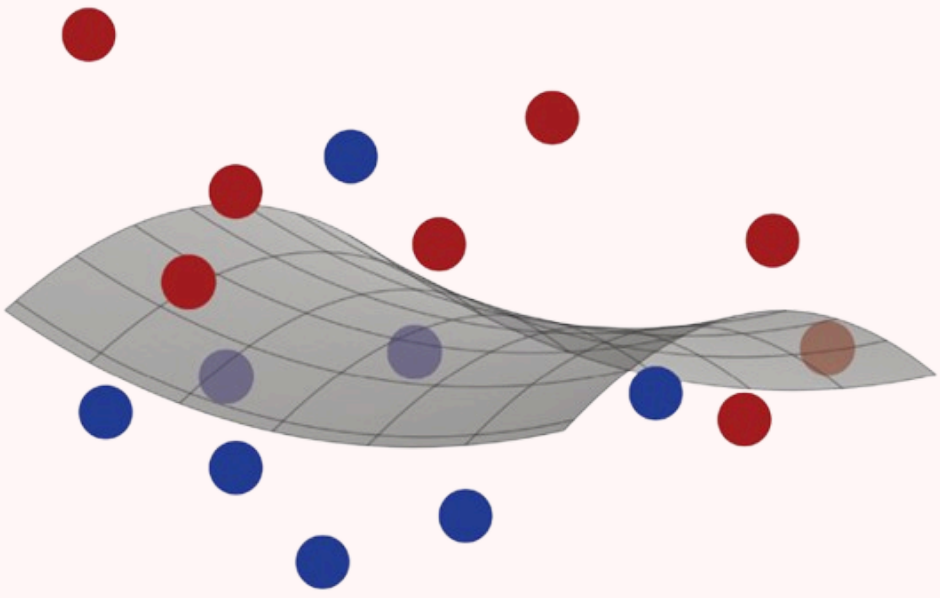
Foundations of Machine Learning

DAY - 8

The PAC Learning Model



What is PAC Learning?

Created By: **Birva Dave**
medium.com/@birva1809
github.com/Birva1809
linkedin.com/in/birva-dave
birvadave1809@gmail.com

--> When we design algorithms that learn from data, several fundamental questions naturally arise:

- What kinds of problems can these algorithms solve efficiently?
- Which ones are inherently difficult or even impossible to learn?
- How many examples do we really need to train a model well?
- And crucially, is there a general framework that helps us understand learning in a rigorous way?

The Probably Approximately Correct (PAC) learning framework gives us a powerful way to answer these questions. PAC learning formalizes what it means for a concept (or function) to be learnable by quantifying how many sample points are needed to reach an approximately correct solution, with high confidence. This framework also considers the computational resources required — like time and space — to represent and process concepts. PAC learning provides a rigorous way to characterize when and how efficiently a concept class can be learned from data, based on:

- The number of training samples needed (sample complexity),
- The computational cost (time and space complexity) of the learning algorithm, and
- The quality of the learned hypothesis measured in terms of accuracy and confidence.

The PAC model is foundational in theoretical machine learning because it connects sample size, computational resources, and learning guarantees into a unified framework.

The PAC Learning Model

Created By: **Birva Dave**
medium.com/@birva1809
github.com/Birva1809
linkedin.com/in/birva-dave
birvadave1809@gmail.com

To begin, let's establish the basic terminology and notation we'll use throughout.

- Input space (X): This is the set of all possible examples or instances the learner might encounter. Sometimes called the "instance space."
- Label space (Y): The set of possible labels or target values. For now, we focus on the simple binary classification setting, where Y = {0, 1}.
- Concept (c): A function mapping inputs to labels, c: X → Y. Since Y = {0, 1}, a concept can be viewed as a subset of X where the function outputs 1 (positive examples). For instance, if the concept is "points inside a triangle," then c is the indicator function that returns 1 for points inside that triangle and 0 otherwise.
- Concept class (C): A set of possible concepts we want to learn from. For example, C might be all possible triangles in the plane.
- Hypothesis set (H): The set of concepts (hypotheses) the learner considers. Note that H might or might not contain the true target concept c.

