aim for a **probabilistic guarantee**!

PAC Definition (Informal)

For any desired accuracy ($\epsilon > 0$) and confidence ($1 - \delta < 1$), we want an algorithm $\mathcal{A}$ that finds a hypothesis h satisfying:

$$\mathbf{Pr}_{S \sim \mathcal{D}^m} \left[\underbrace{\text{err}_{\mathcal{D}} \left(h \equiv \mathcal{A}(S) \right) < \epsilon}_{ApproximatelyCorrect} \right] > \left(\underbrace{1 - \delta}_{Probably} \right) \tag{1}$$

Statistical View of Learning - III

$\mathrm{err}_S(h)$ is a random variable: depends on our "luck" while drawing S
$\mathrm{err}_{\mathcal{D}}(h)$ is a fixed, (but unkown) value
Can't deterministically relate training error and test error...
Aim for a **probabilistic guarantee**!

PAC Definition (Informal)

For any desired accuracy $(\epsilon > 0)$ and confidence $(1 - \delta < 1)$, we want an algorithm $\mathcal{A}$ that finds a hypothesis h satisfying:

$$\mathbf{Pr}_{S \sim \mathcal{D}^m}\left[\underbrace{\mathrm{err}_{\mathcal{D}}\left(h \equiv \mathcal{A}(S)\right) < \epsilon}_{\mathit{ApproximatelyCorrect}}\right] > \left(\underbrace{1 - \delta}_{\mathit{Probably}}\right) \tag{1}$$

To Trust a Model or Not To Trust a Model?

If we have a PAC algorithm $\mathcal{A}$, we can **trust*** a model h generated by it

Generalization Guarantee for ERM Algorithm

Let's say $\mathcal{A}$ minimizes empirical error:

Generalization Guarantee for ERM Algorithm

Let's say $\mathcal{A}$ minimizes empirical error:

$$\mathcal{A}(S) = \arg\min_{h \in \mathcal{H}} \left(\mathrm{err}_S(h) \right)$$

Generalization Guarantee for ERM Algorithm

Let's say $\mathcal{A}$ minimizes empirical error:

$$\mathcal{A}(S) = \arg\min_{h \in \mathcal{H}} \left(\mathrm{err}_S(h) \right)$$

(Simplyfing Assumption) The **Realizable Case**.

Generalization Guarantee for ERM Algorithm

Let's say $\mathcal{A}$ minimizes empirical error:

$$\mathcal{A}(S) = \arg\min_{h \in \mathcal{H}} (\mathrm{err}_S(h))$$

(Simplyfing Assumption) The **Realizable Case**. Assume that the 'correct' rule $f \in \mathcal{H}$.

Generalization Guarantee for ERM Algorithm

Let's say $\mathcal{A}$ minimizes empirical error:

$$\mathcal{A}(S) = \arg\min_{h \in \mathcal{H}} \left(\mathrm{err}_S(h)\right)$$

(Simplyfing Assumption) The **Realizable Case**. Assume that the 'correct' rule $f \in \mathcal{H}$.

$$\mathcal{A}(S) = h \ : \ \mathrm{err}_S(h) = 0$$

Generalization Guarantee for ERM Algorithm

Let's say $\mathcal{A}$ minimizes empirical error:

$$\mathcal{A}(S) = \arg\min_{h \in \mathcal{H}} \left(\mathrm{err}_S(h)\right)$$

(Simplyfing Assumption) The **Realizable Case**. Assume that the 'correct' rule $f \in \mathcal{H}$.

$$\mathcal{A}(S) = h \;:\; \mathrm{err}_S(h) = 0$$

There are possibly many rules, including the correct rule f, that achieve 0-training error. ERM algorithm $\mathcal{A}$ may report any such hypothesis.

Generalization Guarantee for ERM Algorithm

Let's say $\mathcal{A}$ minimizes empirical error:

$$\mathcal{A}(S) = \arg\min_{h \in \mathcal{H}} (\mathrm{err}_S(h))$$

(Simplyfing Assumption) The **Realizable Case**. Assume that the 'correct' rule $f \in \mathcal{H}$.

$$\mathcal{A}(S) = h \ : \ \mathrm{err}_S(h) = 0$$

There are possibly many rules, including the correct rule f, that achieve 0-training error. ERM algorithm $\mathcal{A}$ may report any such hypothesis.

ERM to PAC

When can we say $\mathrm{err}_{\mathcal{D}}(h) < \epsilon$ with probability $1 - \delta$?

ERM to PAC

What must have happended if we found h with $\mathrm{err}_S(h) = 0$ but with $\mathrm{err}_{\mathcal{D}}(h) > \epsilon$?

ERM to PAC

What must have happended if we found h with $\mathrm{err}_S(h) = 0$ but with $\mathrm{err}_{\mathcal{D}}(h) > \epsilon$?
We must have been very unlucky in our draw of training data!

