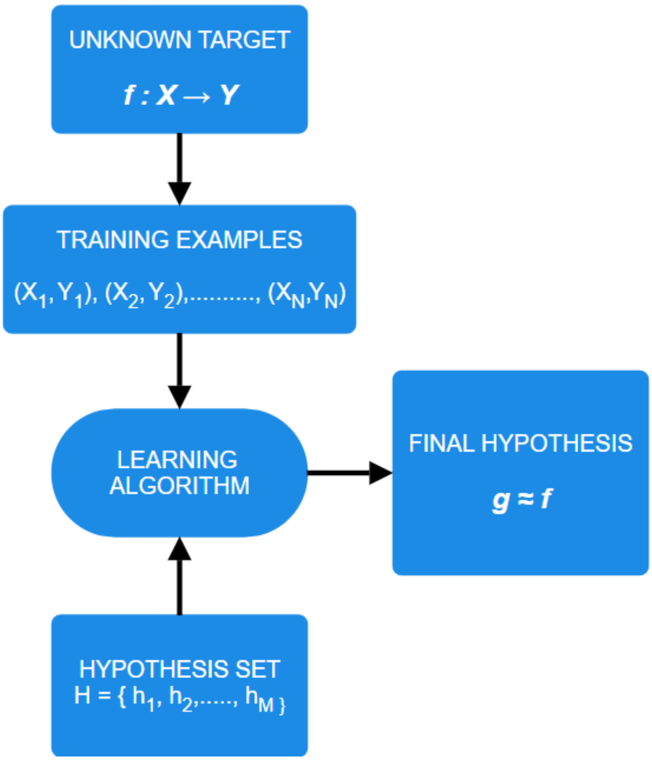
FOUNDATIONS OF ALGORITHM DESIGN AND MACHINE LEARNING

Scribenote for 03-03-2025

Mohsin Parmar
24BM6JP33

The Learning Framework :



This diagram represents a fundamental process in machine learning. Here's a breakdown of its components:

1. **Unknown Target Function f : X → Y :**This represents the true function mapping input X to output Y. However, this function is unknown and needs to be approximated.
2. **Training Examples $(X_1, Y_1), (X_2, Y_2), \ldots\ldots\ldots, (X_N, Y_N)$:** These are the data points collected, consisting of input-output pairs. The goal is to learn a function that generalizes well from these examples.
3. **Hypothesis Set $H = \{ h_1, h_2, \ldots\ldots, h_M \}$:** This represents the set of possible functions (hypotheses) the learning algorithm can consider. The choice of hypothesis set depends on the model used (e.g., linear regression, decision trees, neural networks).
4. **Learning Algorithm :**The core of machine learning, where an algorithm selects the best hypothesis from H using training examples. This selection is often based on minimizing a loss function.
5. **Final Hypothesis :** The output of the learning algorithm is a function ggg that serves as an approximation of the true function f. The goal is for g to generalize well to unseen data.

Understanding Generalization Through Hoeffding's Inequality

1. Introduction

Machine learning models aim to generalize well from training data to unseen examples. To understand this concept, we examine an analogy involving marble selection from a bin. This analogy helps us explore the relationship between training error and true error using Hoeffding’s inequality, a fundamental result in probability theory.

The key question we aim to answer is:

> ***How well does the performance of a model on training data approximate its performance on unseen data?***

Through this experiment, we establish that with a sufficiently large sample size, training error reliably estimates true error with high probability.

2. Experimental Setup: The Marble Selection Analogy

We consider a bin filled with marbles of two types:

- **Shaded marbles** (representing incorrect predictions or misclassifications by a hypothesis).
- **Unshaded marbles** (representing correct predictions).

A learner does not know the total proportion of shaded marbles in the bin but can draw a random sample of marbles to estimate it. The goal is to determine how well this sample-based estimate approximates the true fraction of shaded marbles in the entire bin.

Key elements of the analogy:

- The **bin** represents the overall data distribution.
- Each **marble** represents a data point.
- **Picking marbles** corresponds to drawing a training dataset.
- The **fraction of shaded marbles in the sample** represents the observed training error E_{in}.
- The **fraction of shaded marbles in the bin** represents the true generalization error E_{out}.