We assume that examples are drawn independently and identically distributed (i.i.d.) from some unknown distribution D over X. The learner receives a sample $S = \{(x_1, c(x_1)), \ldots, (x_m, c(x_m))\}$ of m labeled examples drawn according to D.

The goal is to find a hypothesis $h \in H$ that approximates the target concept c well—not just on the training sample S, but also on future unseen examples drawn from D.

The PAC Learning Model

Created By: **Birva Dave**
medium.com/@birva1809
github.com/Birva1809
linkedin.com/in/birva-dave
birvadave1809@gmail.com

Error Metrics in PAC Learning

To evaluate how good a hypothesis h is, we define two kinds of errors:

1. **Generalization error R(h):** This is the probability that h misclassifies a random example drawn from D

$$R(h) = \Pr_{x \sim D}[h(x) \neq c(x)]$$

   Since the true distribution D and target concept c are unknown, the learner cannot compute R(h) directly.

2. **Empirical error R_S(h):** This is the fraction of errors h makes on the training sample S:

$$R_S(h) = \frac{1}{m} \sum_{i=1}^{m} \mathbf{1}[h(x_i) \neq c(x_i)]$$

   While R_S(h) is accessible, it only approximates R(h). Intuitively, if m is large enough and S is representative, then R_S(h) should be close to R(h).

Indeed, due to the i.i.d. assumption, the expected empirical error equals the generalization error:

$$\mathbb{E}_S[R_S(h)] = R(h)$$

The PAC Learning Model

Created By: **Birva Dave**
medium.com/@birva1809
github.com/Birva1809
linkedin.com/in/birva-dave
birvadave1809@gmail.com

What Does PAC Learning Mean?

Now we come to the heart of the matter.

Definition (PAC Learnability): A concept class C is PAC-learnable if there exists an algorithm A and a polynomial function poly(·) such that for any desired accuracy ε > 0 and confidence δ > 0, for all distributions D over X, and all target concepts c ∈ C, the following holds:

- Given a sample size

$$m \geq \text{poly}\left(\frac{1}{\epsilon}, \frac{1}{\delta}, n, \text{size}(c)\right)$$

  - (where n relates to the size or dimension of the input and size(c) is the complexity of representing c),

- The algorithm A, after observing m labeled examples drawn i.i.d. from D, returns a hypothesis $h_S \in H$ such that

$$\Pr_{S \sim D^m}[R(h_S) > \epsilon] < \delta$$

  - In other words, with probability at least 1 − δ, the hypothesis h_S has error no greater than ε.

If A also runs in time polynomial in these parameters, then C is said to be efficiently PAC-learnable.

This definition formalizes the intuition that we want learning algorithms to produce hypotheses that are probably (with high confidence) approximately correct (with small error), hence the name PAC.

The PAC Learning Model

Created By: **Birva Dave**
medium.com/@birva1809
github.com/Birva1809
linkedin.com/in/birva-dave
birvadave1809@gmail.com

Key Features of the PAC Framework

- **Distribution-free:** PAC learning makes no assumptions about the unknown distribution D from which examples are drawn. This is a very strong and desirable property, as it means learning guarantees hold universally.
- **Training and testing under the same distribution:** The training samples and future test points are assumed to come from the same distribution D. This assumption is necessary for generalization; without it, learning is generally impossible.
- **Focus on concept classes, not individual concepts:** PAC learnability is a property of a whole class C, not just one fixed target concept. The learner knows C, but the specific target concept $c \in C$ is unknown.