ERM to PAC

What must have happended if we found h with $\mathrm{err}_S(h) = 0$ but with $\mathrm{err}_{\mathcal{D}}(h) > \epsilon$?
We must have been very unlucky in our draw of training data!
Some *bad* hypothesis h **fooled** our training set S (i.e., had 0-empirical error)

ERM to PAC

What must have happended if we found h with $\text{err}_S(h) = 0$ but with $\text{err}_{\mathcal{D}}(h) > \epsilon$?
We must have been very unlucky in our draw of training data!
Some *bad* hypothesis h **fooled** our training set S (i.e., had 0-empirical error)

Probability of getting fooled by a single *bad* hypothesis

Fix a *bad* hypothesis $h \in \mathcal{H}$ i.e with $\text{err}_{\mathcal{D}}(h) > \epsilon$

ERM to PAC

What must have happended if we found h with $\text{err}_S(h) = 0$ but with $\text{err}_{\mathcal{D}}(h) > \epsilon$?
We must have been very unlucky in our draw of training data!
Some *bad* hypothesis h **fooled** our training set S (i.e., had 0-empirical error)

Probability of getting fooled by a single *bad* hypothesis

Fix a *bad* hypothesis $h \in \mathcal{H}$ i.e with $\text{err}_{\mathcal{D}}(h) > \epsilon$

When can it *fool* a labeled data point (x, y)?

ERM to PAC

What must have happended if we found h with $\mathrm{err}_S(h) = 0$ but with $\mathrm{err}_{\mathcal{D}}(h) > \epsilon$?
We must have been very unlucky in our draw of training data!
Some *bad* hypothesis h **fooled** our training set S (i.e., had 0-empirical error)

Probability of getting fooled by a single *bad* hypothesis

Fix a *bad* hypothesis $h \in \mathcal{H}$ i.e with $\mathrm{err}_{\mathcal{D}}(h) > \epsilon$

When can it *fool* a labeled data point (x, y)?

$$h(x) = y.$$

ERM to PAC

What must have happended if we found h with $\mathrm{err}_S(h) = 0$ but with $\mathrm{err}_{\mathcal{D}}(h) > \epsilon$?
We must have been very unlucky in our draw of training data!
Some *bad* hypothesis h **fooled** our training set S (i.e., had 0-empirical error)

Probability of getting fooled by a single *bad* hypothesis

Fix a *bad* hypothesis $h \in \mathcal{H}$ i.e with $\mathrm{err}_{\mathcal{D}}(h) > \epsilon$

When can it *fool* a labeled data point (x, y)?

$h(x) = y$. But $\mathbf{Pr}_{(x,y)\sim\mathcal{D}}\left[h(x) = y\right]$

ERM to PAC

What must have happended if we found h with $\mathrm{err}_S(h) = 0$ but with $\mathrm{err}_{\mathcal{D}}(h) > \epsilon$?
We must have been very unlucky in our draw of training data!
Some *bad* hypothesis h **fooled** our training set S (i.e., had 0-empirical error)

Probability of getting fooled by a single *bad* hypothesis

Fix a *bad* hypothesis $h \in \mathcal{H}$ i.e with $\mathrm{err}_{\mathcal{D}}(h) > \epsilon$

When can it *fool* a labeled data point (x, y)?

$h(x) = y$. But $\mathbf{Pr}_{(x,y)\sim\mathcal{D}}[h(x) = y] = 1 - \mathrm{err}_{\mathcal{D}}(h)$

ERM to PAC

What must have happended if we found h with $\text{err}_S(h) = 0$ but with $\text{err}_{\mathcal{D}}(h) > \epsilon$?
We must have been very unlucky in our draw of training data!
Some *bad* hypothesis h **fooled** our training set S (i.e., had 0-empirical error)

Probability of getting fooled by a single *bad* hypothesis

Fix a *bad* hypothesis $h \in \mathcal{H}$ i.e with $\text{err}_{\mathcal{D}}(h) > \epsilon$

When can it *fool* a labeled data point (x, y)?

$h(x) = y$. But $\mathbf{Pr}_{(x,y)\sim\mathcal{D}}[h(x) = y] = 1 - \text{err}_{\mathcal{D}}(h) < (1 - \epsilon)$.

ERM to PAC

What must have happended if we found h with $\mathrm{err}_S(h) = 0$ but with $\mathrm{err}_{\mathcal{D}}(h) > \epsilon$?
We must have been very unlucky in our draw of training data!
Some *bad* hypothesis h **fooled** our training set S (i.e., had 0-empirical error)

Probability of getting fooled by a single *bad* hypothesis

Fix a *bad* hypothesis $h \in \mathcal{H}$ i.e with $\mathrm{err}_{\mathcal{D}}(h) > \epsilon$

When can it *fool* a labeled data point (x, y)?

$h(x) = y$. But $\mathbf{Pr}_{(x,y)\sim\mathcal{D}}[h(x) = y] = 1 - \mathrm{err}_{\mathcal{D}}(h) < (1 - \epsilon)$.

The probability that h *fooled* S with m data points:

ERM to PAC

What must have happended if we found h with $\text{err}_S(h) = 0$ but with $\text{err}_{\mathcal{D}}(h) > \epsilon$?
We must have been very unlucky in our draw of training data!
Some *bad* hypothesis h **fooled** our training set S (i.e., had 0-empirical error)

Probability of getting fooled by a single *bad* hypothesis

Fix a *bad* hypothesis $h \in \mathcal{H}$ i.e with $\text{err}_{\mathcal{D}}(h) > \epsilon$

When can it *fool* a labeled data point (x, y)?

$h(x) = y$. But $\mathbf{Pr}_{(x,y)\sim\mathcal{D}}[h(x) = y] = 1 - \text{err}_{\mathcal{D}}(h) < (1 - \epsilon)$.