The fundamental question becomes:

> ***If we take a random sample of marbles, how close is the fraction of shaded marbles in our sample to the true fraction in the bin?***

3. Theoretical Foundation: Hoeffding’s Inequality

Hoeffding’s inequality is a fundamental result in probability theory that provides a **statistical guarantee** about how well training error E_{in} approximates the true generalization error E_{out}. It is expressed as:

$$\boxed{P(|\, Ein - Eout \,|> \epsilon) \leq 2e^{-2\epsilon^2 N}}$$

where:

- E_{in} = Error on the training dataset (observed error)

- E_{out} = Error on the entire population or unseen data (true error)
- ϵ = Tolerance threshold (how much deviation we allow between E_{in} and E_{out})
- N = Number of samples (size of the training dataset)

What Does This Inequality Tell Us?

- **With a large enough sample size (N), the probability that E_{in} deviates significantly from E_{out} becomes very small.**
  - This means that if we train a model on a large enough dataset, its performance on the test data (generalization ability) will be close to its performance on the training data.
- **Exponentially decreasing bound:**
  - The probability of a large deviation decreases exponentially with N.
  - For example, if we double the sample size, the probability of bad generalization shrinks exponentially.
  - This justifies why **having more data improves model generalization**.
- **Independent of the learning algorithm:**
  - Hoeffding's inequality applies to any learning algorithm and any hypothesis class.
  - It **only depends on the number of samples** and does not assume anything about the data distribution.
- **Key Assumption: The samples are independent and drawn from the same distribution (i.i.d. data).**
  - If the training data is biased or not representative of the real-world distribution, Hoeffding's bound might not hold.

Who Defines ϵ?

1. **The User (or Analyst)**:
   - In a learning scenario, ϵ is chosen based on how precise we want our empirical estimate E_{in} to be.
   - A **smaller** ϵ means we demand a very tight generalization bound, which requires a **larger** N.
   - A **larger** ϵ allows more deviation but needs **fewer** samples to satisfy the inequality.
2. **The Problem's Requirements**:
   - If the problem requires a highly accurate model, a small ϵ is necessary.
   - If approximate learning is acceptable, a larger ϵ might be used.
3. **Theoretical Analysis**:
   - In PAC (Probably Approximately Correct) Learning, ϵ is often chosen to bound the generalization error within an acceptable range (e.g., "we want the true error to be within 5% of the empirical error with high probability").
   - A typical choice in statistical learning is to set ϵ based on confidence requirements (e.g., setting $\delta = 2e^{-2\epsilon^2 N}$and solving for ϵ in terms of δ and N).

$$\delta = 2e^{-2\epsilon^2 N}$$

Taking natural log on both sides,

$ln(\delta)$ = $ln(2)$ -2ϵN$ln(e)$

$$\therefore \epsilon = \sqrt{\frac{ln(\frac{2}{\delta})}{2N}}$$

Interpretation

- This formula gives the value of ϵ for a given confidence level δ\deltaδ and sample size N.
- If you want a high confidence (i.e., very small δ), then ln(2/δ) grows, making ϵ larger.
- If you increase the number of samples N, then ϵ decreases, meaning that E_{in} and E_{out} are likely to be closer.

Example

Let’s say:

- N=1000 (1000 training examples)
- Confidence 1−δ=0.95 (i.e. δ=0.05)

Compute ϵ:

$$\epsilon = \sqrt{\frac{ln(\frac{2}{0.05})}{2(1000)}}$$

$$\epsilon = 0.043$$

This means with **95% confidence**, the difference between empirical and true error is at most **4.3%**.

4. Application in the Experimental Setup

In the marble selection problem, E_{in} represents the fraction of red marbles in the random sample, while E_{out} represents the true fraction of red marbles in the entire jar. The goal is to ensure that E_{in} is a good estimate of E_{out} with high confidence.

To control the probability of deviation, we set a confidence level 1−δ, where δ is the probability of failure. Setting

$\epsilon = \sqrt{\frac{ln(\frac{2}{\delta})}{2N}}$ and rearranging the terms we get,

$$N = \frac{ln(2/\delta)}{2\epsilon^2}$$

For instance, if we want to estimate E_{out} within ϵ=0.05 with 95% confidence (δ=0.05), we calculate:

$$N \;=\; \frac{ln(2/0.05)}{2(0.05)^2} \approx 738$$

Thus, we need at least **738 samples** to ensure that E_{in} is within 5% of E_{out} with 95% confidence.