Illustrative Example: Learning Axis-Aligned Rectangles

- To ground these abstract definitions, let's look at a concrete example.
- Suppose the input space is the 2D plane $\mathbb{R}^2$, and the concept class C consists of all axis-aligned rectangles. Each concept $c \in C$ corresponds to a particular rectangle in the plane: points inside the rectangle are labeled 1, points outside are labeled 0.
- The learning problem is: Given labeled examples from an unknown target rectangle R, can we find a hypothesis rectangle $\hat{R}$ that approximates R well?
- Here's a simple PAC learning algorithm A for this problem:
  - Given the sample S, find the tightest axis-aligned rectangle $\hat{R}$ that contains all points labeled 1 in S.
- Intuitively, since $\hat{R}$ must include all positive points, it won't label any positive point incorrectly (no false negatives). The only possible errors come from points inside R but outside $\hat{R}$ — the false positives.

The PAC Learning Model

Created By: **Birva Dave**
medium.com/@birva1809
github.com/Birva1809
linkedin.com/in/birva-dave
birvadave1809@gmail.com

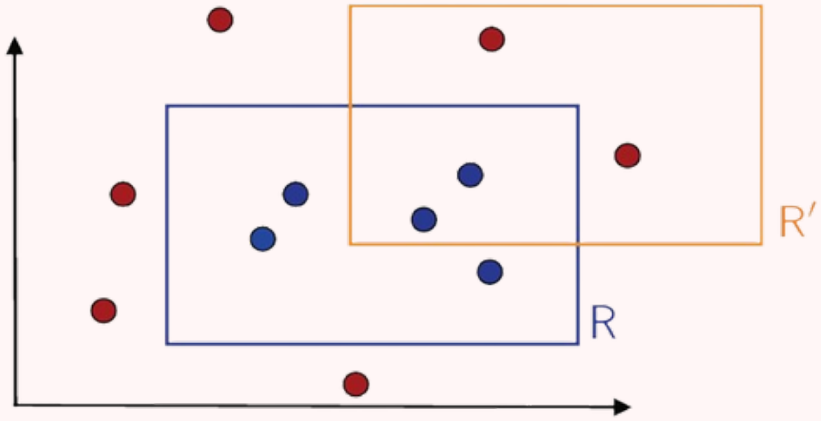


Figure: Target concept R and possible hypothesis R. Circles represent training instances. A blue circle is a point labeled with 1, since it falls within the rectangle R. Others are red and labeled with 0

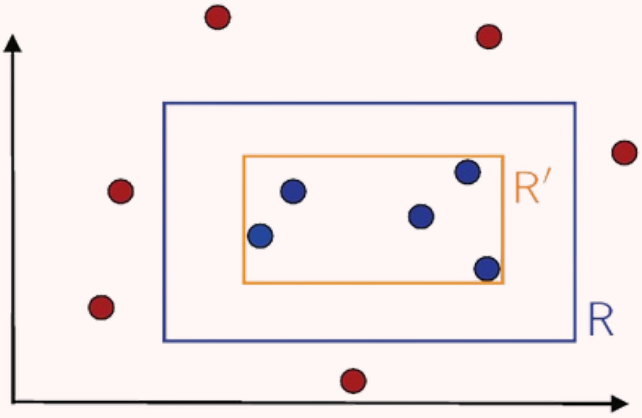


Figure: Illustration of the hypothesis R= RS returned by the algorithm.

- By carefully analyzing the geometry and probability mass of the error regions (the difference between R and $\hat{R}$), it can be shown that for any accuracy ε and confidence δ, if the sample size m is at least proportional to $(1/\varepsilon) \log (1/\delta)$, the error of $\hat{R}$ will be less than ε with probability at least $1 - \delta$.
- This proves that the concept class of axis-aligned rectangles is PAC-learnable — and the sample complexity grows reasonably with the desired accuracy and confidence.

The PAC Learning Model

Created By: **Birva Dave**
medium.com/@birva1809
github.com/Birva1809
linkedin.com/in/birva-dave
birvadave1809@gmail.com

Summary

- The PAC framework provides a rigorous foundation to understand what it means for a concept class to be learnable from data, independent of distribution assumptions. It links the number of samples needed, the complexity of concepts, and the runtime of learning algorithms to guarantees on prediction accuracy.
- By studying specific examples like axis-aligned rectangles, we gain intuition for how PAC learning works and how to design algorithms that perform well with high probability, given enough data.