The probability that h *fooled* S with m data points:

$$\mathbf{Pr}_{S\sim\mathcal{D}^m}[h \text{ fooled } S] = \underbrace{\left(\mathbf{Pr}_{(x,y)\sim\mathcal{D}}[h(x) = y]\right)^m}$$

ERM to PAC

What must have happended if we found h with $\mathrm{err}_S(h) = 0$ but with $\mathrm{err}_{\mathcal{D}}(h) > \epsilon$?
We must have been very unlucky in our draw of training data!
Some *bad* hypothesis h **fooled** our training set S (i.e., had 0-empirical error)

Probability of getting fooled by a single *bad* hypothesis

Fix a *bad* hypothesis $h \in \mathcal{H}$ i.e with $\mathrm{err}_{\mathcal{D}}(h) > \epsilon$

When can it *fool* a labeled data point (x, y)?

$h(x) = y$. But $\mathbf{Pr}_{(x,y)\sim\mathcal{D}}[h(x) = y] = 1 - \mathrm{err}_{\mathcal{D}}(h) < (1 - \epsilon)$.

The probability that h *fooled* S with m data points:

$$\mathbf{Pr}_{S\sim\mathcal{D}^m}[h \text{ fooled } S] = \underbrace{\left(\mathbf{Pr}_{(x,y)\sim\mathcal{D}}[h(x) = y]\right)^m}_{\text{i.i.d assumption}}$$

ERM to PAC

What must have happended if we found h with $\text{err}_S(h) = 0$ but with $\text{err}_{\mathcal{D}}(h) > \epsilon$?
We must have been very unlucky in our draw of training data!
Some *bad* hypothesis h **fooled** our training set S (i.e., had 0-empirical error)

Probability of getting fooled by a single *bad* hypothesis

Fix a *bad* hypothesis $h \in \mathcal{H}$ i.e with $\text{err}_{\mathcal{D}}(h) > \epsilon$

When can it *fool* a labeled data point (x, y)?

$h(x) = y$. But $\mathbf{Pr}_{(x,y)\sim\mathcal{D}}[h(x) = y] = 1 - \text{err}_{\mathcal{D}}(h) < (1 - \epsilon)$.

The probability that h *fooled* S with m data points:

$$\mathbf{Pr}_{S\sim\mathcal{D}^m}[h \text{ fooled } S] = \underbrace{\left(\mathbf{Pr}_{(x,y)\sim\mathcal{D}}[h(x) = y]\right)^m}_{\text{i.i.d assumption}}$$

$$< (1 - \epsilon)^m$$

ERM to PAC

What must have happended if we found h with $\mathrm{err}_S(h) = 0$ and $\mathrm{err}_{\mathcal{D}}(h) > \epsilon$?

Some *bad* hypothesis h **fooled** our training set S (i.e., had 0-empirical error)

ERM to PAC

What must have happended if we found h with $\mathrm{err}_S(h) = 0$ and $\mathrm{err}_{\mathcal{D}}(h) > \epsilon$?

Some *bad* hypothesis h **fooled** our training set S (i.e., had 0-empirical error)

A **single** bad hypothesis h can fool us with probability

$$\mathbf{Pr}_{S \sim \mathcal{D}^m}[h \text{ fooled } S] < (1 - \epsilon)^m$$

ERM to PAC

What must have happended if we found h with $\text{err}_S(h) = 0$ and $\text{err}_{\mathcal{D}}(h) > \epsilon$?

Some *bad* hypothesis h **fooled** our training set S (i.e., had 0-empirical error)

A **single** bad hypothesis h can fool us with probability

$$\mathbf{Pr}_{S \sim \mathcal{D}^m} [h \text{ fooled } S] < (1 - \epsilon)^m$$

We got fooled by **some** hypothesis $h \in \mathcal{H}$:

ERM to PAC

What must have happended if we found h with $\mathrm{err}_S(h) = 0$ and $\mathrm{err}_{\mathcal{D}}(h) > \epsilon$?

Some *bad* hypothesis h **fooled** our training set S (i.e., had 0-empirical error)

A **single** bad hypothesis h can fool us with probability

$$\mathbf{Pr}_{S \sim \mathcal{D}^m}\left[h \text{ fooled } S\right] < (1-\epsilon)^m$$

We got fooled by **some** hypothesis $h \in \mathcal{H}$:

$$\mathbf{Pr}\left[\mathrm{err}_{\mathcal{D}} > h\right] = \mathbf{Pr}\left[\text{some } h \text{ fooled us}\right]$$

ERM to PAC

What must have happended if we found h with $\mathrm{err}_S(h) = 0$ and $\mathrm{err}_{\mathcal{D}}(h) > \epsilon$?

Some *bad* hypothesis h **fooled** our training set S (i.e., had 0-empirical error)

A **single** bad hypothesis h can fool us with probability

$$\mathbf{Pr}_{S \sim \mathcal{D}^m}\left[h \text{ fooled } S\right] < (1-\epsilon)^m$$

We got fooled by **some** hypothesis $h \in \mathcal{H}$:

$$\mathbf{Pr}\left[\mathrm{err}_{\mathcal{D}} > h\right] = \mathbf{Pr}\left[\text{some } h \text{ fooled us}\right] \leq |\mathcal{H}| \cdot \mathbf{Pr}\left[\text{a specific } h \text{ fooled us}\right]$$

ERM to PAC

What must have happended if we found h with $\text{err}_S(h) = 0$ and $\text{err}_{\mathcal{D}}(h) > \epsilon$?