If we reduce ϵ to 1% (more precise estimation), N increases significantly to around 18,400. Similarly, increasing the confidence level (reducing δ) also increases N. This shows the trade-off between accuracy, confidence, and sample size when estimating an unknown probability.

However, in practice, we don't evaluate just one hypothesis—we search through **multiple** hypotheses and select the best one based on E_{in}. This selection introduces an additional risk:

"the more hypotheses we check, the higher the chance of selecting one that ***fits the sample well but generalizes poorly****."*

So how do we take this into account?

Modified Hoeffding's Inequality with M Hypotheses :

When we choose a hypothesis by minimizing E_{in}, we must consider the worst case over all M hypotheses. Using the **union bound**, the probability that at least one hypothesis deviates significantly from its true error is:

Union Bound :

The union bound states that for any finite set of events $A_1,A_2,A_3,.....A_M$ the probability that at least one of them occurs is at most the sum of their individual probabilities:

$$P(A_1 \cup\ A_2\ \cup\ A_3\ \cup........\cup\ A_M)\ \le\ \sum_{i=1}^{M}\quad P(A_i)$$

We can use this property to prove modified hoeffding's inequality to prove M hypotheses, that is :

$$\boxed{P(h\ \in H\ such\ that\ \mid Ein(h) - Eout(h)\ \mid > \epsilon) \le 2Me^{-2\epsilon^2 N}}$$

This shows that increasing the number of hypotheses M **increases the probability of picking a misleading hypothesis**, i.e., one where E_{in} is low but E_{out} is high.

Choosing h such that $E_{in}(h)$=0 :

"*If we select a hypothesis where $E_{in}(h) = 0$, it means the hypothesis perfectly fits the training data. But does that guarantee $E_{out}(h) = 0$?"*

Not necessarily!

From the modified bound:

$$P(|Eout(h)\ |> \epsilon) \leq 2Me^{-2\epsilon^2 N}$$

For small N, even if E_{in}=0, there's a **nonzero chance** that E_{out} is still large. This happens in **overfitting**, where the hypothesis memorizes the training set but fails to generalize.

Key takeaways from this :

1. More hypotheses M increase the risk of choosing a misleading hypothesis.
2. Even if E_{in} = 0, E_{out} may not be small due to limited data.
3. To control generalization error, we should balance the number of hypotheses and sample size N.
4. Regularization and model complexity control help mitigate overfitting.

Coin Flip Experiment for understanding dilation of bounds:

In probability and machine learning, dilation of bounds refers to the widening of confidence intervals or probability bounds when multiple hypotheses are tested. This phenomenon is crucial in understanding why testing many models increases the risk of false positives or overfitting.

Case 1 :

Suppose you flip a fair coin (P(Heads)=0.5) N times(for eg, N = 10)

We ask : *What is the probability that all heads occur?*

$$P(all\ heads) = (0.5)^{10} \approx\ 0.00098$$

Case 2 :

Instead of flipping one coin, we flip 1,000 different coins.

Each coin is flipped N times (eg N = 10).

We ask: *What is the probability that at least one of these 1,000 coins lands all heads?*

The probability of **one coin** getting **all heads** in 10 flips:

$$P(all\ heads) = (0.5)^{10} \approx\ 0.00098$$

Now, for **1,000 coins**, the probability that **at least one** of them lands all heads:

$$P(at\ least\ one\ all - heads\ ) = 1 - (1 - (0.5)^{10})^{1000} \approx 0.625$$

So, there is a **62.5% chance** that at least one of the 1,000 coins will land all heads **just by random chance**.

Key Takeaway:

- With a single hypothesis (one coin), the probability of an extreme event (all heads) is very low.
- With multiple hypotheses (1,000 coins), the probability of an extreme event happening in at least one of them is much higher.
- This demonstrates why testing many hypotheses in machine learning increases the risk of overfitting.

Aspect	First Experiment (Hoeffding's Bound)	Second Experiment (Hypothesis Selection)
What is tested?	How well does a sample estimate the true probability?	What is the probability of seeing an extreme outcome?
Key finding	More samples (NNN) → better approximation	More hypotheses → higher risk of finding something that fits by chance
Machine learning link	More training data improves generalization	Testing too many hypotheses can lead to overfitting
Example in ML	A well-trained model generalizes well to test data	A model with too many parameters might fit training noise