Some *bad* hypothesis h **fooled** our training set S (i.e., had 0-empirical error)

A **single** bad hypothesis h can fool us with probability

$$\mathbf{Pr}_{S\sim\mathcal{D}^m}\left[h \text{ fooled } S\right] < (1-\epsilon)^m$$

We got fooled by **some** hypothesis $h \in \mathcal{H}$:

$$\begin{aligned}\mathbf{Pr}\left[\text{err}_{\mathcal{D}} > h\right] = \mathbf{Pr}\left[\text{some } h \text{ fooled us}\right] &\leq |\mathcal{H}| \cdot \mathbf{Pr}\left[\text{a specific } h \text{ fooled us}\right] \\ &\leq |\mathcal{H}| \cdot (1-\epsilon)^m\end{aligned}$$

ERM to PAC

What must have happended if we found h with $\text{err}_S(h) = 0$ and $\text{err}_{\mathcal{D}}(h) > \epsilon$?

Some *bad* hypothesis h **fooled** our training set S (i.e., had 0-empirical error)

A **single** bad hypothesis h can fool us with probability

$$\mathbf{Pr}_{S \sim \mathcal{D}^m} [h \text{ fooled } S] < (1-\epsilon)^m$$

We got fooled by **some** hypothesis $h \in \mathcal{H}$:

$$\begin{aligned} \mathbf{Pr}[\text{err}_{\mathcal{D}} > h] = \mathbf{Pr}[\text{some } h \text{ fooled us}] &\leq |\mathcal{H}| \cdot \mathbf{Pr}[\text{a specific } h \text{ fooled us}] \\ &\leq |\mathcal{H}| \cdot (1-\epsilon)^m < \delta, \end{aligned}$$

$$\boxed{\text{if } m \geq \frac{1}{\epsilon}\left(\ln|\mathcal{H}| + \ln\left(\frac{1}{\delta}\right)\right)}$$

ERM to PAC - What just happened?!

Observation 2.1

Under the fundamental assumption of ML[a]*, for a set $\mathcal{H}$ of hypotheses,*

ERM to PAC - What just happened?!

Observation 2.1

*Under the fundamental assumption of ML[a], for a set $\mathcal{H}$ of hypotheses, **any** algorithm that finds a hypothesis $h \in \mathcal{H}$ with 0 empirical error for S with $m \geq \frac{1}{\epsilon}\left(\ln|\mathcal{H}| + \ln\left(\frac{1}{\delta}\right)\right)$ examples is PAC!*

[a] that samples are drawn i.i.d. from a fixed, unknown $\mathcal{D}$

ERM to PAC - What just happened?!

Observation 2.1

Under the fundamental assumption of ML[a]*, for a set $\mathcal{H}$ of hypotheses,* ***any*** *algorithm that finds a hypothesis $h \in \mathcal{H}$ with 0 empirical error for S with $m \geq \frac{1}{\epsilon}\left(\ln|\mathcal{H}| + \ln\left(\frac{1}{\delta}\right)\right)$ examples is PAC!*

[a] that samples are drawn i.i.d. from a fixed, unknown $\mathcal{D}$

- Valid for non-ERM algorithms as well

ERM to PAC - What just happened?!

Observation 2.1

*Under the fundamental assumption of ML[a], for a set $\mathcal{H}$ of hypotheses, **any** algorithm that finds a hypothesis $h \in \mathcal{H}$ with 0 empirical error for S with $m \geq \frac{1}{\epsilon}\left(\ln|\mathcal{H}| + \ln\left(\frac{1}{\delta}\right)\right)$ examples is PAC!*

[a] that samples are drawn i.i.d. from a fixed, unknown $\mathcal{D}$

- Valid for non-ERM algorithms as well
- More genrally, we can show that $\text{err}_{\mathcal{D}} < \text{err}_S + \epsilon$

ERM to PAC - What just happened?!

Observation 2.1

*Under the fundamental assumption of ML[a], for a set $\mathcal{H}$ of hypotheses, **any** algorithm that finds a hypothesis $h \in \mathcal{H}$ with 0 empirical error for S with $m \geq \frac{1}{\epsilon}\left(\ln|\mathcal{H}| + \ln\left(\frac{1}{\delta}\right)\right)$ examples is PAC!*

[a]that samples are drawn i.i.d. from a fixed, unknown $\mathcal{D}$

- Valid for non-ERM algorithms as well
- More genrally, we can show that $\text{err}_{\mathcal{D}} < \text{err}_S + \epsilon$ i.e err_S is a represntative of $\text{err}_{\mathcal{D}}$ provided we have sufficient samples

ERM to PAC - What just happened?!

Observation 2.1

*Under the fundamental assumption of ML[a], for a set $\mathcal{H}$ of hypotheses, **any** algorithm that finds a hypothesis $h \in \mathcal{H}$ with 0 empirical error for S with $m \geq \frac{1}{\epsilon}\left(\ln|\mathcal{H}| + \ln\left(\frac{1}{\delta}\right)\right)$ examples is PAC!*

[a]that samples are drawn i.i.d. from a fixed, unknown $\mathcal{D}$

- Valid for non-ERM algorithms as well
- More genrally, we can show that $\text{err}_{\mathcal{D}} < \text{err}_S + \epsilon$ i.e err_S is a represntative of $\text{err}_{\mathcal{D}}$ provided we have sufficient samples
- m depends logarithmically on the size of $\mathcal{H}$

ERM to PAC - What just happened?!

Observation 2.1

Under the fundamental assumption of ML[a]*, for a set $\mathcal{H}$ of hypotheses,* ***any*** *algorithm that finds a hypothesis $h \in \mathcal{H}$ with 0 empirical error for S with $m \geq \frac{1}{\epsilon}\left(\ln|\mathcal{H}| + \ln\left(\frac{1}{\delta}\right)\right)$ examples is PAC!*

[a]that samples are drawn i.i.d. from a fixed, unkn

Sample Size (m)

Error (ϵ) ← Observation 2.1 → **Confidence ($1-\delta$)**

Model Complexity ($|\mathcal{H}|$)

- Valid for ERM ell
- More ge… $\text{err}_{\mathcal{D}} < \text{err}_S + \epsilon$ i.e err_S is a represntative of $\text{err}_{\mathcal{D}}$ provided we have sufficient samples
- m depends logarithmically on the size of $\mathcal{H}$

Occam's Razor

Observation 3.1 (Simplified Sample Complexity Bound)

$h \in \mathcal{H}$ *with* $\mathrm{err}_S(h) = 0 \rightarrow \mathrm{err}_{\mathcal{D}}(h) < \epsilon$, *provided* S *had* $m \geq \frac{\ln |\mathcal{H}|}{\epsilon}$ *samples.*

Occam's Razor

Observation 3.1 (Simplified Sample Complexity Bound)

$h \in \mathcal{H}$ with $\text{err}_S(h) = 0 \rightarrow \text{err}_{\mathcal{D}}(h) < \epsilon$, provided S had $m \geq \frac{\ln|\mathcal{H}|}{\epsilon}$ samples.

A smaller $|\mathcal{H}|$ means fewer samples to get the same guarantee!

Occam's Razor

Observation 3.1 (Simplified Sample Complexity Bound)

$h \in \mathcal{H}$ *with* $\mathrm{err}_S(h) = 0 \to \mathrm{err}_{\mathcal{D}}(h) < \epsilon$, *provided* S *had* $m \geq \frac{\ln |\mathcal{H}|}{\epsilon}$ *samples.*

A smaller $|\mathcal{H}|$ means fewer samples to get the same guarantee!

"Among competing hypotheses, the one with the fewest assumptions should be selected."

Occam's Razor

Observation 3.1 (Simplified Sample Complexity Bound)

$h \in \mathcal{H}$ *with* $\mathrm{err}_S(h) = 0 \rightarrow \mathrm{err}_{\mathcal{D}}(h) < \epsilon$, *provided* S *had* $m \geq \frac{\ln |\mathcal{H}|}{\epsilon}$ *samples.*

A smaller $|\mathcal{H}|$ means fewer samples to get the same guarantee!

"Among competing hypotheses, the one with the fewest assumptions should be selected."

Occam's Razor (c. 1320)

Plurality should not be posited without necessity

Occam's Razor

Observation 3.1 (Simplified Sample Complexity Bound)

$h \in \mathcal{H}$ *with* $\text{err}_S(h) = 0 \rightarrow \text{err}_{\mathcal{D}}(h) < \epsilon$, *provided* S *had* $m \geq \frac{\ln |\mathcal{H}|}{\epsilon}$ *samples.*

A smaller $|\mathcal{H}|$ means fewer samples to get the same guarantee!

"Among competing hypotheses, the one with the fewest assumptions should be selected."

Occam's Razor (c. 1320)

Plurality should not be posited without necessity

- Observation 2.1 provides a formal, mathematical justification for this ancient principle

Occam's Razor

Observation 3.1 (Simplified Sample Complexity Bound)

$h \in \mathcal{H}$ with $\text{err}_S(h) = 0 \rightarrow \text{err}_{\mathcal{D}}(h) < \epsilon$, provided S had $m \geq \frac{\ln |\mathcal{H}|}{\epsilon}$ samples.

A smaller $|\mathcal{H}|$ means fewer samples to get the same guarantee!

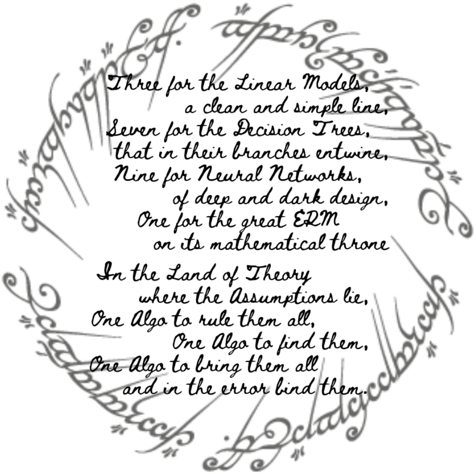
"Among competing hypotheses, the one with the fewest assumptions should be selected."

Occam's Razor (c. 1320)

Plurality should not be posited without necessity

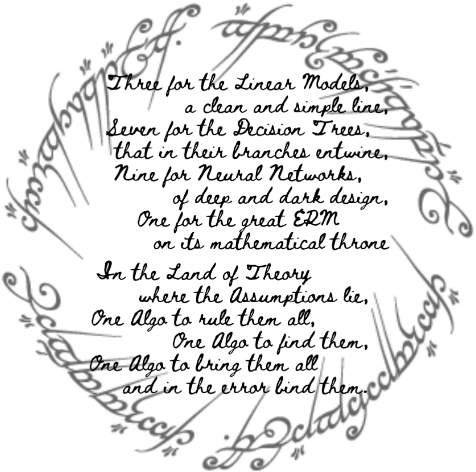
- Observation 2.1 provides a formal, mathematical justification for this ancient principle
- Simpler models (smaller $|\mathcal{H}|$) are easier to work with **and** they provide better generalization

ERM: A master algorithm to rule them all?



ERM: the master algorithm for ML?

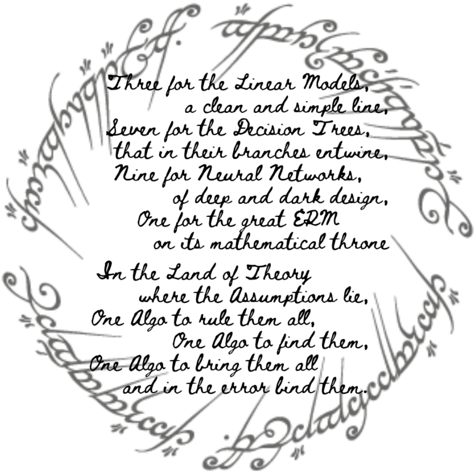
ERM: A master algorithm to rule them all?



ERM: the master algorithm for ML?

Pattern is *geniune* and not noise $\rightarrow$ *simple* hypothesis class $\mathcal{H}$ capturing it

ERM: A master algorithm to rule them all?

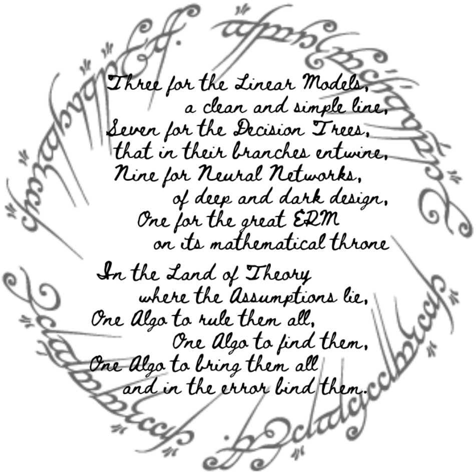


ERM: the master algorithm for ML?

Pattern is *geniune* and not noise $\rightarrow$ *simple* hypothesis class $\mathcal{H}$ capturing it

Need only $\approx |\mathcal{H}|/\epsilon$ samples to learn $\mathcal{H}$

ERM: A master algorithm to rule them all?



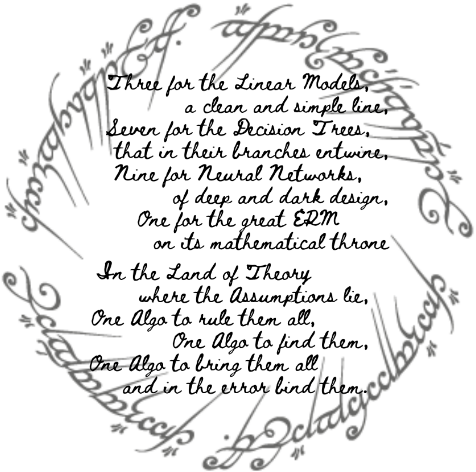
ERM: the master algorithm for ML?

Pattern is *geniune* and not noise $\rightarrow$ *simple* hypothesis class $\mathcal{H}$ capturing it

Need only $\approx |\mathcal{H}|/\epsilon$ samples to learn $\mathcal{H}$

Every "learnable" class of hypotheses can be learnt by ERM!

ERM: A master algorithm to rule them all?



ERM: the master algorithm for ML?

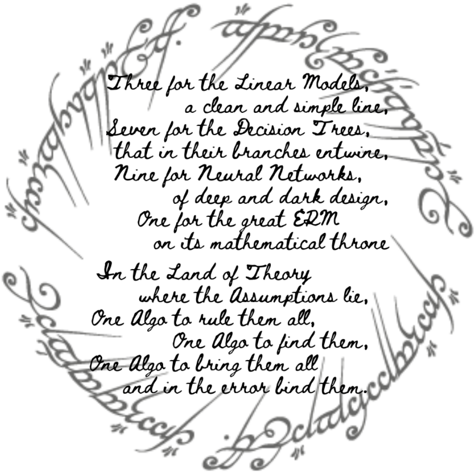
Pattern is *geniune* and not noise $\rightarrow$ *simple* hypothesis class $\mathcal{H}$ capturing it

Need only $\approx |\mathcal{H}|/\epsilon$ samples to learn $\mathcal{H}$

Every "learnable" class of hypotheses can be learnt by ERM!

ML should then be a solved problem!

ERM: A master algorithm to rule them all?



ERM: the master algorithm for ML?

Pattern is *geniune* and not noise $\rightarrow$ *simple* hypothesis class $\mathcal{H}$ capturing it

Need only $\approx |\mathcal{H}|/\epsilon$ samples to learn $\mathcal{H}$

Every "learnable" class of hypotheses can be learnt by ERM!

ML should then be a solved problem!

Except it is not...

Issues with ERM : Computational Barrier

ERM is an *principle*, not an *algorithm*.

How to actually *compute* $\arg\min \mathrm{err}_S(h)$?

Issues with ERM : Computational Barrier

ERM is an *principle*, not an *algorithm*.

How to actually *compute* $\arg\min \mathrm{err}_S(h)$?

Finding the simplest 10-line Python program that fits a dataset is **computationally impossible**

Issues with ERM : Computational Barrier

ERM is an *principle*, not an *algorithm*.

How to actually *compute* $\arg\min \mathrm{err}_S(h)$?

Finding the simplest 10-line Python program that fits a dataset is **computationally impossible**
Finding the wieghts of a neural network to correctly classify a dataset takes **exponential time**

Issues with ERM : Computational Barrier

ERM is an *principle*, not an *algorithm*.

How to actually *compute* $\arg\min \mathrm{err}_S(h)$?

Finding the simplest 10-line Python program that fits a dataset is **computationally impossible**
Finding the wieghts of a neural network to correctly classify a dataset takes **exponential time**

Efficiently PAC Algorithms

Just because a simple solution exists doesn't mean we can find it efficiently

Issues with ERM : Computational Barrier

ERM is an *principle*, not an *algorithm*.

How to actually *compute* $\arg\min \mathrm{err}_S(h)$?

Finding the simplest 10-line Python program that fits a dataset is **computationally impossible**
Finding the wieghts of a neural network to correctly classify a dataset takes **exponential time**

Efficiently PAC Algorithms

Just because a simple solution exists doesn't mean we can find it efficiently

We need different algorithms like SVM, Regression, Naive Bayes,...

Issues with ERM : Statistical Barrier

What if the world isn't simple?

Every algorithm makes a bet on the nature of the problem

Issues with ERM : Statistical Barrier

What if the world isn't simple?

Every algorithm makes a bet on the nature of the problem

- A Linear Classifier bets the data is 'mostly' separable by a line

Issues with ERM : Statistical Barrier

What if the world isn't simple?

Every algorithm makes a bet on the nature of the problem

- A Linear Classifier bets the data is 'mostly' separable by a line
- A Deep Neural Network bets the solution is a complex, hierarchical function

Issues with ERM : Statistical Barrier

What if the world isn't simple?

Every algorithm makes a bet on the nature of the problem

- A Linear Classifier bets the data is 'mostly' separable by a line
- A Deep Neural Network bets the solution is a complex, hierarchical function
- Current LLMs bet *context* is everything and true understanding emerges from mastering statistical patterns

Issues with ERM : Statistical Barrier

What if the world isn't simple?

Every algorithm makes a bet on the nature of the problem

- A Linear Classifier bets the data is 'mostly' separable by a line
- A Deep Neural Network bets the solution is a complex, hierarchical function
- Current LLMs bet *context* is everything and true understanding emerges from mastering statistical patterns

No Free Lunch Theorem (Informal)

For any learning algorithm $\mathcal{A}$, there exists a distribution $\mathcal{D}$ on which it performs poorly.

Issues with ERM : Statistical Barrier

What if the world isn't simple?

Every algorithm makes a bet on the nature of the problem

- A Linear Classifier bets the data is 'mostly' separable by a line
- A Deep Neural Network bets the solution is a complex, hierarchical function
- Current LLMs bet *context* is everything and true understanding emerges from mastering statistical patterns

No Free Lunch Theorem (Informal)

For any learning algorithm $\mathcal{A}$, there exists a distribution $\mathcal{D}$ on which it performs poorly. Averaged over all $\mathcal{D}$, the performance of any two algorithms is **exactly the same**.

The Data-Bias Spectrum

Complete Inductive Bias
(Hardcoded System)

Zero Inductive Bias
(Generic Learner)

The Data-Bias Spectrum

Expert Systems

Complete Inductive Bias (Hardcoded System) — Zero Inductive Bias (Generic Learner)

Noam Chomsky: The ability to learn grammars is hard-wired into the brain. It is not possible to "learn" linguistic ability — rather, we are born with it.

The Data-Bias Spectrum

Expert Systems | **No Free Lunch**

Complete Inductive Bias
(Hardcoded System)

Zero Inductive Bias
(Generic Learner)

Noam Chomsky: The ability to learn grammars is hard-wired into the brain. It is not possible to "learn" linguistic ability — rather, we are born with it.

Geoff Hinton: There exists some "universal" learning algorithm that can learn anything: language, vision, speech, etc. The brain is based on it, and we're working on uncovering it.

The Data-Bias Spectrum

Expert Systems **No Free Lunch**

Complete Inductive Bias (Hardcoded System) — Zero Inductive Bias (Generic Learner)

<— More Assumptions **More Data Required —>**

Noam Chomsky: The ability to learn grammars is hard-wired into the brain. It is not possible to "learn" linguistic ability — rather, we are born with it.

Geoff Hinton: There exists some "universal" learning algorithm that can learn anything: language, vision, speech, etc. The brain is based on it, and we're working on uncovering it.

The Data-Bias Spectrum

Expert Systems | Linear Models, Simple Trees | **No Free Lunch**

Complete Inductive Bias (Hardcoded System) — Zero Inductive Bias (Generic Learner)

<— **More Assumptions** | **More Data Required** —>

Noam Chomsky: The ability to learn grammars is hard-wired into the brain. It is not possible to "learn" linguistic ability — rather, we are born with it.

Geoff Hinton: There exists some "universal" learning algorithm that can learn anything: language, vision, speech, etc. The brain is based on it, and we're working on uncovering it.

The Data-Bias Spectrum

Expert Systems	Linear Models, Simple Trees	Kernel Methods, Random Forests	**No Free Lunch**
Complete Inductive Bias (Hardcoded System)			Zero Inductive Bias (Generic Learner)

<— **More Assumptions** **More Data Required** —>

Noam Chomsky: The ability to learn grammars is hard-wired into the brain. It is not possible to "learn" linguistic ability — rather, we are born with it.

Geoff Hinton: There exists some "universal" learning algorithm that can learn anything: language, vision, speech, etc. The brain is based on it, and we're working on uncovering it.

The Data-Bias Spectrum

Expert Systems	Linear Models, Simple Trees	Kernel Methods, Random Forests	Neural Networks	**No Free Lunch**

Complete Inductive Bias (Hardcoded System) — Zero Inductive Bias (Generic Learner)

<— More Assumptions **More Data Required —>**

Noam Chomsky: The ability to learn grammars is hard-wired into the brain. It is not possible to "learn" linguistic ability — rather, we are born with it.

Geoff Hinton: There exists some "universal" learning algorithm that can learn anything: language, vision, speech, etc. The brain is based on it, and we're working on uncovering it.

The Takeaway

Theory of Learning

The Takeaway

Theory of Learning

Formalized learning in the context of **PAC** (Porbably Approximately Correct) Model

The Takeaway

Theory of Learning

Formalized learning in the context of **PAC** (Porbably Approximately Correct) Model

Minimizing trainig error $\rightarrow$ Minimized test error, but with sufficient trainig data

The Takeaway

Theory of Learning

Formalized learning in the context of **PAC** (Porbably Approximately Correct) Model

Minimizing trainig error $\rightarrow$ Minimized test error, but with sufficient trainig data

Computational Barrier: How to **efficiently** minimize training error?

The Takeaway

Theory of Learning

Formalized learning in the context of **PAC** (Porbably Approximately Correct) Model

Minimizing trainig error $\rightarrow$ Minimized test error, but with sufficient trainig data

Computational Barrier: How to **efficiently** minimize training error?

Occam's Razor: Higher confidence in the generalization of "simpler models"

The Takeaway

Theory of Learning

Formalized learning in the context of **PAC** (Porbably Approximately Correct) Model

Minimizing trainig error $\rightarrow$ Minimized test error, but with sufficient trainig data

Computational Barrier: How to **efficiently** minimize training error?

Occam's Razor: Higher confidence in the generalization of "simpler models"

No Free Lunch: Learning impossible without assumptions on reality

The Takeaway

Theory of Learning

Formalized learning in the context of **PAC** (Porbably Approximately Correct) Model

Minimizing trainig error $\rightarrow$ Minimized test error, but with sufficient trainig data

Computational Barrier: How to **efficiently** minimize training error?

Occam's Razor: Higher confidence in the generalization of "simpler models"

No Free Lunch: Learning impossible without assumptions on reality

For Practitioners of Learning

The Takeaway

Theory of Learning

Formalized learning in the context of **PAC** (Porbably Approximately Correct) Model

Minimizing trainig error $\rightarrow$ Minimized test error, but with sufficient trainig data

Computational Barrier: How to **efficiently** minimize training error?

Occam's Razor: Higher confidence in the generalization of "simpler models"

No Free Lunch: Learning impossible without assumptions on reality

For Practitioners of Learning

No point in finding a **"master algorithm"**

The Takeaway

Theory of Learning

Formalized learning in the context of **PAC** (Porbably Approximately Correct) Model

Minimizing trainig error $\rightarrow$ Minimized test error, but with sufficient trainig data

Computational Barrier: How to **efficiently** minimize training error?

Occam's Razor: Higher confidence in the generalization of "simpler models"

No Free Lunch: Learning impossible without assumptions on reality

For Practitioners of Learning

No point in finding a **"master algorithm"**

Understand the problem well enough to **choose** an algorithm whose inductive bias matches the underlying structure of the data.

Questions?

The Linear Models, born of light,
The branching Trees of wood and might,
The Neural Networks, deep as night,
Each claims its rule, and thinks it right.

But no single Algo to rule them all,
Where one triumphs, one must fall.
No master key for every door,
No champion on every shore.
Quest is not to find the One,
Hone your bias, 'til the work is done.

Questions?

The Linear Models, born of light,
The branching Trees of wood and might,
The Neural Networks, deep as night,
Each claims its rule, and thinks it right.

But no single Algo to rule them all,
Where one triumphs, one must fall.
No master key for every door,
No champion on every shore.
Quest is not to find the One,
Hone your bias, 'til the work is done.

References

An excellent introductory course on Theory of ML: TTIC 31250 by Avrim Blum

THE textbook of ML: Understanding Machine Learning: From Theory to Algorithms by Shai Shalev-Shwartz and Shai Ben-David